mathematics

Article

A Novel Deep Reinforcement Learning Based Framework for Gait Adjustment

Ang Li [1,2], Jianping Chen [2,3,*], Qiming Fu [1,2,*], Hongjie Wu [1,2], Yunzhe Wang [1,2] and You Lu [1,2]

[1] School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China
[2] Jiangsu Province Key Laboratory of Intelligent Building Energy Efficiency, Suzhou University of Science and Technology, Suzhou 215009, China
[3] School of Architecture and Urban Planning, Suzhou University of Science and Technology, Suzhou 215009, China
* Correspondence: alan@usts.edu.cn (J.C.); fqm_1@mail.usts.edu.cn (Q.F.)

Abstract: Nowadays, millions of patients suffer from physical disabilities, including lower-limb disabilities. Researchers have adopted a variety of physical therapies based on the lower-limb exoskeleton, in which it is difficult to adjust equipment parameters in a timely fashion. Therefore, intelligent control methods, for example, deep reinforcement learning (DRL), have been used to control the medical equipment used in human gait adjustment. In this study, based on the key-value attention mechanism, we reconstructed the agent's observations by capturing the self-dependent feature information for decision-making in regard to each state sampled from the replay buffer. Moreover, based on Softmax Deep Double Deterministic policy gradients (SD3), a novel DRL-based framework, key-value attention-based SD3 (AT_SD3), has been proposed for gait adjustment. We demonstrated the effectiveness of our proposed framework in gait adjustment by comparing different gait trajectories, including the desired trajectory and the adjusted trajectory. The results showed that the simulated trajectories were closer to the desired trajectory, both in their shapes and values. Furthermore, by comparing the results of our experiments with those of other state-of-the-art methods, the results proved that our proposed framework exhibited better performance.

Keywords: deep reinforcement learning; attention mechanism; state reconstruction; gait adjustment

MSC: 03D80; 68Q30

check for updates

Citation: Li, A.; Chen, J.; Fu, Q.; Wu, H.; Wang, Y.; Lu, Y. A Novel Deep Reinforcement Learning Based Framework for Gait Adjustment. *Mathematics* **2023**, *11*, 178. https://doi.org/10.3390/math11010178

Academic Editor: Jianping Gou, Weihua Ou, Shaoning Zeng and Lan Du

Received: 18 November 2022
Revised: 15 December 2022
Accepted: 21 December 2022
Published: 29 December 2022

1. Introduction

Regaining the ability to walk is a primary goal of recovery for stroke patients. However, patients often experience restrictions on their daily communication and freedom of movement. Therefore, gait rehabilitation is urgently needed for these patients [1]. In the fields of gait rehabilitation and walking assistance, most lower-limb exoskeletons are developed for assisting paraplegic patients with disabilities of both of their legs. Through gait rehabilitation, we can achieve the goal of helping patients with mobility disorders in the rehabilitation of their musculoskeletal strength, motor control, and gait.

In traditional rehabilitation therapies, intensive labor is involved, and physical therapists have to provide patients with highly repetitive training that is usually inefficient and time-consuming [2]. The inherent shortcomings of these therapies include their failure to autonomously adapt to the user's changing needs, as well as the lack of sensory feedback that they provide to the user regarding the states of the limb and of the device. Compared to traditional physical therapies, exoskeleton-assisted rehabilitation has the advantages of reducing the work of therapists, and it is more convenient to use for quantitatively assessing the patient's level of recovery by measuring force and movement patterns [3].

To date, studies on exoskeleton control methods have achieved remarkable results. Mendoza-Crespo, Rafael et al. [4] developed and presented a method to acquire and saliently analyze subject-specific gait data, with the subject donning a passive lower-limb exoskeleton. In [5], a trajectory tracking controller based on the boundary layer augmented sliding control (BASMC) law was implemented to guide the subject's limbs along physiological gait trajectories. However, patients are normally trained to passively follow a predefined gait reference trajectory and their initiatives or motivations are usually not considered in the abovementioned methods. Therefore, adaptive control techniques and deep reinforcement learning (DRL)-based control methods have been proposed. DRL can potentially be used for exoskeleton control, and a predefined gait trajectory is not required. More importantly, interaction between the exoskeleton of the lower extremity and the patient during rehabilitation can be achieved. Thus, in this study, we focused on the control of a lower-limb exoskeleton using DRL.

2. Novelty and Contribution of the Study

In this study, in order to achieve the goal of gait rehabilitation and walking assistance, we simulated an exoskeleton based on the lower-limb musculoskeletal model used in the 2019 NeurIPS "Learning to Move–Walk Around" challenge.

Firstly, we adopted the Markov decision process (MDP) to model the gait adjustment problem, which provided an intelligent policy for the control of the exoskeleton. Secondly, in order to solve the curse of dimensionality caused by the complexity of the musculoskeletal model, we proposed a DRL-based framework named AT_SD3, which incorporated key-value attention-based state reconstruction and Softmax Deep Double Deterministic policy gradients (SD3). Based on the key-value attention mechanism, we presented a novel state reconstruction framework, in which all sampled sates are used in order to be fused proportionally with the initial observations, which enables the model to extract the self-dependent feature information of each sampled state to reconstruct an effective and interpretive state. Then, the DRL agent can select a better action in accordance with the same policy. Moreover, we used the autoencoder to extract features from the reconstructed state to solve the curse of dimensionality. Finally, we compared gait trajectories, including the desired trajectory, the unadjusted trajectory obtained in previous works, and the adjusted trajectory obtained in this work. The results showed that the adjusted trajectory was closer to the desired trajectory, in terms of its shape and value, than the unadjusted trajectory, and the performance of our proposed framework was better than that of other state-of-the-art DRL algorithms.

The related code and dataset are available at https://github.com/li0516/opensim-rl.git (accessed on 17 November 2022).

3. Related Works

3.1. Adaptive Control Techniques

Adaptive control techniques utilize dynamics models for both the user and the exoskeleton. Fatai Sado proposed a control strategy that integrated a dual unscented Kalman Filter (DUKF) for trajectory generation/the prediction of the spatio-temporal features of human walking and used an impedance-cum-supervisory controller to enable the exoskeleton to follow this trajectory in order to synchronize human walking [6]. In order to improve the control performance, the authors introduced a linear quadratic regulator with integral action (LQRi) and an unknown input observer (UIO) to compensate for disturbances [7]. In [8], an adaptive oscillator method named the amplitude omega adaptive iscillator (AωAO), comprising both low-level classifiers (to detect activities) and high-level classifiers to detect transitions between activities, was proposed to provide bilateral hip assistance for human locomotion. Sado, F. et al. [9] proposed a exoskeleton controller, with the design of a low-level linear quadratic gaussian (LQG) torque controller, a middle-level user-input torque estimator based on the use of a dual extended Kalman filter (EKF), and a novel high-level supervisory algorithm for the detection of movement and the synchronization of the exoskeleton with the user.

3.2. DRL-Based Control Methods

As one of learning-based control methods, Deep Reinforcement Learning (DRL), has been used in lower limb exoskeletons control. A human–robot interactive control, designed with Sigmoid function and the reinforcement learning algorithm, was proposed to govern the assistance provided by a lower limb exoskeleton robot to patients in the gait rehabilitation training [10]. In [11], Zhang, Y. et al. proposed a reinforcement-learning-based impedance controller, which actively reshapes the stiffness of the force-field to the subject's performance. In [12], an optimal adaptive compliance control was proposed for a Robotic walk assist device, where the reinforcement-learning-based strategy is a completely dynamic-model-free scheme, and this scheme employed joint position and velocity feedback as well as sensed joint torque (applied by user during walk) for compliance control. In [13], Rose, L. et al. presented for the first time an end-to-end model-free deep reinforcement learning method for an exoskeleton that can learn to follow a desired gait pattern, while considering a user's existing gait pattern and being robust to their perturbations and interactions. Oghogho, Martin et al. [14] employed the Twin Delayed Deep Deterministic Policy Gradient (TD3) method for rapid learning of the appropriate controller's gain values and delivering personalized assistive torques by the exoskeleton to different joints to assist the wearer in a weight handling task. In [15], Kumar, V.C.V. et al. took the Proximal Policy Optimization (PPO) to develop a human locomotion policy which can imitates the human walking reference motion. Based on all these achievements above, DRL-based control is inherently both adaptive and optimal, which can adapt to uncertainty and unforeseen changes in the robot dynamics [12].

Previous studies have shown that DRL is effective in the lower limb exoskeleton control. Moreover, with the concept of strengthening the discrimination among all the similar classes using the specific weights [16], in this paper, we propose a DRL-based framework, which incorporates a novel DRL algorithm SD3 and the key-value attention mechanism. Compared with the previous DRL methods, our framework can deal with the curse of dimensionality caused by the musculoskeletal model with high degree of freedom. From this perpective, our framework can greatly improve the performance of the DRL algrithm when a reinforcement learning (RL) agent observes a high dimensional state, and more importantly, experimental results show that our proposed framework has the state-of-art performance for the gait adjustment.

4. Preliminaries

4.1. Reinforcement Learning

We usually model the reinforcement learning problem as a MDP. A MDP is a quintuple (S, A, R, P, γ), where S is the state space, A is the action space, R is the reward function, P is the transition probability distribution and γ is the discount factor. At time step t, the agent selects and executes an action $a_t \in A$ according to the policy π, which maps from the state s to the probability of an action a. Then, the environment moves to a new state $s_{t+1} \in S$, where s_{t+1} is determined from the transition probability $P(s_{t+1}|s_t, a_t)$. Simultaneously, the agent receives the immediate reward $r_{t+1} \sim R(s_t, a_t)$. The dynamic diagram of the agent interaction with the environment is shown in Figure 1.

In RL, we aim to find an optimal policy which maximizes the return $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$. To achieve this, we evaluate the policy π by estimating the value function, including state-value function V_π and action-value function Q_π. Here, the state-value function V_π is the expected return G_t when starting in state s and following policy π thereafter:

$$V_\pi(s) = E_\pi[G_t \mid s_t = s], \tag{1}$$

where the $E_\pi[\cdot]$ denotes the expected value of the return G_t given that the agent follows policy π. The action-value function, also called Q-value, $Q_\pi(s,a)$, represents the expected return G_t after taking an action a in state s and thereafter following policy π:

$$Q_\pi(s,a) = E_\pi[G_t \mid s_t = s, a_t = a]. \tag{2}$$

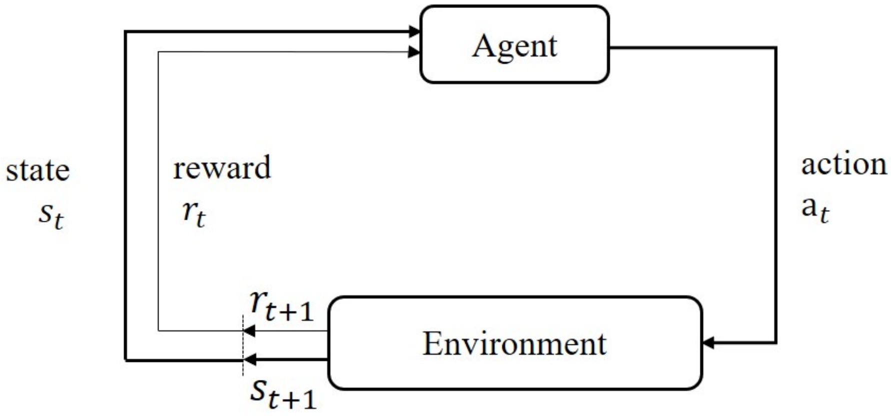


Figure 1. The interaction between the agent and the environment in RL.

Thereafter, the optimal policy π_* can be obtained by maximizing the state-value function or the action-value function, denoted V_* and Q_*, respectively. These two functions can be defined as follows:

$$V_*(s) = \max_\pi V_\pi(s), \tag{3}$$

$$Q_*(s,a) = \max_\pi Q_\pi(s,a) = E\Big[R_{t+1} + \gamma \max_a Q_*(s_{t+1},a) \mid s_t = s, a_t = a\Big]. \tag{4}$$

4.2. Softmax Deep Double Deterministic Policy Gradients

DDPG algorithm is often used to solve continuous control problems [17,18]. However, one of the dominant concerns for DDPG is that it suffers from the overestimation problem caused by selecting an action with highest action-value estimates according to the critic network [19]. To reduce the adverse impact of the overestimation, double estimators were proposed for the critic in TD3 [20]. Nevertheless, another problem is the large underestimation bias caused by direct adoption of taking minimum estimation of action-value from the two critics in TD3 [21].

To tackle this problem, Pan, L. [19] proposed a method, called SD3, which combines the softmax operator with the estimation of the action-value based on double critic estimators. In SD3, double actor networks and critic networks are built to select multiple actions and evaluate the corresponding action-values, respectively. To be specific, alternative actions will be selected via different actor networks, and then the minimum action-value can be obtained by calculating and comparing the action value functions of the corresponding actions evaluated by two critic networks:

$$\hat{Q}_{i=1,2}(s',a') = \min\big(Q_{i=1}(s',a';\theta^-_{i=1}), Q_{i=2}(s',a';\theta^-_{i=2})\big). \tag{5}$$

Thereafter, the minimum Q-value will be induced by the softmax operator in expectation by the importance sampling, and the specific definition of the softmax Q-value is as follows:

$$\text{softmax}_\beta\big(Q(s',\cdot;\theta^-)\big) = \frac{E_{\hat{a}'\sim p}\left[\frac{\exp(\beta Q(s',\hat{a}';\theta^-))Q(s',\hat{a}';\theta^-)}{p(\hat{a}')}\right]}{E_{\hat{a}'\sim p}\left[\frac{\exp(\beta Q(s',\hat{a}';\theta^-))}{p(\hat{a}')}\right]}, \tag{6}$$

where β is the parameter of the softmax operator, and the implication of $p(\hat{a}')$ is the probability density function of the Gaussian distribution for the importance sampling. The $E_{\hat{a}'\sim p}[\cdot]$ denotes the expected value of a random variable given that $\hat{a}'$ are sampled from the Gaussian distribution $p(\hat{a}')$. And $\hat{a}'$ is the action with additional noises for exploration, which are sampled from the Gaussian distribution $p(\hat{a}')$. Finally, the softmax Q-value can be obtained to calculate the target value:

$$y = r + \gamma(1-d)\,\mathrm{softmax}_{\beta}\big(Q(s', \cdot; \theta^{-})\big). \tag{7}$$

4.3. Key-Value Attention Mechanism

Attention mechanism [22] in neural networks is introduced to focus on the information which is critical to the current task among the numerous input information. Therefore, the attention mechanism is often used to solve the problem of information overload and improve the efficiency and accuracy of task processing.

However, it is not suitable for some specific problems. So, Vaswani, A. et al. [22] introduced the key-value attention mechanism, which uses the format of a key-value pair to represent input information. The key is used to calculate the attention distribution α_i, and the value is used to calculate aggregate information. As shown in Figure 2, $(K, V) = [(k_1, v_1), \ldots, (k_n, v_n)]$ is used to represent N sets of the input information and the vector q is used to represent the query vector for a given task. Then, the attention function can be defined as follows:

$$\mathrm{att}(X, q) = \sum_{i=1}^{N} \alpha_i x_i = \sum_{i=1}^{N} \frac{\exp(s(k_i, q))}{\sum_j \exp(s(k_i, q))} v_i, \tag{8}$$

where s is the attention evaluation function, and x_i is equal to v_i which is used to represent the value of N sets of the input information. Finally, a weighted average of the input information v_i, the final output a , can be achieved according to the distribution α_i, which is computed based on the function s.

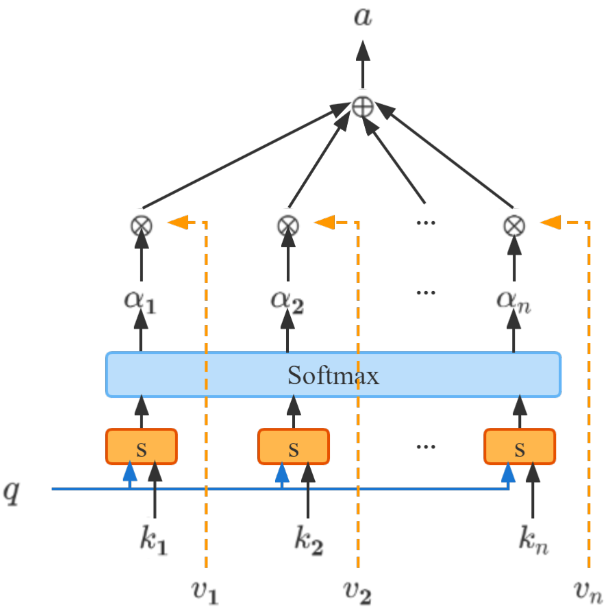


Figure 2. The key-value attention mechanism.

4.4. Parameter Space Noise for Exploration

Traditional RL methods increase exploration by adding noise, for example the Gaussian noise, to the output of the actor network. That is to say, the noise added to the actor network is independent of the state s_t, in other words, state-independent exploration. Hence, even for the same state s_t, a different action a_t will be certainly achieved and even sometimes it has nothing to do with s_t.

Therefore, Fortunato, Meire et al. [23] and Plappert, Matthias et al. [24] proposed to add noise to the agent's parameters. They sampled from a set of policies by adding the noise sampled from the Gaussian noise to the current policy $\pi(s_t)$, and in this case, the same action $a_t = \hat{\pi}(s_t)$ can be achieved every time the same state s_t is taken as the input to the actor network.

5. Problem Modeling

In the previous work, we conducted gait simulation experiments with DRL algorithms based on the lower limb musculoskeletal model. The experimental results show that DRL algorithm is effective in gait simulation. However, sometimes during the simulation, there will be abnormal gait. In this paper, we adopt MDP to model the gait adjustment problem based on the musculoskeletal model.

5.1. The Lower Limb Musculoskeletal Model

In our work, the simulated environment used for the gait adjustment, named osim-rl, used in 2019 NeurIPS "Learning to Move–Walk Around" challenge, incorporates the lower limb musculoskeletal model and DRL to provide the accurate human movement simulation. The lower limb musculoskeletal model built in OpenSim has 8 internal degrees of freedom (4 per leg) and is actuated by 22 muscles (11 per leg). During the simulation, muscles are driven by muscle activations (the control signals that muscles produce power), and then states of the musculoskeletal model including joint angles, body location and ground reaction forces will be returned. The lower limb musculoskeletal model is shown in Figure 3. More detailed environment description can be found at the page: http://osim-rl.kidzinski.com/docs/nips2019/environment/ (accessed on 17 November 2022).

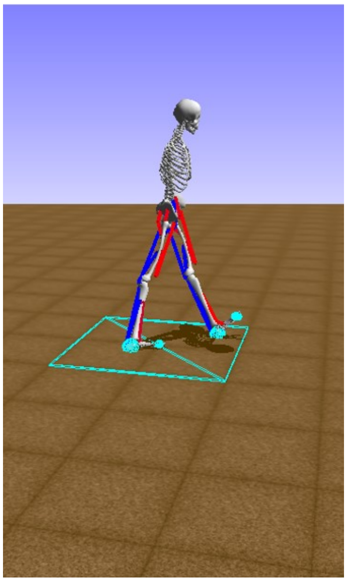


Figure 3. The lower limb musculoskeletal model.

5.2. MDP Modeling

5.2.1. State Space

The observation of the DRL agent consists of two parts: a target velocity map T and a body state S. Firstly, as shown in Figure 4, the target velocity map T is represented as a randomly generated target velocity matrix, which is a 2-dimensional target velocity vector, consisting of the target position and the current position of the model. Then, a target velocity vector can be achieved based on these positions. Secondly, the body state S is expressed by a 97-dimensional vector which consists of the pelvis state, ground reaction forces, joint angles and states of lower limb muscles. To be specific, the varibles of state space is listed in Table 1.

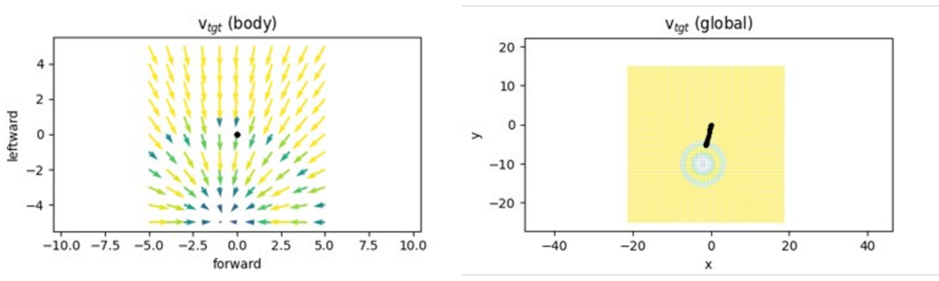


Figure 4. The target velocity map.

Table 1. State space.

	Symbols	Description
Body state S	S_p	pelvis state
	S_g	ground reaction forces
	S_j	joint angles
	S_m	muscle states
Target velocity map T	T_g	target velocity (global)
	T_b	target velocity (body)

5.2.2. Action Space

The action space $[0, 1]^{22}$ represents muscle activations of 22 muscles. Muscles responds to these activations and generate forces, and then the model will act accordingly, for example, moving forward. At the same time, states of the model change accordingly.

5.2.3. Reward Function

The DRL agent will obtain a reward $J(\pi)$. The specific definition is as follows:

$$J(\pi) = R_b + R_g, \tag{9}$$

where R_b and R_g refer to the reward for the initial gait simulation and the gait adjustment according to the desired trajectory. To be specific, R_b ensures that a basis human gait can be obtained based on the musculoskeletal. However, during the simulation, deformed gaits sometimes appeared. So R_g is designed to make up for the gait defects, which is reflected in the deviation between the simulated angle and the desired angle of each joint of the lower limb.

Firstly, the specific definition of R_b is as follows:

$$R_b = M_{alive} + M_{step}, \tag{10}$$

where M_{alive} and M_{step} refer to the model remaining standing as long as possible and moving with minimal forces according to the target velocity map, respectively. Here, M_{alive} and M_{step} are defined as follows:

$$M_{alive} = \sum_i m_{alive}, \tag{11}$$

$$M_{step} = \sum_{step_i} \Big(w_{step} m_{step} - w_{vel} c_{vel} - w_{eff} c_{eff} \Big). \tag{12}$$

In Equation (11), m_{alive} refers to the unit time of "model survival". In addition, in Equation (12), on the one hand, m_{step} is stepping reward which represents the total elapsed time-steps of "model survival" in simulation. c_{vel} and c_{eff} are the velocity and effort costs, respectively. On the other hand, w_{step}, w_{vel} and w_{eff} are weights for the stepping reward, velocity and effort costs. Another point needed to note is that w_{step} is used to avoid getting higher reward by making small steps in human gait simulation.

Secondly, R_g is designed based on the changes of the real-time angle of each joint relative to the desired trajectory, for example, approaching or even exceeding in each episode. The specific definition is as Equation (13):

$$R_g = \sum_{i=0}^{n} \left(w_h r_{i_h} + w_k r_{i_k} + w_a r_{i_a} \right), \tag{13}$$

where r_i and w_i are the reward for each of the three joints in the lower limb and the corresponding weight, respectively. The reward r_i for timestep i is defined as follows:

$$r_i = w_F F(q_i) + w_G G(q_i), \tag{14}$$

where w_F and w_G are the weights for the reward $F(q_i)$ and the penalty $G(q_i)$, respectively. Here, on one hand, the function $F(q_i)$, representing the reward for the tendency approaching the desired trajectory, is defined based on the Gaussian function:

$$F(q_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{d-\mu}{\sigma}\right)^2}, \tag{15}$$

where μ and σ represent the mean and the SD of the desired joint angle, respectively. In addition, d, the absolute value of the difference between the real-time angle q_i and the desired joint angle q_{d_i} is defined as follows:

$$d = |q_i - q_{d_i}|. \tag{16}$$

On the other hand, the function $G(q_i)$, representing the penalty for exceeding the desired trajectory, is defined as Equation (17).

$$G(q_i) = -M(y_{max}) - M(y_{min}), \tag{17}$$

where $M(\cdot)$ is defined as follows:

$$M(y) = \begin{cases} 0 & y \leq 0 \\ y & y > 0 \end{cases}, \tag{18}$$

and

$$y_{max} = q_i - q_{max}, \tag{19}$$

$$y_{min} = q_{min} - q_i, \tag{20}$$

where q_{max} and q_{min} are the maximum and the minimum joint angle, respectively.

6. Methodology

6.1. Overall Framework

As depicted in Figure 5, the overall framework for gait adjustment consists of two parts: state reconstruct and SD3. First of all, the simulated environment initialization. Secondly, we reconstruct the initial observation via extracting features from existing states based on the attention mechanism, where the states are sampled in pairs with actions from the replay buffer randomly.

In the second part, the reconstructed state is taken as the input of SD3. Then, the actor network selects an action a_i according to the observation where i refers to the serial number of the action corresponding to different actor networks, and following, the critic network evaluates the value of the state action pair $Q(s, a_i)$. Moreover, the final action a depends on the result of comparing action-values which are evaluated by two critic networks. It is worth noting that, we add noise directly to the actor network parameters for a state-dependent exploration, which ensures a dependency between the sampled state and the corresponding selected action.

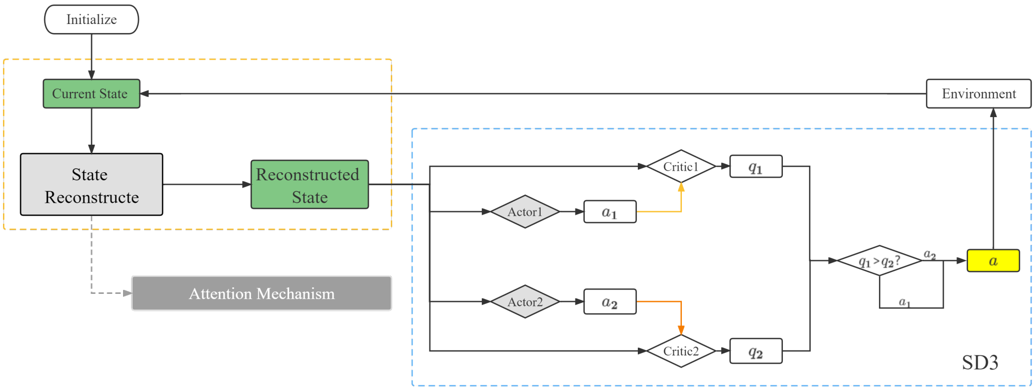


Figure 5. The overall framework for gait adjustment.

6.2. Key-Value Attention-Based State Reconstruction

In this work, the initial observation is a 339-dimensional state which consists of a 97-dimensional body state and a 242-dimensional target velocity map. Therefore, the RL agent cannot extract effective information easily, and then choose better actions due to too much redundant information in this high-dimensional observation. Moreover, in RL, the observed state s and the selected action a of an RL agent often plays a significant role for the training of RL algorithms, and the information in each state usually play an important role in the choice of the action. For example, in the case of the same policy and different states, RL agent takes different actions without active exploration. As shown in Figure 6, the actions taken to reach s_3, s_4 are shown by arrows. Although s_1 and s_2 are very close in space, they are functionally different, and these states contain necessary self-dependent feature information for the agent to perform the corresponding action. In other words, the self-dependent feature information in a state, for example s_1, is different from shared information that exists in all states, and necessary for decision making, for example a_1, which differs to the action a_2. In our work, the musculoskeletal model moves accoring to the target velocity map, if the musculoskeletal model moves to the target position, and then a new target position will be randomly generated. Immediately, the RL agent will make a new action, for example turning right, to move towards another target position. Therefore, in this case, we refer to the specific information contained in the state that signals that the musculoskeletal model has reached the target position as the self-dependent information, which makes the agent makes a specific action.

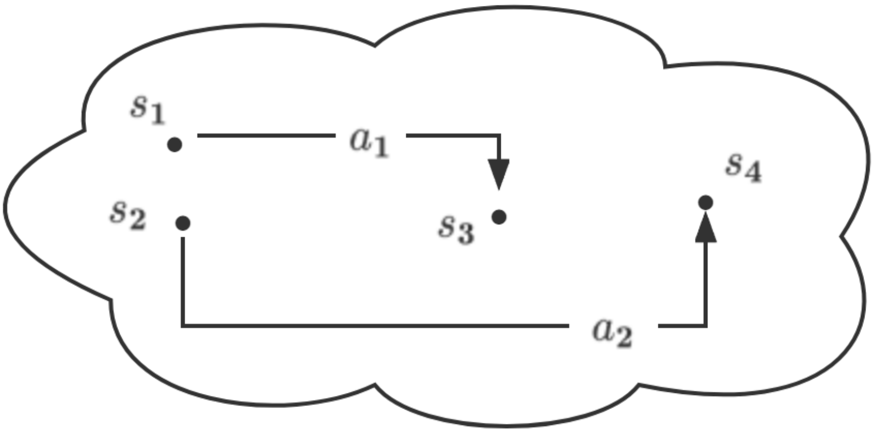


Figure 6. The choice of different actions under the same policy and different states.

The attention mechanism is introduced to focus on the information which is critical to the current task among the input information. Therefore, on one hand, based on the key-value attention mechanism, we try to reconstruct the current observation via capturing self-dependent feature information in each sampled state. To be specific, firstly, we randomly sample n sets of state action pairs $(s_1, a_1), (s_2, a_2), \ldots, (s_n, a_n)$ from the replay buffer. Here, the role of the sampled state action pairs (s_i, a_i) in our proposed framework is equal to (k_i, v_i) in the key-value attention mechanism. The state s_i and the action a_i are used to calculate the attention distribution and aggregate information, respectively. Moreover, we take the state-dependent exploration for the dependency between the sampled state s_i and the sampled action a_i. In other words, in the case of the same policy, the selected action is only related to the state inputted to the policy. Secondly, considering the advantage of the critic network in dealing with continuous action spaces, for example the simulated environment in our work, the critic network is usually used to approximate action-value function [25], so we take the critic network as the attention evaluation function. Thus, we calculate the action-value q_i of the above sampled actions with the critic network which takes the current observation and each sampled action a_i as input.

Based on the above method, a series of action-value q_i for the sampled actions can be achieved, which will serve as a basis for distinguishing the corresponding sampled state and reconstructing the initial observation. Thus, next to this operation, Softmax is used to normalize the corresponding action-value q_i, where the normalized action-value w_i represents the proportion of the sampled state in the reconstructed state. Significantly, the computed proportion w_i can be seen as the attention distribution α_i in key-value attention mechanism. Then, based on the attention distribution w_i, the sampled states s_i will be fused with the initial observation proportionally. In a word, the self-dependent feature information in each sampled state corresponding to the sampled action with higher action-value q_i will account for a larger proportion in reconstructed state. It is worth noting that, the way we perform feature fusion is element-wise addition. Based on this approach, the reconstructed state is influenced by the agent's action, and accordingly the state contains the information necessary to the action. Thus, the RL agent can select the corresponding action based on the information.

On the other hand, notably, autoencoder [26] is a kind of unsupervised neural network, and the goal of dimensionality reduction can be achieved by adjusting the number of hidden layers in both modules including the encoder and the decoder. Therefore, we use autoencoders to overcome the curse of dimensionality caused by the high-dimensional musculoskeletal model. The specific process is depicted as Figure 7.

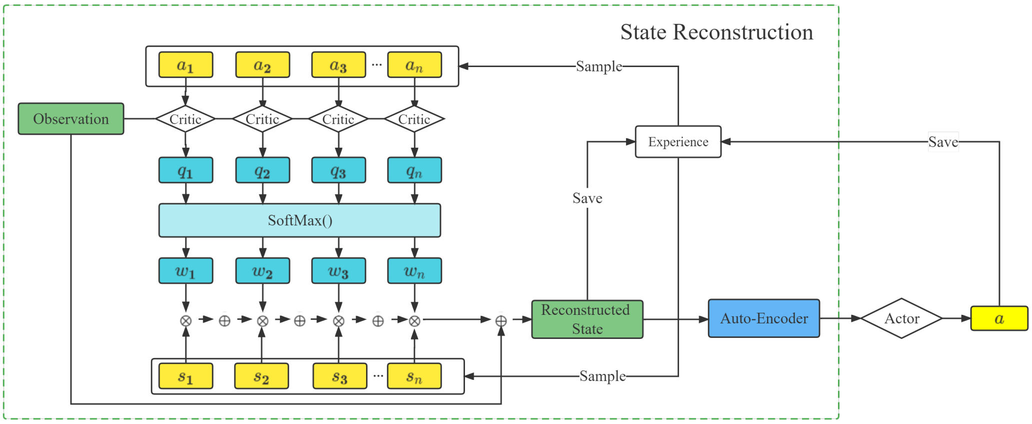


Figure 7. State reconstruction.

6.3. AT_SD3 for Gait Adjustment

Algorithm 1 presents the pseudocode of AT_SD3 for the gait adjustment.

Algorithm 1: AT_SD3 for the gait adjustment.

```
Initialize the simulated musculoskeletal model environment
Initialise critic networks $\alpha_1$,$\alpha_2$ and actor networks $\beta_1$,$\beta_2$ with random parameters
  $\theta_1$,$\theta_2$,$\phi_1$,$\phi_2$
Initialise target networks $\theta_1^- \leftarrow \theta_1, \theta_2^- \leftarrow \theta_2, \phi_1^- \leftarrow \phi_1, \phi_2^- \leftarrow \phi_2$
Initialise replay buffer $B$
Add noise to the actor network $\beta_1$, $\beta_2$
for $t$ = $1$ to $T$ do
    Observe the environmental state $s$ (including the musculoskeletal state $S$ and
      the target velocity map $T$)
    if $t$ >$10000$ then
        $n$ state-action pairs $(s_1, a_1), (s_2, a_2), \ldots, (s_n, a_n)$ from the replay buffer $B$
        Calculate the action value $q_i$ of the sampled action $a_i$ and the current
          observation with the critic network
        Get the attention distribution $w_i$ by normalizing the Q-value $(q_1, q_2, \ldots, q_n)$
          with Softmax operation
        Get state $s'$ through fusing sampled state $s_i$ according to the $w_i$
        Fuse the current observation $s$ and the $s'$ to get reconstructed state
        Based on the reconstructed state $s''$, use auto-encoder to extract state
          features $\varphi(s'')$
        Store transition tuple ($\varphi(s'')$,$a$, $J$,$s$,$d$) in $B$
    else if $t$ <$10000$ then
        Execute an action $a$ referring to the muscle activations
        Observe reward $J$ using Equation (12), new state $s$ and done flag $d$
        Store transition tuple $(s, a, J, s, d)$ in $B$
    for $i$ = $1$, $2$ do
        Sample a batch of $N$ transitions from $B$
        Sample $K$ noises $\epsilon \sim N(0, \bar{\sigma})$
        Add the additional noises to the action $a$
        Compute the action value using Equation (5)
        Compute the target value using Equation (7)
        Update actor networks using $1/N \sum_s \left[\nabla_a \alpha_i(s, a \mid \theta_i) \nabla_{\phi_i} \beta(s \mid \phi_i)\right]$
        Update critic networks using $1/N \sum_s (y_i - \alpha_i(s, a \mid \theta_i))^2$
        Update target networks using
          $\theta_i^- \leftarrow \tau\theta_i + (1-\tau)\theta_i^-$,$\phi_i^- \leftarrow \tau\phi_i + (1-\tau)\phi_i^-$
```

7. Experiment Analysis

7.1. Experiment Preparation

7.1.1. Dataset

To validate the effectiveness of the kinematic and ground reaction forces obtained via the simulation based on DRL algorithms, we compare the simulated data with the experimental data in a public dataset [27], where more details of the experiment refer to Section 7.2.2. The dataset contains a single-source, readily accessible repository of comprehensive gait data for a large group of children walking at a wide variety of speeds including very slow (below average speed), slow, free, fast and very fast (above average speed). Specifically, there are seven kinds of gait data: joint rotations, ground reaction forces, joint moments, joint power, EMG (electromyographic), cycle events and an ANOVA table with results for selected parameters in this dataset.

7.1.2. Evaluation Metrics

In order to compare the similarity between the experimental gait data and the simulated gait data, two evaluation metrics are adopted in this paper, namely mean absolute error (MAE), root mean square error (RMSE). These two metrics are defined as follows:

$$MAE = 1/m \sum_{i=1}^{m} |y_i - y_i'|, \tag{21}$$

$$RMSE = \sqrt{1/m \sum_{i=1}^{m} (y_i - y_i')^2}, \tag{22}$$

where m denotes the total number of gait data, y_i and $y_i^{'}$ represent the simulated and experimental data of the $i - th$ sample, respectively.

7.1.3. Parameter Settings

The hyperparameters of all methods are summarized in Table 2. It can be observed that two hidden layers are used, and the number of neurons in each hidden layer are 128 and 64, respectively. Considering the high-dimensional environment, we set the replay buffer size to 5 × 10^6 and the batch size is 256. Regarding the learning rate, TD3, AT_SD3, SD3, SD3_AE and PPO methods are all set to 0.0001, while DDPG method is set to 0.01. In addition, the hyperparameters, related to the noise added to the actor network, are also listed in Table 1. Note that all parameters are obtained through extensive numerical experiments.

Table 2. Hyperparameters of TD3 [14], DDPG [13], SD3, SD3_AE, PPO [15] and AT_SD3.

Method	Parameters	Results
Shared hyperparameters	Batch size	256
	Critic network	256 →128→64→1
	Actor network	256 →128→64→22
	Optimizer	Adam
	Replay buffer size	5 ×10^6
	Discount factor	0.99
	Target update rate	0.01
SD3	Learning rate	0.0001
	TAU	0.005
	Policy noise	0.2
	Sample size	50
	Noise clip	0.5
	Beta	0.05
	Importance sampling	0
PPO	Learning rate	0.0001
	Iteration	8
AT_SD3	Learning rate	0.0001
	Encoder	256→128→64→32→16→3
	Decoder	3→16→32→64→100
	Initial standard deviation (SD)	1.55
	Desired action SD	0.001
	Adaptation coefficient	1.05
DDPG	Learning rate	0.01
TD3	Learning rate	0.0001
	TAU	0.005
SD3_AE	Encoder	256→128→64→32→16→3
	Decoder	3→16→32→64→100
	Learning rate	0.0001

7.2. Results and Analysis

7.2.1. Algorithm Performance

In order to verify the effectiveness of AT_SD3 in the respect of gait adjustment based on the musculoskeletal model, we compare it with other state-of-the-art DRL algorithms, including TD3 [14], DDPG [13], PPO [15], SD3_AE and SD3, on the gait adjustment problem. The result is shown in Figure 8.

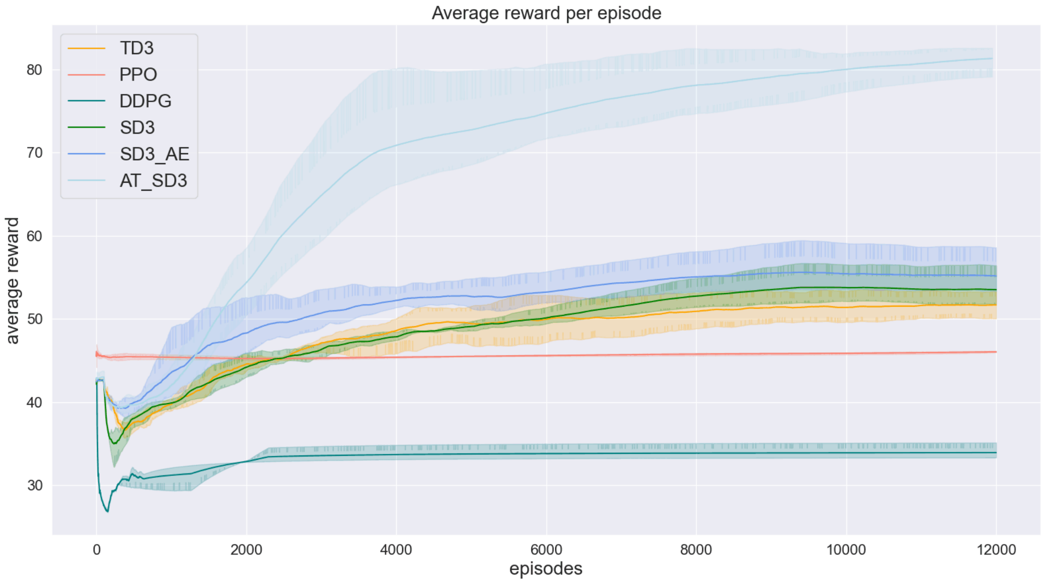


Figure 8. Performance of AT_SD3 and other state-of-the-art DRL algorithms.

Figure 8 shows the performance of AT_SD3 and other state-of-the-art DRL algorithms for the gait adjustment, where the horizontal axis represents the number of episodes and the vertical axis is the average reward. In this figure, each curve indicates the average reward for the gait adjustment using different DRL algorithms over a total of 12,000 episodes. The shaded area represents the SD varying from the mean value of the three independent experiments with same hyperparameters.

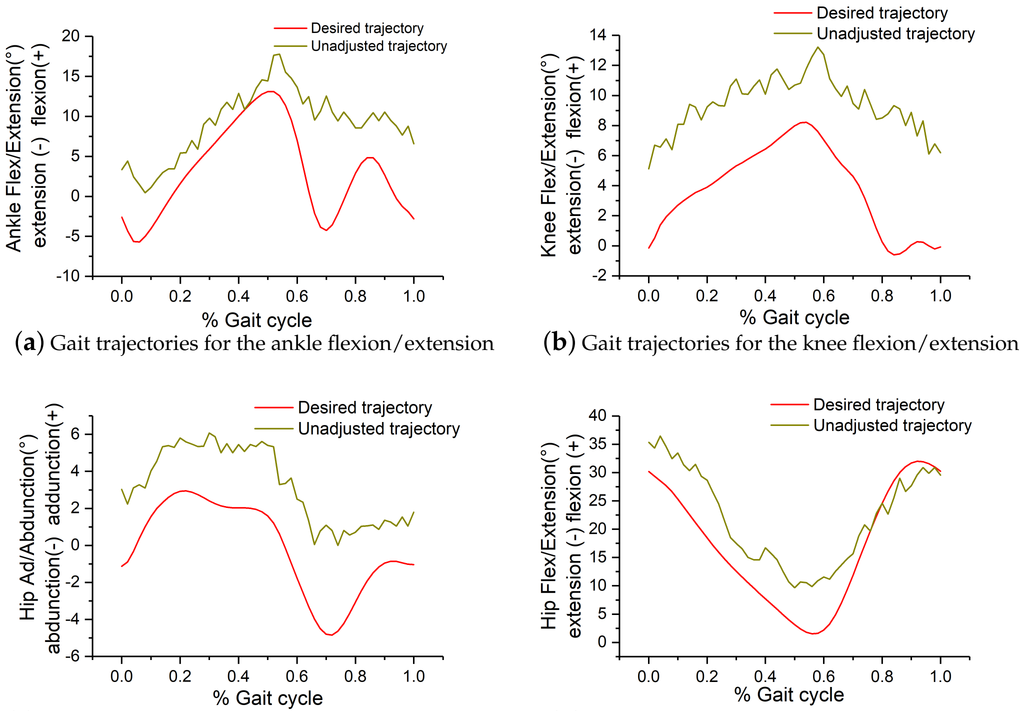
It can be noted that, on the one hand, the performance of AT_SD3 outperforms other traditional DRL algorithms after a certain number of episodes, including DDPG, PPO, TD3 and a novel DRL algorithm SD3. On the other hand, the performance of PPO keeps stable throughout the simulation, and the performance of DDPG is always poor compared to other algorithms, which may result from the limited algorithmic power in dealing with the curse of dimensionality in DRL. On the contrary, TD3, with more complex network structure, has better performance than PPO and DDPG. In our work, the current observation is a 339-dimensional musculoskeletal state, which may lead to this phenomenon. So, we introduce SD3 into our work to deal with the difficulty of gait adjustment caused by this problem. Due to the complexity of network structure, SD3 has a relative advantage over other RL algorithms in dealing with 'the curse of dimensionality'. However, as can be seen from Figure 8, after a certain number of episodes, the performance of SD3 keeps stable gradually but the rewards are relatively low. Therefore, an attention mechanism-based framework for gait adjustment is proposed. Based on the reward difference between AT_SD3 and other algorithms observed in Figure 8, we can conclude that AT_SD3 is more efficient than other traditional algorithms for the gait adjustment. Moreover, we provide an ablation experiment, named SD3_AE, to prove the effectiveness of our proposed framework. To be specific, we combine SD3 with the autoencoder for the gait adjustment. As can be seen in Figure 8, the performance of SD3_AE is better than SD3 due to the advantage of feature extraction and solving the curse of dimensionality. More importantly, by comparing the performance of AT_SD3 and SD3_AE, we can conclude that state reconstruction through the key-value attention mechanism is effective in gait adjustment. Through the above groups of comparative experiments, the experimental result demonstrates the effectiveness of fusing the self-dependent feature information necessary for decision making in each sampled state with the current observation.

7.2.2. Gait Adjustment

We compare different gait trajectories including the unadjusted trajectory obtained in previous work, the adjusted trajectory obtained in this work and the desired trajectory obtained in [27].

a. Unadjusted Trajectory and Desired Trajectory

Figure 9 shows the gait trajectories for different joints, including the ankle flexion/extension, the knee flexion/extension, the hip adduction/abduction and the hip flexion/extension corresponding to sub-figure (a) to (d), respectively, where the horizontal axis represents the gait cycle and the vertical axis represents different gait trajectories. In each sub-figure, red curve indicates the desired trajectory and another curve represents the unadjusted trajectory obtained by the human gait simulation in previous work. In terms of RMSE and MSE, Table 3 shows these similarity metrics between the desired trajectory and the unadjusted trajectory simulated in previous work.



(**a**) Gait trajectories for the ankle flexion/extension (**b**) Gait trajectories for the knee flexion/extension

(**c**) Gait trajectories for the hip adduction/abduction (**d**) Gait trajectories for the hip flexion/extension

Figure 9. The simulated kinematics compared to the experimental data in [27].

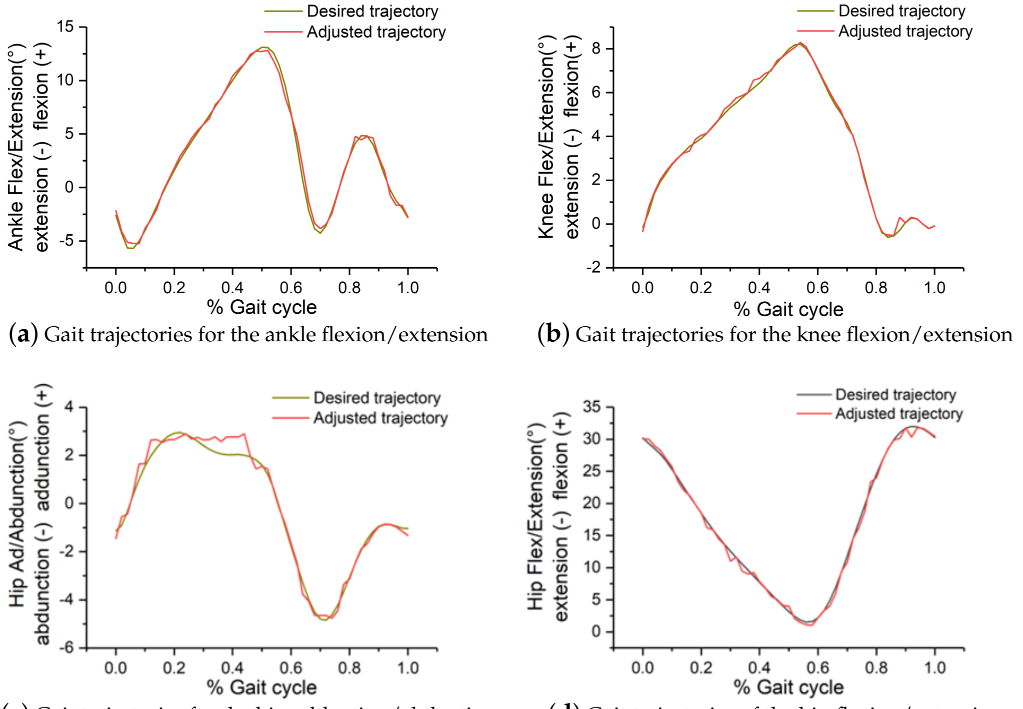
As can be seen from Figure 9, the unadjusted trajectory for different joints obtained in previous work are similar in shape to the desired trajectory, which is the mean kinematics calculated from the maximum and minimum value of the kinematics. However, as shown in Table 3 and Figure 9, there is a deviation between the unadjusted trajectory and the desired trajectory, which result from the randomness of the gait simulated by the algorithms in previous work. As can be seen from Table 3, these two kinds of metrics obtained in previous work are no more than 2.64 SD and no less than 1.22 SD.

Table 3. Metrics between desired trajectory and the unadjusted trajectory.

Metrics	Hip Ad/Abduction	Hip Flex/Extension	Knee	Ankle
RMSE	1.66	2.64	1.58	2.13
MAE	1.22	2.19	1.32	1.74

b. Adjusted Trajectory and Desired Trajectory

Figure 10 shows the trajectories for different joints, including the ankle flexion/extension, the knee flexion/extension, the hip adduction/abduction and the hip flexion/extension corresponding to sub-figure (a) to (d), respectively, where the horizontal axis represents the gait cycle and the vertical axis represents the gait trajectory for different joints. In each sub-figure, red curve indicates the desired trajectory and another curve represents the adjusted trajectory obtained in this work. Table 4 summarizes the similarity metrics between the desired trajectory and the adjusted trajectory obtained in this work, in terms of RMSE and MSE.



(**a**) Gait trajectories for the ankle flexion/extension (**b**) Gait trajectories for the knee flexion/extension

(**c**) Gait trajectories for the hip adduction/abduction (**d**) Gait trajectories of the hip flexion/extension

Figure 10. Desired trajectory and the adjusted trajectory based on the simulated lower limb exoskeleton.

Table 4. Metrics between desired trajectory and the adjusted trajectory.

Metrics	Hip Ad/Abduction	Hip Flex/Extension	Knee	Ankle
RMSE	0.32	0.18	0.14	0.28
MAE	0.23	0.1	0.1	0.22

As can be found from Figure 10, the gait trajectories for different joints obtained in this work are almost consistent with the desired trajectory in shape and value. This phenomenon demonstrates the effectiveness of gait adjustment with the simulated lower limb exoskeleton, which is modeled as a MDP problem in this work. However, in sub-figure (c), the adjusted trajectory for the hip adduction/abduction deviate from the desired trajectory in part of the gait cycle, which may result from the randomness. As can be found from Table 3, these metrics are no more than 0.32 SD which is much lower the figures in

Table 4, and these figures also demonstrate the effectiveness of the gait adjustment with the simulated exoskeleton.

8. Conclusions and Future Work

In order to verify the effect of gait rehabilitation for patients with mobility disorders, one available approach is to adjust gait without using physical equipment, where the musculoskeletal model is used in 2019 NeurIPS "Learning to Move–Walk Around" challenge. In this paper, we adopt MDP to model the gait adjustment problem. Moreover, based on DRL algorithms and the attention mechanism, a framework named AT_SD3 for the gait adjustment is proposed. Taking advantages of the attention mechanism, the self-dependent feature information for decision making in the sampled states generated by the agent's actions can be captured, with which we can reconstruct the initial observation with more interpretive information. Considering the high dimension of RL state and the advantage of autoencoder, the autoencoder is applied to solve the problem of 'the curse of dimensionality'. To investigate the performance of the proposed framework, the proposed framework and other traditional DRL algorithms are applied to the gait adjustment. The comparison results suggest that the performance of the proposed framework is superior to other traditional RL algorithms. Moreover, we compare different trajectories, including the unadjusted trajectory and adjusted trajectory obtained in previous work and in this paper, respectively, and comparative results suggest the trajectories simulated by using our proposed framework are closer to the desired trajectory in both shape and value, which outperforms the related previous work. In terms of the evaluation metrics of MAE and RMSE, results show the trajectories obtained in this paper are more accurate than those obtained in previous work.

As for the future work, the way to extract the information in each sampled state that is critical to the selected action is still worth studying. Moreover, we will purchase an actual lower limb exoskeleton to verify the effectiveness of the proposed exoskeleton control framework. Therefore, in the process of controlling the actual lower limb exoskeleton, the adjustment of exoskeleton parameters and the RL modeling for the exoskeleton control are worth studying.

Author Contributions: Conceptualization, A.L.; data curation, A.L.; formal analysis, A.L.; funding acquisition, J.C., Q.F. and Y.W.; investigation, H.W., Y.W. and Y.L.; methodology, A.L. and Q.F.; project administration, J.C., H.W., Y.W. and Y.L.; software, A.L.; supervision, J.C., Q.F. and Y.L.; validation, A.L.; writing—original draft, A.L.; writing—review and editing, Q.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by National Key R&D Program of China (No. 2020YFC2006602), National Natural Science Foundation of China (No. 62172324, No. 62072324, No. 61876217, No. 61876121), University Natural Science Foundation of Jiangsu Province (No. 21KJA520005), Primary Research and Development Plan of Jiangsu Province (No. BE2020026), Natural Science Foundation of Jiangsu Province (No. BK20190942).

Data Availability Statement: The data presented in this study are openly available at https://doi.org/10.1016/j.jbiomech.2008.03.015 (accessed on 17 November 2022)

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
MDP	Markov Decision Process
SD3	Softmax Deep Double Deterministic policy gradients
DDPG	Deep Deterministic policy gradients

TD3	Twin Delayed Deep Deterministic policy gradient
PPO	Proximal Policy Optimization
SD	Standard Deviation
MAE	Mean Absolute Error
RMSE	Root Mean Square Error

References

1. Louie, D.R.; Eng, J.J. Powered robotic exoskeletons in post-stroke rehabilitation of gait: A scoping review. *J. Neuroeng. Rehabil.* **2016**, *13*, 53. [CrossRef] [PubMed]
2. Riener, R.; Lunenburger, L.; Jezernik, S.; Anderschitz, M.; Colombo, G.; Dietz, V. Patient-cooperative strategies for robot-aided treadmill training: First experimental results. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2005**, *13*, 380–394. [CrossRef] [PubMed]
3. Chen, B.; Ma, H.; Qin, L.Y.; Gao, F.; Chan, K.M.; Law, S.W.; Qin, L.; Liao, W.H. Recent developments and challenges of lower extremity exoskeletons. *J. Orthop. Transl.* **2015**, *5*, 26–37. [CrossRef] [PubMed]
4. Mendoza-Crespo, R.; Torricelli, D.; Huegel, J.C.; Gordillo, J.L.; Rovira, J.L.P.; Soto, R. An Adaptable Human-Like Gait Pattern Generator Derived From a Lower Limb Exoskeleton. *Front. Robot. AI* **2019**, *6*, 36. [CrossRef] [PubMed]
5. Hussain, S.; Xie, S.Q.; Jamwal, P.K. Control of a robotic orthosis for gait rehabilitation. *Robot. Auton. Syst.* **2013**, *61*, 911–919. [CrossRef]
6. Sado, F.; Yap, H.J.; Ghazilla, R.A.B.R.; Ahmad, N. Exoskeleton robot control for synchronous walking assistance in repetitive manual handling works based on dual unscented Kalman filter. *PLoS ONE* **2018**, *13*, e0200193. [CrossRef] [PubMed]
7. Castro, D.L.; Zhong, C.H.; Braghin, F.; Liao, W.H. Lower Limb Exoskeleton Control via Linear Quadratic Regulator and Disturbance Observer. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018; pp. 1743–1748.
8. Chinimilli, P.T.; Subramanian, S.C.; Redkar, S.; Sugar, T. Human Locomotion Assistance using Two-Dimensional Features Based Adaptive Oscillator. In Proceedings of the 2019 Wearable Robotics Association Conference (WearRAcon), Scottsdale, AZ, USA, 25–27 March 2019; pp. 92–98.
9. Sado, F.; Yap, H.J.; Ghazilla, R.A.B.R.; Ahmad, N. Design and control of a wearable lower-body exoskeleton for squatting and walking assistance in manual handling works. *Mechatronics* **2019**, *63*, 102272. [CrossRef]
10. Bingjing, G.; Jianhai, H.; Xiangpan, L.; Lin, Y.Z. Human–robot interactive control based on reinforcement learning for gait rehabilitation training robot. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419839584. [CrossRef]
11. Zhang, Y.; Li, S.; Nolan, K.J.; Zanotto, D. Adaptive Assist-as-needed Control Based on Actor-Critic Reinforcement Learning. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4066–4071.
12. Khan, S.G.; Tufail, M.; Shah, S.H.; Ullah, I. Reinforcement learning based compliance control of a robotic walk assist device. *Adv. Robot.* **2019**, *33*, 1281–1292. [CrossRef]
13. Rose, L.; Bazzocchi, M.C.F.; Nejat, G. End-to-End Deep Reinforcement Learning for Exoskeleton Control. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 4294–4301.
14. Oghogho, M.; Sharifi, M.; Vukadin, M.; Chin, C.; Mushahwar, V.K.; Tavakoli, M. Deep Reinforcement Learning for EMG-based Control of Assistance Level in Upper-limb Exoskeletons. In Proceedings of the 2022 International Symposium on Medical Robotics (ISMR), Atlanta, GA, USA, 13–15 April 2022; pp. 1–7.
15. Kumar, V.C.V.; Ha, S.; Sawicki, G.; Liu, C.K. Learning a Control Policy for Fall Prevention on an Assistive Walking Device. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2019; pp. 4833–4840.
16. Gou, J.; He, X.; Lu, J.; Ma, H.; Ou, W.; Yuan, Y. A class-specific mean vector-based weighted competitive and collaborative representation method for classification. *Neural Netw.* **2022**, *150*, 12–27. [CrossRef] [PubMed]
17. Silver, D.; Lever, G.; Heess, N.M.O.; Degris, T.; Wierstra, D.; Riedmiller, M.A. Deterministic Policy Gradient Algorithms. In Proceedings of the ICML, Beijing, China, 21–26 June 2014.
18. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.M.O.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
19. Pan, L.; Cai, Q.; Huang, L. Softmax Deep Double Deterministic Policy Gradients. *arXiv* **2020**, arXiv:2010.09177.
20. Fujimoto, S.; van Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. *arXiv* **2018**, arXiv:1802.09477.
21. Ciosek, K.; Vuong, Q.H.; Loftin, R.T.; Hofmann, K. Better Exploration with Optimistic Actor-Critic. In Proceedings of the NeurIPS, Vancouver, BC, Canada, 8–14 December 2019.
22. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. *arXiv* **2017**, arXiv:1706.03762.
23. Fortunato, M.; Azar, M.G.; Piot, B.; Menick, J.; Osband, I.; Graves, A.; Mnih, V.; Munos, R.; Hassabis, D.; Pietquin, O.; et al. Noisy Networks for Exploration. *arXiv* **2017**, arXiv:1706.10295.
24. Plappert, M.; Houthooft, R.; Dhariwal, P.; Sidor, S.; Chen, R.Y.; Chen, X.; Asfour, T.; Abbeel, P.; Andrychowicz, M. Parameter Space Noise for Exploration. *arXiv* **2017**, arXiv:1706.01905.

25. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.
26. Baldi, P. Autoencoders, Unsupervised Learning, and Deep Architectures. In Proceedings of the ICML Unsupervised and Transfer Learning, Bellevue, WA, USA, 2 July 2011.
27. Schwartz, M.H.; Rozumalski, A.; Trost, J.P. The effect of walking speed on the gait of typically developing children. *J. Biomech.* **2008**, *41*, 1639–1650. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.