Learning Whole-body Motion Control through Instruction Learning and Human Motion Data

Zhipeng Xu[1], Kaixuan Chen[1], Linqi Ye[1], and Boyang Xing[2]

[1] School of Future Technology, Shanghai University, 200444 Shanghai, China
yelinqi@shu.edu.cn
[2] National and Local Co-Built Humanoid Robotics Innovation Center, 201203 Shanghai, China

Abstract. This paper proposes a novel imitation learning approach for the whole-body motion control of humanoid robots. Based on the instruction learning framework proposed in our prior work, we integrate human motion capture data as a feedforward action in this paper, which is combined with a feedback action driven by reinforcement learning to achieve human-like whole-body movements. Compared to other imitation learning methods, the proposed method can significantly enhance the training efficiency due to the application of a feedforward action. Furthermore, since the motion-mimic capability is mainly determined by the feedforward action while the neural network only plays a role as a stabilizer, it enables the control of multiple motion skills using a single neural network. The effectiveness of the proposed method in whole-body motion imitation learning is verified through several simulation tasks performed on the Unitree H1 robot. The attached video can be found at https://linqi-ye.github.io/video/icira25.mp4

Keywords: Humanoid Robot, Reinforcement Learning, Imitation Learning, Instruction Learning

1 Introduction

With the escalating complexity of modern robotic applications, data-driven learning methods have demonstrated substantial potential, especially the reinforcement learning (RL) algorithm, which aims to maximize a reward through interactions with observations and environments. Many powerful RL methods have been proposed, such as Proximal Policy Optimization (PPO) [1], Soft Actor Critic (SAC) [2], and Curriculum Learning (CL) [3]. However, despite these advancements, directly applying these methods to achieve diverse and agile whole-body motion control for complex humanoid robots often faces challenges related to sample efficiency, the complexity of reward design, and the difficulty of acquiring varied and robust policies for multiple skills within a single framework.

Imitation learning (IL), which integrates motion reference data into reward functions to guide robotic agents. There have been many studies applying imitation learning to robot locomotion, Behavior Cloning (BC)[4], an early direct

policy learning approach, in its relative research, researchers relied on Hidden Markov Models(HMMs) to enable simple human-dance imitation on robot[5], however, its effectiveness was often contingent on highly specific conditions and a well-defined control strategy, or conversely, struggled in its absence. Subsequent advancements, from Dynamic Movement Primitives(DMPs)[6] to Variational Learning(VL)[7], addressed some control strategy limitations. Yet, these methods often exhibited poor generalizability when trained with limited sample sizes.Inverse Reinforcement Learning(IRL)[8][9] emerged as a novel imitation learning paradigm. While maximum margin-based IRL frameworks incurred high computational costs, probabilistic models paved the way for data-driven deep reinforcement learning approaches. Peng et al. proposed a framework that enabled the training of humanoid robots in virtual environments through the algorithm of DeepMimic, yielding highly robust policies[10]. Nevertheless, this approach, too, demands substantial quantities of data. A persistent challenge across these imitation learning methods has been the arduous design of effective reward functions. Generative Adversarial Imitation Learning(GAIL)[11][12] offered a solution to this reward design problem, but its training process remains computationally intensive and requires a large volume of high-quality demonstration data. In general, the substantial data requirements, the key limitation of the inherent incompleteness of real-world demonstrations and susceptibility to local optima in imitation learning often hinder the acquisition of optimal policies [13].

Instruction learning as an alternative imitation learning strategy takes advantage of a feedforward action as an initial guide policy[14]. This feedforward-guided policy optimization enables more efficient exploration of the policy landscape, as the learning process starts from a known motion, albeit suboptimal. By refining this initial policy, agents achieve accelerated convergence to policies that surpass the original demonstration's performance. This method also mitigates the need for intricate reward shaping, as the feedforward itself provides a strong directional cue, simplifying the learning problem and potentially leading to more robust behaviors with reduced training complexity. Based on instruction learning, we developed Unity RL Playground, which is a simple, yet efficient and versatile framework for RL in mobile robotics[15]. Unity RL Playground is build upon Unity ML-Agents[16] and tailored for robotic reinforcement learning. A key feature is its ease of use and versatility to achieve various locomotion behaviors for mobile robots. This characteristic significantly broadens its accessibility to a wider research and development community. We leverage this framework for the simulation training in this paper. However, the previous version of instruction learning has adopted hand-crafted trajectories as feedforward, which is difficult to extend to whole-body motion control.

Human motion data is crucial for whole-body imitation learning. Extensive research has explored the integration of adaptive motion functions (AMFs) or dynamics with RL for continuous gait design and control, yielding highly robust control strategies[17][18]. Zhou et al. combined IL with biological data collected in nature to instruct a multi-jointed robot to learn animal behavior[19]. Prior research has extensively investigated the direct retargeting of human motion

capture data to humanoid robot kinematics[20][21][22]. Building upon this foundation, the work[23][24] created a novel, large-scale motion dataset specifically optimized for humanoid robot compatibility, which systematically adapts the human-derived motions to ensure their kinematic and dynamic feasibility within the operational constraints of real-world humanoid robot platforms.

Motivated by the aforementioned work, we take advantage of human motion data to serve as a feedforward and thus extend instruction learning to whole-body motion control. The contributions of this paper are summarized as follows. First, we extend the instruction learning method to whole-body motion control by applying human motion capture data as a feedforward action, which enables humanoid robots to learn multiple motion skills efficiently through a single neural network. Second, we verify the effectiveness of the proposed method in whole-body motion imitation learning through several simulation tasks with the Unitree H1 robot, which are presented in the attached video.

2 Method

The framework of the proposed method is shown in Fig. 1. Compared to the previous version of instruction learning[14], the primary modifications lie in the feedforward component and the reward aspect. The details are introduced in the following.

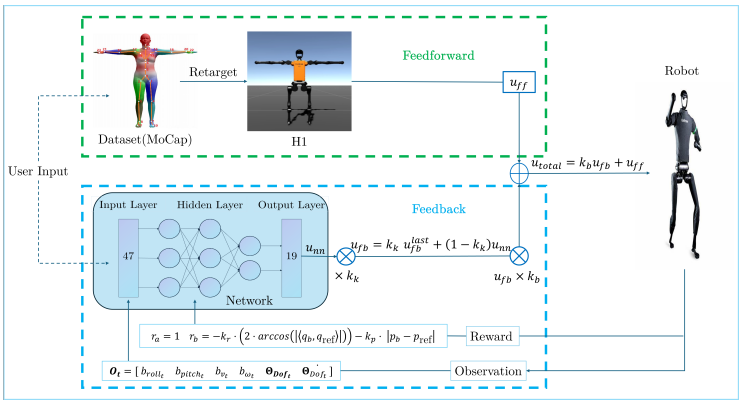


Fig. 1: Framework

2.1 Feedforward Design

In this study, a retargeted joint angle reference from human motion data is employed to define the robot's feedforward action. With these predefined motion, the robot will be trained to optimize the feedback action sequences, aiming to achieve higher reward values.

Our dataset, sourced from AMASS Dataset[25], necessitated a retargeting process for its integration into the Unity environment. This was primarily due to fundamental discrepancies in their coordinate systems and angular representations: AMASS Dataset employs a Z-up orientation with joint angles recorded in radians, whereas Unity utilizes a Y-up orientation and expresses angles in degrees. Subsequently, these transformed joint angles were converted into quaternion representations to ensure robust motion blending within the Unity engine. The transformation as follows:

$$\begin{cases} \mathbf{p}_{Unity} = \mathbf{R}_c \mathbf{p}_{MoCap}, \quad \mathbf{R}_c = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \\ \mathbf{q}_{Unity} = \mathcal{Q}(\mathbf{q}_{MoCap}) = \begin{bmatrix} q_y \\ -q_z \\ q_x \\ q_w \end{bmatrix}, \quad \mathbf{q}_{MoCap} = \begin{bmatrix} q_x \\ q_y \\ q_z \\ q_w \end{bmatrix} \\ \boldsymbol{\theta}_{Unity} = \boldsymbol{\theta}_{MoCap} \times \dfrac{180}{\pi} = \begin{bmatrix} \theta_1^{rad} \\ \vdots \\ \theta_{19}^{rad} \end{bmatrix} \times \dfrac{180}{\pi} \end{cases} \tag{1}$$

2.2 Learning Settings

(A) **Hyperparameters:**
In this study, proximal policy optimization (PPO) is used as the reinforcement learning algorithm. The hyperparameters of the corresponding network are configured as follows: batch size 2048, buffer size 20480, beta 0.005, epsilon 0.2, lambda 0.95, learning rate 0.0003, num epoch 3.

(B) **Neural network:**
The neural network architecture employs an actor-critic structure. The actor network is implemented as a Multi-Layer Perceptron (MLP) comprising three hidden layers, each with 512 hidden units. Complementing this, the critic network is also an MLP, featuring two hidden layers, each containing 128 hidden units.

(C) **Observation:**
The observations are

$$\mathbf{O_t} = \left[b_{roll_t} \; b_{pitch_t} \; b_{v_t} \; b_{\omega_t} \; \boldsymbol{\Theta}_{Dof_t} \; \dot{\boldsymbol{\Theta}}_{Dof_t} \right] \tag{2}$$

where b_{roll_t},b_{pitch_t} represent the roll and pitch angles of the root(Unit: radian), b_{v_t},b_{ω_t} represents the linear velocity and angular velocity of the root, $\boldsymbol{\Theta}_{Dof_t}$,$\dot{\boldsymbol{\Theta}}_{Dof_t}$ are the joint angle,joint angular velocity, respectively, which contains the same number of variables as the joint numbers of the robot.

(D) **Action:**
The action is a combination of the feedforward action, which is obtained from human motion data, and a feedback action, which is a filtered output of the neural network. The feedback action is expressed as

$$u_{fb} = k_k u_{fb}^{last} + (1 - k_k) u_{nn} \tag{3}$$

where u_{fb}^{last} represents the value of the feedback term from the preceding time step, u_{nn} denotes the raw output of the actor network. Then, the feedback action u_{fb} is weighted by k_b and combined with the feedforward component to yield the final action:

$$u_{total} = k_b u_{fb} + u_{ff} \tag{4}$$

2.3 Reward Design

Instruction learning alleviates the complexities associated with traditional reward engineering, thereby simplifying reward definition. In this study, we use some simple reward components, which can be applied to all motion imitation tasks.

(1) **Alive reward:**
r_a is designed as $r_a = 1$, its existence encourages the robot to maintain balance and not fall. In all time steps before termination, the robot can obtain this reward.

(2) **Body reward:**
r_b is designed as

$$r_b = -k_r \cdot (2 \cdot \arccos(|\langle q_b, q_{\text{ref}} \rangle|)) - k_p \cdot \|p_b - p_{\text{ref}}\| \tag{5}$$

where q_b,q_{ref} represents the orientation of the body expressed by a quaternion, p_b,p_{ref} represents the real and reference body position.

(3) **Feedforward switching processing:**
Imperfections in feedforward compensation yield substantial state disparities across temporal frames preceding and succeeding a change in feedforward control paradigms. Such pronounced discontinuities impede the effective learning or optimization of the control policy during transitional phases. To facilitate a robust feedforward transition, a specialized control scheme adaptation is employed. During the initial temporal window subsequent to a feedforward set update, the system's reward evaluation is restricted to only account for elemental survival objectives, thereby temporarily decoupling the body-specific reward components from the overall objective function.

2.4 Episode Termination Condition

During the training process, the termination condition is set to:

$$2 \cdot \arccos(|\langle q_b, q_{\text{ref}} \rangle|) > 30 \quad or \quad \|p_b - p_{\text{ref}}\| > 0.3 \tag{6}$$

it indicates to terminate when the orientation or position of the robot is too far from the reference.

2.5 Curriculum Learning

Our work introduces a generalized curriculum learning scheme which demonstrably leads to faster convergence and enhanced training stability. We select multiple motion tasks to train simultaneously. During the first ten million steps, we train each motion for 300 seconds and then switch to the next motion, in order to make sure that every action is fully practiced. When it comes to the last two million steps, we spent 30 seconds training on each motion, namely increasing the switching frequency, which can reduce the impact of forgetting effectively.

3 Simulation

The simulations were conducted with Unity RL Playground. This platform was selected for its proven efficacy in providing an efficient and user-friendly Reinforcement Learning (RL) development environment specifically tailored for robotic applications. The discrete action timestep was set to 0.02 seconds. To accelerate the training process, a parallelized learning paradigm was adopted, leveraging 24 concurrent instances of the robot to train in parallel.

3.1 Simulation Settings

(A) **Robot Model and Specifications:**

The robot model utilized for simulation is the Unitree-H1 bipedal humanoid robot, as depicted in the accompanying figure. This robot features a total of 19 revolute joints. Each leg possesses 5 degrees of freedom, comprising three at the hip (Hip Y, Hip R, Hip P), one at the knee (Knee P), and one at the ankle (Ankle P). Each arm has 4 degrees of freedom.The main specifications of the Unitree-H1 robot are shown in Table 1(a)(b), and Fig.2 shows the zero position of the robot.

(B) **Feedforward data:**

The feedforward component of our control strategy is generated from a dataset of human motion capture, which has been meticulously retargeted to ensure kinematic and dynamic compatibility with humanoid robots. Specifically, our comprehensive feedforward set includes motions for golf, tennis, guitar, violin, and general waving gestures.

(C) **training settings:**

$k_b = 30, k_k = 0.9, k_r = 0.01, k_p = 1$

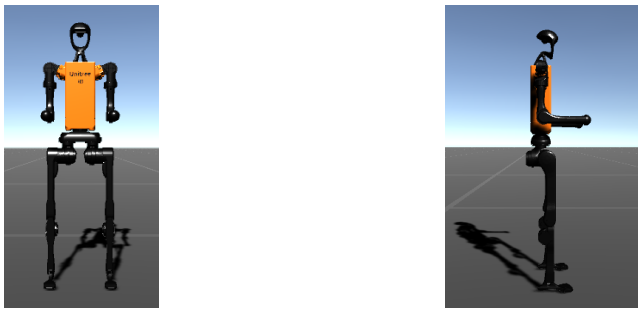
(a) Joint Unit Limit Torques

Joint Type	Limit Torque (Nm)
Knee Joints	360
Hip Joints	220
Ankle Joints	59
Arm Joints	75

(b) Joint Motion Ranges (rad)

Joint Type	Motion Range (rad)
Hip Y	-0.43 to +0.43
Hip R	-0.43 to +0.43
Hip P	-3.14 to +2.53
Knee P	-0.26 to +2.05
Ankle P	-0.87 to +0.52
Torso Y	-2.35 to +2.35
Shoulder Pitch	-2.87 to +2.87
Right Shoulder Roll	-3.11 to +0.34
Left Shoulder Roll	-0.34 to +3.11
Right Shoulder Yaw	-4.45 to +1.3
Left Shoulder Yaw	-1.3 to +4.45
Elbow Pitch	-1.25 to +2.61

Table 1: Joint Specifications: (a) Torque Limits, (b) Motion Ranges



(a) Front view of H1 (b) Side view of H1

Fig. 2: Unitree-H1 zero position

3.2 Simulation Results

The process of our simulation is dived by tow parts, training and playing. We design 13 motion tasks (some are repeated) for H1 robot to learn, the order and name of these motion tasks are listed in the following. We repeated some of the tasks for several times because they are more difficult to learn, and the following results prove that our hypothesis is correct.

Index	Motion Name	Index	Motion Name
1	golf drive poses	8	golf drive poses
2	golf drive poses	9	golf drive poses
3	tennis forehand left poses	10	tennis forehand left poses
4	tennis forehand left poses	11	tennis forehand left poses
5	guitar right poses	12	wave left poses
6	violin left poses	13	wave right poses
7	wave both poses		

Table 2: Motion Tasks

We used reinforcement learning to train the robot for twelve million steps. During training, we found that the reward decreased when the feedforward motion was switched, as shown in Fig. 3. This suggested that the policy network just trained may forget the motion it just learned, implying that we use curriculum learning to reduce the time allocated on each motion.

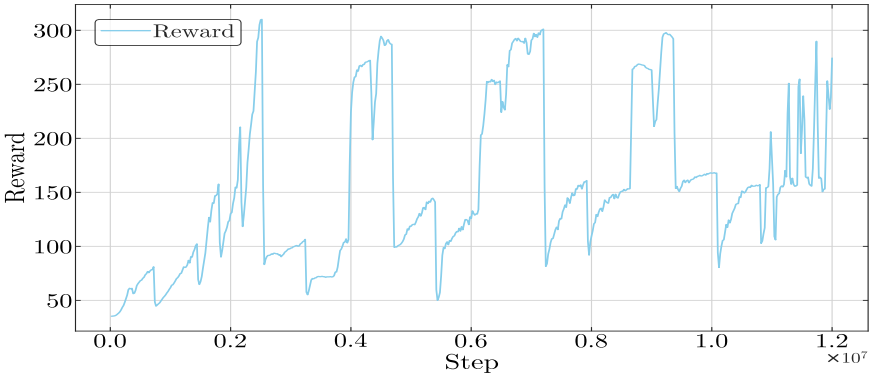
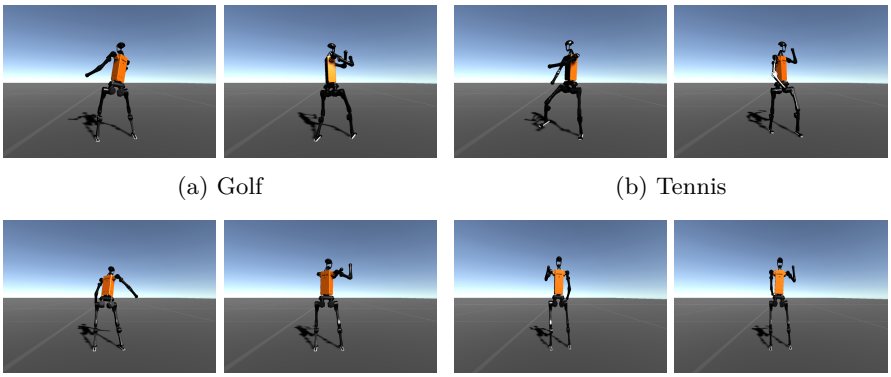


Fig. 3: Cumulative Reward

As a result, despite the fact that the reward function decreases at motion switch, the extent to which it decreases is gradually decreasing over time and the reward stably remains at a relatively high value in the end.



(a) Golf (b) Tennis

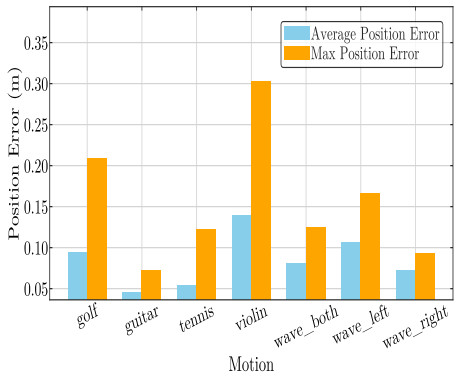
(c) Violin (d) Wave

Fig. 4: Learned locomotion behavior

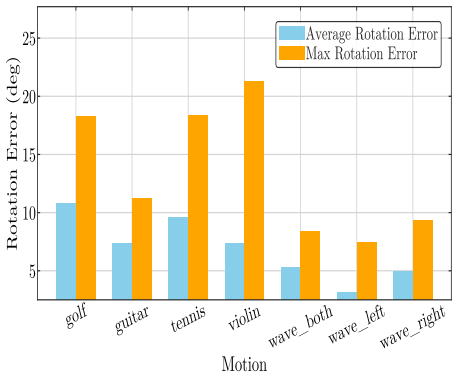
Fig. 4 depicts the trained H1 robot performing different motion tasks, but it is difficult to objectively quantify and evaluate the training performance based on visual observation alone. Therefore, we collected statistics on body position and posture errors of the robot during playing, and conducted a comprehensive analysis by combining them with charts.

In Fig. 5, the comparison results of "Average Error" and "Max Error" of seven different tasks (golf, guitar, tennis, violin, wave both, wave left, wave right) are

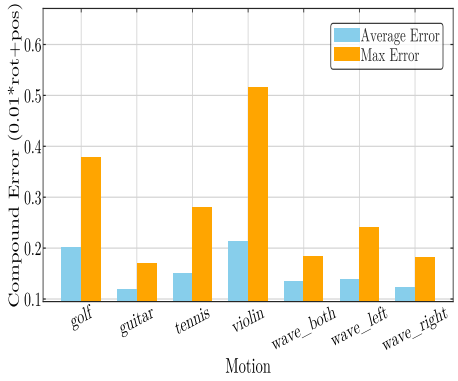
shown. According to the difference in imitation accuracy, these tasks can be divided into two categories.



(a) Position error



(b) Rotation error



(c) Compound error (0.01*rot+pos)

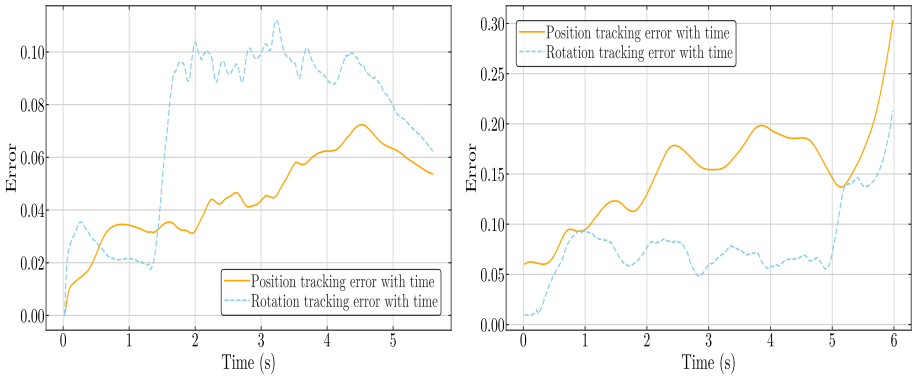
Fig. 5: Motion tracking error

One is tasks with good imitation effects, including guitar and three wave tasks. The average error and maximum error of these tasks are both at a low level. Their common feature is that the movements are mainly concentrated in the upper limbs and the posture changes are small. These characteristics reduce the complexity of action control, making it easier for the policy network to converge to a stable solution, thereby achieving accurate imitation. Therefore, it can be considered that upper limb actions that do not involve obvious movement of the lower limbs are easier to master by the robot, and show good stability and consistency in the reproduction process.

The other is tasks that are more difficult to imitate, including violin, golf and tennis. These actions generally show high average and maximum errors, especially golf and violin, where the error values are much higher than other tasks. The reason for the difficulty in imitation is that the actions involve complex multi-joint coordinated control, accompanied by significant trunk twisting and lower limb movements. The results above offer us insights into the robot itself,

we may not deny that it is the structural bottleneck (each leg only has one ankle joint) that severely limits its body coordination ability, leading to significant deviations in the imitation process.

Considering the joint parameters displayed in Table. 1, one of the important reasons for this phenomenon is that the ankle of the current robot platform (Unitree-H1) has only one degree of freedom (ankle pitch), which limits the flexibility of center of gravity adjustment and gait control. In subsequent work, it is possible to consider introducing a hardware platform with higher joint freedom (such as Unitree-G1) to improve the imitation accuracy and control stability of highly dynamic and complex movements.



(a) Guitar (b) Violin

Fig. 6: Tracking error with time

To further explore the imitation consistency of actions in different time periods, we selected two tasks, guitar with the smallest compound error and violin with the largest compound error, and compared their tracking error with time(as shown in Fig. 6). We employ the position error and rotation error to quantify the deviations of the robot's movement from the desired track. Specifically, the former is calculated as the Euclidean distance between the robot's current position and the desired position, while the latter is determined by the angular difference between the robot's current orientation and the target orientation. The results show that guitar has a low tracking error in the entire time series(less than 0.1), reflecting the stability and robustness of the action imitation process. In contrast, violin shows significant deviations in the middle and later stages(more than 0.1), especially in its position tracking error, where there are significant oscillations and fluctuations, reflecting the limitation of the controller's ability to cope with multi-joint coordination and lower limb dynamic balance tasks.

4 Conclusion

This paper proposes a method for learning whole-body movements of humanoid robots based on instruction learning. By introducing human motion capture data

as feedforward signals and combining it with a reinforcement learning framework, unified learning and control of multiple movements are achieved. Compared with the traditional imitation learning method, this method uses reference trajectories as the initial strategy to guide the policy network to efficiently explore in the policy space, significantly improving the training convergence speed and reducing the risk of policy degradation. In addition, by building a simulation environment on the Unity RL Playground platform and adjusting the action switching frequency in combination with the curriculum learning strategy, the model's adaptability to multi-skill switching is further improved. The simulation results show that this method has good generalization ability in a variety of movements.

In order to further improve the imitation accuracy and stability of the system in highly complex action scenarios, future work will introduce a robot platform with higher lower limb freedom (such as Unitree G1) to enhance the overall action coordination ability and reduce the impact of structural constraints on strategy optimization, thereby achieving higher quality whole-body action learning and control.

5 References

1. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
2. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International Conference on Machine Learning (ICML), pp. 1861–1870. PMLR (2018)
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning (ICML), pp. 41–48 (2009)
4. Osa, T., Pajarinen, J., Neumann, G., Bagnell, J.A., Abbeel, P., Peters, J.: An algorithmic perspective on imitation learning. Foundations and Trends in Robotics 7(1-2), 1–179 (2018)
5. Takeda, T., Hirata, Y., Kosuge, K.: Dance step estimation method based on HMM for dance partner robot. IEEE Transactions on Industrial Electronics 54, 699–706 (2007)
6. Gams, A., Nemec, B., Ijspeert, A.J., Ude, A.: Coupling movement primitives: Interaction with the environment and bimanual tasks. IEEE Transactions on Robotics 30(4), 816–830 (2014)
7. Zhang, T., McCarthy, Z., Jow, O., Lee, D., Chen, X., Goldberg, K., Abbeel, P.: Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 5628–5635 (2018)
8. Ng, A.Y., Russell, S.: Algorithms for inverse reinforcement learning. In: Proceedings of the 17th International Conference on Machine Learning (ICML), vol. 1, no. 2, p. 2 (2000)
9. Krishnan, S., Garg, A., Liaw, R., Thananjeyan, B., Miller, L., Pokorny, F.T., Goldberg, K.: SWIRL: A sequential windowed inverse reinforcement learning algorithm for robot tasks with delayed rewards. The International Journal of Robotics Research 38(2–3), 126–145 (2019)

10. Peng, X.B., Abbeel, P., Levine, S., Van de Panne, M.: DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. ACM Transactions on Graphics (TOG) 37(4), 1–14 (2018)
11. Kuefler, A., Morton, J., Wheeler, T., Kochenderfer, M.: Imitating driver behavior with generative adversarial networks. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), pp. 204–211. IEEE (2017)
12. Cai, Q., Hong, M., Chen, Y., Wang, Z.: On the global convergence of imitation learning: A case for linear quadratic regulator. arXiv preprint arXiv:1901.03674 (2019)
13. Peng, X.B., Coumans, E., Zhang, T., Lee, T.W., Tan, J., Levine, S.: Learning agile robotic locomotion skills by imitating animals. arXiv preprint arXiv:2004.00784 (2020)
14. Ye, L., Li, J., Cheng, Y., Wang, X., Liang, B., Peng, Y.: From knowing to doing: learning diverse motor skills through instruction learning. arXiv preprint arXiv:2309.09167 (2023)
15. Ye, L., Li, R., Hu, X., Li, J., Xing, B., Peng, Y., Liang, B.: Unity RL Playground: A versatile reinforcement learning framework for mobile robots. arXiv preprint arXiv:2503.05146 (2025)
16. Juliani, A., Berges, V.P., Teng, E., Cohen, A., Harper, J., Elion, C., Lange, D.: Unity: A general platform for intelligent agents. arXiv preprint arXiv:1809.02627 (2018)
17. Liu, Q., Guo, J., Lin, S., Ma, S., Zhu, J., Li, Y.: MASQ: Multi-agent reinforcement learning for single quadruped robot locomotion. arXiv preprint arXiv:2408.13759 (2024)
18. Zhao, Z., Huang, H., Sun, S., Li, C., Xu, W.: Fusing dynamics and reinforcement learning for control strategy: Achieving precise gait and high robustness in humanoid robot locomotion. In: Proceedings of the 2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids), pp. 1072–1079. IEEE (2024)
19. Zhou, Q., Li, G., Tang, R., Xu, Y., Wen, H., Shi, Q.: Stable jumping control based on deep reinforcement learning for a locust-inspired robot. Biomimetics 9(9), 548 (2024)
20. Darvish, K., Tirupachuri, Y., Romualdi, G., Rapetti, L., Ferigo, D., Chavez, F.J.A., Pucci, D.: Whole-body geometric retargeting for humanoid robots. In: 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), pp. 679–686. IEEE (2019)
21. Cisneros-Limón, R., Dallard, A., Benallegue, M., Kaneko, K., Kaminaga, H., Gergondet, P., Kheddar, A.: A cybernetic avatar system to embody human telepresence for connectivity, exploration, and skill transfer. International Journal of Social Robotics, 1–28 (2024)
22. Radosavovic, I., Zhang, B., Shi, B., Rajasegaran, J., Kamat, S., Darrell, T., Malik, J.: Humanoid locomotion as next token prediction. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS) (2024)
23. He, T., Luo, Z., Xiao, W., Zhang, C., Kitani, K., Liu, C., Shi, G.: Learning human-to-humanoid real-time whole-body teleoperation. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 8944–8951. IEEE (2024)
24. He, T., Luo, Z., He, X., Xiao, W., Zhang, C., Zhang, W., Shi, G.: OmniH2O: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. arXiv preprint arXiv:2406.08858 (2024)
25. Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., Black, M. J. (2019). AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 5442-5451).