SCIENCE CHINA
Information Sciences

• LETTER •

February 2026, Vol. 69, Iss. 2, 129208:1–129208:2
https://doi.org/10.1007/s11432-024-4755-6

Domain-aware behavior cloning for bridging the sim-to-real gap of legged robots

Yong ZHOU, Jiawei JIANG*, Bo DU, Hengyi YANG & Qianfan HU

School of Computer Science, Wuhan University, Wuhan 430072, China

Received 23 December 2024/Revised 7 June 2025/Accepted 5 August 2025/Published online 19 January 2026

Citation Zhou Y, Jiang J W, Du B, et al. Domain-aware behavior cloning for bridging the sim-to-real gap of legged robots. Sci China Inf Sci, 2026, 69(2): 129208, https://doi.org/10.1007/s11432-024-4755-6

Locomotion controllers play a critical role in enabling robots to perform complex behaviors in dynamic environments. Recent advances in reinforcement learning have demonstrated significant potential for developing more efficient and effective robotic controllers. Peng et al. [1] proposed a paradigm to train a locomotion policy in simulation and then transfer it to the real world. Environmental and dynamic factors, which can also be summarized as domain factors, have been taken into consideration to train more robust policies [2–5]. These methods can result in a robust policy that can adjust the gait to adapt to perturbations in the simulation that may also occur in the real world, such as random push, adding mass, and slippery or rocky terrains. However, these methods rely on the choice of the range of the domain factors. If the chosen range does not cover the real-world scenario, the policy will not be effective.

In this study, we demonstrate a new perspective on viewing the domain gap. Intuitively, the researchers deploy the trained policy on a real robot only when the performance is acceptable in simulation. Thus, the goal of bridging the sim-to-real gap is to replicate the accepted performance to different sets of domain factors. Motivated as such, we present a method, called domain-aware behavior cloning (DABC), which first trains a base policy in a specific environment and then trains an adapting policy to generalize to new environments with different domain factors. The key distinction between our proposed method and the typical teacher-student framework used by [3, 4] is that we train the adaptive policy with reinforcement learning rather than supervised learning. Reinforcement learning allows the adaptive policy to explore a wider spectrum of environmental conditions than the base policy, which results in a more resilient policy for deployment. We evaluate the effectiveness of the method using a quadrupedal robot that walks in various environments in both simulation and the real world. Compared to baseline methods based on domain identification and adaptation, DABC demonstrates superior performance in command following and terrain navigation in the real world.

Our proposed method involves two phases: training a base locomotion policy using a specific environment and cloning its behavior to other environments with varied unknown domain factors with an adapting policy.

In training the base locomotion policy, the key components of reinforcement learning are defined as follows.

Agent (Phase 1). The agent π_{base} generates desired joint angles for legged robots with the current state $s_t \in \mathbf{R}^{48}$ and the embedding e_t of domain factors $z_t \in \mathbf{R}^{32}$. The agent is in the form of multi-layer perceptrons (MLPs) sized $[512, 256, 128]$ and trained through interactions with the environments.

Environment (Phase 1). The environment provides the states s_t to the agent and transitions to the next state based on the agent's actions a_t. The state s_t comprises commands $c_t \in \mathbf{R}^3$, the previous action $a_{t-1} \in \mathbf{R}^{12}$, gravity $g_t \in \mathbf{R}^3$, heading $h_t \in \mathbf{R}^3$, angular velocity $w_t \in \mathbf{R}^3$, joint position $q_t \in \mathbf{R}^{12}$, and joint velocity $\dot{q}_t \in \mathbf{R}^{12}$. The commands c_t refer to the task target, including velocity tracking in the x-y plane and the target heading direction. The actions a_t are the desired joint positions of the actuators. The embedding of domain factors e_t is encoded from ground truth domain factors z_t including friction f, the robot's external mass m, the robot's center of mass x_{com}, and the K_P, K_D values of the actuators.

Reward (Phase 1). We design the reward to achieve the goal and prevent harmful behaviors. The goal includes tracking the command and avoiding falls. The penalty includes energy consumption, motor torques, action scales, contacts on robot links and feet contact forces. We designed N_r reward functions and used their weighted sum as the final reward $r_t = \Sigma_{i=0}^{N_r} f_i(s_t)$.

There are two modules to be learned for a locomotion policy

$$e_t = \mu(z_t), \tag{1}$$

$$a_t = \pi_{base}(s_t, e_t). \tag{2}$$

An autoencoder μ is trained to minimize the reconstruction loss of the input domain factors while the agent π is trained to optimize the accumulated reward

$$\max J_\pi = \max E_{\tau\ p(\tau|\pi)}[\Sigma_{t=0}^{T-1} \gamma^t r_t], \tag{3}$$

where τ is the trajectory collected from rollouts of simulated robots, which is defined as $\tau_t = (x_t, x_{t+1}, a_t, r_t)$. The first training phase involves training the locomotion policy with reinforcement learning and training the autoencoder for the domain factors with supervised learning. Then, the locomotion policy and the autoencoder will be frozen to train the adapting policy in the second training phase.

* Corresponding author (email: jiawei.jiang@whu.edu.cn)

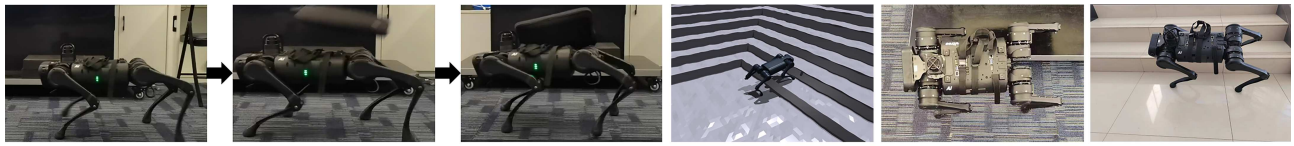

Figure 1 (Color online) Environment of our experiment. (a) Locomotion under disturbances. The robot is commanded to walk with 0.7 m/s, and a 3-kg payload will be dropped during operation. (b) Simulated stairs. The robot is trained to walk on stairs with a curriculum of speeds. (c) Real stairs with an 11-cm height. The robot is required to walk from plain ground onto a step with an 11-cm height. (d) Real stair with a 14-cm height.

The second training phase is designed to enable the agent to mimic its trajectory across a wider range of domain factors. We refer to the specified domain factors from which the agent is cloned as the source domain. The unknown domain to which the agent is being adapted is called the target domain. To achieve this goal, we model the task as a Markov decision process (MDP), highlighting three major differences compared to the agent trained specifically for locomotion.

Agent (Phase 2). Since our goal is to clone the agent's trajectory within a specific set of domain factors, the source domain factors are observable by the agent. Therefore, we include the embedding of the source domain factors in the state s_t. With this setup, we can obtain the reference action

$$a_t^{ref} = \pi(s_t, e_{t-1}). \tag{4}$$

Our primary objective in the adapting policy is to compute the residual action a^{res}, which is a complement of the original action with a smaller absolute scale. This setting prevents the exploration of the adapting policy harming the performance of the base locomotion policy too much, which may result in falling. The intuition of using residual action has also been applied in several studies to stabilize training and facilitate faster convergence [2–4]. We confine the range of the residual action to 0.1, 0.15, and 0.2 for the hip, thigh, and calf joints, respectively, according to their maximum range of motion. Finally, the agent interacts with the environment by executing the action summation $a_t^{adapt} = a_t^{ref} + a_t^{res}$.

Environment (Phase 2). State history, which consists of joint motion and action history, can be an effective source for estimating the motor dynamics [3]. We use the history, which is defined by $s_t^{adapt} = \{s_{t-n}, \ldots, s_{t-1}, s_t\}$ to perceive the dynamic in the target domain. Thus, the residual action given by the adapting policy can be computed by

$$a_t^{res} = \pi^{adapt}(s_t^{adapt}). \tag{5}$$

Reward (Phase 2). The base locomotion policy in Phase 1 is trained to optimize the expected return in the source domain while the adapting policy in Phase 2 is specifically crafted to replicate the trajectory from the source domain to the target domain. Using the same action given by the same base locomotion policy trained in Phase 1, the agents in the source and the target domain will reach different subsequent states determined by the state transition function.

$$s_{t+1} = P_{ss'}(s_t, a_t, z_t). \tag{6}$$

The goal of the residual action can be interpreted as minimizing the distance between the next states of agents that act in different environments given the same current state. Hence, the reward function can be designed to minimize the summation of the error in the source domain and the target domain including heading tracking, velocity tracking, joint position tracking, and joint velocity tracking.

Experiment. The trained adapting policy can be deployed directly on a real robot for evaluation. To evaluate the domain adapting capability of different methods, we design experiments in both simulation and real-world environments. We design velocity tracking tasks including free walking, walking under a dropped payload, and a terrain traversing task, which is shown in Figure 1. The payload weighs 3 kg and is dropped from 1.0 m above the robot after the robot has started walking for 3 s. We evaluate the performance of walking tasks with the mean squared error (MSE) between the commanded velocity and actual velocity. We report the success rate of walking without falling in 100 trials for terrain traversing tasks. Detailed results are demonstrated in Appendix A.

Conclusion and limitation. In this study, we propose a novel workflow of domain adaptation for quadrupedal robots. By learning to replicate the trajectory from the source domain to the target domain, the proposed adaptation mechanism achieved both a higher velocity tracking reward and a higher success rate for traversing stairs in the real world. Nevertheless, the distortion of robot feet is difficult to simulate by merely changing the domain factors used in the study. Such distortion can surpass the tolerance of the adaptation policy, causing locomotion failure.

Acknowledgements This work was supported by Key R&D Program of Hubei Province (Grant Nos. 2023BAB077, 2023BAB170) and National Natural Science Foundation of China (Grant No. 62472327).

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Peng X, Andrychowicz M, Zaremba W, et al. Sim-to-real transfer of robotic control with dynamics randomization. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2018. 3803–3810
2 Cheng X, Shi K, Agarwal A, et al. Extreme parkour with legged robots. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2024. 11443–11450
3 Kumar A, Fu Z, Pathak D, et al. RMA: rapid motor adaptation for legged robots. 2021. ArXiv:2107.04034
4 Lee J, Hwangbo J, Wellhausen L, et al. Learning quadrupedal locomotion over challenging terrain. Sci Robot, 2020, 5: eabc5986
5 Yu W, Tan J, Liu C K, et al. Preparing for the unknown: learning a universal policy with online system identification. 2017. ArXiv:1702.02453