Zhang *et al*. *Intell Robot* 2022;2(3):275-97
**DOI:** 10.20517/ir.2022.20

**Intelligence & Robotics**

Review

Open Access

# Deep reinforcement learning for real-world quadrupedal locomotion: a comprehensive review

**Hongyin Zhang, Li He, Donglin Wang**

School of Engineering, Westlake University, Hangzhou 310000, Zhejiang, China.

**Correspondence to:** Dr. Donglin Wang, School of Engineering, Westlake University, Dunyu Road No.600, Xihu District, Hangzhou 310000, Zhejiang, China. E-mail: wangdonglin@westlake.edu.cn; ORCID: 0000-0002-8188-3735

**How to cite this article:** Zhang H, He L, Wang D. Deep reinforcement learning for real-world quadrupedal locomotion: a comprehensive review. *Intell Robot* 2022;2(3):275-97. http://dx.doi.org/10.20517/ir.2022.20

**Received:** 30 Jun 2022 **First Decision:** 25 Jul 2022 **Revised:** 27 Jul 2022 **Accepted:** 22 Aug 2022 **Published:** 1 Sep 2022

**Academic Editor:** Simon X. Yang **Copy Editor:** Jia-Xin Zhang **Production Editor:** Jia-Xin Zhang

## Abstract

Building controllers for legged robots with agility and intelligence has been one of the typical challenges in the pursuit of artificial intelligence (AI). As an important part of the AI field, deep reinforcement learning (DRL) can realize sequential decision making without physical modeling through end-to-end learning and has achieved a series of major breakthroughs in quadrupedal locomotion research. In this review article, we systematically organize and summarize relevant important literature, covering DRL algorithms from problem setting to advanced learning methods. These algorithms alleviate the specific problems encountered in the practical application of robots to a certain extent. We first elaborate on the general development trend in this field from several aspects, such as the DRL algorithms, simulation environments, and hardware platforms. Moreover, core components in the algorithm design, such as state and action spaces, reward functions, and solutions to reality gap problems, are highlighted and summarized. We further discuss open problems and propose promising future research directions to discover new areas of research.

**Keywords:** Deep reinforcement learning, quadrupedal locomotion, reality gap

## 1. INTRODUCTION

Wheeled and tracked robots are still unable to navigate the most challenging terrain in the natural environment, and their stability may be severely compromised. Quadrupedal locomotion, on the other hand, can greatly expand the agility of robot behavior, as legged robots can choose safe and stable footholds within their kinematic

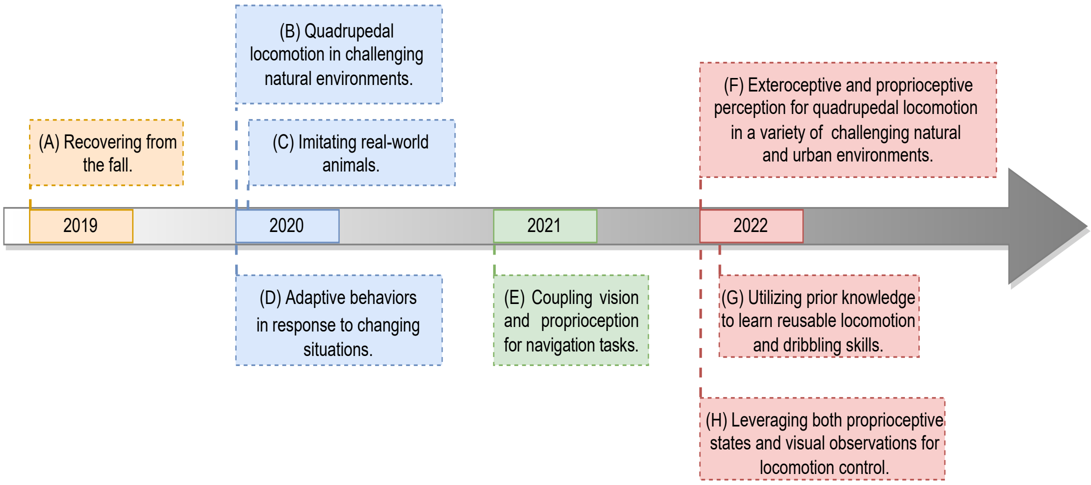

(B) Quadrupedal locomotion in challenging natural environments.
(A) Recovering from the fall.
(C) Imitating real-world animals.
(F) Exteroceptive and proprioceptive perception for quadrupedal locomotion in a variety of challenging natural and urban environments.
2019
2020
2021
2022
(D) Adaptive behaviors in response to changing situations.
(E) Coupling vision and proprioception for navigation tasks.
(G) Utilizing prior knowledge to learn reusable locomotion and dribbling skills.
(H) Leveraging both proprioceptive states and visual observations for locomotion control.

**Figure 1.** Several typical quadrupedal locomotion studies based on DRL algorithm: (A) recovering from a fall[6]; (B) a radically robust controller for quadrupedal locomotion in challenging natural environments[7]; (C) learning agile locomotion skills by imitating real-world animals[10]; (D) producing adaptive behaviors in response to changing situations[9]; (E) coupling vision and proprioception for navigation tasks[11]; (F) integrating exteroceptive and proprioceptive perception for quadrupedal locomotion in a variety of challenging natural and urban environments over multiple seasons[8]; (G) utilizing prior knowledge of human and animal movement to learn reusable locomotion and dribbling skills[12]; and (H) leveraging both proprioceptive states and visual observations for locomotion control[13].

reach and rapidly change the kinematic state according to the environment. To further study quadrupedal locomotion on uneven terrain, the complexity of traditional control methods is gradually increased as more scenarios are considered[1–4]. As a result, the associated development and maintenance becomes rather time-consuming and labor-intensive, and it remains vulnerable to extreme situations.

With the rapid development of the artificial intelligence field, deep reinforcement learning (DRL) has recently emerged as an alternative method for developing legged motor skills. The core idea of DRL is that the control policy learns to make decisions to obtain the maximum benefit based on the reward received from the environment[5]. DRL has been used to simplify the design of locomotion controllers, automate parts of the design process, and learn behaviors that previous control methods could not achieve[6–9]. Research on DRL algorithms for legged robots has gained wide attention in recent years. Meanwhile, several well-known research institutions and companies have publicly revealed their implementations of DRL-based legged robots, as shown in Figure 1.

Currently, there are several reviews on applying DRL algorithms to robots. Some works summarize the types of DRL algorithms and deployment on several robots such as robotic arms, bipeds, and quadrupeds[14]. They discuss in detail the theoretical background and advanced learning algorithms of DRL, as well as present key current challenges in this field and ideas for future research directions to stimulate new research interests. There is also a work summarizing some case studies involving robotic DRL and some open problems[15]. Based on these case studies, they discuss common challenges in DRL and how the work addresses them. They also provide an overview of other prominent challenges, many of which are unique to real-world robotics settings. Furthermore, a common paradigm for DRL algorithms applied to robotics is to train policies in simulations and then deploy them on real machines. This can lead to the reality gap[16] (also known as sim-to-real gap) problem, which is summarized for the robotic arm in[17]. These reviews introduce the basic background behind sim-to-real transfer in DRL and outline the main methods currently used: domain randomization, domain adaptation, imitation learning, meta-learning, and knowledge distillation. They categorize some of the most relevant recent works and outline the main application scenarios while also discussing the main opportunities and challenges of different approaches and pointing out the most promising directions. The closest work to our review simply surveys current research on motor skills learning via DRL algorithms[18], without systematically combing through the relevant literature and without an in-depth analysis of the existing open problems and future research directions.

In this survey, we focus on quadrupedal locomotion research from the perspective of algorithm design, key challenges, and future research directions. The remainder of this review is organized as follows. Section 2 formulates the basic settings in DRL and lists several important issues that should be alleviated. The classification and core components of the current algorithm design (e.g., the DRL algorithm, simulation environment, hardware platform, observation, action, and reward function) are introduced in Section 3. Finally, we summarize and offer perspectives on potential future research directions in this field.

## 2. BASIC SETTINGS AND LEARNING PARADIGM

In this section, we first formulate the basic settings of standard reinforcement learning problems and then introduce the common learning paradigm.

Quadrupedal locomotion is commonly formulated as a reinforcement learning (RL) problem, which in the framework of Markov decision processes (MDPs) is specified by the tuple $M := (\mathcal{S}, \mathcal{A}, R, P, \rho_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively; $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function; $P(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ is the stochastic transition dynamics; $\rho_0(\mathbf{s})$ is the initial state distribution; and $\gamma \in [0, 1]$ is the discount factor. The objective is to learn a control policy $\pi$ that enables a legged robot to maximize its expected return for a given task[19]. A state $s_t$ is observed by the robot from the environment at each time step $t$, and an action $\mathbf{a}_t \sim \pi(\mathbf{a}_t \mid \mathbf{s}_t)$ is derived from robot's policy $\pi$. The robot next applies this action, which results in a novel state $s_{t+1}$ and a scalar reward $r_t = R(\mathbf{s}_t, \mathbf{a}_t)$. As a result, a trajectory $\tau := (\mathbf{s}_0, \mathbf{a}_0, r_0, \mathbf{s}_1, \mathbf{a}_1, r_1, \ldots)$ is obtained by repeating applications of this interaction process. Formally, the RL problem requires the robot to learn a decision making policy $\pi(\mathbf{a}|\mathbf{s})$ that maximizes the expected discounted return:

$$\mathcal{J}(\pi) := \mathbb{E}_{\tau \sim p^\pi(\tau)}\left[\sum_{t=0}^{H-1} \gamma^t r_t\right], \tag{1}$$

where $H$ denotes the time horizon of each episode and $p^\pi(\tau) = p(\mathbf{s}_0) \prod_{t=0}^{H-1} p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\, \pi(\mathbf{a}_t \mid \mathbf{s}_t)$ represents the likelihood of a trajectory $\tau$ under a given policy $\pi$, with $p(\mathbf{s}_0)$ being the initial state distribution.

For quadrupedal locomotion tasks, most of the current research is based on a similar learning paradigm, as shown in Figure 2. First, we need to build a simulation environment (e.g., ground, steps, and stairs), and then design the state and action space, reward function and other essential elements. DRL-based algorithms are further designed and used to train policies in the simulation. The trained policy is finally deployed on the real robot to complete the assigned task.

## 3. DRL-BASED CONTROL POLICY DESIGN FOR QUADRUPEDAL LOCOMOTION

In this section, we detail the key components of a DRL-based controller. The classification results are presented in Tables 1 and 2 in the Appendix. After the most relevant publications in this field are summarized, their key parts are further condensed. As shown in Figure 3, we firstly review and analyze the general state and development trend of current research (e.g., DRL algorithms, simulators, and hardware platforms). Then, important components of DRL algorithm (state and action design, reward function design, solution to reality gap, *etc.*) are presented, as shown in Figure 4. These specific designs would help to alleviate open questions, which are further discussed in Section 4. Please refer to the Appendix for more details.

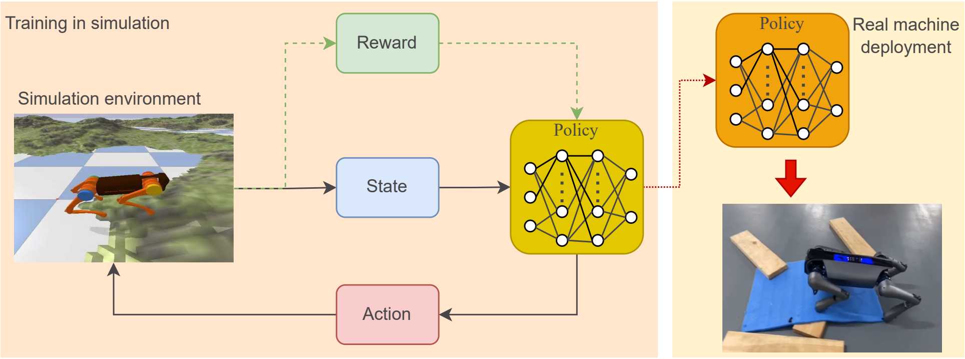
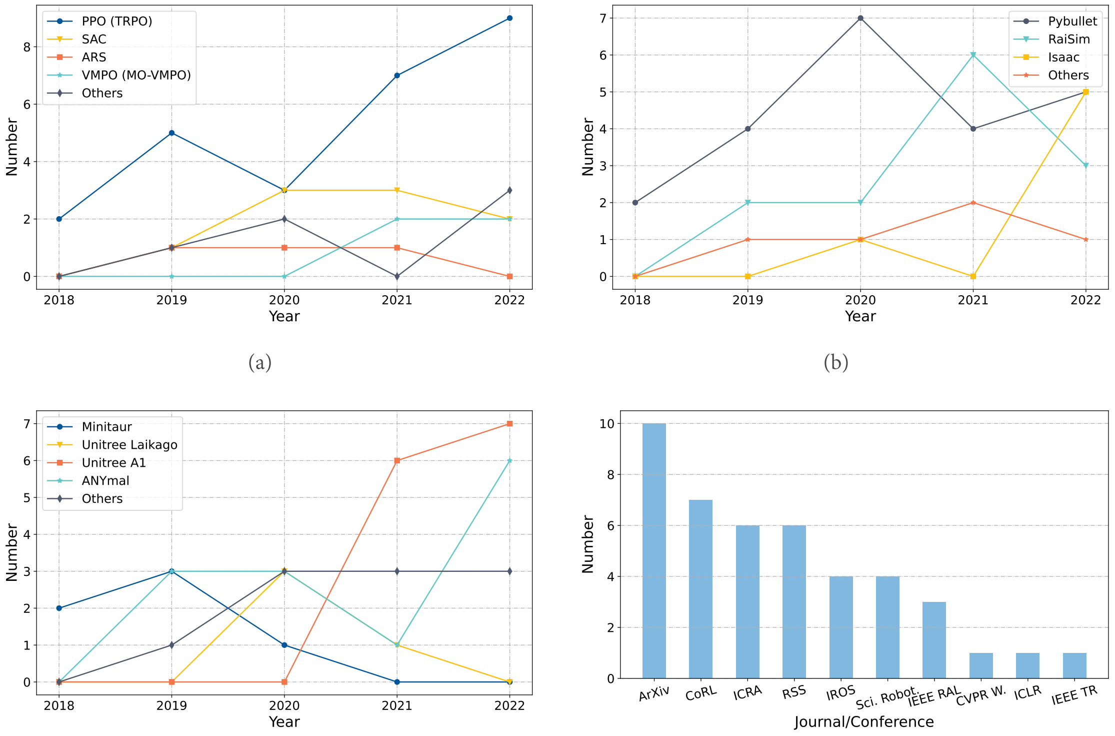


**Figure 2.** A common paradigm for DRL-based quadrupedal locomotion research. This paradigm is mainly divided into training and testing phases. The policy interacts with the simulated environment and collects data for iterative updates, and then the trained policy is deployed to the real robot.



(a)

(b)

(c)

(d)

**Figure 3.** Several statistical results from important papers on quadrupedal locomotion research. A full summary of classification results of the most relevant publications is presented in Tables 1 and 2 in the Appendix. These papers were selected from journals and conferences (ArXiv, CoRL, ICRA, RSS, IROS, Science Robotics, ICLR, *etc.*) in recent years. (a-c) Trends in the usage times of several DRL-based algorithms, simulation platforms, and real robots. The x and y axes represent the year and the number used, respectively. (d) Number of papers accepted by the journal or conference. The x and y axes represent journals (or conferences) and the number of papers published, respectively.

### 3.1. DRL algorithm

Although many novel algorithms have been developed in the DRL community, most current quadrupedal locomotion controller designs still use model-free DRL algorithms, especially PPO and TRPO[20,21]. For a complex high-dimensional nonlinear system such as robots, stable control is the fundamental purpose. Most researchers choose the PPO (TRPO) algorithm for utilization in their research due to its simplicity, stability, theoretical justification, and empirical performance[20–22].

Similar to on-policy algorithms, PPO (TRPO) has been criticized for its sample inefficiency; thus, more efficient model-free algorithms (ARS[23], SAC[24], V-MPO[25], *etc.*) are sometimes considered. Some researchers have also recently used advanced algorithms for more challenging tasks. For example, the multi-objective variant of the VMPO algorithm (MO-VMPO)[26] has been utilized to train a policy to track the planned trajectories[27]. Some researchers have introduced guided constrained policy optimization (GCPO) method for tracking base velocity commands while following defined constraints[28]. Moreover, for more efficient real-world fine-tuning and to avoid overestimation problems, REDQ, an off-policy algorithm[29], is used for real robots[30].

### 3.2. Simulator

The robot simulator should be able to simulate the dynamic physical laws of the robot itself more realistically and efficiently solve the collisions generated when the robot interacts with the environment. Over the past few years, the Pybullet[31] and RaiSim[32] simulation platforms have been the choice of most researchers. However, the current robotic simulators in academia are still relatively simple, and the precision is far less than that of simulators in games. For robots, directly realizing end-to-end decision making from perception to control is difficult without an accurate and realistic simulator. Common robotic simulators, such as Pybullet and RaiSim, can only solve control-level simulations, but they are stretched for real-world simulations. They have been developed to run on CPUs with reduced parallelism. On the other hand, while mujoco[33] is a popular DRL algorithm verification simulator, it is rarely used as a deployment and testing platform for real-world quadrupedal locomotion algorithms. A possible explanation is that the highly encapsulated mujoco simulator makes it difficult for researchers to develop it further.

Recently, NVIDIA released a new simulator, Isaac Gym[34], which simulates the environment with much higher accuracy than the aforementioned simulators, and can simulate and train directly on GPUs. This simulator is scalable and can simulate a large number of scenarios in parallel, so researchers can use DRL algorithms for large-scale training. It can also build large-scale realistic complex scenes, and its underlying PhysX engine can accurately and realistically model and simulate the motion of objects. Therefore, more researchers have begun to use Isaac Gym as the implementation and verification platform of DRL algorithm[35–38].

### 3.3. Hardware platform

In the early research stage, *Minitaur*[39] with only eight degrees of freedom was used to verify the feasibility of the DRL algorithm in simple experimental scenarios. To accomplish more complex tasks, robots (Unitree Laikago, Unitree A1, *ANYmal*[40], *etc.*) with more than 12 degrees of freedom are used by researchers. While the *ANYmal* series robots are well known for their high hardware costs, low-cost robots such as *Unitree A1* are a more prevailing choice among researchers. Lower-cost hardware platforms allow DRL algorithms to be more widely used. More recently, a wheel-legged quadruped robot[38] demonstrated skills learned from existing DRL controllers and trajectory optimization, such as ducking and walking, and new skills, such as switching between quadrupedal and humanoid configurations.

### 3.4. Publisher

Currently, DRL-based quadrupedal locomotion research is an emerging and promising field, and many papers have not been officially published. The published papers are mainly in journals or conferences related to the field of robotics, and there are four outstanding works[6–9] published on Science Robotics. It is worth noting that the field is actually an intersection of several fields, and some excellent studies have been published at

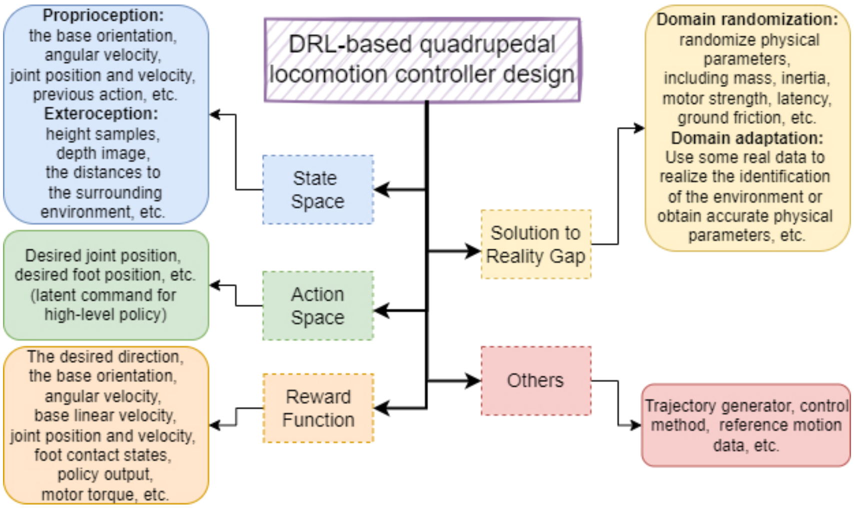

Proprioception:
the base orientation, angular velocity, joint position and velocity, previous action, etc.
Exteroception:
height samples, depth image, the distances to the surrounding environment, etc.
DRL-based quadrupedal locomotion controller design
Domain randomization:
randomize physical parameters, including mass, inertia, motor strength, latency, ground friction, etc.
Domain adaptation:
Use some real data to realize the identification of the environment or obtain accurate physical parameters, etc.
State Space
Solution to Reality Gap
Desired joint position, desired foot position, etc. (latent command for high-level policy)
Action Space
The desired direction, the base orientation, angular velocity, base linear velocity, joint position and velocity, foot contact states, policy output, motor torque, etc.
Reward Function
Others
Trajectory generator, control method, reference motion data, etc.

**Figure 4.** The key components of the DRL-based controller design from the classification result of the most relevant publications. Tables 1 and 2 in the Appendix provide a completed summary.

conferences in the machine learning field.

### 3.5. State, action, reward, and others

State, action, and reward are integral and important components for training controllers. The design of these components will directly affect the performance of the controller. However, there is no fully unified standard and method for the specific design.

For the design of state space, on the one hand, considering too few observations can lead to a partially observable controller. On the other hand, providing all available readings results in a brittle controller that is overfitted to the simulation environment. Both affect the performance of the controller in the real machine, so researchers can only make trade-offs based on practical problems. In current research works, for simple tasks (walking, turning on flat ground, *etc.*), proprioception alone (base orientation, angular velocity, joint position and velocity, *etc.*) is sufficient to solve the problem[10,39,41]. For more complex tasks (walking on uneven ground, climbing stairs or hillsides, avoiding obstacles, *etc.*), exteroception, such as visual information, needs to be introduced[8,13,42]. Adding additional sensors alleviates the partial observation issues to some extent.

Most researchers use the desired joint positions (residuals) as the action space and then calculate the torque through a PD controller to control the robot locomotion. Early studies[43] experimentally demonstrated that controllers with such action space can achieve better performance. However, recent studies also attempt to use lower-level control commands to obtain highly dynamic motion behavior to avoid the use of PD controllers and control torque directly[44]. Although the current DRL-based controllers have achieved outstanding performance[6–8], their stability is still not as good as the common control methods, such as MPC controllers[45]. The force–position hybrid control method adopted by MPC is worthy of reference and further research. Furthermore, in some studies based on hierarchical DRL, the latent commands serve as the action space of the high-level policy to guide the behavior of low-level policies[46,47].

In general, the design of the reward function is fairly laborious, especially for complex systems such as robots. Small changes in the reward function hyperparameters have the potential to have a large impact on the final performance of the controller. In order for the robot to complete more complex tasks, the reward function must be designed with sufficient detail[6–8,48]. Some specific factors include the desired direction, base orientation, angular velocity, base linear velocity, joint position and velocity, foot contact states, policy output, and motor

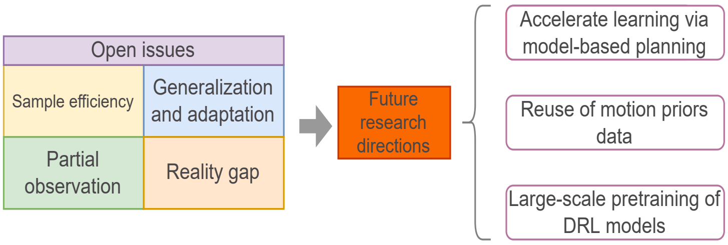

Open issues
Sample efficiency
Generalization and adaptation
Partial observation
Reality gap
Future research directions
Accelerate learning via model-based planning
Reuse of motion priors data
Large-scale pretraining of DRL models

**Figure 5.** In the DRL-based real-world quadrupedal locomotion field, open problems mainly include sample efficiency, generalization and adaptation, partial observation, and reality gap. Future research directions are highlighted and pointed out around these open problems. Based on the current research states of quadrupedal locomotion, we expound the future research prospects from multiple perspectives. In particular, world models, skill data, and pre-trained models require significant attention, as these directions will play an integral role in realizing legged robot intelligence.

torque.

Many studies have also considered additional information, such as trajectory generators [46,49–51], control methods [52–54], motion data [10,12,55,56], *etc.* Trajectory generators and control methods mainly introduce prior knowledge in the action space, narrowing the search range of DRL control policies, which greatly improves the sample efficiency under a simple reward function. Motion data are often generated by other suboptimal controllers or assessed via public datasets. Through imitation learning based on the motion data, the robot can master behaviors and skills such as walking and turning. In both simulations and real-world deployment, the robot eventually manages to generate natural and agile movement patterns and completes the assigned tasks according to the external reward function.

### 3.6. Solution to reality gap

Under the current mainstream learning paradigm, the reality gap is an unavoidable problem that must be addressed. The domain randomization method is used by most researchers due to its simplicity and effectiveness. The difference between simulation and real environment is mainly reflected in physical parameters and sensors. Therefore, researchers mainly randomize physical parameters (mass, inertia, motor strength, latency, ground friction, *etc.*), add Gaussian noise to observations, and apply disturbing force, *etc.* [35,48,50,57,58]. However, domain randomization methods trade optimality for robustness, which can lead to conservative controllers [59]. Some studies have also used domain adaptation methods, that is, use real data to identify the environment [60,61] or obtain accurate physical parameters [62]. Furthermore, these methods can improve the generalization (adaptation) performance of robots in challenging environments. For more solutions to the reality gap, please refer to the relevant review paper [63].

## 4. OPEN PROBLEMS AND FUTURE PROSPECTS

In this section, we discuss the long-standing open questions and promising future research directions in the DRL-based quadrupedal locomotion field around these issues, as shown in Figure 5. Solutions to these open problems are described in Section 3.

### 4.1 Open problems

*4.1.1. Sample efficiency*

In many popular DRL algorithms, millions or billions of gradient descent steps are required to train policies that can accomplish the assigned task [64–66]. For real robotics tasks, therefore, such a learning process requires a significant number of interactions, which is infeasible in practical applications. In the face of increasingly complex robotic tasks, without improvement in the sample efficiency of algorithms, the number of training samples needed will only increase with model size and complexity. Furthermore, a sample-efficient DRL algorithm can deal with sparse-reward tasks, which greatly reduces the difficulty of designing reward functions. It also alleviates the serious time burden for the researchers to tune the parameters of reward function.

#### *4.1.2. Generalization and adaptation*

Generalization is another fundamental problem of the DRL algorithm. Current algorithms perform well in single-task and static environments, but they struggle with multi-task and dynamically unstructured environments. That is, it is difficult for robots to acquire novel skills and quickly adapt to unseen environments or tasks. Generalization or adaptation to new scenarios remains a long-standing unsolved problem in the DRL community. In general, there are two broad categories of problems in robotics tasks: the observational generalization (adaptation) problem and the dynamic generalization (adaptation) problem. The former is a learning problem for robots considering high-dimensional state spaces, such as raw visual sensor observations. High-dimensional observations may incorporate redundant, task-irrelevant information that may impair the generalization ability of robot learning. Currently, there are many related studies published on physical manipulation[67–71] but only a few cutting-edge works on quadrupedal locomotion tasks[8,11,13]. The latter mainly takes into account the dynamic changes of the environment (e.g., robot body mass and ground friction coefficient)[72–74]. This causes the transition probability of the environment to change, i.e., the robot takes the same action in the same state, but it transitions to a different next state.

#### *4.1.3. Partial observation*

Simulators can significantly reduce the training difficulty of the DRL algorithms because we have access to the ground-truth state of the robots. However, due to the limitations of the onboard sensors of real robots, the policies are limited to partial observations that are often noisy and delayed. For example, it is difficult to accurately measure the root translation and body height of a legged robot. This problem is more pronounced when faced with locomotion or navigation tasks in complex and unstructured environments. Several approaches have been proposed to alleviate this problem, such as applying system identification[75], removing inaccessible states during training[39], adding more sensors[8,11,13], and learning to infer privileged information[7,76].

#### *4.1.4. Reality gap*

This problem is caused by differences between the simulation and real-world physics[16]. There are many sources of this discrepancy, including incorrect physical parameters, unmodeled dynamics, and random real-world environments. Furthermore, there is no general consensus on which of these sources plays the most important role. A straightforward approach is domain randomization, a class of methods that uses a wide range of environmental parameters and sensor noises to learn robust robot behaviors[39,77,78]. Since this method is simple and effective, most studies on quadrupedal locomotion have used it to alleviate the reality gap problem.

### 4.2 Future prospects

#### *4.2.1. Accelerate learning via model-based planning*

For sequential decision making problems, model-based planning is a powerful approach to improve sample efficiency and has achieved great success in applied domains such as game playing[79–81] and continuous control[82,83]. These methods, however, are both costly to plan over long horizons and struggle to obtain accurate world models. More recently, the strengths of model-free and model-based methods are combined to achieve superior sample efficiency and asymptotic performance on continuous control tasks[84], especially on fairly challenging, high-dimensional humanoid and dog tasks[85]. How to use model-based planning in DRL-based quadrupedal locomotion research is an issue worthy of further exploration.

#### *4.2.2. Reuse of motion priors data*

Current vanilla DRL algorithms have difficulty producing life-like natural behaviors for legged robots. Furthermore, reward functions capable of accomplishing complex tasks often require a tedious and labor-intensive tuning process. Robots also struggle to generalize or adapt to other environments or tasks. To alleviate this problem to a certain extent, there have been recent DRL studies based on motion priors[86–90], which have been successfully applied to quadrupedal locomotion tasks[12,56,91]. However, the variety of motion priors in these studies is insufficient, and the robot's behavior is not agile and natural. This makes it difficult for robots to cope with complex and unstructured natural environments. Improving the diversity of motion priors is also an interesting direction in quadrupedal locomotion research. On the other hand, there is currently a lack of general real-world legged motion skills datasets and benchmarks, which would have significant value for DRL-based quadrupedal locomotion research. If many real-world data were available, we could study and verify offline RL[92] algorithms for quadrupedal locomotion. The main feature of offline RL algorithms is that the robot does not need to interact with the environment during the training phase, so we can bypass the notorious reality gap problem.

### *4.2.3. Large-scale pre-training of DRL models*

The pre-training and fine-tuning paradigms for new tasks have emerged as simple yet effective solutions in supervised and self-supervised learning. Pre-trained DRL-based models enable robots to rapidly and efficiently acquire new skills and respond to non-stationary complex environments. Meta-learning methods seem to be a popular solution for improving the generalization (adaptation) performance of robots to new environments. However, current meta-reinforcement learning algorithms are limited to simple environments with narrow task distributions[93–96]. A recent study showed that multi-task pre-training with fine-tuning on new tasks performs as well as or better than meta-pre-training with meta test-time adaptation[97]. Research considering large-scale pre-trained models in quadrupedal locomotion research is still in its infancy and needs further exploration. Furthermore, this direction is inseparable from the motor skills dataset mentioned above, but it focuses more on large-scale pre-training of DRL-based models and online fine-tuning for downstream tasks.

## 5. CONCLUSIONS

In the past few years, there have been some breakthroughs in quadrupedal locomotion research. However, due to the limitations of algorithms and hardware, the behavior of robots is still not agile and intelligent. This review provides a comprehensive survey of several DRL algorithms in this field. We first introduce basic concepts and formulations, and then condense open problems in the literature. Subsequently, we sort out previous works and summarize the algorithm design and core components in detail, which includes DRL algorithms, simulators, hardware platforms, observation and action space design, reward function design, prior knowledge, solution of reality gap problems, *etc*. While this review considers as many factors as possible in systematically collating the relevant literature, there are still many imperceptible factors that may affect the performance of DRL-based control policies in real-world robotics tasks. Finally, we point out future research directions around open questions to drive important research forward.

## DECLARATIONS

### Authors' contributions

Made substantial contributions to conception and design of the study and performed data analysis and interpretation: Zhang H, Wang D

Performed data acquisition, as well as provided administrative, technical, and material support: He L

### Availability of data and materials

Please refer to Table 1 and Table 2 in the appendix.

### Financial support and sponsorship

This work was supported by the National Science and Technology Innovation 2030 - Major Project (Grant No. 2022ZD0208800), and NSFC General Program (Grant No. 62176215).

### Conflicts of interest

All authors declared that there are no conflicts of interest.

### Ethical approval and consent to participate

Not applicable.

### Consent for publication

Not applicable.

### Copyright

## REFERENCES

1. Bledt G, Wensing PM, Ingersoll S, Kim S. Contact Model Fusion for Event-Based Locomotion in Unstructured Terrains. In: 2018 IEEE International Conference on Robotics and Automation (ICRA); 2018. pp. 4399–406. DOI
2. Hwangbo J, Bellicoso CD, Fankhauser P, Hutter M. Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2016. pp. 3872–78. DOI
3. Camurri M, Fallon M, Bazeille S, et al. Probabilistic contact estimation and impact detection for state estimation of quadruped robots. *IEEE Robotics and Automation Letters* 2017;2:1023–30. DOI
4. Bloesch M, Gehring C, Fankhauser P, et al. State estimation for legged robots on unstable and slippery terrain. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2013. pp. 6058–64. DOI
5. Sutton RS, Barto AG. Reinforcement learning: an introduction. *IEEE Trans Neural Netw* 2005;16:285–86. Available from: https://books.google.com.hk/books?hl=zh-CN&lr=&id=uWV0DwAAQBAJ&oi=fnd&pg=PR7&dq=Reinforcement+Learning:+An+Introduction&ots=mitFv1__l3&sig=tpgx07M0IomIGA4K13idTvhYFbo&redir_esc=y#v=onepage&q=Reinforcement%20Learning%3A%20An%20Introduction&f=false [Last accessed on 30 Aug 2022].
6. Hwangbo J, Lee J, Dosovitskiy A, et al. Learning agile and dynamic motor skills for legged robots. *Sci Robot* 2019;4. DOI
7. Lee J, Hwangbo J, Wellhausen L, Koltun V, Hutter M. Learning quadrupedal locomotion over challenging terrain. *Sci Robot* 2020;5. Available from: https://robotics.sciencemag.org/content/5/47/eabc5986 [last accessed on 30 Aug 2022].
8. Miki T, Lee J, Hwangbo J, et al. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci Robot* 2022;7:eabk2822. Available from: https://www.science.org/doi/abs/10.1126/scirobotics.abk2822 [Last accessed on 30 Aug 2022].
9. Yang C, Yuan K, Zhu Q, Yu W, Li Z. Multi-expert learning of adaptive legged locomotion. *Sci Robot* 2020;5. Available from: https://robotics.sciencemag.org/content/5/49/eabb2174 [Last accessed on 30 Aug 2022].
10. Peng XB, Coumans E, Zhang T, et al. Learning agile robotic locomotion skills by imitating animals. In: Robotics: Science and Systems; 2020. DOI
11. Fu Z, Kumar A, Agarwal A, et al. Coupling vision and proprioception for navigation of legged robots. *arXiv preprint arXiv:211202094* 2021. Available from: https://openaccess.thecvf.com/content/CVPR2022/html/Fu_Coupling_Vision_and_Proprioception_for_Navigation_of_Legged_Robots_CVPR_2022_paper.html [Last accessed on 30 Aug 2022].
12. Bohez S, Tunyasuvunakool S, Brakel P, et al. Imitate and repurpose: learning reusable robot movement skills from human and animal behaviors. *arXiv preprint arXiv:220317138* 2022. DOI
13. Yang R, Zhang M, Hansen N, Xu H, Wang X. *Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers* .
14. Zhang T, Mo H. Reinforcement learning for robot research: a comprehensive review and open issues. *Int J Advanc Robot Syst* 2021;18:17298814211007305. DOI
15. Ibarz J, Tan J, Finn C, et al. How to train your robot with deep reinforcement learning: lessons we have learned. *Int J Robot Res* 2021;40:698–721. DOI
16. Koos S, Mouret JB, Doncieux S. Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In: Conference on Genetic and Evolutionary Computation. United States: ACM, publisher; 2010. pp. 119–26. DOI Available from: https://hal.archives-ouvertes.fr/hal-00633927.
17. Zhao W, Queralta JP, Westerlund T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE; 2020. pp. 737–44. DOI
18. Yue J. Learning locomotion for legged robots based on reinforcement learning: a survey. In: 2020 International Conference on Electrical Engineering and Control Technologies (CEECT). IEEE; 2020. pp. 1–7. DOI

19. Sutton RS, Barto AG. *Introduction to reinforcement learning* 1998. Available from: https://login.cs.utexas.edu/sites/default/files/legacy_files/research/documents/1%20intro%20up%20to%20RL%3ATD.pdf [Last accessed on 30 Aug 2022].
20. Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. Trust region policy optimization. In: International conference on machine learning. PMLR; 2015. pp. 1889–97. Available from: https://proceedings.mlr.press/v37/schulman15.html [Last accessed on 30 Aug 2022].
21. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. *CoRR* 2017;abs/1707.06347. Available from: http://arxiv.org/abs/1707.06347 [Last accessed on 30 Aug 2022].
22. Schulman J, Moritz P, Levine S, Jordan MI, Abbeel P. High-dimensional continuous control using generalized advantage estimation. In: Bengio Y, LeCun Y, editors. 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings; 2016. Available from: http://arxiv.org/abs/1506.02438 [Last accessed on 30 Aug 2022].
23. Mania H, Guy A, Recht B. Simple random search provides a competitive approach to reinforcement learning. *CoRR* 2018;abs/1803.07055. Available from: http://arxiv.org/abs/1803.07055 [Last accessed on 30 Aug 2022].
24. Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Dy JG, Krause A, editors. Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. vol. 80 of Proceedings of Machine Learning Research. PMLR; 2018. pp. 1856–65. Avaialble from: http://proceedings.mlr.press/v80/haarnoja18b.html [Last accessed on 30 Aug 2022].
25. Song HF, Abdolmaleki A, Springenberg JT, et al. V-MPO: on-policy maximum a posteriori policy optimization for discrete and continuous control. OpenReview.net; 2020. Available from: https://openreview.net/forum?id=SylOlp4FvH [Last accessed on 30 Aug 2022].
26. Abdolmaleki A, Huang SH, Hasenclever L, et al. A distributional view on multi-objective policy optimization. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. vol. 119 of Proceedings of Machine Learning Research. PMLR; 2020. pp. 11–22. Avaialble from: http://proceedings.mlr.press/v119/abdolmaleki20a.html [Last accessed on 30 Aug 2022].
27. Brakel P, Bohez S, Hasenclever L, Heess N, Bousmalis K. Learning coordinated terrain-adaptive locomotion by imitating a centroidal dynamics planner. *CoRR* 2021;abs/2111.00262. Avaialble from: https://arxiv.org/abs/2111.00262 [Last accessed on 30 Aug 2022].
28. Gangapurwala S, Mitchell AL, Havoutis I. Guided constrained policy optimization for dynamic quadrupedal robot locomotion. *IEEE Robotics Autom Lett* 2020;5:3642–49. DOI
29. Chen X, Wang C, Zhou Z, Ross KW. Randomized ensembled double Q-learning: learning fast without a model. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net; 2021. Avaialble from: https://openreview.net/forum?id=AY8zfZm0tDd [Last accessed on 30 Aug 2022].
30. Smith L, Kew JC, Peng XB, et al. Legged robots that keep on learning: fine-tuning locomotion policies in the real world. In: 2022 IEEE International Conference on Robotics and Automation (ICRA); 2022. pp. 1–7. DOI
31. Coumans E, Bai Y. PyBullet, a Python module for physics simulation for games, robotics and machine learning; 2016–2021. http://pybullet.org.
32. Hwangbo J, Lee J, Hutter M. Per-Contact Iteration Method for Solving Contact Dynamics. *IEEE Robotics Autom Lett* 2018;3:895–902. Avaialble from: https://doi.org/10.1109/LRA.2018.2792536 [Last accessed on 30 Aug 2022].
33. Todorov E, Erez T, Tassa Y. MuJoCo: a physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2012. pp. 5026–33. DOI
34. Makoviychuk V, Wawrzyniak L, Guo Y, et al. Isaac gym: high performance GPU based physics simulation for robot learning. In: Vanschoren J, Yeung S, editors. Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual; 2021. Avaialble from: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/28dd2c7955ce926456240b2ff0100bde-Abstract-round2.html [Last accessed on 30 Aug 2022].
35. Rudin N, Hoeller D, Reist P, Hutter M. Learning to walk in minutes using massively parallel deep reinforcement learning. In: Conference on Robot Learning. PMLR; 2022. pp. 91–100. Avaialble from: https://proceedings.mlr.press/v164/rudin22a.html [Last accessed on 30 Aug 2022].
36. Margolis GB, Yang G, Paigwar K, Chen T, Agrawal P. Rapid locomotion via reinforcement learning. *arXiv preprint arXiv:220502824* 2022. DOI
37. Escontrela A, Peng XB, Yu W, et al. Adversarial motion priors make good substitutes for complex reward functions. *arXiv e-prints* 2022:arXiv:2203.15103. DOI
38. Vollenweider E, Bjelonic M, Klemm V, et al. Advanced skills through multiple adversarial motion priors in reinforcement learning. *arXiv e-prints* 2022:arXiv:2203.14912. DOI
39. Tan J, Zhang T, Coumans E, et al. Sim-to-real: learning agile locomotion for quadruped robots. In: Kress-Gazit H, Srinivasa SS, Howard T, Atanasov N, editors. Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018; 2018. Avaialble from: http://www.roboticsproceedings.org/rss14/p10.html [Last accessed on 30 Aug 2022].
40. Hutter M, Gehring C, Jud D, et al. Anymal-a highly mobile and dynamic quadrupedal robot. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE; 2016. pp. 38–44. DOI
41. Ha S, Xu P, Tan Z, Levine S, Tan J. *Learning to walk in the real world with minimal human effort* 2020;155:1110–20. Available from: https://proceedings.mlr.press/v155/ha21c.html [Lasta accessed on 30 Aug 2022].
42. Gangapurwala S, Geisert M, Orsolino R, Fallon M, Havoutis I. Rloc: terrain-aware legged locomotion using reinforcement learning and optimal control. *IEEE Trans Robot* 2022. DOI
43. Peng XB, van de Panne M. Learning locomotion skills using DeepRL: does the choice of action space matter? In: Teran J, Zheng C, Spencer SN, Thomaszewski B, Yin K, editors. Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation,

Los Angeles, CA, USA, July 28-30, 2017. Eurographics Association / ACM; 2017. pp. 12:1–2:13. Avaialble from: https://doi.org/10.1145/3099564.3099567 [Last accessed on 30 Aug 2022].

44. Chen S, Zhang B, Mueller MW, Rai A, Sreenath K. Learning torque control for quadrupedal locomotion. *CoRR* 2022;abs/2203.05194. Avaialble from: https://doi.org/10.48550/arXiv.2203.05194 [Last accessed on 30 Aug 2022].

45. Carlo JD, Wensing PM, Katz B, Bledt G, Kim S. Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018. IEEE; 2018. pp. 1–9. Avaialble from: https://doi.org/10.1109/IROS.2018.8594448 [Last accessed on 30 Aug 2022].

46. Jain D, Iscen A, Caluwaerts K. Hierarchical reinforcement learning for quadruped locomotion. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019. IEEE; 2019. pp. 7551–57. Avaialble from: https://doi.org/10.1109/IROS40897.2019.8967913 [Last accessed on 30 Aug 2022].

47. Li T, Lambert NO, Calandra R, Meier F, Rai A. Learning generalizable locomotion skills with hierarchical reinforcement learning. In: 2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020. IEEE; 2020. pp. 413–19. Avaialble from: https://doi.org/10.1109/ICRA40945.2020.9196642 [Last accessed on 30 Aug 2022].

48. Lee J, Hwangbo J, Hutter M. Robust recovery controller for a quadrupedal robot using deep reinforcement learning. *CoRR* 2019;abs/1901.07517. Available from: http://arxiv.org/abs/1901.07517 [Lasta accessed on 30 Aug 2022].

49. Iscen A, Caluwaerts K, Tan J, et al. Policies modulating trajectory generators. In: 2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings. vol. 87 of Proceedings of Machine Learning Research. PMLR; 2018. pp. 916–26. Available from: http://proceedings.mlr.press/v87/iscen18a.html [last accessed on 30 Aug 2022].

50. Rahme M, Abraham I, Elwin ML, Murphey TD. Dynamics and domain randomized gait modulation with Bezier curves for sim-to-real legged locomotion. *CoRR* 2020;abs/2010.12070. Avaialble from: https://arxiv.org/abs/2010.12070 [Last accessed on 30 Aug 2022].

51. Zhang H, Wang J, Wu Z, Wang Y, Wang D. Terrain-aware risk-assessment-network-aided deep reinforcement learning for quadrupedal locomotion in tough terrain. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021. IEEE; 2021. pp. 4538–45. Avaialble from: https://doi.org/10.1109/IROS51168.2021.9636519 [Last accessed on 30 Aug 2022].

52. Yang Y, Zhang T, Coumans E, Tan J, Boots B. Fast and efficient locomotion via learned gait transitions. In: Faust A, Hsu D, Neumann G, editors. Conference on Robot Learning, 8-11 November 2021, London, UK. vol. 164 of Proceedings of Machine Learning Research. PMLR; 2021. pp. 773–83. Avaialble from: https://proceedings.mlr.press/v164/yang22d.html [Last accessed on 30 Aug 2022].

53. Gangapurwala S, Geisert M, Orsolino R, Fallon MF, Havoutis I. Real-time trajectory adaptation for quadrupedal locomotion using deep reinforcement learning. In: IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021. IEEE; 2021. pp. 5973–79. Avaialble from: https://doi.org/10.1109/ICRA48506.2021.9561639 [Last accessed on 30 Aug 2022].

54. Yao Q, Wang J, Wang D, et al. Hierarchical terrain-aware control for quadrupedal locomotion by combining deep reinforcement learning and optimal control. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021. IEEE; 2021. pp. 4546–51. Avaialble from: https://doi.org/10.1109/IROS51168.2021.9636738 [Last accessed on 30 Aug 2022].

55. Singla A, Bhattacharya S, Dholakiya D, et al. Realizing learned quadruped locomotion behaviors through kinematic motion primitives. In: International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019. IEEE; 2019. pp. 7434–40. Avaialble from: https://doi.org/10.1109/ICRA.2019.8794179 [Last accessed on 30 Aug 2022].

56. Vollenweider E, Bjelonic M, Klemm V, et al. Advanced skills through multiple adversarial motion priors in reinforcement learning. *arXiv preprint arXiv:220314912* 2022. DOI

57. Li A, Wang Z, Wu J, Zhu Q. Efficient learning of control policies for robust quadruped bounding using pretrained neural networks. *arXiv preprint arXiv:201100446* 2020. DOI

58. Shao Y, Jin Y, Liu X, et al. Learning free gait transition for quadruped robots via phase-guided controller. *IEEE Robotics and Automation Letters* 2021;7:1230–37. DOI

59. Luo J, Hauser KK. Robust trajectory optimization under frictional contact with iterative learning. *Auton Robots* 2017;41:1447–61. Avaialble from: https://doi.org/10.1007/s10514-017-9629-x [Last accessed on 30 Aug 2022].

60. Kumar A, Fu Z, Pathak D, Malik J. RMA: rapid motor adaptation for legged robots. In: Proceedings of Robotics: Science and Systems. Virtual; 2021. DOI

61. Liu J, Zhang H, Wang D. DARA: dynamics-aware reward augmentation in offline reinforcement learning. *CoRR* 2022;abs/2203.06662. Avaialble from: https://doi.org/10.48550/arXiv.2203.06662 [Last accessed on 30 Aug 2022].

62. Shi H, Zhou B, Zeng H, et al. Reinforcement learning with evolutionary trajectory generator: A general approach for quadrupedal locomotion. *IEEE Robotics Autom Lett* 2022;7:3085–92. Avaialble from: https://doi.org/10.1109/LRA.2022.3145495 [Last accessed on 30 Aug 2022].

63. Zhao W, Queralta JP, Westerlund T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In: 2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, Canberra, Australia, December 1-4, 2020. IEEE; 2020. pp. 737–44. Avaialble from: https://doi.org/10.1109/SSCI47803.2020.9308468 [Last accessed on 30 Aug 2022].

64. Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:13125602* 2013. DOI

65. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. *arXiv preprint arXiv:170706347* 2017. DOI

66. Wang C, Yang T, Hao J, et al. ED2: an environment dynamics decomposition framework for world model construction. *CoRR* 2021;abs/2112.02817. Avaialble from: https://arxiv.org/abs/2112.02817 [Last accessed on 30 Aug 2022].

67. Kostrikov I, Yarats D, Fergus R. Image augmentation is all you need: regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:200413649* 2020. DOI
68. Yarats D, Fergus R, Lazaric A, Pinto L. Mastering visual continuous control: improved data-augmented reinforcement learning. *arXiv preprint arXiv:210709645* 2021. DOI
69. Ahmed O, Träuble F, Goyal A, et al. Causalworld: a robotic manipulation benchmark for causal structure and transfer learning. *arXiv preprint arXiv:201004296* 2020. DOI
70. Dittadi A, Träuble F, Wüthrich M, et al. The role of pretrained representations for the OOD generalization of RL agents. *arXiv preprint arXiv:210705686* 2021. DOI
71. Hsu K, Kim MJ, Rafailov R, Wu J, Finn C. Vision-based manipulators need to also see from their hands. *arXiv preprint arXiv:220312677* 2022. DOI
72. Eysenbach B, Asawa S, Chaudhari S, Levine S, Salakhutdinov R. Off-dynamics reinforcement learning: training for transfer with domain classifiers. *arXiv preprint arXiv:200613916* 2020. DOI
73. Liu J, Zhang H, Wang D. DARA: dynamics-aware reward augmentation in offline reinforcement learning. *arXiv preprint arXiv:220306662* 2022. DOI
74. Lee K, Seo Y, Lee S, Lee H, Shin J. Context-aware dynamics model for generalization in model-based reinforcement learning. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. vol. 119 of Proceedings of Machine Learning Research. PMLR; 2020. pp. 5757–66. Avaialble from: http://proceedings.mlr.press/v119/lee20g.html [Last accessed on 30 Aug 2022].
75. Yu W, Tan J, Liu CK, Turk G. Preparing for the unknown: learning a universal policy with online system identification. *arXiv preprint arXiv:170202453* 2017. DOI
76. Chen D, Zhou B, Koltun V, Krähenbühl P. Learning by cheating. In: Conference on Robot Learning. PMLR; 2020. pp. 66–75. Available from: http://proceedings.mlr.press/v100/chen20a.html [Last accessed on 30 Aug 2022].
77. Tobin J, Fong R, Ray A, et al. *Domain randomization for transferring deep neural networks from simulation to the real world* 2017. DOI
78. Peng XB, Andrychowicz M, Zaremba W, Abbeel P. *Sim-to-real transfer of robotic control with dynamics randomization* 2017. DOI
79. Kaiser L, Babaeizadeh M, Milos P, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:190300374* 2019. DOI
80. Schrittwieser J, Antonoglou I, Hubert T, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* 2020;588:604–9. DOI
81. Ye W, Liu S, Kurutach T, Abbeel P, Gao Y. Mastering atari games with limited data. *Adv Neural Inform Proc Syst* 2021;34:25476–88. Available from: https://proceedings.neurips.cc/paper/2021/hash/d5eca8dc3820cad9fe56a3bafda65ca1-Abstract.html [Last accessed on 30 Aug 2022].
82. Chua K, Calandra R, McAllister R, Levine S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Adva neural inform proc syst* 2018;31. Available from: https://proceedings.neurips.cc/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html [Last accessed on 30 Aug 2022].
83. Janner M, Fu J, Zhang M, Levine S. When to trust your model: model-based policy optimization. *Adv Neural Inform Proc Syst* 2019;32. Available from: https://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html [Last accessed on 30 Aug 2022].
84. Hansen N, Wang X, Su H. Temporal difference learning for model predictive control. *arXiv preprint arXiv:220304955* 2022. DOI
85. Tassa Y, Doron Y, Muldal A, et al. Deepmind control suite. *arXiv preprint arXiv:180100690* 2018. DOI
86. Peng XB, Ma Z, Abbeel P, Levine S, Kanazawa A. Amp: adversarial motion priors for stylized physics-based character control. *ACM Trans Graph (TOG)* 2021;40:1–20. DOI
87. Peng XB, Guo Y, Halper L, Levine S, Fidler S. ASE: large-scale reusable adversarial skill embeddings for physically simulated characters. *arXiv preprint arXiv:220501906* 2022. DOI
88. Merel J, Hasenclever L, Galashov A, et al. Neural probabilistic motor primitives for humanoid control. *arXiv preprint arXiv:181111711* 2018. DOI
89. Hasenclever L, Pardo F, Hadsell R, Heess N, Merel J. Comic: complementary task learning & mimicry for reusable skills. In: International Conference on Machine Learning. PMLR; 2020. pp. 4105–15. Available from: https://proceedings.mlr.press/v119/hasenclever20a.html [Last accessed on 30 Aug 2022].
90. Liu S, Lever G, Wang Z, et al. From motor control to team play in simulated humanoid football. *arXiv preprint arXiv:210512196* 2021. DOI
91. Escontrela A, Peng XB, Yu W, et al. Adversarial motion priors make good substitutes for complex reward functions. *arXiv preprint arXiv:220315103* 2022. DOI
92. Levine S, Kumar A, Tucker G, Fu J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:200501643* 2020. DOI
93. Duan Y, Schulman J, Chen X, et al. Rl ˆ2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:161102779* 2016. DOI
94. Nichol A, Achiam J, Schulman J. On first-order meta-learning algorithms. *arXiv preprint arXiv:180302999* 2018. DOI
95. Rakelly K, Zhou A, Finn C, Levine S, Quillen D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In: International Conference on Machine Learning. PMLR; 2019. pp. 5331–40. Available from: http://proceedings.mlr.press/v97/rakelly19a.html [Last accessed on 30 Aug 2022].
96. Seo Y, Lee K, James SL, Abbeel P. Reinforcement learning with action-free pre-training from videos. In: International Conference on

Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA. vol. 162 of Proceedings of Machine Learning Research. PMLR; 2022. pp. 19561–79. Avaialble from: https://proceedings.mlr.press/v162/seo22a.html [Last accessed on 30 Aug 2022].

97. Mandi Z, Abbeel P, James S. On the Effectiveness of Fine-tuning Versus Meta-reinforcement Learning. *arXiv preprint arXiv:220603271* 2022. DOI

98. Haarnoja T, Zhou A, Ha S, et al. *Learning to walk via deep reinforcement learning* 2019. DOI

99. Yang Y, Caluwaerts K, Iscen A, et al. Data efficient reinforcement learning for legged robots. In: Kaelbling LP, Kragic D, Sugiura K, editors. 3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings. vol. 100 of Proceedings of Machine Learning Research. PMLR; 2019. pp. 1–10. Avaialble from: http://proceedings.mlr.press/v100/yang20a.html [Last accessed on 30 Aug 2022].

100. Tsounis V, Alge M, Lee J, Farshidian F, Hutter M. DeepGait: planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robot Autom Lett* 2020;5:3699–706. DOI

101. Da X, Xie Z, Hoeller D, et al. Learning a contact-adaptive controller for robust, efficient legged locomotion. PMLR; 2020. Available from: https://proceedings.mlr.press/v155/da21a.html [Lasta accessed on 30 Aug 2022].

102. Liang J, Makoviychuk V, Handa A, et al. GPU-accelerated robotic simulation for distributed reinforcement learning. *CoRR* 2018;abs/1810.05762. Avaialble from: http://arxiv.org/abs/1810.05762 [Last accessed on 30 Aug 2022].

103. Escontrela A, Yu G, Xu P, Iscen A, Tan J. Zero-shot terrain generalization for visual locomotion policies. *CoRR* 2020;abs/2011.05513. Avaialble from: https://arxiv.org/abs/2011.05513 [Last accessed on 30 Aug 2022].

104. Jiang Y, Zhang T, Ho D, et al. *SimGAN: hybrid simulator identification for domain adaptation via adversarial reinforcement learning* 2021:2884–90. Available from: https://doi.org/10.1109/ICRA48506.2021.9561731 [last accessed on 30 Aug 2022].

105. Tan W, Fang X, Zhang W, et al. A hierarchical framework for quadruped locomotion based on reinforcement learning. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021. IEEE; 2021. pp. 8462–68. Avaialble from: https://doi.org/10.1109/IROS51168.2021.9636757 [Last accessed on 30 Aug 2022].

106. Michel O. WebotsTM: professional mobile robot simulation. *CoRR* 2004;abs/cs/0412052. Avaialble from: http://arxiv.org/abs/cs/0412052 [Last accessed on 30 Aug 2022].

107. Fu Z, Kumar A, Malik J, Pathak D. Minimizing energy consumption leads to the emergence of gaits in legged robots. *CoRR* 2021;abs/2111.01674. Avaialble from: https://arxiv.org/abs/2111.01674 [Last accessed on 30 Aug 2022].

108. Kim S, Sorokin M, Lee J, Ha S. Human motion control of quadrupedal robots using deep reinforcement learning. *arXiv preprint arXiv:220413336* 2022. Avaialble from: http://www.roboticsproceedings.org/rss18/p021.pdf [Last accessed on 30 Aug 2022].

109. Bogdanovic M, Khadiv M, Righetti L. Model-free reinforcement learning for robust locomotion using trajectory optimization for exploration. *arXiv preprint arXiv:210706629* 2021. DOI

110. Fernbach P, Tonneau S, Stasse O, Carpentier J, Taïx M. C-CROC: continuous and convex resolution of centroidal dynamic trajectories for legged robots in multicontact scenarios. *IEEE Trans Robot* 2020;36:676–91. DOI

111. Zhang H, Starke S, Komura T, Saito J. Mode-adaptive neural networks for quadruped motion control. *ACM Trans Graph (TOG)* 2018;37:1–11. DOI

112. Feldman A, Goussev V, Sangole A, Levin M. Threshold position control and the principle of minimal interaction in motor actions. *Progr brain res* 2007 02;165:267–81. DOI

113. Winkler AW, Bellicoso CD, Hutter M, Buchli J. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robot Autom Lett* 2018;3:1560–67. DOI

114. Liu H, Jia W, Bi L. Hopf oscillator based adaptive locomotion control for a bionic quadruped robot. *2017 IEEE Int Confer Mechatr Autom (ICMA)* 2017:949–54. DOI

115. Carlo JD, Wensing PM, Katz B, Bledt G, Kim S. Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2018. pp. 1–9. DOI

116. Bellicoso D, Jenelten F, Fankhauser P, et al. Dynamic locomotion and whole-body control for quadrupedal robots. *2017 IEEE/RSJ Int Conf Intell Robots Sys (IROS)* 2017:3359–65. DOI

117. Sethian JA. Fast marching methods. *SIAM Rev* 1999;41:199–235. DOI

118. Ponton B, Khadiv M, Meduri A, Righetti L. Efficient multi-contact pattern generation with sequential convex approximations of the centroidal dynamics. *CoRR* 2020;abs/2010.01215. Avaialble from: https://arxiv.org/abs/2010.01215 [Last accessed on 30 Aug 2022].

119. Zhang T, McCarthy Z, Jow O, et al. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. *CoRR* 2017;abs/1710.04615. Avaialble from: http://arxiv.org/abs/1710.04615 [Last accessed on 30 Aug 2022].

120. Thor M, Kulvicius T, Manoonpong P. Generic neural locomotion control framework for legged robots. *IEEE Trans Neural Netw Learn Syst* 2021;32:4013–25. DOI

## APPENDIX

**Table 1. Classification of the most relevant publications in DRL-based quadrupedal locomotion research**

| Publication | Pub. Year | Description | Algorithm | State Space | Action Space | Reward Function | Simulator | Robot |
|---|---|---|---|---|---|---|---|---|
| Sim-to-Real: Learning Agile Locomotion For Quadruped Robots [39] | RSS 2018 | A system to design agile locomotion by leveraging DRL algorithms. | PPO | Orientation (2-dim), Base Angular Velocities (2-dim), and Motor Angles (8-dim). | Desired Leg Pose. | Current and Previous Base Positions, Desired Running Direction, Motor Torques and Velocities. | Pybullet | Minitaur |
| Policies Modulating Trajectory Generators [49] | CoRL 2018 | An architecture for learning behaviors by using PMTG that provides memory and prior knowledge. | PPO | Orientation, Base Angular Velocities, Desired Velocity (control input), and Phase of the TG. | Leg Swing Angles and Extensions, Frequency, Amplitude, Walking Height. | Speed Gap (Desired vs. Acutual). | Pybullet | Minitaur |
| Robust Recovery Controller for a Quadrupedal Robot using Deep Reinforcement Learning [48] | ArXiv 2019 | A model-free DRL approach to control recovery maneuvers using a hierarchical controller. | TRPO + GAE [22] | Self-Righting: Gravity, Base Angular Velocity, Joint Positions, Velocities and History. Standing Up: Base Linear Velocity, State Space (Self-Righting). Locomotion: Command, Base Height, State Space (Standing Up). | Desired Joint Positions. | Angular and Linear Velocity, Height, Orientation, Torque, Power, Joint State, Body Impulse and Slippage, Foot Slippage and Clearance, Self Collision, and Action Gap. | RaiSim | ANYmal |
| Learning to Walk via Deep Reinforcement Learning [98] | RSS 2019 | A sample-efficient Max. Entropy RL algorithm requiring minimal per-task tuning to learn neural network policies. | SAC | Motor Angles (8-dim), Orientation (2-dim), and Base Angular Velocities (2-dim). | Leg Swing Angles and Extensions. | Walking Distance, Joint Accelerations and Angles, and Base Roll Angle. | Pybullet | Minitaur |
| Data Efficient Reinforcement Learning for Legged Robots [99] | CoRL 2019 | A model-based RL framework for learning locomotion from only 4.5 minutes of data collected on a quadruped robot. | MPC + CEM | Base Linear Velocity, Orientation (3-dim), and Motor Positions. | Leg Swing Angles and Extensions, and Phase Scales. | Speed Gap, and Base Orientation. | Pybullet | Minitaur |
| Hierarchical Reinforcement Learning for Quadruped Locomotion [46] | IROS 2019 | A hierarchical framework to automatically decompose complex locomotion tasks. | ARS | High-Level: Base Position and Orientation. Low-Level: PMTG State, Orientation, Base Angular Velocities, and Latent Command. | High-Level: Command, Duration. Low-Level: Motor Position, PMTG Param. | Steering Angle, Moving Distance. | Pybullet | Minitaur |
| Realizing Learned Quadruped Locomotion Behaviors through Kinematic Motion Primitives (kMPs) [55] | ICRA 2019 | kMPs is effective to learn quadrupedal walking using DRL, and realize these behaviors in Stoch. | PPO | Joint Angles, Velocities and Torques, and Orientation. | Leg End-Point Positions. | Position Change, Energy Consumption, and Motor Torques and Velocities. | Pybullet | Stoch |
| DeepGait: Planning and Control of Quadrupedal Gaits using Deep Reinforcement Learning [100] | ICRA 2019 | A technique for training terrain-aware locomotion, which combines Model-Based Planning and RL. | PPO, TRPO, GAE | Planner: Terrain Elevation, Base State and Velocity, Joint Torques, Feet State, Phase Variables. Controller: Phase (Current&Next), Joint Angles and Velocities. | Planner: Candidate Phase. Controller: Desired Joint Positions. | Planner: Goal Distance and Orientation, Work, Stance Phases. Controller: Target and Slip, Joint Velocities and Torques, Base Velocities, Angular Deviation. | RaiSim | ANYmal |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Learning agile and dynamic motor skills for legged robots [6] | Science Robotics 2019 | A Sim2Real method leveraging fast, automated and cost-effective data generation schemes. | TRPO | Height, Base Velocities, Joint State (with History), Previous Action, Command, Gravity. | Desired Joint Positions. | Locomotion: Base Velocities, Cost, Torque, Joint Speed, Foot State, Direction, Fluency. Recovery: Torque, Joint Motion, HAA, HFE, KFE, Slip and Impulse, Internal Contact, Direction, Fluency. | RaiSim | ANYmal |
| Learning to Walk in the Real World with Minimal Human Effort [41] | CoRL 2020 | A system for learning quadrupedal locomotion policies with Deep RL in the real world with minimal human effort. | SAC | Motor Angles (6-step), IMU Readings (6-step), and Previous Action (6-step). | Desired Joint Positions. | Base Position and Yaw, and Smoothness. | Pybullet | Minitaur |
| Dynamics and Domain Randomized Gait Modulation with Bezier Curves for Sim-to-Real Legged Locomotion [50] | ArXiv 2020 | A quadrupedal Sim2Real framework utilizing offline RL with dynamics and domain randomized to allow traversing uneven terrain. | ARS [23] | Orientation (2-dim), Base Angular Velocities (3-dim) and Linear Accelerations (3-dim), and Foot Phase. | Foot Position Residuals. | Distance, Orientation (2-dim), Base Angular Velocities (3-dim). | Pybullet | Mini Spot |
| Guided Constrained Policy Optimization for Dynamic Quadrupedal Robot Locomotion [28] | IEEE Robotics Autom 2020 | A CPPO-based RL framework for tracking velocity commands under constraints. | GCPO | Base Height and Velocities, Orientation, Joint State, Policy Output, Desired Base Velocity. | Desired Joint Positions. | Linear and Angular Velocity, Torque, Foot Acceleration and Slip, Smoothness, and Orientation. | RaiSim, PyBullet and MuJoCo. | ANYmal |
| Learning a Contact-Adaptive Controller for Robust, Efficient Legged Locomotion [101] | CoRL 2020 | A hierarchical framework combining Model-Based Control and RL to synthesize robust quadrupedal controllers. | DQN | Pose (without linear positions and foot positions), Primitive (previously-used). | One-Hot Primitive Selection Vector (9-dim) | Torques, Base Linear Velocity, Desired Base Linear Velocity. | IsaacGym [102] | Unitree Laikago |
| Zero-Shot Terrain Generalization for Visual Locomotion Policies [103] | ArXiv 2020 | A learning approach for terrain locomotion using exteroceptive inputs without ground-truth height maps. | PPO | Distance to Env., Orientation, Base Velocity, Joint Angles, Target Distance and Direction, Previous Action, Trajectory Generator Param. | Gait Frequency, Swing Height, Stride Length, Residual Action. | Euclidean Distance (Base to Target) and Timestep Duration. | Pybullet | Unitree Laikago |
| Learning Generalizable Locomotion Skills with Hierarchical Reinforcement Learning [47] | ICRA 2020 | A sample-efficient and generalizable hierarchical framework for learning locomotion skills on real-world robots. | SAC | Phase (1-dim) | Desired Joint Positions. | Forward Distance and Orientation Deviation. | PyBullet | Daisy Hexapod |
| Learning Agile Robotic Locomotion Skills by Imitating Animals [10] | RSS 2020 | A system enabling legged robot to learn agile locomotion skills by imitating real-world animals. | PPO | Pose (Orientation (3-dim) and Joint Rotations) and Action Sequence. | Joint Torques for Desired Positions. | Motion Gap (Current vs. Reference in Joint Velocities and State, End-Effector Positions, Base Pose and Velocity). | PyBullet | Unitree-Laikago |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Learning Quadrupedal Locomotion over Challenging Terrain [7] | Science Robotics 2020 | A novel Sim2Real solution incorporating proprioception showing remarkable zero-shot generalization. | TRPO | Goal Direction, Gravity, Base Velocities and Frequency, Joint States and Velocities, FTG Phases and Frequencies, Joint History, Terrain Normal Vector, Foot Height, Contact Forces and Target History, Contact States, Friction, External Force. | Leg Frequencies and Foot Position Residuals. | Linear and Angular Velocity, Base Motion Reward and Collision, Foot Clearance, Target Smoothness, and Torque. | RaiSim | ANYmal-B, ANYmal-C |
| Multi-expert learning (MEL) of adaptive legged locomotion [9] | Science Robotics 2020 | A MEL architecture to generate adaptive skills from a group of expert skills. | SAC | Joint Position, Gravity, Base Velocities, Phase Vector, and Goal Position. | Expert: Desired Joint Positions. Gating: Variable Weights. | Base Pose, Height and Velocity, Regularisation (Torque, Velocity), Foot State, Body State, Reference Positions and Contacts, Goal Position. | PyBullet | Jueying[3] |
| Efficient Learning of Control Policies for Robust Quadruped Bounding using Pretrained Neural Networks [57] | ArXiv 2021 | A training method for learning bounding gaits, which combines pre-training and DRL. | PPO2 | Base Height, Gravity Direction, Base Angular Velocity and Linear Acceleration, and Joint Position and Angular Velocity. | Desired Joint Positions. | Base Velocity, Joint Torque and Velocity, Gait, Position uniformity, Torque uniformity, Smoothness, and Pitch Limitations. | RaiSim | Jueying-Mini robot |
| Learning Coordinated Terrain-Adaptive Locomotion by Imitating a Centroidal Dynamics Planner [27] | ArXiv 2021 | A terrain adaptive controller obtained by training policies to reproduce trajectories planned by a non-linear solver. | V-MPO [25], MO-VMPO [26], | Image, State (Base, End-Effector, Joint, CoM), Velocities (Base, Joint), Orientation, Previous Action, Command. | Desired Joint Positions. | Joint Positions, Base Position, End-Effector Positions, Base Linear and Angular Velocities, and Quaternion. | Mujoco | ANYmal |
| Learning Free Gait Transition for Quadruped Robots via Phase-Guided Controller [58] | IEEE Robotics and Automation Letters 2021 | A novel quadrupedal framework for training a control policy to locomote in various gaits. | PPO | Velocity Command, Sine and Cosine Values (4 phases), Joint Position and Velocity, Angular Velocities, Gravity. | Desired Joint Positions. | Joint Torque, Desired Velocity, Base Balance, and Foot Contact. | RaiSim | Black Panther robot |
| Fast and Efficient Locomotion via Learned Gait Transitions [52] | CoRL 2021 | A hierarchical learning framework in which gait transitions emerge automatically with a reward of min. energy. | ARS [23] | Desired and Actual Base Linear Velocity | Desired Leg Frequency, Cutoff (Swing and Stance Phase), Phase Offset. | Torques, Base Velocities, and Survival. | Pybullet | Unitree A1 |
| SimGAN: Hybrid Simulator Identification for Domain Adaptation via Adversarial RL [104] | ICRA 2021 | A framework for domain adaptation by identifying a simulator to match the simulated trajectories to the target ones. | PPO | Orientation, Base Height, Base Linear Velocities and Joint Angles (12-dim). | Desired Joint Torques. | Base Forward Velocity, Joint Limit Count, and Torque. | Pybullet | Unitree Laikago |
| Hierarchical Terrain-Aware Control (HTC) for Quadrupedal Locomotion by Combining DRL and Optimal Control [54] | IROS 2021 | A novel HTC framework leveraging DRL for the high-level and optimal control for the low-level. | SAC | Global Height Map, Motors Positions, Orientation, and Gait Phase. | Goal Swing, Base Height and Velocity, Orientation. | Desired Forward Velocity to Target and Orientation. | Pybullet | Unitree A1 |

[3] https://www.deeprobotics.cn/

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A Hierarchical Framework for Quadruped Locomotion Based on RL [105] | IROS 2021 | A well-performing quadruped robot system for learning locomotion in real-world terrains without pre-training. | SAC | Angle Error, Orientation, and Command. | Goal Position, Base Velocity, Pitch, and Leg Lift Max. Height. | Base Target Positions (with Previous), Orientation, and Survival. | Webots [106] | Yobogo |
| Terrain-Aware Risk-Assessment-Network-Aided (RAN) DRL for Quadrupedal Locomotion in Tough Terrain [51] | IROS 2021 | A terrain-aware DRL-based controller integrating a RAN to guarantee the action stability. | SAC | Elevation Map, Goal Direction, Base Velocities, Joint State and Velocity, FTG Phases, Frequencies (Base and FTG), Joint History, Foot Targets and Contact Forces, Contact States, Env. Param. | Desired Joint Positions. | Linear and Angular Velocity, Base Motion and Collision, Foot Clearance, Target Smoothness, Torque, and Traversability Map. | Pybullet | Unitree A1 |
| Real-Time Trajectory Adaptation for Quadrupedal Locomotion using DRL [53] | ICRA 2021 | A policy using DRL to get noisy reference trajectory in order to generate a quadrupedal tracking system. | PPO | Robot Positions and Velocities, Reference Positions and Velocities, Corrected Trajectory Positions and Velocities, Height Maps. | Base State and Velocity Deviation, End-Effector State and Velocities Deviation. | Torque, Foot State, Smoothness, Orientation, Joint Motion, Trajectory Tracking, Goal. | RaiSim | ANYmal |
| Coupling Vision and Proprioception for navigation of Legged Robots [11] | CVPR workshop 2021 | Incorporating vision and proprioception in navigation tasks of legged robots. | PPO | Proprioception, Command Velocities, Previous Action and Extrinsic Vector. | Desired Joint Positions. | Velocity Gap, Energy Consumption, Lateral Movement, and Hip Joints. | RaiSim | Unitree A1 |
| Minimising Energy Consumption Leads to the Emergence of Gaits in Legged Robots [107] | CoRL 2021 | Energy constraints leading to the emergence of natural locomotion, and the choice is related to the desired speed. | PPO | Joint Positions and Velocities, Orientation, Foot Contact, Previous Action. | Desired Joint Positions. | Linear and Angular Velocity, and Joint Torques and Velocities. | RaiSim | Unitree A1 |
| RMA: Rapid Motor Adaptation for Legged Robots [60] | RSS 2021 | RMA algorithm for real-time online adaptation problems in quadruped robots. | PPO | Joint Positions and Velocities, Orientation (2-dim), and Foot Contact Vector. | Desired Joint Positions. | Base Motion and Orientation, Work, Ground Impact, Smoothness, Joint Speed, Foot Slip. | RaiSim | Unitree A1 |
| Human Motion Control of Quadrupedal Robots using DRL [108] | RSS 2022 | A quadrupedal motion control system allowing human operation. | PPO | Sensor Data (with history), Actions History, and Reference Poses (with history). | Desired Joint Positions [43] | Joint Imitation, End-Effector, Base Sate, Deviation, and Acceleration. | RaiSim | Unitree A1 |
| Learning Torque Control for Quadrupedal Locomotion [44] | ArXiv 2022 | A quadrupedal torque control framework predicting high-frequency joint torques via RL. | PPO | Base Velocities, Gravity, Joint Position and Velocity, Command, Last Action. | Desired Torques. | Base Velocities and Height, Orientation, Joint Motion, Foot State, Knee Collision, Action Rate, Gaits, and Hips Reward. | Isaac, Pybullt | Unitree A1 |
| Model-free RL for Robust Locomotion using Demonstrations from Trajectory Optimization [109] | ArXiv 2022 | A RL approach to create robust policies deployable on real robots without additional training using a single optimised demonstration. | PPO | Joint Positions and Velocities, Orientation (Yall and Roll) and Angular Velocities (Yall and Roll). | Desired Joint Positions. | Joint Position, and Base Position, Quaternion and Angular Velocity. | Pybullet | Solos |
| Legged Robots that Keep on Learning: Fine-Tuning Locomotion Policies in the Real World [30] | ICRA 2022 | A robot RL system for fine-tuning real-world locomotion policies. | REDQ [29] | Orientation (3-step), Joint Angles (3-step), Actions (3-step), Future Target Poses. | Desired Joint Positions. | Joint States and Velocities, End-Effector State, Base Pose and Velocity. | Pybullet | Unitree A1 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| RLOC: Terrain-Aware Legged Locomotion using RL and Optimal Control [42] | IEEE Transactions on Robotics 2022 | A unified model-based and data-driven approach for quadrupedal locomotion over uneven terrain. | SAC, TD3, GCPO [28] | Planning: Base State, Joint States, Goal Velocity, Elevation Map. Adaption: Base State, Feet Goal, Torques, Elevation Map. Recovery: Joint Position, Goal Positions and Velocity. | Planning: Coordinates. Adaption: Joint Torques. Recovery: Desired Joint Positions. | Planning: Base Velocities, Torque, Foot Slip, Stability. Adaption: State Deviation, Robot State. Recovery: State Space (Planning), Foot Motion (Foot, Joint), Smoothness. | RaiSim | ANYmal B, ANYmal C |
| Rapid Locomotion via Reinforcement Learning [36] | RSS 2022 | A MIT Mini Cheetah controller achieving record agility. | PPO | Joint Angles and Velocities, Gravity, Previous Actions, Goal Velocity. | Desired Joint Positions. | Velocity Tracking, Base Pose, Self-Collision, Joint Limits, Torques, Action Rate, Airtime. | IsaacGym | MIT Mini Cheetah |
| Learning to Walk in Minutes Using Massively Parallel DRL [35] | CoRL 2022 | A robotic training framework achieving fast policy generation via parallelism. | PPO | Base Velocities, Gravity, Joint Motion, Previous Actions, Terrain Measurements. | Desired Joint Positions. | Velocity Tracking, Joint Motion, Torques, Action Rate, Collisions, Feet Airtime. | IsaacGym | ANYmal B,<br>ANYmal C,<br>Unitree A1 |
| Adversarial Motion Priors Make Good Substitutes for Complex Reward Functions [91] | ArXiv 2022 | Substituting reward functions with stylish rewards learned from motion captures. | PPO | Joint Angles and Velocities, Orientations and Previous Actions. | Joint Torques for Desired Positions. | Linear and Angular Velocity Tracking, and Motion Prior Discrimination | Issac Gym | Unitree A1 |
| Advanced Skills through Multiple Adversarial Motion Priors in RL [56] | ArXiv 2022 | An adversarial motion prior-based RL approach to allow for multiple, discretely switchable styles. | PPO | Base State and Velocities, Gravity, Joint Positions and Velocity, and Wheel Positions. | Desired Joint Positions. | Linear and Angular Velocity Tracking, Pose, and Joint Velocity and Position. | Issac Gym | Quadruped Humanoid Transformer |
| Learning robust perceptive locomotion for quadrupedal robots in the wild [8] | Science Robotics 2022 | A quadrupedal locomotion solution integrating exteroceptive and proprioceptive perception. | PPO | Command, Base Pose and Motion, Joint History, CPG Phase, Height Samples, Contact States, Friction, External Forces, Airtime. | Phase Offset, Joint Position Target. | Velocities, Body Motion, Foot Clearance, Collisions, Joint Motion and Constraint, Smoothness, Torque, Slip. | RaiSim | ANYmal-C |
| Imitate and Repurpose: Learning Reusable Robot Movement Skills From Human and Animal Behaviors [12] | ArXiv 2022 | Learning reusable locomotion skills for real legged robots using prior knowledge of human and animal movement. | V-MPO, MO-VMPO, | Base States and Motion, Latent Command, Joint States and Velocities, Gravity, Goal Velocity and Position, Ball Position, End-Effector Position, Clip ID. | High-Level: Latent Command. Low-Level: Desired Joint Positions. | Imitation: Tracking CoM, Joint Velocities, End-Effector Positions, Body Quaternions, Current Draw. Walking: Tracking Velocity. Ball Dribbling: Tracking Ball Positions. | MuJoCo | ANYmal |
| Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers [13] | ICLR 2022 | An end-to-end RL method leveraging both proprioceptive states and visual observations for locomotion control. | PPO | Orientation, Joint Rotations, Previous Actions (3-step), and Image (4 dense depth). | Desired Joint Positions. | Distance, Motor Torques, Survival, and Collected Sphere Count. | Pybullet | Unitree A1 |
| RL with Evolutionary Trajectory Generator: A General Approach for Quadrupedal Locomotion [62] | IEEE Robotics Autom 2022 | A novel RL-based approach containing an evolutionary foot trajectory generator. | SAC | Orientation (3-dim), Joint Angles and Angular Velocities, Feet Contact Vector (4-dim), and Base Velocity. | Desired Joint Positions. | Base Position, Desired Direction, and Consumed Energy. | Pybullet | Unitree A1 |

**Table 2. More information about publications (Supplement to Table 1)**

| Publication | Others | Solution to Reality Gap | Open-Source Package |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Sim-to-Real: Learning Agile Locomotion For Quadruped Robots [39] | Open-loop Controller. | Improving Simulation Fidelity (Actuator Model, Latency), and Dynamics Randomization (Mass, Motor Strength, Inertia, Control Step, Latency, Battery, Friction, IMU bias and noise. | https://github.com/bulletphysics/bullet3/tree/master/examples/pybullet/gym/pybullet_envs/minitaur/envs |
| Policies Modulating Trajectory Generators [49] | Trajectory Generator. | Random Directional Virtual Forces. | / |
| Robust Recovery Controller for a Quadrupedal Robot using Deep Reinforcement Learning [48] | / | Randomized Physical Properties, Actuator Model, Additive Noise to the Observation. | / |
| Data Efficient Reinforcement Learning for Legged Robots [99] | Trajectory Generators. | / | / |
| Hierarchical Reinforcement Learning for Quadruped Locomotion [46] | PMTG [49] | / | / |
| Realizing Learned Quadruped Locomotion Behaviors through Kinematic Motion Primitives [55] | Recorded Data (Joint Angles and Orientation for 4800 steps). | / | / |
| DeepGait: Planning and Control of Quadrupedal Gaits using Deep Reinforcement Learning [100] | CROC [110] | / | / |
| Learning agile and dynamic motor skills for legged robots [6] | A controller generating foot trajectories to train the actuator model. | Actuator Model, Curriculum Training, Randomised Bodies (Size and Position), Random Command and Initial State. | / |
| Dynamics and Domain Randomized Gait Modulation with Bezier Curves for Sim-to-Real Legged Locomotion [50] | Open-loop Bezier curve Gait Generator. | Domain Randomization (Base Mass , Leg Link Masses, Foot Friction, XYZ Mesh Magnitude). | https://github.com/OpenQuadruped/spot_mini_mini |
| Guided Constrained Policy Optimization for Dynamic Quadrupedal Robot Locomotion [28] | / | Noisy Observations and Actions, and Domain Randomisation (Gravity, Actuator Torque Scaling, Link Mass and Size, Actuator Damping, and Step Time). | / |
| Learning a Contact-Adaptive Controller for Robust, Efficient Legged Locomotion [101] | A simple model-based method [45]. | / | / |
| Zero-Shot Terrain Generalization for Visual Locomotion Policies [103] | PMTG | / | / |
| Learning Generalizable Locomotion Skills with Hierarchical Reinforcement Learning [47] | MPC (High-Level Planning), Sinusoidal Policy (TG, Low-Level Controller). | / | / |

| | | | |
|---|---|---|---|
| Learning Agile Robotic Locomotion Skills by Imitating Animals [10] | MoCap Clips [111]. | Domain Randomization and Domain Adapation. | https://github.com/erwincoumans/motion_imitation |
| Learning Quadrupedal Locomotion over Challenging Terrain [7] | Foot Trajectory Generator. | Actuator Model, Randomized Physical Parameters, Teach-Student Training Set-up, and Automated Curriculum synthesizing Terrains. | / |
| Multi-expert learning of adaptive legged locomotion [9] | Reference Gait to provide Joint Position Reward and Foot Contact Reward. | Action Filtering and Smoothing Loss [112]. | / |
| Efficient Learning of Control Policies for Robust Quadruped Bounding using Pretrained Neural Networks [57] | SLIP: Spring Linear Inverted Pendulum (Model-based Controller). | Domain randomization (Link Mass, Inertia and CoM, Initial Direction and Ground Friction and Restitution). | / |
| Learning Coordinated Terrain-Adaptive Locomotion by Imitating a Centroidal Dynamics Planner [27] | TOWR [113] | / | / |
| Learning Free Gait Transition for Quadruped Robots via Phase-Guided Controller [58] | Hopf Oscillator and manually designed functions [114]. | Domain Randomization (External Force and Torque, Ground Friction and Height, Mass, Body Size, Noise of Joint Position and Velocity, Body Posture, and Angular Velocity). | https://github.com/ZJU-XMech/PhaseGuidedControl |
| Fast and Efficient Locomotion via Learned Gait Transitions [52] | Centroidal Dynamics Model [115]. | / | / |
| SimGAN: Hybrid Simulator Identification for Domain Adaptation via Adversarial Reinforcement Learning [104] | / | Hybrid Simulator Identification. | / |
| Hierarchical Terrain-Aware Control for Quadrupedal Locomotion by Combining Deep Reinforcement Learning and Optimal Control [54] | Optimal Control. | Domain Randomization (Mass, Inertia, Motor Strength and Friction, Latency, Lateral Friction, Battery, Joint Friction, CoM position noise, External force, and Step Height and width. | / |
| A Hierarchical Framework for Quadruped Locomotion Based on Reinforcement Learning [105] | Trajectory Generator. | Domain Randomization (Leg Profile and Mass, Base Mass Distribution, Leg Inertia Matrix, and Communication Delay). | / |
| Terrain-Aware Risk-Assessment-Network-Aided Deep Reinforcement Learning for Quadrupedal Locomotion in Tough Terrain [51] | PMTG [49]. | Domain Randomization (Mass, Inertia, Motor Strength and Friction, Latency, Lateral friction, Battery, Joint friction). | / |

| | | | |
|---|---|---|---|
| Real-Time Trajectory Adaptation for Quadrupedal Locomotion using Deep Reinforcement Learning [53] | TOWR [113], WBC [116]. | Domain Randomization, Actuator Modelling, Shifting Initial Position, Changing Gravity, Actuator Torque Scaling, and Perturbing the Robot Base. | / |
| Coupling Vision and Proprioception for navigation of Legged Robots [11] | FMM [117], librealsense[4]. | RMA-based Adaption Module [60]. | / |
| Minimizing Energy Consumption Leads to the Emergence of Gaits in Legged Robots [107] | / | RMA-based Adaption Module. | / |
| RMA: Rapid Motor Adaptation for Legged Robots [60] | / | RMA-based Adaption Module. | / |
| Human Motion Control of Quadrupedal Robots using Deep Reinforcement Learning [108] | Human Motions. | Domain randomization (Link Mass, Ground Friction Coefficients and Slope, Proportional and Derivative Gain, and Communication Delay). | / |
| Learning Torque Control for Quadrupedal Locomotion [44] | / | Domain randomization (Base Linear and Angular Velocity, Projected Gravity, Joint Position and Velocity, Ground Friction and External Disturbances). | / |
| Model-free Reinforcement Learning for Robust Locomotion using Demonstrations from Trajectory Optimization [109] | Trajectory Optimization Algorithm [118]. | / | / |
| Legged Robots that Keep on Learning: Fine-Tuning Locomotion Policies in the Real World [30] | MoCap Dog Recording [111] and Side-Step motion for A1 [10]. | Real-World Fine-Tuning. | https://github.com/lauramsmith/fine-tuning-locomotion |
| RLOC: Terrain-Aware Legged Locomotion using Reinforcement Learning and Optimal Control [42] | Dynamic Gaits Controller. | Domain randomization (Gravity, Actuator Torque Scaling, Link Mass and Size, Actuator Damping), Perturbation on Robot Base, and Smoothing Filters for Elevation Map, Actuator Model. | / |
| Rapid Locomotion via Reinforcement Learning [36] | Curriculum Strategy. | Domain Randomization. | / |
| Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning [35] | / | Domain Randomization (Ground Friction and External Force), and Noisy Observations. | https://github.com/leggedrobotics/legged_gym |

[4] https://github.com/IntelRealSense/librealsense

| | | | |
|---|---|---|---|
| Adversarial Motion Priors Make Good Subsitutes for Complex Reward Functions [91] | German Shepherd Motion Dataset [119]. | Domain Randomization (Friction, Base Mass, Velocity Perturbation, Motor Gain Multiplier). | https://github.com/Alescontrela/AMP_for_hardware |
| Advanced Skills through Multiple Adversarial Motion Priors in Reinforcement Learning [56] | Motion Data from another RL policy or an MPC controller. | Actuator Model, Domain Randomisation (Rough Terrin, Disturbances, External Force), Curriculum Training, and Joint-Velocity-Based Trajectory Termination. | / |
| Learning robust perceptive locomotion for quadrupedal robots in the wild [8] | Foot Trajectory Generator. | Actuator Model, Domain Randomisation (Robot Mass, Initial Joint Position and Velocity, Orientation, External Force, Friction Coefficient, Curriculum Learning, and Randomized Height Sampling. | / |
| Imitate and Repurpose: Learning Reusable Robot Movement Skills From Human and Animal Behaviors [12] | MoCap Dataset of dog walking and turning behaviors [111]. | Domain Randomization (Body Mass, Centre of Mass, Joint Position and Reference, Joint Damping and Friction loss, Geom Friction, P Gain, and Torque Limit). | / |
| Learning Vision-Guided Quadrupedal Locomotion End-to-End with Cross-Modal Transformers [13] | / | Domain Randomization (KP, KD, Inertia, Lateral Friction, Mass, Motor Friction and Strength, and Sensor Latency), and Random Depth Input. | https://github.com/Mehooz/vision4leg |
| Reinforcement Learning with Evolutionary Trajectory Generator: A General Approach for Quadrupedal Locomotion [62] | Evolutionary Trajectory Generator [120]. | Teacher-Student Learning Setting, Domain Adaptation,and Domain Randomization (Control Latency, Foot Friction, Base Mass, Leg Mass, and Motor Kp, Kd), and Noisy Sensor Input. | https://github.com/PaddlePaddle/PaddleRobotics/tree/main/QuadrupedalRobots/ETGRL |