DOI：10.13973/j.cnki.robot.240239 CSTR：32165.14.robot.240239

封面文章

基于预训练－微调框架的四足机器人结构－控制协同设计

陈 词[1]，余纪宇[1]，李 超[2]，陆豪健[1]，高洪波[3]，熊 蓉[1]，王 越[1]

（1. 浙江大学工业控制技术国家重点实验室，浙江 杭州 310027； 2. 杭州云深处科技有限公司，浙江 杭州 310058；
3. 中国科学技术大学信息科学技术学院，安徽 合肥 230026）

摘 要：在自然界中，美洲狮等具有卓越运动能力的动物通常具有不对称的前腿和后腿，其强壮的后腿能够为跳跃提供强劲的动力。受这一类生物启发，提出了一种机械结构与控制策略协同设计的方法，通过优化机器人的腿部长度来提升其整体运动性能。首先，引入一种预训练－微调框架，该框架不仅能够为每个候选的机械结构提供最佳控制策略，还提升了算法的训练效率。此外，将空间域随机化与折扣正则化方法相结合，显著提高了预训练网络的泛化能力。实验结果表明，所提出的预训练－微调框架显著增强了协同设计算法的性能，并且减少了时间消耗。此外，所提算法在提高机器人运动性能方面远远超越了传统的对控制策略进行独立优化的方法，为提升四足机器人的极限跑酷能力提供了一种新型的解决方案。

关键词：四足机器人；结构－控制协同设计；深度强化学习

Structure-control Co-design of Quadruped Robots Based on Pre-training-Fine-tuning Framework

CHEN Ci[1]， YU Jiyu[1]， LI Chao[2]， LU Haojian[1]， GAO Hongbo[3]， XIONG Rong[1]， WANG Yue[1]

(1. *The State Key Laboratory of Industrial Control and Technology*, *Zhejiang University*, *Hangzhou* 310027, *China*;
2. *Deep Robotics Company*, *Hangzhou* 310058, *China*;
3. *School of Information Science and Technology*, *University of Science and Technology of China*, *Hefei* 230026, *China*)

Abstract: In nature, animals with exceptional locomotion abilities, such as cougars, often possess asymmetric fore and hind legs, with their powerful hind legs acting as reservoirs of energy for leaps. Inspired by this biological concept, a co-optimization method for mechanical structures and control strategies is proposed, focusing on optimizing the leg length of the robot to enhance its overall motion performance. Firstly, a novel pretraining-finetuning framework is introduced, which not only guarantees optimal control strategies for each mechanical candidate but also improves the training efficiency of the algorithm. Additionally, spatial domain randomization is integrated with discount regularization, which remarkably improves the generalization ability of the pretraining network. Experimental results indicate that the proposed pretraining-finetuning framework significantly enhances the overall co-design performance with less time consumption. Moreover, the co-design strategy substantially exceeds the conventional method of independently optimizing control strategies, providing an innovative approach to enhancing the extreme parkour capabilities of quadruped robots.

Keywords: quadrupedal robot; structure-control co-design; deep reinforcement learning

自然界中，四足动物通常具有卓越的运动能力，例如，一只身长约 1.8 m 的红袋鼠能够轻松向前跳跃超过 9 m。值得注意的是，四足哺乳动物的腿部结构通常是不对称的，这是自然选择的结果，有助于其实现灵活而高效的运动[1]。相对而言，当前常见的四足机器人在腿部长度的设计上多依赖于机械工程师的经验，缺乏明确的理论指导。因此，受上述生物学启发，本文提出了一种基于预训练－微调的框架，旨在通过优化机器人的腿部长度来提升其整体运动性能。

机器人的机械结构与控制策略的联合优化被称为协同设计。目前，已有众多研究开展了四足机器人的协同设计[2-6]，但主要局限于行走任务。例如，Belmonte-Baeza 等[2] 探讨了大腿与小腿之间的理想比例，旨在优化速度跟踪性能和能效等目标，以应对不同地形的挑战。Yu 等[5] 强调了基于 Transformer 架构的控制策略的泛化性，声称其能够适应多种比例的腿长，并展示了仅通过更新结构参

基金项目：国家自然科学基金（62373322，62173293，62303407，U2013601）.
通信作者：王越，wangyue@iipc.zju.edu.cn 收稿/录用/修回：2024-09-18/2024-11-19/2024-12-11

数、使用固定控制策略的机器人可以跨越 10 cm 的平台。Chen 等[3-4] 则通过协同设计方法提升了自由度受限的小型四足机器人在平地上的运动能力。然而，在简单的行走任务中，机器人协同设计的潜力尚未得到充分发挥。在更具挑战性的控制任务中，尤其是跑酷任务（如跳高和跳远）[7-10]，协同设计的重要性愈加突出。机器人的物理能力及其执行的任务范围受其机械结构的限制，当机械结构设计不当时，即便使用最巧妙的运动控制策略也可能难以实现预期任务。因此，通过机械结构与控制策略的相互增强，可以显著提高机器人运动性能的上限。

机器人的协同设计主要可分为基于最优控制的方法[11-13] 和基于深度强化学习（DRL）的方法[1-6,14-17]。然而，基于最优控制的方法需要丰富的专业知识和工程经验，以构建动力学模型并设计等式或者不等式约束。对于具有不同机械结构的机器人，运动控制还必须结合系统辨识方法[18]，这一过程通常极为繁琐。基于 DRL 的方法通常将协同设计建模为双层优化问题，其中下层为在特定结构参数下的优化控制策略，而上层则在给定最佳控制策略的情况下寻找最佳结构。根据获取方式的不同，控制策略主要可分为两类：第一类是在上层优化过程中为给定的候选结构训练特定的控制器，如图 1(a) 所示。例如，Gupta 等[1] 在上层使用锦标赛算法，将每种候选结构的适应度作为输入，以推导出最佳结构；在下层，利用 1152 个 CPU 收集训练数据，为各种候选结构学习最优的控制策略，这一过程耗时较长且需要大量的计算资源。相比之下，第二类集中于开发一种具有泛化性的控制策略[2-3,5-6]，该策略能够提供适用于各种结构的替代策略，如图 1(b) 所示。例如，Chen 等[3] 采用耦合网络架构，Yu 等[5] 利用 Transformer 架构，Bjelonic 等[6] 采用教师－学生架构训练替代策略。然而，这种用泛化策略替代最优策略的方法虽然节省了时间，却不能确保在特定结构下控制策略的最优性，进一步影响了上层优化决策的有效性。

为了在保持控制策略最优性的同时节省计算资源，本文提出了一种针对四足机器人协同设计的预训练－微调框架，旨在提升机器人跑酷性能，如图 1(c) 所示。首先训练了一种适用于多种结构的泛化性策略，具体而言，结合了空间域随机化和折扣正则化技术。前者使得数千种不同结构的机器人能够同时进行训练，而后者则有助于减少价值网络中的记忆偏差。这两者的结合显著增强了预训练模型的泛化能力。在随后的结构优化阶段，对这一泛化性模型进行微调，以适应特定的结构参数。实验结果表明，仅需 400 步的微调（约为常规训练步骤的 6.67%），新模型的各项评估指标趋于稳定，性能能够与针对每种结构专门设计的控制策略相媲美。

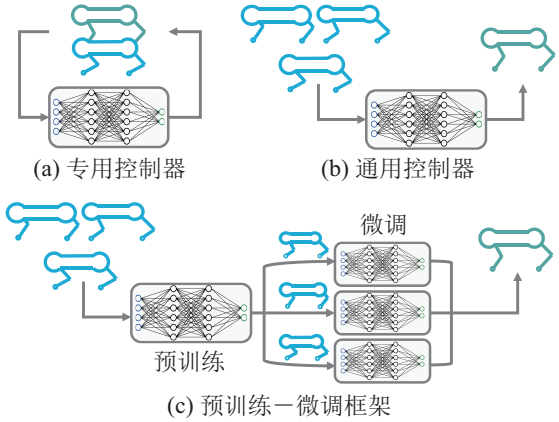


图 1 不同结构－控制协同设计方法的对比

Fig.1 Comparison of different structure-control co-design methods

1 预训练－微调框架（Pre-training-fine-tuning framework）

1.1 系统总览

算法整体框架如图 2 所示，分为预训练阶段和微调阶段 2 个阶段。在预训练阶段，开发了一种适用于不同腿长机器人的跑酷策略，作为预训练策略。在微调阶段，采用贝叶斯优化算法迭代地将候选结构参数的适应度作为先验知识，寻找最优结构参数。由于预训练策略并非是每组结构参数的最优控制策略，因此在特定结构参数下对其进行微调以提高准确性。这种微调使得在贝叶斯优化中进行更精确的适应度计算成为可能，从而提升了协同设计的整体性能。

本文将跑酷运动的控制问题建模为马尔可夫决策过程（MDP），由一个元组 $\langle S,A,R,P,\gamma\rangle$ 定义，其中 S 是状态空间，A 是动作空间，$R(s_t,a_t)$ 是奖励函数，$P(s_{t+1}|s_t,a_t)$ 是状态转移概率，γ 是折扣因子。在每一个时间步，智能体从环境中接收状态 $s_t\in S$，并根据策略 $\pi_\theta(a_t|s_t)$ 选择动作 $a_t\in A$，该动作产生奖励 r_t。DRL 的目标是获得一个最优策略 π^*，该策略能够最大化折扣累计奖励值的期望 $E\left(\sum_{t=0}^{T}\gamma^t r_t\right)$。采用近端策略优化算法（PPO）[19] 来解决机器人的运动控制问题。

1.1.1 状态与动作空间的设计

状态空间由 5 部分组成：本体感知 x_t，显式特权信息 e_t'，特权信息 e_t，外部感知 m_t，以及历史本

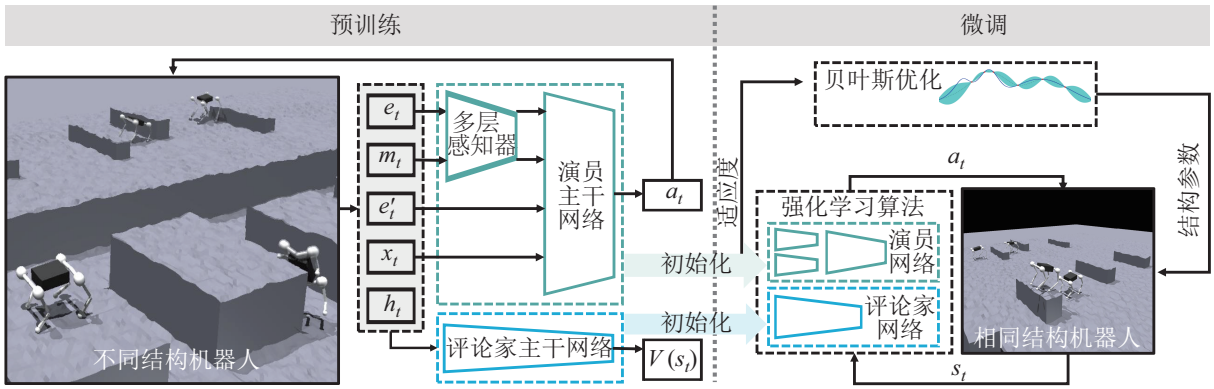


图 2　基于预训练－微调框架的结构－控制协同设计算法整体框架

Fig.2　The schematic of structure-control co-design algorithm based on pre-training-fine-tuning framework

体感知 h_t。其中，x_t 包含身体的角速度、横滚角、俯仰角、身体偏航角与目标偏航角的差值、关节的位置和速度、上一步中的动作以及足端触地信息，e_t' 指身体的线速度，e_t 包含机器人的质量、质心位置、机器人的结构参数、足端与地面之间的摩擦系数以及电动机阻尼比，m_t 指地形高度采样点。动作 a_t 为通过比例－微分（PD）控制器将机器人 12 个关节的目标角度转化为扭矩，然后传输给机器人的动作。

1.1.2　奖励函数的设计

四足机器人在常规的运动控制任务中是将速度跟踪作为主要的奖励函数项[20-23]。在跑酷任务中，手工指定机器人的线速度和偏航角会限制动作的解空间，导致任务失败。本文奖励函数的设计参考了 Cheng 等[9] 的工作，将主要奖励部分设置为“目标追踪”，鼓励机器人向航向点标记移动，当遇到高台时，该奖励项鼓励机器人跳上去而不是绕过去。奖励函数项的设置如表 1 所示。其中，p 表示后续航向点，$\boldsymbol{x}$ 表示机器人在世界坐标系下的位置，机器人在世界坐标系下的速度为 $\boldsymbol{v}\in\mathbb{R}^3$，目标速度由 $v^{\mathrm{cmd}}\in\mathbb{R}^1$ 表示；指示函数 c_i 在第 i 个足端与地面接触时设置为 1，布尔函数 M 在点 p_i 位于高台边缘 5 cm 范围内返回 1，每个足端的位置由点坐标 $\boldsymbol{p}_i$ 给出；符号 $\exp(\cdot)$ 表示指数运算符，上标 $(\cdot)^{\mathrm{des}}$ 用于指定期望值；$\boldsymbol{\omega}\in\mathbb{R}^3$ 表示身体的角速度；$\boldsymbol{a}_t\in\mathbb{R}^{12}$ 表示动作；$\boldsymbol{q}\in\mathbb{R}^{12}$、$\ddot{\boldsymbol{q}}\in\mathbb{R}^{12}$ 和 $\boldsymbol{\tau}\in\mathbb{R}^{12}$ 分别表示关节的角度、角加速度和扭矩；$\boldsymbol{g}$ 表示投影到机器人本体坐标系中的重力矢量。

表 1　奖励函数的设计

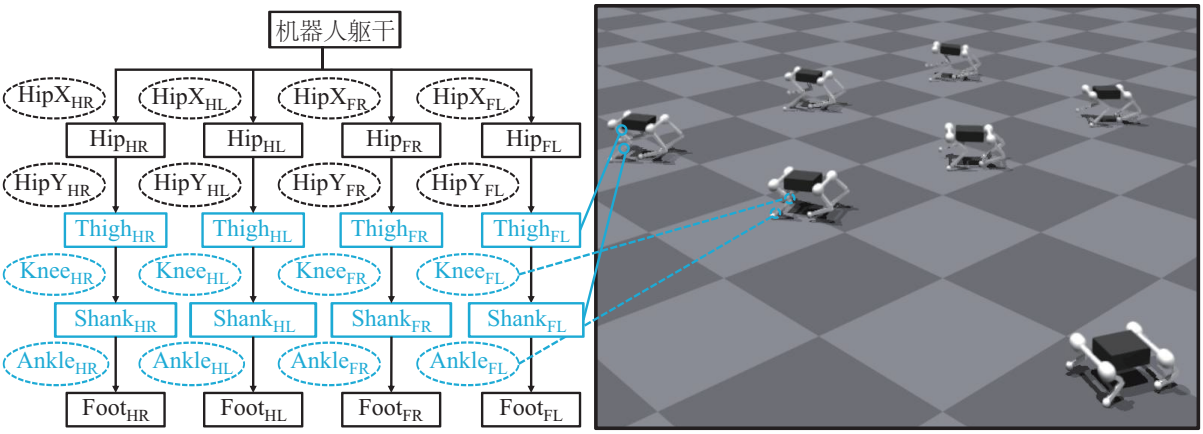
Tab.1　Design of the reward function

奖励项	公式	权重
目标追踪函数	$\min(\langle\boldsymbol{v},\hat{\boldsymbol{d}}_w\rangle,v^{\mathrm{cmd}})$ $\hat{\boldsymbol{d}}_w=\dfrac{\boldsymbol{p}-\boldsymbol{x}}{\lVert\boldsymbol{p}-\boldsymbol{x}\rVert}$	1.5
足端间隙	$-\sum_{i=0}^{4}c_i\cdot M(\boldsymbol{p}_i)$	−1.0
偏航角追踪函数	$\exp\{-\lVert\boldsymbol{\omega}_z-\boldsymbol{\omega}_z^{\mathrm{des}}\rVert_1\}$	0.5
线速度	v_z^2	−1.0
角速度	ω_{xy}^2	−0.05
动作改变项	$\lVert\boldsymbol{a}_t-\boldsymbol{a}_{t-1}\rVert_2$	−0.1
髋关节角度	$(\boldsymbol{q}_{\mathrm{hip}}-\boldsymbol{q}_{\mathrm{hip}}^{\mathrm{des}})^2$	−0.5
关节加速度	$\ddot{\boldsymbol{q}}^2$	-2.5×10^{-7}
关节目标项	$(\boldsymbol{q}-\boldsymbol{q}^{\mathrm{des}})^2$	−0.04
力矩改变项	$(\boldsymbol{\tau}_t-\boldsymbol{\tau}_{t-1})^2$	-1.0×10^{-7}
力矩惩罚项	$\boldsymbol{\tau}_t^2$	-1.0×10^{-5}
旋转惩罚项	$\boldsymbol{g}^2$	−1.0

1.2　预训练阶段

1.2.1　空间域随机化法

域随机化是一项提升强化学习（RL）模型泛化能力的关键技术，通过在指定的合理范围内对动态参数进行采样、并在新参数下训练模型，以增强模型的鲁棒性。这一方法先前主要用于弥合 RL 中仿真到现实（sim2real）的差距[22,24-25]，不同于已有方法，本文旨在获得具有良好泛化能力的预训练模型，从而为后续的微调阶段打下坚实基础。在已有方法中，域随机化涉及在训练过程中改变摩擦系数、电动机阻尼比等变量，这一过程在文中称为时间域随机化。然而，不同于上述易于更改的参数，机器人的结构参数在加载仿真器时就已经确定，对结构参数进行修改需要重新加载仿真器，如果在每一个训练回合中都进行修改，无疑会大大降低算法的训练速度。为了解决这个问题，本文引入了空间域随机化的概念，利用 Isaac Gym 仿真器的大规模并行能力，在同一个环境中同时训练数千个不同腿长的机器人。这种方法能够在短短几小时内开发出具有良好泛化能力的运动控制策略。



(a) 四足机器人的 URDF 模型 (b) 不同结构的四足机器人

图 3 四足机器人的 URDF 模型以及不同结构四足机器人的示意图

Fig.3 Schematic of the URDF model of the quadruped robots, and the quadruped robots with different structures

本文通过修改机器人的 URDF（united robotics description format）文件来创建具有不同结构参数的机器人仿真模型。四足机器人的 URDF 结构如图 3(a) 所示，其中实线矩形代表机器人的连杆，虚线椭圆代表机器人的关节。尽管所提方法适用于修改机器人各个构件的结构参数，如身体长度、宽度等，但是为了更加清晰地分析结构优化的结果，选择仅关注大腿和小腿长度，如图 3(a) 中的蓝色部分所示。对于每一个需要修改的连杆和关节，待更改的具体参数详见表 2，其中，$i=0$ 代表左/右前大腿，$i=1$ 代表左/右前小腿，$i=2$ 代表左/右后大腿，$i=3$ 代表左/右后小腿，l_i 代表每条腿的原始长度，m_i 代表每条腿的原始质量，腿部的碰撞模型采用具有正方形截面的长方体表示，b_i 代表截面正方形的边长，z_{knee} 代表膝关节在 z 方向的原始位置，z_{ankle} 代表踝关节在 z 方向的原始位置。为了确保机器人的稳定性，令机器人左右腿相应部分保持对称。因此，可调节的参数是前大腿、前小腿、后大腿、后小腿的缩放因子。在训练泛化性模型时，允许缩放因子在可行范围内随机取值，即 $\xi_i \sim U(c_{\min}, c_{\max})$。得到的参数 ξ_i 随后用于构建具有不同腿长的机器人，如图 3(b) 所示。

表 2 URDF 模型的待修改参数

Tab.2 The modified parameters of the URDF model

标签	属性	参数	值
连杆	惯性	原点 (x,y,z)	$(0,\ 0,\ \frac{-l_i\times\xi_i}{2})$
		质量	$m_i\times\xi_i$
		转动惯量	x 轴：$\frac{m_i}{12}(b_i^2+l_i^2)$，$y$ 轴：$\frac{m_i}{12}(b_i^2+l_i^2)$，$z$ 轴：$\frac{m_i b_i^2}{6}$
	视觉	原点 (x,y,z)	$(0,\ 0,\ \frac{-l_i\times\xi_i}{2})$
		几何形状	尺寸：$b_i, b_i, l_i\times\xi_i$
	碰撞	原点 (x,y,z)	$(0,\ 0,\ \frac{-l_i\times\xi_i}{2})$
		几何形状	尺寸：$b_i, b_i, l_i\times\xi_i$
膝关节	/	原点 (x,y,z)	$(0,\ 0,\ z_{\text{knee}}\times\xi_{i,i=0,2})$
踝关节	/	原点 (x,y,z)	$(0,\ 0,\ z_{\text{ankle}}\times\xi_{i,i=1,3})$

当机器人肢体长度发生变化时，PD 参数也需要要进行相应的修改。因此，借鉴 Luo 等[26] 提出的方法，采用多项式来计算修正后的比例因子：

$$\eta_i = a\times\xi_i^3 + b\times\xi_i^2 + c\times\xi_i + d \tag{1}$$

其中，a、b、c 和 d 为手工指定的参数，鉴于文中对大腿和小腿长度进行了修改，对应于图 3(a) 中所示的HipY 和Knee 关节，因此将修正因子$_i$应用示的 HipY 和 Knee 关节，因此将修正因子 η_i 应用于这 8 个关节的 PD 参数，具体如下：

$$k_{\text{p}}^i = \eta_i\times\tilde{k}_{\text{p}}^i,\quad k_{\text{d}}^i = \eta_i\times\tilde{k}_{\text{d}}^i \tag{2}$$

其中，$\tilde{k}_{\text{p}}^i$ 和 $\tilde{k}_{\text{d}}^i$ 分别代表原始参数，取值分别为 40 和 0.7。

1.2.2 折扣正则化法

在 1.2.1 节中，介绍了空间域随机化方法，该方法使得算法能够使用具有不同腿长机器人的数据进行训练。然而，如文 [27-28] 所述，在多个机器人上训练的价值网络有可能对训练时的数据过拟合，导致对未访问状态的泛化能力差，这不仅阻碍了训练时算法的性能，还对新环境中的测试性能产生负面影响。为了缓解这一问题，采用折扣正则化法将折扣因子调整为一个较小的值 γ_{reg}，其中 $0<\gamma_{\text{reg}}<\gamma<1$。折扣因子的修改会影响 PPO 算法

中优势函数 A_t 和时间差分 δ_t 的计算，修改后的计算公式如下所示：

$$A_t^{\gamma_{\text{reg}}} = \sum_{l=0}^{\infty} (\gamma_{\text{reg}}\lambda)^l \delta_{t+l}^{\gamma_{\text{reg}}} \tag{3}$$

$$\delta_t^{\gamma_{\text{reg}}} = r_t + \gamma_{\text{reg}} V_\phi(s_{t+1}) - V_\phi(s_t) \tag{4}$$

其中，λ 是在广义优势估计（GAE）中使用的超参数[29]，ϕ 代表评论家网络（Critic）的参数。从理论上来讲，较低的折扣因子减少了对未来奖励的依赖，使得算法专注于短期收益，该方法通过降低方差来改善模型的泛化能力。进一步的，优势函数和时间差分的更改会影响 PPO 算法中演员网络（Actor）和 Critic 损失函数的计算，修改后的损失函数如下所示：

$$L_{\text{Actor}} = E_{\tau\sim\pi_{\theta_{\text{old}}}}\left[\min\left(\frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}A_t^{\gamma_{\text{reg}}}, f_{\text{clip}}\left(\frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}, 1-\varepsilon, 1+\varepsilon\right)A_t^{\gamma_{\text{reg}}}\right)\right] \tag{5}$$

$$L_{\text{Critic}} = E_{s_t\sim\tau}\left[\frac{1}{2}\left(r_t + \gamma_{\text{reg}} V_\phi(s_{t+1}) - V_\phi(s_t)\right)^2\right] \tag{6}$$

其中，θ 代表 Actor 网络的参数，τ 是使用策略 $\pi_{\theta_{\text{old}}}$ 采样得到的轨迹，ε 是一个超参数，表示进行截断的范围，$f_{\text{clip}}(x,l,r) = \max(\min(x,r),l)$，即把 x 限制在 $[l,r]$ 内。

1.3　微调阶段

1.3.1　基于预训练模型的微调

通过集成空间域随机化和折扣正则化法，能获得适用于不同腿长机器人的控制策略。然而，该策略不能保证在每种结构参数下均为最优的。为了给结构优化算法提供更为可靠的适应度，对每一组候选结构参数进行微调。与训练泛化性模型不同，本节对训练过程进行了两项修改：1) 去除空间域随机化，专注于特定结构参数下的训练；2) 将课程训练的难度级别提升至最高。这些修改能够令微调后的控制策略更加适合特定结构参数的机器人。

1.3.2　结构参数优化

结构参数优化的目标是在指定的跑酷任务 $\kappa \in K$ 和给定的控制策略 $\pi \in \Pi$ 下，找到最优的结构参数 $\xi \in \Xi$，使得目标函数 $f(\xi|\kappa,\pi): \Xi\times K\times\Pi \to \mathbb{R}$ 最大化。本文主要关注跳高和跳远任务，记为 $K = \{\kappa_{\text{high}}, \kappa_{\text{long}}\}$。优化目标（也称为适应度）$f(\cdot)$ 被设置为机器人在单个回合中非折扣累计奖励值，如下所示：

$$f = \frac{1}{N}\sum_{i=0}^{N}\sum_{t=0}^{T} r(s_t^i, a_t^i) \tag{7}$$

其中，T 表示一个回合中的步数，N 代表与环境交互的机器人的数量。通常，ξ 相对于 $f(\cdot)$ 是不可微分的，原因在于腿部运动涉及到接触导致的离散变化。因此，利用黑盒优化方法——贝叶斯优化（BO），来进行结构参数的优化，目标优化过程为

$$\xi^* = \arg\max_{\xi\in\Xi} f(\xi|\kappa,\pi_\xi) \tag{8}$$

BO 算法是一个迭代的过程，在每次迭代中，选取一个候选结构参数 ξ，基于选定的结构参数，RL 算法在预训练模型的基础上对控制策略进行微调，当在该结构参数下获得微调后的模型 π_ξ 后，具有结构参数 ξ 的机器人在 π_ξ 的指导下与环境互动，以获得 $f(\cdot)$，然后将该结果反馈至 BO 算法，迭代继续进行，直至完成指定轮数，获得最优结构参数以及对应的控制策略。

2　实验结果（Experimental results)

2.1　实验设置

在算法的训练过程中，采用 Isaac Gym 作为仿真器。网络构建基于 PyTorch 库，并利用 NVIDIA GeForce RTX 3090 GPU 进行加速。利用控制策略生成电动机的目标角度，并使用 PD 控制器将目标角度转化为目标力矩。控制策略的频率为 50 Hz，PD 控制器以 200 Hz 运行，最终执行力矩的计算公式如下所示[7,9,20]。

$$\tau_{\text{target}}^i = k_{\text{p}}^i(q_i^* - q_i) + k_{\text{d}}^i(\dot{q}_i^* - \dot{q}_i) \tag{9}$$

$$\tau_{\text{scale}}^i = \tau_{\text{target}}^i \times \alpha_i \tag{10}$$

$$\tau_{\text{real}}^i = f_{\text{clip}}(\tau_{\text{scale}}^i, -\tau_{\max}, \tau_{\max}) \tag{11}$$

其中，i 为电动机编号，τ_{target}^i 为目标输出力矩，k_{p}^i 和 k_{d}^i 分别为 PD 控制器的比例系数和微分系数，如式 (2) 所示。q_i^* 和 q_i 分别为电动机 i 的目标角度和当前实际角度，$\dot{q}_i^*$ 为电动机 i 的目标角速度，通常设置为 0，$\dot{q}_i$ 为电动机 i 的当前实际角速度，α_i 为电动机 i 的阻尼比，该值在训练过程中会随着训练回合的进行而不断变化，以缓解强化学习中的 sim2real 问题。$\tau_{\max}$ 为最大关节力矩，τ_{real}^i 为对关节力矩施加限制后的实际力矩，也是电动机最终的执行力矩。

采用杭州云深处科技有限公司的绝影 Lite3 作为机器人的基础构形，训练过程中，机器人数量用 N 表示，这也意味着在预训练阶段，使用空间域随机化方法生成了 N 个不同腿长的机器人。具体而言，如 1.2.1 节所示，在环境初始化时，首先在 $U(0.6,1.4)$ 范围内选取 $\xi_{i,i=0,1,2,3}$，获得 N 组

$[\xi_0,\xi_1,\xi_2,\xi_3]$ 参数后，根据表 2 更新原始 URDF 文件，以获取 N 组不同的 URDF 文件，随后将其加载至仿真器中。

2.2 控制策略性能比较

2.2.1 域随机化方法比较

为了验证空间域随机化方法的有效性，设计了 3 组对比实验以供比较。

- **时间域随机化方法**：该方法在训练过程中改变机器人的结构参数，具体而言，该方法从采样范围内均匀采样了 81 组结构参数。从第 1 组结构参数开始训练，当完成总体训练过程的 1/81 时，切换到下一组结构参数，以此类推，直至训练结束。
- **空间域随机化方法**：本文所提方法，使用 N 组不同腿长的机器人同时收集交互数据以训练策略。
- **无随机化方法**：不采用随机化策略，所有机器人共享相同的默认结构参数，即 $\xi_{i,i=0,1,2,3}=[1,1,1,1]$。

为了验证 3 种域随机化方法的性能，本文随机采样了 100 组结构参数，并记录了 3 种方法在这些结构参数下获得的平均累计奖励值。此外，还使用独立 T 检验方法对比了所提的空间域随机化方法与另外 2 种方法，对比结果见表 3。显而易见，所提方法获得了最高的平均累计奖励值，并且与 2 种对比方法之间的 P 值均小于 0.05 的阈值，在统计学意义上证明了所提方法具有显著的优越性。这些结果表明，空间域随机化方法有效提升了控制策略的泛化性，而无需额外的计算资源。

表 3 域随机化方法对比

Tab.3 Comparison of domain randomization methods

	时间域随机化法	空间域随机化法	无随机化法
平均值	3.51 ± 0.30	15.58 ± 0.67	9.91 ± 0.37
P 值		4.42×10^{-5}	4.55×10^{-4}

2.2.2 正则化方法比较

为了验证折扣正则化方法在提升控制策略性能方面的有效性，设计了如下实验。

- **激活正则化方法**：向 Critic 目标函数中添加正则化项，旨在惩罚较大的值估计，这种方法通过减小虚假近似误差的影响，潜在地促进了模型的泛化性。
- **折扣正则化方法**：本文所提方法，修改 PPO 算法中的折扣因子 γ 为 γ_{reg}，经过多次测试，选择 $\gamma_{\text{reg}}=0.98$。
- **无正则化方法**：该方法不进行正则化，保留原始的 γ 值 0.99。
- **阶段性策略梯度方法**：用于增强 RL 算法泛化能力的 SOTA（state-of-the-art）方法，在训练 PPO 算法期间整合辅助损失，以提升算法的泛化性，详见 Cobbe 等[28] 的研究。

为了比较不同正则化方法的性能，图 4 绘制了在训练过程中每种方法的累计奖励曲线。显而易见，折扣正则化方法取得了最佳结果，其次是激活正则化方法，而阶段性策略梯度方法的表现不佳，甚至落后于不使用任何正则化技术的方法。综合各种方法的有效性以及实现的简洁性，折扣正则化方法在运动控制任务中被证明是最优的。值得注意的是，阶段性策略梯度方法表现最差，推测这可能是因为该方法是为基于图像的 RL 算法开发的，可能不适用于存在低维观测的运动控制任务。

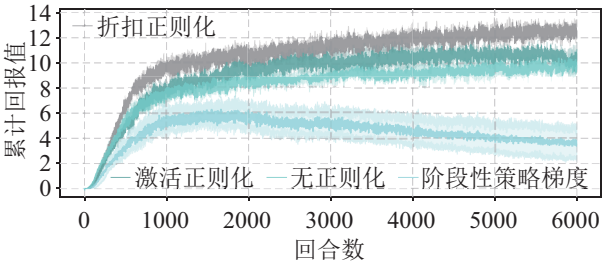


图 4 不同正则化方法在训练期间的累计奖励曲线

Fig.4 Cumulative reward curves during training with different regularization methods

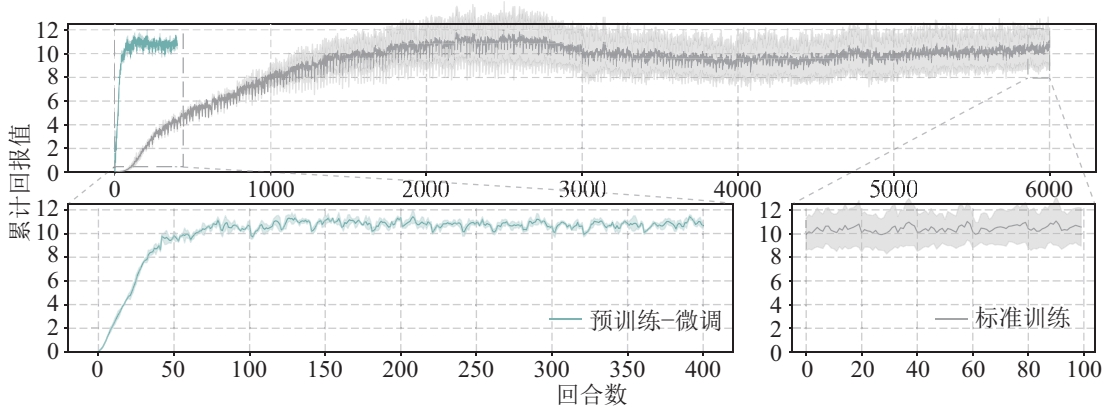
2.2.3 训练效率比较

本节通过以下 2 种方法进行对比实验，验证所提方法在训练效率上的优势。

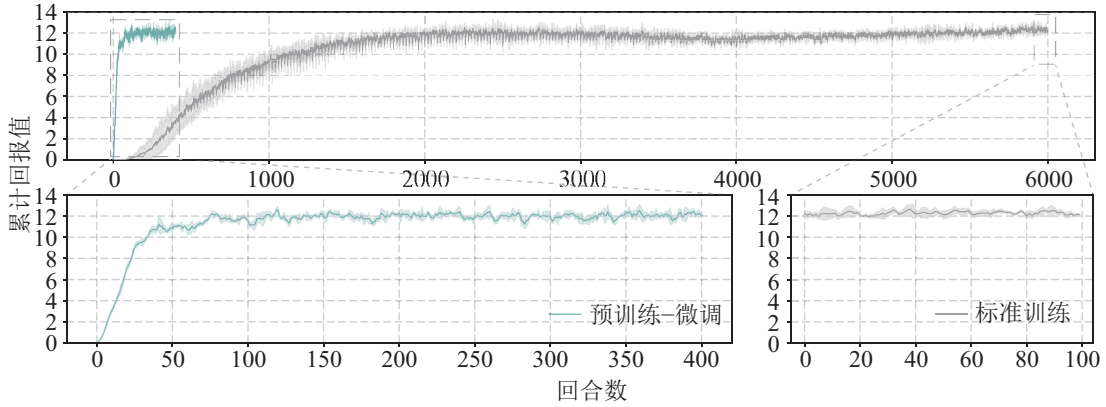
- **标准训练方法**：在特定结构参数下训练控制策略，不涉及预训练。
- **预训练—微调方法**：本文所提方法，首先使用空间域随机化方法结合正则化技术进行预训练，随后在特定结构参数下进行微调。

由图 5 可知，标准训练在第 2 000 个回合才开始收敛，而微调仅在第 100 个回合就开始收敛。这一结果证明了所提方法可以显著加速训练过程。例如，当需要获得 30 种不同结构参数下的策略时，标准训练需要 $30\times6000=1.8\times10^5$ 个回合，而微调仅需要 $6000+30\times400=1.8\times10^4$ 个回合，其中 6 000 代表预训练/标准训练的步骤，400 代表针对特定结构参数量身训练的微调步骤，仅相当于标准训练的 6.67%。

此外，对 2 种方法进行了性能分析，如图 5 所示。对于结构 1，2 种方法的累计回报值的收敛值



(a) 结构 1 累计回报值曲线对比



(b) 结构 2 累计回报值对比

图 5　训练效率比较实验结果

Fig.5　Comparative experimental results of training efficiency

均为 10 左右，对于结构 2，2 种方法的收敛值均为 12 左右。同时计算了 2 种方法最后 100 个奖励值的均值，并评估其差异。具体而言，对于结构 1，2 种方法的差异为 2.88%，对于结构 2，2 种方法的差异为 2.08%。这些结果表明，所提方法在保持与标准训练相当的性能的同时，训练时间仅为标准训练的 1/10，证明了所提方法的有效性。

2.3　协同设计性能比较

本节将所提方法与 3 种协同设计的基线算法进行比较，下面介绍这些方法。

- **基于结构参数的在线策略方法（在线 PCOD-P）[6]：**该方法将结构参数视为特权信息，在训练过程中取值范围与 2.1 节保持一致，均为 $U(0.6,1.4)$。该方法假设采用基于结构参数的策略可以获得接近最优的性能。在本文的跑酷任务中，由于在每个回合中修改结构参数难以实现，因此采用了时间域随机化的方法来训练控制策略，所采用的 81 组结构参数如图 6(a) 所示，训练顺序为从上至下，从左至右。
- **基于结构参数的离线策略方法（离线 PCOD-P）[3]：**使用在线 RL 算法收集各种结构参数下的交互数据，然后使用离线 RL 算法从这些数据中创建一个通用策略，作为结构参数优化过程中各个最优控制策略的替代。具体来说，在本文中，使用了 PPO 算法为 16 种典型结构参数（采用了图 6(b) 除去中间点外对角线上的参数）训练单独的控制策略，然后使用 TD3BC 算法 [30] 训练通用策略。
- **具身感知变换器方法（EAT）[5]：**类似于离线 PCODP 方法，该方法最初为各种结构参数训练单独的控制策略以收集交互数据，然后使用基于 Transformer 架构的网络对这些数据进行训练，最终获得具有泛化能力的控制策略。
- **预训练−微调方法：**本文所提方法。首先训练具有泛化能力的初始模型，然后基于该模型，对不同结构参数下的控制策略进行微调。

本文从取值范围内均匀选取了 81 种不同的结构参数，并在图 6 中绘制了不同方法在这些结构参数下的累计奖励，每个方块代表在特定结构参数下多任务中使用相应控制策略所获得的累计奖励值，其颜色越深，代表累计奖励值越高，表明算法

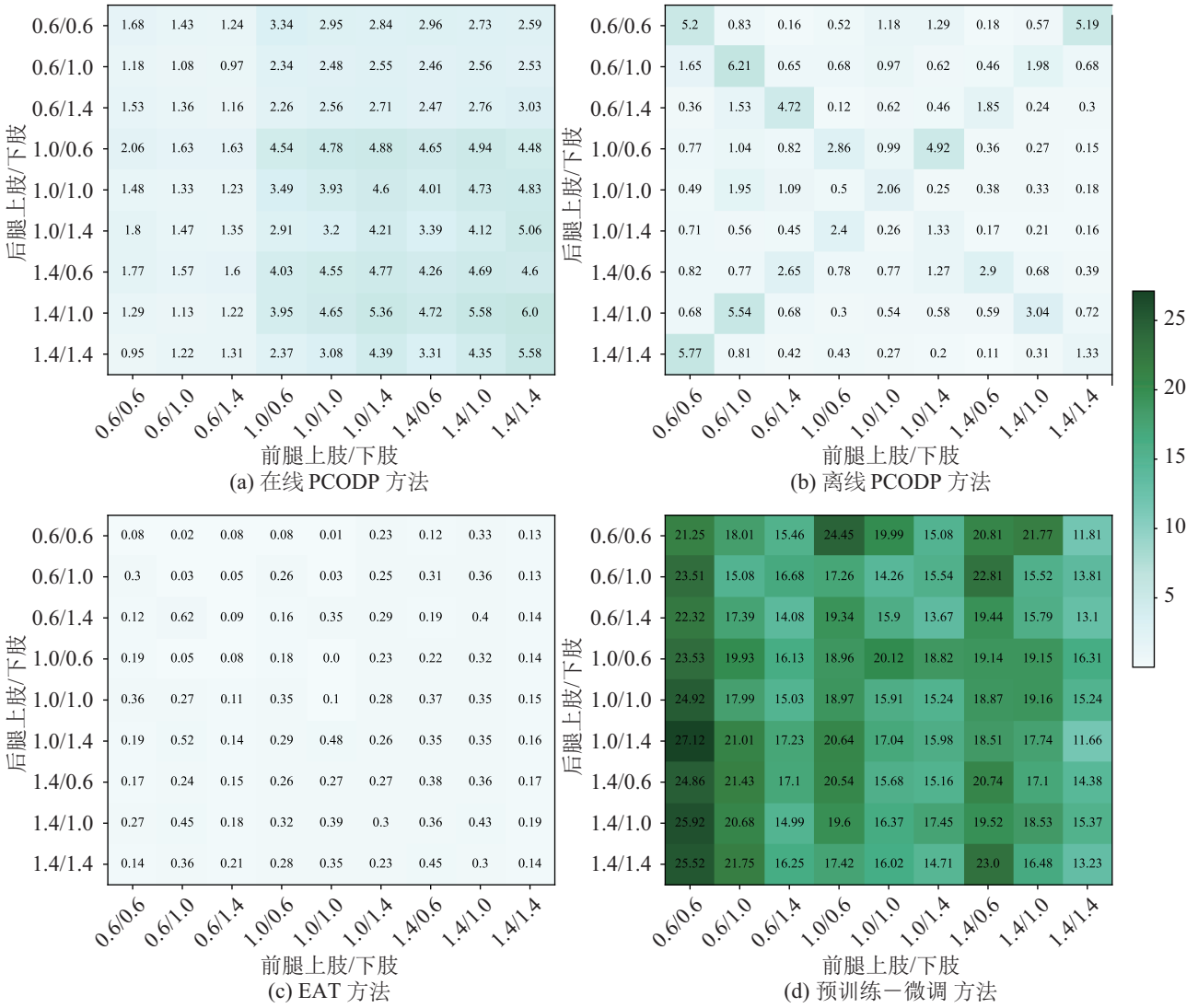


图 6 协同设计对比实验热力图

Fig.6 Heatmap of co-design experiments

的性能越好。可以看出，所提方法在各种结构参数下均取得了最高的奖励值，表明预训练－微调方法可以为结构参数优化提供更有效的控制策略。就在线 PCODP 方法而言，如图 6(a) 所示，其较高的奖励值集中在最后训练的结构参数下，对于先前遇到的结构参数，该方法获得较低的累计奖励，这些结果表明，该方法倾向于遗忘之前结构参数下的控制策略，转而专注于最新结构参数下的策略。就离线 PCODP 方法而言，如图 6(b) 所示，其较高的奖励值出现在训练期间遇到过的 16 种典型结构参数附近，对于训练阶段未遇到的结构参数，该方法获得较低的累计奖励值，证明了该方法难以泛化到训练期间未见过的结构参数。至于 EAT 方法，如图 6(c) 所示，其在所有结构参数上的表现都比较差，原因可能在于原始 EAT 方法关注的是具有低维状态的简单平地行走任务，状态仅包含身体线速度和角速度等，在机器人跑酷任务中，地形信息至关重要，但却使得状态维度更高，导致基于 Transformer 架构的算法失效。

进一步的，本文将上述协同设计方法应用于单独的跳高和跳远任务中，并记录了单个回合内每种方法获得的累计奖励值，如表 4 所示。显然，所提方法在 2 个任务中均取得了最高的累计奖励值，是其他基线算法的 3 倍以上，这证明了在协同设计的背景下，所提出的预训练－微调框架的有效性。该表还记录了各种方法训练消耗的时间。其中，在线 PCODP 方法和所提方法消耗的时间包括训练控制策略以及贝叶斯优化所用时间，而离线 PCODP 方法和 EAT 方法包含训练专家控制策略、收集专家数据、训练通用控制器以及贝叶斯优化时间，本文方

法由于在贝叶斯优化期间引入微调导致了一些额外的时间损耗。然而，从总体上看，所提方法所消耗的时间仍远低于离线 PCODP 和 EAT 方法，时间效率提升了 90%，综合考虑性能与时间效率，所提方法的优势显而易见。

表 4　协同设计算法在不同任务中的累计奖励值和耗时对比

Tab.4　Comparison of accumulated reward values in different tasks and time comparison of co-design algorithms

	跳远任务	跳高任务	耗时
在线 PCODP 方法	6.18	10.37	**3 h 5 min**
离线 PCODP 方法	4.78	4.63	86 h 45 min
EAT 方法	0.66	0.71	86 h 17 min
预训练－微调方法	**28.23**	**29.06**	8 h 51 min

2.4　机器人跑酷性能比较

2.4.1　单任务下的比较

不同运动任务对机器人的动力学性能具有特定要求，因此在各类任务下，优化结果会有所不同。对这些差异进行深入分析，能够为特定任务的需求提供更为精确的设计方案。

Robot Parkour Learning[7] 和 Extreme Parkour[9] 是当前先进的跑酷算法。其中，前者在杭州宇树科技公司 A1 机器人平台上实现了 0.6 m 的极限跳远距离和 0.4 m 的极限跳跃高度，后者在同一个机器人平台上则实现了 0.8 m 极限跳远距离和 0.5 m 极限跳跃高度。因此，本文选择性能更为优越的 Extreme Parkour 算法与所提方法进行对比分析。鉴于可变结构机器人的实物实验受到硬件条件的限制，选择在仿真环境中对这 2 种方法进行验证，并在绝影 Lite3 平台上对 Extreme Parkour 算法重新进行仿真实验，以确保机器人的物理结构与所提方法的一样，从而增强比较的合理性。表 5 详细记录了在成功率达到 100% 时，2 种方法所实现的最高跳跃高度和最远跳远距离。通过对 Extreme Parkour 与预训练－微调方法的对比分析，能够进一步确认，除了提升运动控制器的性能，适当的机械结构优化同样能够显著改善四足机器人在极限运动任务中的表现。

表 5　仿真环境跑酷性能对比

Tab.5　Parkour performance comparison in simulation environment

	跳远距离 /m	跳跃高度 /m
Extreme Parkour 算法[9]	0.87	0.47
预训练－微调方法	1.00	0.55

此外，图 7 展示了机器人在跳远和跳高两项极限任务中，结构参数优化前后的表现，优化前后的结构参数如图 7 中左侧所示。对于每个任务，第 1 行和第 2 行的图片分别展示了在跳跃相同距离/高度时，原始结构与优化结构下机器人的运动过程。显而易见，未经优化的结构在两项任务中均未能成功（机器人出现跌倒现象），而结构优化后的机器人则在这两项任务中均表现出色。具体来说，在跳远任务中，结构优化后的机器人配备了较长的后腿，这可以提供更强大的推进力，而较短的前腿则可以帮助机器人抓住前方地面以防止跌落；在跳高任务中，机器人的前后腿均得到了延长，这有助于机器人攀爬。两项实验都强调了协同设计方法在释放机器人跑酷性能潜力方面的可能性。鉴于特定结构通常仅适用于特定运动场景，未来可以考虑探索新型可伸缩的腿部构件，以实现可变结构的足式机器人[31]。这将使机器人能够在不同场景下动态调节其构形，从而进一步提升其适用性。

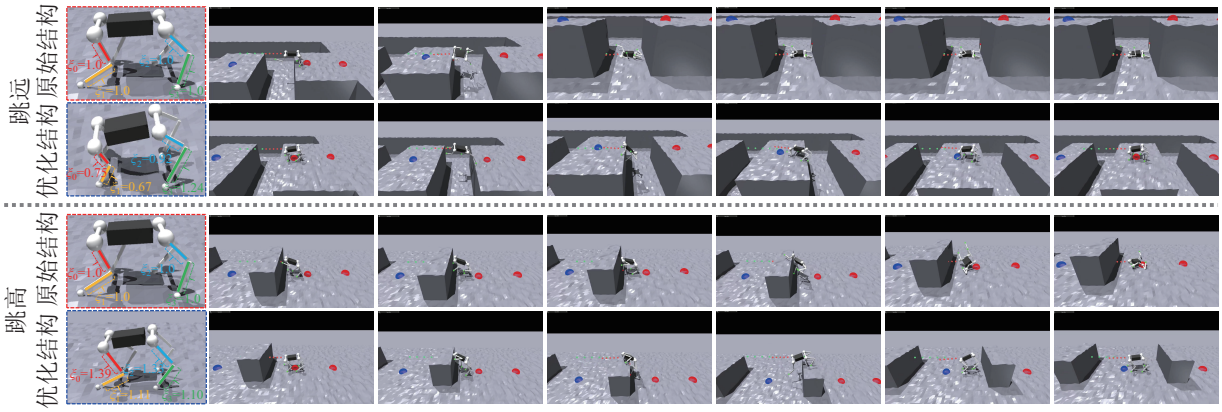


图 7　跳高和跳远任务中不同结构参数下机器人在同一时刻运动状态的对比

Fig.7　Comparison of robot motion states at the same moment under different structural parameters in high jump and long jump tasks

2.4.2 多任务下的比较

目前，受机械结构的限制，机器人在不同任务之间难以灵活切换腿长。因此，探讨四足机器人在多任务环境下的协同设计，旨在说明在刚性腿的情况下，机器人腿长该如何进行设计。微调过程同时包含了跳高和跳远所需的 2 种实验地形。考虑到强化学习算法中出现的不可避免的随机性，选择胫腓比作为定性分析的比较指标。胫腓比定义为胫骨（小腿）与股骨（大腿）长度之比乘以 100。研究表明，跳跃性能与胫腓比呈正相关，人类的平均胫腓比约为 80，而在擅长跳跃的动物中，这一比值通常超过 100[32]。测量微调过程中胫腓比高于和低于 100 的结构参数的适应度，并采用 Levene 检验法和 T 检验法进行数据分析。表 6 中的结果显示，胫腓比更高的结构在适应度上的表现更优越，这与自然观察结果相符。

表 6 不同胫腓比下的适应度取值

Tab.6 Fitness values of different crural index

胫腓比		Levene 检验法 P 值[1]	T 检验法 P 值[2]
>100	$\leqslant 100$		
29.177 ± 2.02	27.898 ± 1.76	0.608	0.027

[1] Levene 检验法用于评估数据集的方差是否相似，值大于 0.05 表示方差具有相似性。

[2] 进一步使用 T 检验法分析这两组数据，P 值低于 0.05 则表明它们在统计上显著不同。

此外，由杭州云深处科技有限公司设计的标准的机器人，其大腿长度为 200 mm，小腿长度为 210 mm，这进一步验证了实验结果与人工经验的一致性，为现有机器人的设计提供了数据支持。

3 结论（Conclusion）

提出了一种预训练－微调框架，用于机器人结构与控制的协同设计，旨在提升四足机器人在极限任务中的运动性能。在预训练阶段，采用空间域随机化与折扣正则化方法，训练出一种具有良好泛化能力的控制策略。微调阶段则嵌入到贝叶斯优化的迭代过程中，使预训练模型能够在每一轮贝叶斯优化中针对候选结构参数进行快速调整。这一方法不仅确保了在不同结构参数条件下策略的最优性，还显著提高了算法的训练效率。

在单独的跳高和跳远任务中对所提方法与基线算法进行了比较，证明了除了提高运动控制器性能外，适当的机械结构优化同样能够改善四足机器人在极限运动任务中的表现。此外，本文还进行了多任务环境下的协同设计，以胫腓比作为定性指标，验证了所提方法获得的结果与自然观察以及现有机器人设计的一致性，进一步证明了所提方法的实际价值。

目前，四足机器人多采用刚性腿部设计，在面对不同任务时腿长不可调，本研究的多任务实验可为刚性腿长的设计提供指导。然而，正如单任务实验所示，不同任务对应的最优腿长存在差异，因此，未来可探索新型可伸缩的腿部组件，研制出自适应结构的四足机器人，进一步增强其适用性。

参考文献（References）

[1] GUPTA A, SAVARESE S, GANGULI S, et al. Embodied intelligence via learning and evolution[J]. Nature Communications, 2021, 12(1). DOI: 10.1038/s41467-021-25874-z.

[2] BELMONTE-BAEZA Ã, LEE J, VALSECCHI G, et al. Meta reinforcement learning for optimal design of legged robots[J]. IEEE Robotics and Automation Letters, 2022, 7(4): 12134-12141.

[3] CHEN C, XIANG P Y, LU H J, et al. C^2: Co-design of robots via a concurrent-network coupling online and offline reinforcement learning[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2023: 7487-7494.

[4] CHEN C, XIANG P Y, ZHANG J Y, et al. Deep reinforcement learning based co-optimization of morphology and gait for small-scale legged robot[J]. IEEE/ASME Transactions on Mechatronics, 2024, 29(4): 2697-2708.

[5] YU C, ZHANG W N, LAI H, et al. Multi-embodiment legged robot control as a sequence modeling problem[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2023: 7250-7257.

[6] BJELONIC F, LEE J, ARM P, et al. Learning-based design and control for quadrupedal robots with parallel-elastic actuators[J]. IEEE Robotics and Automation Letters, 2023, 8(3): 1611-1618.

[7] ZHUANG Z, FU Z, WANG J, et al. Robot parkour learning[DB/OL]. (2023-09-12) [2024-09-13]. https://arxiv.org/abs/2309.05665.

[8] HOELLER D, RUDIN N, SAKO D, et al. ANYmal parkour: Learning agile navigation for quadrupedal robots[J]. Science Robotics, 2024, 9(88). DOI: 10.1126/scirobotics.adi7566.

[9] CHENG X X, SHI K X, AGARWAL A, et al. Extreme parkour with legged robots[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2024: 11443-11450.

[10] RUDIN N, HOELLER D, BJELONIC M, et al. Advanced skills by learning locomotion and local navigation end-to-end[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2022: 2497-2503.

[11] HA S, COROS S, ALSPACH A, et al. Joint optimization of robot design and motion parameters using the implicit function theorem[C]//Robotics: Science and Systems XIII. 2017. DOI: 10.15607/RSS.2017.XIII.003.

[12] GEILINGER M, PORANNE R, DESAI R, et al. Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels[J]. ACM Transactions on Graphics, 2018, 37(4): 1-12.

[13] FADINI G, FLAYOLS T, DEL PRETE A, et al. Computational design of energy-efficient legged robots: Optimizing for size and actuators[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2021: 9898-9904.
[14] HEJNA III D J, ABBEEL P, PINTO L. Task-agnostic morphology evolution[DB/OL]. (2021-02-25) [2024-09-13]. https://arxiv.org/abs/2102.13100.
[15] WANG T W, ZHOU Y H, FIDLER S, et al. Neural graph evolution: Towards efficient automatic robot design[DB/OL]. (2019-06-12) [2024-09-13]. https://arxiv.org/abs/1906.05370.
[16] LUCK K S, AMOR H B, CALANDRA R. Data-efficient co-adaptation of morphology and behaviour with deep reinforcement learning[C]//Conference on Robot Learning. PMLR, 2020: 854-869.
[17] SCHAFF C, YUNIS D, CHAKRABARTI A, et al. Jointly learning to construct and control agents using deep reinforcement learning[C]//International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2019: 9798-9805.
[18] NAGARAJAN U, MAMPETTA A, KANTOR G A, et al. State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile robot[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2009: 998-1003.
[19] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[DB/OL]. (2017-08-28) [2024-09-13]. https://arxiv.org/abs/1707.06347.
[20] RUDIN N, HOELLER D, REIST P, et al. Learning to walk in minutes using massively parallel deep reinforcement learning [C]//Conference on Robot Learning. PMLR, 2022: 91-100.
[21] KUMAR A, FU Z P, PATHAK D, et al. RMA: Rapid motor adaptation for legged robots[C]//Robotics: Science and Systems XVII. 2021. DOI: 10.15607/RSS.2021.XVII.011.
[22] JI G, MUN J, KIM H, et al. Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion[J]. IEEE Robotics and Automation Letters, 2022, 7(2): 4630-4637.
[23] NAHRENDRA I M A, YU B, MYUNG H. DreamWaQ: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2023: 5078-5084.
[24] TAN J, ZHANG T, COUMANS E, et al. Sim-to-real: Learning agile locomotion for quadruped robots[C]// Robotics: Science and Systems XIV. 2018. DOI: 10.15607/RSS.2018.XIV.010.
[25] SHI H J, ZHOU B, ZENG H S, et al. Reinforcement learning with evolutionary trajectory generator: A general approach for quadrupedal locomotion[J]. IEEE Robotics and Automation Letters, 2022, 7(2): 3085-3092.
[26] LUO Z R, DONG Y Z, LI X Q, et al. Moral: Learning morphologically adaptive locomotion controller for quadrupedal robots on challenging terrains[J]. IEEE Robotics and Automation Letters, 2024, 9(5): 4019-4026.
[27] MOON S, LEE J, SONG H O. Rethinking value function learning for generalization in reinforcement learning[DB/OL]. (2021-01-08) [2024-09-13]. https://arxiv.org/abs/2210.09960.
[28] COBBE K W, HILTON J, KLIMOV O, et al. Phasic policy gradient[C]//International Conference on Machine Learning. PMLR, 2021: 2020-2027.
[29] SCHULMAN J, MORITZ P, LEVINE S, et al. High-dimensional continuous control using generalized advantage estimation[DB/OL]. (2018-10-20) [2024-09-13]. https://arxiv.org/abs/1506.02438.
[30] FUJIMOTO S, GU S S. A minimalist approach to offline reinforcement learning[DB/OL]. (2021-12-03) [2024-09-13]. https://arxiv.org/abs/2106.06860.
[31] NYGAARD T F, MARTIN C P, TORRESEN J, et al. Real-world embodied AI through a morphologically adaptive quadruped robot[J]. Nature Machine Intelligence, 2021, 3(5): 410-419.
[32] DAVENPORT C B. The crural index[J]. American Journal of Biological Anthropology, 1933, 17: 333-353.

作者简介：

陈　词（1994 –），女，博士。研究领域：深度强化学习，四足机器人，计算机视觉。

王　越（1988 –），男，博士，教授。研究领域：机器人自主导航，智能操作，计算机视觉，深度学习。