ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 39,No. 5,Sep. 2024,pp. 1182－1191
DOI:10. 16337/j. 1004-9037. 2024. 05. 010

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

# 基于多任务强化学习的地形自适应模仿学习方法

余　昊[1]，梁宇宸[2]，张　驰[2]，刘跃虎[2]

(1. 西安交通大学软件学院，西安 710049;2. 西安交通大学人工智能学院，西安 710049)

**摘　要：**地形自适应能力是智能体在复杂地形条件下稳定运动的基础,而由于机器人动力学系统的复杂性,传统逆动力学方法通常难以使其具备这种能力。现有利用强化学习在解决序列决策问题上的优势训练智能体地形适应能力的单任务学习方法无法有效学习各类地形中的相关性。事实上,复杂地形自适应任务可以认为是一种多任务,子任务间的关系可以用不同地形影响因素来衡量,通过子任务模型的相互学习解决数据分布信息获取不全面的问题。基于此,本文提出一种多任务强化学习方法。该方法包含1个由子任务预训练模型组成的执行层和1个基于强化学习方法、采用软约束融合执行层模型的决策层。在LeggedGym地形仿真器上的实验证明,本文方法训练的智能体运动更加稳定,在复杂地形上的摔倒次数更少,并且表现出更好的泛化性能。

**关键词：**多任务学习;模仿学习;强化学习;地形影响因素;LeggedGym地形仿真器
**中图分类号：**TP18　　**文献标志码：**A

## Terrain-Adaptive Motion Imitation Based on Multi-task Reinforcement Learning

YU Hao[1], LIANG Yuchen[2], ZHANG Chi[2], LIU Yuehu[2]

(1. School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China;2. College of Artificial Intelligence, Xi'an Jiaotong University, Xi'an 710049, China)

**Abstract:** Terrain adaptive ability is the basis for the stable movement of agents under complex terrain conditions. Due to the complexity of the dynamical systems of these agents, such as humanoid robots, it is usually difficult for traditional inverse dynamics methods to have such ability. Recent research has used the advantages of reinforcement learning in solving sequential decision-making problems to train agents to adapt to terrain. However, these single-task learning methods cannot effectively learn the correlation in various terrains. In fact, complex terrain adaptive tasks can be considered as a multi-task problem, and the relationship between sub-tasks can be measured by different terrain factors. And then, the problem of incomplete acquisition of data distribution information can be solved by mutual learning of sub-task models. Therefore, this paper proposes a multi-task reinforcement learning method. It contains an execution layer which is consist of pre-trained subtask models and a decision layer based on reinforcement learning method. Moreover, the decision layer uses soft constraints to fuse models of the execution layer. Experiments on LeggedGym terrain simulator prove that the agent trained by the method in this paper is more stable in movement and has fewer falls down on complex terrains, showing better generalization performance.

**Key words:** multi-task learning; learning by imitation; reinforcement learning; terrain influencing factor; LeggedGym terrain simulator

---

**基金项目：**科技创新2030“新一代人工智能”重大项目(2018AAA0102504)。
**收稿日期：**2023-02-28;**修订日期：**2023-10-29

## 引　言

地形自适应能力是智能体在复杂地形保持稳定运动的基础。地形自适应技术增强了机器人在复杂地形下的运动性能,可用于外附骨骼可保证残障人士在日常生活中的安全行走,同时该技术也被用于角色动画或游戏引擎,使角色在复杂地形上的运动更加自然[1-5]。苏黎世联邦理工的机器人系统实验室设计了一个包含5类地形影响因素的复杂地形仿真器LeggedGym,并采用游戏激励课程策略通过大规模并行训练,使智能体获得地形自适应技能[6]。DeepMimic方法[7]以地势高度图和腿式机器人本体状态为输入,奖励函数通过模仿奖励和任务奖励使腿式机器人获得地形适应能力。然而,这一类单任务学习模式的不同类型地形适应任务是彼此独立的,忽略了任务之间的潜在共享因素,不同任务模型之间无法相互学习,获取到的任务数据分布不够全面,当出现新的复杂地形时,模型需要重新训练。事实上,复杂地形适应任务可以因其含有的不同地形影响因素种类被看作是一种多任务,子任务被认为是对特定地形影响因素的适应任务,因此可以利用多任务学习的优势来解决。基于以上想法,本文针对地形自适应问题,提出一种地形自适应运动模仿的多任务学习方法。该方法利用地形影响因素种类衡量子任务关系,在子任务上预训练适应性策略组成执行层,采用强化学习训练的策略作为决策层,根据地形信息和决策层奖励,建立多个子任务之间的共享因素表示,融合执行层策略。

具体地,执行层为在单一影响因素所构成的地形上预训练的符合高斯分布的策略集合,根据地形高度图和智能体当前状态输出最优动作。决策层策略被建模为一种混合高斯模型,它根据输入地形高度图和智能体的本体状态信息,融合执行层策略完成模型融合,使得融合模型能够根据地形高度图和智能体自身状态信息输出关节坐标。然后利用PD控制器将关节坐标转化为关节力矩作用于智能体。在LeggedGym地形仿真器下的智能体运动结果可视化示例如图1所示。

本文工作的主要贡献包括两个方面:(1)提出一种多任务学习强化学习解决机器人复杂地形自适应任务,有效利用单一影响因素地形适应任务模型间的互学习提高运动模仿性能的稳定性;(2)一个可扩展的地形适应模型,可以利用与任务无关的运动剪辑和地形高度图训练可重复使用的地形适应策略。

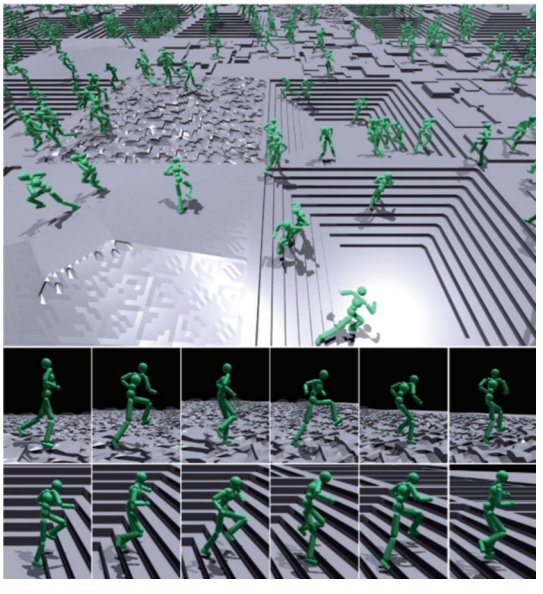

图1　本文方法训练智能体在LeggedGym地形仿真器上的运动可视化

Fig.1　Motion visualization of agents on the LeggedGym terrain simulator by using the proposed method

## 1　相关工作

在LeggedGym地形仿真器的实验设置中,复杂地形被认为是含有以下影响因素的地形:discrete, stairs down, stairs up, rough slope, smooth slope等[6]。现有的方法中,DeepMimic采用近端策略优化(Proximal policy optimization, PPO)算法,将地势高度图和腿式机器人自身的状态作为输入[7],并将奖励函数分为模仿奖励和任务奖励来使腿式机器人具备了地形适应能力。Merel-GAIL是一种修改的生成式对抗模仿学习(Generative adversarial imitation learning, GAIL)方法[8]。该方法对原本GAIL中策略的输入进行简化,使其不再关注上一时刻输出的动作,而只关注上一时刻的状态,简化了计算量,同时提出了一种分层强化学习方法来解决地形自适应任务。对抗性运动先验(Adversarial

motion priors, AMP)采用生成对抗模仿学习方法使机器人模仿示教运动轨迹[9],并在此基础上加入了一个任务奖励,使其完成地形适应任务,然而由于其采用了生成对抗网络,导致模型效率差,容易出现模式崩溃现象。DeepMimic采用PPO算法,通过游戏激励课程策略在IsaacGym平台首次实现大规模并行训练,使得ANYmal等机器人具备了地形适应能力[7],然而由于其奖励函数设计得过于复杂,难以将其迁移到其他机器人,例如人形机器人。

多任务学习可以将一个复杂任务按照合理的衡量因素分解成多个相关子任务,对子任务分别训练相应模型,最后通过软约束或硬约束实现模型融合。多任务学习可以有效利用子任务的相关性,促进子任务间的相互学习,以及为新任务的学习提供额外信息,使融合后的模型具有更好的表现效果和鲁棒性[10]。

## 2 基本定义

在本文的方法中,执行层的预训练和决策层的策略融合都被建模为强化学习问题,其中智能体采取相应的策略$\pi$与环境交互,以优化给定的目标。在每一时间步$t$,智能体根据自身状态从策略中采样一个动作$a_t\sim\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$,然后智能体执行该动作,并根据环境动态$\boldsymbol{s}_{t+1}\sim p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)$和标量奖励$r_t=r(s_t,a_t,s_{t+1})$产生一个新状态$\boldsymbol{s}_{t+1}$。智能体的目标是学习一个策略,最大化目标收益$J(\pi)$[11-15]为

$$J(\pi)=E_{\pi_\theta}\left[\sum_{t=0}^{T-1}\gamma^t r_t\right] \tag{1}$$

$$\pi_\theta=p(\boldsymbol{\tau}|\pi)=p(\boldsymbol{s}_0)\prod_{t}^{T-1}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)\pi(\boldsymbol{a}_t|\boldsymbol{s}_t) \tag{2}$$

式中:$\theta$表示策略$\pi$的参数;$p(\boldsymbol{\tau}|\pi)$表示采取策略$\pi$时,智能体产生相应轨迹$\boldsymbol{\tau}=\{\boldsymbol{s}_0,\boldsymbol{a}_0,\boldsymbol{r}_0,\boldsymbol{s}_1,\boldsymbol{s}_1,\boldsymbol{a}_1,\boldsymbol{r}_1,\boldsymbol{s}_2\cdots,\boldsymbol{s}_{T-1},\boldsymbol{a}_{T-1},\boldsymbol{r}_{T-1},\boldsymbol{s}_T\}$的概率;$p(\boldsymbol{s}_0)$表示初始的状态分布;$T$表示轨迹的时间范围;$\gamma\in[0,1]$表示折扣因子。

在本文方法中,执行层和决策层都用到了PPO算法[16],这是一种从置信域策略优化(Trust region policy optimization, TRPO)[17]改进来的新型策略梯度算法。原本策略梯度算法对步长十分敏感,但又难以选择合适步长,PPO提出的新目标函数实现了训练步骤的小批量更新,解决了策略梯度算法中步长难以确定的问题,也解决了TRPO计算过程复杂、每一步更新运算量大的问题,能够采用更少的时间达到甚至超越TRPO训练的智能体。

自从PPO算法被提出,OpenAI和DeepMind等实验室都将其作为强化学习的首选算法来使用。PPO算法的实现方式分为两种:PPO惩罚和PPO截断。PPO惩罚用拉格朗日乘数法直接将KL散度的限制放进了目标函数中,使其变成了一个无约束的优化问题,在迭代的过程中不断更新KL散度前的系数为

$$\arg\max_{\theta}\mathbb{E}_{\boldsymbol{s}\sim\nu^{\pi_{\theta_k}}}\mathbb{E}_{\boldsymbol{a}\sim\pi_{\theta_k}(\cdot|\boldsymbol{s})}\left[\frac{\pi_\theta(\boldsymbol{a}|\boldsymbol{s})}{\pi_{\theta_k}(\boldsymbol{a}|\boldsymbol{s})}A^{\pi_{\theta_k}}(\boldsymbol{s},\boldsymbol{a})-\beta D_{\mathrm{KL}}[\pi_{\theta_k}(\cdot|\boldsymbol{s}),\pi_\theta(\cdot|\boldsymbol{s})]\right] \tag{3}$$

式中:$\theta_k$表示策略$\pi$第$k$次迭代时的参数;$A$为优势函数。

PPO截断更加直接,它在目标函数中进行限制,以保证新的参数和旧的参数的差距不会太大,即

$$\arg\max_{\theta}\mathbb{E}_{\boldsymbol{s}\sim\nu^{\pi_{\theta_k}}}\mathbb{E}_{\boldsymbol{a}\sim\pi_{\theta_k}(\cdot|\boldsymbol{s})}\left[\min\left(\frac{\pi_\theta(\boldsymbol{a}|\boldsymbol{s})}{\pi_{\theta_k}(\boldsymbol{a}|\boldsymbol{s})}A^{\pi_{\theta_k}}(\boldsymbol{s},\boldsymbol{a}),\ \mathrm{clip}\left(\frac{\pi_\theta(\boldsymbol{a}|\boldsymbol{s})}{\pi_{\theta_k}(\boldsymbol{a}|\boldsymbol{s})},1-\epsilon,1+\epsilon\right)A^{\pi_{\theta_k}}(\boldsymbol{s},\boldsymbol{a})\right)\right] \tag{4}$$

式中:clip(·)为截断函数;$\epsilon$表示截断的范围。

虽然强化学习的奖励函数提供了一种交互方式,然而仅通过设计一个有效奖励函数使智能体完成

复杂地形适应任务,通常运动效果难以保证[18]。这是因为针对任务的单个策略对环境探索难以做到完备,获得的任务数据分布不完整,难以保证模型性能。

## 3　本文方法

基于多任务学习模式,本文提出一种地形自适应运动模仿的多任务强化学习方法,如图2所示。该方法的整体架构包含决策层和执行层两部分,均采用PPO算法。决策层用于环境探索,并依据智能体的状态-地势对$\{\boldsymbol{s},\boldsymbol{h}\}$和决策层奖励$r$采用软约束方式输出混合系数$\{w_1,w_2,\cdots,w_n\}$,将多个任务模型融合,产生混合动作$\boldsymbol{a}_{\mathrm{mix}}$,使各个单一影响因素地形适应任务模型能够相互学习,优化提升融合模型在任意影响因素的地形适应任务效果。执行层由多个预训练策略构成,这些策略在单一影响因素地形适应任务中由示教数据训练。执行层在每一时刻智能体的状态-地势对$\{\boldsymbol{s},\boldsymbol{h}\}$输出智能体动作$\boldsymbol{a}$,由PD控制器将动作转化为对应关节的关节力矩。

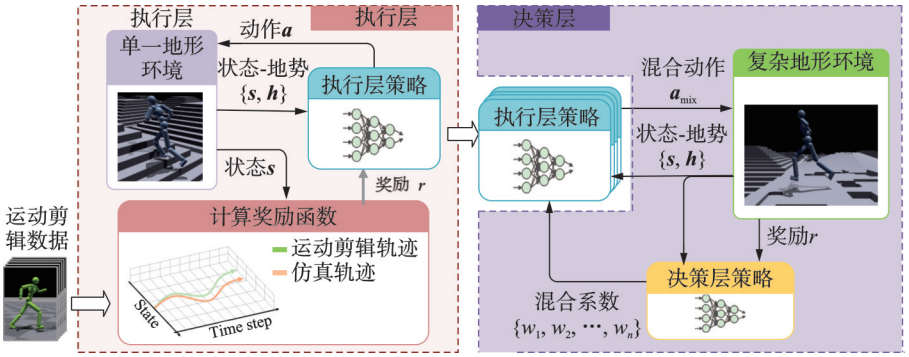


图2　本文方法框架图

Fig.2　Framework of the proposed method

### 3.1　策略融合

本文方法将执行层构建为在多个在单一影响因素构成的地形上(子任务)预训练策略的集合$\Pi=\{\pi_b^1(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{h}),\pi_b^2(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{h}),\cdots,\pi_b^n(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{h})\}$,这些策略在单一的地形上具备一定适应性。每个预训练策略$\pi_b^n$由一个3层全连接网络构成,它将智能体自身状态$\boldsymbol{s}$和地势高度图$\boldsymbol{h}$映射到动作$a=\{j_p^1,j_p^2,\cdots,j_p^k\}$的分布$\pi_b^n(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{h})$上,$j_p^k$为第$k$个关节的位置。本文采用高斯分布建模的执行层策略$\pi_b^n(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{h})$,具有参数化的均值$\mu(\boldsymbol{s},\boldsymbol{h})$,和一个固定的对角线协方差矩阵$\boldsymbol{\Sigma}$,即有

$$\pi_b^n(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{h})=N(\boldsymbol{a}|\mu_n(\boldsymbol{s},\boldsymbol{h}),\boldsymbol{\Sigma}_n) \tag{5}$$

执行层策略输出关节位置,并由PD控制器转化为关节力矩,其中PD控制器采用Isaac Gym内部派生的隐式spring公式来解算关节力矩。决策层策略$\pi_d(\Omega|\boldsymbol{s},\boldsymbol{h})$,根据地形高度图$\boldsymbol{h}$和智能体自身状态信息$\boldsymbol{s}$输出混合系数$\boldsymbol{\Omega}=\{w_1,w_2,\cdots,w_n\}$,加权执行层预训练策略构成混合策略$P(\boldsymbol{a}_{\mathrm{mix}}|\boldsymbol{s},\boldsymbol{h})$。

每一时间步所采取的动作$\boldsymbol{a}_{\mathrm{mix}}$为

$$\begin{aligned}&\operatorname{argmax}P(\boldsymbol{a}_{\mathrm{mix}}|\boldsymbol{s},\boldsymbol{h})=\operatorname{argmax}\sum_n\omega_n\pi_b^n(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{h})=\operatorname{argmax}\sum_n\omega_nN(\boldsymbol{a}_{\mathrm{mix}}|\mu_n(\boldsymbol{s},\boldsymbol{h}),\boldsymbol{\Sigma}_n)=\\&\operatorname{argmax}\sum_n\omega_n\frac{1}{\sqrt{(2\pi)^k\det(\boldsymbol{\Sigma}_n)}}\exp\left[-\frac{1}{2}(\boldsymbol{a}_{\mathrm{mix}}-\boldsymbol{\mu}_n)^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{a}_{\mathrm{mix}}-\boldsymbol{\mu}_n)\right]\end{aligned} \tag{6}$$

### 3.2　执行层设计

执行层是多个在单一影响因素地形适应任务中预训练策略构成的集合。执行层的每一个策略是

一个3层全连接网络，每层包含512个神经元，采用ReLU作为激活函数，使用Adam优化器，学习率为5e−5，采用PPO算法的截断实现形式，截断范围$\epsilon=0.25$，折扣因子$\gamma=0.99$。执行层关注点在于：使智能体在地形因素影响下运动更加自然，同时使预训练的策略能够有彼此的任务侧重，为最终的融合策略提供更为全面的任务数据分布信息，因此执行层通过在每个单一影响因素构成的地形中通过模仿示教运动轨迹建立预训练策略集合。执行层通过最大化累积位姿奖励$r_{\mathrm{p}}$来保证智能体在子任务复现示教轨迹，位姿奖励$r_{\mathrm{p}}$计算公式为

$$r_{\mathrm{p}}=w_{\mathrm{r}}r_{\mathrm{r}}+w_{\mathrm{jo}}r_{\mathrm{jo}}+w_{\mathrm{jv}}r_{\mathrm{jv}}+w_{\mathrm{ee}}r_{\mathrm{ee}},\ w_{\mathrm{r}}=0.1,\ w_{\mathrm{jo}}=0.6,\ w_{\mathrm{jv}}=0.1,\ w_{\mathrm{ee}}=0.2 \tag{7}$$

式中根节点奖励$r_{\mathrm{r}}$包含根节点角度奖励$r_{\mathrm{rr}}$、根节点位置奖励$r_{\mathrm{location}}$、根节点线速度奖励$r_{\mathrm{linv}}$和根节点角速度奖励$r_{\mathrm{angv}}$，即

$$r_{\mathrm{r}}=r_{\mathrm{rr}}+r_{\mathrm{location}}+r_{\mathrm{linv}}+r_{\mathrm{angv}} \tag{8}$$

根节点角度奖励$r_{\mathrm{rr}}$鼓励智能体与示教的运动方向保持一致，根节点角度由世界坐标下的四元数表示，目的是给予智能体方位信息。$\boldsymbol{q}_{\mathrm{r}}$为智能体根节点角度四元数，$\hat{\boldsymbol{q}}_{\mathrm{r}}$为示教根节点角度的四元数，$\boldsymbol{q}_{\mathrm{r}}\ominus\hat{\boldsymbol{q}}_{\mathrm{r}}$表示四元数运算，$\|\boldsymbol{q}\|$计算四元数围绕其轴的标量旋转弧度，则有$r_{\mathrm{rr}}$的计算公式为

$$r_{\mathrm{rr}}=\exp\left[-300\left(\|\boldsymbol{q}_{\mathrm{r}}\ominus\hat{\boldsymbol{q}}_{\mathrm{r}}\|^2\right)\right] \tag{9}$$

根节点位置奖励$r_{\mathrm{location}}$计算智能体根节点的局部坐标$\boldsymbol{p}_{\mathrm{w}}$与示教的局部坐标$\hat{\boldsymbol{p}}_{\mathrm{w}}$的距离，有

$$r_{\mathrm{location}}=\exp\left[-300\left(\|\boldsymbol{p}_{\mathrm{w}}-\hat{\boldsymbol{p}}_{\mathrm{w}}\|^2\right)\right] \tag{10}$$

根节点线速度奖励$r_{\mathrm{linv}}$计算智能体根节点的线速度$\boldsymbol{v}_{\mathrm{rlinv}}$与示教根节点的线速度$\hat{\boldsymbol{v}}_{\mathrm{rlinv}}$的差值为

$$r_{\mathrm{linv}}=\exp\left[-0.1\left(\|\boldsymbol{v}_{\mathrm{rlinv}}-\hat{\boldsymbol{v}}_{\mathrm{rlinv}}\|^2\right)\right] \tag{11}$$

根角速度奖励$r_{\mathrm{angv}}$为智能体根节点的角速度$\boldsymbol{v}_{\mathrm{rangv}}$与示教根节点的角速度$\hat{\boldsymbol{v}}_{\mathrm{rangv}}$的差值为

$$r_{\mathrm{angv}}=\exp\left[-0.1\left(\|\boldsymbol{v}_{\mathrm{rangv}}-\hat{\boldsymbol{v}}_{\mathrm{angv}}\|^2\right)\right] \tag{12}$$

关节位姿奖励$r_{\mathrm{jo}}$鼓励智能体匹配示教各关节位姿。式(13)中$q_{\mathrm{jo}}^k$和$\hat{q}_{\mathrm{jo}}^k$分别表示智能体和示教第$k$个关节位姿的四元数，$q_{\mathrm{jo}}^k\ominus\hat{q}_{\mathrm{jo}}^k$表示四元数运算，有

$$r_{\mathrm{jo}}=\exp\left[-2\left(\sum_k\|q_{\mathrm{jo}}^k\ominus\hat{q}_{\mathrm{jo}}^k\|^2\right)\right] \tag{13}$$

关节角速度奖励$r_{\mathrm{jv}}$鼓励智能体匹配示教各关节角速度，$v_{\mathrm{jv}}^k$为智能体第k个关节的角速度。目标速度$\hat{v}_{\mathrm{jv}}^k$是通过有限差分法从示教中计算出来的。关节角速度奖励计算为

$$r_{\mathrm{jv}}=\exp\left[-0.1\left(\sum_k\|v_{\mathrm{jv}}^k-\hat{v}_{\mathrm{jv}}^k\|^2\right)\right] \tag{14}$$

末端执行器指手、脚、头部等部位，奖励$r_{\mathrm{ee}}$鼓励智能体手、脚、头部与示教相匹配。所有末端执行器位置采用局部坐标表示。$p_{\mathrm{ee}}^k$与$\hat{p}_{\mathrm{ee}}^k$分别表示智能体与示教第$k$个末端执行器的位置，有

$$r_{\mathrm{ee}}=\exp\left[-20\left(\sum_k\|p_{\mathrm{ee}}^k-\hat{p}_{\mathrm{ee}}^k\|^2\right)\right] \tag{15}$$

### 3.3 决策层设计

决策层采用PPO截断实现方式，在网络构建与超参数选择上与决策层一致。决策层根据复杂地形高度图和决策层奖励函数，将执行层中的预训练策略融合，通过执行层策略间的信息共享实现相互学习，获得关于复杂地形适应任务更为全面的数据分布信息，使智能体获得复杂地形的适应能力。式(16)为决策层奖励函数$r_{\mathrm{d}}$，它由任务奖励$r_{\mathrm{t}}$和位姿奖励$r_{\mathrm{p}}$构成，即有

$$r_{\mathrm{d}}=w_{\mathrm{t}}r_{\mathrm{t}}+w_{\mathrm{p}}r_{\mathrm{p}} \tag{16}$$

任务奖励$r_{\mathrm{t}}$保证了智能体完成任务所需要的对环境进行的充分探索,中和执行层中根节点角度奖励$r_{\mathrm{rr}}$采用世界坐标对智能体运动轨迹的限制,并鼓励智能体更快速的适应地形,计算公式为

$$r_{\mathrm{t}}=w_{\mathrm{linv}}r_{\mathrm{linv}}+w_{\mathrm{dis}}r_{\mathrm{dis}} \tag{17}$$

速度奖励$r_{\mathrm{linv}}$鼓励智能体以更快的速度探索并适应地形,权重$w_{\mathrm{linv}}$取值为0.5。智能体速度用速度在世界坐标下$x$轴的分量$v_x$,在$y$轴的分量$v_y$和在$z$轴上的分量$v_z$表示,$r_{\mathrm{linv}}$计算公式为

$$r_{\mathrm{linv}}=\ln\left[20\left(\|v_x+v_y+v_z\|^2\right)\right] \tag{18}$$

位置奖励$r_{\mathrm{dis}}$鼓励智能体向地形边缘移动,保证智能体更加充分地探索地形,权重$w_{\mathrm{dis}}$取值为0.5。智能体与地形边缘的距离计算为智能体根节点坐标$r_{\mathrm{l}}$与地形中心边缘位置$t_{\mathrm{l}}$的L2距离,$r_{\mathrm{dis}}$的具体形式为

$$r_{\mathrm{dis}}=\ln\left[20\left(\|r_{\mathrm{l}}-t_{\mathrm{l}}\|^2\right)\right] \tag{19}$$

位姿奖励$r_{\mathrm{p}}$与执行层中的位姿奖励形式基本一致,但是根奖励$r_{\mathrm{r}}$直接以根节点角度奖励$r_{\mathrm{rr}}$代替,目的是减少示教数据对智能体运动状态的限制,在决策层进行策略融合时,保证智能体自然的运动状态。具体算法如算法1所示。

**算法1**:地形自适应模仿的多任务强化学习算法伪代码

(1) $\theta$←随机权值

(2) $\psi$←随机权值

(3) $\mathrm{II}_e$←执行策略$\left\{\pi\frac{1}{e},\pi\frac{2}{e},\cdots,\pi\frac{n}{e}\right\}$

(4) while not done do

(5)　$\boldsymbol{s}_0$←从参考运动采样初始状态

(6)　初始化智能体状态$\boldsymbol{s}_0$

(7)　for step $=1,2,\cdots,m$, do

(8)　　$\boldsymbol{s}$←开始状态

(9)　　$\boldsymbol{h}$←开始的地形图

(10)　　$\boldsymbol{\Omega}\{w_1,w_2,\cdots,w_n\}\sim\pi\theta(\boldsymbol{\Omega}\{w_1,w_2,\cdots,w_n\}|\boldsymbol{s},\boldsymbol{h})$

(11)　　$\boldsymbol{a}_{\mathrm{mix}}$←通过式(13)计算混合策略,并采取混合策略$\boldsymbol{a}_{\mathrm{mix}}$推进仿真到下一时间步

(12)　　$\boldsymbol{s}'$←结束状态

(13)　　$r$←通过式(14)计算奖励,并将$(\boldsymbol{s},\boldsymbol{h},\boldsymbol{\Omega},\boldsymbol{r},\boldsymbol{s}',\boldsymbol{h})$保存进内存$D$

(14)　end for

(15)　$\theta_{\mathrm{old}}$←$\theta$

(16)　for 每一个更新步骤do

(17)　　从内存$D$中采样小批量$n$个数据$\{(\boldsymbol{s}_i,\boldsymbol{h}_i,\boldsymbol{\Omega}_i,\boldsymbol{r}_i,\boldsymbol{s}_i',\boldsymbol{h}_i')\}$

(18)　　更新值函数:

(19)　　for 每一个$(\boldsymbol{s}_i,\boldsymbol{h}_i,\boldsymbol{\Omega}_i,\boldsymbol{r}_i,\boldsymbol{s}_i',\boldsymbol{h}_i')$ do

(20)　　　$\boldsymbol{y}_i$←用时序误差TD($\lambda$)计算目标价值

(21)　　end for

(22)　　$\psi$←$\psi+\boldsymbol{a}_v(\frac{1}{n}\sum_i\nabla_\psi V_\psi(\{\boldsymbol{s}_i,\boldsymbol{h}_i\})(\boldsymbol{y}_i-V(\{\boldsymbol{s}_i,\boldsymbol{h}_i\}))$

(23) 更新策略:

(24) for 每一个$(\boldsymbol{s}_i,\boldsymbol{h}_i,\boldsymbol{\Omega}_i,\boldsymbol{r}_i,\boldsymbol{s}_i',\boldsymbol{h}_i')$ do

(25) $A_i \leftarrow$用$V_\phi$和GAE计算优势

(26) $w_i(\theta)\leftarrow\dfrac{\pi_\theta(\Omega_i|\boldsymbol{s}_i,\boldsymbol{h}_i)}{\pi_{\theta_{old}}(\Omega_i|\boldsymbol{s}_i,\boldsymbol{h}_i)}$

(27) end for

(28) $\theta\leftarrow\theta+\alpha\pi\dfrac{1}{n}\sum_i\nabla_\theta\min(w_i(\theta)A_i,\mathrm{clip}(w_i(\theta),1-\epsilon,1+\epsilon)A_i)$

(29) end for

(30) end while

## 4 实验与结果分析

本文在IsaacGym仿真环境设计实验,实验主要关注人形机器人(Humanoid)控制任务,该机器人包含28个自由度,远高于先前研究工作使用的ANYmal机器狗(只有12自由度),任务更具挑战性[19]。实验数据采用CMU动作捕捉数据集[20]。算法用PyTorch框架实现并使用了elegantRL[21]强化学习算法库,所有算法均在1块GTX titan xp上训练。

实验基于LeggedGym地形仿真器,构建了10×20的地形网格,每一列为相同地形,其中discrete为4列,stairs down为5列,stairs up为7列,rough slope为2列,smooth slope为2列,每一行地形难度等级相同,共分为0~9十级难度,由第1行开始向后地形难度等级上升,地形难度等级的上升意味着地势的高度差增加,其中最大地势高度差为2.25 m,对智能体运动的影响更大。

智能体初始化的位置全部在第1行,初始地形难度等级为0,运动方向为地形难度等级增加的方向。智能体通过相应难度地形时能够维持正常的运动状态,被认为适应了该难度等级的地形。此时若对智能体进行初始化,则初始地形难度等级加1,否则初始地形难度等级不变。能够引发智能体初始化的事件有两个:(1)智能体与环境的交互达到1 000时间步;(2)智能体无法维持正常运动状态而摔倒。

### 4.1 对比实验

本文通过在LeggedGym地形仿真器中与解决地形自适应问题的3种方法AMP、Merel-GAIL和DeepMimic进行对比实验,来评估方法的稳定性。3种对比方法和本文方法都构建了4 096个环境。4 096个智能体平均初始化在每一列的第1行位置,即地形难度等级为1的位置,运动的方向为地形难度等级增加的方向。

图3是本文方法和其他3种方法训练出来的4 096个智能体在复杂地形上,每次与环境交互时摔倒的智能体数量。其中横轴表示智能体每次与环境的交互(时间步),纵轴代表此次交互智能体摔倒的数量。随着智能体与环境交互次数的增加,智能体所处的地形难度等级呈现增加趋势,横轴向右地形难度呈现增加趋势。可以看出,本文方法训练出的智能体在每次与环境交互时摔倒的数量更少,地形适应能力更强。同时也可以看出,基于生成对抗模仿学习的两种方法:Merel-GAIL和AMP相比于基于PPO的本文方法和DeepMimic方法曲线波动更加剧烈,这说明生成对抗模仿学习方法容易出现模式崩溃,限制了智能体技能的多样性。

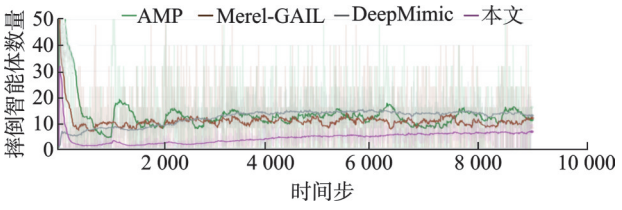


图3 对比实验结果

Fig.3 Comparative experimental results

### 4.2　消融实验

为验证本文方法能够促进单一任务模型,及执行层预训练模型间的相互学习,使得最终的融合模型相比于任意单一任务的模型在相应的单一影响因素地形上的效果更稳定,本文进行了相应的消融实验设计如下:(1)对照组,每种单一影响因素地形的4 096个仿真环境中的智能体采用与之相应的预训练策略;(2)实验组,将对照组的预训练策略作为执行层中的预训练策略,然后加入决策层策略进行融合得到融合策略,使该融合策略作为5种单一影响因素地形仿真环境中智能体的策略。

表2为消融实验结果。通过表2可以得出,融合策略训练出的智能体在5种单一影响因素地形中,每次与环境交互的摔倒平均次数都要比相应预训练策略少19%～50%。这表明本文方法的决策层可有效共享执行层单一任务模型信息,模型之间可以相互学习,优化并提升了最终融合模型性能,使得训练出的智能体更加稳定,同时在子任务上的效果也优于单一预训练模型。

**表2　消融实验结果**

**Table 2　Ablation experiment results**

| 地形影响因素 | 实验组摔倒智能体数量(融合策略) | 对照组摔倒智能体数量(预训练策略) |
|---|---|---|
| discrete | 2.11(↓45.90%) | 3.90 |
| stairs down | 15.23(↓18.95%) | 18.79 |
| stairs up | 8.38(↓30.69%) | 12.09 |
| rough slope | 1.69(↓46.18%) | 3.14 |
| smooth slope | 1.95(↓50.26%) | 3.92 |

### 4.3　泛化性能实验

在测试泛化效果的实验中,本文方法和其他3种方法的训练环境地形难度等级使用0～4五个等级。验证阶段的测试地形难度使用5～9五个等级,智能体均为4 096个。验证阶段统计4 096个智能体在每一时间步的摔倒数量,然后求均值和方差,如表3所示,该数据可用于反映不同方法将已知难度地形上训练模型泛化到未知难度地形能力。由表3可以看出,本文方法训练的智能体在5种地形上平均每一时间步的摔倒数量更少的同时,具有更小的方差;基于生成对抗模仿学习的方法Merel-GAIL和AMP相比于基于PPO的DeepMimic和本文方法具有更大的方差,这与对比实验结果一致,说明基于生成对抗模仿学习的方法容易出现模式崩溃问题。综上可以得出,本文方法训练的智能体更加稳定,训练的效果不会出现巨大波动,同时泛化性更好,能够将低难度等级地形上训练的模型更好地泛化到高难度等级的地形上。

**表3　泛化性能对比**

**Table 3　Comparison of generalization performance**

| 地形影响因素 | 本文方法 | | DeepMimic | | Merel-GAIL | | AMP | |
|---|---|---|---|---|---|---|---|---|
| | 均值 | 方差 | 均值 | 方差 | 均值 | 方差 | 均值 | 方差 |
| discrete | 3.01 | 4.73 | 4.91 | 5.21 | 7.66 | 123.94 | 19.84 | 157.35 |
| stairs down | 26.05 | 30.61 | 26.07 | 28.69 | 124.96 | 1019.46 | 71.62 | 546.86 |
| stairs up | 14.07 | 18.91 | 16.19 | 20.47 | 12.05 | 150.40 | 8.43 | 77.92 |
| rough slope | 2.51 | 2.62 | 4.58 | 4.78 | 6.42 | 74.30 | 3.62 | 29.97 |
| smooth slope | 2.28 | 3.77 | 6.60 | 6.95 | 6.98 | 111.74 | 3.60 | 29.18 |

## 5 结束语

本文提出了一种多任务强化学习方法,使人形机器人具备地形自适应能力。这项工作能够利用非结构化的运动剪辑数据和地形高度图使智能体学习一组可重用的技能及执行层策略。同时,基于多任务学习思想,提出了一种多策略融合方法及决策层策略。消融实验展示了决策策略能够有效利用多任务学习的优势,促进执行策略间的相互学习。泛化性实验展示了本文方法基于多任务学习模式的另一优势,能够使模型具备泛化性。然而,由于这项工作基于深度强化学习方法,因此奖励函数的设计难以考虑数据的所有分布情况,之后的工作将考虑基于生成对抗模仿学习。但是就像对比实验和泛化实验中对AMP和Merel-GAIL的讨论一样,生成对抗网络容易出现模式崩溃,以致于限制模型的运动多样性。所以下一步工作应避免模式崩溃问题,提高模型运动的多样性。此外,将仿真模仿迁移到真实机器人也将是未来研究的方向之一。

**参 考 文 献:**


[1] YUAN Y, KITANI K. Ego-pose estimation and forecasting as real-time PD control[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. [S.l.]: IEEE, 2019: 10082-10092.

[2] PENG X B, GUO Y, HALPER L, et al. ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters[J]. ACM Transactions on Graphics (TOG), 2022, 41(4): 1-17.

[3] JURAVSKY J, GUO Y, FIDLER S, et al. PADL: Language-directed physics-based character control[C]//Proceedings of SIGGRAPH Asia 2022 Conference. New York: ACM, 2022: 1-9.

[4] SONG S, KIDZIŃSKI Ł, PENG X B, et al. Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation[J]. Journal of Neuroengineering and Rehabilitation, 2021, 18: 1-17.

[5] SMITH L, KEW J C, PENG X B, et al. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world [C]//Proceedings of 2022 International Conference on Robotics and Automation (ICRA). [S.l.]: IEEE, 2022: 1593-1599.

[6] RUDIN N, HOELLER D, REIST P, et al. Learning to walk in minutes using massively parallel deep reinforcement learning [C]//Proceedings of Conference on Robot Learning. [S.l.]: PMLR, 2022: 91-100.

[7] PENG X B, ABBEEL P, LEVINE S, et al. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills[J]. ACM Transactions on Graphics (TOG), 2018, 37(4): 1-14.

[8] MEREL J, TASSA Y, TB D, et al. Learning human behaviors from motion capture by adversarial imitation[EB/OL]. (2017-07-10)[2023-02-28]. https://doi.org/10.48550/arXiv.1707.02201.

[9] PENG X B, MA Z, ABBEEL P, et al. AMP: Adversarial motion priors for stylized physics-based character control[J]. ACM Transactions on Graphics (TOG), 2021, 40(4): 1-20.

[10] RUDER S. An overview of multi-task learning in deep neural networks [EB/OL]. (2017-06-06)[2023-02-28]. https://doi.org/10.48550/arXiv:1706.05098.

[11] KAELBLING L P, LITTMAN M L, MOORE A W. Reinforcement learning: A survey[J]. Journal of Artificial Intelligence research, 1996, 4: 237-285.

[12] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. Cambridge, MA, USA: MIT Press, 2018.

[13] LI Y. Deep reinforcement learning: An overview[EB/OL]. (2018-11-26) [2023-02-28]. https://doi. org/10.48550/arXiv: 1810.06339.

[14] LIANG Y, LIU Y. Adaptive frequency hopping policy for fast pose estimation[C]//Proceedings of 2021 IEEE International Conference on Image Processing (ICIP). [S.l.]: IEEE, 2021: 324-328.

[15] WIERING M A, VAN OTTERLO M. Reinforcement learning[J]. Adaptation, Learning, and Optimization, 2012, 12(3): 729.

[16] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[EB/OL]. (2017-08-28)[2023-02-28]. https://doi.org/10.48550/arXiv.1707.06347.

[17] SCHULMAN J, LEVINE S, ABBEEL P, et al. Trust region policy optimization[C]//Proceedings of International Conference on Machine Learning. [S.l.]: PMLR, 2015: 1889-1897.

[18] LEVINE S, KUMAR A, TUCKER G, et al. Offline reinforcement learning: Tutorial, review, and perspectives on open problems[EB/OL]. (2020-11-01)[2023-02-28]. https://doi.org/10.48550/arXiv.2005.01643.

[19] MAKOVIYCHUK V, WAWRZYNIAK L, GUO Y, et al. ISAAC GYM: High performance gpu-based physics simulation for robot learning[EB/OL]. (2021-08-25)[2023-02-28]. https://doi.org/10.48550/arXiv.2108.10470.

[20] Carnegie Mellon University. CMU graphics lab motion capture database[EB/OL]. (2003-05-14)[2023-02-28]. http://mocap.cs.cmu.edu.

[21] LIU X Y, LI Z, YANG Z, et al. ElegantRL-Podracer: Scalable and elastic library for cloud-native deep reinforcement learning [EB/OL]. (2021-12-11)[2023-02-28]. https://doi.org/10.48550/arXiv.2112.05923.

**作者简介:**

**余昊**(1999-),男,硕士研究生,研究方向:强化学习、模仿学习和机器人的复杂地形自适应任务,E-mail:tsfking@stu.xjtu.edu.cn。

**梁宇宸**(1997-),男,博士研究生,研究方向:强化学习、模仿学习和人机协同,E-mail:liangyc@stu.xjtu.edu. cn。

**张驰**(1990-),男,博士,助理教授,研究方向:机器学习,计算机视觉与模式识别和混合增强智能。

**刘跃虎**(1962-),**通信作者**,男,博士,教授,研究方向:模式识别与计算机视觉、人机交互与混合智能和增强现实与仿真测试,E-mail:liuyh@xjtu.edu.cn。

(编辑:刘彦东)