doi:10.11887/j.issn.1001-2486.23050020 http://journal.nudt.edu.cn

元学习探索隐变量的强化学习方法

李艺颖[1,2],周 伟[3*]

(1. 军事科学院 国防科技创新研究院,北京 100071;2. 中国人民解放军32806部队,北京 100091;
3. 西安卫星测控中心,陕西 西安 710043)

摘 要:针对智能体传统探索工作对交互数据利用率低或需要其他任务数据的问题,创新性地引入一个表征当前任务特点的可在线学习的探索隐变量,辅助策略网络进行行为决策。无须其他多任务数据,也无须多余的当前任务的环境交互步,探索隐变量在所引入的可学习的环境模型中进行更新;而环境模型又进而通过智能体与真实环境的交互数据进行监督式更新。因此,探索隐变量即在模拟真实环境的模型中提前帮助"探路",该任务探索信息可帮助智能体在真实环境中加强探索、提高性能。实验在强化学习典型连续控制任务上有约30%的性能提升,对单任务探索工作和元强化学习研究具有指导和借鉴意义。

关键词:元强化学习;基于模型强化学习;隐变量;环境探索

中图分类号:TP181 **文献标志码**:A **文章编号**:1001-2486(2025)05-197-09

论文拓展

Reinforcement learning method via meta-learning the exploring latent variable

LI Yiying[1,2], *ZHOU Wei*[3*]

(1. National Innovation Institute of Defense Technology, Academy of Military, Beijing 100071, China;
2. The PLA Unit 32806, Beijing 100091, China; 3. Xi'an Satellite Control Center, Xi'an 710043, China)

Abstract: Aiming at the issues of low utilization of interaction data or the need for additional task data in traditional agent exploration work, an online-learnable exploration latent variable that characterizes the current task features to assist the policy network in behavioral decision-making was innovatively introduced. There was no need for additional multi-task data or additional environmental interaction steps in the current task. The exploring latent variable was updated in the learnable environment model, and the environment model underwent supervised updates based on the intelligent agent and real environment interaction data. The exploration in advance in the simulated environment model was assisted by the exploring latent variable, and thus the performance of agents in the real environment was improved. The performance in typical continuous control tasks was raised by about 30% in the experiments, which was of guiding significance for the single-task exploration and meta reinforcement learning research.

Keywords: meta reinforcement learning; model based reinforcement learning; latent variables; environment exploration

当前,强化学习(reinforcement learning,RL)在游戏、机器人等领域都有着出色表现。然而,如何有效地对环境进行探索、解决"探索-利用"(exploration-exploitation)困境[1-2]仍是该领域的重要挑战。"探索"是帮助智能体遍历更多未知的状态空间以收集更多环境信息,"利用"是基于当前的已知环境信息做决策以最大限度获取高收益。"利用"的方案已较为明确,而"探索"仍面临许多悬而未决的研究。当前常用的探索策略大多是基于动作空间中的随机抖动实现的,即在贪婪策略或确定性策略输出的动作上添加随机噪声,例如Q-learning[3]和深度Q网络(deep Q-network,DQN)[4]中使用的ϵ-greedy朴素探索;或是通过乐观初始估计[5]、不确定性优先[6-8]、概率匹配[9-12]、信息状态搜索[13-15]等方法实现。然而,这种基于动作空间的探索策略在简单强化学习任务[16-17]上效果尚可,在深度神经网络等复杂结构强化学习上的效果还难以满足需求。因此,逐渐

收稿日期:2023-05-21

基金项目:国家自然科学基金青年基金资助项目(62206307)

第一作者:李艺颖(1992—),女,山东淄博人,助理研究员,博士,E-mail:liyiying10@nudt.edu.cn

*通信作者:周伟(1990—),男,山西忻州人,工程师,博士,E-mail:zhouwei14@nudt.edu.cn

引用格式:李艺颖,周伟. 元学习探索隐变量的强化学习方法[J]. 国防科技大学学报,2025,47(5):197-205.

Citation: LI Y Y, ZHOU W. Reinforcement learning method via meta-learning the exploring latent variable[J]. Journal of National University of Defense Technology, 2025, 47(5): 197-205.

有工作聚焦于在参数空间[18-19]进行探索研究,例如,内在好奇心模块(intrinsic curiosity module, ICM)[20]、自动内在奖励塑造(automatic intrinsic reward shaping, AIRS)[21]通过内在奖励来鼓励特征参数上的探索;DeepMind 利用环境模型进行想象[22]以辅助决策,在决策网络参数上体现探索;基于结构噪声的模型无关探索(model-agnostic exploration with structured noise, MAESN)[2]和概率嵌入行动者-评论家的元强化学习(probabilistic embeddings for actor-critic meta-RL, PEARL)[23]引入探索隐变量(latent variable)[2,23-26],通过在多任务上对探索隐变量进行学习,使策略网络能对新任务快速有效探索,这体现了对任务本质特点的"结构化"探索,即通过各任务的上下文数据,后验地学习表征任务特性的结构化信息,以快速有效探索新任务。然而,强化学习本身更强调在线地学习,即智能体在当前任务中进行行动不断"试错"。于是,如何在当前单任务中在线发掘表征任务本质特征的信息,帮助快速智能体有效探索当前环境,且无须利用其他任务数据去学习如何探索,是一个值得研究的课题。

因此,本文聚焦于强化学习中对当前单任务在线探索策略的学习,先验地引入一个可学习的表征任务特点的隐变量,并通过引入一个环境模型,以元学习[23,27-28]的方式在环境模型中更新任务隐变量,从而帮助智能体在真实环境中高效探索。该元学习的探索方法结合了免模型(model-free)[29-36]和基于模型(model-based)[23,37-39]强化学习思想,适用于典型的在策略(on-policy)[40]强化学习算法,旨在当前任务中无须其他任务数据,也无须多余的环境交互步,即可有效提高性能。

1 方法

元学习探索隐变量的强化学习方法主要通过在元层和基层中迭代地实现:为在较高采样利用率情况下进行有效探索,结合免模型和基于模型强化学习思想,引入了一个可学习的环境模型(参数为 ω),通过多次演绎推理去更新优化所引入的探索隐变量 z(参数为 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$),这在元层进行实现;而智能体的价值网络(参数为 θ)或策略网络(参数为 ϕ)仍然通过与环境交互以免模型的方式进行更新,这在基层进行实现,而且这也可以从源头上避免传统基于模型的方法中环境模型的偏差所带来的对智能体性能的直接影响,即基于环境模型所学习的探索隐变量辅助智能体在真实环境中进行有效探索且不损失采样效率,以此提高智能体在当前单个任务中的学习性能。上文提及的所有参数都是逐步迭代优化的:探索隐变量会基于环境模型逐步加深对任务特点的深刻认识;策略网络会在探索隐变量的指导下在真实环境中进行优化,以获得尽可能高的收益;智能体真实行为轨迹会提供给环境模型的学习,使得环境模型更拟合于真实环境特性;更好的环境模型会帮助学习得到该环境下任务更准确的探索隐变量。

该方法适用于对在策略强化学习的性能提升,因为在策略算法根据当前策略的执行轨迹去更新函数逼近器的过程可以为基于隐变量的基层更新过程和利用环境模型演绎评判探索隐变量的元层更新过程的一致性[36]提供保证。采用经典的在策略 Actor-Critic[41] 架构的近端策略优化(proximal policy optimization, PPO)[40,42]算法作为基本算法,并将所提出的元学习探索隐变量方法标为 PPO-latent。

1.1 整体架构

PPO-latent 的整体架构如图 1 所示。图 1 中间为免模型的基本算法网络(策略网络和价值网络),智能体会基于在环境模型(图右侧)中学到的探索隐变量(图右上角)直接与真实环境(图左上角)进行交互,交互的经验(图下侧)会用于更新环境模型。

针对当前单任务在线强化学习,探索隐变量定义为来自可迭代学习的先验高斯分布 $N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ 的采样,其中 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$ 是可学习的参数,将会随着元强化学习过程逐步加深和完善对当前任务特性的表征。可学习的分布是在当前任务的整体学习过程中逐步适应和优化的,即"自举"式地体现了对自我任务的认知,因此无须其他任务数据及其他任务数据所提供的后验隐变量分布。

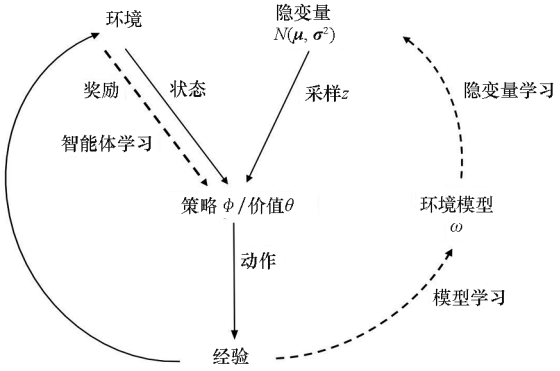


图 1 基于探索隐变量元学习的在线强化学习

Fig. 1 Online reinforcement learning based on exploratory latent variable meta-learning

元优化流程:元优化流程可以描述为一个双层优化[43]问题:元层学习为基于环境模型对探索隐变量的学习,基层学习为基于探索隐变量指导的基本网络(策略网络和价值网络)学习和环境模型的学习。PPO-latent 在当前单个强化学习任务中在线迭代地进行基层和元层的学习。PPO-latent 采用了基于模型的强化学习的特点,引入环境模型进行多次演绎,进而对探索隐变量进行学习。该可学习的环境模型连接了基层和元层的学习过程,该双层优化过程由以下公式进行描述,其中加星号的参数表示该公式中所要进行优化或刚刚优化过的参数:

$$\begin{cases} \boldsymbol{\mu}^* \boldsymbol{\sigma}^* = \arg\min_{\boldsymbol{\mu},\boldsymbol{\sigma}} L^{\text{Latent}}_{\text{Mod}_{\omega^*}}(s, \pi_{\phi^*}(s, z \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2))) \\ \omega^* = \arg\min_{\omega} L^{\text{Mod}}(Mod_\omega(s, \pi_\phi(s,z)), Env(s, \pi_\phi(s,z))) \\ \phi^* = \arg\min_{\phi} J^{\text{PPO}}(s, \pi_\phi(s,z)) \end{cases} \tag{1}$$

其整体工作流程具体解释如下:

1) 基层:基层学习在真实的环境中进行。固定当前探索隐变量分布 $N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ 的值,从中进行采样 z,进而 PPO 策略网络 $\pi_\phi(a|s,z)$ 会输入状态 s 和隐变量分布采样 z,输出动作 a。智能体在真实环境中运行一个轨迹 $\tau = (s_0, a_0, r_0, s_1, \cdots, s_T, a_T, r_T, s_{T+1})$,根据环境反馈的累积奖励,按照 PPO 原有的损失函数形式[40]更新策略网络参数 ϕ 和价值网络参数 θ;并且基于真实环境的状态转移 Env 所获得的若干个 (s_i, a_i, r_i, s_{i+1}) 元组,采用监督学习的方式,以均方误差(mean squared error, MSE)损失为目标函数来更新环境模型参数 ω,即环境模型 $Mod_\omega(s', r|s, a)$ 学习拟合环境状态转移,输入是真实数据当前状态 s 和动作 a,期望环境模型输出的下一步状态 s' 和奖励 r 要尽可能与真实环境状态转移中下一步状态、奖励相同。需要注意的是,PPO 的价值网络与通常情况下的更新方式相同,即期望状态价值函数越准确越好,这也在基层学习中进行更新,此处聚焦于策略网络和环境模型的优化过程,因此对价值网络 θ 的更新过程不再做过多强调。

2) 元层:元层学习利用环境模型来评估当前智能体的探索水平并对探索隐变量进行更新。具体来说,环境模型参数 ω 和策略网络参数 ϕ 暂时固定,通过在环境模型 Mod_ω 中进行多次演绎(rollout)来更新探索隐变量的分布,即更新 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$。每次演绎时都从该隐变量分布中随机采样一个值 $z \in N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$,随后依据此隐变量的值,利用 $\pi_\phi(a|s,z)$ 在环境模型中运行一定的训练步,因此多次演绎便意味着多条满足探索隐变量分布下的探索路径的试探,目的是使得多次演绎的平均累积回报越高越好,根据平均累积回报反向传播来更新 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$。另外,为了降低基于模型方法中可能存在的模型偏差带来的结果偏差,根据基于模型强化学习中不确定性的理论证明[37],智能体仅在环境模型中运行少量的行动步,这也类似于蒙特卡罗树搜索中对枝干长度的控制。而且,为使隐变量分布较为稳定,基于信息瓶颈(information bottleneck)[44-45]理论,本文希望分布不至于偏离标准高斯分布太远,这也是 PPO-latent 的隐变量分布初始值从标准高斯分布开始进行探索优化的理论支撑,其具体设计会在之后进行详细介绍。

1.2　基于探索隐变量的策略执行和学习

智能体通过探索隐变量获得指导,从而在当前任务中实现更高效的探索。为使智能体在决策时利用探索隐变量,在 PPO-latent 中将传统的策略网络设计 $\pi_\phi(a|s)$ 修改为 $\pi_\phi(a|s,z)$,在每一个训练回合中对 z 进行一次随机采样使用,以保证在同一回合中探索方案的一致性和稳定性。z 从 $N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ 分布中随机采样而得,并随后提供给策略网络 π_ϕ 作为其输入的一项,π_ϕ 采用非线性神经网络,可以将探索隐变量中蕴含的信息体现在复杂的动作分布中,以此来作用于智能体在真实环境中动作的执行。类似地,策略网络的输入和表达方式在之前的一些对智能体多任务的迁移学习或快速适应的工作[2,32,35]中也有相关使用,而 PPO-latent 重点关注在单任务中在线元学习探索隐变量以提高当前任务中智能体的学习性能。在环境中每一回合结束后,价值网络参数 θ(见式(2),其中 $\hat{R}_t$ 为 t 时刻之后的累积回报)和策略网络参数 ϕ(见式(3))会按照 PPO 算法中的更新方式[40]进行优化。

$$\theta = \arg\min_{\theta} L_{\text{MSE}}\left(\sum_{t=0}^{T} (V_\theta(s_t) - \hat{R}_t)^2\right) \tag{2}$$

$$\phi = \arg\max_{\phi} J^{\text{PPO}}(s, \pi_\phi(s,z)) \tag{3}$$

1.3　基于环境交互数据的环境模型学习

在真实环境强化学习的每一回合中,智能体依据当前采样的 z 值,利用策略 $\pi_\phi(a|s,z)$ 与真实环境交互,获得轨迹经验数据 $\tau = (s_0, a_0, r_0, s_1, \cdots, s_T, a_T, r_T, s_{T+1})$,也就获得了很多个 (s_i, a_i, r_i, s_{i+1}) 元组,即智能体处于 s_i 状态,执行了动作 a_i,然后观察到环境转移到的下一状态 s_{i+1} 以及收到奖励 r_i,这些信息会用来以监督学习的方式提

高环境模型 Mod_ω 对其状态转移函数 $\mathcal{P}_\omega$（属于密度估计问题）和奖励预测函数 $\mathcal{R}_\omega$（属于回归问题）的估计的准确性。环境模型的优化可描述如式(4)所示，以 MSE 损失的形式来更新其参数。

$$\begin{cases} s_{i+1} \in \mathcal{P}_\omega(s_i, a_i) \\ r_i \in \mathcal{R}_\omega(s_i, a_i) \end{cases} \tag{4}$$

1.4 利用环境模型对探索隐变量的元学习

这里重点描述探索隐变量的更新方式，PPO-latent 使用了基于梯度的元学习和变分推断相结合的方式来对探索隐变量进行优化。在每一轮训练中，基层学习利用隐变量做出行动并更新策略网络，元层学习评估智能体当前的探索性能水平并对探索隐变量进行更新，即元层通过向自己发问："基于当前探索隐变量指导下所更新的策略网络，使得智能体对自身任务的性能提升了吗？"，以期望性能提升为目标来更新隐变量。本文在元层学习中利用所学习的环境模型对其性能进行评估，这可以看作是一个在线的推理网络，无须再与环境进行真实交互，而是通过拟合真实环境模型中的演绎推理不断优化探索隐变量分布。

隐变量分布的设计：就探索隐变量而言，不同于先前的隐变量元学习工作如 PEARL[23] 等，它们的隐变量服从的分布大多是需要基于多任务数据所学习的后验分布，PPO-latent 隐变量分布是先验高斯分布 $N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$，无须其他任务数据进行辅助学习，可以直接在线迭代地进行优化，不断加强对自身任务的探索认知。PPO-latent 利用神经网络反向传播直接优化可学习的参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$，根据信息瓶颈理论，探索隐变量体现了当前任务中的普适规律和特性，其训练前的初始值对齐为标准高斯分布 $\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\sigma} = \mathbf{1}$，这利于规整解耦隐变量分量，方便对任务特征解耦和建模，增强单任务探索泛化性能。

基于隐变量的模型演绎：固定环境模型参数 ω 和策略网络参数 ϕ，更新隐变量分布参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$。类似于基于想象的学习[23]，利用模型的多次演绎可看作在一定探索水平下在模型中展开多条路径的想象，第 i 条演绎路径就是从隐变量分布中随机采样一个 $z^{\text{rollout}_i} \in N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ 值，从初始状态开始运行一个探索轨迹 τ_{rollout_i}，即在环境模型中该条演绎路径的每一步基于当前 s_t^{Mod} 执行 $\pi_\phi(s_t^{\text{Mod}}, z^{\text{rollout}_i})$，获得模型预测的下一步状态 s_{t+1}^{Mod} 和奖励 r_t^{Mod}。为降低模型可能存在的偏差所带来的影响，需要控制每次演绎的步数不至于过大。需要注意的是，为使采样过程仍然可以保持对可学习的分布参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$ 的神经网络计算图（保留计算图即可以保留神经网络对参数梯度的计算和梯度对神经网络参数的更新），PPO-latent 采用重参数[46]技巧进行采样，以保证隐变量分布参数的连续性和可导性，否则普通的随机采样操作是不可导的。因此模型给出的累积预测奖励可以为隐变量分布参数的优化提供指导。这种在模型中演绎多条路径的思想与进化算法[47]中对多个方向的尝试在形式上有相似之处，但是却有本质的不同：进化学习是零阶前馈的，直接尝试众多参数梯度方向并从中选择更新方向；而 PPO-latent 是一阶反馈的，通过多次演绎探索的期望表现来反向传播优化分布参数。

元损失的设计和隐变量分布参数的学习：元损失通过智能体在环境模型中的多次演绎而获得，PPO-latent 利用元损失 $L_{\text{Mod}}^{\text{Latent}}$ 来更新隐变量分布参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$。元损失的定义见式(5)：

$$L_{\text{Mod}}^{\text{Latent}} = E_{\text{rollout}}\left(-\sum_{t=1}^{T} r_t^{\text{Mod}}(s_t, \pi_\phi(s_t, z^{\text{rollout}} \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)))\right) + KL(N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) \| N(\mathbf{0}, \mathbf{1})) \tag{5}$$

其中，$-\sum_{t=1}^{T} r_t^{\text{Mod}}(s_t, \pi_\phi(s_t, z^{\text{rollout}} \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)))$ 体现了在环境模型中利用当前策略网络多次演绎的结果来更新隐变量分布的过程，智能体在环境模型中运行的每一步所获得的即时奖励 r_t^{Mod} 都是 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$ 的函数，因此 PPO-latent 会利用多次演绎的期望累积奖励基于梯度优化来训练隐变量分布，目的是希望期望累积奖励越大越好。另外，基于变分推断[44]原则，PPO-latent 也在元损失中添加了训练的隐变量分布同标准高斯分布之间的 KL（Kullback-Leibler，KL）散度。类似于 PEARL 中对隐变量分布的分析，KL 形成的损失可以理解为信息瓶颈下的变分估计，以试图用相对更加简单的分布和更低的成本来完成任务，这意味着得到泛化性能更好的模型。高斯分布下的 KL 散度具有解析解，在变分估计中，高斯分布常常是一种简单合理的先验分布被加以使用。综上，对模型进行多次演绎后，基于元损失 $L_{\text{Mod}}^{\text{Latent}}$ 来更新 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$ 以对隐变量分布进行学习。

这样的一种元损失的设计从理论上体现了变分证据因子下界（evidence lower bound，ELBO）[2]，即：

$$\ln p(x) \geqslant E_q[\ln p(x|z)] - KL(q(z) \| p(z)) \tag{6}$$

对于 PPO-latent，式(6)中的 $p(z)$ 为标准高斯分布 $N(\mathbf{0}, \mathbf{1})$，$q(z)$ 为要学习的隐变量分布 $N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$，后验概率 $\ln p(x|z)$ 为在当前隐变量 z 下

运行策略得到的累积奖励。因此最小化元损失 $L_{\text{Mod}}^{\text{Latent}}$ 即为最大化变分证据因子下界 $E_q[\ln p(x|z)] - KL(q(z)\|p(z))$,即以智能体可以获得的回报的最小值越大越好为目标来更新隐变量分布。

综上所述,PPO-latent 元学习探索隐变量的高效强化学习在算法 1 中进行了总结。

算法 1　基于探索隐变量元学习的高效强化学习

Alg. 1　Reinforcement learning using exploring latent variable based on meta-learning

1. 初始化:环境模型 Mod_ω,探索隐变量分布参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$,策略网络 π_ϕ,价值网络 V_θ
2. **for** 每次迭代 **do**
3.　元训练
4.　采样 $z \in N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$
5.　基于 z 和当前状态 s_0 运行策略 π_ϕ 得到轨迹 $\tau = (s_0, a_0, r_0, s_1, \cdots, s_T, a_T, r_T, s_{T+1})$
6.　基于轨迹 τ 使用 PPO 算法更新策略网络参数 ϕ 和价值网络参数 θ(见式(2)~(3))
7.　基于轨迹 τ 更新环境模型参数 ω(见式(4))
8.　元测试
9.　**for** 在环境模型 Mod_ω 中每次演绎 i **do**
10.　采样 $z^{\text{rollout}_i} \in N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$
11.　初始化演绎 i 累积奖励 $R_{\text{rollout}_i} = 0$
12.　**for** 模型中的每个行动步 t **do**
13.　执行动作 $\pi_\phi(s_t^{\text{Mod}}, z^{\text{rollout}_i})$
14.　获得下一步状态 s_{t+1} 和奖励 r_t
15.　计算累积奖励 $R_{\text{rollout}_i} += r_t$
16.　**end for**
17.　**end for**
18.　计算隐变量分布损失函数 $L_{\text{Mod}}^{\text{Latent}} = -E(R_{\text{rollout}}) + KL$(见式(5))
19.　基于 $L_{\text{Mod}}^{\text{Latent}}$ 更新隐变量分布参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$
20. **end for**

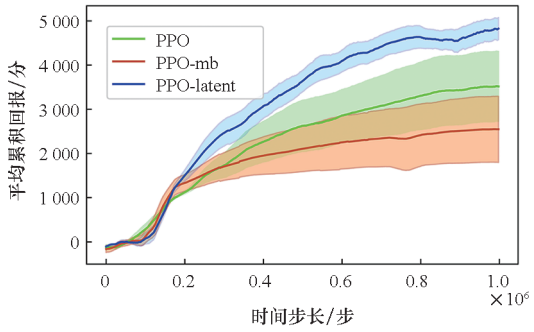
2　实现与评估

2.1　算法参数设置

除 PPO-latent 与 PPO 基本算法对比之外,还测试了基于模型进行策略学习的 PPO 方法(PPO-mb),即基于 Dyna 框架[48]的 PPO 方法进行对照。其他现有的基于隐变量的探索工作均需要额外的多任务数据进行训练,无法直接应用于单强化学习任务中。实验在 OpenAI Gym[49]中的 MuJoCo(v2 版本)连续控制任务上进行了评估。学习率设置为:$\phi_{\text{lr}} = 0.000\ 3$,$\theta_{\text{lr}} = 0.000\ 3$,$\omega_{\text{lr}} = 0.001$,$l_{\text{lr}} = 0.001$($l_{\text{lr}}$为隐变量的学习率);隐变量中 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$ 分别设置为一个 5 维的向量,隐变量分布的初始化设为 $\boldsymbol{\mu} = \mathbf{0}$,$\boldsymbol{\sigma} = \mathbf{1}$。网络中的优化器均使用 Adam。在环境模型中每次演绎的步数设为 30 步。

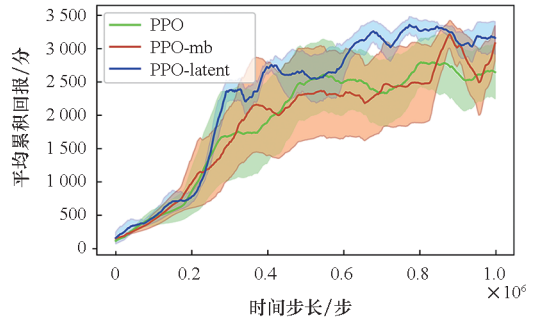
2.2　实验结果评估

图 2 展示了各方法下智能体的性能曲线,实验结果展示的是在 5 个随机种子和网络初始化情况下进行实验的结果平均值,这 5 次的标准差显示为时间步长上的阴影区域。遵照 PPO 原始实验中的设置,智能体在各任务中与环境交互的步数为 10^6 步,且每隔 2 048 步会对智能体当前的性能进行 10 次评估并取均值。在评估时仅进行性能结果的统计验证,该步数时不再进行训练,即避免评估时的数据也是训练数据所导致的过拟合问题。对于 PPO-latent,每次评估时都会从当前隐变量分布中采样出一个新的探索隐变量。为清晰起见,对曲线进行了统一平滑处理(滑动窗口大小为 10)。从图 2 可以看出,元学习探索隐变量的方法(PPO-latent)在学习速度和渐进性能方面通常优于基准方法和一般直接基于模型的方法(PPO-mb),而且,PPO-mb 并没有表现出比 PPO 更优的性能,甚至可能出现性能下降;此外,PPO-latent 通常具有较小的方差。表 1 和图 3 显示了



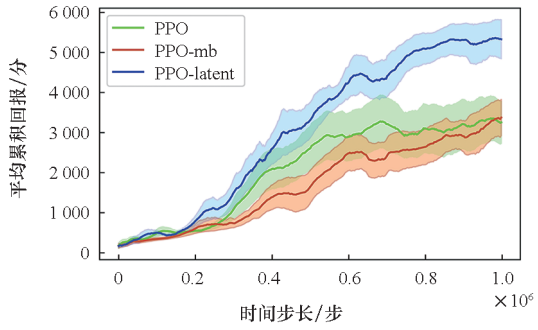
(a) HalfCheetah-v2 任务学习曲线

(a) Learning curve on HalfCheetah-v2 task



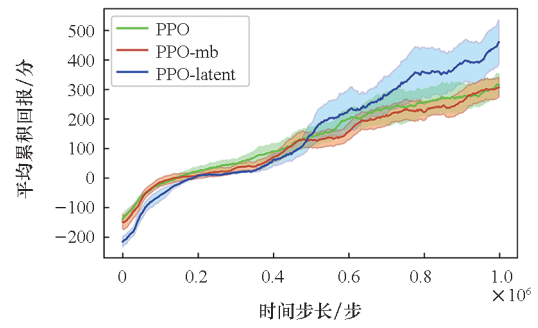
(b) Hopper-v2 任务学习曲线

(b) Learning curve on Hopper-v2 task



（c）Walker2d-v2 任务学习曲线

（c）Learning curve on Walker2d-v2 task



（d）Ant-v2 任务学习曲线

（d）Learning curve on Ant-v2 task

图 2　在连续控制任务上 PPO-latent 方法及对比方法的学习曲线

Fig. 2　The learning curve of PPO-latent method and the comparison methods in the continuous control task

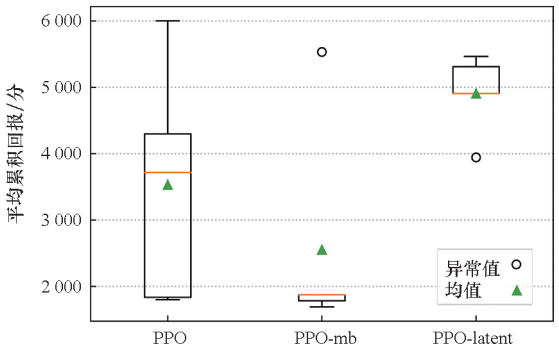
所有对比方法在任务上最大平均回报的汇总结果，其中表 1 报告的结果为在 5 个随机种子上获得的平均回报的最大值，在每个任务上的所有方法中获得的最大值已加粗表示，图 3 显示了在所有时间步上 5 次试验的最大平均回报的盒形图（若盒形图中未显示异常值图标，则说明数据中不存在超出合理范围的异常值），可以看出 PPO-latent 提供了一致的更高的最大回报。

表 1　PPO-latent 方法及对比方法的最高得分

Tab. 1　Maximal score of PPO-latent and comparisons

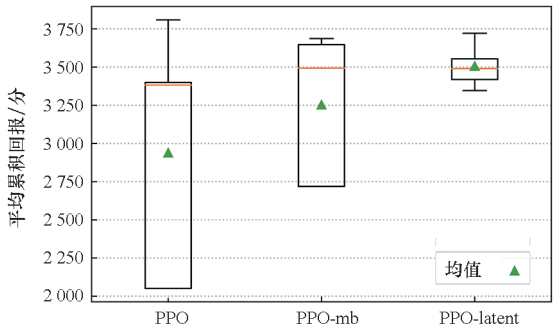
单位：分

方法	HalfCheetah-v2	Hopper-v2	Walker2d-v2	Ant-v2
PPO	3 576.43 ±1 036.07	3 040.42 ±894.00	3 669.94 ±1 023.68	359.35 ±93.74
PPO-mb	2 573.93 ±1 021.31	3 525.98 ±620.54	3 691.32 ±936.96	369.67 ±96.45
PPO-latent	5 006.78 ±350.98	3 520.85 ±311.73	5 542.52 ±744.35	515.37 ±101.32



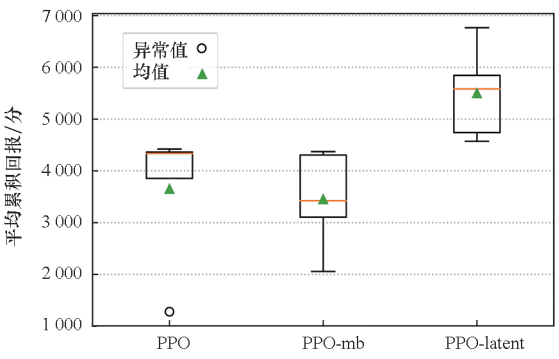
（a）HalfCheetah-v2 任务盒形图

（a）Box plot of HalfCheetah-v2 task



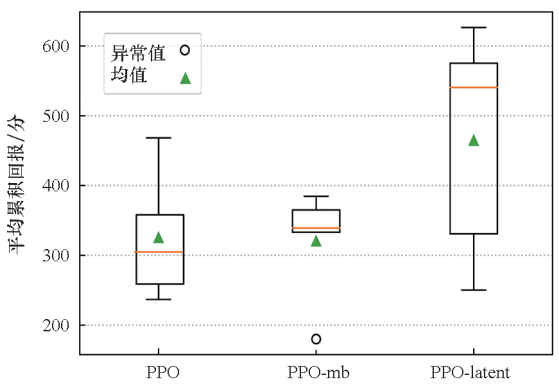
（b）Hopper-v2 任务盒形图

（b）Box plot of Hopper-v2 task



（c）Walker2d-v2 任务盒形图

（c）Box plot of Walker2d-v2 task



（d）Ant-v2 任务盒形图

（d）Box plot of Ant-v2 task

图 3　在所有时间步上 5 次试验的最大平均回报的盒形图

Fig. 3　Box plot of maximum average return for 5 trials through all timesteps

2.3　消融实验结果

2.3.1　KL 损失函数的消融实验

该消融实验分析 KL 损失函数的作用,消融实验中元损失的定义去除 KL 散度一项,如式(7)所示:

$$L_{\text{Mod}}^{\text{Latent}} = E_{\text{rollout}}\left(-\sum_{t=1}^{T} r_t^{\text{Mod}}\left(s_t, \pi_\phi\left(s_t, z^{\text{rollout}} \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)\right)\right)\right) \tag{7}$$

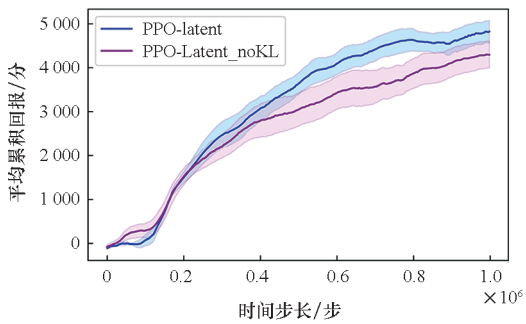
对比实验以 HalfCheetah-v2 环境为例,图 4(a)展示了不同元损失设计情况下的智能体评估性能,式(5)的默认设计获得了更高的渐进性能,展现了 KL 散度的设计在当前任务的学习阶段更具性能泛化和提升的潜力。

2.3.2　针对计算开销的消融实验

智能体在环境中进行强化学习时,其最昂贵的代价在于与环境进行交互。PPO-latent 为不引入额外的与环境交互代价,而是引入一个可学习的环境模型进行多次演绎从而学习训练智能体的探索能力。其模型和探索隐变量的学习引入了部分计算开销,本消融实验在控制 PPO 和 PPO-latent 方法与环境交互步数相同且不变的情况下,增加了 PPO 中重要性采样和梯度更新网络的步数,使得该基准方法(PPO_moreEpoch)的计算开销与 PPO-latent 方法等同。该消融实验以 HalfCheetah-v2 环境为例,图 4(b)学习曲线显示了 PPO-latent 的良好性能并不能简单地通过相应增加基准方法的重要性采样梯度更新步数来实现。

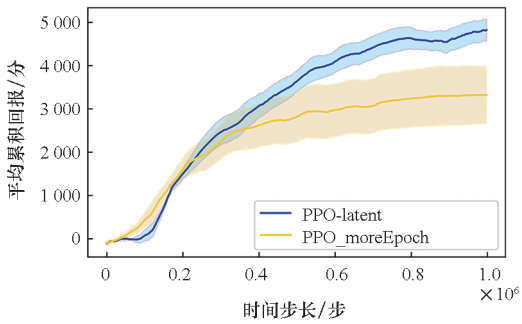
2.3.3　环境模型演绎的消融实验

对于基于模型的强化学习,模型的不确定性所带来的偏差可能会在长远多步演绎中对智能体性能带来影响。本文控制演绎长度分别为 5 步、30 步、50 步、100 步,以 HalfCheetah-v2 和 Ant-v2 任务为例,统计智能体在 5 个随机种子上的实验渐进性能,如表 2 所示。当演绎长度过长时,其模型不确定性带来的影响较为明显,从而导致智能体性能受损;演绎长度在 5 步和 30 步时具有相当的较好渐进性能,但 5 步时表现出了更大的方差,说明较小的演绎长度使得智能体还不具有长远眼光,从而探索能力不够深入和精准。因此,综合考虑模型不确定性影响和探索的长远性,PPO-latent 在环境模型中采用演绎长度为 30 步。



(a) KL 损失函数的消融实验结果

(a) Ablation results of KL loss function



(b) 针对计算开销的控制实验结果

(b) Results of a control experiment for computational overhead

图 4　消融实验结果

Fig. 4 Ablation experimental results

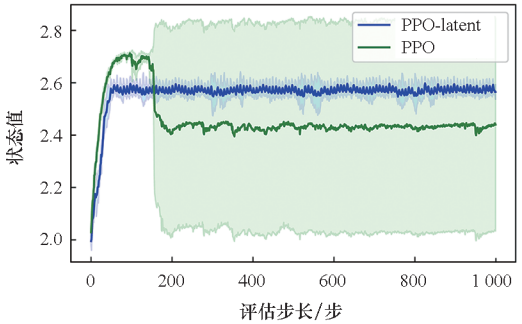
表 2　不同演绎长度 PPO-latent 平均渐进得分

Tab. 2　Average progressive scores of PPO-latent under different rollout length

演绎长度/步	HalfCheetah-v2/分	Ant-v2/分
5	4 683.66 ± 683.25	**488.04 ± 142.93**
30	**4 779.78 ± 317.51**	475.08 ± 98.47
50	3 691.14 ± 376.49	336.45 ± 83.52
100	3 008.90 ± 401.38	280.82 ± 107.63

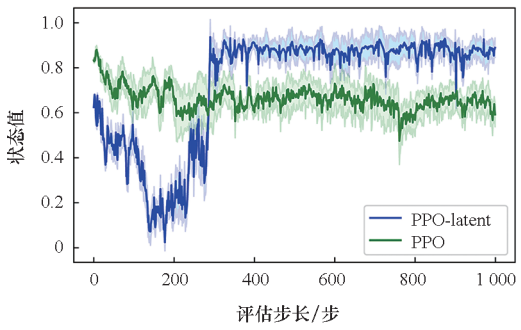
2.3.4　模型探索策略的消融实验

为更好地体现经训练的模型在环境中的探索策略优势,图 5 以 HalfCheetah-v2 和 Ant-v2 环境为例,展示了智能体在环境中运行过程所到之处的状态价值情况。从结果看出,相比于基本的



(a) HalfCheetah-v2 任务价值状态曲线

(a) State value curve of HalfCheetah-v2 task



(b) Ant-v2 任务价值状态曲线

(b) State value curve of Ant-v2 task

图 5 模型应用于在环境中运行时的状态价值曲线

Fig. 5 State value curve of model as it runs in environment

PPO 方法,PPO-latent 在智能体运行初期可能并不会直接选择 Value 值更大的状态,而是着眼于长远的总收益,去探索更多可能性。从长远来看,PPO-latent 也的确比 PPO 更有可能到达 Value 值更高的状态,且 PPO-latent 的表现更加稳定,可以维持在一个较高的状态值水平。因此,可学习的探索策略会为智能体获得更高的总收益提供有效的支持。

3 结论

围绕在线提升智能体在当前单强化学习任务中的探索能力这一挑战,提出了基于元学习探索隐变量的高效强化学习方法。创新性引入一个表征当前任务特点的可学习的探索隐变量,并通过引入一个环境模型,以元学习的方式在环境模型中更新该任务隐变量,使得策略网络在真实环境中可以借助该具有任务探索信息的隐变量,帮助在真实环境中高效探索。探索隐变量、环境模型、策略和价值网络均在当前强化学习任务中迭代进行更新,无须其他任务数据;而且,探索任务本质特点的隐变量利用环境模型更新,帮助策略网络提前"探路",也不需要花费额外的真实环境探索步。实验证明,元学习探索隐变量的强化学习方法可以对智能体单任务学习提供明显性能收益。

参考文献(References)

[1] GARAFFA L C, BASSO M, KONZEN A A, et al. Reinforcement learning for mobile robotics exploration: a survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(8): 3796 – 3810.

[2] GUPTA A, MENDONCA R, LIU Y X, et al. Meta-reinforcement learning of structured exploration strategies[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018.

[3] WATKINS C J C H, DAYAN P. *Q*-learning[J]. Machine Learning, 1992, 8: 279 – 292.

[4] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[C]//Proceedings of NIPS Deep Learning Workshop, 2013.

[5] SZITA I, LRINCZ A. Optimistic initialization and greediness lead to polynomial time learning in factored MDPs[C]// Proceedings of the 26th Annual International Conference on Machine Learning, 2009: 1001 – 1008.

[6] BRAFMAN R I, TENNENHOLTZ M. R-max—a general polynomial time algorithm for near-optimal reinforcement learning[J]. Journal of Machine Learning Research, 2002, 3: 213 – 231.

[7] AN G, MOON S, KIM J H, et al. Uncertainty-based offline reinforcement learning with diversified *Q*-ensemble[C]// Proceedings of the 35th International Conference on Neural Information Processing Systems, 2021.

[8] GEIST M, PIETQUIN O. Managing uncertainty within value function approximation in reinforcement learning[C]// Proceedings of Workshop on Active Learning and Experimental Design, Collocated with AISTATS 2010, 2010.

[9] KVETON B, KONOBEEV M, ZAHEER M, et al. Meta-thompson sampling[C]//Proceedings of the 38th International Conference on Machine Learning, 2021.

[10] MARCUS R, NEGI P, MAO H Z, et al. Bao: making learned query optimization practical[C]//Proceedings of the International Conference on Management of Data, 2021: 1275 – 1288.

[11] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//Proceedings of the 35th International Conference on Machine Learning, 2021.

[12] HAARNOJA T, ZHOU A, HARTIKAINEN K, et al. Soft actor-critic algorithms and applications[EB/OL]. (2019 – 01 – 29)[2023 – 01 – 14]. https://arxiv.org/abs/1812.05905.

[13] AKAM T, WALTON M E. What is dopamine doing in model-based reinforcement learning?[J]. Current Opinion in Behavioral Sciences, 2021, 38: 74 – 82.

[14] MOERLAND T M, BROEKENS J, PLAAT A, et al. Model-based reinforcement learning: a survey[J]. Foundations and Trends® in Machine Learning, 2023, 16(1): 1 – 118.

[15] FENG F, WANG R S, YIN W, et al. Provably efficient exploration for reinforcement learning using unsupervised learning[C]//Proceedings of the 34th Conference on Neural Information Processing Systems, 2020.

[16] BERRY D A, FRISTEDT B. Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability)[M]. [S.l.]: Springer, 1985.

[17] KAELBLING L P, LITTMAN M L, MOORE A W. Reinforcement learning: a survey[J]. Journal of Artificial Intelligence Research, 1996, 4: 237 – 285.

[18] FORTUNATO M, AZAR M G, PIOT B, et al. Noisy networks for exploration[C]//Proceedings of the International Conference on Learning Representations, 2018.

[19] PLAPPERT M, HOUTHOOFT R, DHARIWAL P, et al. Parameter space noise for exploration[C]//Proceedings of the International Conference on Learning Representations, 2018.

[20] PATHAK D, AGRAWAL P, EFROS A A, et al. Curiosity-driven exploration by self-supervised prediction[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017: 488 –

489.

[21] YUAN M Q, LI B, JIN X, et al. Automatic intrinsic reward shaping for exploration in deep reinforcement learning[C]// Proceedings of the International Conference on Machine Learning, 2023.

[22] THABET M. Imagination-augmented deep reinforcement learning for robotic applications [D]. Manchester: the University of Manchester (United Kingdom), 2022.

[23] RAKELLY K, ZHOU A, QUILLEN D, et al. Efficient off-policy meta-reinforcement learning via probabilistic context variables [C]//Proceedings of the 36th International Conference on Machine Learning, 2019.

[24] RUSU A A, RAO D, SYGNOWSKI J, et al. Meta-learning with latent embedding optimization [C]//Proceedings of International Conference on Learning Representations, 2019.

[25] SÆMUNDSSON S, HOFMANN K, DEISENROTH M P. Meta reinforcement learning with latent variable Gaussian processes[C]//Proceedings of Conference on Uncertainty in Artificial Intelligence, 2018.

[26] HAUSMAN K, SPRINGENBERG J T, WANG Z Y, et al. Learning an embedding space for transferable robot skills[C]// Proceedings of the International Conference on Learning Representations, 2018.

[27] PAL M, KUMAR M, PERI R, et al. Meta-learning with latent space clustering in generative adversarial network for speaker diarization [J]. IEEE/ACM Trans Audio Speech Lang Process, 2021, 29: 1204 – 1219.

[28] 李凡长, 刘洋, 吴鹏翔, 等. 元学习研究综述[J]. 计算机学报, 2021, 44(2): 422 – 446.
LI F Z, LIU Y, WU P X, et al. A survey on recent advances in meta-learning[J]. Chinese Journal of Computers, 2021, 44(2): 422 – 446. (in Chinese)

[29] FINN C, ABBEEL P, LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks [C]// Proceedings of the International Conference on Machine Learning, 2017.

[30] LIU Y S, HALEV A, LIU X. Policy learning with constraints in model-free reinforcement learning: a survey [C]// Proceedings of the 30th International Joint Conference on Artificial Intelligence, 2021: 4508 – 4515.

[31] PAN E, PETSAGKOURAKIS P, MOWBRAY M, et al. Constrained model-free reinforcement learning for process optimization[J]. Computers & Chemical Engineering, 2021, 154: 107462.

[32] RAMÍREZ J, YU W, PERRUSQUÍA A. Model-free reinforcement learning from expert demonstrations: a survey[J]. Artificial Intelligence Review, 2022, 55: 3213 – 3241.

[33] FEINBERG V, WAN A, STOICA I, et al. Model-based value estimation for efficient model-free reinforcement learning[C]// Proceedings of the 35th International Conference on Machine Learning, 2018.

[34] PETERS J, SCHAAL S. Policy gradient methods for robotics[C]// Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.

[35] ZHOU W, LI Y Y, YANG Y X, et al. Online meta-critic learning for off-policy actor-critic methods[C]//Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020.

[36] MAO W C, ZHANG K, ZHU R H, et al. Near-optimal model-free reinforcement learning in non-stationary episodic MDPs[C]//Proceedings of the 38th International Conference on Machine Learning, 2021.

[37] JANNER M, FU J, ZHANG M, et al. When to trust your model: model-based policy optimization[C]//Proceedings of the 33th International Conference on Neural Information Processing Systems, 2019.

[38] SHLEZINGER N, WHANG J, ELDAR Y C, et al. Model-based deep learning[J]. Proceedings of the IEEE, 2023, 111(5): 465 – 499.

[39] 黄文振, 尹奇跃, 张俊格, 等. 基于模型的强化学习中可学习的样本加权机制[J]. 软件学报, 2022, 34(6): 2765 – 2775.
HUANG W Z, YIN Q Y, ZHANG J G, et al. Learnable weighting mechanism in model-based reinforcement learning[J]. Journal of Software, 2022, 34(6): 2765 – 2775. (in Chinese)

[40] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[EB/OL]. (2017 – 08 – 28)[2023 – 02 – 02]. https://arxiv.org/abs/1707.06347.

[41] KONDA V R, TSITSIKLIS J N. Actor-critic algorithms[C]// Proceedings of the 13th International Conference on Neural Information Processing Systems, 1999.

[42] GitHub. Pytorch-a2c-ppo-acktr-gail [EB/OL]. [2025 – 08 – 15]. https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail/tree/master/a2c_ppo_acktr.

[43] FRANCESCHI L, FRASCONI P, SALZO S, et al. Bilevel programming for hyperparameter optimization and meta-learning [C]//Proceedings of the 35th International Conference on Machine Learning, 2018.

[44] LEE J, CHOI J, MOK J, et al. Reducing information bottleneck for weakly supervised semantic segmentation[C]// Proceedings of the 35th Conference on Neural Information Processing Systems, 2021.

[45] TISHBY N, ZASLAVSKY N. Deep learning and the information bottleneck principle [C]//Proceedings of the IEEE Information Theory Workshop (ITW), 2015.

[46] KINGMA D P, WELLING M. Auto-encoding variational Bayes[C]//Proceedings of the 2nd International Conference on Learning Representations, 2014.

[47] HOUTHOOFT R, CHEN R Y, ISOLA P, et al. Evolved policy gradients[C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018.

[48] SUTTON R S. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming[C]// Proceedings of the 7th International Machine Learning Conference, 1990.

[49] BROCKMAN G, CHEUNG V, PETTERSSON L, et al. OpenAI gym[EB/OL]. (2016 – 06 – 05)[2023 – 04 – 12]. https://arxiv.org/abs/1606.01540.