计算机科学与探索
Journal of Frontiers of Computer Science and Technology
1673-9418/2023/17(08)-1917-11
doi: 10.3778/j.issn.1673-9418.2203074

结合对比预测的离线元强化学习方法

韩　旭,吴　锋+
中国科学技术大学 计算机科学与技术学院,合肥 230032
+ 通信作者 E-mail: wufeng02@ustc.edu.cn

摘　要:传统的强化学习算法与环境进行大量交互才能获取稳定的动作选择策略,且无法有效应对任务环境发生改变的情况,因此难以应用于实际问题。离线元强化学习通过使用包含多个任务的经验回放数据集进行离线策略学习,为复杂情况下智能体快速适应新任务提供了一种有效方法。将离线元强化学习算法应用于复杂任务将会面临两个挑战:首先,由于无法与环境进行充分交互,离线强化学习算法会错误估计数据集外动作的价值,进而选择次优动作;其次,元强化学习算法不仅需要学习动作选择策略,还需要具备稳健而高效的任务推理能力。针对上述挑战,提出一种结合对比预测的离线元强化学习算法。为了应对价值函数的错误估计问题,该算法使用行为克隆技术鼓励策略选择包含在数据集中的动作。为了提高元学习的任务推理能力,该算法使用循环神经网络对智能体上下文轨迹进行任务推理,并利用对比学习和预测网络来分析辨别不同任务轨迹中的潜在结构。实验结果表明,相比现有方法,使用该算法训练得到的智能体在面对未见过的任务时的得分提高了25个百分点以上,并且具有更高的元训练效率和更好的泛化性能。

关键词:深度强化学习;离线元强化学习;对比学习
文献标志码:A　　**中图分类号:**TP181

Offline Meta-Reinforcement Learning with Contrastive Prediction

HAN Xu, WU Feng+
School of Computer Science and Technology, University of Science and Technology of China, Hefei 230032, China

Abstract: Traditional reinforcement learning algorithms require lots of online interaction with the environment for training and cannot effectively adapt to changes in the task environment, making them difficult to apply to real-world problems. Offline meta-reinforcement learning provides an effective way to quickly adapt to a new task by using replay datasets of multiple tasks for offline policy learning. Applying offline meta-reinforcement learning to complex tasks will face two challenges. Firstly, reinforcement learning algorithms overestimate the value of state-action pairs not contained in the dataset and thus select non-optimal actions, resulting in poor performance. Secondly, meta-reinforcement learning algorithms need not only to learn the policy but also to have robust and efficient task inference capabilities. To address the above problems, this paper proposes an offline meta-reinforcement learning algorithm based on contrastive prediction. To cope with the problem of overestimation of value functions, the proposed algorithm uses behavior cloning to encourage policy to prefer actions included in the dataset. To improve the task inference capability of meta-learning, the proposed algorithm uses recurrent neural networks for task inference on the contextual trajectories of the agents and uses contrastive learning and prediction networks to analyze and distinguish potential structures in different task trajectories. Experimental results show that the agents

基金项目:国家自然科学基金(92048301)。
This work was supported by the National Natural Science Foundation of China (92048301).
收稿日期:2022-03-18　　**修回日期:**2022-06-24

trained by the proposed algorithm can score more than 25 percentage points when faced with unseen tasks, and it has higher meta-training efficiency and better generalization performance compared with existing methods.
Key words: deep reinforcement learning; offline meta-reinforcement learning; contrastive learning

强化学习(reinforcement learning, RL)已成功应用于一系列复杂的决策控制任务[1-3],但是智能体通常需要与环境进行大量在线交互以学习如何选取最佳行动。对于真实世界中的复杂任务场景,与环境进行大量交互通常非常昂贵、耗时且有可能是危险的(例如,在机器人、自动驾驶或医疗保健等场景中)[4],这限制了强化学习在许多实际场景中的应用。上述问题的一个解决思路是借鉴监督学习,使用静态数据集学习智能体的策略。这种方法被称为离线强化学习(OfflineRL)[5-6]或批量强化学习(BatchRL)[7],近年来得到了国内外学者的广泛关注和研究。

离线强化学习打破了强化学习中智能体通过与环境在线交互进行学习的基本假设,允许智能体通过预先收集的数据集进行策略学习。缺乏环境交互导致智能体无法纠正数据集之外的状态-动作对的错误价值估计。而强化学习中常用的“自举”操作(Bootstrapping)会进一步放大这种误差,从而导致训练得到的智能体在与环境交互过程中将倾向于选择价值被高估的分布外动作[5]。为了解决这个问题:一类方法通过对数据集内的动作分布进行建模并对策略选择进行约束,从而限制智能体选择数据集中所包含的动作或与之接近的动作[5,8];另一类方法通过惩罚分布外动作的价值函数来避免乐观估计以达到同样的效果[6]。

无需额外训练即可泛化到未见过的新任务上是强化学习的理想特性之一,在现实应用中也有大量需求。由于无法有效利用其他任务中的经验样本数据或学习到的策略模型,传统的强化学习方法在遇到新任务时通常需要从零开始学习一个新的策略[3]。而现实应用中的许多任务往往具有相似的内部结构,元强化学习(Meta-RL)[9-11]可以通过从不同任务中进行归纳偏差以学习这种“结构”,无需交互或仅需少量交互即可适应新的任务。其中,基于梯度下降的元强化学习方法[9]尝试通过学习通用模型初始化参数,并在适应新任务的过程中更新参数的方法达到快速适应新任务的目的。但是这种方法在进行梯度下降时需要利用收集大量的同策略(On-policy)样本,导致其样本利用效率较低。Rakelly等人[12]提出了基于上下文的离策略(Off-policy)元强化学习算法PEARL(probabilistic embeddings for actor-critic meta-RL)以解决这一问题。该算法将元强化学习中的任务推断和策略控制进行解耦,并在两个阶段中使用不同的数据:从近期收集的同策略经验样本中随机采样得到上下文数据以进行任务推断,同时使用先前收集的离策略经验样本进行策略学习。

离线元强化学习结合了离线强化学习和元强化学习的优势,可以从包含多个任务的经验样本数据集中进行元学习,并快速适应到新的任务中。Li等人[13]将离线强化学习算法BCQ(batch-constrained deep Q-learning)[5]与上下文任务推断编码器相结合,提出了用于离线元强化的MBML(multi-task batch reinforcement learning with metric learning)算法,并使用度量学习和奖励重标签技术增加了任务推断的准确性。然而这种方法需要对不同任务的奖励函数分别进行建模,训练效率较低。Li等人[14]以PEARL算法为基础构建了适用于离线环境的FOCAL算法,其中使用行为正则化方法[8]来应对离线强化学习中的分布偏移问题,同时使用负幂度量学习(negative-power variant metric learning)来增强不同任务之间的可区分度。这两种算法均选择状态转换元组作为基础单位进行上下文任务推断,这种方法会带来较大的任务推断误差;并且在处理离线数据时都需要引入额外的网络结构或在实现层面进行修改。Mitchell等人[15]提出的MACAW(meta-actor critic with advantage weighting)算法将优势加权回归(advantage-weighted regression, AWR)算法[16]和MAML(model-agnostic meta-learning)算法相结合,首次将基于优化的方法应用在离线元强化学习场景中。上述方法都需要额外的样本数据才能适应新任务,这将减小离线元强化学习的应用范围。

针对上述问题,本文提出了结合对比预测的离线元强化学习方法,该方法使用包含多个任务的离线数据集进行策略学习,并无需额外数据即可在新任务上获得较好的表现。针对离线环境下的策略学习问题,所提算法通过在策略损失函数中引入行为克隆项来限制动作的选择范围,且无需增加额外网络结构和复杂的算法改动。为了实现高效的任务推理,该算法使用循环神经网络从上下文序列中提取

上下文变量，利用对比学习[17-20]来辨别不同任务对应轨迹中的潜在结构，同时使用了预测网络来增加上下文变量中的任务相关信息。本文主要贡献可以总结如下：

（1）通过在标准策略更新步骤中添加行为克隆正则化项引导策略选择数据集中包含的动作，避免了智能体选择价值函数被错误估计的动作。

（2）使用经验样本序列作为上下文数据，并使用对比学习和预测网络对上下文编码器进行训练，增强了智能体的任务推断能力。

（3）在不同任务环境下测试了所提算法，证明了所提算法能够使智能体在面对未见过的任务时做到快速使用，最终性能较已有算法有25个百分点以上的提升，且具有更高的元训练效率和泛化性能。

1　背景知识

本章主要介绍了结合对比预测的离线元强化学习算法的背景知识和基础概念。表1列出了本文所使用的重要符号和对应说明。

1.1　强化学习

强化学习常被建模为马尔可夫决策过程（Markov decision process，MDP），由五元组 $M\equiv(S,A,P,R,\gamma)$ 表示。其中 S 和 A 分别表示状态和动作空间，P 表示状态转移概率函数，$P(s'|s,a)$ 表示在状态 s 下执行动作 a 得到的下一时刻的状态 s' 的概率值。R 表示奖励函数，$R(s,a)$ 表示智能体在状态 s 执行动作 a 之后获得的奖励大小。$\gamma\in[0,1]$ 表示奖励折扣因子，用于权衡智能体的长期收益和短期收益。策略 $\pi(a|s)$ 表示在状态 s 选择动作 a 的概率，值函数 $Q(s,a)$ 表示状态动作对的预期回报。在强化学习中，智能体的最终目的是通过与环境的大量交互来寻找最优策略 $\pi_*(a|s)$ 以最大化折扣累计回报 $G=\sum_{i=t}^{T}\gamma^{i-t}r_i$。

1.2　离线元强化学习

与专注于解决单一任务的传统强化学习算法不同，元强化学习的目标是让智能体获得快速适应新任务的能力。具体来说，智能体需要通过在一组任务中进行训练以学习如何完成不同任务；智能体在一组新任务上的表现将被作为元强化学习算法泛化能力的评价指标。本文考虑这样一种元强化学习任务场景：所有任务对应的状态空间和动作空间相同，而不同的任务由不同的状态转移概率函数和奖励函数所定义。对一组任务环境，将任务分布记为 $p(T)$，从中采样得到的每个任务 $T_i\sim p(T)$ 都可以表示为一个马尔可夫决策过程 $T_i\equiv(S,A,P_i,R_i,\gamma)$。由于不同任务之间共享状态和动作空间，任务分布可以表示为奖励函数和环境动态的联合分布 $p(T)=p(R)p(P)$。

在离线元强化学习场景中，智能体仅能使用包含多个任务的经验样本数据集 D 来进行任务推断和策略学习。对于每一个任务数据集 $D_i\in D$，其对应的

表1　符号注释表

Table 1　Symbol annotation

符号	符号注释	符号	符号注释
M	马尔可夫决策过程	S	状态空间
A	动作空间	P	状态转移概率函数
R	奖励函数	γ	奖励折扣因子
s	$s\in S$，表示状态空间中的一个状态	a	$a\in A$，表示动作空间中的一个动作
r	某时刻的奖励大小	s'	表示 s 对应的下一个状态
π	智能体所持有的策略	$Q(s,a)$	动作值函数，表示状态动作对的预期回报
π_*	最优策略	$\mathbb{E}$	求期望函数
G	折扣累计回报	T	从任务分布中采样得到的某个任务
$p(T)$	任务分布	p_η	网络参数为 η 的预测网络
q_φ	网络参数为 φ 的上下文编码器	c	上下文数据
z	上下文变量，也称任务变量	ϵ	动作噪声
$\mathcal{N}(0,\sigma^2)$	均值为0、方差为 σ^2 的正态分布	τ	软更新中行为网络参数所占权重
q	查询数据，表示需要进行表征学习的数据	k	键数据，表示与查询数据做对比的数据
c^q	使用上下文数据作为查询数据	c^k	使用上下文数据作为键数据
z^q	用上下文编码器 q_φ 处理 c^q 得到的上下文查询变量	z^k	用上下文编码器 q_φ 处理 c^k 得到的上下文键变量

任务 T_i 都是从任务分布 $P(T)$ 中采样得到的。值得注意的是,智能体并不知道每个任务对应的具体的状态转移概率函数与奖励函数,而是需要通过使用任务推断网络进行有效任务推理继而完成策略学习过程。在测试阶段,智能体需要快速适应从任务分布 $P(T)$ 中采样得到的新任务。

1.3 对比学习

对比学习[17-20]被广泛应用于学习图像或序列数据的特征表示中,其核心思想是提高相同类别样本的特征表示相似度,同时降低不同样本的特征表示相似度。以序列数据为例,正键数据和查询数据通常是从同一类型的序列数据中截取得到的,而负键数据是从其他类型的序列数据中获得的。这种方法通过学习不同样本间的差异性和相似性,能够让模型充分学习数据的高级语义特征。具体来说,给定查询数据 q 和键数据集合 $\{k_0,k_1,\cdots,k_n\}$,对比学习的目标是确保 q 与键数据集合中对应的正键数据 k_i 匹配程度高于任何其他键数据。为了提高正键数据和查询数据之间的相似性,Oord等人[17]提出了InfoNCE损失函数:

$$L_{\mathrm{NCE}}=-\mathbb{E}\left[\ln\frac{\exp(f(q,k_i))}{\sum_{j=1}^{K}\exp(f(q,k_j))}\right] \tag{1}$$

其中,函数 f 用于计算查询数据和键数据之间的相似度得分,常用方法包括点积、双线性乘积或欧氏距离等。

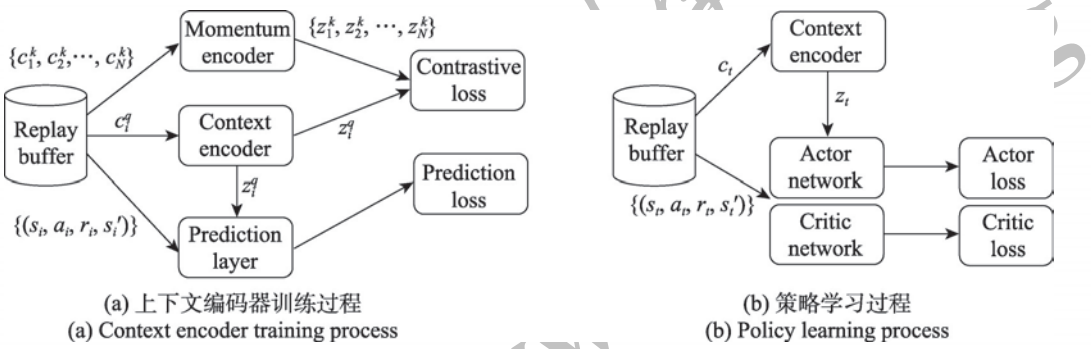
2 结合对比预测的离线元强化学习

本文主要对结合对比预测的离线强化学习算法进行介绍。图1展示了所提算法的整体结构。离线元强化学习需要同时解决离线强化学习中的价值函数估计问题和元强化学习中的任务推断问题。所提算法分别使用了行为克隆正则化和基于对比预测的上下文编码器以应对上述问题。为了有效利用离线数据,该算法通过在标准策略更新步骤中添加行为克隆正则化项以引导策略倾向于选择数据集中包含的动作[21]。此外,该算法将当前时刻之前的若干个经验样本视为上下文数据,并使用确定性的上下文编码器从中提取任务相关信息。同时本文利用对比学习来对不同任务轨迹背后的潜在结构加以区分,从而将不同任务所对应的隐变量区分开来。为了使得隐变量中包含更多的任务相关信息,所提算法还使用了预测网络拟合任务的状态转移函数和奖励函数。

2.1 上下文数据选择

基于上下文的元强化学习通常包括对上下文数据进行任务推断和基于任务推断的策略学习两个阶段,而选择何种形式的上下文数据将对后续策略学习产生直接影响。上下文数据的选择通常有两种方式:基于转换元组(transition tuple)的上下文数据和基于转换元组序列(transition sequence)的上下文数据。

Li等人[22]指出,转换元组 (s,a,r,s') 中包含了与任务相关的状态转移概率函数和奖励函数相关信息,因此可以用于任务推断。而Rakelly等人[12]和Li等人[13]指出在同一任务中使用不同策略所收集到的转换元组的概率分布是不同的,这将对任务推断产生影响。Rakelly等人[12]使用不同的数据分布分别进行任务推断和策略学习:从当前策略收集的同策略经验样本中随机采样数据作为上下文数据以进行任务推断,并基于此任务推断使用离策略经验样本进行策略学习。在离线元强化学习环境中智能体无法使用同策略收集经验样本,Li等人[22]选择使用从整个数据集中随机采样的经验样本计算上下文变量,并将上下文编码器和策略网络分开训练。由于使用基于转



(a) 上下文编码器训练过程
(a) Context encoder training process

(b) 策略学习过程
(b) Policy learning process

图1 算法结构示意图

Fig.1 Algorithm structure diagram

换元组的上下文数据需要利用当前任务经验样本数据进行任务推断,使用这种上下文数据的离线元强化学习算法在元测试阶段仍需要从测试任务中收集的经验样本,这种做法会减小离线元强化学习的应用范围。

此外,Fakoor 等人[23]提出了一种更为简单的方法:采用当前时刻 t 之前长度为 n 的转换元组序列 $\{(s_i,a_i,r_i)\}_{i=t-n,t-n+1,\cdots,n-1}$ 作为上下文数据($s_{t-i+1}=s_{t-i}'$,故省略 s_{t-i}')。由于相邻的经验样本序列在时间上具有连贯性,这种上下文数据有助于捕捉不同任务之间的转换函数和奖励函数中的内在区别。更重要的是,这种方法在选择动作时只需要使用当前轨迹上的数据,因此在测试阶段无需任何来自测试任务的额外数据,这是已有的离线元强化学习算法所无法做到的。该方法仅需同一轨迹中的经验样本在数据集中按顺序存储即可。本文中选择使用这种简单的上下文数据构建算法,上下文编码器的详细结构将在下文中介绍。

通过这两种方式得到的上下文数据中均包含任务相关信息。元强化学习算法将上下文数据输入任务推断网络得到上下文变量 z ,则 Q 值函数和策略可以表示为 $Q(s,a,z)$ 和 $\pi(s,z)$ 。基于上下文的元强化学习算法可与任意离策略强化学习算法相结合以进行策略学习[12,23]。

2.2　对比预测上下文编码器

由于离线元强化学习智能体无法与环境交互以获取更多数据来纠正任务推断错误,且策略学习过程依赖于对上下文数据的任务推断,如何有效地训练上下文编码器是离线元强化学习的关键问题之一。Rakelly 等人[12]使用端到端的方式直接使用策略学习的损失函数训练上下文编码器,这种方法无法有效捕捉任务之间的相关性,且离线强化学习中对价值函数的估计误差会对该方法产生较大影响。另一些工作仅通过区分任务之间的环境动态来学习任务变量[24-25],这种完全依赖低层次的状态预测或动作重建的方法往往会过度拟合不同任务中共享的动力学特征,进而阻碍后续策略学习过程。

作为离线元强化学习的重要组成部分,上下文变量应具有以下两点品质:(1)可区分性,不同的任务对应的上下文变量需要能够有效地进行区分;(2)任务相关性,各个任务对应的上下文变量需要包含足够的任务相关信息。本文算法将任务推断和策略学习进行解耦。在使用上下文编码器获得上下文变量后,引入对比学习方法分析隐藏在不同轨迹背后的潜在任务结构以提高可区分性,并利用预测网络拟合环境动态和奖励函数以增加上下文变量的任务相关性。

2.2.1　上下文编码器

本文所使用的上下文编码器主要包括全连接神经网络和基于门控循环单元(gated recurrent unit, GRU)[26]的循环神经网络两部分,网络结构如图 2 所示。如上文所述,所提方法采用时刻 t 之前长度为 n 的转换元组序列 $\{(s_i,a_i,r_i)\}_{i=t-n,t-n+1,\cdots,n-1}$ 作为上下文数据。图 2 中使用 x 代指元组 (s,a,r) 。为计算上下文变量 z ,所提算法首先使用全连接神经网络初步提取转换元组中的特征,再使用 GRU 提取包含在序列中的时序特征。由于 GRU 中的隐藏状态包含了之前所有时刻的相关信息,将上下文中最后一个时刻对应的隐藏状态作为上下文变量 z 。

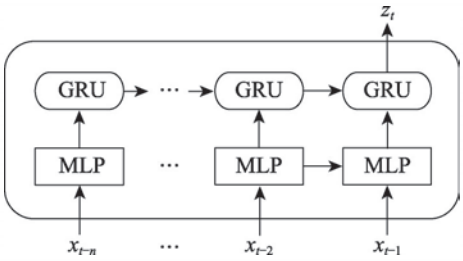


图 2　上下文编码器

Fig.2　Context encoder

2.2.2　上下文对比学习

本文通过引入对比学习方法将从相同任务中得到的上下文变量拉近,并将从不同任务中得到的上下文变量推开。传统的对比学习常使用数据增强方法生成查询数据和键数据,而这种方法并不适用于转换元组序列数据。因此,本文将从相同任务中采样得到的上下文数据视为正数据对,将不同任务中的上下文数据视为负数据对。具体来说,在每次训练中首先从数据集 D 中选择 N 个任务数据集,并从这些数据集中分别进行两次上下文数据采样,得到查询数据集合 $\{c_1^q,c_2^q,\cdots,c_N^q\}$ 和键数据集合 $\{c_1^k,c_2^k,\cdots,c_N^k\}$ 。其中键数据 c_i^k 是查询数据 c_i^q 对应的正键数据,而键数据 $\{c_j^k\}_{j=1,2,\cdots,N,j\neq i}$ 是查询数据 c_i^q 对应的负键数据。在获得查询数据集合和键数据集合之后,遵循 He 等人[18]的设置,本文使用上下文编码器 q_φ 对查询数据进行编码得到 $\{z_1^q,z_2^q,\cdots,z_N^q\}$,使用上下文编码器 q_φ 的动量平均版本 $q_{\bar{\varphi}}$ 对键数据进行编码得 $\{z_1^k,z_2^k,\cdots,z_N^k\}$,

计算相似度得分和对比损失函数：

$$L_{\text{cont}}=-\mathbb{E}\left[\sum_{i=1}^{N}\ln\frac{\exp(f(z_i^q,z_i^k))}{\sum_{j=1}^{N}\exp(f(z_i^q,z_j^k))}\right] \tag{2}$$

其中，f 表示用于计算相似性得分的双线性乘积法[17]，并使用动量平均[18]的方式更新键编码器参数。

2.2.3 预测网络

引入对比学习方法可以使得不同任务之间的任务变量能够满足可区分性的要求，但是由于所提方法将任务推断和策略学习解耦，所得仅使用对比学习的任务变量并不包含足够的任务相关信息。在本文中，不同的任务共享状态和动作空间而状态转移概率和奖励函数不同，因此本文使用预测网络预测下一时刻状态和奖励作为辅助任务，以捕获更多任务信息。具体来说，在每次训练中本文从数据集中采样转换元组 $\{s_t,a_t,r_t,s_t'\}$ 和对应的上下文数据 c_t。在使用上下文编码器获得上下文变量 z_t 之后，将状态 s_t、动作 a_t 和上下文数据 c_t 放入预测网络 p_θ 中以获取奖励和下一状态的预测值 $\hat{s}_{t+1},\hat{r}_t=p_\theta(s_t,a_t,z_t)$，通过最小化预测值和真实值之间的均方误差（mean squared error，MSE），促使上下文变量包含更多任务信息：

$$L_{\text{pred}}=\mathbb{E}[(\hat{s}_{t+1}-s_{t+1})^2+(\hat{r}_t-r_t)^2] \tag{3}$$

2.3 基于行为克隆的策略约束方法

“外推误差”（extrapolation error）是离线强化学习中的关键问题，可以概括为无法正确评估数据集外状态-动作对的价值[21]。已有的离线强化学习算法通过限制学习到的策略与数据集对应策略的距离来避免“外推误差”带来的影响。为此，多数算法除了在离策略强化学习的基础上进行算法层面的修改外，还使用了实现层面上的技巧：如对数据集中的状态-动作对分布进行建模[5]、修改网络架构[6]和预训练策略网络等[27]。这些技巧对算法结果有着显著影响，并降低了训练稳定性。而行为克隆使用监督学习方法，可以直接学习数据集中状态和动作的对应关系。因此将行为克隆项添加到标准策略更新步骤中，可以使策略倾向于选择数据集中包含的动作，从而减小外推误差带来的影响[21]。具体的，行为克隆项的损失函数可以写作：

$$L_{\text{BC}}=\mathbb{E}[(a-\pi(s))^2] \tag{4}$$

将此约束项加入到行动者-评论家（actor-critic）方法框架中，则行动者网络和评论家网络的损失函数分别可以写作：

$$L_{\text{Actor}}=-\mathbb{E}[\lambda Q(s,\pi(s))-L_{\text{BC}}] \tag{5}$$

$$L_{\text{Critic}}=\mathbb{E}[(r+\gamma Q'(s',\pi(s'))-Q(s,a))^2] \tag{6}$$

其中，λ 是用来调节强化学习（最大化价值函数）和模仿学习（最小化行为克隆损失函数）之间的权重比例的参数。本文选取 $\lambda=1/\mathbb{E}|Q(s,a)|$ 对价值函数绝对值大小进行正则化，此项不参与梯度反向传播过程，且能够避免不同任务中价值函数取值量级所带来的影响。

本节方法通过简单的变形即可适用于基于上下文的元强化学习情景，详见2.4节。

2.4 算法实现

所提方法以常用的离策略强化学习算法TD3算法[28]为基础，使用基于对比预测的上下文编码器进行有效任务推断，并引入行为克隆项以完成离线场景下的策略学习。最终的元训练算法和元测试算法如算法1和算法2所示。

算法1 元训练算法

输入：M 个经验回放数据集 $D_i=\{(s_{i,j},a_{i,j},r_{i,j},s_{i,j+1})\}_{j=1,2,\cdots,T}$，其中 D_i 其对应的任务 T_i 都是从任务分布 $P(T)$ 中采样得到。

1. 初始化行动者网络 π_θ，评论家网络 Q_{ϕ_1}、Q_{ϕ_2}，目标值网络 $Q_{\phi_1'}$、$Q_{\phi_2'}$，上下文编码器 q_φ 和预测网络 p_η

2. 初始化目标网络参数 $\theta'\leftarrow\theta,\phi_1'\leftarrow\phi_1,\phi_2'\leftarrow\phi_2$

3. While not done do

4. 随机采样 N 个数据集 $\{D_i'\}_{i=1,2,\cdots,N}$

5. 从 N 个数据集中分别进行数据采样，得到上下文查询数据集合 $\{c_1^q,c_2^q,\cdots,c_N^q\}$ 和其对应的经验样本集合 $\{s_i,a_i,r_i,s_i'\}_{i=1,2,\cdots,N}$，再次从数据中采样，得到上下文键数据集合 $\{c_1^k,c_2^k,\cdots,c_N^k\}$

6. 通过上下文编码器 q_φ 计算上下文变量，并计算对比学习损失函数 L_{cont}

7. 通过预测网络 p_η 预测下一状态和奖励，并计算预测损失函数 L_{pred}

8. 最小化损失函数 L_{cont} 和 L_{pred} 以更新上下文编码器 q_φ 和预测网络 p_η

9. For each $D_i'\in\{D_i'\}_{i=1,2,\cdots,N}$ do

10. 从 D_i' 中采样RL训练批次 b^i 和对应的上下文批次 c^i，并计算 L_{Critic}

11. 最小化 L_{Critic} 以更新评论家网络

12. If update Actor network do

13. 计算 L_{Actor} 并最小化 L_{Actor} 以更新行动者网络和评论家网络

14. $\phi_i'\leftarrow\tau\phi_i+(1-\tau)\phi_i',i=1,2;\theta'\leftarrow\tau\theta+(1-\tau)\theta'$

15. End if

16. End for

17. End while

算法 2 元测试算法

输入：测试任务 T'，行动者网络 π_θ，上下文编码器 q_φ。

1. 初始化定长上下文队列 c
2. While not done do
3. 　计算上下文变量 $z_t = q_\varphi(c)$
4. 　执行动作 $\pi_\theta(s_t, z_t)$，并收集数据更新 c
5. End while

所提算法中，离线元强化学习智能体主要由以下四种网络组成，分别是初始化行动者网络 π_θ，评论家网络 Q_{ϕ_1}、Q_{ϕ_2}，上下文编码器 q_φ 和预测网络 p_η。其中，上下文编码器 q_φ 用于从上下文数据 c 中提取包含任务相关信息的上下文变量 z；预测网络 p_η 用于隐式地对不同任务的奖励函数和环境动态进行建模，促使上下文变量中包含更多任务相关信息；评论家网络 Q_ϕ 利用上下文变量 z 估计在当前状态 s 下采取的动作 a 后的累计奖励的期望值，行动者网络 π_θ 利用上下文变量 z 在当前状态 s 下进行动作选择。所提算法用于解决离线元强化学习问题，本算法仅使用预先收集的离线数据集进行训练，并且在策略学习阶段，该算法并不会与环境进行在线交互或从环境中获得额外数据以改进算法。在算法测试阶段，使用该算法学习得到的智能体直接与从未见过的任务环境交互，并将得到的累积回报大小作为该算法的评价标准。

在元训练过程中，本文算法将任务推断和策略学习进行解耦，从而避免了已有算法在离线条件下值函数估计误差对任务推断过程的影响。具体地，在任务推断学习阶段，首先从所有数据集中随机采样 N 个数据集 $\{D_i'\}_{i=1,2,\cdots,N}$，并从中分别进行数据采样得到上下文查询数据集合 $\{c_1^q, c_2^q, \cdots, c_N^q\}$ 和其对应的经验样本集合 $\{s_i, a_i, r_i, s_i'\}_{i=1,2,\cdots,N}$，再次采样得到上下文键数据集合 $\{c_1^k, c_2^k, \cdots, c_N^k\}$。通过上下文编码器 q_φ 计算对应的上下文变量，并计算对比学习损失函数 L_{cont}；通过预测网络 p_η 预测下一状态和奖励，并计算预测误差损失 L_{pred}，最后使用梯度下降法最小化损失函数以更新上下文编码器和预测网络的参数。

在策略学习阶段，准备训练数据批次以更新行动者网络和评价者网络的参数。可以重复使用上阶段的上下文查询数据集合 $\{c_1^q, c_2^q, \cdots, c_N^q\}$ 和其对应的经验样本集合 $\{s_i, a_i, r_i, s_i'\}_{i=1,2,\cdots,N}$ 以节省采样时间。计算评论家的损失函数：

$$L_{\text{Critic}} = \mathbb{E}[(r + \gamma \min_{i=1,2} Q_{\phi_i'}(s', a', \bar{z}) - Q_\phi(s, a, \bar{z}))^2] \quad (7)$$

在计算 L_{Critic} 时，使用了两个技巧以提高训练过程的鲁棒性：(1)在估计目标值网络时使用的目标动作 $a' = \pi_{\theta'}(s', \bar{z}) + \epsilon, \epsilon \in \mathcal{N}(0, \sigma^2)$ 是由目标行动者网络输出结果叠加高斯随机噪声得到的；(2)使用截断双 Q 学习，即将 $Q_{\phi_i'}(s', a', \bar{z}), i \in \{1,2\}$ 中较小的 Q 值作为目标 Q 值，以减小对目标值网络的过估计。由于本文将上下文编码器的训练过程和策略学习过程解耦，在更新行动者和评价者网络时梯度没有通过上下文变量 z 进行传播，因此将此处上下文变量表示为 $\bar{z}$。

引入了行为克隆正则化和上下文变量的行动者网络的损失函数可以表示为：

$$L_{\text{BC}} = \mathbb{E}[(\pi_\theta(s, \bar{z}) - a)^2] \quad (8)$$

$$L_{\text{Actor}} = -\mathbb{E}[\lambda Q_{\phi_1}(s, \pi_\theta(s), \bar{z}) - L_{\text{BC}}] \quad (9)$$

为了进一步减小值函数估计误差对策略网络更新带来的影响，所提算法对策略网络进行延迟更新，即每更新两次值函数网络后再更新一次策略网络。通过最小化损失函数对行动者和批评者网络参数进行更新后，所提算法对目标网络参数进行软更新以提高训练稳定性。

在元测试阶段，智能体需要初始化一个定长上下文队列 c 用于存储上下文数据，在执行每个动作之前，智能体需要使用上下文编码器 q_φ 得到上下文变量 z，并基于上下文变量 z 和当前状态 s 进行动作选择 $a = \pi(s, z)$。在执行动作之后将准换元组 (s, a, r) 放入定长上下文队列中以更新上下文数据。重复上述步骤直至智能体完成测试。由于在测试阶段上下文数据是通过收集当前时刻之前的轨迹得来的，所提算法无需额外数据即可快速适应新任务。

3　实验

3.1　实验设计

为了验证智能体适应新任务的能力，本文在 MuJoCo[29] 实验平台中的 6 个元强化学习基准环境中进行实验。MuJoCo 是一款被广泛应用于强化学习算法测试的仿真物理引擎，旨在快速准确地模拟仿真物体与环境的相互作用。

对于每一个基准环境，智能体都需要完成多个不同的任务目标，并且智能体无法得知任务中的环境动态和奖励函数设置，需要智能体通过上下文数据对其所处任务进行推理。具体来说，在 Ant-Dir、Cheetah-Dir 和 Humanoid-Dir 中，不同的任务被设定

为智能体的目标运动方向,智能体需要在指定方向上快速前进以获得奖励。在Ant-Goal中,不同的目标位置定义了不同的任务,智能体需要快速到达指定目标点以最大化奖励。在Cheetah-Vel中,任务由智能体应达到的恒定速度所定义,奖励被设定为当前速度和目标速度之间差值的绝对值。上述5个基准环境中的不同任务对应环境动态相同而奖励函数不同。而在Walker-Params中,智能体需要适应不同的环境动态:不同任务对应着智能体不同的质量分布或者摩擦系数等,此类环境中的奖励大小取决于智能体的前进速度。实验环境如图3所示[29],表2列出了各任务环境信息及超参数设置。

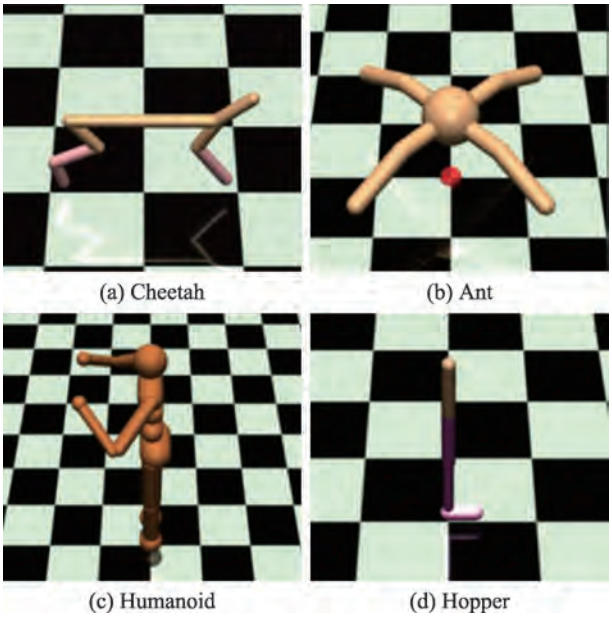


图3 仿真实验环境

Fig.3 Simulation environment

本文选择近年针对离线元强化学习问题所提出的几种代表性算法作为对比算法,其中包括基于确定性上下文变量的FOCAL算法[14]、基于双循环优化的MACAW算法[15]、将上下文推断网络直接引入离线Q学习的Context BCQ算法和基于策略蒸馏和奖励重标签的MBML算法[13]。离线元强化学习使用从多个任务中获取的固定数据集进行学习,并且在学习过程中不与环境发生交互。为此,对于每个基准环境,本文首先从任务分布中随机采样多个任务,并为每个任务使用最大熵强化学习算法SAC(soft actor-critic)[30]训练模型,使用训练好的策略来收集轨迹。具体的,针对每个任务收集500条轨迹用于后续离线学习过程。

和其他算法保持一致,本文选择使用训练次数作为横坐标,纵坐标是模型在元测试任务中的平均累计奖励。评估实验性能的指标通常是训练得到的模型在元测试任务集上的评估结果,使用训练得到的模型在测试任务集上收集到的轨迹的累计奖励大小反映了模型在新任务中的泛化能力。具体地,每一个基准环境中进行5次重复实验,并取最终多次实验结果的平均值。实验代码使用Python 3.7编写,基于Pytorch框架实现。运行环境为Ubuntu 18.04操作系统,Intel® Core™ i7-9700K 8核心处理器和NVIDIA GTX 2060显卡。

3.2 实验结果分析

3.2.1 总体实验结果

通过图4和表3中的实验结果可以看出,本文方法在模型性能和算法收敛速度方面都取得了较好的结果。在模型性能方面,所提算法的最终结果相较于FOCAL算法平均提升约25%,相较于MACAW算法提升约40%。在模型稳定性方面,所提算法的最终结果的方差大小较FOCAL算法降低约30%,这说明本文算法更为稳定。由于MACAW算法在测试阶段需要进行额外训练,而本文算法和FOCAL未进行额外训练,该算法的方差总体较低。在多数环境中(Cheetah-Dir、Cheetah-Vel、Humanoid-Dir和Walker-Params),提出方法的训练效果在训练速度和最终性能相较于对比算法都有显著提升,这是由于所提出的对比预测方法既可以增加不同任务的上下文变量的区分度,

表2 各任务环境信息及超参数设置

Table 2 Environment information and hyperparameter settings

环境	Obs Dim	Action Dim	Train Tasks	Test Tasks	Context Length	Z dimension	Policy Noise
Cheetah-Dir	20	6	2	2	10	30	0.2
Ant-Dir	27	8	2	2	15	20	0.3
Ant-Goal	113	8	100	30	25	30	0.3
Cheetah-Vel	20	6	100	30	20	20	0.2
Humanoid-Dir	376	17	100	30	20	20	0.2
Walker-Params	17	6	50	20	10	30	0.1

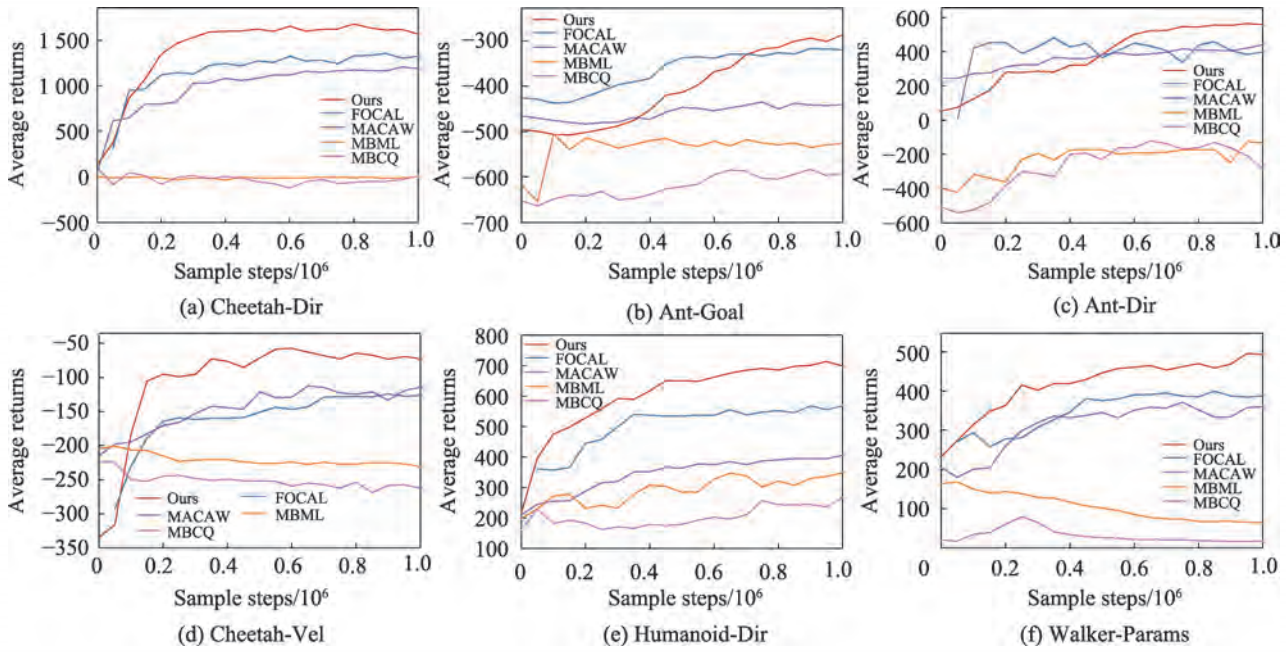


图4　算法性能比较

Fig.4　Comparison of algorithm performance

表3　各算法最终实验结果

Table 3　Final experimental results of each algorithm

环境	Ours	FOCAL	MACAW	MBML	MBCQ
Cheetah-Dir	1 567.9±103.5	1 323.4±168.7	1 190.1±80.0	−5.0±53.6	8.9±131.9
Ant-Goal	−286.4±38.3	−319.2±50.9	−441.1±42.6	−526.5±22.1	−591.2±17.8
Ant-Dir	558.8±16.6	397.1±83.4	440.3±26.4	−132.4±108.6	−281.0±70.8
Cheetah-Vel	−78.5±17.2	−126.9±32.8	−118.6±8.5	−230.7±9.1	−262.5±9.3
Humanoid-Dir	689.2±65.0	568.4±43.1	405.9±28.8	348.7±66.3	267.8±88.4
Walker-Params	493.2±28.7	387.6±43.4	358.9±59.7	64.6±8.5	17.4±2.2

同时能够使得上下文变量中包含更多的任务信息。在其余两个环境中(Ant-Goal和Ant-Dir),所提算法虽然收敛较慢,但是最终结果仍优于其他对比算法。出现这种现象的原因在于这两组环境中不同任务下的智能体行为具有相似性,所提算法需要更多训练过程才能对不同任务进行有效区分;表现较好的FOCAL算法在测试阶段可以利用新任务上的额外数据进行任务推断,而本文算法使用直接测试(Zero-Shot)的方式。

3.2.2　对比预测上下文编码器作用分析

为了验证对比学习和预测网络的作用,本文在Half-Cheetah-Vel基准环境中分别测试了所提算法去除预测网络(w/o P)、去除对比学习(w/o C)和使用策略学习损失函数学习上下文编码器(w/o P&C)的效果,实验结果如图5所示。结果表明在Cheetah-Vel基准环境,仅使用对比学习时的收敛速度优于仅使用预测网络,这是由于对比学习可以使得智能体更好地分辨不同任务对应的上下文变量,而不使用对比学习时智能体需要大量训练过程才能区分不同环境的状态转移和奖励函数特征。仅使用策略学习损失

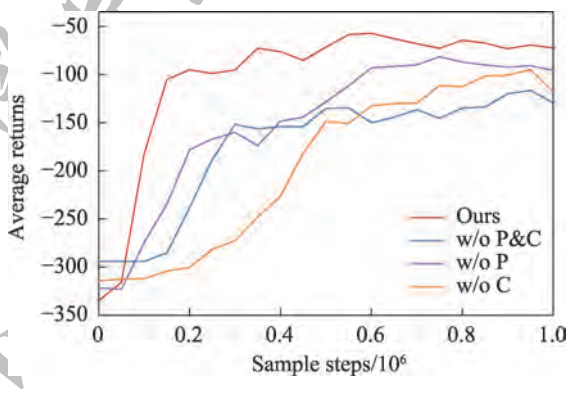


图5　上下文编码器消融实验

Fig.5　Ablation study on context encoder

函数学习上下文编码器时会由于奖励策略学习损失函数无法有效捕捉任务之间的相关性，最终结果表现最差。

3.2.3 上下文数据长度作用分析

为了验证上下文数据长度对实验结果的影响，本文在Cheetah-Vel基准环境对三种不同长的上下文数据进行测试。实验结果如图6所示，其中 $CL=n$ 表示使用长度为 n 的上下文数据的实验结果。在对比实验部分，本文采用的上下文数据长度为 $n=20$ 。实验结果表明使用不同长度的上下文数据会对结果产生一定影响。当使用的上下文序列较长或较短时，都会导致上下文编码器对任务的推理能力下降。当上下文序列较短时，所包含信息较少，将影响学习速度，但是对最终结果影响较小；当上下文序列较长时，包含了冗余信息，同样会影响上下文编码器的推理能力，且会影响算法的最终表现。

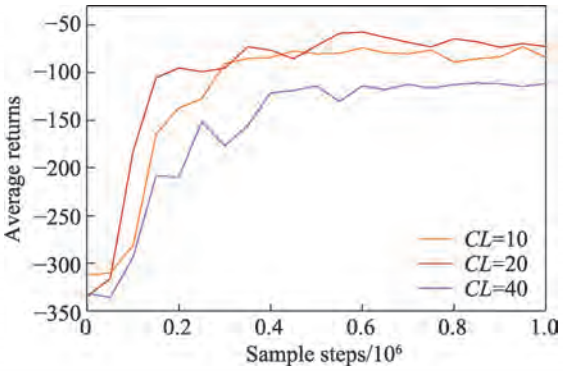


图6 上下文数据长度消融实验

Fig.6 Ablation study on context length

上述对比实验的结果表明：本文方法不仅具有更快的收敛速度，收敛结果也相对于对比算法有25个百分点以上的提升。因此本文算法具有更高的训练效率和更好的泛化性能。本文还验证了对比预测上下文编码器中的对比学习和预测网络对结果的影响，以及不同长度的上下文数据对结果的影响。

4 结束语

有限数据下的任务推理和离线场景中的策略学习是离线元强化学习算法的主要问题。针对这两个问题，本文提出了结合对比预测的离线元强化学习算法。所提算法首先使用行为克隆正则化项引导策略倾向于选择数据集中包含的动作，避免了离线学习得到的智能体选择未知动作。该算法使用上下文编码器从转换元组序列得到上下文变量，并使用对比学习和预测网络增强了智能体的任务推断能力。不同任务环境下的测试结果证明了所提算法能够加快训练速度以及增加模型的稳定性，在面对新任务时的泛化能力较已有算法有显著提升。

在未来的工作中，将探究如何在测试阶段通过少量与环境的交互进一步增强智能体在新环境中的适应能力，从而将这类方法扩展到机械臂控制等实际应用场景中。

参考文献：

[1] ARULKUMARAN K, MARC D, MILES B, et al. Deep reinforcement learning: a brief survey[J]. IEEE Signal Processing Magazine, 2016, 34(6): 26-38.

[2] KOBER J, BAGNELL J A, PETERS J. Reinforcement learning in robotics: a survey[J]. The International Journal of Robotics Research, 2013, 32(11): 1238-1274.

[3] SUTTON R S, BARTO A G. Reinforcement learning: an introduction[M]. Cambridge: MIT Press, 2018.

[4] LEVINE S, KUMAR A, TUCKER G, et al. Offline reinforcement learning: tutorial, review, and perspectives on open problems[J]. arXiv:2005.01643, 2020.

[5] FUJIMOTO S, MEGER D, PRECUP D. Off-policy deep reinforcement learning without exploration[C]//Proceedings of the 36th International Conference on Machine Learning, Long Beach, Jun 9-15, 2019. Cambridge: JMLR, 2019: 2052-2062.

[6] KUMAR A, ZHOU A, TUCKER G, et al. Conservative Q-learning for offline reinforcement learning[C]//Advances in Neural Information Processing Systems 33, Dec 6-12, 2020: 1179-1191.

[7] ERNST D, GEURTS P, WEHENKEL L. Tree-based batch mode reinforcement learning[J]. Journal of Machine Learning Research, 2005, 6: 503-556.

[8] WU Y, TUCKER G, NACHUM O. Behavior regularized offline reinforcement learning[J]. arXiv:1911.11361, 2019.

[9] FINN C, ABBEEL P, LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks[C]//Proceedings of the 34th International Conference on Machine Learning, Sydney, Aug 6-11, 2017. Cambridge: JMLR, 2017: 1126-1135.

[10] GUPTA A, MENDONCA R, LIU Y, et al. Meta-reinforcement learning of structured exploration strategies[C]//Advances in Neural Information Processing Systems, 31, Montréal, Dec 3-8, 2018. New York: Curran Associates, 2018: 5302-5311.

[11] ROTHFUSS J, LEE D, CLAVERA I, et al. ProMP: proximal meta-policy search[C]//Proceedings of the 2019 International Conference on Learning Representations, New Orleans, May 6-9, 2019: 1-25.

[12] RAKELLY K, ZHOU A, FINN C, et al. Efficient off-policy

meta-reinforcement learning via probabilistic context variables[C]//Proceedings of the 36th International Conference on Machine Learning, Long Beach, Jun 9-15, 2019: 5331-5340.
[13] LI J, VUONG Q, LIU S, et al. Multi-task batch reinforcement learning with metric learning[C]//Advances in Neural Information Processing Systems 33, Dec 6-12, 2020. New York: Curran Associates, 2020: 6197-6210.
[14] LI L, YANG R, LUO D. FOCAL: efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization[C]//Proceedings of the 9th International Conference on Learning Representations, May 3-7, 2021: 1-11.
[15] MITCHELL E, RAFAILOV R, PENG X B, et al. Offline meta-reinforcement learning with advantage weighting[C]//Proceedings of the 38th International Conference on Machine Learning, Jul 18-24, 2021: 7780-7791.
[16] PENG X B, KUMAR A, ZHANG G, et al. Advantage-weighted regression: simple and scalable off-policy reinforcement learning[J]. arXiv:1910.00177, 2019.
[17] OORD A V, LI Y, VINYALS O. Representation learning with contrastive predictive coding[J]. arXiv:1807.03748, 2018.
[18] HE K, FAN H, WU Y, et al. Momentum contrast for unsupervised visual representation learning[C]//Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, Jun 13-19, 2020. Piscataway: IEEE, 2020: 9729-9738.
[19] LASKIN M, SRINIVAS A, ABBEEL P. CURL: contrastive unsupervised representations for reinforcement learning[C]//Proceedings of the 38th International Conference on Machine Learning, Jul 12-18, 2020: 5639-5650.
[20] FU H, TANG H, HAO J, et al. Towards effective context for meta-reinforcement learning: an approach based on contrastive learning[C]//Proceedings of the 35th AAAI Conference on Artificial Intelligence, the 33rd Conference on Innovative Applications of Artificial Intelligence, the 11th Symposium on Educational Advances in Artificial Intelligence, Feb 2-9, 2021. Palo Alto: AAAI Press, 2021: 7457-7465.
[21] FUJIMOTO S, GU S S. A minimalist approach to offline reinforcement learning[C]//Advances in Neural Information Processing Systems 34, Dec 6-14, 2021: 20132-20145.
[22] LI L, HUANG Y, CHEN M, et al. Provably improved context-based offline meta-RL with attention and contrastive learning[J]. arXiv:2102.10774, 2021.
[23] FAKOOR R, CHAUDHARI P, SOATTO S, et al. Meta-Q-Learning[C]//Proceedings of the 2020 International Conference on Learning Representations, Apr 26-May 1, 2020: 1-17.
[24] ZHOU W, PINTO L, GUPTA A. Environment probing interaction policies[C]//Proceedings of the 2019 International Conference on Learning Representations, New Orleans, May 6-9, 2019: 1-13.
[25] LEE K, SEO Y, LEE S, et al. Context-aware dynamics model for generalization in model-based reinforcement learning[C]//Proceedings of the 37th International Conference on Machine Learning, Jul 12-18, 2020: 5757-5766.
[26] CHUNG J, GULCEHRE C, CHO K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv:1412.3555, 2014.
[27] KOSTRIKOV I, FERGUS R, TOMPSON J, et al. Offline reinforcement learning with fisher divergence critic regularization[C]//Proceedings of the 38th International Conference on Machine Learning, Jul 18-24, 2021: 5774-5783.
[28] FUJIMOTO S, HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[C]//Proceedings of the 35th International Conference on Machine Learning, Stockholmsmässan, Jul 10-15, 2018: 1582-1591.
[29] TODOROV E, EREZ T, TASSA Y. MuJoCo: a physics engine for model-based control[C]//Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Oct 7-11, 2012. Piscataway: IEEE, 2012: 5026-5033.
[30] 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, Stockholmsmässan, Jul 10-15, 2018: 1861-1870.

韩旭（1997—），男，安徽亳州人，硕士研究生，主要研究方向为强化学习。
HAN Xu, born in 1997, M.S. candidate. His research interest is reinforcement learning.

吴锋（1984—），男，福建宁德人，博士，副教授，主要研究方向为智能机器人、强化学习、多智能体系统。
WU Feng, born in 1984, Ph.D., associate professor. His research interests include intelligent robots, reinforcement learning and multi-agent systems.