论文复现：For SALE: State-Action Representation Learning for Deep Reinforcement Learning

摘要

TD7 (TD3 + 4 additions) 算法在 Twin Delayed Deep Deterministic policy gradient (TD3) 算法的基础上提出了四项改进：状态-动作表征、检查点、优先经验回放和行为克隆。通过这四项改进，TD7 在多个连续控制问题上表现出了比 TD3 算法更高的样本效率和稳定性。其中稳定性的提升主要来自检查点的使用，而 TD7 作者认为检查点这一改进与进化强化学习算法的研究十分相关。基于此，本项目首先对 TD7 算法的结果进行了复现，然后尝试借鉴进化强化学习算法中的相关设计，对 TD7 算法进行了改进。具体而言，本项目的方法将维护一个智能体种群，将整个学习分为两个阶段来分别注重于探索和利用，以最终提升算法的探索能力和效率。在多个测试环境的对比结果表明，改进方法的性能在一些环境上较原算法有一定的提升。

关键词：强化学习；状态-动作表征

1 引言

强化学习（Reinforcement Learning，RL）作为机器学习的一个重要分支，已经在一些任务上成功训练出能够达到甚至超过人类水平的智能体。例如，DeepMind 在《Nature》上发表的深度 Q 网络算法（Deep Q-Network，DQN）[12] 在雅达利游戏上展现了相当于人类测试者的表现；在此之后，DeepMind 又推出了 AlphaGo [16]，并在围棋领域中击败了人类的世界冠军。RL 能够在近些年取得许多突破性成就的一个重要原因是其与深度神经网络的成功结合，这一成功的结合通常被称之为“深度强化学习”（Deep Reinforcement Learning, DRL）。近些年，DRL 已经成功地应用在机器人控制 [1]、推荐系统 [20]、游戏 AI [19] 等领域。

本课程项目复现的论文所提出的 DRL 算法的名称为 TD7 (TD3 + 4 additions) [4]。TD7 算法在 Twin Delayed Deep Deterministic policy gradient (TD3) 算法 [5] 的基础上提出了四项改进：状态-动作表征、检查点、优先经验回放和行为克隆。通过这四项改进，TD7 在多个连续控制问题上表现出了比 TD3 算法更高的样本效率和稳定性。其中稳定性的提升主要来自检查点的使用，而 TD7 作者认为检查点这一改进与进化强化学习算法的研究 [10] 十分相关。基于此，本项目首先对 TD7 算法的结果进行了复现，然后尝试借鉴进化强化学习算法中的相关设计，对 TD7 算法进行了改进。具体而言，本项目的方法将维护一个智能体种群，将整个学习分为两个阶段来分别注重于探索和利用，以最终提升算法的探索能力和效率。本项目在多

个基于 MuJoCo 物理引擎的连续控制环境进行了测试，实验结果表明改进方法的性能在一些环境上较原算法有一定的提升。

2 相关工作

2.1 TD3

TD3 算法指出 Deep Deterministic policy gradient（DDPG）[11] 算法会高估动作的价值的问题，并通过利用两个价值网络来缓解这个问题，这个方式类似于双 Q 学习（Double Q-learning）[8] 中所使用的方法。在 TD3 中，两个目标价值网络分别给出对动作价值的单独估计，然后取更小的估计值来计算目标值。TD3 的价值网络的目标价值通过下式（1）计算：

$$y_i = r_i + \gamma \min_{j=1,2} Q'_j(s_{i+1}, a_{i+1}) \tag{1}$$

为了在更新过程中减小累计的估计误差，TD3 提出延迟策略更新（delayed policy update）技术，即多次更新价值网络后才更新策略和目标网络。此外，在目标网络估计动作的值时，TD3 为目标策略网络选择的动作添加了随机噪声，这减小了目标网络的估计方差，使得值函数的近似值更加平滑。所以实际上，式（1）的 a_{i+1} 通过下式（2）来计算：

$$a_{i+1} = \mu'(s_{i+1}) + \epsilon, \quad \epsilon \sim \text{clip}\left(\mathcal{N}(0, \sigma), -c, c\right) \tag{2}$$

式中，clip 函数表示对从分布中采样的结果进行约束，约束范围为 $[-c, c]$，c 为一个较小的正数，通常取值为 0.2。

2.2 进化强化学习

为了结合进化算法（Evolution Algorithm, EA）和强化学习的优点，已经提出了各种混合算法。其中一种方法是 ERL [10]，它结合了遗传算法（Genetic Algorithm，GA）和 DDPG，分别优化策略网络种群和额外的 RL 策略。策略种群生成多样化的经验以训练 RL 策略，而 RL 策略则定期插入到群体中，以提供新的个体。尽管 ERL 在连续控制任务中展示了其优势，但其遗传算子具有破坏性，可能会导致已学习信息的遗忘。为了解决这个问题，近端蒸馏进化强化学习（Proximal Distilled ERL，PDERL）[3] 提出了近端突变和蒸馏交叉。近端突变通过利用梯度和减少每个权重的突变强度，增加了突变的稳定性。蒸馏交叉利用父代的经验，通过模仿学习（Imitation Learning）[13] 训练其子代。这两个算子在相同任务上表现优于原始算子。

受到 ERL 框架的启发，一些研究探索了一些能够利用更多方法优势的方案。交叉熵方法强化学习（CEM-RL）[14] 提出了一个不同的框架，将 RL 算法与交叉熵方法（Cross-Entropy Method，CEM）相结合。CEM-RL 直接将 RL 梯度步应用于一半的群体，并使用表现最好的那一半来生成个体，从而减少了那些会降低性能的梯度步的影响。基于进化的柔性演员 - 评论家（Evolution-based Soft Actor Critic，ESAC）[17] 算法结合了进化策略（Evolution Strategy，ES）[2] 和 SAC [7]，并且利用了其提出的后见交叉来促进个体之间的技巧迁移。此外，ESAC 还提出了自动变异调整来改进算法对超参数的敏感性。

3 复现算法介绍

TD7 算法 [4] 在 TD3 算法 [5] 的基础上提出了四项改进：状态-动作表征、检查点、优先经验回放和行为克隆。通过这四项改进，TD7 分别在离线和在线强化学习的测试问题上表现出了比 TD3 算法更好的性能和稳定性。在这一章节中，这四项改进将会被分别详细介绍。

3.1 状态-动态表征

TD7 作者认为表征学习已经在图像输入的强化学习任务中验证了其有效性，然而在矢量输入的强化学习任务，例如物理控制问题中，表征学习经常被忽略。因此，作者提出一个新颖的状态-动作表征方法来对环境的动态进行建模，并通过学习到的表征来有效地提高算法学习的效率。

首先，作者使用两个表征网络 f 和 g 来分别学习状态表征和状态-动作表征：

$$z^s := f(s), \quad z^{sa} := g(z^s, a). \tag{3}$$

式中状态表征网络 f 输入状态 s，输出状态表征 z^s; 状态-动作表征网络 g 输入状态表征 z^s 和动作 a, 输出状态-动作表征 z^{sa}。

接着，作者通过使用状态-动态表征来预测下一步的状态表征来计算 loss 以更新两个表征网络：

$$\mathcal{L}(f, g) := \left(g(f(s), a) - |f(s')|_\times\right)^2 = \left(z^{sa} - |z^{s'}|_\times\right)^2. \tag{4}$$

式中 s' 表示下一步的状态，$\times$ 表示该部分不进行梯度运算。

最后，作者使用两个表征分别作为策略、价值网络的输入来提高其学习效率：

$$Q(s, a) \to Q(z^{sa}, z^s, s, a), \quad \pi(s) \to \pi(z^s, s). \tag{5}$$

式中 Q 表示价值网络，π 表示策略网络。

3.2 检查点

深度强化学习算法的训练过程经常出现不稳定的情况 [9]，因此作者认为深度强化学习算法需要一些技术来提高其训练的稳定性。在本节中，作者提出使用检查点来保持策略的评估性能，而不受当前学习策略质量的影响。

检查点是模型参数在训练过程中某个特定时间点的快照。在监督学习中，检查点常用于根据验证误差回溯到一组高性能的参数，并在评估中保持一致的性能 [18]。然而，这种技术在深度强化学习工具集中却出奇地缺失，尽管它可以用于稳定策略性能。在强化学习中，使用在训练期间获得高回报的策略的检查点，而非当前策略，可以提高测试时性能的稳定性。对于离策略深度强化学习算法，标准的训练范式是在每个时间步后进行训练（通常是一对一的比例：一个梯度步骤对应一个数据点）。然而，这意味着策略会在每个回合中不断变化，使得性能评估变得困难。类似于许多基于策略的算法 [15]，作者建议在若干评估回合中保持策略固定，然后批量进行原本应发生的训练，也即在多个评估回合中收集 N 个数据点后训练 N 次。

检查点这一技术十分类似于进化强化学习的方式 [3, 10]，即通过多个评估回合的得分来判断当前策略是否优于之前的最佳策略，并相应地保存最优的策略。

3.3 优先经验回放和行为克隆项

TD7 算法的第三项改进技术是一个优先经验回放 [6]。在该优先经验回放缓存中，每一个经验元组 $i := (s, a, r, s')$ 的采样概率为：

$$p(i) = \frac{\max\left(|\delta(i)|^{\alpha}, 1\right)}{\sum_{j \in D} \max\left(|\delta(j)|^{\alpha}, 1\right)}. \tag{6}$$

其中 $\delta(i) := Q(s, a) - y$，y 是学习目标，α 是控制优先级使用程度的超参数。$\delta(i)$ 也即时序差分损失，从公式 (6) 可以看出时序差分损失越大的经验元组，被采样到的概率也越大。该技术正是通过时序差分误差来对经验的采样进行优先级排序，使得更有价值的经验被更频繁地回放，从而提高算法学习效率。

TD7 的最后一项改进主要针对离线强化学习任务，通过在策略网络的损失上增加一个行为克隆项，来引导策略学习离线数据的行为：

$$\pi \approx \arg\max_{\pi} \mathbb{E}_{(s,a)\sim D}\left[Q(s, \pi(s)) - \lambda \left|\mathbb{E}_{s\sim D}\left[Q(s, \pi(s))\right]\right| \cdot (\pi(s) - a)^2\right]. \tag{7}$$

上式中 λ 是一个超参数，用来调控行为克隆项对策略损失的影响，当 λ 取 0 时，该式就转变成了在线强化学习任务中的策略损失。

4 复现细节

4.1 与已有开源代码对比

本项目尝试使用进化强化学习的方法对原算法进行改进，将 TD7 的算法框架改成了进化强化学习算法的训练框架。具体而言，本项目的方法将维护一个智能体种群，种群中的每个个体维护它们自己的策略和价值网络。此外，将整个训练过程分为两个阶段来分别注重于探索和利用。在第一阶段，每个个体都分别通过强化学习方法进行优化，这使得它们可以相对独立地进行探索。许多进化强化学习方法没有维护完整的智能体种群的主要原因是同时训练多个智能体的计算成本较高。而本项目的方法通过公共的回放缓存来在种群的个体共享学到的信息以提高学习效率并同时避免使用过多的计算资源。在第二阶段，更多的计算资源将会被分配给最佳个体，即只有最佳个体通过强化学习算法进行进一步的优化以实现更好的最终性能。

4.2 实验环境搭建

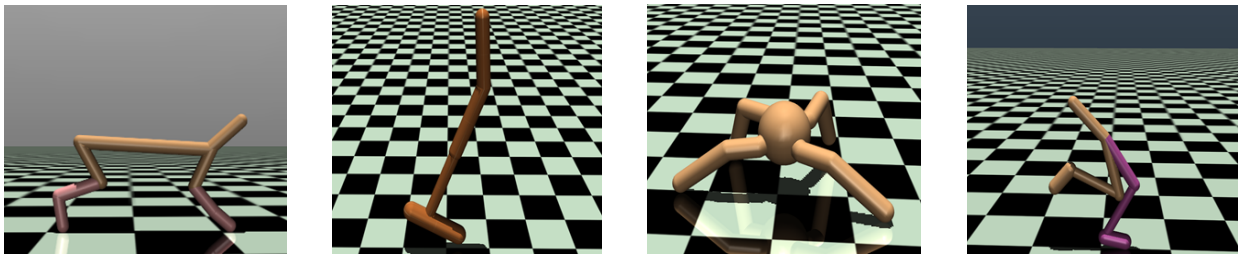
对比实验在 anaconda 软件搭建的虚拟环境中进行。虚拟环境主要依赖如下：python 3.8, pytorch 1.8, cudatoolkit 10.2, gym 0.23, mujoco-py 2.1。此外需要在额外安装 MuJoCo 引擎的相关依赖，例如 mujoco210。

4.3 创新点

本项目主要借鉴进化强化学习算法中的相关设计，尝试对 TD7 算法进行了改进。具体而言，本项目的方法将维护一个智能体种群，将整个学习分为两个阶段来分别注重于探索和利用，以最终提升算法的探索能力和效率。

5 实验结果分析

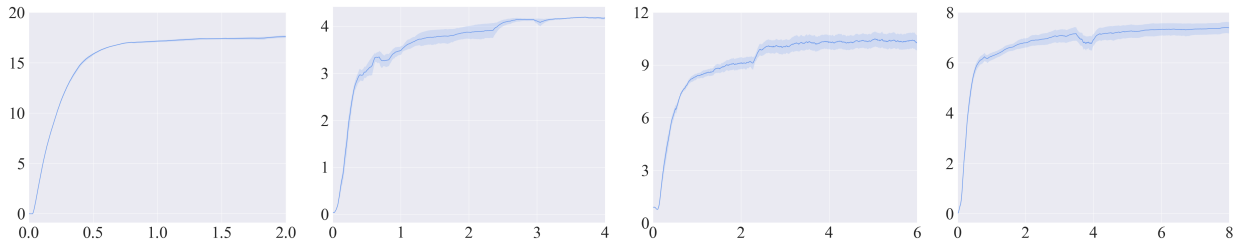
本项目首先使用 TD7 算法开源的代码对论文中的实验结果进行了复现。采用的测试环境是和原文一样的 MuJoCo 仿真环境，每个测试环境使用不同随机种子独立运行五次，取平均值作为最终结果。每个测试环境的示意图如下图 1所示。



(a) HalfCheetah (b) Hopper (c) Ant (d) Walker2d

图 1. MuJoCo 连续控制环境。

原算法复现的学习曲线如下图 2, 图中横坐标为训练步数（单位：百万），纵坐标为得分（单位：千）。曲线旁边的阴影部分表示结果的标准差。通过和原文的结果对比，复现的结果基本和原文的结果保持一致。



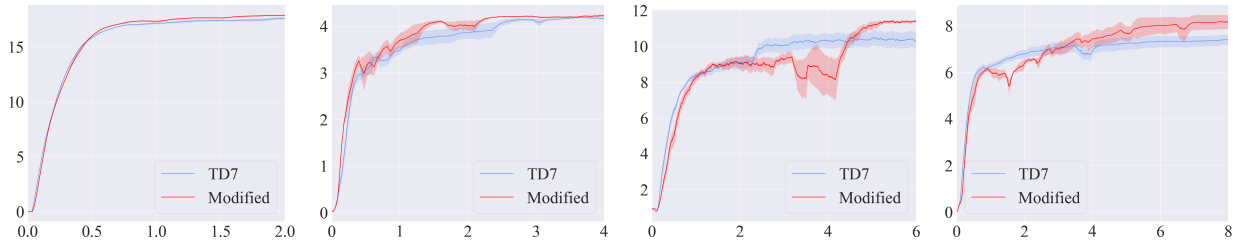
(a) HalfCheetah (b) Hopper (c) Ant (d) Walker2d

图 2. 原算法复现的结果。

在此之后，本项目将修改后的方法在相同环境下进行测试，每个测试环境同样使用不同的随机种子独立运行五次，取平均值作为最终结果。将修改后的方法的学习曲线与复现的学习曲线对比如下图 3，图 3中横纵坐标的意义和图 2 保持一致，其中红色曲线表示修改后方法的结果。从图 3中的结果可以看出，修改的方法在 Ant 和 Walker2d 两个测试环境相对原算法有一定提升，在另外两个环境和原算法基本持平。

6 总结与展望

本项目首先对 TD7 算法的结果进行了复现。接着，本项目借鉴进化强化学习算法中的相关设计，尝试对 TD7 算法进行了改进。具体而言，本项目的方法将维护一个智能体种群，将整个学习分为两个阶段来分别注重于探索和利用，以最终提升算法的探索能力和效率。在多个测试环境的对比结果表明，改进方法的性能在一些环境上较原算法有一定的提升。本项目



(a) HalfCheetah (b) Hopper (c) Ant (d) Walker2d

图 3. 修改后方法和原方法的结果对比。

对进化强化学习算法框架的设计相对较简单，对算法框架进行进一步的优化和设计是一个未来可能的研究方向。

参考文献

[1] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on robot learning*, pages 403–415. PMLR, 2023.

[2] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies–a comprehensive introduction. *Natural computing*, 1:3–52, 2002.

[3] Cristian Bodnar, Ben Day, and Pietro Lió. Proximal distilled evolutionary reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3283–3290, 2020.

[4] Scott Fujimoto, Wei-Di Chang, Edward Smith, Shixiang Shane Gu, Doina Precup, and David Meger. For sale: State-action representation learning for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[5] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[6] Scott Fujimoto, David Meger, and Doina Precup. An equivalence between loss functions and non-uniform sampling in experience replay. *Advances in neural information processing systems*, 33:14219–14230, 2020.

[7] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[8] Hado Hasselt. Double q-learning. *Advances in neural information processing systems*, 23, 2010.

[9] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[10] Shauharda Khadka and Kagan Tumer. Evolution-guided policy gradient in reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[11] TP Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[13] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[14] Aloïs Pourchot and Olivier Sigaud. Cem-rl: Combining evolutionary and gradient-based methods for policy search. *arXiv preprint arXiv:1810.01222*, 2018.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[16] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[17] Karush Suri. Off-policy evolutionary reinforcement learning with maximum mutations. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 1237–1245, 2022.

[18] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[19] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.

[20] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 world wide web conference*, pages 167–176, 2018.