arXiv:2601.14705v1 [cs.NE] 21 Jan 2026

结合进化变异的近端策略优化

Casimir Czworkowski CCZWORK1@JH.EDU
约翰斯·霍普金斯大学
Stephen Hornish SHORNIS1@JH.EDU
约翰斯·霍普金斯大学
Alhassan S. Yasin AYASIN1@JHU.EDU
约翰斯·霍普金斯大学

摘要

近端策略优化（PPO）是一种广泛使用的强化学习算法，以稳定性和样本效率著称，但由于探索有限，常会遭遇过早收敛。本文提出 POEM（结合进化变异的近端策略优化），这是对 PPO 的一种创新改进，引入了受进化算法启发的自适应探索机制。POEM 通过监测当前策略与先前策略移动平均之间的 Kullback–Leibler（KL）散度，增强策略多样性。当策略变化趋于微小、表明陷入停滞时，POEM 会触发策略参数的自适应变异，以促进探索。我们在四个 OpenAI Gym 环境中评估了 POEM：CarRacing、MountainCar、BipedalWalker 和月球着陆器。通过使用贝叶斯优化技术进行广泛的微调，并采用 Welch t检验进行统计检验，我们发现 POEM 在四项任务中的三项上显著优于 PPO（BipedalWalker：$t = -2.0642$,$p = 0.0495$；CarRacing：$t = -6.3987$, $p = 0.0002$；MountainCar：$t = -6.2431$, $p < 0.0001$），而在月球着陆器上的表现差异在统计学上并不显著（$t = -1.8707$, $p = 0.0778$）。我们的结果凸显了将进化原理融入策略梯度方法、克服探索与利用权衡的潜力。

1 引言

强化学习（RL）算法往往难以平衡探索与利用。近端策略优化（PPO）等模型在通过小幅、渐进式的策略更新进行学习时，通常表现更好。这些微小更新也有负面影响：模型可能因此过早收敛于局部最优，尤其是在奖励结构无效的环境中。标准 PPO 通常采用基于 Kullback–Leibler（KL）散度的策略更新，以防止策略发生过度变化，但这可能会无意中抑制探索，因为它会阻止策略偏离已有认知太远。结果便是策略停滞：模型实际上学到的是次优策略，而不是积极探索不同的行为。

在本文中，我们提出了一种对近端策略优化（PPO）的新颖改进，其中融入了深受进化算法启发的自适应探索机制。该改进的核心机制是使用KL散度监测策略随时间发生的行为变化。具体而言，新算法跟踪过去策略的移动平均值，以计算当前策略与过去参考策略之间的KL散度。这实际上成为衡量策略新颖性的指标，其值较高则定量表明

该策略与近期历史相比存在显著差异。当该值过低时，表明策略变得过于可预测，或可能陷入停滞，从而触发对策略参数进行进化变异。这样，如果算法在一段时间内没有显著变化，就会尝试采取不同的做法。这些变化是对策略参数进行随机置换，类似于遗传算法中的变异。变异具有自适应性，其幅度会根据多样性指标降低的程度进行调整。变异后，模型恢复标准的近端策略优化（PPO）操作。

我们假设，将基于KL散度的多样性触发机制与自适应进化变异整合到近端策略优化中，相较于标准近端策略优化，将显著提升其在控制任务上的表现。

2 相关工作

我们对近端策略优化的改进主要通过使用KL散度监控策略多样性来触发变异，这与其他现有的强化学习方法有所不同。多篇论文表明，多样性是一个值得探索的重要指标。例如，Wu等人（2023）介绍了一种通过最大化互信息，以无监督方式学习多样技能的方法。与我们以监督学习和奖励为重点的策略方法不同，他们旨在生成一套技能，而非仅生成单个近端策略优化策略，从而增强探索能力。

另一个例子是（Gangwani和Peng，2018）所述的GPO模型中的基于梯度的探索。如文中所示，该模型利用梯度多样性来增强探索，但它并非使用单一的KL散度度量来触发修正，而是进行持续调整。这些调整通过基于梯度的正则化实现。（Plappert等人，2018）也研究了这种持续扰动模型的方法，他们在训练过程中向神经网络参数注入持续的噪声流。有趣的是，这些噪声采样自一个学习得到的分布。

（Eysenbach等人，2018）从不同的角度看待模型合并，将近端策略优化纳入现有的进化方法中，这与我们的方向相反。他们的工作实质上是将近端策略优化作为进化模型中的一个子程序，该模型专注于演化多种多样的策略，而不使用任何外在奖励。他们的方法旨在产生广泛的进化行为。相比之下，我们的方法致力于为奖励驱动的任务优化单一策略。

3 方法

我们的 POEM 算法详见算法 1，它通过整合一种自适应探索机制扩展了标准近端策略优化，该机制利用过去策略参数的移动平均。与仅依赖裁剪代理目标和熵正则化的近端策略优化不同，POEM 会监测当前策略与其移动平均之间的 KL散度，以量化策略新颖性。当这一散度低于预设阈值时，就会触发进化变异，即自适应地向策略参数注入噪声，以摆脱局部最优并鼓励更广泛的探索。这一额外机制使 POEM 有别于传统的近端策略优化，为克服策略停滞、实现更稳健的性能提供了一种系统化方法。

为了在公平的条件下充分评估我们算法的性能，并与近端策略优化进行比较，我们在四种 OpenAI Gym 环境中进行了基准测试：CarRacing（v0）、MountainCar（v0）、BipedalWalker（v3）和月球着陆器（v2）。

基于 Box2D 物理引擎的 CarRacing 环境使用以车辆为中心的 96×96 像素 RGB观测。我们将帧转换为灰度图，并在每个步骤堆叠四帧，以便进行速度估计。动作空间为连续型，包含三种动作：转向、加速（油门）和制动。我们采用了以下奖励函数

$$\text{Reward} = -0.1 \times (\text{Number of frames}) + \frac{1000}{\text{Number of track tiles visited}},$$

该函数会惩罚较慢的完赛速度，并奖励高效的赛道覆盖率。

MountainCar 和 BipedalWalker 使用默认设置和奖励结构：MountainCar 的每步稀疏惩罚（到达旗帜前每个时间步惩罚 –1），以及 BipedalWalker 的连续前进进度奖励和能耗奖励。对于月球着陆器，我们修改了默认环境，引入燃料机制，确保每个回合都以安全着陆或坠毁结束。燃料系统的初始总量为 600 单位，主引擎每个时间步消耗 3 单位燃料，而每个侧引擎每个时间步消耗 1 单位燃料。这项修改促使智能体在每次着陆尝试过程中进行资源感知决策。

强化学习近端策略优化模型基于开源 Stable-Baselines3 (SB3) 框架构建，并通过 PyTorch 库进行修改，以实现我们的自适应探索机制。如算法 1 所示，POEM 在标准近端策略优化更新中加入了自适应的基于KL散度的变异步骤。我们对策略参数维护一个指数移动平均

$$\hat{\theta} \;\leftarrow\; \beta\,\hat{\theta} + (1-\beta)\,\theta,$$

其中，θ 是当前策略权重，$\beta \in [0,1]$ 控制平滑程度。对于每个小批量，我们首先计算常规的近端策略优化裁剪代理损失 $L_{\text{PPO}}(\theta)$，以及价值函数损失$L_{\text{VF}}(\theta)$和熵奖励 $H(\pi_\theta)$。然后，我们构造总损失

$$L_{\text{total}}(\theta) = L_{\text{PPO}}(\theta) - \lambda_{\text{div}}\, D_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\hat{\theta}}\big) + \alpha_{\text{vf}}\, L_{\text{VF}}(\theta) - \alpha_{\text{ent}}\, H(\pi_\theta),$$

其中，$D_{\text{KL}}(\pi\|\pi')$是小批量上的平均KL散度，λ_{div} 用于加权多样性奖励，而 $\alpha_{\text{vf}}, \alpha_{\text{ent}}$ 是从近端策略优化继承的价值系数和熵系数。应用梯度步后，我们重新计算

$$d_{\text{post}} \;=\; D_{\text{KL}}\big(\pi_{\theta_{\text{new}}} \| \pi_{\hat{\theta}}\big).$$

如果 $d_{\text{post}} < \delta$，我们设定

$$\sigma \;=\; \sigma_{\min} + (\sigma_{\max} - \sigma_{\min})\,\frac{\delta - d_{\text{post}}}{\delta},$$

其中，δ 是散度阈值，σ 是自适应插值后的标准差，并将其限制在 $[\sigma_{\min}, \sigma_{\max}]$，然后采样一个扰动候选解 $\theta' = \theta_{\text{new}} + \mathcal{N}(0,\ \sigma^2 I)$。随后，我们评估 $L_{\text{total}}(\theta')$，并且仅当 $L_{\text{total}}(\theta') < L_{\text{total}}(\theta_{\text{new}})$时替换 $\theta_{\text{new}} \leftarrow \theta'$。这一变异步骤（算法 1 的第 10–17 行）使 POEM 能够在标准近端策略优化更新变得过于保守时逃离局部最优。

算法 1 结合进化变异的近端策略优化

初始化策略参数 θ
初始化策略参数的方差 σ
设置KL散度阈值 δ
设置最小和最大变异方差 σ 最小值 $\sigma_{\max}$
当未收敛 **时**
    使用策略 π_θ采集动作
    计算优势 $\hat{A}_t$ 和未来累积奖励 $\hat{R}_t$
    更新移动平均策略 $\hat{}\pi_\theta$
    计算 KL散度:

$$D_{\mathrm{KL}} = \frac{1}{N}\sum_{i=1}^{N} \log \frac{\pi_\theta(a_i \mid s_i)}{\hat{\pi}_\theta(a_i \mid s_i)}$$

    如果 $D_{\mathrm{KL}} < \delta$ **那么**
        $\sigma \leftarrow \sigma_{\min} + (\sigma_{\max} - \sigma_{\min}) \times \dfrac{\delta - D_{\mathrm{KL}}}{\delta}$
        生成候选变异:

$$\theta'(j) = \theta + \mathcal{N}(0, \sigma^2)$$

        评估组合目标函数:

$$L_{\mathrm{total}}(\theta') = L_{\mathrm{PPO}}(\theta') - \lambda_{\mathrm{div}}\, D_{\mathrm{KL}}\big(\pi_{\theta'} \| \pi_{\hat{\theta}}\big)$$

        选择最佳候选变异:
        如果 $L_{\mathrm{total}}(\theta'(j)) < L_{\mathrm{total}}(\theta)$ **那么**
            $\theta \leftarrow \theta'(j)$
        结束条件判断
        更新策略参数 $\theta \leftarrow \theta^*$
    else
        对 θ执行标准近端策略优化更新
    结束条件判断
结束循环

最终训练阶段之前使用 Optuna 进行了超参数调参。超参数的取值范围基于原始近端策略优化算法的最优值确定，允许偏差最多为 $\pm 10\%$。Optuna 利用树结构 Parzen 估计器（TPE）算法，无需穷举网格搜索即可高效确定最优超参数。POEM 和近端策略优化均采用相同的方法，在所有环境中进行超参数调参。对于 CarRacing、BipedalWalker 和月球着陆器，每组超参数组合均通过一次包含 100,000 个时间步的短时训练运行进行评估，随后进行 4 回合评估；对于 MountainCar，则采用较短的 50,000 个时间步训练运行，随后进行 4 回合评估。所有运行均使用固定的随机种子并采用确定性执行，以确保可复现性。对于每个环境，选择平均评估奖励最高的超参数集用于最终训练。

针对每个环境确定最佳超参数后，最终训练阶段分别在 MountainCar、月球着陆器、CarRacing 和 BipedalWalker 上运行了 150,000、250,000、500,000 和 1,500,000 个时间步。训练完成后，使用随机种子对每个模型进行 15 个确定性回合的评估，并记录和计算总奖励的平均值。为确保可复现性，我们保存了最终模型检查点和完整的评估日志，其中包括以 CSV 文件格式整理的奖励。

在调参和最终训练阶段，TensorBoard 日志都提供了实时监控，并生成性能图，以直观呈现训练动态和评估结果。这种系统且可复现的方法使我们能够在多种控制任务中公平比较 POEM 与近端策略优化。

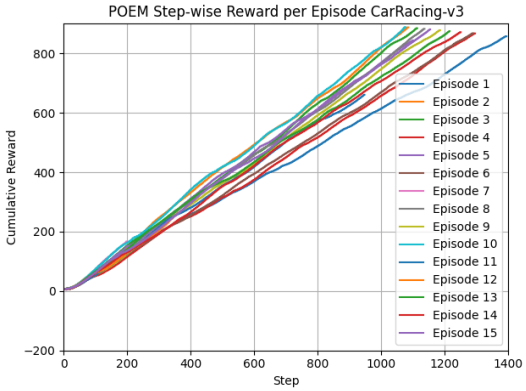
4 实验结果

我们在四个 OpenAI Gym 环境（CarRacing、MountainCar、BipedalWalker 和月球着陆器）中，分别对近端策略优化和 POEM 进行了 10 次实验。每次运行均以固定随机种子初始化，并以确定性方式执行，以确保可复现性。表 1 展示了 Welch 双样本 t 检验对比 POEM 与近端策略优化在各环境中表现的结果。与近端策略优化相比，POEM 在 BipedalWalker（$t = -2.0642$, $p = 0.0495$）、CarRacing（$t = -6.3987$, $p = 0.0002$）和 MountainCar（$t = -6.2431$, $p < 0.0001$）上取得了统计显著的改进，而在月球着陆器（$t = -1.8707$, $p = 0.0778$）上的差异则不显著。

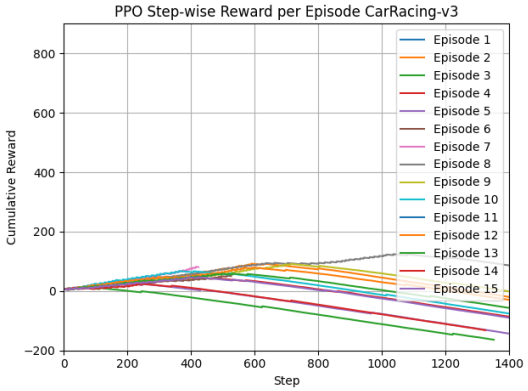
表 1：近端策略优化与 POEM 的 t 检验结果汇总

检验名称	t 统计量	p 值	POEM 是否显著更优?
双足行走者	-2.0642	0.0495	Yes
CarRacing	-6.3987	0.0002	Yes
山地车	-6.2431	< 0.0001	Yes
月球着陆器	-1.8707	0.0778	No

图 1–4 展示了近端策略优化和 POEM 在四个测试环境中的逐步累积奖励。每张图均报告了 15 个固定随机种子评估回合的结果。在所有环境中，POEM 累积奖励的速度更快，最终回报也始终高于近端策略优化。

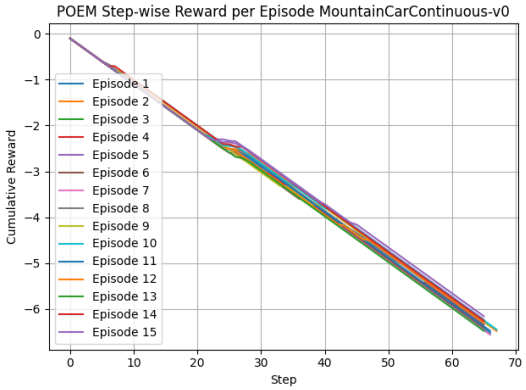


(a) POEM CarRacing-v3

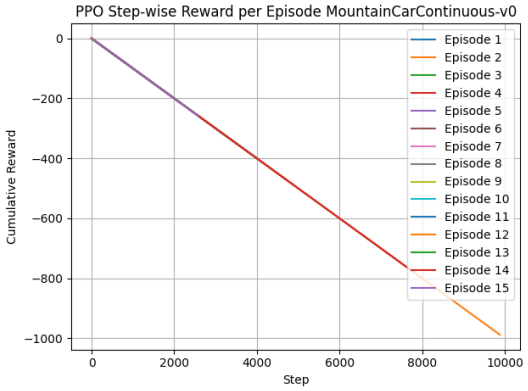


(b) 近端策略优化 CarRacing-v3

图 1：POEM 和近端策略优化在 15个评估回合中，于 CarRacing-v3 上每步的累计奖励。**(a) POEM：**在不同随机种子下的方差较低，能够稳定地沿赛道前进，并完成除四次以外的所有运行。**(b) 近端策略优化：**经常在转弯处陷入停滞，无法完成任何回合。

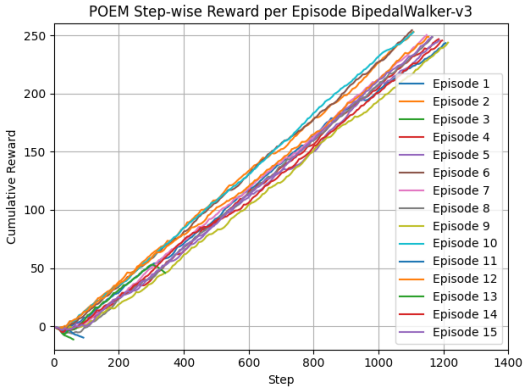


(a) POEM MountainCarContinuous-v0

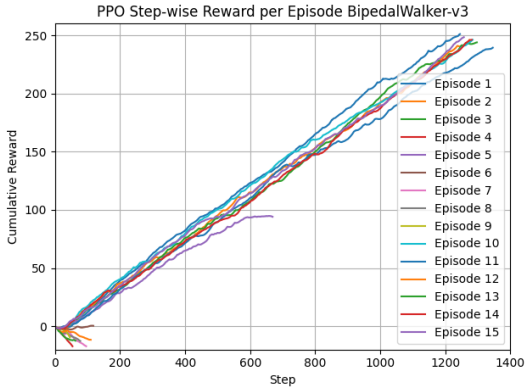


(b) 近端策略优化 MountainCarContinuous-v0

图 2：POEM 和近端策略优化在 15个评估回合中，于 MountainCarContinuous-v0 上每步的累计奖励。**(a) POEM：**成功探索环境，驶出山谷，并始终能够到达目标，同时只产生极少的负奖励。**(b) 近端策略优化：**在所有回合中均未能完成任务，达到时间步上限，或驶出赛道边界，并产生完整的 1000 步累计成本。请注意，两张图的 Y 轴刻度不同；为便于阅读，图中未显示完成任务可获得的最终奖励 $+100$ 。

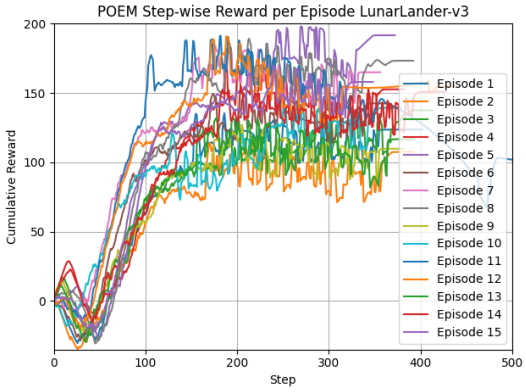


(a) POEM

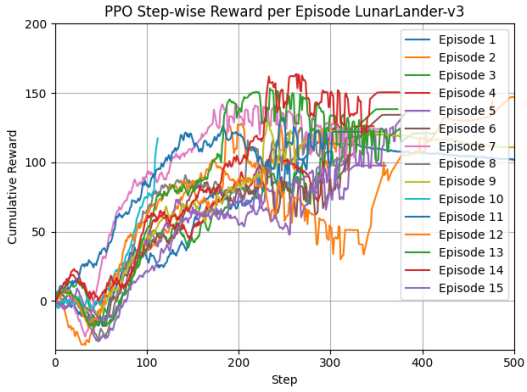


(b) 近端策略优化

图 3：在 BipedalWalker-v3 上，POEM 和近端策略优化在 15个评估回合中的每步累积奖励。**(a) POEM：**奖励轨迹在不同随机种子间的方差较低，表明性能稳定。POEM 能够稳定地沿赛道前进，除三个回合外，其余回合均完成了赛道，且所有成功回合均在 1200 步内完成。**(b) 近端策略优化：**仅在 15 个回合中的 7 个回合完成赛道，其余回合均以失败告终。近端策略优化的所有运行中，没有一次能在少于 1200 步的情况下完成赛道。



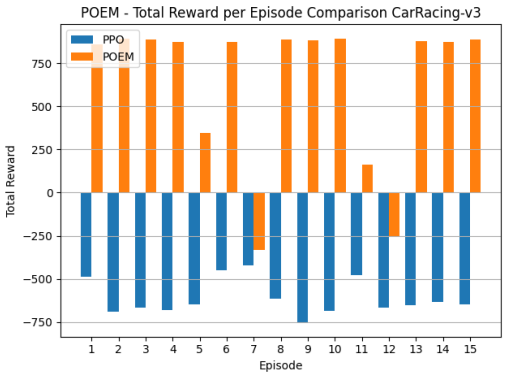
(a) POEM



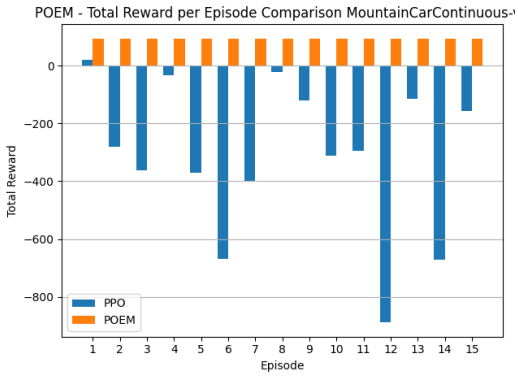
(b) 近端策略优化

图 4：在 LunarLander-v3 上，POEM 和近端策略优化在 15个评估回合中的每步累积奖励。**(a) POEM：**智能体在前 100 步中迅速提高奖励，同时稳定下降过程；随后奖励趋于平稳，着陆器会微调位置，然后执行受控着陆。**(b) 近端策略优化：**奖励增长较为缓慢，反映出一种策略：先在较高海拔寻找着陆区域，再下降并尝试执行最终着陆操作。

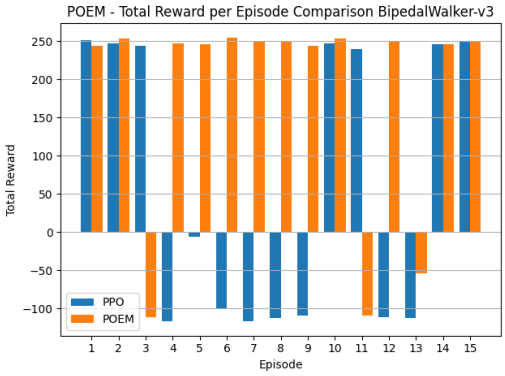
图5展示了 10 次确定性随机种子运行中每次获得的总奖励（蓝色 =近端策略优化，橙色 = POEM）。POEM 在几乎每个回合中的表现都优于近端策略优化；近端策略优化仅在 BipedalWalker 中有一次勉强胜过 POEM，在月球着陆器中有四次。



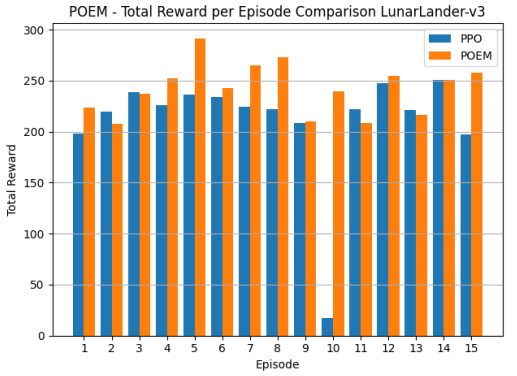
(a) CarRacing



(b) MountainCar



(c) BipedalWalker



(d) 月球着陆器

图5：近端策略优化（蓝色）和 POEM（橙色）在 15 次运行中各回合的总奖励。POEM 在绝大多数运行中胜出；近端策略优化仅在 BipedalWalker 的 1 个回合和月球着陆器的 4 个回合中超过 POEM。

表2：15个评估回合中各环境的平均智能体奖励

环境	POEM	PPO
CarRacing-v3	640.01	-610.83
MountainCarContinuous-v0	93.52	-311.75
BipedalWalker-v3	180.58	62.43
LunarLander-v3	242.10	210.94

5 讨论

在四个 OpenAI Gym 环境——CarRacing、MountainCar、BipedalWalker 和月球着陆器——中的实验结果表明，在十次使用确定性随机种子进行的运行中，POEM 在四项任务中的三项上优于近端策略优化。使用显著性阈值为 $\alpha = 0.05$ 的 Welch t 检验，我们发现 POEM 在 BipedalWalker（$t = -2.0642$, $p = 0.0495$）、CarRacing（$t = -6.3987$, $p = 0.0002$）和 MountainCar（$t = -6.2431$, $p < 0.0001$）上取得了显著提升，而在月球着陆器（$t = -1.8707$, $p = 0.0778$）上的差异未达到显著水平。累积奖励曲线（图1 - 4）和回合奖励柱状图（图5）进一步展示了 POEM 一贯的优越表现；近端策略优化仅在 BipedalWalker 中有一次勉强超过 POEM，在月球着陆器中有四次。

CarRacing-v3. 在 CarRacing 中，两种方法之间的性能差距最为明显。如图1a所示，POEM 在每个回合早期便迅速累积奖励，并在整个过程中保持稳定的控制；而 PPO 则反复陷入低奖励轨迹（图1b）。较大的 t 统计量和极小的 p值进一步凸显了 POEM 的显著优势，这得益于它能够避免回合早期不稳定性。

MountainCarContinuous-v0. 在 MountainCar 中，POEM 同样更具优势。PPO 经常无法积累足够的动量以逃离山谷并获得稀疏奖励，往往会停滞并超时；或者花费很长时间才积累起足够的动量逃出山谷，最终承受大量负奖励，如图2b所示。POEM 则能可靠地到达目标并获得显著更高的回报；如图2a所示，它在回合中产生的负奖励较少，且到达目标前所需步数有限。统计结果（$p < 0.0001$）以及表2中的平均奖励均反映了这种稳定的行为优势。

BipedalWalker-v3。 在 BipedalWalker 中，两种方法最终都学会了可行的移动方式，但 POEM 生成的轨迹更平滑、更稳定，因此平均回报和成功率更高。PPO 穿越地形的表现不稳定，在 15 个回合中有 8 个回合发生了碰撞。即使在 PPO 成功的回合中，其完成时间仍长于 POEM 表现最差的一次运行，如对比图 3a 和图 3b 所示。

LunarLander-v3. LunarLander呈现出更为微妙的对比。统计显著性不足，可能源于该环境中较高的奖励方差，或超参数敏感性仍未完全消除。在评估图 4a 中的逐步奖励时，值得注意的是，POEM采用了不同于近端策略优化的策略。POEM倾向于在早期快速下降，迅速获得奖励，随后通过微调着陆过程以获取终端奖励，奖励增长逐渐趋于平缓。相比之下，近端策略优化的奖励逐步增加，它在高空盘旋，寻找最佳着陆点，然后才开始下降。然而，如图 4b 所示，近端策略优化的盘旋策略导致了一次灾难性失败，整体表现也逊于 POEM。这反映出一种内在风险：在高空盘旋可能导致燃料耗尽并坠毁。POEM采取早期下降、临近地面时再进行修正的策略，因此燃料耗尽不太可能导致彻底失败，同时也带来了更高的平均奖励。

总体而言，这些发现表明，POEM的自适应变异机制能够有效地在多种控制任务中平衡探索与利用，从而在稀疏奖励和密集奖励环境中提升稳定性与性能。

6 结论

在本研究中，我们提出了 POEM，这是一种近端策略优化变体；当当前策略与移动平均策略之间的 KL散度低于阈值时，该方法会注入进化变异。在四个 OpenAI Gym 控制任务上进行了 15 次确定性运行，结果显示，POEM 的回报始终高于标准 PPO；在 CarRacing、MountainCar 和 BipedalWalker 上取得了统计显著的提升，在月球着陆器上则与 PPO 表现相当。这些结果证实了我们的假设：自适应的多样性驱动的变异步骤能够带来额外的探索，同时不牺牲 PPO 标志性的稳定性和样本效率。

未来的工作包括在具有连续和离散动作空间的环境中明确评估 POEM 的性能，以确定其优势是否取决于动作空间。此外，将 POEM 扩展到更复杂、更高维的环境中，或可深入揭示其在现实世界应用中的可扩展性和鲁棒性。最后，尽管 PPO 仍被广泛使用，未来的实验应将我们的改进与其他当代 PPO 变体进行基准比较。一个例子是结合内在好奇心模块的 PPO （ Wu 等，2024 ）。

参考文献

Eysenbach, B., Gupta, A., Ibarz, J. 和 Levine, S.（2018）。多样性就是你所需要的一切：无需奖励函数的技能学习。Gangwani, T. 和 Peng, J.（2018）。通过遗传蒸馏优化策略。Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P. 和 Andrychowicz, M.（2018）。用于探索的参数空间噪声。Wu, C., Yu, W., Liao, W. 和 Ou, Y.（2024）。基于内在好奇心模块的无人水面艇深度强化学习轨迹跟踪控制。《海洋工程》, 308:118342。Wu, S., Yao, J., Fu, H., Tian, Y., Qian, C., Yang, Y., FU, Q. 和 Wei, Y.（2023）。通过基于种群的强化学习实现质量相近的多样性。载于第十一届国际学习表征会议。