



南开大学
Nankai University

V-MPO:

ON-POLICY MAXIMUM A POSTERIORI POLICY OPTIMIZATION
FOR DISCRETE AND CONTINUOUS CONTROL

组员:

颜铭
毛荣贞
姚文君
杨宪

目 录

CONTENTS

- 一. 研究背景
- 二. 研究内容
- 三. 论文实现细节分析
- 四. 思考与展望


01

研究背景



摘要:

本文介绍了一种在线适应的最大后验策略优化（V-MPO）算法，作为策略梯度算法的替代方案。该算法在多任务设置下超过了以前报告的基准套件，并且可以可靠地实现，同时避免了策略崩溃等问题。V-MPO适用于具有高维连续动作空间的问题，并已经在多个任务中得到应用和验证，可实现比以前报告的更高的分数。算法的成功应用意味着政策梯度算法在处理一些高挑战控制领域时可以被替代或增强。



V-MPO: On-Policy Maximum a Posteriori Policy Optimization for Discrete and Continuous Control

1 code implementation • ICLR 2020

Some of the most successful applications of deep reinforcement learning to challenging domains in discrete and continuous control have used policy gradient methods in the on-policy setting.

Continuous Control

OpenAI Gym

+1

★ 37

Paper

Code



研究背景介绍：

深度强化学习（RL）的算法在多种机器人应用领域中的超人类表现。在这些领域中，深度RL已被应用于困难环境中，如Dota 2、夺旗战（Capture the Flag）、星际争霸II和灵巧物体操纵等。而在此应用中最成功的RL算法技术包括策略梯度法和近似策略迭代法等，尤其以Proximal Policy Optimization（PPO）和Importance-Weighted Actor-Learner Architecture（IMPALA）最为常用。其中，策略梯度法常常受到方差变化的影响，特别是在高维度的动作空间时，此时往往采用熵正则化技术进行调整。与之不同，本文研究提出了一种近似策略迭代算法V-MPO，该算法主要在于使用了一个学习的状态价值函数 $V(s)$ ，而不再使用状态动作价值函数，这可以有效地解决策略梯度法中的方差问题。这一算法的应用也涵盖了离散行为和连续行为的RL学习问题。在理论和实践中，V-MPO已有较为突出的表现，并在多任务应用中取得了比以前报道的更好的表现。

背景介绍:

值得探索的研究意义在于折扣强化学习（RL）的发展前景，其中我们寻求优化策略 π ，这个过程用马尔可夫决策过程（MDP）来描述，包括状态 s 、行动 a 、初始状态分布、过渡概率 $P^{\text{env}}(s_{t+1}|s_t, a_t)$ 、奖赏函数 $r(S_t, A_t)$ 和折扣因子 $\gamma \in (0, 1)$ 。在深度RL中，策略 $\pi_{\theta}(a_t|s_t)$ 由具有参数 θ 的神经网络描述。我们考虑问题中 s 和 a 可能是离散或连续的。RL中有两个核心函数：状态值函数 $V_{\pi}(s_t)$ 和状态行动值函数 $Q_{\pi}(s_t, a_t)$ 。在通常的RL问题中。

目标是找到最大化预期回报的策略 π ，即 $J(\pi)$ ，其中 $\pi(s) = \arg\max_a [Q^{\pi}(s, a) - V^{\pi}(s)]$ ，如果至少有一个状态行动对具有正优势且访问该状态的概率为非零，则精确的策略迭代可以改善策略，基于这个结果，我们可以选择优势函数技巧 $A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s)$ 。

本文所述的新算法V-MPO分享了很多相似之处和相关工作与原始的MPO算法。然而，V-MPO使用EM算法通过适当的约束复杂度，能够可靠地训练功能强大的神经网络。



相关工作：

与原始的MPO算法相比，V-MPO使用KL约束来限制策略更新的大小，这个思想在TRPO和PPO中也存在，但这对应于V-MPO中的E-step约束。与此同时，M-step KL约束的引入以及使用top-k优势使V-MPO与相对熵策略搜索（REPS）不同。先前尝试使用神经网络函数逼近器的REPS报告了非常差的性能，特别是容易陷入局部极值。此外，与V-MPO类似的Supervised Policy Update（SPU）旨在实现一个优化问题并将参数策略拟合到这个解决方案中。但我们认为SPU与V-MPO的使用方式有很大不同，因此最终算法更接近于PPO等策略梯度算法。

03

算法实现细节

A. DERIVATION OF THE V-MPO TEMPERATURE LOSS

V-MPO温度损失的推导

- 寻求 $\psi(s, a)$, 使其最小化:

$$\mathcal{J}(\psi(s, a)) = D_{\text{KL}}(\psi(s, a) \| p_{\theta_{\text{old}}}(s, a | \mathcal{I} = 1)) \quad (1)$$

$$\propto - \sum_{s, a} \psi(s, a) A^{\pi_{\theta_{\text{old}}}}(s, a) + \eta \sum_{s, a} \psi(s, a) \log \frac{\psi(s, a)}{p_{\theta_{\text{old}}}(s, a)} + \lambda \sum_{s, a} \psi(s, a) \quad (2)$$

- 希望自动调优 η , 因此考虑约束优化问题:

$$\psi(s, a) = \arg \max_{\psi(s, a)} \sum_{s, a} \psi(s, a) A^{\pi_{\theta_{\text{old}}}}(s, a) \quad (3)$$

$$\text{s.t. } \sum_{s, a} \psi(s, a) \log \frac{\psi(s, a)}{p_{\theta_{\text{old}}}(s, a)} < \epsilon_{\eta} \text{ and } \sum_{s, a} \psi(s, a) = 1. \quad (4)$$

- 使用拉格朗日松弛将约束优化问题转化为一个最大化无约束目标的问题 ($\eta \geq 0$):

$$\mathcal{J}(\psi(s, a), \eta, \lambda) = \sum_{s, a} \psi(s, a) A^{\pi_{\theta_{\text{old}}}}(s, a) + \eta \left(\epsilon_{\eta} - \sum_{s, a} \psi(s, a) \log \frac{\psi(s, a)}{p_{\theta_{\text{old}}}(s, a)} \right) + \lambda \left(1 - \sum_{s, a} \psi(s, a) \right) \quad (5)$$

A. DERIVATION OF THE V-MPO TEMPERATURE LOSS

V-MPO温度损失的推导

- 求 \mathcal{J} 关于 $\psi(s, a)$ 的微分并使其等于 0 得到:

$$\psi(s, a) = p_{\theta_{\text{old}}}(s, a) \exp\left(\frac{A^{\pi_{\theta_{\text{old}}}}(s, a)}{\eta}\right) \exp\left(-1 - \frac{\lambda}{\eta}\right). \quad (6)$$

- 归一化 s, a (使用 λ 给出的自由度) 得到:

$$\psi(s, a) = \frac{p_{\theta_{\text{old}}}(s, a) \exp\left(\frac{A^{\pi_{\theta_{\text{old}}}}(s, a)}{\eta}\right)}{\sum_{s, a} p_{\theta_{\text{old}}}(s, a) \exp\left(\frac{A^{\pi_{\theta_{\text{old}}}}(s, a)}{\eta}\right)} \quad (7)$$

- 现在可以通过优化相应的对偶函数来找到 η 的值, 将公式 7 插入公式 5 中的无约束目标, 就会产生与 η 相关的项:

$$\mathcal{L}_{\eta}(\eta) = \eta \epsilon_{\eta} + \eta \log \left[\sum_{s, a} p_{\theta_{\text{old}}}(s, a) \exp\left(\frac{A^{\pi_{\theta_{\text{old}}}}(s, a)}{\eta}\right) \right] \quad (8)$$

B. DECOUPLED KL CONSTRAINTS FOR CONTINUOUS CONTROL

针对连续控制的
解耦KL约束

- 对于以高斯分布为参数的连续动作空间，在M步骤中使用解耦 KL 约束。
两个均值为 μ_1, μ_2 ，协方差 Σ_1, Σ_2 的 d 维多变量正态分布之间的 KL 散度可以写为：

$$D_{\text{KL}}(\mathcal{N}(\mu_1, \Sigma_1) \parallel \mathcal{N}(\mu_2, \Sigma_2)) = \frac{1}{2} \left[(\mu_2 - \mu_1)^T \Sigma_1^{-1} (\mu_2 - \mu_1) + \text{Tr}(\Sigma_2^{-1} \Sigma_1) - d + \log \frac{|\Sigma_2|}{|\Sigma_1|} \right] \quad (9)$$

- 可以把整个 KL 散度分成一个平均分量和一个协方差分量：

$$D_{\text{KL}}^{\mu}(\pi_{\theta_{\text{old}}} \parallel \pi_{\theta}) = \frac{1}{2} (\mu_{\theta} - \mu_{\theta_{\text{old}}})^T \Sigma_{\theta_{\text{old}}}^{-1} (\mu_{\theta} - \mu_{\theta_{\text{old}}}) \quad (10)$$

$$D_{\text{KL}}^{\Sigma}(\pi_{\theta_{\text{old}}} \parallel \pi_{\theta}) = \frac{1}{2} \left[\text{Tr}(\Sigma_{\theta}^{-1} \Sigma_{\theta_{\text{old}}}) - d + \log \frac{|\Sigma_{\theta}|}{|\Sigma_{\theta_{\text{old}}}|} \right] \quad (11)$$

- 对约束损失公式 (*) 进行替换 $D_{\text{KL}}(\pi_{\theta_{\text{old}}} \parallel \pi_{\theta}) \rightarrow D_{\text{KL}}^C(\pi_{\theta_{\text{old}}} \parallel \pi_{\theta})$ $C = \mu, \Sigma$ $\alpha \rightarrow \{\alpha_{\mu}, \alpha_{\Sigma}\}$
地，可以得到总损失（公式12）：

$$\mathcal{L}_{\alpha}(\theta, \alpha) = \alpha \left(\epsilon_{\alpha} - \mathbb{E}_{s \sim p(s)} \left[\text{sg} \left[[D_{\text{KL}}(\pi_{\theta_{\text{old}}} \parallel \pi_{\theta})] \right] \right] \right) + \text{sg}[[\alpha]] \mathbb{E}_{s \sim p(s)} \left[D_{\text{KL}}(\pi_{\theta_{\text{old}}} \parallel \pi_{\theta}) \right] \quad (*)$$

$$\mathcal{L}_{\text{V-MPO}}(\theta, \eta, \alpha_{\mu}, \alpha_{\Sigma}) = \mathcal{L}_{\pi}(\theta) + \mathcal{L}_{\eta}(\eta) + \mathcal{L}_{\alpha_{\mu}}(\theta, \alpha_{\mu}) + \mathcal{L}_{\alpha_{\Sigma}}(\theta, \alpha_{\Sigma}) \quad (12)$$

C. IMPORTANCE-WEIGHTING FOR OFF-POLICY CORRECTIONS

- 在常见的分布式、异步实现中，生成数据的网络可能会落后于目标网络，可以通过将指数化的优势乘以重要性权重 $\rho(s, a)$ 来补偿这一点：

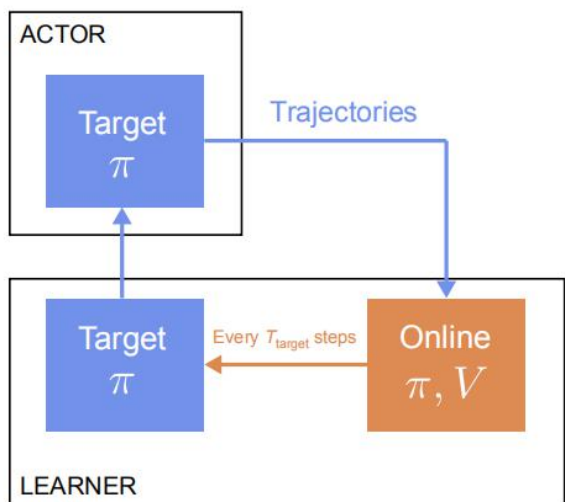
$$\psi(s, a) = \frac{\rho(s, a) p_{\theta_{\mathcal{D}}}(s, a) \exp\left(\frac{A^{\pi_{\theta_{\mathcal{D}}}}(s, a)}{\eta}\right)}{\sum_{s, a} \rho(s, a) p_{\theta_{\mathcal{D}}}(s, a) \exp\left(\frac{A^{\pi_{\theta_{\mathcal{D}}}}(s, a)}{\eta}\right)}$$

$$\mathcal{L}_{\eta}(\eta) = \eta \epsilon_{\eta} + \eta \log \left[\sum_{s, a} \rho(s, a) p_{\theta_{\mathcal{D}}}(s, a) \exp\left(\frac{A^{\pi_{\theta_{\mathcal{D}}}}(s, a)}{\eta}\right) \right]$$

$$\rho(s, a) = \min \left(1, \frac{\pi_{\theta_{\text{old}}}(a|s)}{\pi_{\theta_{\mathcal{D}}}(a|s)} \right)$$

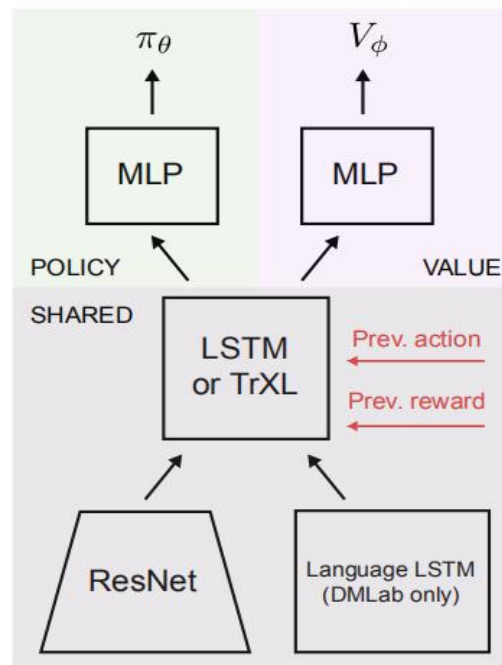
为简单起见，本论文工作中提出的实验没有使用重要性加权。

D. NETWORK ARCHITECTURE AND HYPERPARAMETERS



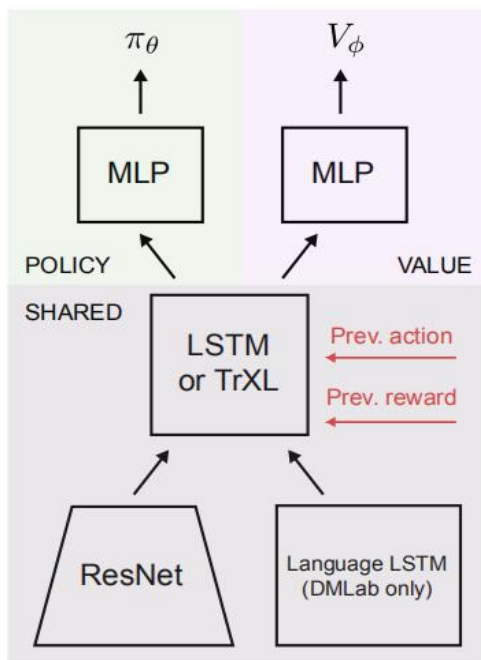
图a 带有目标网络的actor-learner架构

目标网络用于生成环境中的行为体经验，每隔 T_{target} 学习步骤从在线网络中更新



图b V-MPO agent示意图

D. NETWORK ARCHITECTURE AND HYPERPARAMETERS



图b V-MPO agent示意图

策略policy (θ) 和价值 value(ϕ) 网络通过一个共享的输入编码器和 LSTM [或Transformer-XL(TrXL), 单一Atari水平] 共享其大部分参数, agent还接收上一步的行动和奖励作为LSTM的输入。对于DMLab, 还会有一个额外的LSTM用来处理简单的语言指令。

在所有情况下, 策略对数 the policy logits (对离散动作) 和高斯分布参数 Gaussian distribution parameters (对连续动作) 都是由256个单元的MLP接着线性读出组成的, 对于价值函数也是如此。

HYPERPARAMETER	VALUE		
	DMLab	Atari	Continuous control
Initial η	1.0	1.0	1.0
Initial α	5.0	5.0	-
Initial α_μ	-	-	1.0
Initial α_Σ	-	-	1.0

表1: 常用的V-MPO参数的值

SETTING	SINGLE-TASK	MULTI-TASK
Agent discount		0.99
Image height		72
Image width		96
Number of action repeats		4
Number of LSTM layers	2	3
Pixel-control cost		2×10^{-3}
T_{target}		10
ϵ_η	0.1	0.5
ϵ_α (log-uniform)	[0.001, 0.01)	[0.01, 0.1)

表2: DMLab的设置

HYPERPARAMETERS

SETTING	SINGLE-TASK	MULTI-TASK
Environment discount on end of life	1	0
Agent discount	0.997	0.99
Clipped reward range	no clipping	$[-1, 1]$
Max episode length	30 mins (108,000 frames)	
Image height	84	
Image width	84	
Grayscale	True	
Number of stacked frames	4	
Number of action repeats	4	
TrXL: Key/Value size	32	.
TrXL: Number of heads	4	.
TrXL: Number of layers	8	.
TrXL: MLP size	512	.
T_{target}	1000	100
ϵ_η	2×10^{-2}	
ϵ_α (log-uniform)	[0.005, 0.01)	[0.001, 0.01)

表3: Atari的设置. TrXL: Transformer-XL

04

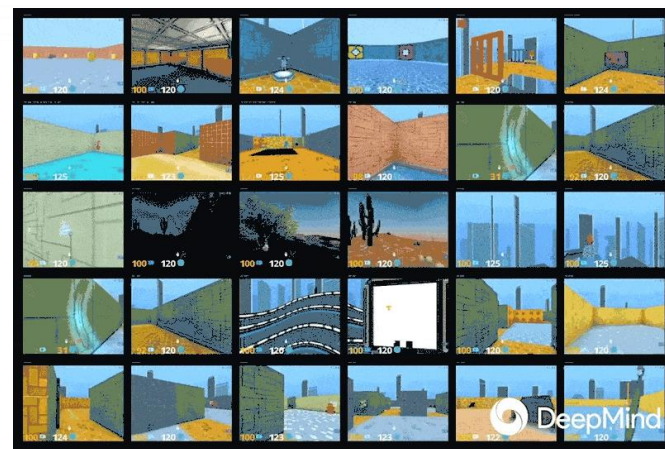
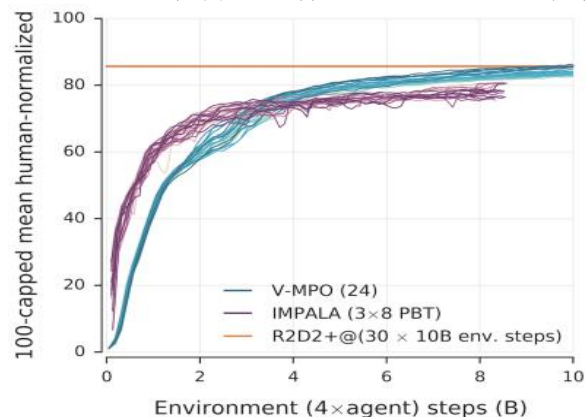
思考与展望

实验总结

离散动作游戏环境:基于DMLab

• DMLab-30环境是由deepmind团队开发的动态可视化和分布可观察的立体第一人称视角游戏环境，以模仿学习的构建思路研究三维迷宫寻路的动作-状态。

在实验超参数设置上，将V-MPO算法应用到基于视觉的深度强化学习任务上设计了像素控制损失函数，且并没有像以往一样设计乐观的奖励机制和之前PPO网络推广研究中使用的预训练改善超参数的PBT技巧。因此期望参照朴素的原始时序学习的R2D2网络(复合Q学习的基线)，对比上述包含重要性加权的异步自适应方法方法，获得更鲁棒与直接性能以研究完备环境中的泛化能力提升的轨迹

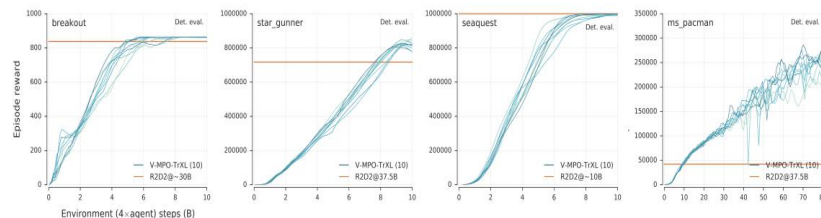
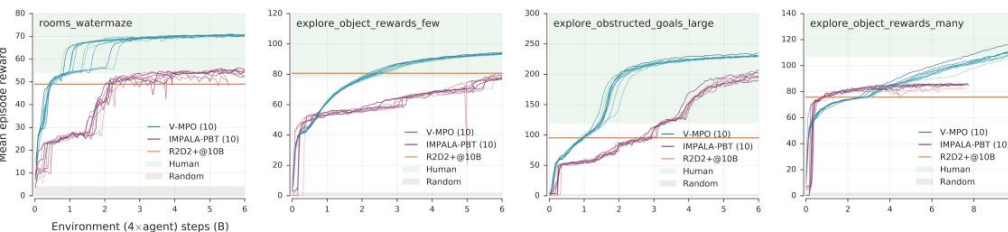


• 对标控制曲线，稳定曲线由R2D2训练的可数行为评估均值给出。从稳定性和可靠性的角度可以看出V-MPO的学习曲线相比重要性加权IMPALA方法更加平滑，而对于快速性的分析可以认为是之前的IMPALA方法对乐观奖励利用没有冗余，其敏感的兴趣导向与V-MPO引入M步的推理相反且没有发挥多余的探索作用。

实验总结

细化指标的离散动作实验—ATARI验证

• 沿用使用R2D2迁移深度Q学习训练效果基线的作为参照的思路。通过使用模仿学习关于目标和障碍物的感知例程的反馈，可以从不同的评估层级确定中研究V-MPO算法是否可以针对不同奖励结构表现较好的泛化或者适应性能。根据结果来看，在对于多任务并行的奖励系统中使用裁剪奖励因子得到了发挥原有奖励结构带来性能同时，V-MPO算法依然可以通过牺牲可控制速度性能来得到更高的阈值分数。



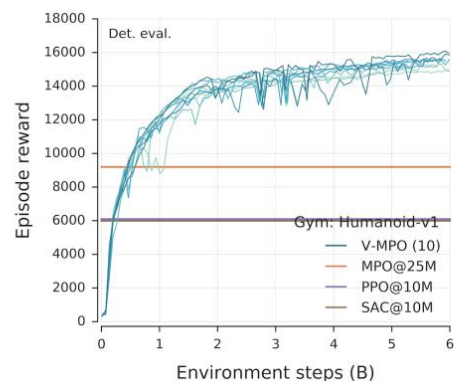
• 关于经典奖励的游戏环境，论文设计了关于雅达利的验证性实验。然后较为特别地发现V-MPO算法可以使用更少的交互次数达到超过记录分数的效果。由于雅达利不同于DMLab是完全观察的，也就意味着对于游戏仿真设计的先验模型知识更加强。论文对于代码实现中将网络核心设计为了Transformer模型而替换了Q学习那套LSTM核心(这也是DeepMind团队的工作主流)。

除此之外，放弃使用冗余的训练技巧如使用奖励裁剪和非线性值函数设计，依然达到了基线性能的要求。

思考

1. 论文提出了可以规模和集成化后端训练的深度强化学习算法V-MPO，对于推进近端策略优化的强化学习在工业仿真中的应用发挥了重要作用。
2. 在充分研究离散动作和连续动作的基础上。V-MPO的价值还体现在没有使用重要度加权和正则化熵等优化技巧，而是依靠更加扎实的EM算法数理分析，实现了更好的迁移学习效果。特别地，由于基于PP0一系沿用了使用KL散度进行网络参数规模限制的方法，可以依据EM思路可以使用相同学习率泛化到所有实验上。因此，需要纵使V-MPO算法的易用性和拓展性。
3. 一致性的超参数类别减少，取而代之的是条件性地评估裁剪奖励因子和结合相关符号量与定义式的增广的参数更新或者基于具体任务时间步长的批量化参数

ACTION	NATIVE DMLAB ACTION
Forward (FW)	[0, 0, 0, 1, 0]
Backward (BW)	[0, 0, 0, -1, 0]
Strafe left	[0, 0, -1, 0, 0]
Strafe right	[0, 0, 1, 0, 0]
Small look left (LL)	[-10, 0, 0, 0, 0]
Small look right (LR)	[10, 0, 0, 0, 0]
Large look left (LL)	[-60, 0, 0, 0, 0]
Large look right (LR)	[60, 0, 0, 0, 0]
Look down	[0, 10, 0, 0, 0]
Look up	[0, -10, 0, 0, 0]
FW + small LL	[-10, 0, 0, 1, 0]
FW + small LR	[10, 0, 0, 1, 0]
FW + large LL	[-60, 0, 0, 1, 0]
FW + large LR	[60, 0, 0, 1, 0]
Fire	[0, 0, 0, 0, 1]



展望

1. 因此，在大模型环境下，达到更好的强化学习性能不一定需要使用前面讨论过的过于复杂技巧，而是辩证地结合机器学习落地的算法成果加以创新。
2. 局限在于优化思路是同轨策略的这使得模型在在针对异轨的开发与探索上改进难度加大。
3. 事实上在相同法则下，MPO原型算法的off-policy版本并没有V-MPO算法效果更为理想。这启示业界同行更加关注策略梯度基础理论研究风向，也可以是一定手段部署的on-policy类型算法。

谢谢大家

组员分工:

- 颜铭-组织分工。负责第四部分PPT制作，研读论文。
- 杨宪-负责PPT第一部分制作，负责三四部分PPT发言。
- 毛荣贞-阅读论文并负责第三部分PPT制作。
- 姚文君-阅读论文并负责第二部分PPT制作与PPT学术化修改工作，负责一二部分PPT发言。