


策略探索能力补充材料

策略 (Policy) 是指基于状态 (State) 决定动作 (Action) 的方法。强化学习旨在寻找一个最优策略，以便在序列决策中最大化回报。根据策略函数生成动作的数学性质，策略可以分为确定性策略 (deterministic policy) 和随机性策略 (stochastic policy) 两种类型。

 更多确定性与随机性策略的细节，请参考[策略之道：探索强化学习中的随机性策略与确定性策略](#)

探索与利用的平衡 (The balance between exploration and exploitation) 是强化学习中最核心的问题之一。利用指的是根据当前已知知识或数据学习到最优动作，而探索则是指探索未知环境以获取新知识 (从而潜在地可能找到回报更大的动作)。探索和利用之间平衡的关键在于：**如何在选择探索和选择利用之间做出权衡，以便在未知世界中高效地探索，以实现回报最大化**。总体来看，探索方法大致可以分为状态空间探索技术和动作空间探索技术。本文主要关注强化学习中动作空间的探索。关于状态空间探索和利用的方法，请参考这个综述链接 [Awesome Exploration Methods in RL](#)。

此外，不同的环境类型和设定对策略探索能力的需求也有所不同。例如，在简单的确定性环境中，动作产生的影响是确定的，因此可以通过确定性规划甚至解析解来获得最优策略。然而，在现实世界任务中，环境通常具有随机性，且可能拥有庞大的状态-动作空间。这就要求智能体的策略具备更强的探索能力，以便找到最优的轨迹。

接下来，本文将分别介绍随机性策略与确定性策略算法在动作空间的探索方法。

1. 随机性策略：PPO 与 SAC 中的策略熵

在强化学习中，策略熵 (Policy Entropy) 被用作衡量策略分布随机性 (或不确定性) 的度量。策略熵的概念源于信息论中的熵 [10]，用于描述概率分布的不确定性。在策略梯度方法中，引入策略熵有助于增加探索性行为，进而可能找到更优的策略。具体来说，对于离散动作空间，给定一个策略 $\pi(a|s)$ ，其中 a 表示动作， s 表示状态，那么策略熵 $H(\pi)$ 被定义为：

$$H(\pi) = - \sum_a [\pi(a|s) * \log(\pi(a|s))]$$

策略熵的值越大，表示策略分布的不确定性越大，即采取不同动作的概率更相近。相反，策略熵的值越小，表示策略分布的确定性越大，即某些动作的概率明显高于其他动作。

对于不同的基于随机性策略的方法，对策略熵有不同的处理。具体地，PPO (Proximal Policy Optimization, 近端策略优化) [3] 通过在优化原始目标函数时，额外增加策略熵作为一个正则项，可以鼓励策略在学习前期保持一定的探索性，有助于防止过早地收敛到局部最优解，从而可能找到更好的全局最优策略。而 SAC (Soft Actor Critic) [4] 算法直接把策略熵作为策略的优化目标的一部分，

从原理上寻找一个不同的最优策略。它们使用策略熵的方式从原理上来说是不同的，从而有不同的性能表现与实践方法，下面依次介绍 PPO 和 SAC 中策略熵的使用方法。

1.1 PPO 中的熵正则化

原理

PPO 算法是 OpenAI 于2017年提出的算法，旨在解决策略梯度方法中的稳定性和数据利用率问题。核心思想是在优化策略时，限制策略更新的幅度，从而避免在策略空间中一步改变过大，导致性能下降。为了实现这个目标，PPO 引入了一个目标函数 L^{CLIP} ，它限制了新策略与旧策略之间的相对变化。这种限制通过在目标函数中引入一个比例项（即新策略与旧策略的概率比值）来实现，该比例项被限制在一个预先设定的范围内。

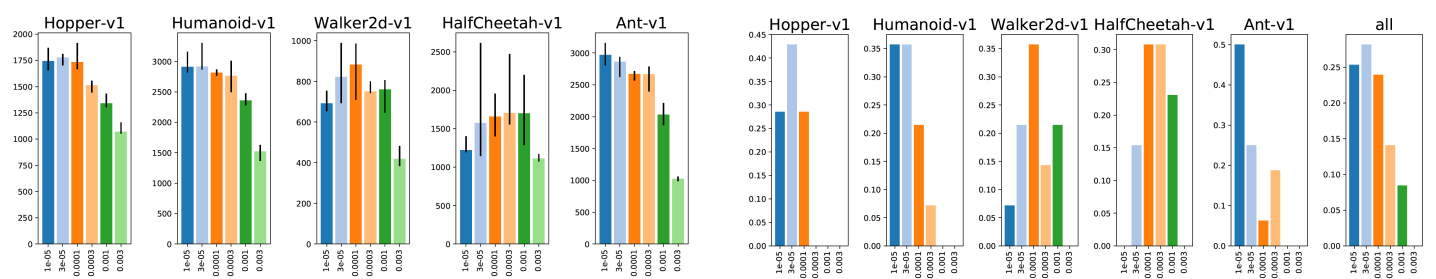
PPO 通过在原始 CLIP 后的策略损失 L^{CLIP} 和值函数损失 L^{Value} 的基础上（注：这2项的具体表达式请参考 PPO 原论文 [3]），增加一个策略熵的正则化项，来保证足够的探索，具体如下：

$$L(\theta) = \mathbb{E}_{\mathcal{D}} [L^{CLIP}(\theta) + L^{Value}(\theta) + c_2 \times H[\pi_{\theta}(s_t)]]$$

其中策略熵整体是作为一个正则项， c_2 表示策略熵系数，其值越大，则越鼓励策略保持较高的熵，从而探索能力更强。合理设置策略熵系数可以有效防止策略早熟并实现高效收敛。在 PPO 原文 [3] 中，在 Atari 环境上，默认设置是 **0.01**。但不同环境具有不同的动力学属性，其最优的策略熵系数的控制机制应该也是不同的。

实验

如图1所示，论文 [8] 给出了在 MuJoCo 的不同子环境和不同熵正则化系数下 PPO 的性能表现：



（图1：左图：展示了在不同熵正则化系数下，性能分数（performance scores）的第95百分位。这意味着，对于每个熵正则化系数，作者观察到的性能分数中有95%的分数低于此值。这有助于了解在不同熵正则化系数下，算法性能的上界。右图：展示了在最优秀的5%配置中熵正则化系数的分布情况。通过观察这个分布，可以了解在最优秀的5%性能的配置中，各个熵正则化系数的出现频率，可以看到在 MuJoCo 环境上 3×10^{-5} 是一个总体上最好的选择。）

PPO 算法在策略更新过程中能保持稳定性，并能高效利用采样数据。由于其较少的超参数易于调整，PPO 已成为许多强化学习任务的首选算法。然而，在不同的环境中，所需的探索程度是不同的，甚至在同一个环境的训练过程中，不同阶段所需的探索程度也有很大差异。例如，在训练初期，智能体通

常需要充分探索环境，尽可能访问充足的状态-动作空间。但是当智能体性能达到一定程度时，探索程度需降低，以便更好地找到最大化回报的策略，因为此时策略熵更多地起到干扰作用。

如上所述，探索程度，例如策略熵系数，是一个需要手动调整的超参数。那么，是否存在能自适应调整动作空间探索程度的算法呢？答案是肯定的。SAC 算法里面就实现了自动调整策略熵的机制。但值得注意的是，SAC 中策略熵的使用原理和更新方式与 PPO 有较大差异。下面简要介绍 SAC 算法。

1.2 SAC 中的最大熵目标

SAC 是一种结合了最大熵强化学习和 Actor-Critic 方法优势的 off-policy 算法。它的目标是学习一个随机策略，该策略不仅最大化预期回报，同时也需要最大化状态访问的熵。通过这种方法，SAC 算法鼓励策略在学习过程中进行更多的探索，从而使得学习过程更加稳定和高效。

原理

标准的强化学习 agent 最大化累计折扣奖励的期望值：

$$J(\pi) = \mathbb{E}_{s_t, a_t \sim \rho_\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$

而 SAC 算法考虑的是一个**最大熵强化学习目标**，即通过在优化目标中增加访问状态上的策略熵的期望，从而使最优策略部分偏向于随机策略：

$$J(\pi) = \mathbb{E}_{s_t, a_t \sim \rho_\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) + \alpha H(\pi(\cdot | s_t)) \right]$$

其中温度参数 α 决定了策略熵项与奖励之间的相对重要性，从而控制了最优策略的随机性。当 $\alpha \rightarrow 0$ 时，最大熵目标退化为上面的标准强化学习中的最大期望回报。

这个最大熵目标在概念上和实际应用中具有以下优势：

- 首先，策略被激励在更广泛的范围内探索，同时避免了明显表现差的轨迹。
- 其次，策略能够捕捉到近似最优行为的多个模式。在某些问题设置中，若多个动作具有相同的回报，策略将为这些动作分配相等的概率质量。
- 最后，在一些先前的研究 [5] 中观察到，使用这个目标可以显著改善探索性能。在 SAC 论文的实验中，作者也发现相较于传统的强化学习目标函数，它明显提高了学习速度。

温度系数 α 的自适应调整

自动调整熵的原理

SAC [4] 推导出了一种实用的 off-policy 算法，用于学习给定温度 α 下的最大熵策略。但是选择最佳温度并非易事，这需要为每个任务调整温度。SAC [9] 通过制定一个最大熵强化学习目标来自动化这个过程，将熵视为约束条件，而不是要求用户手动设置温度。简单地将熵固定在一个定值上并不是一个好的解决方案，因为**策略应该在最佳动作不确定的区域内更自由地探索，但在好动作和坏动作之间有明显区别的状态下更加具有确定性。**

基于此思想，SAC 论文中提出了一个约束优化问题，在该问题中，**策略的平均熵受到约束，但不同状态下的熵是可以变化**。类似的方法在先前的工作 [2] 中也被采用过，不过它里面是将当前策略与前一个策略尽可能接近作为约束。SAC 论文证明，这种约束优化的对偶形式将推导得到 soft actor critic 的更新，以及对偶变量（即温度系数）的更新。具体地，约束优化问题的目标是找到一个满足**最小期望熵的约束**而且具有**最大期望回报**的随机策略。即希望解决以下约束优化问题：

$$\max_{\pi_{0:T}} \mathbb{E}_{\rho_{\pi}} \left[\sum_{t=0}^T r(\mathbf{s}_t, \mathbf{a}_t) \right] \quad \text{s.t.} \quad \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_{\pi}} [-\log(\pi_t(\mathbf{a}_t | \mathbf{s}_t))] \geq \mathcal{H} \quad \forall t$$

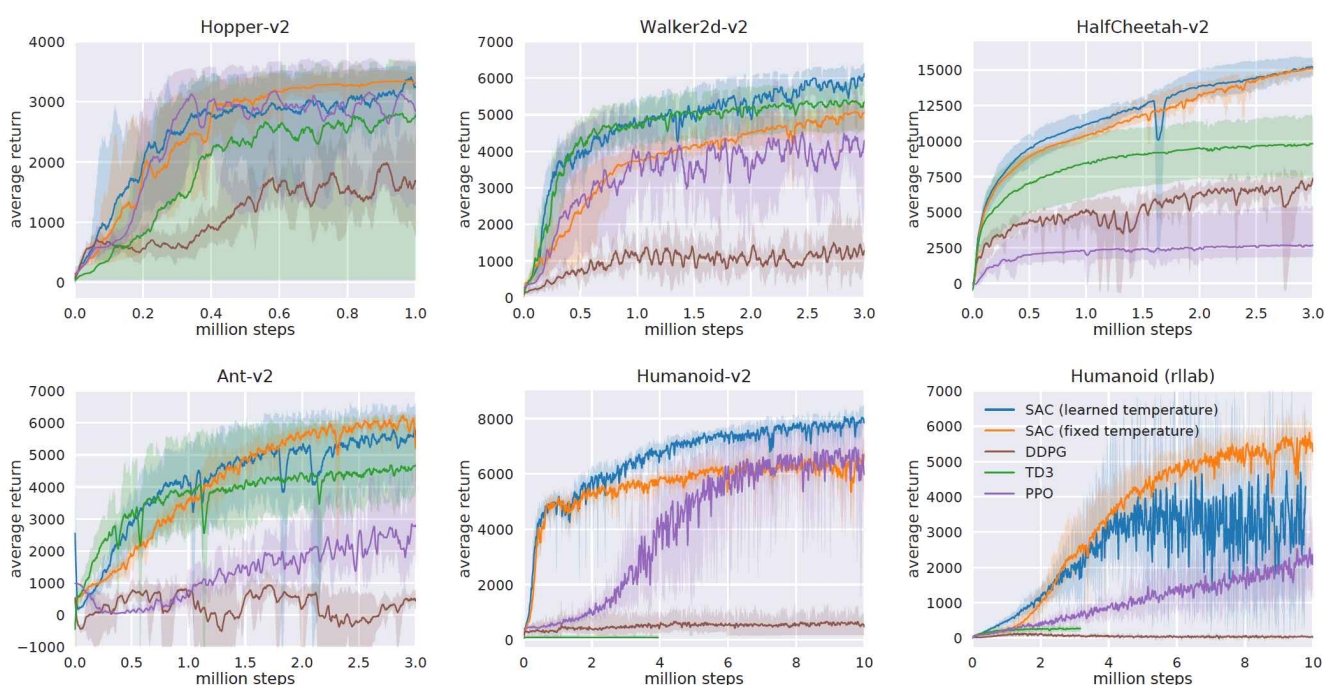
其中， \mathcal{H} 是期望的最小动作熵，在实践中，一般设置为 $-\dim(\mathcal{A})$ ，其中 \mathcal{A} 是动作空间的维度，例如对于环境 HalfCheetah-v3，其动作空间维度为6，目标的动作熵就是 -6。注意，对于 fully observed MDPs，优化期望回报得到的策略一般是确定性的，因此期望这个约束通常是紧的，不需要对熵设置上界。接着通过拉格朗日乘子法，将该约束优化问题转换为其对偶问题。具体推导请参考论文 [9] 的第5节。

实际的算法

对偶梯度下降法在下面2个步骤之间交替进行：关于原始变量优化拉格朗日函数直到收敛，关于对偶变量执行一步梯度下降。虽然在实践中针对原始变量进行优化到最优值是不切实际的，但在凸性假设下，一个进行不完全优化（甚至只执行单个梯度步）的截断版本可以证明也是收敛的。虽然这样的假设并不适用于神经网络等是非线性函数逼近器的情况，但作者发现这种方法在实践中仍然有效。因此，最终作者使用以下目标来自适应的更新温度系数 α ：

$$J(\alpha) = \mathbb{E}_{\mathbf{a}_t \sim \pi_t} [-\alpha \log \pi_t(\mathbf{a}_t | \mathbf{s}_t) - \alpha \bar{\mathcal{H}}]$$

实验



（图2：连续控制基准的训练曲线。Soft actor-critic（蓝线和橙线）在所有任务中表现一致，并且在具有挑战性的高维动作空间的任務中优于 on-policy 和 off-policy 方法。实验结果表明，自动调整温度

系数的方案在所有环境中都表现良好，从而避免了手动调整温度系数的要求。两个版本的 SAC 的区别：橙线表示针对每个环境单独调整温度参数并在训练过程中固定；蓝线表示像上文所述自动调整温度系数。）

2. 确定性策略：DDPG 与 TD3 中的动作噪声

典型的确定性策略方法，如 DDPG [1] 和 TD3 [2] 算法，都是通过在策略输出的动作上添加噪声来引入随机性，从而提高策略的探索能力。DDPG 使用的是时间相关的 Ornstein-Uhlenbeck 噪声（OU Noise），而 TD3 则采用了时间无关的高斯噪声（Gaussian Noise）。这两种噪声对策略收敛速度和最终性能有着不同的影响。关于是否存在其他可用的探索噪声以及如何选择探索噪声的问题，最近的研究 [7] 提供了部分解答。下面将对该论文的核心内容进行简要介绍。分别介绍 Colored noise 的形式化定义，以及 Pink noise 作为默认动作噪声的普适性。

2.1 Colored noise 的定义

在确定性策略算法，其学习的策略 μ 是确定性的。为了增加策略的探索能力，直接在输出的确定性动作上增加动作噪声，即

$$a_t = \mu(s_t) + \sigma \epsilon_t$$

其中， $\epsilon_{1:T} = (\epsilon_1, \dots, \epsilon_T)$ 由随机过程采样得到， σ 是一个尺度参数。如果在每个时间步都独立地采样 ϵ_t ，例如从高斯分布中采样，那么 $\epsilon_{1:T}$ 被称为白噪声（white noise, WN），这是之前 RL 算法中动作噪声的主要选择。另外一个常用的时间相关的 Ornstein-Uhlenbeck 噪声 $\epsilon_{1:T} \sim OU_T$ 。

参数化随机策略的算法，如 SAC 和 MPO [6]，也使用动作噪声。在连续动作空间中，最常见的策略分布是对角高斯分布，由函数 $\mu(s_t)$ 和 $\sigma(s_t)$ 表示： $a_t \sim N(\mu(s_t); \text{diag}(\sigma(s_t))^2)$ 。这可以等价地写成：

$$a_t = \mu(s_t) + \sigma(s_t) \odot \epsilon_t$$

其中， $\epsilon_t \sim N(0, I)$ 。在这种情况下，动作噪声 $\epsilon_{1:T}$ 同样是高斯白噪声，其尺度由函数 σ 调制。白噪声在时间上不相关（ $\text{cov}[\epsilon_t; \epsilon_{t'}] = 0$ ）。在某些环境中，这会导致探索速度非常慢，从而导致状态空间覆盖不足，无法发现高回报区域。因此，使用具有时间相关性的动作噪声（ $\text{cov}[\epsilon_t; \epsilon_{t'}] > 0$ ），如 Ornstein-Uhlenbeck（OU）噪声，通常是有益的。OU 噪声被推荐为 DDPG 算法的默认选择，并已被证明能显著提高状态空间的覆盖率。

OU 噪声由随机微分方程（SDE）定义：

$$\dot{\epsilon}_t = -\theta \epsilon_t + \sigma \eta_t,$$

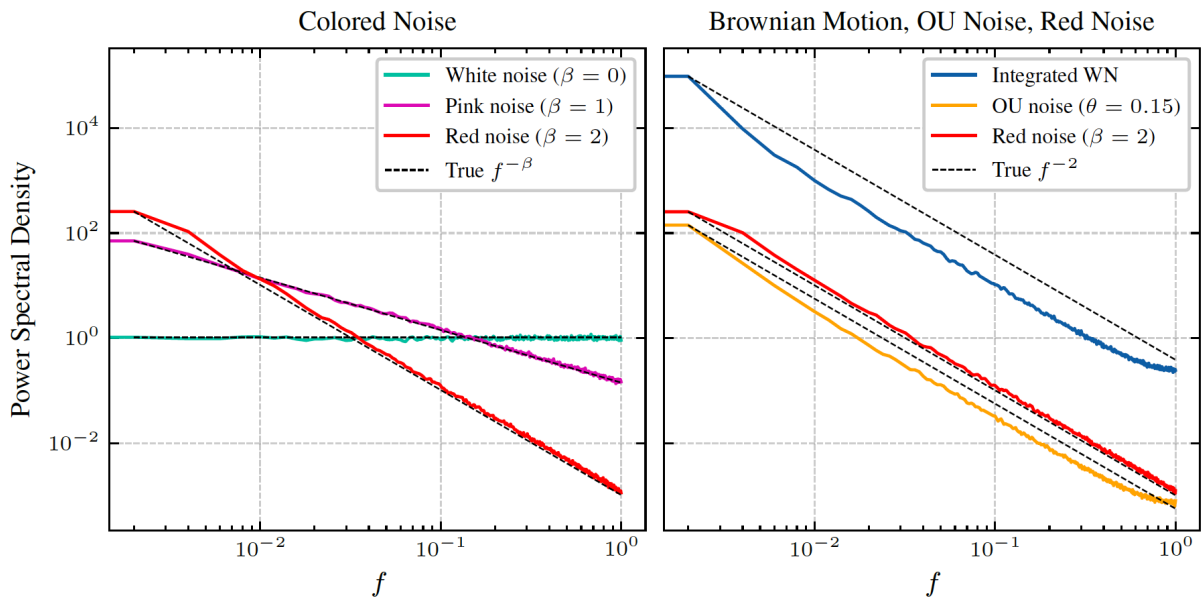
其中， η_t 是白噪声过程。如果 $\theta = 0$ ，那么这个方程定义了积分白噪声，也称为布朗运动。布朗运动在时间上相关，但如果以这种方式生成，则不能将其用作动作噪声，因为其方差会随时间无限增加，违反了动作空间的限制。通过设置 $\theta > 0$ （典型选择是 $\theta = 0.15$ ）来解决这个问题，这将限制方差的幅度。

对于噪声的一个统一定义是具有不同时间相关程度的彩色噪声，它概括了白噪声和布朗运动（布朗运动有时也被称为红噪声）。

定义: 彩色噪声 (Colored noise, CN) 如果从信号 ϵ_t 中抽取的 Fourier 变换满足性质 $|\hat{\epsilon}(f)|^2 \propto f^{-\beta}$, 那么该随机过程被称为具有颜色参数 β 的彩色噪声, 其中 $\hat{\epsilon}(f) = \mathcal{F}[\epsilon(t)](f)$ 表示 ϵ_t 的傅里叶变换 (f 是频率), $|\hat{\epsilon}(f)|^2$ 称为功率谱密度 (power spectral density, PSD)。

颜色参数 β 控制信号中的**时间相关程度**。不同的颜色参数 β 对应的彩色噪声的 PSD 如图3所示。

- 白噪声: $\beta = 0$ 的彩色噪声, 那么信号是不相关的, PSD 是平坦的, 这意味着所有频率都以相等的权重表示, 这种噪声被称为白噪声, 类似于光, 在光中所有可见频率上的信号功率相等时, 被认为是白光。
- 红噪声: $\beta = 2$ 的彩色噪声, 之所以这么命名, 是因为它在较低频率上具有更大的权重, 在光中对应于光谱的红色部分。具有恒定方差的高斯彩色噪声可以有高效的生成方式, 即一次性采样一个 episode 的完整噪声信号, 像这样生成噪声可以用 $\epsilon_{1:T} \sim CN_T(\beta)$ 表示。如果生成的是白噪声, 这样采样的噪声与每个时间步独立地从高斯分布中采样是等效的。如果生成的是红噪声 ($CN_T(\beta = 2)$) 与默认设置 $\theta = 0.15$ 的 OU 噪声非常相似, 因为它们本质上都是方差有界的布朗运动。通过设置 $0 < \beta < 2$, 彩色噪声允许我们寻找具有介于白噪声和红噪声之间的中间时间相关性的更好的默认动作噪声类型。
- 粉红噪声: $\beta = 1$ 的彩色噪声被称为粉红噪声。

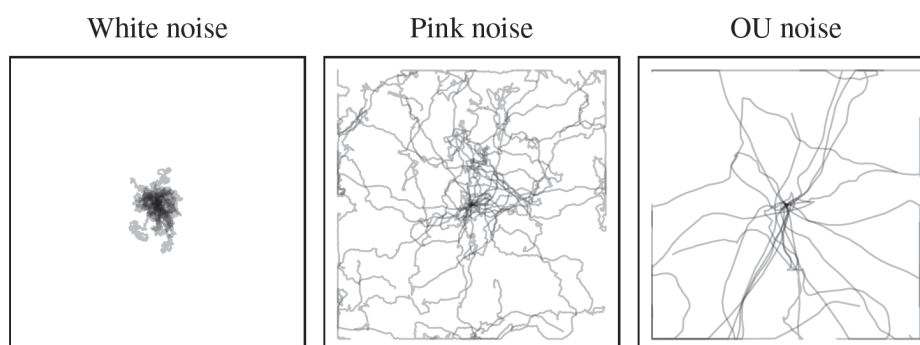


(图3: 左图: 在采样的彩色噪声信号的功率谱密度 (PSD) 中, 可以看到幂律趋势 [11]。右图: 通过积分从 $N(0, I)$ 采样的白噪声生成的布朗运动与两种相关的平稳噪声的比较: Ornstein-Uhlenbeck 噪声 ($\theta = 0.15$) 和红噪声。OU 噪声与红噪声之间的相似性是比较明显的。所有信号的时间长度均为 $T = 1000$ 。)

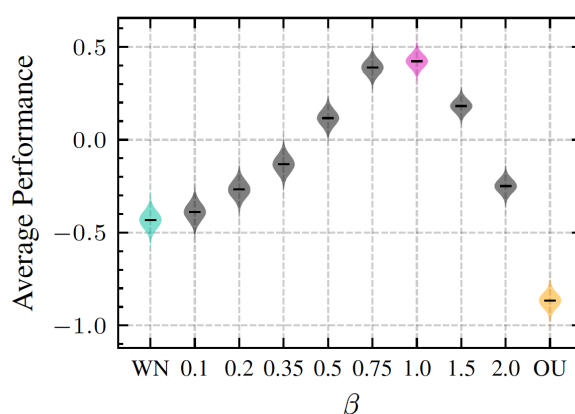
幂律趋势: 是指一种数据分布或现象的特征, 它在双对数坐标图 (双轴均为对数刻度的坐标图) 上呈现出直线关系。幂律分布广泛存在于自然科学、社会科学和经济学等领域。其数学公式为: $y = a * x^k$, 其中 y 和 x 是变量, a 和 k 是常数。幂律趋势的特点包括长尾分布、无标度性和自相似性等, 具体参考 [11]。

2.2 Pink noise 的普适性

- 简介：在具有连续动作空间的 off-policy 深度强化学习中，探索通常通过向动作选择过程中注入动作噪声来实现。基于随机策略的流行算法，如 SAC [4] 或 MPO，通过从不相关的高斯分布中采样动作来注入白噪声。然而，在许多任务中，白噪声不能提供足够的探索，因此使用时间相关的噪声。一个常见的选择是 Ornstein-Uhlenbeck (OU) 噪声，它与布朗运动（红噪声）密切相关。红噪声和白噪声都属于彩色噪声的大家族。在这项工作中，作者对 MPO 和 SAC 进行了全面的实验评估，以探讨其他颜色的噪声作为动作噪声的有效性。作者发现，**粉红噪声（位于白噪声和红噪声之间）在各种环境中明显优于白噪声、OU 噪声和其他替代方案。**
- 下面展示 [7] 中一些主要的实验结论。



(图4：在有界积分器环境中，作者研究了纯噪声智能体的轨迹，包括白噪声、粉红噪声和 OU 噪声。白噪声作用下的动作（左图）在这个环境中不能达到很远，因此无法收集到位于边缘处的稀疏奖励：它主要在**局部范围内探索**。而 OU 噪声（右图）只在**全局范围内探索**，容易在边缘处陷入困境。粉红噪声（中图）则在局部和全局探索之间取得了平衡，并比其他两种噪声**更均匀地覆盖了状态空间**。（细节参考原论文 [7] 第6节））



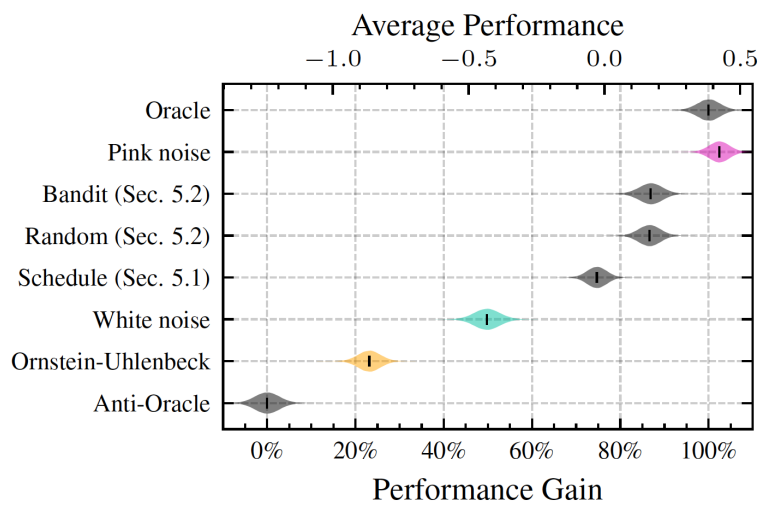
(图5：使用不同动作噪声类型的 MPO 和 SAC 在包含 DeepMind Control Suite 等**10**个任务上的期望平均性能分布。WN 表示白噪声、 $\beta=1$ 表示粉红噪声和 OU 表示 Ornstein-Uhlenbeck 噪声。可以看到**粉红噪声（位于白噪声和红噪声之间）在各种环境中明显优于白噪声、OU 噪声和其他方案。**）

作者发现，在各种各样的环境中，粉红噪声作为默认动作噪声表现最佳。然而，在某些环境中，粉红噪声表现仍然不如其他噪声类型，特别是在 Reacher 和 Pendulum 任务中，分别被白噪声和红噪声所

超越。这暗示可能不存在一个能在所有环境中都表现最佳的单一噪声类型。然而，这种考虑仅在训练过程中噪声保持不变的情况下适用。

[7] 的作者尝试了另一种思路，即在每个环境上和（或）训练进程中的噪声类型都采用变化的方式，因为这理论上可能会找到更好的探索机制。具体地，[7] 讨论了两种这样的自适应噪声的方法，在选择探索的噪声类型时有不同的机制：

- 一种是颜色调度（color-schedule）方法，即把颜色噪声的参数从 $\beta = 2$ 逐渐变为 $\beta = 0$ （即由全局探索逐渐变化为局部探索），细节参考原论文 [7] 第5.1节
- 另一种是多臂老虎机方法，对于每一个环境都单独寻找特定的最优的颜色噪声参数，细节参考原论文 [7] 第5.2节



（图6：[7] 中讨论的所有方法的期望平均性能分布。用彩色标注的是白噪声和 OU 噪声（之前 RL 算法经常的选择），以及粉红噪声（论文的建议选择）。虽然 OU 噪声和白噪声只实现了 Oracle 方法性能增益的约25%和50%，但粉红噪声的性能与 Oracle 相当！粉红噪声的表现也不逊于 color-schedule 方法以及多臂老虎机算法）。）

3. 总结

探索与利用的平衡是强化学习中的一个核心问题。在 RL 算法中，可以通过不同的方法来调整策略在动作空间中的探索能力。随机性策略（如 PPO 和 SAC）通常利用动作分布的熵来控制策略的探索能力，而确定性策略（如 DDPG 和 TD3）则通过添加动作噪声来实现探索。不同的环境类型和设置对策略探索能力的需求也有所不同，因此在实际应用中，需要根据具体情况选择合适的探索方法和参数设置。

4. 参考文献

- [1] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.
- [2] Fujimoto, Scott, Herke Hoof, and David Meger. "Addressing function approximation error in actor-critic methods." *International conference on machine learning*. PMLR, 2018.
- [3] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.
- [4] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//International conference on machine learning. PMLR, 2018: 1861-1870.
- [5] Schulman, J., Abbeel, P., and Chen, X. Equivalence between policy gradients and soft Q-learning. arXiv preprint arXiv:1704.06440, 2017a.
- [6] Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. Maximum a posteriori policy optimisation. arXiv preprint arXiv:1806.06920, 2018.
- [7] Onno_Eberhard, Jakob Hollenstein, Cristina Pinneri, Georg Martius. Pink Noise Is All You Need: Colored Noise Exploration in Deep Reinforcement Learning. ICLR 2023.
- [8] Andrychowicz M, Raichuk A, Stańczyk P, et al. What matters in on-policy reinforcement learning? a large-scale empirical study[J]. arXiv preprint arXiv:2006.05990, 2020.
- [9] Haarnoja T, Zhou A, Hartikainen K, et al. Soft actor-critic algorithms and applications[J]. arXiv preprint arXiv:1812.05905, 2018.
- [10] Shannon C E. A mathematical theory of communication[J]. ACM SIGMOBILE mobile computing and communications review, 2001, 5(1): 3-55.
- [11] Erdil E, Sevilla J. Power Law Trends in Speedrunning and Machine Learning[J]. arXiv preprint arXiv:2304.10004, 2023.