

隐变量因果模型视角下的策略梯度方差优化*

蔡瑞初¹, 林富艺¹, 陈薇¹, 朱海鹏¹, 郝志峰^{1,2}

¹(广东工业大学 计算机学院, 广东 广州 510006)

²(汕头大学 数学与计算机学院, 广东 汕头 515063)

通信作者: 陈薇, E-mail: weichen@gdut.edu.cn



摘要: 深度强化学习已在多个领域取得了显著突破, 其中策略梯度算法因适用于处理非线性和高维状态空间的问题而被广泛采用. 然而, 现有策略梯度算法在实际应用中仍面临高方差问题, 这会导致算法收敛速度变慢, 甚至可能陷入次优解. 针对这一挑战, 从隐因果模型的视角提出一种策略梯度方差优化方法. 通过引入隐变量刻画未观测随机信息, 构建并学习隐变量因果模型. 基于隐变量因果模型, 提出因果价值函数, 结合长短期记忆网络, 根据时效性区分衡量未观测随机信息对价值估计的影响作用, 提高动作优势函数预估的准确性, 降低策略梯度方差. 实验表明, 与前沿的同类算法相比, 基于隐变量因果模型的方法在多个任务更具有优越性和稳定性.

关键词: 深度强化学习; 策略梯度; 方差优化; 隐因果模型; 因果价值函数

中图法分类号: TP183

中文引用格式: 蔡瑞初, 林富艺, 陈薇, 朱海鹏, 郝志峰. 隐变量因果模型视角下的策略梯度方差优化. 软件学报. <http://www.jos.org.cn/1000-9825/7547.htm>

英文引用格式: Cai RC, Lin FY, Chen W, Zhu HP, Hao ZF. Variance Optimization of Policy Gradients from Latent Variable Causal Model Perspective. Ruan Jian Xue Bao/Journal of Software (in Chinese). <http://www.jos.org.cn/1000-9825/7547.htm>

Variance Optimization of Policy Gradients from Latent Variable Causal Model Perspective

CAI Rui-Chu¹, LIN Fu-Yi¹, CHEN Wei¹, ZHU Hai-Peng¹, HAO Zhi-Feng^{1,2}

¹(School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China)

²(College of Mathematics and Computer Science, Shantou University, Shantou 515063, China)

Abstract: Deep reinforcement learning has achieved significant breakthroughs in various fields, with policy gradient algorithms widely adopted due to their suitability for handling nonlinear and high-dimensional state spaces. However, in practical applications, existing policy gradient algorithms still suffer from high variance, which slows convergence and may cause suboptimal solutions. To tackle this challenge, a variance optimization method for policy gradients is proposed from a latent causal model perspective. By introducing latent variables to characterize unobserved random information, a latent variable causal model is constructed and learned. Utilizing this model, a causal value function is proposed and combined with long short-term memory (LSTM) networks to differentiate the temporal impact of unobserved information on value estimation. This approach improves the accuracy of action advantage function estimation and reduces policy gradient variance. Experiments demonstrate that the proposed latent variable causal model outperforms state-of-the-art algorithms across multiple tasks, with better performance and stability.

Key words: deep reinforcement learning; policy gradient; variance optimization; latent causal model; causal value function

深度强化学习^[1]通过深度神经网络高效拟合价值函数和策略函数, 已经在游戏^[2]、自动驾驶^[3]和机器人控制^[4,5]等领域取得了广泛的应用. 其中, 策略梯度算法^[6]通过沿着累计奖励增大的方向直接优化策略参数, 在处理非线性^[7]和高维状态空间^[8]相关的环境场景具有显著优势. 然而, 由于实际应用中策略和环境的随机性^[9], 将现有大多数策略梯度算法应用于实际场景仍面临着高方差问题^[10]. 这种方差会导致算法收敛速度变慢, 甚至可能陷入

* 基金项目: 新一代人工智能国家科技重大专项 (2021ZD0111500); 国家优秀青年科学基金 (62122022); 国家自然科学基金 (62206064)

收稿时间: 2024-07-08; 修改时间: 2025-01-16; 采用时间: 2025-09-03; jos 在线出版时间: 2026-01-14

次优解. 因此, 研究一种优化策略梯度方差的方法具有一定的实际意义和价值.

为了应对策略梯度算法中的高方差问题, 目前的工作通过重要性采样^[11]、改进梯度更新过程^[12]、减基线^[13]等方式来优化策略梯度的估计. 尽管现有方法在一定程度上缓解了策略梯度算法中的高方差问题, 但它们仍然存在一些局限性. 重要性采样方法通过计算不同策略样本的分布差异^[14]来降低方差, 对样本内部的随机信息并不敏感. 改进梯度更新过程的方法通过限制策略更新幅度^[15]或改进梯度估计^[16]来减少方差, 未能充分利用环境的动态特性. 减基线法性能高度依赖于基线函数的选择, 而已有的基线函数未考虑环境随机机制^[17], 应对不同随机机制的环境表现较差.

针对这些局限性, 如何充分考虑环境随机性, 降低策略梯度估计方差, 是进一步提升策略梯度算法性能的关键. 在实际应用中, 环境往往具有复杂的动态性和随机性, 不同状态间存在着错综复杂的关系. 传统方法难以捕捉这些关系, 导致其在面对环境变化时表现出较大的不稳定性. 如图 1(a) 在无风时, 月球登陆车 (Lunar Lander) 在决策动作后的落点为 B1; 但在受到风等未观测随机信息的影响时, 月球登陆车位置在相同动作决策下发生偏移, 落点为 B2, 导致得到的奖励结果不同. 针对类似场景, 传统智能体只建模环境可观测状态、动作和奖励的马尔可夫决策过程 (如图 1(b) 所示), 使用状态作为价值基线判断的拟合信息源, 以此计算动作优势估计值. 在不同未观测随机信息的干扰下, 传统智能体计算的动作优势估计值会出现不同的波动, 使策略网络出现方差抖动问题. 这种情况下, 传统的价值基线便不能辅助智能体从根本上解决环境随机性导致策略梯度高方差的问题.

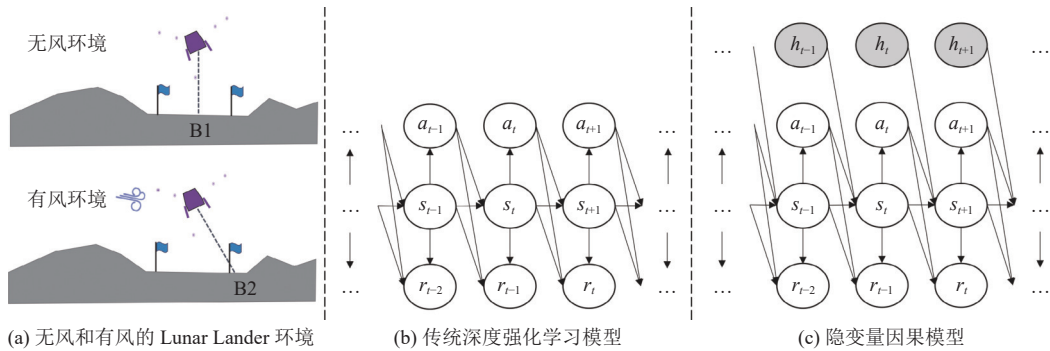


图 1 隐变量因果模型建模

通过上述分析发现, 如果智能体能够获得环境的随机信息, 以此辅助价值评估, 便能让策略梯度算法更准确地估计动作优势的值. 此时, 动态环境中未观测的随机信息可能在智能体做出决策动作后, 扰动下一时刻返回的可观测状态信息. 这部分未观测的随机信息可以刻画为隐变量, 并通过因果模型来建模如图 1(c) 所示的动作、状态和奖励之间的互相影响机制. 因此, 本文从隐因果模型的视角提出了一种策略梯度方差优化方法 (policy gradient variance optimization, PGVO). 该方法首先引入隐变量刻画未观测随机信息, 构建含有隐变量的结构因果模型. 接着, 基于所构建的动态环境的结构因果模型, 以与期望回报存在直接或间接因果关系的外生变量为输入, 构建因果价值函数 (causal value function, CVF), 识别隐变量对期望回报的影响. 最后, 上述步骤可以通过利用深度强化学习过程中收集的轨迹数据训练因果模型, 学习动态环境因果机制. 利用因果模型推断的环境未观测随机信息, 进而使用因果价值函数评估当前时刻的价值, 降低策略梯度估值中动作优势估计值的方差.

本文主要贡献包括以下 3 个方面: (1) 提出一种基于隐变量因果模型的策略梯度方差优化方法. 引入隐变量刻画未观测随机信息, 构建并学习隐变量因果模型. 引入因果编码网络和因果解码网络, 将难以求解的后验分布问题转化为变分下界问题, 确保因果模型对环境因果机制的拟合能力和对未观测随机信息的推断能力. (2) 提出因果价值函数, 结合长短期记忆网络, 根据时效性区分衡量未观测随机信息对价值估计的影响作用, 提高动作优势函数预估的准确性, 降低策略梯度方差. (3) 在 OpenAI Gym 平台^[18]中, 将基于隐变量因果模型的策略梯度方差优化方法 (PGVO) 分别与 PPO (proximal policy optimization)^[19]和 HVF (hindsight value function)^[17]进行对比, 验证了本方法

的优越性和稳定性.

1 相关工作

1.1 隐因果强化学习

因果关系在深度强化学习中的应用已取得显著成效,为解决模型“黑盒”问题提供了新视角,提升了系统可解释性. Lin 等人^[20]提出了面向自动驾驶的安全感知因果表征方法,通过将因果结构融入离线强化学习,显著提高了决策的可信度与安全性. Chen 等人^[21]研究了因果感知大语言模型,展示了如何通过因果学习增强深度强化学习智能体的决策能力,使其能更好地适应和行动. Wang 等人^[22]开发的 ERCI 方法将可解释的经验回放与因果推断相结合,使深度强化学习系统能够解释其学习过程和决策依据. Cao 等人^[23]的研究则聚焦于因果信息优先级排序,通过识别关键因果因素提高了深度强化学习的效率. 这些研究表明,因果推理正成为连接深度强化学习性能与可解释性的重要桥梁,使决策更具可解释性.

在这些因果强化学习方法中,隐因果强化学习作为一个新兴分支与本职工作密切相关. 隐因果强化学习主要通过隐变量建模环境中的本质特征和不确定性,可分为两个主要研究方向.

在环境表征与建模方面,隐变量用于捕获环境中的本质特征和不确定性. Zhang 等人^[24]通过双向训练提取任务相关的因果信息,构建环境的稳定表征,在连续控制任务中实现了优异的泛化性能. Bennett 等人^[25]针对离线强化学习,通过隐变量显式建模未观测的混杂因素,在保持计算效率的同时提升了模型泛化能力.

在环境适应与迁移方面,研究者提出了基于隐变量分解的方法. Huang 等人^[26]设计了变分推断框架,通过预训练实现环境变化的多维度分解. Feng 等人^[27]采用因子分解方法处理环境迁移,利用结构化隐变量表示提升了知识迁移效果和可解释性.

然而,现有基于隐变量建模的因果强化学习方法主要关注环境表征与迁移问题,对策略梯度估计中的高方差问题尚缺乏有效解决方案. 策略梯度方差过大不仅降低收敛速度,还会影响收敛时累计奖励的性能. 本文从因果模型角度出发,提出一种新的方差降低方法,通过因果结构的显式建模提升策略梯度估计的准确性与稳定性.

1.2 降低策略梯度方差的方法

策略梯度方法的发展始于早期的 REINFORCE 算法,其主要基于直接优化策略函数和使用梯度上升的思想. 在策略梯度方法中,策略梯度估值的方差会影响算法的收敛速度和性能,对算法的稳定性和训练效率有着重要的影响. 为了降低策略梯度估值的方差,研究者们提出了多种方法.

一种常见的方法是重要性采样. Thomas 等人^[14]提出了一种数据高效的离策略 (off-policy) 策略评估方法,用重要性采样提升评估准确性和效率. Gu 等人^[28]在 Q-Prop 中结合离策略价值估计,通过重要性采样减小方差,提高样本效率. Espeholt 等人^[29]在 IMPALA 中采用重要性加权修正不同策略采样的数据,实现大规模分布式深度强化学习. Gruslys 等人^[30]在 Reactor 中使用快速且样本高效的 Actor-Critic 算法,结合重要性采样实现高效学习. 基于重要性采样的方法虽然能识别样本间的策略分布差异,降低这部分差异造成的策略梯度估值方差,但它不能捕获样本中环境机制的随机性,无法解决因此产生的高方差问题.

另一种常见的方法是改进梯度更新过程. Schulman 等人^[15]提出了信任域策略优化 (TRPO) 算法,通过约束策略更新的步长,在保证单调提升的同时,有效地控制了策略变化的幅度,提高了训练的稳定性. Schulman 等人^[19]进一步提出了近端策略优化 (PPO) 算法,通过引入替代目标函数和策略裁剪技术,在 TRPO 的基础上简化了优化过程,使得算法更易于实现和调参. Papini 等人^[12]提出了随机方差减少策略梯度 (SVRPG) 算法,通过引入方差减少技术,如 SVRG 和 SAGA,来降低策略梯度估计的方差,提高了训练效率和稳定性. Liu 等人^[16]对策略梯度和自然策略梯度方法进行了改进分析,提出了新的方差减少技术,并给出了收敛性保证,进一步提高了这些方法的理论基础和实践效果. Xu 等人^[31]对 SVRPG 算法进行了改进分析,提供了更紧致的收敛界,展示了该算法在方差减少和收敛速度上的优势. 基于改进梯度更新过程的方法通过限制策略更新幅度或改进梯度估计来减少方差,未能充分利用环境的动态特性.

减基线法也是常见的降低方差的技术. Schulman 等人^[32]提出了广义优势估计 (GAE), 引入状态价值函数的自举, 通过 $TD(\lambda)$ 误差来估计优势函数, 从而在降低方差的同时保持较低的偏差. Wu 等人^[33]提出利用策略的结构性信息构建基于动作的基线函数, 针对每个因素计算独立的基线, 排除其他因素的影响, 从而减少方差. Mao 等人^[34]提出了基于元学习的方法学习一个输入依赖基线, 它综合考虑状态和整个后续输入序列, 可以排除外部产生的随机扰动, 从而更准确地估计一个动作的质量, 降低方差. Guo 等人^[17]提出了一种信息论方法, 通过学习与未来状态和奖励的互信息最大且与当前动作无关的后见向量 (hindsight vector), 然后再获得后见价值函数, 从而持续降低策略梯度方法的奖励方差, 得到稳定的训练效果. 上述基线或因没考虑到未观测随机信息, 而无法消除未观测信息对动作优势估计值的影响; 或只是直观地利用了互信息, 没把握动态环境的数据生成机制, 面对不同环境机制时的效果较差.

不同于现有降低策略梯度方差的方法, 本文从环境数据生成机制的角度出发, 利用因果模型描述动态环境中不变的因果关系, 并引入隐变量刻画未观测随机信息. 基于因果编解码框架推断隐变量, 结合因果价值函数充分考虑未观测随机信息的影响, 从而提高动作优势函数预估的准确性, 有效降低策略梯度方差.

2 基础知识

本文所提方法主要关于深度强化学习、策略梯度方法和结构因果模型, 下面就相关概念和基本知识予以介绍.

2.1 深度强化学习

深度强化学习 (deep reinforcement learning, DRL) 是一种机器学习方法, 通过与环境的交互来学习策略, 以最大化累计奖励. 与监督学习不同, 深度强化学习不需要预先标注的数据, 而是通过试错和反馈机制来进行学习. 深度强化学习中的主要组成部分包括智能体 (Agent)、环境 (Environment)、动作 (Action)、状态 (State) 和奖励 (Reward). 为了系统化地描述深度强化学习问题, 通常使用马尔可夫决策过程 (Markov decision process, MDP), 如图 1(b) 所示. MDP 是一个包含五元组的数学模型 $\langle S, A, P, R, \gamma \rangle$. S 是状态空间, s_t 表示在时刻 t 的状态. A 是动作空间, a_t 表示智能体在时刻 t 的动作. P 是状态转移概率, $P(s_{t+1}|s_t, a_t)$ 表示在状态 s_t 执行动作 a_t 后转移到下一时刻状态 s_{t+1} 的概率. R 是奖励函数, $R(s_t, a_t, s_{t+1})$ 表示在状态转移过程对应的奖励. γ 是折扣因子, 用于衡量未来奖励和当前奖励的相对重要性. 在 MDP 中, 智能体在深度强化学习过程中收集的轨迹为所有时刻的状态、动作和奖励的集合, 记作 $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_T, a_T, r_T, s_{T+1}\}$.

2.2 策略梯度方法

策略梯度方法是一类直接优化策略的深度强化学习算法, 通过最大化策略的期望回报来学习最佳策略. 如果已知智能体在深度强化学习过程中收集的轨迹为 $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_T, a_T, r_T, s_{T+1}\}$, 该轨迹的回报 $G(\tau) = \sum_{t=0}^T \gamma^t r_t$. 假设策略 π 由参数 θ_a 参数化, 其性能指标表示为 $J(\theta_a)$. 策略梯度方法的目标是最大化策略的期望回报, 则 $J(\theta_a) = \mathbb{E}_{\tau \sim \pi_{\theta_a}} [G(\tau)]$. 策略 π_{θ_a} 性能指标的梯度为 $\nabla_{\theta_a} J(\theta_a) = \mathbb{E}_{\tau \sim \pi_{\theta_a}} [G(\tau) \nabla_{\theta_a} \log p(\tau|\pi_{\theta_a})]$. 为了降低策略梯度的方差, 一般引入基线价值函数 V , 使策略 π_{θ_a} 性能指标的梯度为:

$$\nabla_{\theta_a} J(\theta_a) = \mathbb{E}_{\tau \sim \pi_{\theta_a}} [(G(\tau) - V) \nabla_{\theta_a} \log p(\tau|\pi_{\theta_a})] \quad (1)$$

2.3 结构因果模型

结构因果模型 (structural causal model, SCM)^[35]提供了一种描述系统中变量之间因果关系结构的框架. 通过明确定义变量之间的因果影响, 结构因果模型从因果机制层面总结了数据的生成过程, 为因果推断和分析提供了坚实的基础. 结构因果模型 M 是一个四元组 $\langle U, X, F, P(u) \rangle$. U 是一组外生变量的集合 $\{U_1, U_2, \dots, U_n\}$, 外生变量是系统外部的因素或背景条件, 由模型外部因素决定. X 是一组内生变量的集合 $\{X_1, X_2, \dots, X_n\}$, 内生变量是模型内部需要解释或预测的变量, 由其他变量 (其他内生变量和外生变量) 决定; 在内生变量 X_i 的父节点集合中, 由其他内生变量组成的子集被表示为 PA_i . F 是一组函数的集合 $\{f_1, f_2, \dots, f_n\}$, 每个函数 f_i 是其他变量集合 $PA_i \cup \{U_i\}$ 到内生变量子集 $\{Y_i\}$ 的映射. 函数和各个对应的变量集构成结构方程 $Y_i := f_i(PA_i, U_i)$, 其中 $U_i \in U$, $PA_i \subseteq X \setminus X_i$, $i = 1, 2, \dots, n$.

结构方程明确了因果方向, 刻画了系统内部的因果关系. SCM 的一个关键优势在于其模块化和可解释性. 通过将复杂系统分解为相互关联的子模块, 这种模块化使得模型更容易构建、理解和维护, 同时也有助于识别关键的因果关系和潜在的干预目标. 通过明确表示变量之间的因果关系, SCM 使研究者能够提出和回答关于系统行为的问题, 识别关键的因果机制, 并估计因果效应.

3 隐变量因果模型和因果价值函数

3.1 隐变量因果模型

通过上述分析可知, 策略梯度算法的高方差问题主要源于环境中的外部扰动. 这些扰动通常表现为未观测的随机信息, 且可能影响下一时刻可观测的状态. 这使得 t 时刻的状态除了依赖 $t-1$ 时刻可观测的状态和动作外, 还受到 $t-1$ 时刻的未观测的随机扰动的影响, 从而使策略梯度估计的方差增大. 以 Gym 的 Lunar Lander 环境为例, 风就是一个典型的未观测随机信息. 其可能在智能体做出决策动作后, 扰动下一时刻返回的可观测状态信息. 在实际环境中, 类似的未观测随机信息普遍存在, 例如机器人运行过程由于接触材料变化而引起的摩擦力变化或运输车因为货物装卸而产生的不同负重等. 因此, 本文在传统深度强化学习中的马尔可夫决策过程五元组基础上, 引入隐变量 h_t 刻画 t 时刻的动态环境的未观测随机信息, 并假设存在隐变量 h_t 指向下一时刻可观测状态信息 s_{t+1} 的因果边. 其因果关系如图 1(c) 所示. 因此, 在时刻 t 时, 含有隐变量的结构因果模型可以形式化为:

$$\begin{cases} a_t := f_a(s_t) \\ s_t := f_s(s_{t-1}, a_{t-1}, h_{t-1}) \\ r_t := f_r(s_t, a_t, s_{t+1}) \end{cases} \quad (2)$$

3.2 因果价值函数

基于上述因果模型, 状态转移不仅取决于可观察状态信息和采取的动作, 还受到未观测随机信息的影响. 而奖励函数会结合当前时刻的状态、动作和下一时刻的状态返回环境的反馈, 这也意味着未观测随机信息还可能影响奖励信息. 在给定时刻 t 下的状态为 s_t 时, 内生变量 r_t 的结构方程可以展开为:

$$r_t = f_r(s_t, a_t, s_{t+1}) = f_r(s_t, a_t, f_s(s_t, a_t, h_t)) = f_r(s_t, f_a(s_t), f_s(s_t, f_a(s_t), h_t)) \quad (3)$$

可以得知与内生变量 r_t 存在直接或间接因果关系的外生变量集合为 (s_t, h_t) . 相同地, 在给定时刻 $t+1$ 下的状态为 s_{t+1} , 通过结构因果模型的分析, 可以寻找与内生变量 r_{t+1} 存在直接或间接因果关系的外生变量集合为 (s_t, h_t, h_{t+1}) .

在给定时刻 t 下的状态为 s_t 时, 从时刻 t 开始累计, 到轨迹最终结束的時刻 T 的轨迹回报为 $G_t = \sum_{r=t}^T \gamma^{r-t} r_r$, 所以与 G_t 存在直接或间接因果关系的外生变量集合为 $(s_t, h_t, h_{t+1}, \dots, h_T)$. 为了方便表示, 引入从时刻 t 开始到轨迹结束的所有隐变量序列的定义:

$$h_t^+ \triangleq (h_t, h_{t+1}, \dots, h_T) \quad (4)$$

因为价值函数 V 往往就是对轨迹回报 G_t 的期望值的拟合, 所以 V 可以使用 (s_t, h_t^+) 作为网络输入进行拟合. 故提出因果价值函数 (CVF) 的定义如下所示:

$$V^{\text{causal}}(s_t, h_t^+) \triangleq \mathbb{E}_{r \sim \pi_{\theta_t}} [G_t | s_t, h_t^+] \quad (5)$$

根据公式 (3) 可知, 奖励函数会受到隐变量的干扰. 当采用状态价值函数 $V^s(s_t) \triangleq \mathbb{E}_{r \sim \pi_{\theta_t}} [G_t | s_t]$ 作为基线 V_t 时, 动作优势函数 $A_t = G_t - V_t$ 的估计值会受到隐变量序列的影响. 这是因为轨迹回报 G_t 受隐变量序列 h_t^+ 的影响, 而状态价值函数并未考虑隐变量序列 h_t^+ 的影响, 导致动作优势估计存在较大方差. 在样本充足且隐变量均值为 0 的情况下, 对动作优势估计值多次求平均可实现类似无偏的效果, 但会降低数据样本效率. 而且即使通过平均实现了动作优势的无偏估计, 策略梯度在优化过程中仍会受到隐变量序列 h_t^+ 的影响, 导致梯度估计的方差过大并影响收敛速度和收敛时累计奖励的性能. 因此, 本文采用因果价值函数作为基线 V_t , 以此充分考虑隐变量序列 h_t^+ 对轨迹回报 G_t 的影响, 提高动作优势预估的准确性并降低策略梯度的方差.

$$A_t = G_t - V^{\text{causal}}(s_t, h_t^+) \quad (6)$$

定理 1. 假设数据的生成机制满足公式 (2), 那么基于因果价值函数的策略梯度估计是无偏的.

证明: 引入因果价值函数作为基线函数后, 策略梯度可以表示为:

$$\begin{aligned} \nabla_{\theta_a} J(\theta_a) &= \mathbb{E}_{\tau \sim \pi_{\theta_a}} \left[(G(\tau) - V^{\text{causal}}) \nabla_{\theta_a} \log p(\tau | \pi_{\theta_a}) \right] \\ &= \mathbb{E}_{s_{0:T+1}, a_{0:T}, \tau_{0:T}} \left[\sum_{t=0}^T (G_t - V^{\text{causal}}(s_t, h_t^+)) \nabla_{\theta_a} \log \pi_{\theta_a}(a_t | s_t) \right] \\ &= \mathbb{E}_{s_{0:T+1}, a_{0:T}, \tau_{0:T}} \left[\sum_{t=0}^T G_t \nabla_{\theta_a} \log \pi_{\theta_a}(a_t | s_t) \right] - \mathbb{E}_{s_{0:T+1}, a_{0:T}} \left[\sum_{t=0}^T V^{\text{causal}}(s_t, h_t^+) \nabla_{\theta_a} \log \pi_{\theta_a}(a_t | s_t) \right] \end{aligned} \quad (7)$$

根据环境的因果模型展开, $V^{\text{causal}}(s_t, h_t^+)$ 中 s_t 和 h_t^+ 都不是关于 a_t 的函数, 且 $\sum_{t=0}^T V^{\text{causal}}(s_t, h_t^+) \nabla_{\theta_a} \log \pi_{\theta_a}(a_t | s_t)$ 涉及的 $s_{0:T}$ 、 $h_{0:T}$ 和 $a_{0:T}$ 都不依赖于 s_{T+1} , 可得:

$$\begin{aligned} \mathbb{E}_{s_{0:T+1}, a_{0:T}} \left[\sum_{t=0}^T V^{\text{causal}}(s_t, h_t^+) \nabla_{\theta_a} \log \pi_{\theta_a}(a_t | s_t) \right] &= \mathbb{E}_{s_{0:T}, a_{0:T}} \left[\sum_{t=0}^T V^{\text{causal}}(s_t, h_t^+) \nabla_{\theta_a} \log \pi_{\theta_a}(a_t | s_t) \right] \\ &= \sum_{t=0}^T \mathbb{E}_{s_t, a_t} \left[V^{\text{causal}}(s_t, h_t^+) \nabla_{\theta_a} \log \pi_{\theta_a}(a_t | s_t) \right] \\ &= \sum_{t=0}^T \mathbb{E}_{s_t} \left\{ V^{\text{causal}}(s_t, h_t^+) * \mathbb{E}_{a_t} [\nabla_{\theta_a} \log \pi_{\theta_a}(a_t | s_t)] \right\} \\ &= \sum_{t=0}^T \mathbb{E}_{s_t} \left[V^{\text{causal}}(s_t, h_t^+) * 0 \right] \\ &= 0 \end{aligned} \quad (8)$$

综合公式 (7) 和 (8), 易证 $\nabla_{\theta_a} J(\theta_a) = \mathbb{E}_{s_{0:T+1}, a_{0:T}, \tau_{0:T}} \left[\sum_{t=0}^T G_t \nabla_{\theta_a} \log \pi_{\theta_a}(a_t | s_t) \right]$, 故定理 1 成立.

定理 2. 相对于以状态价值函数作为基线函数, 基于因果价值函数的策略梯度算法在动作优势估计值上的方差更小.

证明: 引入因果价值函数作为基线函数后, 策略梯度估值中动作优势估计值的方差为:

$$\begin{aligned} \mathbb{V}[G_t - V^{\text{causal}}(s_t, h_t^+)] &= \mathbb{E} \left\{ \mathbb{V} \left[\left[G_t - V^{\text{causal}}(s_t, h_t^+) \right] | s_t, h_t^+ \right] \right\} + \mathbb{V} \left\{ \mathbb{E} \left[\left[G_t - V^{\text{causal}}(s_t, h_t^+) \right] | s_t, h_t^+ \right] \right\} \\ &= \mathbb{E} \left\{ \mathbb{V} \left[G_t | s_t, h_t^+ \right] \right\} - \mathbb{E} \left\{ \mathbb{V} \left[V^{\text{causal}}(s_t, h_t^+) | s_t, h_t^+ \right] \right\} + \mathbb{V} \left\{ \mathbb{E} \left[G_t | s_t, h_t^+ \right] - \mathbb{E} \left[V^{\text{causal}}(s_t, h_t^+) | s_t, h_t^+ \right] \right\} \\ &= \mathbb{E} \left\{ \mathbb{V} \left[G_t | s_t, h_t^+ \right] \right\} + \mathbb{V} \left\{ \mathbb{E} \left[G_t | s_t, h_t^+ \right] - \mathbb{E} \left[V^{\text{causal}}(s_t, h_t^+) | s_t, h_t^+ \right] \right\} \\ &= \mathbb{E} \left\{ \mathbb{V} \left[G_t | s_t, h_t^+ \right] \right\} + \mathbb{V} \left\{ \mathbb{E} \left[G_t | s_t, h_t^+ \right] - \mathbb{E} \left[V^{\text{causal}}(s_t, h_t^+) | s_t, h_t^+ \right] \right\} \\ &= \mathbb{E} \left\{ \mathbb{V} \left[G_t | s_t, h_t^+ \right] \right\} \end{aligned} \quad (9)$$

其中, 由 $\mathbb{V}[f(x)|x] = \mathbb{E} \{ [f(x) - \mathbb{E}[f(x)|x]]^2 \} = \mathbb{E} \{ [f(x) - f(x)]^2 \} = 0$, 可知 $\mathbb{E} \left\{ \mathbb{V} \left[V^{\text{causal}}(s_t, h_t^+) | s_t, h_t^+ \right] \right\} = 0$. 结合公式 (5) 给出的因果价值函数定义, 可推导出公式 (9) 第 3 个等式.

使用状态价值函数作为基线函数时, 策略梯度估值中动作优势估计值的方差为:

$$\begin{aligned} \mathbb{V}[G_t - V^s(s_t)] &= \mathbb{E}[\mathbb{V}(G_t | s_t)] = \mathbb{E} \left\{ \mathbb{E}[\mathbb{V}(G_t | s_t, h_t^+)] + \mathbb{V}[\mathbb{E}(G_t | s_t, h_t^+)] \right\} \\ &= \mathbb{E}[\mathbb{V}(G_t | s_t, h_t^+)] + \mathbb{E} \left\{ \mathbb{V}[\mathbb{E}(G_t | s_t, h_t^+)] \right\} \\ &\geq \mathbb{E}[\mathbb{V}(G_t | s_t, h_t^+)] = \mathbb{V}[G_t - V^{\text{causal}}(s_t, h_t^+)] \end{aligned} \quad (10)$$

其中, 利用公式 (9) 的证明逻辑可以证明公式 (10) 第 1 个等式, 利用 $\mathbb{V}(y) = \mathbb{E}[\mathbb{V}(y|x)] + \mathbb{V}[\mathbb{E}(y|x)]$ 可以证明公式 (10) 第 2 个等式.

综合公式 (9) 和 (10), 易证得 $\mathbb{V}[G_t - V^s(s_t)] \geq \mathbb{V}[G_t - V^{\text{causal}}(s_t, h_t^+)]$, 故定理 2 成立.

4 算法实现

4.1 学习隐变量因果模型

本文受因果模型和变分推断的启发, 提出了基于隐变量因果模型的因果编解码框架 (如图 2 所示). 该框架通过引入隐变量刻画未观测随机信息, 旨在实现隐变量因果模型的学习和隐变量的推断. 框架主要包含因果编码网络 (causal encoder) 和因果解码网络 (causal decoder) 两个核心组件. 因果编码网络基于变分推断方法, 利用观测信息推断隐变量的变分后验分布. 具体而言, 编码网络首先推断后验分布的参数 (如均值和方差), 随后在假设隐变量服从先验分布的条件下, 采用重参数化技巧实现端到端的有效优化. 这一设计既保证了优化效率, 又保持了因果模型在表示和推理因果关系方面的能力. 因果解码网络则基于推断的隐变量, 结合当前时刻的可观测状态与动作信息, 对下一时刻的可观测状态进行重构预测, 从而拟合状态转移的数据生成机制. 通过对比重构状态与真实观测状态之间的偏差, 可有效评估因果模型的拟合性能. 最终, 该框架通过编码和解码网络的协同作用, 实现了对隐变量因果机制的有效拟合和轨迹中隐变量信息的准确推断.

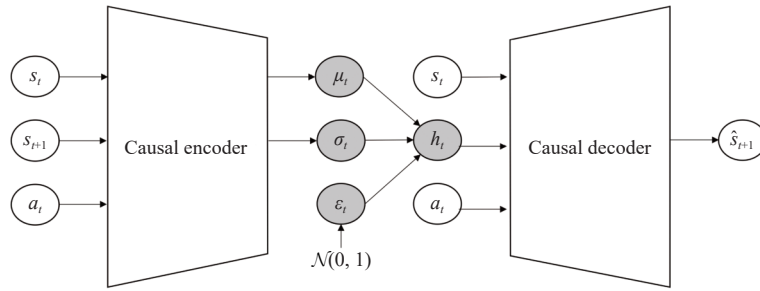


图 2 基于结构因果模型的因果编解码框架

在该框架下, 首先将轨迹中所有时刻信息的联合概率建模为一个边缘对数似然. 这个边缘对数似然是通过将每个时刻的隐变量边缘化而得到的. 然后, 利用因果模型内在的因果依赖关系, 结合变分推断技术, 将这个边缘对数似然进一步拆解, 以便于计算和优化. 通过这种方式, 原本难以求解的后验概率问题被转化为优化变分下界 (evidence lower bound, *ELBO*) 的问题. 设引入的因果编码网络参数为 φ , 因果解码网络为 ψ , 则轨迹中所有时刻信息的对数联合概率为:

$$\log p(s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_T, a_T, r_T, s_{T+1}) = \sum_{t=0}^T D_{\text{KL}}(q_{\varphi}(h_t | s_t, a_t, s_{t+1}) \| p(h_t | s_t, a_t, s_{t+1})) + ELBO \quad (11)$$

其中, 等式右侧第 1 项表示近似后验分布 (由因果编码网络推断) 和真实后验分布的 Kullback-Leibler 散度 (KL 散度) 在所有时刻上的总和, 反映轨迹中因果编码网络在拟合真实后验分布时的误差. 事实上, KL 散度是非负的. 又由于近似后验分布与真实后验分布在实际应用中几乎不可能完全相同, 这一项恒大于 0. 等式右侧第 2 项为 *ELBO*. 通过最大化 *ELBO*, 使近似后验分布更好地拟合真实后验分布, 即提高因果编码网络对隐变量的推断能力. *ELBO* 通过因果模型内在的因果依赖关系, 进一步展开:

$$\begin{aligned} ELBO = & \log p(s_0) + \sum_{t=0}^T \mathbb{E}_{h_t \sim q_{\varphi}(h_t | s_t, a_t, s_{t+1})} \log p_{\psi}(s_{t+1} | s_t, a_t, h_t) - \sum_{t=0}^T D_{\text{KL}}(q_{\varphi}(h_t | s_t, a_t, s_{t+1}) \| p(h_t)) \\ & + \sum_{t=0}^T \log \pi_{\theta_a}(a_t | s_t) + \sum_{t=0}^T \log p_R(r_t | s_t, a_t, s_{t+1}) \end{aligned} \quad (12)$$

因为 $p(s_0)$ 为样本给定的固定数值、 $\pi_{\theta_a}(a_t | s_t)$ 由动作网络决定、 $p_R(r_t | s_t, a_t, s_{t+1})$ 由环境奖励机制决定, 都与因果编码网络或因果解码网络无关, 最大化 *ELBO* 调整为:

$$\begin{aligned}
\max(ELBO) &= \min(-ELBO) \\
&= \min \left(\sum_{t=0}^T D_{\text{KL}}(q_{\varphi}(h_t|s_t, a_t, s_{t+1}) \| p(h_t)) - \sum_{t=0}^T \mathbb{E}_{h_t \sim q_{\varphi}(h_t|s_t, a_t, s_{t+1})} \log p_{\psi}(s_{t+1}|s_t, a_t, h_t) \right) \\
&= \sum_{t=0}^T \min \left[D_{\text{KL}}(q_{\varphi}(h_t|s_t, a_t, s_{t+1}) \| p(h_t)) - \mathbb{E}_{h_t \sim q_{\varphi}(h_t|s_t, a_t, s_{t+1})} \log p_{\psi}(s_{t+1}|s_t, a_t, h_t) \right]
\end{aligned} \quad (13)$$

最大化 $ELBO$ 分成各个时间点内单独控制每一步损失, 则因果模型的损失为:

$$\mathcal{L}_{\text{causal}} = D_{\text{KL}}(q_{\varphi}(h_t|s_t, a_t, s_{t+1}) \| p(h_t)) - \mathbb{E}_{h_t \sim q_{\varphi}(h_t|s_t, a_t, s_{t+1})} \log p_{\psi}(s_{t+1}|s_t, a_t, h_t) \quad (14)$$

其中, 等式右侧第 1 项是近似后验分布 (由因果编码网络推断) 和假设先验分布 (一般假设为高斯分布) 的 KL 散度. 最小化 KL 散度项可以促使因果编码网络学习的隐变量分布尽可能接近先验分布, 同时起到正则化优化问题的作用, 保持参数推断的不确定性. 等式右侧第 2 项是在近似后验分布下对隐藏变量 h_t 进行采样, 并计算因果编码网络中 $p_{\psi}(s_{t+1}|s_t, a_t, h_t)$ 的对数概率负期望值. 最小化对数概率负期望项, 可以使因果编码网络和因果解码网络协同学习, 在给定隐变量的情况下, 使重构数据尽可能接近真实的可观测状态信息, 保持模型对数据的解释能力. 通过平衡这两部分, 保证了变分推断和因果模型拟合的有效性, 进而确保算法可推断因果价值函数所需要的隐变量.

假设隐变量的先验分布服从各向同性多元高斯分布, 即 $p(h_t) = \mathcal{N}(h_t; 0, I)$. 同时, 也假设真实后验分布也满足对角协方差矩阵的多元高斯分布. 通过向因果编码网络输入采样批次样本对应的 $(s_t^{(j)}, a_t^{(j)}, s_{t+1}^{(j)})$ 信息, 获得对应情况的后验分布的参数, 如均值为 $\mu_t^{(j)}$ 、标准差为 $\sigma_t^{(j)}$. 相应地, 基于数量为 J 的采样批次样本, 因果编码网络的输出通过重参数化技巧整理为近似后验分布:

$$q_{\varphi}(h_t^{(j)}|s_t^{(j)}, a_t^{(j)}, s_{t+1}^{(j)}) = \mathcal{N}(h_t^{(j)}; \mu_t^{(j)}, [\sigma_t^{(j)}]^2 I) \quad (15)$$

在公式 (15) 中, 重参数化技巧将隐变量 $h_t^{(j)}$ 的采样过程重新表述为一个确定性函数和一个独立的噪声变量的函数: $h_t^{(j)} = \mu_t^{(j)} + \sigma_t^{(j)} \odot \varepsilon_t^{(j)}$. 其中, $\varepsilon_t^{(j)}$ 是从标准正态分布 $\mathcal{N}(0, 1)$ 中采样的噪声变量. 故因果模型的损失中, 近似后验分布 $q_{\varphi}(h_t^{(j)}|s_t^{(j)}, a_t^{(j)}, s_{t+1}^{(j)})$ 和假设先验分布 $p(h_t)$ 的 KL 散度项整理为:

$$D_{\text{KL}}(q_{\varphi}(h_t^{(j)}|s_t^{(j)}, a_t^{(j)}, s_{t+1}^{(j)}) \| p(h_t)) = -\frac{1}{2} \sum_{j=1}^J \left(1 + \log \left\{ [\sigma_t^{(j)}]^2 \right\} - [\mu_t^{(j)}]^2 - [\sigma_t^{(j)}]^2 \right) \quad (16)$$

而对于因果解码网络, 直接优化对数概率负期望值在计算上比较复杂. 在已知可观测状态信息维度为 d 时, 借助真实后验分布 (已假设为高斯分布) 的概率密度函数, 可以如公式 (17) 将对数概率负期望项简化为重构损失项:

$$\begin{aligned}
-\mathbb{E}_{h_t \sim q_{\varphi}(h_t|s_t, a_t, s_{t+1})} \log p_{\psi}(s_{t+1}|s_t, a_t, h_t) &= -\mathbb{E}_{h_t \sim q_{\varphi}(h_t|s_t, a_t, s_{t+1})} \log \left[\frac{1}{\sqrt{(2\pi)^d \sigma^2}} \exp \left(-\frac{1}{2\sigma^2} \|s_{t+1} - F_{\psi}(s_t, a_t, h_t)\|^2 \right) \right] \\
&= \mathbb{E}_{h_t \sim q_{\varphi}(h_t|s_t, a_t, s_{t+1})} \left[\frac{d}{2} \log(2\pi) + d \log \sigma + \frac{1}{2\sigma^2} \|s_{t+1} - F_{\psi}(s_t, a_t, h_t)\|^2 \right] \\
&\propto \frac{1}{J} \sum_{j=1}^J \|s_{t+1}^{(j)} - F_{\psi}(s_t^{(j)}, a_t^{(j)}, h_t^{(j)})\|^2
\end{aligned} \quad (17)$$

学习隐变量因果模型的过程, 如算法 1 所示.

算法 1. 隐变量因果模型学习算法.

输入: 缓存池 β , 采样批次样本量 J , 迭代次数 I , 因果编码网络 F_{φ} , 因果解码网络 F_{ψ} , 因果模型的 KL 散度项权重 w_{KL} , 因果模型的重构损失项权重 w_{recon} ;

输出: 隐变量因果模型更新后的因果编码网络参数 φ , 因果解码网络参数 ψ , 隐变量信息 h .

1. 初始化 F_φ, F_ψ
2. **for** $i = 1 : I$ **do**
3. 从缓冲池 β 中采样 $\{(s_t^{(j)}, a_t^{(j)}, s_{t+1}^{(j)}) | j \in [1, J]\}$
4. **for** $j = 1 : J$ **do**
5. 推断隐变量均值和标准差: $\mu_t^{(j)}, \sigma_t^{(j)} = F_\varphi(s_t^{(j)}, a_t^{(j)}, s_{t+1}^{(j)})$
6. 重参数化: $h_t^{(j)} \sim \mathcal{N}(\mu_t^{(j)}, [\sigma_t^{(j)}]^2 I)$
7. 重构可观测状态信息: $s_{t+1}^{(j)} = F_\psi(s_t^{(j)}, a_t^{(j)}, h_t^{(j)})$
8. **end for**
9. 计算 KL 散度项: $\mathcal{L}_{\text{KL}} = -\frac{1}{2} \sum_{j=1}^J \left(1 + \log \{[\sigma_t^{(j)}]^2\} - [\mu_t^{(j)}]^2 - [\sigma_t^{(j)}]^2 \right)$
10. 计算重构损失项: $\mathcal{L}_{\text{recon}} = \frac{1}{J} \sum_{j=1}^J \|s_{t+1}^{(j)} - F_\psi(s_t^{(j)}, a_t^{(j)}, h_t^{(j)})\|^2$
11. 根据联合损失项 $\mathcal{L}_{\text{causal}} = w_{\text{KL}} \mathcal{L}_{\text{KL}} + w_{\text{recon}} \mathcal{L}_{\text{recon}}$, 更新 φ 和 ψ
12. **end for**

4.2 基于隐变量因果模型的策略梯度算法

基于隐变量因果模型的策略梯度算法的网络训练方法如图 3 所示. 在因果价值网络判定价值之前, 利用长短期记忆网络 (long short-term memory, LSTM) 处理逆序的变长隐变量序列 h_t^+ , 学出隐变量序列特征. 在时间序列中距离当前节点越远的隐变量信息对当前节点价值 V_t 的影响应该越小, 这样的设计与折扣因子的设计是相似的. 而 LSTM 网络处理逆序的变长隐变量序列 h_t^+ , 恰能通过遗忘门将类似的记忆特点表现出来. 结合上述的隐变量序列特征和可观测状态信息, 计算基于当前因果价值网络对当前时刻预估的价值信息. 通过因果价值网络更为准确地衡量当前时刻的综合情况, 可以使动作优势估计值不受动态环境中未观测随机信息的干扰, 进而降低策略梯度的方差. 基于上述描述, 基于隐变量因果模型的策略梯度算法如算法 2 所示.

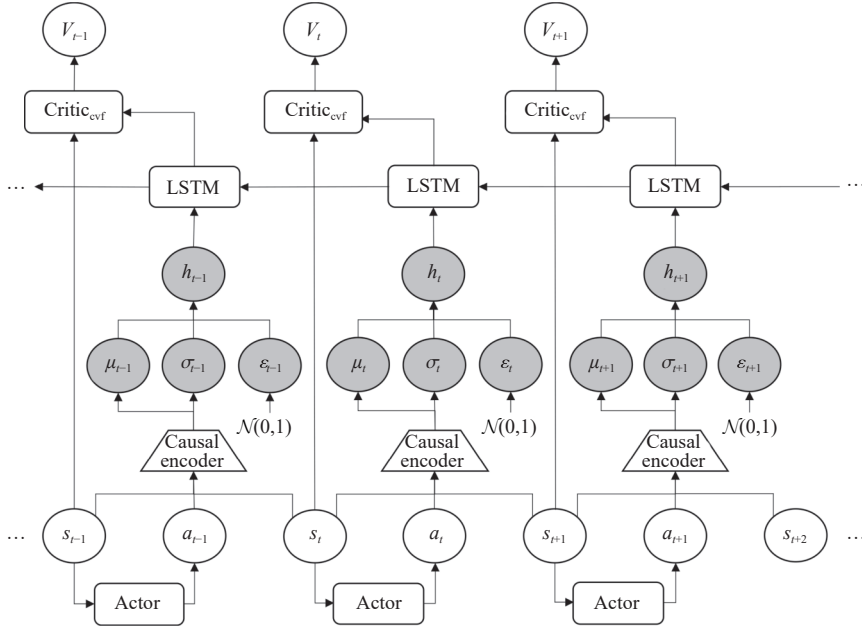


图 3 基于隐变量因果模型的策略梯度算法的网络训练方法示意图

算法 2. 基于隐变量因果模型的策略梯度算法的网络训练方法.

输入: L 条完整轨迹 $\left\{ \tau^{(0)}, \tau^{(1)}, \dots, \tau^{(L)} \mid \tau^{(l)} = \{s_0^{(l)}, a_0^{(l)}, r_0^{(l)}, s_1^{(l)}, a_1^{(l)}, r_1^{(l)}, \dots, s_T^{(l)}, a_T^{(l)}, r_T^{(l)}, s_{T+1}^{(l)}\} \right\}$, 强化样本采样比例 ρ , 迭代次数 I , 折扣因子 γ , 因果价值网络 $F_{\theta_{CVF}}$, 动作网络 F_{θ_a} , 熵的损失加权 ω_e , 裁剪系数 δ , 因果编码网络 F_φ , 已完成训练的因果编码网络参数 φ ;

输出: 因果价值网络参数 θ_{CVF} , 动作网络参数 θ_a .

```

1. 初始化  $F_{\theta_{CVF}}, F_{\theta_a}$ 
2. for  $l = 1 : L$  do
3.   for  $t = 0 : T$  do
4.     推断隐变量均值和标准差:  $\mu_t^{(l)}, \sigma_t^{(l)} = F_\varphi(s_t^{(l)}, a_t^{(l)}, s_{t+1}^{(l)})$ 
5.     重参数化:  $h_t^{(l)} \sim \mathcal{N}(\mu_t^{(l)}, [\sigma_t^{(l)}]^2 I)$ 
6.     计算更新前动作对数概率和熵:  $\log pa\_old_t^{(l)}, e\_old_t^{(l)} = F_{\theta_a}(s_t^{(l)})$ 
7.     计算更新前价值评估:  $V\_old_t^{(l)} = F_{\theta_a}(s_t^{(l)}, h_t^{(l)+})$ 
8.     计算轨迹回报:  $G_t^{(l)} = \sum_{r'=t}^T \gamma^{r'-t} r_{r'}^{(l)}$ 
9.     计算动作优势估计值:  $A_t^{(l)} = G_t^{(l)} - V\_old_t^{(l)}$ 
10.   end for
11. end for
12. for  $i = 1 : I$  do
13.   for  $l = 1 : L$  do
14.     for  $t = 0 : T$  do
15.       更新动作对数概率  $\log pa_t^{(l)}$  和熵  $e_t^{(l)}$ :  $\log pa_t^{(l)}, e_t^{(l)} = F_{\theta_a}(s_t^{(l)})$ 
16.       更新价值评估:  $V_t^{(l)} = F_{\theta_a}(s_t^{(l)}, h_t^{(l)+})$ 
17.     end for
18.   end for
19.   从  $L$  条完整轨迹的数据中采样:  $\{A_1^{(\rho)}, \dots, A_K^{(\rho)}\}, \{\log pa\_old_1^{(\rho)}, \dots, \log pa\_old_K^{(\rho)}\}, \{\log pa_1^{(\rho)}, \dots, \log pa_K^{(\rho)}\}, \{e_1^{(\rho)}, \dots, e_K^{(\rho)}\}, \{G_1^{(\rho)}, \dots, G_K^{(\rho)}\}, \{V_1^{(\rho)}, \dots, V_K^{(\rho)}\}$ 
20.   计算新旧策略比率为:  $\alpha_k = \exp(\log pa\_old_k^{(\rho)} - \log pa_k^{(\rho)})$ 
21.   计算动作网络损失项:  $\mathcal{L}_a = -\frac{1}{K} \sum_{k=1}^K \{\min[\alpha_k, clip(\alpha_k, 1 - \delta, 1 + \delta)] A_k^{(\rho)} + \omega_e e_k^{(\rho)}\}$ 
22.   计算因果价值网络损失项:  $\mathcal{L}_v = \frac{1}{K} \sum_{k=1}^K \|G_k^{(\rho)} - V_k^{(\rho)}\|^2$ 
23.   根据动作网络损失项, 更新  $\theta_a$ 
24.   根据因果网络损失项, 更新  $\theta_{CVF}$ 
25. end for

```

5 实验结果与分析

5.1 实验环境

本文选择使用 OpenAI Gym 平台中的 3 个经典环境: Lunar Lander、Thrower 和 Pusher 作为实验环境, 如图 4 所示. OpenAI Gym 是一个流行的深度强化学习环境平台, 提供了丰富多样的环境供研究者开展实验, 涵盖了从基

础控制任务到复杂模拟环境的多种场景。

Lunar Lander 环境模拟了一艘月球着陆器在二维空间中通过控制其主引擎和侧推器实现平稳着陆的任务。该环境具有高维度的观测空间和离散的动作空间, 其奖励函数设计如下: 当着陆器成功着陆时可获得+100 的奖励, 着陆器保持静止状态可获得+10 的奖励, 每个着陆腿与地面接触可获得+10 的奖励。此外, 奖励值会随着着陆器与着陆台之间的距离动态变化, 距离越近奖励越高; 同时着陆器的运动速度也会影响奖励值, 速度越慢奖励越高。若发生坠毁则获得-100 的惩罚, 且燃料消耗会产生相应的负向奖励。本文选用的 Lunar Lander 环境具有内置的风力干扰机制。通过配置 $wind_power = 1.0$ 及 $turbulence_power = 1.5$ 参数引入显著的环境动态随机性, 构建基准实验环境用于评估各算法的性能表现。

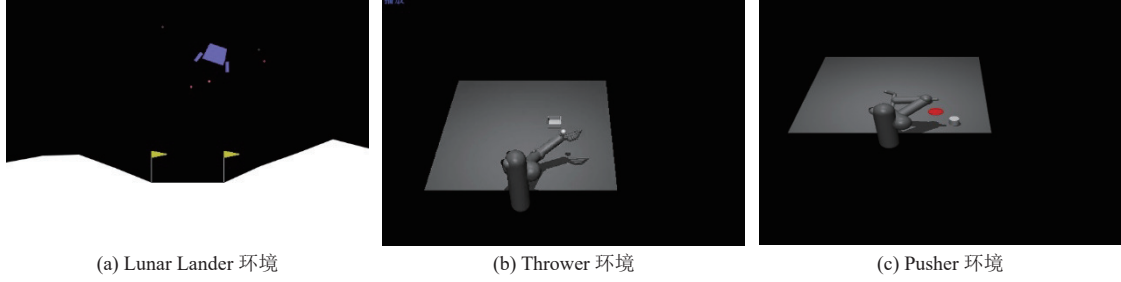


图 4 OpenAI Gym 实验环境

Thrower 环境是一个基于 MuJoCo 物理引擎的三维任务, 模拟了一台机械臂将物体抛向指定目标区域。该任务需要控制机械臂的多个关节, 以生成合适的运动轨迹, 使得物体能够精确地到达目标区域。环境的奖励函数主要由物体与目标的欧氏距离的负值构成, 即距离目标越近奖励越高。本文在 Thrower 环境的目标位置中增加均值为 0, 标准差为 0.01 的高斯分布的随机变量, 以引入动态环境的随机性, 使实验环境具备较为明显的动态随机特性。

Pusher 环境同样基于 MuJoCo 物理引擎, 模拟了一台机械臂将物体推向目标位置的任务。Pusher 环境具有与 Thrower 环境相同的状态空间和动作空间结构, 但其奖励函数由机械臂末端与物体的距离、物体与目标的距离两部分的负值构成。本文也向 Pusher 环境的目标状态增加分布为 $\mathcal{N}(0, 0.01)$ 的动态环境随机信息, 使实验环境存在较为明显的动态环境随机信息。

通过比较在这些环境下使用 PPO^[19]、HVF^[17]和 PGVO 的表现, 可以更全面地评估 PGVO 的适用性和鲁棒性。

5.2 参数设置

本文在不同实验环境下, 使用 5 个随机种子 (12, 34, 56, 78, 90) 对各算法进行对比测试。在离散环境 Lunar Lander 测试 5000 回合, 连续环境 Thrower 和 Pusher 测试 20000 回合, 每 100 个回合取奖励的平均值作为评估结果。

PGVO 的因果编码器 (causal encoder) 和因果解码器 (causal decoder) 均采用多层感知机 (multilayer perceptron, MLP) 实现。其中, 因果编码器以动作维度与两倍状态维度之和作为输入, 通过双输出结构分别输出均值和对数方差, 输出维度均为隐变量维度。基于该结构, 编码器利用重参数化技巧实现随机采样。因果解码器以状态维度、动作维度及隐变量维度之和作为输入, 输出维度与状态维度相同。动作网络以可观测状态作为输入, 通过 MLP 输出动作空间维度的动作值。价值网络采用双流架构, 包含状态处理和隐变量序列处理两个分支。其中, 状态处理分支通过 MLP 提取可观测状态特征, 隐变量序列处理分支利用 LSTM 网络提取隐变量序列特征。最终将两个分支的特征拼接, 通过 MLP 网络输出标量价值估计。

为保证实验的公平性和可比性, 本文统一配置了各算法的网络结构和超参数。PPO 的动作网络结构与 PGVO 保持一致, 其价值网络直接以可观测状态作为输入。HVF 的后见向量 (hindsight vector) 维度与 PGVO 的隐变量维度相同, 其动作网络和价值网络也和 PGVO 使用相同的网络架构。具体超参数配置如后文表 1 所示。

5.3 实验结果

经过第 3.2 节探讨, 动态环境中的未观测随机信息会导致策略梯度估计存在较大方差。这一问题主要表现为:

其一, 训练过程不稳定, 参数更新方向存在较大波动, 影响策略的收敛速度; 其二, 过大的方差可能使策略更新偏离最优方向, 导致算法无法收敛至理想性能. 针对环境随机性引起的策略梯度高方差问题, 本文重点关注以下两个性能指标: 策略的累计奖励达到目标性能阈值所需训练回合数和策略收敛后的累计奖励. 对于目标性能阈值的设定, Lunar Lander 环境已明确规定累计奖励达到 200 即表示任务完成, 因此直接采用该值作为阈值. 由于 Thrower 和 Pusher 环境未设定明确的任务完成标准, 本文将 PPO 算法收敛后的平均奖励值向下取整至最近的 5 的倍数, 即分别以-40 和-50 作为这两个环境的目标性能阈值.

表 1 基于隐变量因果模型的策略梯度算法超参数

超参数	取值	参数描述
β	10 000	因果编解码框架的缓存池容量
J	128	因果编解码框架单轮优化网络批次采样数
ρ	0.8	强化样本采样比例
w_{KL}	0.7	因果编解码框架的KL散度项权重
w_{recon}	0.3	因果编解码框架的重构损失项权重
w_e	0.01	策略的熵权重
γ	0.99	强化折扣因子
δ	0.1	近端策略裁剪系数
lr_{causal}	0.000 3	因果编解码框架的学习率
lr_a	0.000 3	动作网络的学习率
lr_v	0.000 05	因果价值网络的学习率
d_h	4	隐变量的信息维度
d_s	256	价值网络状态处理分支的可观测状态特征的维度
d_o	64	价值网络隐变量序列处理分支的隐变量序列特征的维度

表 2 和图 5 分别展示了 PPO、HVF 和 PGVO 这 3 种算法在不同环境下的性能指标结果. 表 2 记录了累计奖励达到目标性能阈值所需训练回合数、累计奖励均值 ($\pm$ 标准差) 和累计奖励中值, 其中性能优势指标以粗体标注. 对于未能在规定回合数内达到目标性能阈值的算法, 表 2 中标记为“×”. 图 5 用虚线标识达标回合数并在其与累积奖励曲线的交点处标注“(达标回合数, 达标时累计奖励)”, 未达标算法则不显示虚线.

表 2 OpenAI Gym 实验不同算法的最终性能表现

性能指标	Lunar Lander <i>wind power=1</i>			Thrower $h \sim \mathcal{N}(0, 0.01)$			Pusher $h \sim \mathcal{N}(0, 0.01)$		
	PPO	HVF	PGVO	PPO	HVF	PGVO	PPO	HVF	PGVO
累计奖励达到目标性能阈值所需训练回合数	4098	×	1555	6523	489	496	11 503	6459	5564
累计奖励均值 ($\pm$ 标准差)	197.69 (± 13.65)	57.39 (± 129.21)	238.89 (± 25.78)	-39.11 (± 0.98)	-34.47 (± 1.65)	-30.91 (± 0.91)	-46.88 (± 2.21)	-47.88 (± 5.01)	-42.94 (± 1.74)
累计奖励中值	201.37	9.35	247.28	-38.76	-34.36	-30.90	-47.21	-49.39	-42.63

在 Lunar Lander 环境中, PGVO 达到目标性能阈值所需的训练回合数从 PPO 的 4098 回合减少至 1555 回合 (减少 62.05%). 实验表明, 训练完成时 PGVO 在累计奖励的均值和中值方面均显著优于对比算法, 具体而言, 均值较 PPO 提升 41.2 (提升 20.84%), 中值较 PPO 提升 45.91 (提升 22.8%). 值得注意的是, 尽管 PPO 的标准差较低, 但 PGVO 在最大方差情况下的性能仍优于 PPO 的最佳表现. 在价值网络损失方面, PGVO 呈现出最为显著的前期下降趋势, 这从侧面验证了其能以最少训练回合数达到目标性能阈值的合理性. 相比之下, HVF 虽然价值网络损失下降迅速, 但在经过 5000 回合训练后累计奖励仍未达到 200 点的目标阈值. 深入分析表明, 这可能源于 HVF 未能有效捕捉动态环境的因果结构, 在还原额外信息时过度依赖奖励信息, 导致价值网络过拟合. 此外, 作为降低策略梯度方差的算法, PGVO 在网络优化时间上明显优于 HVF.

在 Thrower 环境中, PGVO 达到目标性能阈值所需的训练回合数从 PPO 的 6523 回合显著降低至 496 回合

(减少 92.4%). 实验数据表明, 训练结束时 PGVO 的累计奖励均值较 PPO 提升 8.2 (增长 20.97%), 较 HVF 提升 3.56 (增长 10.33%); 中值较 PPO 提升 7.86 (增长 20.28%), 较 HVF 提升 3.46 (增长 10.07%). 需要指出的是, HVF 虽在达到阈值的回合数上略优于 PGVO (减少 1.41%), 但其最终累计奖励表现不及 PGVO (降低 11.5%). 通过价值网络收敛特性分析, PGVO 较 PPO 表现出更快的下降速度, 较 HVF 表现出更小的稳态振荡. 同时, 网络优化时间的对比结果进一步验证了 PGVO 的优越性.

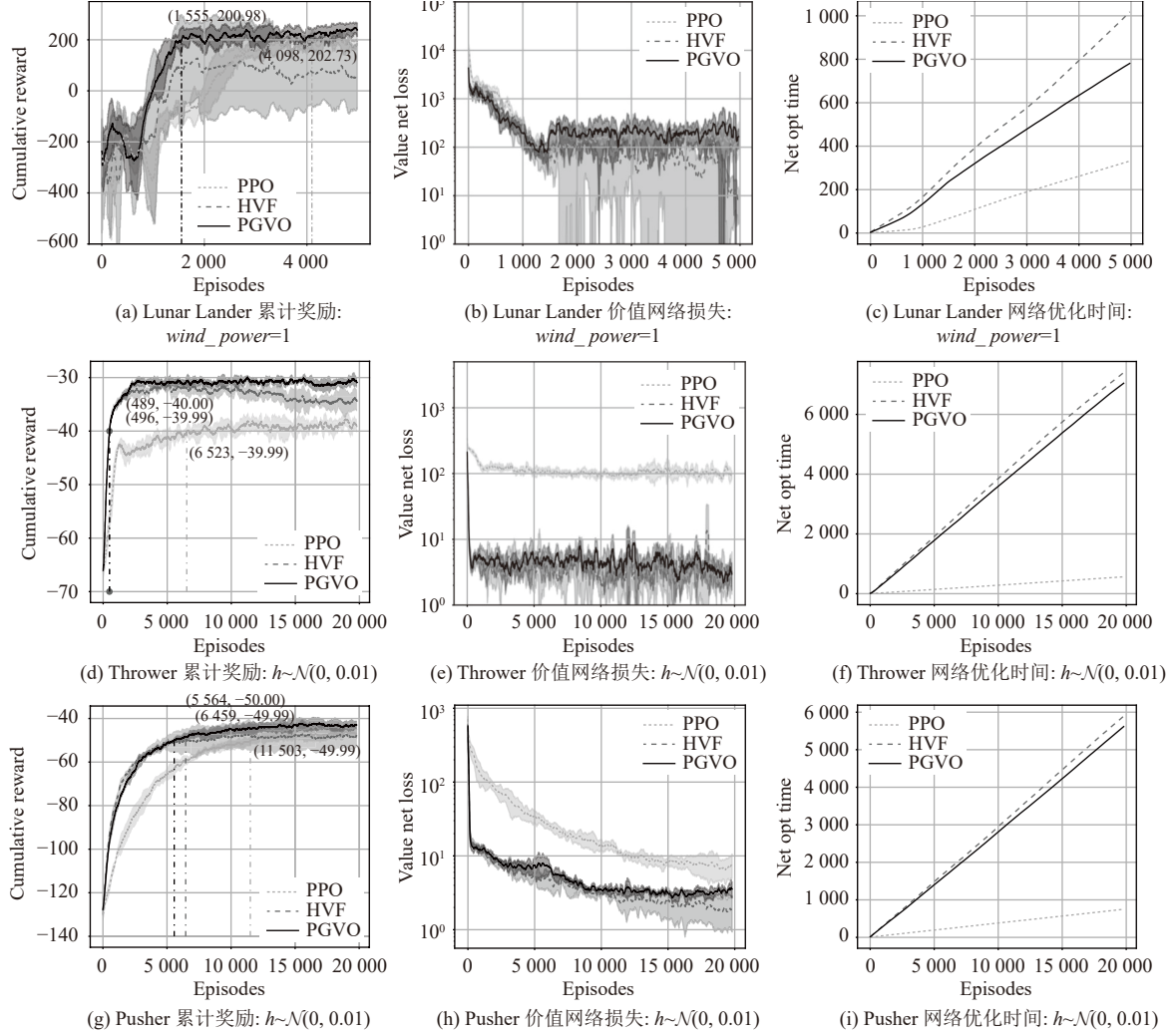


图5 OpenAI Gym 训练过程中累计奖励、网络优化时间和价值损失的可视化曲线

在 Pusher 环境中, PGVO 达到目标性能阈值所需的训练回合数为 5564 回合, 相较于 PPO 的 11503 回合和 HVF 的 6459 回合分别减少 51.6% 和 13.86%. 实验结果表明, 训练完成时 PGVO 的累计奖励均值较 PPO 提升 3.94 (增长 8.4%), 较 HVF 提升 4.94 (增长 10.32%); 中值较 PPO 提升 4.58 (增长 9.7%), 较 HVF 提升 6.76 (增长 13.69%). 通过价值网络损失分析可知, PGVO 不仅具有比 PPO 更快的收敛速度, 且在稳定后较 HVF 呈现更小的方差. 此外, 在网络优化时间方面, PGVO 相较 HVF 也具有一定优势.

通过对 3 个环境的实验分析可知, PGVO 在训练效率、性能稳定性和优化效率等方面均表现出明显优势. 这主要得益于 PGVO 能够通过隐变量因果模型有效刻画动态环境中的数据生成机制, 以此推断未观测随机信息, 进而降低策略梯度优化的方差, 加速策略梯度算法收敛并提升其累计奖励表现.

5.4 消融实验

为了评估本文方法的稳定性和有效性, 基于上述实验, 在 Lunar Lander 环境和 Thrower 环境做了未观测随机信息消融实验, 测试了算法面向不同强度的动态环境随机信息的性能.

如图 6 所示, 在 Lunar Lander 环境中, 面对不同强度的动态环境随机环境信息的影响, PGVO 表现出稳定的优越性能, 是唯一一个在所有不同强度风力影响下能在 5000 回合内达到目标性能阈值的算法. 实验观察到随着风力增大, PGVO 和 PPO 需要更多回合进行训练, 这符合客观规律. 在价值网络损失方面, HVF 始终存在过拟合现象, 而 PGVO 的价值网络损失在训练前期表现出比 PPO 更快的下降速度, 有助于强化算法性能的快速收敛.

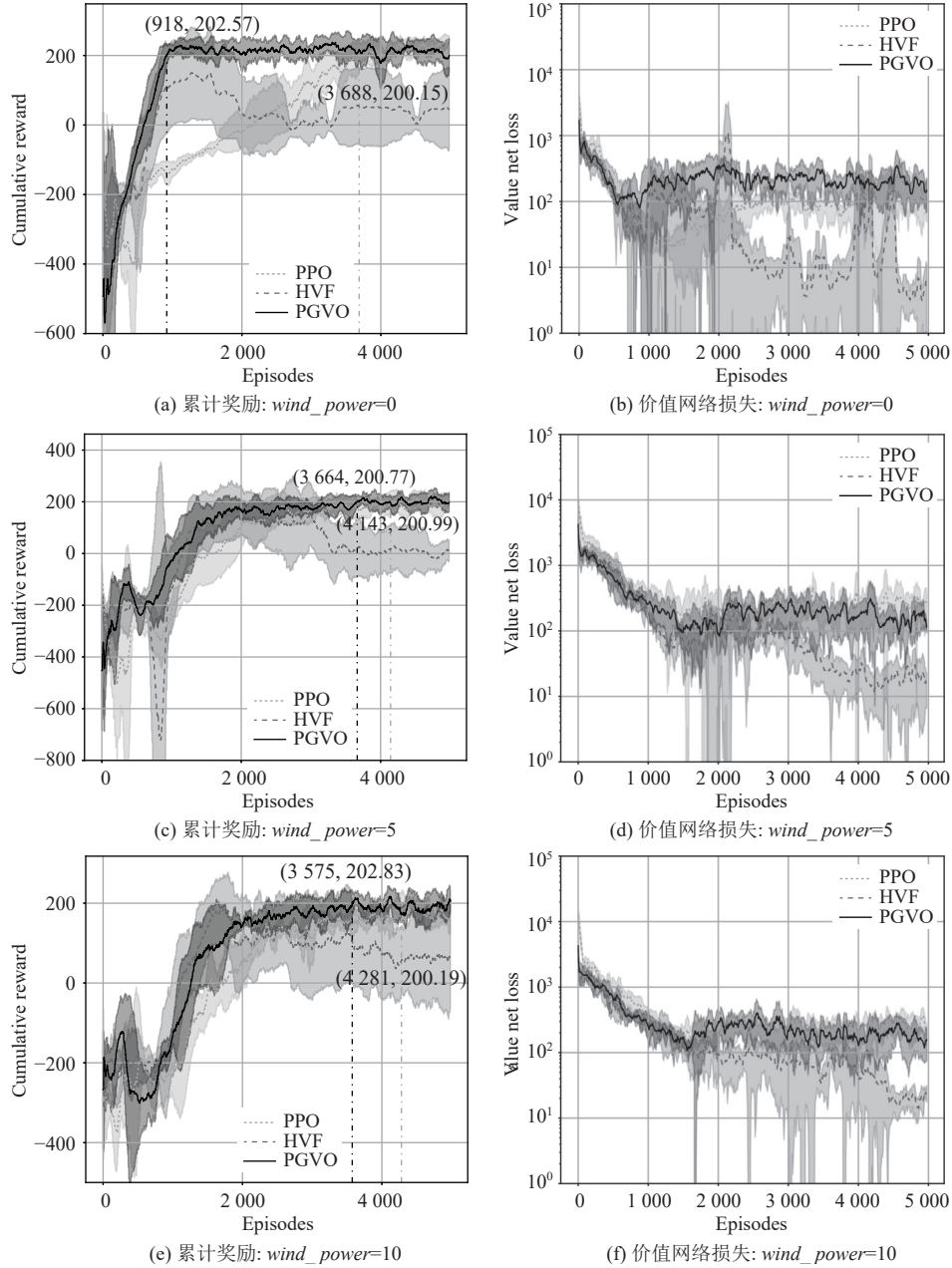


图 6 Lunar Lander 环境随机信息消融的性能曲线

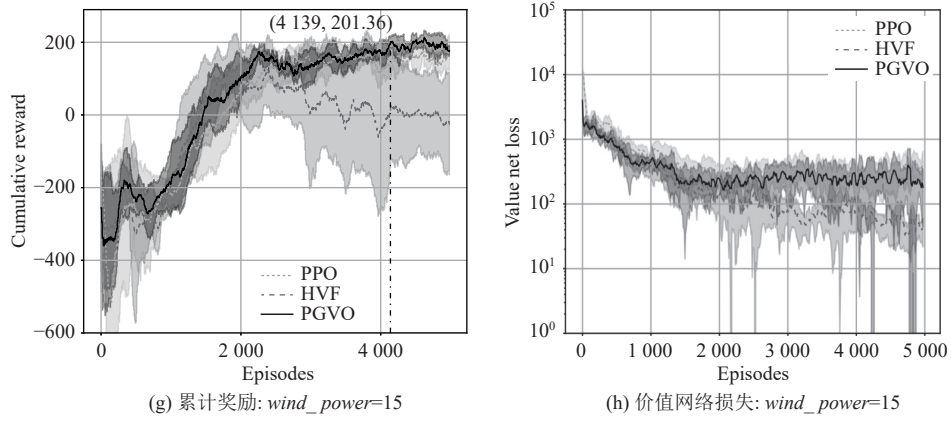


图 6 Lunar Lander 环境随机信息消融的性能曲线 (续)

如图 7 所示, 在 Thrower 环境中, 面对不同强度的动态环境随机信息影响, PGVO 展现出最稳定的优化性能, 且最终收敛的累计奖励最优. 虽然其达到目标性能阈值所需的训练回合并非在所有情况下最少, 但从整体来看, 当 HVF 具有优势时差距并不显著; 而在强干扰环境中, PGVO 较 HVF 减少了 55.6% 的训练回合. 在价值网络损失方面, PGVO 和 HVF 均比 PPO 表现出更快的损失下降速度. 在保持较低价值网络损失值的同时, PGVO 在抖动控制上明显优于 HVF. 总结以上随机信息消融实验, 可以证明本算法面向不同强度的动态环境随机信息, 都能表现出较为稳定的性能, 加快深度强化学习的收敛速度和提高收敛时的累计奖励.

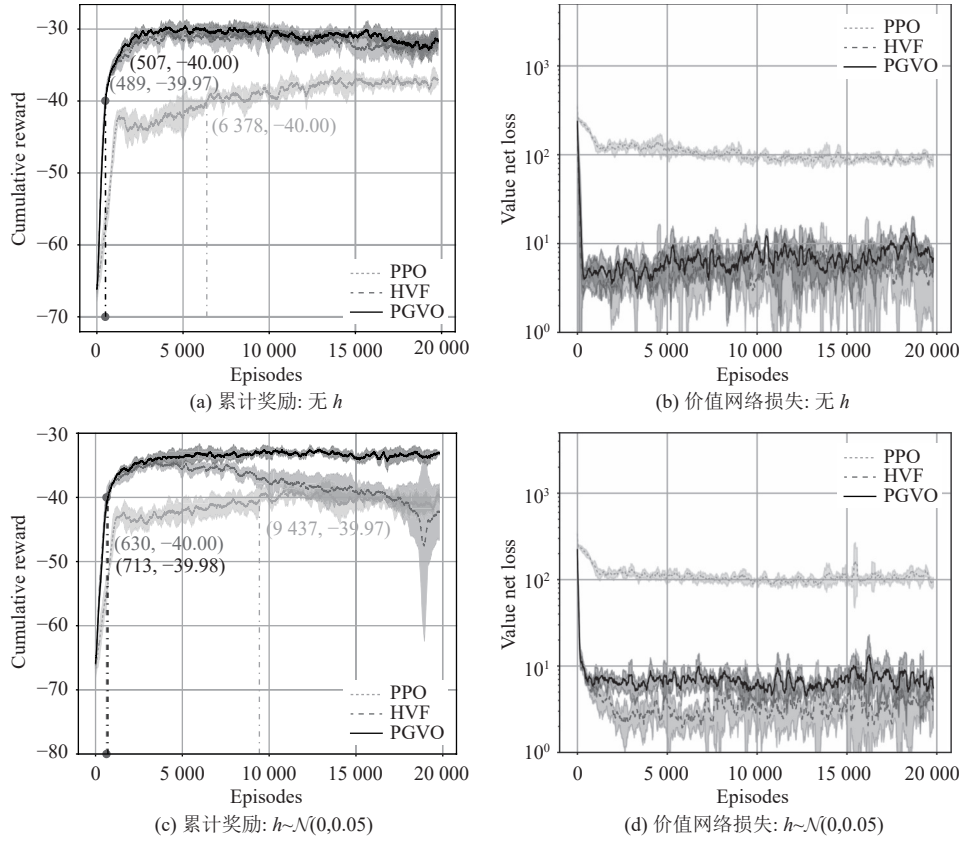


图 7 Thrower 环境随机信息消融的性能曲线

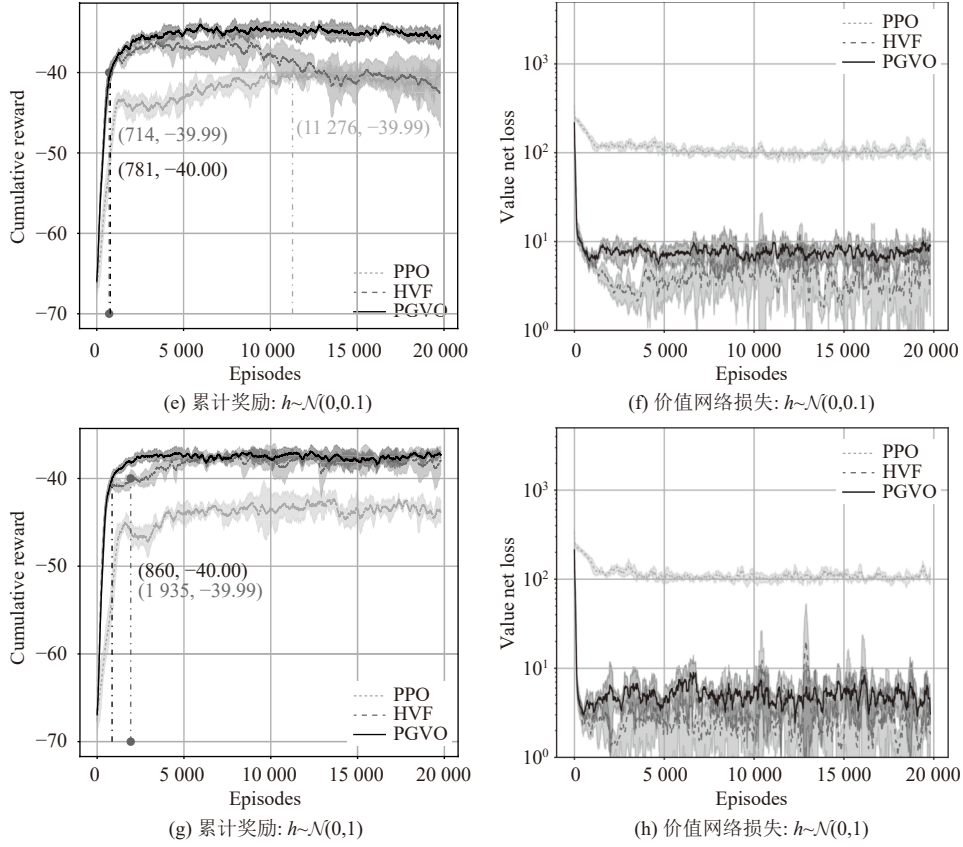


图7 Throwing环境随机信息消融的性能曲线(续)

5.5 参数敏感实验

为了测试算法受模型参数的影响状况,本节测试了不同参数权重对性能带来的影响.在动态环境随机信息由高斯分布 $\mathcal{N}(0, 0.01)$ 产生的 Throwing 环境中,图 8(a)、(b) 测试参数为因果模型的缓存池容量 β 、因果模型单轮优化网络批次采样数 J ,简写成“CVF_buf_{\beta}_{J}”;图 8(c)、(d) 测试参数为隐变量的信息维度 d_h ,简写成“hdim_{d_h}”;图 8(e)、(f) 测试参数为因果模型的重构损失项权重 w_{recon} 和因果模型的 KL 散度项权重 w_{KL} ,简写为“CVF_loss_{w_{\text{recon}}}_{w_{\text{KL}}}”;图 8(g)、(h) 测试参数为近端策略裁剪系数 δ ,简写为“CVF_ppoclip_{\delta}”;图 8(i)、(j) 测试参数为强化折扣因子 γ ,简写为“CVF_gamma_{\gamma}”。

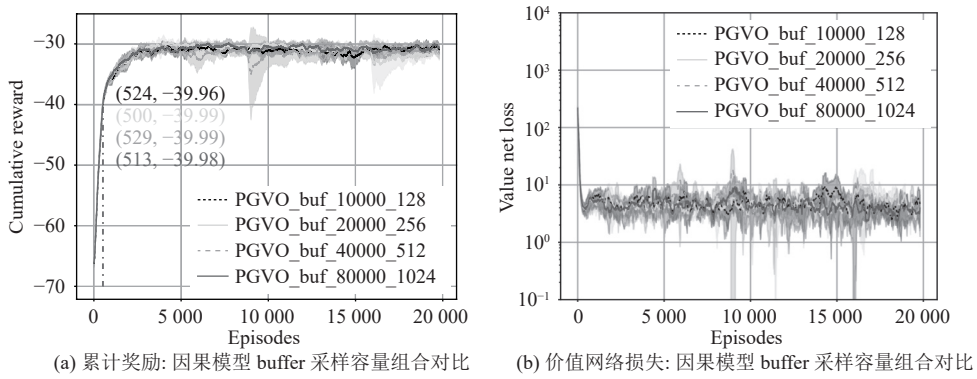


图8 Throwing环境噪声消融的性能曲线

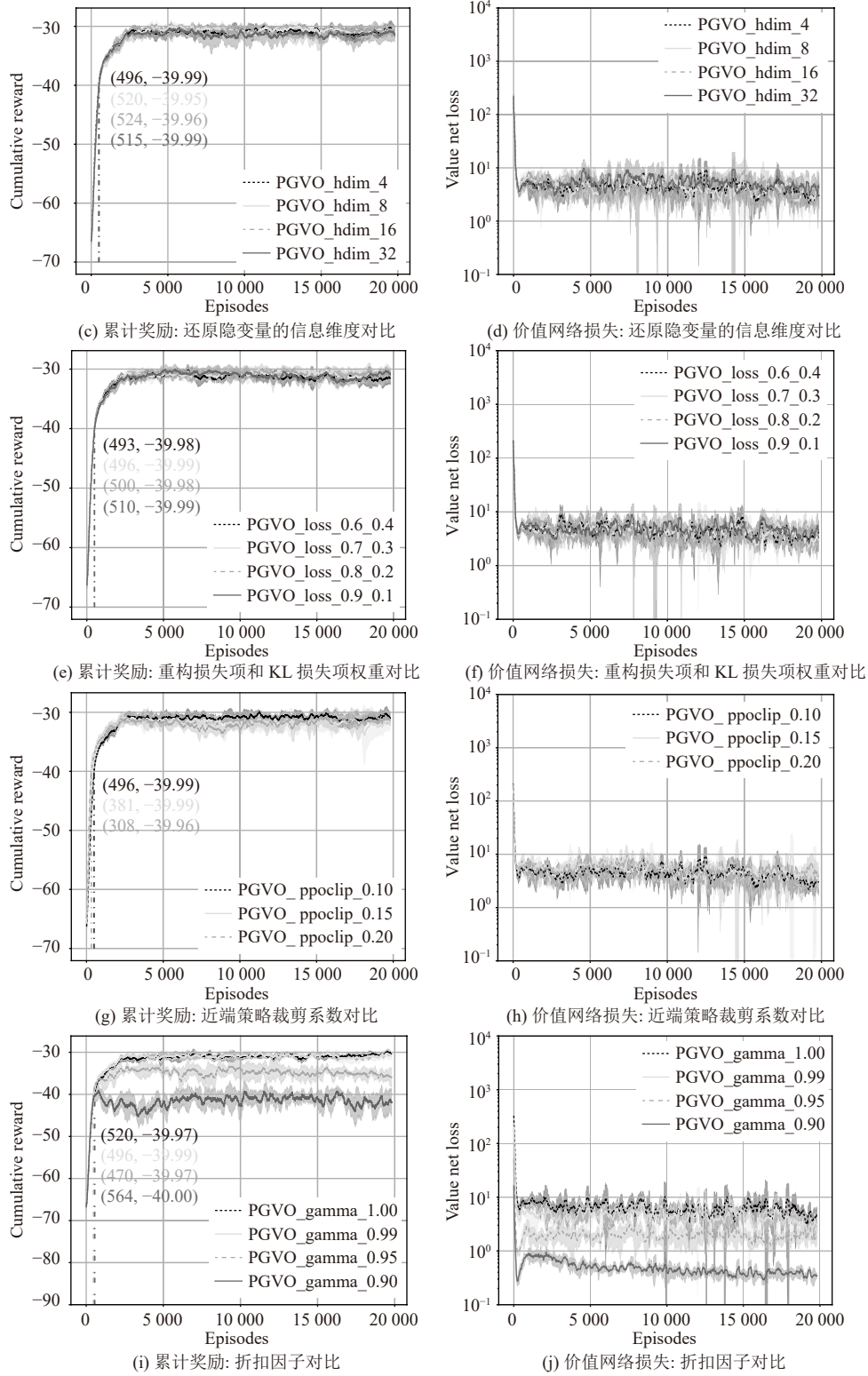


图 8 Throwing 环境噪声消融的性能曲线 (续)

总结参数影响实验, 如图 8 所示可以发现本算法在合理的超参设置范围内, 都能表现出较为稳定的良好性能. 更多需要关注的是经典强化部分的超参设置, 如折扣因子这类的超参设置. 在折扣因子的超参设置上, 使用较高值的折扣因子表现会更优秀, 如 1、0.99.

6 总 结

本文针对动态环境中存在随机信息的情况, 导致现有策略梯度方法存在的高方差问题, 从隐因果模型的视角提出了一种策略梯度方差优化方法 (PGVO), 该方法通过引入显变量和隐变量分别刻画可观测状态信息和动态环境随机信息, 分析并构建隐变量因果模型, 然后根据该模型提出因果价值函数. 该价值函数估计考虑了环境随机信息带来的干扰, 以其为基线计算动作优势估计值, 可有效降低动态环境随机信息对策略梯度算法表现性能的影响. 上述步骤的求解利用因果编解码框架推断动态环境随机信息, 并结合长短期记忆网络处理变长的动态环境随机信息, 借助因果价值函数结合可观测状态信息进行价值函数估计. 实验结果表明, 本文方法在不同随机种子和不同强度的动态环境随机信息下表现出卓越的性能, 显著提升了策略梯度算法的收敛速度和稳定性. 未来的工作将考虑非稳态等更复杂场景所带来的挑战, 进一步提高策略梯度方法的适用性同时优化方差.

References

- [1] Wang X, Wang S, Liang XX, Zhao DW, Huang JC, Xu X, Dai B, Miao QG. Deep reinforcement learning: A survey. *IEEE Trans. on Neural Networks and Learning Systems*, 2024, 35(4): 5064–5078. [doi: [10.1109/TNNLS.2022.3207346](https://doi.org/10.1109/TNNLS.2022.3207346)]
- [2] Perolat J, de Vyllder B, Hennes D, *et al.* Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 2022, 378(6623): 990–996. [doi: [10.1126/science.add4679](https://doi.org/10.1126/science.add4679)]
- [3] Li QY, Peng ZH, Feng L, Zhang QH, Xue ZH, Zhou BL. MetaDrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2023, 45(3): 3461–3475. [doi: [10.1109/TPAMI.2022.3190471](https://doi.org/10.1109/TPAMI.2022.3190471)]
- [4] Ibarz J, Tan J, Finn C, Kalakrishnan M, Pastor P, Levine S. How to train your robot with deep reinforcement learning: Lessons we have learned. *The Int'l Journal of Robotics Research*, 2021, 40(4–5): 698–721. [doi: [10.1177/0278364920987859](https://doi.org/10.1177/0278364920987859)]
- [5] Lin Q, Yu C, Wu XW, Dong YZ, Xu X, Zhang Q, Guo X. Survey on sim-to-real transfer reinforcement learning in robot systems. *Ruan Jian Xue Bao/Journal of Software*, 2024, 35(2): 711–738 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/7006.htm> [doi: [10.13328/j.cnki.jos.007006](https://doi.org/10.13328/j.cnki.jos.007006)]
- [6] Liu JW, Gao F, Luo XL. Survey of deep reinforcement learning based on value function and policy gradient. *Chinese Journal of Computers*, 2019, 42(6): 1406–1438 (in Chinese with English abstract). [doi: [10.11897/SP.J.1016.2019.01406](https://doi.org/10.11897/SP.J.1016.2019.01406)]
- [7] Shen WG, Huan X. Bayesian sequential optimal experimental design for nonlinear models using policy gradient reinforcement learning. *Computer Methods in Applied Mechanics and Engineering*, 2023, 416: 116304. [doi: [10.1016/j.cma.2023.116304](https://doi.org/10.1016/j.cma.2023.116304)]
- [8] Wu CC, Ruan JG, Cui HH, Zhang B, Li TY, Zhang KX. The application of machine learning based energy management strategy in multi-mode plug-in hybrid electric vehicle, part I: Twin delayed deep deterministic policy gradient algorithm design for hybrid mode. *Energy*, 2023, 262: 125084. [doi: [10.1016/j.energy.2022.125084](https://doi.org/10.1016/j.energy.2022.125084)]
- [9] Zhang JW, Lü S, Zhang ZH, Yu JY, Gong XY. Survey on deep reinforcement learning methods based on sample efficiency optimization. *Ruan Jian Xue Bao/Journal of Software*, 2022, 33(11): 4217–4238 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6391.htm> [doi: [10.13328/j.cnki.jos.006391](https://doi.org/10.13328/j.cnki.jos.006391)]
- [10] Liu H, Socher R, Xiong CM. Taming MAML: Efficient unbiased meta-reinforcement learning. In: *Proc. of the 36th Int'l Conf. on Machine Learning*. Long Beach: PMLR, 2019. 4061–4071.
- [11] Munos R, Stepleton T, Harutyunyan A, Bellemare MG. Safe and efficient off-policy reinforcement learning. In: *Proc. of the 30th Int'l Conf. on Neural Information Processing Systems*. Barcelona: Curran Associates Inc., 2016. 1054–1062.
- [12] Papini M, Binaghi D, Canonaco G, Pirota M, Restelli M. Stochastic variance-reduced policy gradient. In: *Proc. of the 35th Int'l Conf. on Machine Learning*. Stockholm: PMLR, 2018. 4023–4032.
- [13] Nguyen DT, Kumar A, Lau HC. Policy gradient with value function approximation for collective multiagent planning. In: *Proc. of the 31st Int'l Conf. on Neural Information Processing Systems*. Long Beach: Curran Associates Inc., 2017. 4322–4332.
- [14] Thomas PS, Brunskill E. Data-efficient off-policy policy evaluation for reinforcement learning. In: *Proc. of the 33rd Int'l Conf. on Machine Learning*. New York: JMLR.org, 2016. 2139–2148.

- [15] Schulman J, Levine S, Abbeel P, Jordan MI, Moritz P. Trust region policy optimization. In: Proc. of the 32nd Int'l Conf. on Machine Learning. Lille: JMLR.org, 2015. 1889–1897.
- [16] Liu YL, Zhang KQ, Başar T, Yin WT. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 639.
- [17] Guo JM, Zhang R, Zhang XS, Peng SH, Yi Q, Du ZD, Hu X, Guo Q, Chen YJ. Hindsight value function for variance reduction in stochastic dynamic environment. In: Proc. of the 30th Int'l Joint Conf. on Artificial Intelligence. Montreal: ijcai.org, 2021. 2476–2482. [doi: [10.24963/ijcai.2021/341](https://doi.org/10.24963/ijcai.2021/341)]
- [18] Palanisamy P. Hands-on Intelligent Agents with OpenAI Gym: A Step-by-step Guide to Develop AI Agents Using Deep Reinforcement Learning. Birmingham: Packt Publishing Ltd., 2018.
- [19] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv:1707.06347, 2017.
- [20] Lin HH, Ding WH, Liu ZX, Niu YR, Zhu JC, Niu YM, Zhao D. Safety-aware causal representation for trustworthy offline reinforcement learning in autonomous driving. IEEE Robotics and Automation Letters, 2024, 9(5): 4639–4646. [doi: [10.1109/LRA.2024.3379805](https://doi.org/10.1109/LRA.2024.3379805)]
- [21] Chen W, Zhang J, Zhu H, Xu B, Hao Z, Zhang K, Ye J, Cai R. Causal-aware large language models: Enhancing decision-making through learning, adapting and acting. In: Proc. of the 34th Int'l Joint Conf. on Artificial Intelligence. Montreal: ijcai.org, 2025. 4292–4300. [doi: [10.24963/IJCAI.2025/478](https://doi.org/10.24963/IJCAI.2025/478)]
- [22] Wang JW, Du DH, Tian LL, Chen YK, Li YD, Li YY. ERCI: An explainable experience replay approach with causal inference for deep reinforcement learning. In: Proc. of the 39th AAAI Conf. on Artificial Intelligence. Philadelphia: AAAI Press, 2025. 27671–27679. [doi: [10.1609/aaai.v39i26.34981](https://doi.org/10.1609/aaai.v39i26.34981)]
- [23] Cao HY, Feng F, Yang TP, Huo J, Gao Y. Causal information prioritization for efficient reinforcement learning. In: Proc. of the 13th Int'l Conf. on Learning Representations. Singapore: OpenReview.net, 2025.
- [24] Zhang A, McAllister RT, Calandra R, Gal Y, Levine S. Learning invariant representations for reinforcement learning without reconstruction. In: Proc. of the 9th Int'l Conf. on Learning Representations. OpenReview.net, 2021.
- [25] Bennett A, Kallus N, Li LH, Mousavi A. Off-policy evaluation in infinite-horizon reinforcement learning with latent confounders. In: Proc. of the 24th Int'l Conf. on Artificial Intelligence and Statistics. PMLR, 2021. 1999–2007.
- [26] Huang BW, Feng F, Lu CC, Magliacane S, Zhang K. AdaRL: What, where, and how to adapt in transfer reinforcement learning. In: Proc. of the 10th Int'l Conf. on Learning Representations. OpenReview.net, 2022.
- [27] Feng F, Huang BW, Zhang K, Magliacane S. Factored adaptation for non-stationary reinforcement learning. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2022. 2316.
- [28] Gu SX, Lillicrap TP, Ghahramani Z, Turner RE, Levine S. Q-Prop: Sample-efficient policy gradient with an off-policy Critic. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.
- [29] Espeholt L, Soyer H, Munos R, Simonyan K, Mnih V, Ward T, Doron Y, Firoiu V, Harley T, Dunning I, Legg S, Kavukcuoglu K. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 1406–1415.
- [30] Gruslys A, Dabney W, Azar MG, Piot B, Bellemare MG, Munos R. The Reactor: A fast and sample-efficient Actor-Critic agent for reinforcement learning. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.
- [31] Xu P, Gao F, Gu QQ. An improved convergence analysis of stochastic variance-reduced policy gradient. In: Proc. of the 35th Conf. on Uncertainty in Artificial Intelligence. Tel Aviv: AUAI Press, 2019. 541–551.
- [32] Schulman J, Moritz P, Levine S, Jordan MI, Abbeel P. High-dimensional continuous control using generalized advantage estimation. arXiv:1506.02438, 2015.
- [33] Wu C, Rajeswaran A, Duan Y, Kumar V, Bayen AM, Kakade SM, Mordatch I, Abbeel P. Variance reduction for policy gradient with action-dependent factorized baselines. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.
- [34] Mao HZ, Venkatakrishnan SB, Schwarzkopf M, Alizadeh M. Variance reduction for reinforcement learning in input-driven environments. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019.
- [35] Pawlowski N, Castro DC, Glocker B. Deep structural causal models for tractable counterfactual inference. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 73.

附中文参考文献

- [5] 林谦, 余超, 伍夏威, 董银昭, 徐昕, 张强, 郭宪. 面向机器人系统的虚实迁移强化学习综述. 软件学报, 2024, 35(2): 711–738. <http://www.jos.org.cn/1000-9825/7006.htm> [doi: [10.13328/j.cnki.jos.007006](https://doi.org/10.13328/j.cnki.jos.007006)]

- [6] 刘建伟, 高峰, 罗雄麟. 基于值函数和策略梯度的深度强化学习综述. 计算机学报, 2019, 42(6): 1406–1438. [doi: [10.11897/SP.J.1016.2019.01406](https://doi.org/10.11897/SP.J.1016.2019.01406)]
- [9] 张峻伟, 吕帅, 张正昊, 于佳玉, 龚晓宇. 基于样本效率优化的深度强化学习方法综述. 软件学报, 2022, 33(11): 4217–4238. <http://www.jos.org.cn/1000-9825/6391.htm> [doi: [10.13328/j.cnki.jos.006391](https://doi.org/10.13328/j.cnki.jos.006391)]

作者简介

蔡瑞初, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为因果发现, 图神经网络, 领域自适应学习, 自然语言处理.

林富艺, 硕士生, 主要研究领域为强化学习, 因果发现.

陈薇, 博士, 讲师, 主要研究领域为机器学习, 因果推断, 因果强化学习.

朱海鹏, 硕士生, 主要研究领域为强化学习, 因果发现.

郝志峰, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为因果发现, 图神经网络, 领域自适应学习, 自然语言处理.