

基于时序一致性约束的 Agent 实时决策优化算法

彭商谦¹, 刘郝霞², 林嘉君¹, 冯 丽¹

1. 成都信息工程大学 计算机学院, 四川 成都 610225;

2. 四川大学 文学与新闻学院, 四川 成都 610207

收稿日期: 2025-11-12

基金项目: 国家社会科学基金(24XTQ005); 四川省科技厅重点研发项目(2023YFG0144); 四川省科技厅川渝创新合作计划项目(2024YF-HZ0086); 成都信息工程大学创新创业训练计划项目(X202510621154)

第一作者: 彭商谦, 男, 讲师, 博士, 现从事人工智能, 数据管理, 数据安全研究。E-mail: psl@cuit.edu.cn

摘 要: 在动态工业环境中, Agent 决策过程需持续应对数据流变化与分布漂移。现有在线优化方法在连续时序决策场景中往往仅关注即时奖励最大化, 缺乏对相邻时间步策略输出时间相关性的显式建模, 易导致策略输出在相邻时间步之间产生高频波动, 从而引发性能不稳定。针对该问题, 本文提出一种基于时序一致性约束的 Agent 实时决策优化算法(Temporal Consistency Constrained Agent, TCC-Agent)。该方法在策略梯度优化框架中引入时序一致性约束项, 通过约束相邻时间步动作输出差异及策略分布差异, 将时间维度平滑性融入联合损失函数, 实现对策略更新过程的稳定调控。方法主要由策略生成网络与一致性约束模块构成, 约束项直接参与损失计算, 无需引入额外可学习参数。在典型时序决策环境与工业负载仿真环境中的实验结果表明, 与对比算法相比, 所提方法在策略稳定性指标与损失波动控制方面均取得较大改善, 表现为损失曲线方差降低且动作变化率下降, 并在非平稳环境下保持更稳定的收敛趋势。结果表明, 引入时序一致性约束能够有效抑制策略高频振荡, 提升决策过程的鲁棒性与稳定性。

关 键 词: Agent; 时序一致性约束; 实时决策优化; 在线学习; 工业智能

中图分类号: TP391

文献标志码: A

文章编号: 1671-8836(2026)04-0487-10

Real-Time Decision Optimization Algorithm for Agents with Temporal Consistency Constraints

PENG Shanglian¹, LIU Haoxia², LIN Jiajun¹, FENG Li¹

1. School of Computer, Chengdu University of Information Technology, Chengdu 610225, Sichuan, China;

2. The College of Literature and Journalism, Sichuan University, Chengdu 610207, Sichuan, China

Abstract: In dynamic industrial environments, agents are required to continuously adapt to evolving data streams and distribution shifts. Existing online optimization methods for sequential decision-making typically focus on maximizing immediate rewards, while lacking explicit modeling of temporal dependencies between policy outputs at adjacent time steps. This may lead to high-frequency fluctuations in consecutive actions and consequently unstable performance. To address this issue, this paper proposes a real-time decision optimization algorithm termed the Temporal Consistency Constrained Agent (TCC-Agent). Within a policy gradient framework, a temporal consistency constraint is incorporated into the objective function by penalizing discrepancies between consecutive actions as well as differences in policy distributions across adjacent time steps. In this way, temporal smoothness is integrated into the joint loss, enabling more stable regulation of the policy update process. The proposed framework consists of a policy network and a consistency constraint module. The constraint term directly participates in loss computation without introducing additional learnable parameters. Experimental results on continuous control benchmarks and an industrial load simulation environment demonstrate that the proposed method achieves improvements in policy stability and loss fluctuation control compared with

引用格式: 彭商谦, 刘郝霞, 林嘉君, 等. 基于时序一致性约束的 Agent 实时决策优化算法[J]. 武汉大学学报(理学版), 2026, 72(4): 487-496. DOI: 10.14188/j.1671-8836.2025.0211.

PENG Shanglian, LIU Haoxia, LIN Jiajun, et al. Real-Time Decision Optimization Algorithm for Agents with Temporal Consistency Constraints [J].

J Wuhan Univ (Nat Sci Ed), 2026, 72(4): 487-496. DOI: 10.14188/j.1671-8836.2025.0211(Ch).

baseline algorithms. Specifically, it reduces loss variance and action variation rates, while maintaining more stable convergence under non-stationary conditions. These results indicate that incorporating temporal consistency constraints can effectively mitigate high-frequency policy variations and enhance the robustness and stability of sequential decision-making processes.

Key words: Agent; temporal consistency constraint; real-time decision optimization; online learning; industrial intelligence

0 引言

随着工业系统的数字化与智能化升级,实时数据的规模与复杂度迅速增长,系统决策过程面临数据分布漂移、动态约束变化和高频扰动等挑战。传统基于静态样本或周期性更新的优化算法难以适应这种流式数据环境,导致模型在应对非平稳输入和突发变化时性能退化。Agent 技术因具备环境感知、自主学习与策略更新能力,逐渐成为动态环境下实现智能决策的重要手段,在工业机器人打磨焊接、自动驾驶车辆控制、智能制造过程参数调节以及能源调度功率控制等场景中,对决策系统的实时响应能力与运行稳定性提出了极高要求。

然而,在此类复杂工业场景中,Agent 的在线学习往往受到数据不连续、反馈滞后及策略振荡等因素的影响,难以在实时性与稳定性之间取得平衡。决策序列在时间维度上的不一致性已成为突出问题:在工业机器人领域,策略输出的轻微抖动可能通过执行器放大,导致加工精度下降、设备磨损加剧;在自动驾驶中,控制指令的不平滑会直接转化为方向盘的“画龙”或油门的“窜动”,影响乘坐舒适性与安全性;在过程控制与能源调度中,决策参数的剧烈跳变可能打破系统平衡,引发生产波动或电网振荡。因此,如何赋予 Agent 在动态环境中的时序一致性决策能力,使其在快速响应变化的同时输出平滑、稳定、可靠的决策序列,已成为推动工业智能从理论走向实践的关键瓶颈。

近年来,学界在强化学习领域围绕时序一致性与策略优化展开了大量研究,取得了一系列进展。这些研究表明,时序一致性机制能够有效缓解策略振荡、改善模型收敛性,并增强决策稳定性。然而,通过对现有文献^[1-6]的梳理与分析,我们发现当前基于时序一致性的方法在应对高动态工业环境时仍存在明显局限,主要体现在以下两个方面:

首先,在机制融合层面,多数方法将时序一致性作为一种辅助性的正则化手段。例如, Mysore 等^[2]提出的 CAPS(Conditioning for Action Policy Smoothness)方法通过在动作输出层施加平滑性正则抑制振荡。而 Kim^[3]的工作则侧重于约束 Q 函数在相邻时间步的变化。这类方法将一致性作为辅

助损失项,并未将其深度耦合至策略优化的核心梯度更新逻辑中。这种“外围式”的约束在高频动态环境下往往显得响应迟缓,难以在实时响应性与策略平滑性之间实现动态、自适应的平衡,从而导致收敛过程不稳定或决策延迟。

其次,在应用场景与约束适应性层面,现有研究的验证多局限于标准仿真环境(如 OpenAI Gym^[5]、Mujoco^[6]),其任务(如 CartPole、Ant)的状态转移相对平稳,与工业场景中常见的数据分布突变、周期性负载振荡等复杂情况存在差距。因此,这些方法在面向多目标工业决策(如同时优化能耗、延迟与吞吐量)时,其约束强度通常是固定不变的,缺乏根据环境动态进行自适应调整的能力。这导致在分布突变点时,策略要么因约束过强而响应迟钝,要么因约束过弱而产生剧烈振荡,无法满足工业系统对稳定性与能效的严苛要求。

在高动态工业场景中,状态转移分布往往呈现快速漂移或周期性突变特征。此时,传统策略梯度方法的更新仅基于即时奖励梯度,其参数更新方向对瞬时噪声高度敏感,容易产生高频策略振荡。这种振荡不仅增加执行能耗,还可能在执行层放大为机械抖动或控制波动。因此,本文提出一种基于时序一致性约束的 Agent 实时决策优化算法(Temporal Consistency Constrained Agent, TCC-Agent)。在不牺牲实时响应能力的前提下,该算法通过在优化目标层面显式建模时序一致性,抑制高频更新波动,提升策略在非平稳环境中的稳定性与鲁棒性。

1 相关工作

1.1 强化学习中的时序一致性与策略优化

在动态环境下优化 Agent 决策,近年来已有不少研究试图在强化学习/在线优化框架中引入时序一致性或稳定性机制以提升性能与鲁棒性。McInroe 等^[7]提出 K-Step Latent (KSL)辅助任务,强制表征在时间维度保持连贯,以提升强化学习中样本利用率与策略表现。Zhang 等^[8]提出了一种基于时序差分约束的奖励建模方法(TDRM),通过在训练过程中最小化时间差分来获得更加平滑、时序一致的奖励函数,显著提升强化学习过程的稳定性和长期目

标一致性。在离线强化学习方向,Zhang 等^[9]提出 Q-Distribution guided Q-Learning(QDQ),通过一致性模型估计 Q 值分布的不确定性,并对高不确定性区域的 Q 值施加惩罚性调整,缓解 Q 值过估计与不稳定性问题。一些研究在价值网络增强或正则方向也具有启发意义。Value Penalized Q-Learning (VPQ)^[10]在推荐系统背景下通过惩罚不稳定 Q 值(基于不确定性权重)来缓解分布偏移风险,提升策略的稳定性与泛化能力。在高度动态、实时响应要求强的工业场景中,这类方法或难以平衡响应速度与策略平滑性的冲突。基于此,本文算法将时序一致性约束融入 Agent 实时策略更新过程,以在流式数据环境下获得稳定、平滑且高效的决策序列。

1.2 基于时序约束的稳定性增强方法

为增强强化学习/在线优化算法在动态环境下的稳定性,研究者从多个层面尝试在策略或价值网络中引入时序一致性或正则化机制。Thodoroff 等^[11]提出,在价值估计过程中对连续时间步的估计值施加平滑正则,可以减少估计方差并稳定训练过程,该方法在离散与连续马尔可夫决策过程(MDP)上均表现出有效性。Shen 等^[12]引入一种平滑性正则化框架(SR²L),在政策训练中加入对策略输出随状态变化的平滑性约束,从而提升策略对测量误差和扰动的鲁棒性。Song 等^[13]设计了一种名为 LipsNet 的网络结构,通过网络级别(而非层级)约束 Lipschitz 常数以实现动作输出的平滑性和鲁棒性,该方法可无缝集成到 RL(Reinforcement Learning)算法中。这些工作表明:将时间或平滑性约束引入策略/价值学习,是提升策略稳定性与收敛性的有效方向。

1.3 Agent 在工业系统、多目标优化中的应用

在工业系统与工程优化背景下,Agent 与强化学习逐步被用于解决高维、非线性和多目标控制任务,其研究聚焦在如何让 Agent 在现实约束下平衡性能、成本、能耗等多重目标。

在制造过程参数调优领域,Paranjape 等^[14]提出一种基于强化学习的框架,使用 MO-MPO(Multi-Objective Machine Process Optimization)算法训练 Agent,在齿轮制造过程中同时优化法兰半径和厚度两个质量指标。该研究在仿真环境中验证了 Agent 在多目标参数选配方面的有效性。在过程控制领域,Li 等^[15]针对工业控制系统提出了一种自适应阈值驱动 MORL(Multi-Objective Reinforcement Learning)方法,引导 Agent 在多个目标间动态切换,并在仿真实验中展示了较高的控制性能。在能效驱动的工业控制问题中,Schäfer 等^[16]设计复合奖励函

数同时惩罚跟踪误差与电功率消耗,并在 Quanser Aero 2 系统中验证了不同权重下的性能折中。该研究体现了在工业级平台上同时优化性能与能耗的可行性。在工业布局与资源规划领域,Klar 等^[17]采用深度 RL 方法,使 Agent 可在工厂布局设计中同时考虑多个目标(如空间利用率、物流成本与连通性),并提出迁移学习方法以提高跨场景通用性。在多 Agent 多目标协作优化中,Chen 等^[18]提出一个多目标多 Agent 的 Proximal Policy Optimization(PPO)框架,支持多个 Agent 在序列任务中协作优化多个目标(如调度效率与能耗),具有较好的扩展性。

这些工作共同展现了 Agent 在工业/多目标场景中的应用潜力:它们在参数调优、过程控制、能耗管理、布局规划及协作调度等方向均取得了成果。然而,绝大多数方法侧重于策略设计、Sequential Multi-Objective Multi-Agent PPO (SMOMA-PPO)目标函数构造和任务场景部署,而较少关注在 Agent 的实时策略更新过程中植入时序一致性机制以提升决策平滑性、稳定性和适应性。

2 本文方法

2.1 模块协同机制与结构设计

TCC-Agent 算法由两个核心部分组成:策略生成网络与一致性约束模块。

策略生成网络以环境状态 s_t 为输入,输出动作 $a_t = \pi_\theta(s_t)$,其中 θ 为主网络参数。该网络通过环境奖励信号驱动参数更新,负责感知环境状态并生成动作策略,是决策过程的核心。

一致性约束模块为无参数结构,不参与策略表达,仅对相邻时间步动作序列进行一致性评估,并生成平滑约束信号反馈至优化过程。该模块通过限制策略变化幅度,抑制由非平稳分布或噪声扰动引起的高频振荡。

在整体架构中,策略生成网络负责即时收益最大化,一致性约束模块负责时间连续性调控。二者通过联合损失函数形成闭环优化机制,在实时响应能力与决策稳定性之间建立平衡。算法框架如图 1 所示。

2.2 时序一致性约束建模与联合损失函数

在上述结构协同机制基础上,本节对时序一致性约束进行数学建模,并定义联合损失函数。

2.2.1 强化学习基本优化目标

策略函数由主网络参数化为 $\pi_\theta(a|s)$ 。Agent 与环境 Env 的交互过程可描述为 MDP。环境 Env 定义为一个马尔可夫决策过程: $\text{Env} = (S, A, P, R)$,其中 S 为

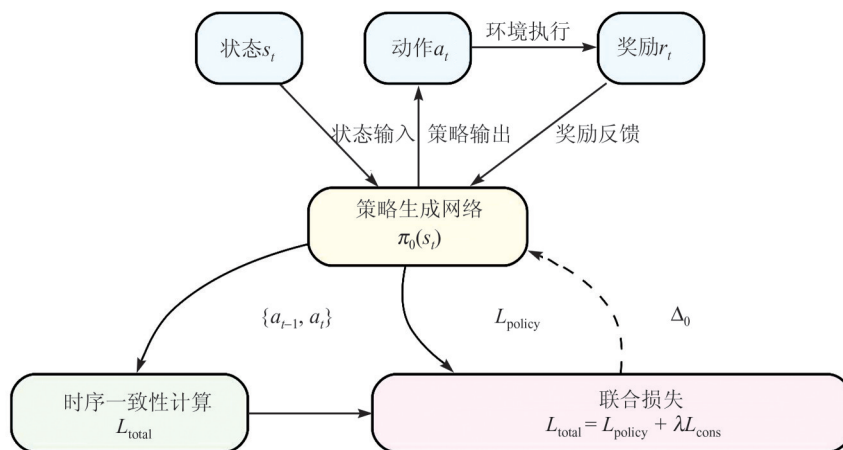


图1 基于时序一致性约束的 Agent 实时决策优化总体框架

Fig. 1 Overall framework of the proposed Agent real-time decision optimization based on temporal consistency constraint

状态空间, A 为动作空间, $P(s_{t+1}|s_t, a_t)$ 为状态转移函数, $R(s_t, a_t)$ 为奖励函数。

强化学习的目标是最大化长期累积期望回报:

$$J(\theta) = \mathbb{E}_{\pi_{\theta}} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] \quad (1)$$

其中, $\mathbb{E}_{\pi_{\theta}}$ 表示在当前策略 π_{θ} 下对轨迹分布的数学期望; $\gamma \in (0, 1)$ 为折扣因子; $\sum_{t=0}^{\infty} \gamma^t r_t$ 表示时间维度上的累积折扣奖励 (Return), 用于刻画策略在无限时间尺度上的长期收益期望。在实际算法实现中, 由于训练过程基于有限步长进行, (1) 式通过有限时间步的样本回报 R_t 进行估计, $R_t = \sum_{k=t}^T \gamma^{k-t} r_k$, 其中 T 为当前步长终止时间步。

策略梯度框架下的优化损失函数表示为:

$$L_{\text{policy}} = -\mathbb{E}[R_t \log \pi_{\theta}(a_t | s_t)] \quad (2)$$

其中, R_t 为时间步 t 的累计折扣回报。

2.2.2 时序一致性约束建模

在动态环境中, 相邻时间步状态通常具有较强的相关性, 因此策略输出应保持时间连续性。

定义动作级一致性约束:

$$L_{\text{cons}}^{(a)} = \|a_t - a_{t-1}\|^2 \quad (3)$$

若考虑策略分布层面的变化, 可在动作分布空间中引入 KL 散度^[19]约束。由于策略函数为条件概率分布 $\pi_{\theta}(a|s)$, 因此一致性约束定义为:

$$L_{\text{KL}} = D_{\text{KL}}(\pi_{\theta}(\cdot | s_{t-1}) \| \pi_{\theta}(\cdot | s_t)) \quad (4)$$

其中, “ $\cdot$ ” 表示对动作空间上的概率分布进行比较。该项用于衡量相邻时间步策略分布在动作空间中的整体形态差异。

综合得到一致性损失函数:

$$L_{\text{cons}} = \beta_1 \|a_t - a_{t-1}\|^2 + \beta_2 L_{\text{KL}} \quad (5)$$

其中 β_1, β_2 为权重参数。实验中, β_1, β_2 设置为 1:1, 以保证两尺度约束的平衡。

2.2.3 联合优化目标

在强化学习基本目标之上, 结合时序一致性约束, 构建联合损失函数:

$$L_{\text{total}} = L_{\text{policy}} + \lambda L_{\text{cons}} \quad (6)$$

其中, λ 为时序一致性权重, λL_{cons} 为一致性约束项。权重 λ 通过网格搜索在 $\{0.01, 0.05, 0.10, 0.50\}$ 范围内选取。当 λ 较小时, 算法更侧重即时奖励最大化; 当 λ 较大时, 策略更新趋于平滑。该联合损失函数通过 λ 在即时奖励梯度与一致性约束梯度之间形成显式权衡结构, 为算法在响应性与平滑性之间提供连续可调的优化路径, 不同 λ 取值对应不同的稳定性-响应性折中关系。

2.3 策略优化算法与收敛性分析

2.3.1 参数更新规则

策略参数采用梯度下降方式更新:

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} L_{\text{total}} \quad (7)$$

其中, α 为学习率。

该更新规则表明, 参数更新方向同时受即时奖励梯度与一致性约束梯度的共同影响。

2.3.2 算法流程

基于上述联合损失函数, TCC-Agent 如算法 1 所示。

2.3.3 联合损失函数的局部收敛性分析

从优化理论角度看, TCC-Agent 的参数更新过程等价于对联合损失函数 $L_{\text{total}}(\theta) = L_{\text{policy}}(\theta) + \lambda L_{\text{cons}}(\theta)$ 执行梯度下降。

定理 1 (联合损失函数的局部收敛性) 设联合损失函数

$$L_{\text{total}}(\theta) = L_{\text{policy}}(\theta) + \lambda L_{\text{cons}}(\theta)$$

算法 1 TCC-Agent

输入: 环境模型 Env; 学习率 α ; 折扣因子 γ ; 时序一致性权重 λ ; 最大训练轮数 T ; 每轮最大步数 N

输出: 最优策略参数 θ^*

```

1  初始化主网络参数  $\theta$ , 经验池  $D \leftarrow \emptyset$ 
2  for episode = 1 to  $T$  do
3      初始化环境状态  $s_0$ 
4      for each step  $t = 1$  to  $N$  do
5          根据策略  $\pi_\theta(a_t|s_t)$  采样动作  $a_t$ 
6          根据状态转移函数  $P(s_{t+1}|s_t, a_t)$  生成新状态, 并获得奖励  $R(s_t, a_t)$ 
7          将  $(s_t, a_t, r_t, s_{t+1})$  存入经验池  $D$ 
8          计算累计回报:  $R_t \leftarrow \sum_{k=t}^N \gamma^{k-t} r^k$ 
9          采用(2)式计算策略梯度损失
10         采用(6)式计算联合损失函数  $L_{\text{total}}$ 
11         更新主网络参数  $\theta \leftarrow \theta - \alpha \cdot \nabla \theta L_{\text{total}}$ 
12     end for
13 end for
14 输出最优参数  $\theta^*$ 

```

在参数邻域内连续可微, 且其梯度满足 Lipschitz 连续条件, 即存在常数 $L > 0$, 使得

$$\|\nabla L_{\text{total}}(\theta_1) - \nabla L_{\text{total}}(\theta_2)\| \leq L \|\theta_1 - \theta_2\|$$

若学习率满足 $0 < \alpha < \frac{2}{L}$, 则梯度下降更新

$$\theta_{k+1} = \theta_k - \alpha \nabla L_{\text{total}}(\theta_k)$$

生成的序列满足:

- 1) 联合损失函数单调下降;
- 2) 梯度平方可求和:

$$\sum_{k=0}^{\infty} \|\nabla L_{\text{total}}(\theta_k)\|^2 < \infty;$$

3) $\lim_{k \rightarrow \infty} \|\nabla L_{\text{total}}(\theta_k)\| = 0$, 即参数序列收敛至一个一阶临界点 θ^* , 满足

$$\nabla L_{\text{total}}(\theta^*) = 0$$

证 1) 梯度 Lipschitz 性

由假设, 存在常数 $L_1, L_2 > 0$, 使得

$$\|\nabla L_{\text{policy}}(\theta_1) - \nabla L_{\text{policy}}(\theta_2)\| \leq L_1 \|\theta_1 - \theta_2\|$$

$$\|\nabla L_{\text{cons}}(\theta_1) - \nabla L_{\text{cons}}(\theta_2)\| \leq L_2 \|\theta_1 - \theta_2\|$$

因此联合梯度满足

$$\|\nabla L_{\text{total}}(\theta_1) - \nabla L_{\text{total}}(\theta_2)\| \leq (L_1 + \lambda L_2) \|\theta_1 - \theta_2\|$$

记 $L = L_1 + \lambda L_2$

2) 下降引理

由梯度 Lipschitz 连续性^[20], 可得经典下降引理:

$$L_{\text{total}}(\theta_{k+1}) \leq L_{\text{total}}(\theta_k) + \langle \nabla L_{\text{total}}(\theta_k), \theta_{k+1} - \theta_k \rangle + \frac{L}{2} \|\theta_{k+1} - \theta_k\|^2$$

代入更新式

$$\theta_{k+1} - \theta_k = -\alpha \nabla L_{\text{total}}(\theta_k)$$

得到

$$L_{\text{total}}(\theta_{k+1}) \leq L_{\text{total}}(\theta_k) - \alpha \|\nabla L_{\text{total}}(\theta_k)\|^2 + \frac{L\alpha^2}{2} \|\nabla L_{\text{total}}(\theta_k)\|^2$$

整理得

$$L_{\text{total}}(\theta_{k+1}) \leq L_{\text{total}}(\theta_k) - \alpha \left(1 - \frac{L\alpha}{2}\right) \|\nabla L_{\text{total}}(\theta_k)\|^2$$

当 $0 < \alpha < 2/L$ 时, $1 - \frac{L\alpha}{2} > 0$, 因此

$$L_{\text{total}}(\theta_{k+1}) \leq L_{\text{total}}(\theta_k)$$

即联合损失函数单调下降。

3) 梯度平方可求和

将上式从 $k=0$ 累加至 N , 得到:

$$\sum_{k=0}^N \alpha \left(1 - \frac{L\alpha}{2}\right) \|\nabla L_{\text{total}}(\theta_k)\|^2 \leq L_{\text{total}}(\theta_0) - L_{\text{total}}(\theta_{N+1})$$

由于损失函数在邻域内存在下界 L_{inf} , 右侧有界, 因此

$$\|\nabla L_{\text{total}}(\theta_k)\|^2 < \infty$$

4) 梯度范数趋于零

由于非负项级数收敛, 必有

$$\lim_{k \rightarrow \infty} \|\nabla L_{\text{total}}(\theta_k)\| = 0$$

因此序列收敛至一个一阶临界点(证毕)。

3 实验与结果分析

3.1 实验设置

实验在统一硬件平台上运行: CPU 为 Intel i9-13900K, GPU 为 NVIDIA RTX 4090 (24 GB), 内存 64 GB, 操作系统为 Ubuntu 22.04, 深度学习框架采用 PyTorch 2.2。

实验采用三个具有代表性的时序决策环境:

1) CartPole-v1 (OpenAI Gym)^[21]: 评估算法在低维、连续状态空间下的收敛速度与稳定性; 2) LunarLanderContinuous-v2 (<https://huggingface.co/sb3ppo-LunarLanderContinuous-v2>): 验证算法在复杂动力学约束任务中的控制精度与时序平滑性; 3) IndustrialLoadSim: 为模拟工业设备能耗调度与控制场景, 本文构建了 IndustrialLoadSim 连续控制环境。状态空间维度为 8, 包含当前负载需求、历史变

化率、设备功率输出及预测误差项等特征;动作空间为连续区间 $[-1,1]$,表示功率调节比例,实际更新形式为 $P_{t+1}=P_t+\alpha a_t$,其中 $\alpha=0.05$ 为最大调节步长。奖励函数定义为 $r_t=-|L_t-P_t|-\beta P_t^2$, $\beta=0.1$,为能耗惩罚权重。为构造非平稳性,每2000步对负载分布均值进行扰动,并随机注入幅值为基准负载10%的脉冲干扰;同时在状态输入中叠加高斯噪声 $\mathcal{N}(0,\sigma^2)$,其中 $\sigma\in\{0.01,0.02,0.04\}$ 。所有实验总训练步数为 1.0×10^6 ,每100步进行一次策略评估,并报告均值±标准差结果。

选取近年来在连续控制与动态优化任务中具有代表性的4个算法进行对比。1) DrQ-v2(Data Regularized Q-learning v2)^[22],该算法通过数据增强与正则化策略改进传统Q-learning,在连续动作控制任务中提升了样本效率与鲁棒性,DrQ-v2代表了近年来基于正则化思想的强化学习最新进展。2) TQC(Truncated Quantile Critics)^[23]通过对分位值分布进行截断以减轻Q值估计中的过估计偏差,从而提升训练过程中的数值稳定性与策略收敛稳定性。其分布式策略估计思想与本文引入的一致性约束在稳定性目标上具有互补性。3) ED2(Ensemble Diversified DDPG)^[24]使用多个异质Actor-Critic网络形成集成结构,通过策略多样性与经验共享机制提升在动态环境下的稳健性与泛化能力,是近年来高引用的稳定强化学习方法之一。4) Meta-SAC^[25]结合元学习与自适应熵正则化机制,实现策略在不同任务间的快速自适应与时序平滑控制。其在线再训练特性与本文实时优化目标一致,是近年来最具代表性的自适应策略优化方法。DrQ-v2与Meta-SAC通过正则化与自适应熵调节机制实现

策略的隐式平滑;L2D引入“延迟决策”以抑制时序波动;ED2与TQC则分别通过集成多样性与分位数截断增强了决策稳定性。PPO算法采用与原论文一致的标准实现框架,本文将它作为基线算法。为保证公平性,其主要参数设置如下:学习率设为 3.0×10^{-4} ,折扣因子 $\gamma=0.99$,广义优势估计参数为0.95,剪切系数 $\epsilon=0.2$,批大小为256,每次更新执行10轮次,熵正则系数设为0.01。策略网络结构与本文方法保持一致,以消除模型容量差异带来的性能影响。所有参数在三个实验环境中保持一致,以确保比较的公平性与可复现性。

TCC-Agent算法学习率,折扣因子,批大小与基线算法一致;一致性权重为0.05,训练轮数为1000,经验池容量 1.0×10^5 。

注1: CAPS^[2]、L2C2^[4]与SR²L^[12]主要针对离线策略平滑或辅助一致性学习场景,部分方法依赖额外模型结构或预训练过程,其原始实现未提供可直接迁移至本文所采用的连续状态决策环境的公开代码框架。因此,本文未将这些方法纳入对比。

3.2 实验指标

本文采用了涵盖收敛性、稳定性、时序平滑性、实时性与能效的多维指标全面评估算法性能,这些指标定义如表1所示。

3.3 实验结果与分析

3.3.1 整体性能对比

表2给出了在三个环境上的性能结果。对于每个环境,首先在5个独立随机种子下取均值;随后对三个环境的结果进行等权算术平均,得到表3中的综合性能指标。其中,响应延迟(Response Delay, RD)定义为单步决策过程中策略网络的前向推理时

表 1 实验评价指标定义与说明

Table 1 Evaluation metrics and definitions

指标名称	计算公式	含义说明
平均回报	$\overline{R}=\frac{1}{N}\sum_{i=1}^N\sum_{t=1}^{T_i}r_t^{(i)}$	衡量智能体在完整任务周期内获得的平均累计奖励,用于反映策略整体性能与收敛水平
稳定性指数	$SI=\frac{\text{Var}(L_t)}{\text{Mean}(L_t)}$	以训练过程中损失函数的方差与均值的比值衡量算法的收敛平稳性,值越小表示收敛越稳定
损失方差	$\sigma_L^2=\frac{1}{T}\sum_{t=1}^T(L_t-\overline{L})^2$	衡量训练阶段中损失函数值在时间维度上的波动幅度,用以反映优化过程的稳定性与收敛平滑程度
动作变化率	$AJ=\sqrt{\frac{1}{T-1}\sum_{t=2}^T\ a_t-a_{t-1}\ _2^2}$	度量连续时间步动作变化的平滑性,值越小表示决策序列越连续、时序一致性越好
能耗优化率	$EOR=\frac{E_{\text{base}}-E_{\text{model}}}{E_{\text{base}}}\times100\%$	反映模型相较基线在能耗上的降低比例,用于工业任务能效评估
噪声扰动性能下降率	$PDN=\frac{R_{\text{clean}}-R_{\text{noise}}}{R_{\text{clean}}}\times100\%$	衡量在输入含噪情况下性能下降的幅度,用于评估算法的鲁棒性

间,即从输入状态 s_t 到输出动作 a_t 所消耗的时间。实验中使用 PyTorch 的时间计量接口在 GPU 上统计每一步前向传播时间,单位为 ms,并在每个环境下取 10 000 步平均值作为最终结果。

如表 2 所示,与基线算法 PPO 相比,TCC-Agent 在平均回报指标上,相对提升 9.7%,稳定性

指数下降约 22.4%,响应延迟下降约 18.0%,动作变化率下降约 25.3%,单位时间平均能耗下降 $13.2\% \pm 0.4\%$,在各项指标上优于其他 4 种算法与基线算法对比数值。双尾独立样本 t 检验结果表明,上述关键指标改进均在显著性水平为 0.05 时达到统计显著($p < 0.05$)。

表 2 各算法在三种环境上的平均综合性能对比

Table 2 Average comprehensive performance of various algorithms across three environments

算法	平均回报 ↑	稳定性指数 ↓	响应延迟/ms ↓	动作变化率 ↓	能耗优化率 ↑
PPO	1.00 ± 0.03	1.00 ± 0.02	100 ± 2.1	1.00 ± 0.04	0%
DrQ-v2	1.063 ± 0.02	0.882 ± 0.03	89 ± 1.8	0.915 ± 0.03	$5.9\% \pm 0.7\%$
TQC	1.071 ± 0.02	0.855 ± 0.03	92 ± 2.0	0.886 ± 0.04	$8.2\% \pm 0.6\%$
ED2	1.084 ± 0.02	0.821 ± 0.02	85 ± 1.6	0.847 ± 0.03	$9.6\% \pm 0.5\%$
Meta-SAC	1.090 ± 0.02	0.804 ± 0.02	83 ± 1.5	0.813 ± 0.03	$10.4\% \pm 0.5\%$
TCC-Agent	1.097 ± 0.01	0.776 ± 0.02	82 ± 1.3	0.747 ± 0.02	$13.2\% \pm 0.4\%$

3.3.2 收敛与稳定性分析

从图 2 可以观察到,TCC-Agent 在训练早期阶段的回报提升速度快于对比算法,曲线斜率更陡,在前 1 000~1 500 轮区间内性能已取得较大幅度提升。随着训练推进,各算法逐步趋于稳定,但 TCC-Agent 始终保持更高的性能水平。与此同时,训练曲线波动幅度较小,优化轨迹更为平滑。这是由于时序一致性约束项的引入,实现了策略输出的时间连续性,减少了梯度更新过程中的高频振荡。

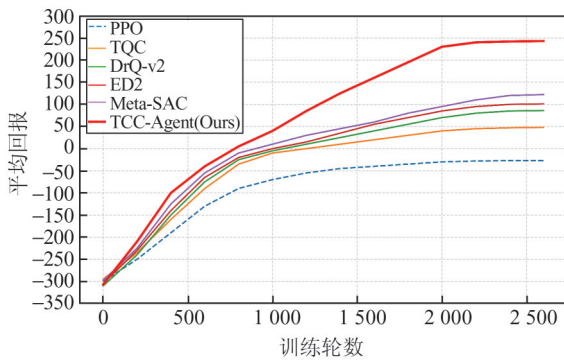


图 2 不同算法在 LunarLanderContinuous-v2 环境下的平均回报收敛曲线

Fig. 2 Convergence curve of average return for different algorithms in the LunarLanderContinuous-v2 environment

3.3.3 噪声与扰动鲁棒性分析

为评估算法在输入扰动下的鲁棒性,在状态输入中注入高斯噪声 $\mathcal{N}(0, \sigma^2)$,并逐步增大噪声方差 $\sigma^2 \in [0, 0.040]$ 。从噪声扰动性能下降率曲线结果(图 3)可见,TCC-Agent 明显低于其他算法(普遍超过 10%),表明其在输入扰动下保持了更优的决策稳定性与鲁棒性。这是因为引入的时序一致性约

束项在联合损失函数中对相邻时间步的策略变化施加了惩罚,在输入扰动存在时抑制策略输出的剧烈波动,提升了模型对噪声的鲁棒性。

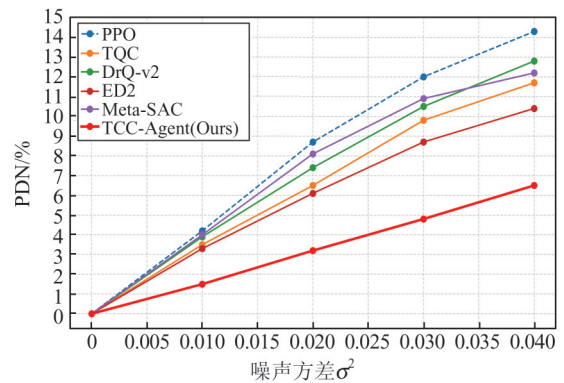


图 3 不同噪声水平下各算法的 PDN 对比

Fig. 3 Comparison of performance degradation under noise among different algorithms

3.3.4 消融实验

本文在 IndustrialLoadSim 工业负载仿真环境下设计了消融实验。表 3 中,w/o Consistency Module 表示去除一致性约束模块,仅保留策略生成网络的优化过程,不引入时序约束项,用于验证一致性约束在抑制决策波动方面的作用;w/o Distribution-Level Term 表示去除分布级约束,仅保留动作级约束项 $\|a_t - a_{t-1}\|^2$,去除基于 Kullback-Leibler 散度的分布平滑项,用于评估多尺度约束结构对稳定性提升的贡献;Full Model 为完整模型,即同时包含动作级与分布级一致性约束项的 TCC-Agent 算法。

在相同训练轮数与环境配置下,Full Model 的平均回报较 w/o Consistency Module 提升约

14.2%, 较 w/o Distribution-Level Term 提升约 8%, 训练损失方差下降约 28%。这表明一致性约束模块在提升模型稳定性与收敛质量方面具有较好作用。在能耗指标方面, 表 3 中的能耗优化率以 w/o Consistency Module 作为能耗基准进行归一化计算。Full Model 的能耗优化率较基准模型提升

4.7 个百分点; 相较于 w/o Distribution-Level Term, 提升 2.9 个百分点。该结果表明, 多尺度一致性约束不仅有助于抑制策略振荡, 也在一定程度上改善了工业负载调度过程中的能源利用效率。

表 3 中收敛回合数表示训练过程中联合损失函数, 该数据首次达到稳定区间时对应的轮数。

表 3 消融实验结果
Table 3 Ablation study results

模型	平均回报 $\uparrow$	损失方差 $\downarrow$	收敛回合数 $\downarrow$	性能下降率/% $\downarrow$	能耗优化率/% $\uparrow$
w/o Consistency Module	0.84	0.021	9 200	15.9	87.3
w/o Distribution-Level Term	0.89	0.017	8 700	14.3	89.1
Full Model (TCC-Agent)	0.96	0.015	8 000	6.50	92.0

3.3.5 参数敏感性与能耗权衡分析

为进一步验证联合损失函数中一致性权重参数 λ 对算法性能的影响, 本文设置 $\lambda \in \{0.01, 0.05, 0.10, 0.50\}$ 进行参数敏感性实验。

实验结果如图 4 所示。随着 λ 从 0.01 增大到 0.50, 模型的平均回报先升后降, 而动作变化率呈单调下降趋势。当 $\lambda = 0.10$ 时, 平均回报与动作平滑性达到最优平衡点, 这一发现为不同应用场景的参数配置提供了指导: 在对平滑性要求极高的精密控制场景(如手术机器人)建议采用较大的 λ 值(0.10~0.30)以确保操作稳定性; 在需要快速响应的场景(如自动驾驶避障)则应选择较小的 λ 值(0.01~0.05)以保障决策时效性; 而在注重能效的工业过程控制场景中, 适中的 λ 值(0.05~0.100)最能兼顾响应速度与能耗优化。这种可调节的特性使得 TCC-Agent 能够通过单一参数灵活适应多样化的工业需求, 增强了算法在实际部署中的实用价值。

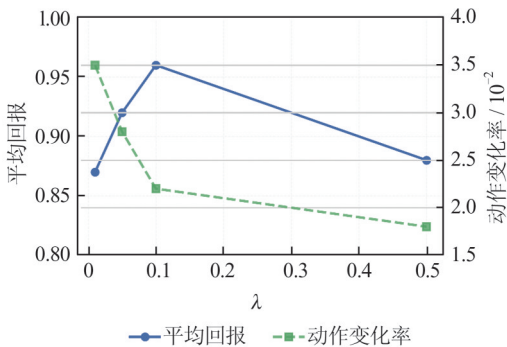


图 4 λ 对平均回报与动作变化率的影响
Fig. 4 Effect of λ on average reward and action jitter

为评估算法在工业负载控制任务中的节能效果, 在 IndustrialLoadSim 环境下绘制了能耗优化率与平均回报的关系曲线, 如图 5 所示。结果显示,

TCC-Agent 在保持较高回报的同时, 能耗优化率稳定提升 10~13 个百分点, 优于 Meta-SAC(基于自适应熵调节的策略优化方法)与 ED2(基于集成结构的鲁棒控制方法)等对比算法。TCC-Agent 通过引入时序一致性约束, 在实现高动态响应的同时具备更优的能耗-性能折中能力, 体现出在工业控制场景下的节能潜力。

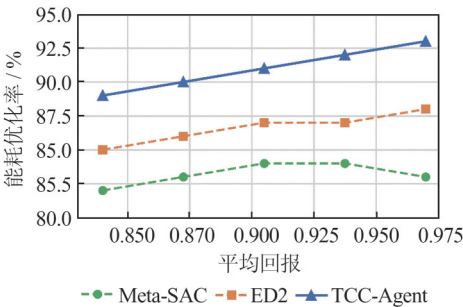


图 5 能耗优化率与平均回报关系
Fig. 5 Trade-off curve between energy optimization rate (EOR) and average reward

3.4 机制分析与讨论

时序一致性约束对性能提升的作用机制, 可以从优化过程与统计特性两个方面进行解释。在联合损失中引入一致性约束项后, 策略更新不仅受即时奖励梯度驱动, 同时受到相邻时间步动作差异的约束, 使参数变化在时间维度上保持连续性。当输入存在噪声或分布漂移时, 传统方法往往对瞬时异常样本产生过度响应, 导致梯度方向频繁波动, 而一致性约束在更新过程中对这种波动形成抑制, 从实验结果上表现为损失方差与动作变化率同步下降, 训练曲线抖动幅度明显减小。另一方面, 一致性约束项增强了策略输出的时间相关性, 使当前决策在一定程度上继承历史信息, 从而削弱高频扰动对单步决策的放大效应, 这也与噪声扰动实验中性

能下降率较低的现象相一致。不过,该机制仍依赖于权重参数的合理设置,当一致性强度过大时,策略可能在突发环境变化下响应滞后;同时,本文验证主要基于仿真环境,真实工业系统中的执行延迟与硬件约束仍有待进一步评估。因此,该方法更适用于存在持续扰动或渐进漂移的动态场景,而在剧烈突变条件下需要结合自适应调节策略加以改进。

在连续控制与动态优化任务中,多种强化学习算法通过不同机制提升策略稳定性与样本效率。例如,DrQ-v2^[22]通过数据增强与正则化机制改进 Q-learning 框架,提高了样本利用效率与泛化能力;TQC^[23]采用截断分位值估计以缓解值函数高估问题,从分布式角度增强了训练稳定性;ED2^[24]通过构建多样化 Actor-Critic 集成结构,提高了在动态环境下的鲁棒性;基于元学习的 Soft Actor-Critic 变体 Meta-SAC^[25]引入自适应更新机制,使策略在分布变化时具备更快的适应能力。这些方法在提升策略稳定性方面取得了一定进展,但其稳定性增强机制多依赖于正则化、数据增强或结构集成等手段,并未在损失函数层面对相邻时间步策略输出的连续性进行显式建模。从实验结果看,本文通过在优化目标中引入时序一致性约束项,从时间维度直接刻画策略输出的平滑性,在响应性与稳定性之间实现可调节的权衡,取得了较好的效果。

4 结 语

围绕动态工业环境中实时决策稳定性与响应性之间的矛盾,本文提出了一种引入时序一致性约束的策略优化方法 TCC-Agent。该方法在策略梯度框架下将时间维度的平滑约束直接嵌入联合损失函数,使策略更新过程在追求即时收益的同时受到相邻时间步输出差异的调节,在非平稳数据流环境中实现较为平衡的响应速度与收敛稳定性。与仅依赖结构正则或值函数稳定增强的策略不同,本文方法在优化目标层面对策略变化幅度进行约束,使时间连续性成为参数更新过程的内在组成部分。

实验结果显示,在连续控制任务与工业负载仿真环境中,TCC-Agent 在平均回报、收敛速度、动作变化率及能耗优化率等指标显示算法性能稳定。损失方差与动作抖动幅度的同步下降表明,一致性约束在时间维度上有效抑制了高频更新波动;在噪声扰动条件下性能下降幅度较小,说明该机制对梯度方差具有一定抑制作用。消融实验验证了动作级与分布级一致性约束对整体稳定性的贡献。

需要指出的是,本文方法的性能仍依赖于一致性权重参数 λ 的合理设置。在约束强度过大时,策略可能在突发分布变化场景下出现短期响应滞后,说明响应性与平滑性之间仍存在权衡关系。此外,本文实验主要基于仿真环境构建,真实工业系统中的执行延迟、硬件约束及安全边界问题仍需进一步验证。因此,该方法更适用于存在持续扰动或渐进漂移的高动态场景,在剧烈突变环境下可结合自适应权重调节或异常检测机制加以改进。

未来工作将围绕一致性权重的动态调节策略、多智能体协同情境下的时序约束扩展以及真实工业系统部署验证等方向展开,以进一步提升算法在复杂工程场景中的适用性与稳定性。

参考文献:

- [1] ZHAO Y, ZHAO W S, BONEY R, *et al.* Simplified temporal consistency reinforcement learning [C]//Proceedings of the 40th International Conference on Machine Learning. Cambridge: ML Research Press, 2023: 42227-42246. DOI:10.5555/3618408.3620184.
- [2] MYSORE S, MABSOUT B, MANCUSO R, *et al.* Regularizing action policies for smooth control with reinforcement learning[C]//2021 IEEE International Conference on Robotics and Automation (ICRA). New York: IEEE Press, 2021: 1810-1816. DOI:10.1109/ICRA48506.2021.9561138.
- [3] KIM C. Temporal consistency-based loss function for both deep Q-networks and deep deterministic policy gradients for continuous actions[J]. *Symmetry*, 2021, **13**(12): 2411. DOI: 10.3390/sym13122411.
- [4] KOBAYASHI T. L2C2: Locally Lipschitz continuous constraint towards stable and smooth reinforcement learning [C]//2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). New York: IEEE Press, 2022: 4032-4039. DOI:10.1109/IROS47612.2022.9981812.
- [5] BROCKMAN G, CHEUNG V, PETERSSON L, *et al.* OpenAI Gym[DB/OL]. [2025-04-02]. <https://arxiv.org/pdf/1606.01540>.
- [6] TODOROV E, EREZ T, TASSA Y. MuJoCo: A physics engine for model-based control[C]//2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. New York: IEEE Press, 2012: 5026-5033. DOI: 10.1109/IROS.2012.6386109.
- [7] MCINROE T, SCHÄFER L, ALBRECHT S V. Learning temporally-consistent representations for data-efficient reinforcement learning[DB/OL]. [2025-04-02]. <https://arxiv.org/pdf/2110.04935>.
- [8] ZHANG D, CAI M, LI J, *et al.* TDRM: Smooth re-

- ward models with temporal difference for LLM RL and inference[DB/OL]. [2025-04-02]. <https://arxiv.org/pdf/2509.15110>.
- [9] ZHANG J, FANG LIN J J, SHI K X, *et al.* Q-Distribution guided Q-learning for offline reinforcement learning: Uncertainty penalized Q-value *via* consistency model[DB/OL]. [2025-04-02]. <https://www.proceedings.com/content/079/079017-1724open.pdf>.
- [10] GAO C Q, XU K, ZHOU K Q, *et al.* Value penalized Q-learning for recommender systems[DB/OL]. [2025-04-02]. <https://dl.acm.org/doi/pdf/10.1145/3477495.3531796>.
- [11] THODOROFF P, DURAND A, PINEAU J, *et al.* Temporal regularization for Markov decision process[DB/OL]. [2025-04-02]. https://proceedings.neurips.cc/paper_files/paper/2018/file/4b0250793549726d5c1ea3906726ebfe-Paper.pdf.
- [12] SHEN Q L, LI Y, JIANG H M, *et al.* Deep reinforcement learning with robust and smooth policy[DB/OL]. [2025-04-02]. <http://proceedings.mlr.press/v119/shen20b/shen20b.pdf>.
- [13] SONG X, DUAN J, WANG W, *et al.* LipsNet: A smooth and robust neural network with adaptive Lipschitz constant for high accuracy optimal control[DB/OL]. [2025-04-02]. <https://proceedings.mlr.press/v202/song23b/song23b.pdf>.
- [14] PARANJAPE A, QUADER N, UHLMANN L, *et al.* Reinforcement learning agent for multi-objective online process parameter optimization of manufacturing processes[J]. *Applied Sciences*, 2025, **15**(13): 7279. DOI: 10.3390/app15137279.
- [15] LI D Z, GU W T, SONG T H. Multi-objective reinforcement learning in process control: A goal-oriented approach with adaptive thresholds[J]. *Journal of Process Control*, 2023, **129**: 103063. DOI:10.1016/j.jprocont.2023.103063.
- [16] SCHÄFER G, SELIGER R, REHRL J, *et al.* Multi-objective reinforcement learning for energy-efficient industrial control[C]//Database and Expert Systems Applications – DEXA 2025 Workshops. Cham: Springer, 2026: 67–72. DOI:10.1007/978-3-032-02003-1_6.
- [17] KLAR M, SCHWORM P, WU X Q, *et al.* Transferable multi-objective factory layout planning using simulation-based deep reinforcement learning[J]. *Journal of Manufacturing Systems*, 2024, **74**: 487–511. DOI:10.1016/j.jmsy.2024.04.007.
- [18] CHEN Y, LIU C. Sequential multi-objective multi-agent reinforcement learning approach for predictive maintenance[J]. *Advanced Engineering Informatics*, 2025, **67**: 103553. DOI: <https://doi.org/10.1016/j.aei.2025.103553>.
- [19] CUI J, TIAN Z T, ZHONG Z S, *et al.* Decoupled Kullback-Leibler divergence loss[DB/OL]. [2025-04-02]. https://proceedings.neurips.cc/paper_files/paper/2024/file/87ee1bbac4635e7c948f3eea83c1f262-Paper-Conference.pdf.
- [20] HAGER W W. Lipschitz continuity for constrained processes[J]. *SIAM Journal on Control and Optimization*, 1979, **17**(3): 321–338. DOI:10.1137/0317026.
- [21] HSIAO J Y, DU Y X, CHIANG W Y, *et al.* Unentangled quantum reinforcement learning agents in the OpenAI Gym[DB/OL]. [2025-04-02]. <https://arxiv.org/pdf/2203.14348>.
- [22] YARATS D, FERGUS R, LAZARIC A, *et al.* Mastering visual continuous control: Improved data-augmented reinforcement learning[DB/OL]. [2025-04-02]. <https://arxiv.org/pdf/2107.09645>.
- [23] KUZNETSOV A, SHVECHIKOV P, GRISHIN A, *et al.* Controlling overestimation bias with truncated mixture of continuous distributional quantile critics[DB/OL]. [2025-04-02]. <http://proceedings.mlr.press/v119/kuznetsov20a/kuznetsov20a.pdf>.
- [24] JANUSZEWSKI P, OLKO M, KRÓLIKOWSKI M, *et al.* Continuous control with ensemble deep deterministic policy gradients[DB/OL]. [2025-04-02]. <https://arxiv.org/pdf/2111.15382>.
- [25] HONARI H, SOUFI ENAYATI A M, GHAFARIAN TAMIZI M, *et al.* Meta SAC-lag: Towards deployable safe reinforcement learning *via* MetaGradient-based hyperparameter tuning[C]//2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). New York: IEEE Press, 2024: 619–626. DOI: 10.1109/IROS58592.2024.10802547.

□