

基于扩散模型与对抗模仿学习的智能模型预测控制策略

任凯楠¹, 付东飞^{1†}, 金志豪¹, 黎明^{1,2}

(1. 中国海洋大学 工程学院, 山东 青岛 266404;

2. 中国海洋大学 海洋工程与技术创新研究院, 山东 青岛 266100)

摘要: 模型预测控制 (MPC) 的性能高度依赖精确的系统模型与精心设计的代价函数, 这限制了其在复杂非线性系统中的应用. 为克服该局限性, 提出一种融合扩散模型与生成式对抗模仿学习 (DGAIL) 的智能 MPC 策略. 该方法利用扩散模型的分布建模能力, 从专家示范中更准确地学习隐式奖励函数, 从而有效减少对研究者经验与人工奖励设计的过度依赖, 并兼具无模型学习与基于模型优化的优势. 进一步引入一种基于粒子群优化 (PSO) 的在线更新策略, 以高效求解非线性 MPC 的约束优化问题. 在多个基准仿真环境中的实验结果表明, 所提方法在控制性能与安全性方面均优于现有模仿学习与强化学习算法, 验证了其有效性与泛化能力.

关键词: 模仿学习; 模型预测控制; 约束控制; 逆向强化学习; 扩散模型; 粒子群优化

中图分类号: TP273 文献标志码: A

DOI: 10.13195/j.kzyjc.2025.1036

引用格式: 任凯楠, 付东飞, 金志豪, 等. 基于扩散模型与对抗模仿学习的智能模型预测控制策略 [J]. 控制与决策.

An intelligent predictive control strategy with diffusion model and adversarial imitation learning

REN Kai-nan¹, FU Dong-fei^{1†}, JIN Zhi-hao¹, LI Ming^{1,2}

(1. College of Engineering, Ocean University of China, Qingdao 266404, China; 2. Institution of Marine Engineering and Technological Innovation, Ocean University of China, Qingdao 266100, China)

Abstract: The performance of model predictive control (MPC) relies heavily on precise system models and well-designed cost functions, which limits its applicability to complex nonlinear systems. To address this limitation, this paper proposes an intelligent MPC scheme that integrates diffusion models (DM) with generative adversarial imitation learning (DGAIL). By using the strong distribution modeling capability of DM, the proposed method learns an implicit reward function more accurately from expert demonstrations, thereby significantly reducing reliance on manual reward design and expert prior knowledge. It also combines the benefits of model-free learning and model-based optimization. Furthermore, a particle swarm optimization (PSO)-based online update strategy is introduced to efficiently solve the constrained nonlinear MPC problem. Experimental results across multiple benchmark simulation demonstrate that the proposed approach outperforms existing imitation learning and reinforcement learning methods in terms of control performance and safety, confirming its effectiveness and generalization ability.

Keywords: imitation learning; model predictive control; constraint control; inverse reinforcement learning; diffusion model; particle swarm optimization

0 引言

模型预测控制 (MPC) 能够有效处理多变量约束控制问题, 在流程工业、自动驾驶及无人系统等领域获得了广泛应用^[1-3]. 然而, 传统 MPC 的性能严重依赖精确的系统模型与人为设计的代价函数. 对于复杂的非线性系统, 其动力学模型难以精确建立, 且人

工设计的代价函数往往无法充分刻画实际控制目标与专家行为偏好^[4], 这极大地限制了其在未知或复杂环境下的应用.

为降低对精确模型的依赖, 数据驱动方法被引入 MPC 框架. 一类方式是将描述系统的复杂数学方程抽象为神经网络映射, 策略网络直接选择最优动

收稿日期: 2025-10-06; 录用日期: 2026-03-16.

基金项目: 山东省自然科学基金项目 (ZR2022MF280).

责任编委: 侯忠生.

†通信作者. E-mail: fudongfei@ouc.edu.cn.

作^[5],这种端到端的控制方法避免了对精确物理模型的依赖.除了直接的策略学习之外,神经网络也被广泛用于学习系统动力学模型,近似观测模型、状态估计器甚至约束函数.这就使得被控对象模型在显式形式不可用或过于复杂的情况下,仍然可以实现高精度预测,从而构建数据驱动的MPC控制器^[6].另一类方式是将强化学习(RL)与MPC的结合,通过将RL学得的价值函数嵌入MPC的优化目标,为短期优化引入长期视野^[7].然而,无论是MPC的代价函数还是RL的价值函数,其设计通常都依赖于研究者的人工先验与大量调试.这种对显式奖励塑造的依赖,使得控制器难以学习那些易于演示但难以量化(即不可观测奖励)的复杂专家行为.

模仿学习(IL)为奖励设计难题提供了可行的解决路径,借鉴生成对抗网络(GAN)思想的生成式对抗模仿学习(GAIL)通过引入判别器来区分专家与智能体的行为,并以此隐式地恢复奖励函数,克服了行为克隆(BC)中固有的分布漂移问题^[8].但GAIL方法的判别器常采用二分类器结构,在面对高维、复杂的专家级状态-动作对分布时,建模能力有限,易导致训练不稳定与模仿性能不佳.近年来,研究者们借鉴图像生成任务中的研究经验^[9],将扩散模型(DM)强大的分布匹配能力引入对抗模仿学习(AIL)的判别器中,在复杂数据分布建模中展现出强大能力.例如DiffAIL^[10]将扩散模型融入对抗学习框架,利用其卓越的分布匹配特性来增强判别器,显著提升了模仿学习的稳定性与表达能力.

基于DM的模仿学习在策略学习上取得了一定的进展,但如何将其与具有约束保证、滚动优化特性的MPC框架有效结合,仍是一个有待深入探索的问题.单纯的无模型模仿学习缺乏对系统动态和约束的显式考虑,而传统的MPC又难于设计合理有效的奖励函数.因此,一个自然的思路是将基于DM的模仿学习作为MPC的上层决策,为其提供源自专家数据的、隐式的性能导向.同时保留MPC作为底层控制策略,利用其模型(即便是近似的)与约束处理能力来保证系统的安全性与动态性能.基于这些考虑,本文提出一种融合扩散模型与生成式对抗模仿学习(DGAIL)的智能模型预测控制架构(DiffMPC).其核心在于通过逆强化学习(IRL),利用DM判别器从专家数据中隐式学习更精确、更鲁棒的奖励函数,以替代MPC中的人工设计代价项,从而将专家智能嵌入优化回路.同时,为高效求解由此产生的复杂非线性、非凸优化问题,引入了基于粒子群优化(PSO)的在线滚动优化策略PSO-MPC,以适应高维动作空间.

本文主要贡献可总结如下:1)针对非线性系统,提出了DGAIL与MPC相结合的新型控制决策架构DiffMPC,实现了专家知识的隐式奖励学习与模型约束控制的统一;2)设计了PSO-MPC方法,在保证控制精度的同时,显著提升了高维非线性系统约束优化问题的求解效率;3)在基准环境上的多个非线性动态系统实验结果表明,所提方法在控制性能与安全性上均优于现有的模仿学习与强化学习算法.

1 问题描述

1.1 非线性系统预测控制

本文研究在显式奖励信号稀疏或难以设计的场景下,为非线性约束动力学系统学习最优控制策略的问题.控制目标是在满足系统安全约束的前提下,使智能体的行为表现逼近专家水平.考虑如下离散时间非线性系统:

$$\begin{cases} x(k+1) = f(x(k), u(k)); \\ y(k) = g(x(k), u(k)). \end{cases} \quad (1)$$

同时,该系统需满足如下时域约束:

$$\begin{cases} u_{\min} \leq u(k) \leq u_{\max}, \forall k \geq 0; \\ \Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max}, \forall k \geq 0; \\ y_{\min}(k) \leq y(k) \leq y_{\max}(k), \forall k \geq 0. \end{cases} \quad (2)$$

其中: $x(k) \in \mathbb{R}^n$ 和 $y(k) \in \mathbb{R}^b$ 分别为系统在 k 时刻的状态向量和输出, $u(k) \in \mathbb{R}^m$ 为控制输入, $\Delta u(k)$ 表示控制输入变化率.

控制目标可描述为在预测时域内 $k \in [1, H]$,求解最优控制序列 $\mathbf{u}_k := \{\bar{u}(k), \dots, \bar{u}(k+H-1)\}$,使得控制系统在满足约束的同时,控制性能尽可能接近专家示范.传统MPC通过求解如下带约束优化问题实现该目标:

$$\min_{\bar{u}(\cdot)} J(\bar{x}(\cdot), \bar{u}(\cdot)), \text{ s.t. (1), (2)}. \quad (3)$$

其中:代价函数 J 通常为人工设计, $\bar{x}(\cdot)$ 和 $\bar{u}(\cdot)$ 分别为预测的系统状态和控制输入.然而,当专家行为所对应的真实奖励函数 R_{true} 未知或难以显式表达时,如何构建有效的代价函数是核心挑战.

为解决此问题,假设能够获得专家的示范数据 $\tau_E = \{(s_k^E, a_k^E)\}_{k=1}^T$,但无法直接获取由环境定义的奖励函数.通过IL,用DM判别器来估计状态-动作对 (s, a) 来源于专家行为的概率,从 τ_E 中学习一个隐式的奖励模型 $R_\theta(s, a)$,并根据任意RL算法利用 R_θ 来训练智能体策略 $\pi_\phi(a|s)$,其优化问题为:

$$\max_{\pi_\phi} \mathbb{E}_{\pi_\phi} \left[\sum_{t=0}^{\infty} \gamma^t R_\theta(s_t, a_t) + \alpha \mathcal{H}(\pi_\phi(\cdot | s_k)) \right]. \quad (4)$$

其中: \mathcal{H} 表示用于促进探索的策略熵正则化项, γ 是

折扣因子, R_θ 为学习得到的奖励函数, 并以此构建 MPC 的代价函数, 从而引导控制器复现专家行为。

1.2 生成式对抗模仿学习 GAIL

模仿学习旨在从专家示范中直接学习一个最优策略 $\pi_\phi^*(a|s)$, 其主流方法包括行为克隆 (BC) 与逆向强化学习 (IRL). 考虑到 BC 面临分布漂移导致的累积误差问题, 更可取的方式是采用 IRL 从 τ_E 中推断出潜在的奖励函数. 其核心思想在于专家策略之所以最优, 是因为智能体会更频繁地访问那些对获得高回报至关重要的 (s, a) 以最大化累积奖励^[11]. 为描述 IRL, 引入策略 π 的占用度量 $\rho(s, a)$ 以表征在该策略下 (s, a) 的访问频率:

$$\rho(s, a) = \pi(a|s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \pi). \quad (5)$$

其中: $P(s_t = s | \pi)$ 表示在策略 π 下, 智能体/专家在时刻 t 访问状态 s 的概率, $\gamma \in (0, 1)$. IRL 的目标是寻找一个奖励函数, 能够最小化智能体策略的占用度量 $\rho_\phi(s, a)$ 和专家策略的占用度量 $\rho_E(s, a)$ 之间的 f -散度 D_f :

$$\min_{\pi} D_f(\rho_\phi(s, a) || \rho_E(s, a)). \quad (6)$$

GAIL^[10] 借鉴 GAN 框架求解问题 (6), 它由生成器与判别器组成. 其通过二分类判别器的输出概率 $D_\theta(s, a)$ 区分 ρ_E 与 ρ_ϕ , 并形成对抗博弈:

$$\min_{\pi_\phi} \max_D \mathbb{E}_{\rho_\phi} [\log D_\theta(s, a)] + \mathbb{E}_{\rho_E} [\log(1 - D_\theta(s, a))]. \quad (7)$$

博弈策略 (7) 致使生成器产生的轨迹在判别器看来与专家行为无法区分. 同时, 生成器通过最大化由 $\log D_\theta(s, a)$ 构成的替代奖励来优化策略:

$$R(s, a) = -\mathbb{E}_{\rho_\phi} [\log D_\theta(s, a)]. \quad (8)$$

因此, 智能体策略 π_ϕ 使其行为分布 ρ_ϕ 与专家分布 ρ_E 尽可能接近, 亦即 π_ϕ 与专家策略 π_E 对齐.

1.3 扩散模型 DM

DM 是一类深度生成模型^[9], 通过逐步去噪机制以稳健地捕捉底层数据分布, 从噪声中迭代式地合成高质量且稳定的样本数据. 本节引入去噪扩散概率模型 (DDPM), 它是一个包含前向扩散与反向去噪的两阶段随机过程. 一方面, 前向过程涉及在每一个时间步 $t \in \{1, \dots, T\}$, 对一个原始样本 x_0 逐步添加高斯噪声, 得到 x_1, x_2, \dots, x_T . 在此马尔可夫链中, 每一步的扰动由预定义的噪声调度 β_t 控制:

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t \mathbf{I}). \quad (9)$$

另一方面, 作为 DM 核心的反向过程, 旨在逆转

前向扩散的加噪过程, 通过逐步去噪重构出原始数据样本. 它是一个神经网络 θ , 学习近似真实的反向条件分布:

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \quad (10)$$

其中 μ_θ 与 Σ_θ 分别表示由参数 θ 建模的均值与协方差, 它们依赖于当前样本 x_t 和时间步 t . 模型通过优化如下损失函数来训练, 以匹配真实去噪方向:

$$\mathcal{L}_{DM} = \mathbb{E}[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|^2]. \quad (11)$$

其中: $\bar{\alpha}_t = \prod_{j=1}^t (1 - \beta_j)$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, ϵ_θ 表示由扩散模型估计的噪声.

需要指出, 本文所处理的系统控制任务与图像生成不同, 其 DM 的 (s, a) 空间具有相对较低的维度和特定结构. 因此, 只需要较少的扩散步数即可获得令人满意的性能^[12], 这可大幅提高计算效率. 应用 DM 主要有两种范式: 一是作为策略表示的直接建模, 通过扩散过程生成动作序列; 二是作为判别器增强传统 GAIL 框架, 如在 DiffAIL^[10] 中提升了专家与智能体轨迹的区分能力, 在 DRAIL^[12] 中构建了更平滑且信息量更丰富的奖励函数. 受这些工作的启发, 本文构建一个强大的扩散模型判别器, 使其能够更精确地度量专家与智能体行为分布之间的差异, 从而为 MPC 控制器提供更准确的奖励信号.

2 基于 DM 与 GAIL 的智能模型预测控制

本节提出融合扩散模型与生成对抗模仿学习的智能预测控制框架 DiffMPC, 实现从专家示范中学习隐式奖励函数, 以此引导约束优化过程. 如图 1 所示, DiffMPC 主要由两个核心组件构成: 扩散判别器, 负责提供精确的奖励信号; 基于 PSO 的预测控制器, 求解带约束的非线性 MPC 优化问题.

2.1 DGAIL

采用基于 DM 的判别器可增强传统 GAIL 框架. 与使用简单二分类器的传统判别器不同, 扩散判别器通过多步去噪过程来评估 (s, a) 的质量, 从而实现了对专家行为分布更精细的建模.

在每一控制时刻, DiffMPC 同时采样专家演示数据和当前智能体交互数据, 分别构成专家状态-动作对 (s^E, a^E) 与智能体状态-动作对 (s^ϕ, a^ϕ) . 首先对状态与动作进行拼接操作, 形成联合变量, 并将专家与智能体样本统一映射至同一特征空间中. 随后, 对拼接后的状态-动作对引入前向扩散过程, 通过逐步向样本中注入高斯噪声, 将原始数据映射为一系列扩散时间步上的随机变量, 从而获得多噪声层级下的状态-动作分布表示.

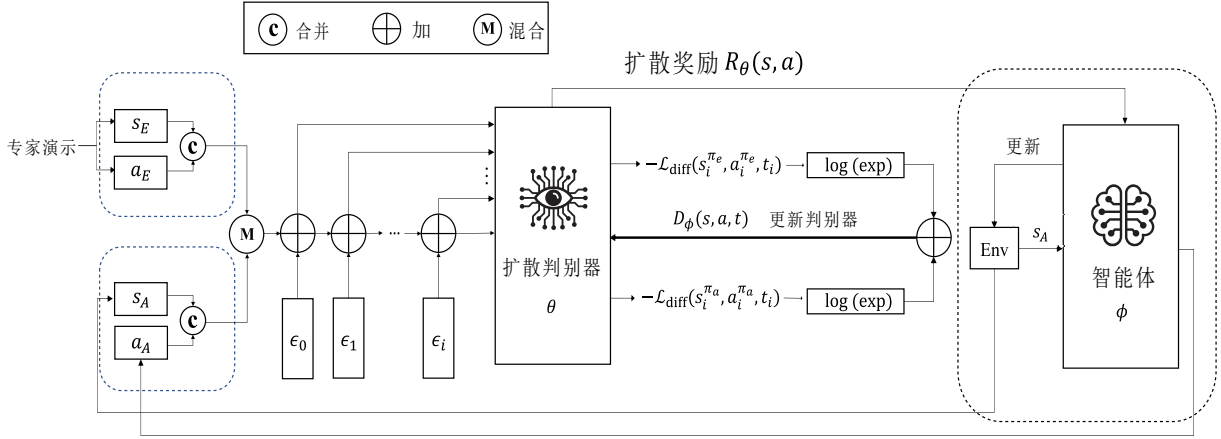


图1 DiffMPC 控制框架

在扩散判别器设计中,采用固定的马尔可夫过程 (9) 设计前向扩散,即每一步按照标准高斯分布 $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ 逐步加入噪声.在反向去噪过程 (10) 中,去噪后的变量 (x_{t-1}, a_{t-1}) 由当前带噪样本 (x_t, a_t) 结合预测噪声 $\epsilon_\theta(x_t, a_t, t)$ 推断得到.高斯分布在反向过程中的条件均值 $\mu_\theta(x_t, a_t, t)$ 为:

$$\mu_\theta(x_t, a_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, a_t, t) \right). \quad (12)$$

其中: $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ 为累计噪声调度参数, $\epsilon_\theta(x_t, a_t, t)$ 表示由判别器预估的噪声.不同于传统对抗判别器仅输出二分类概率,设计的网络并不生成分类概率,而是直接在每一个扩散时间步反向过程中预估噪声得到 $\hat{\epsilon}_\theta(s, a, t)$,进而获得更平滑、更稳定的训练奖励.

在判别器训练完成后,其输出被用于构造隐式奖励函数.该奖励函数以状态-动作对为输入,反映当前策略行为在扩散分布意义下与专家策略的一致程度.在奖励生成设计中,模型的训练目标是 minimized 预测噪声与真实噪声 ϵ_t 之间的差异,即优化如下扩散损失函数:

$$\mathcal{L}_{\text{Diff}}(s, a, t) = \mathbb{E}[\|\hat{\epsilon}_\theta(s, a, t) - \epsilon_t\|^2]. \quad (13)$$

为实现对抗训练,基于预估的噪声误差定义伪概率 D_θ 作为奖励信号,用于指导生成器的更新:

$$D_\theta(s, a, t) = \exp(-\mathcal{L}_{\text{Diff}}(s, a, t)). \quad (14)$$

其优化目标为类似二元交叉熵的对抗损失函数 \mathcal{L}_D :

$$\mathcal{L}_D = \mathbb{E}_{(s^\phi, a^\phi) \in \tau_\phi, t \sim T} [\log(1 - D_\theta(s^\phi, a^\phi, t))] + \mathbb{E}_{(s^E, a^E) \in \tau_E, t \sim T} [\log D_\theta(s^E, a^E, t)]. \quad (15)$$

在策略优化设计中,采用 Soft Actor-Critic (SAC)^[13] 算法作为生成器,在最大熵框架下优化策略 (4).策略被优化为一个由噪声驱动的生成过程,且生成器

不依赖环境提供的奖励,而是根据判别器的伪概率 D_θ 替代奖励函数 $R_\theta(s, a)$:

$$R_\theta(s, a) = -\frac{1}{T} \sum_{t=1}^T \log(1 - D_\theta(s, a, t)). \quad (16)$$

由于判别器输出并非严格概率,直接使用其作为奖励易导致梯度饱和及训练不稳定.通过对判别器伪概率 (14) 进行对数变换,奖励函数 R_θ 在数学上近似于专家分布与策略分布之间的对数密度比,从而与对抗模仿学习的理论目标保持一致.通过最大化伪概率来有效引导生成器更好地欺骗判别器,在实质上最小化预测误差.与标准 GAIL 相比, DGAIL 能够更稳定地训练,并提供更丰富的奖励信号,这对于后续的 MPC 优化至关重要.

2.2 PSO-MPC

提出的 PSO-MPC 控制框架以数据驱动或学习得到的预测模型为基础,在每个控制时刻利用粒子群优化算法对有限时域内的控制序列进行滚动优化,并仅执行首个控制量,从而实现实时闭环控制.其整体流程可划分为"状态感知—预测建模—粒子群优化—滚动执行"四个核心阶段.在时刻 t ,智能体 ϕ 从环境中获取当前系统状态 $s_k \in \mathbb{R}^n$,并将其作为预测与优化的初始条件.该状态可以是系统的物理状态 $x(k)$,也可以是经深度网络编码后的高维特征表示.然后将给定初始控制输入序列 \mathbf{u}_t 作为粒子群初始位置,通过群体智能搜索最优解.

在初始化阶段,粒子在可行动作空间内均匀分布,其速度约束为 $[v_{min}, v_{max}]$.每个粒子 i 在时刻 k 的适应度 P_k^i 通过一个改进的 MPC 代价函数来评估,该代价函数综合考虑了轨迹跟踪精度、控制能耗以及约束违反的惩罚项:

$$P_k^i = \beta_0 \cdot \mathcal{A}(\text{dist}(a_k^i, a_\phi) | s_k) + \beta_1 \cdot \sigma(a_k^i | s_k) + \beta_2 \cdot \tau(a_k^i). \quad (17)$$

其中: σ 为与控制相关的代价项, τ 则量化约束违反程度, 这两类代价项通常可采用二次型形式进行建模, 以保证代价函数的连续性与优化过程的数值稳定性; $\varphi(\cdot)$ 表示当前动作 a_k^i 与智能体 ϕ 在状态 s_k 下提供的参考动作 a_ϕ 之间的相似度; $\beta_0, \beta_1, \beta_2$ 作为超参数用于平衡各代价的影响。

粒子 i 在预测时域 $k \in [0, H]$ 内的总适应度函数为:

$$P^i = \sum_{k=1}^H P_k^i. \quad (18)$$

在每一次迭代中, 所有粒子的适应度都会基于其在最小化适应度函数过程中的表现重新评估. 适应度最优的动作序列被指定为该粒子的个体最优值 P_*^i . 在整个粒子群中, 具有最低总体代价的粒子被标记为全局最优 G_* , 并被作为引导参考驱动所有粒子向解空间中更优的区域演进. 据此, 粒子的速度 v^i 与位置 u^i 按照以下方式进行更新:

$$v_{k+1}^i = wv_k^i + c_1r_1(P_*^i - u_k^i) + c_2r_2(G_* - u_k^i); \quad (19)$$

$$u_{k+1}^i = u_k^i + v_{k+1}^i. \quad (20)$$

其中: 惯性权重 w 用于控制粒子保持当前运动的程度; 学习因子 c_1, c_2 分别用来调节个体经验与群体经验的影响; $r_1, r_2 \sim U(0, 1)$ 用于增加搜索的随机性. 如算法 1 所示, 每个粒子的适应度、速度和位置迭代更新, 此过程重复执行直至达到最大迭代次数或满足收敛条件. 迭代结束后, 选取全局最优控制序列 G_{best} 的第一个控制量, 施加到真实系统中. 系统在执行控制后更新至下一状态, 控制时域向前滚动一个时间步, 并重新进入下一轮 PSO-MPC 优化过程. 完整过程见**算法 1**.

算法 1 PSO-MPC.

step 1: 初始化系统环境, 设定最大迭代次数 N 和粒子数 S , 以及参数 $w, c_1, c_2, \beta_0, \beta_1, \beta_2$. 随机初始化所有粒子的状态 (位置) \mathbf{x}_0 与速度 \mathbf{v}_0 : 计算其初始适应度 P_0^i 并记录个体最优 P_0^i , 选取得到的最优值作为全局最优 G_{best} ; 迭代计数置 $n \leftarrow 0$.

step 2: 在每次迭代中, 所有粒子并行执行:

1. 依据 s_k 与策略模型, 计算粒子 i 的适应度 P_k^i ;
2. 若 $P_k^i > P_*^i$, 则更新个体最优 $P_*^i \leftarrow P_k^i$;
3. 在所有 P_*^i 中选取 $G_* \leftarrow \arg \max_i P_*^i$, 依据式 (19)-(20) 更新粒子速度—位置. 必要时对 $\mathbf{x}^i, \mathbf{v}^i$ 做边界裁剪或约束投影.

更新迭代计数 $n \leftarrow n + 1$, 若 $n \geq N$ 或全局最优增益小于阈值则转入 **step 3**, 否则返回本步继续搜

索.

step 3: 将 G_* 解码为当前时刻的候选控制序列, 采纳其第一个控制量作为执行输入 u_k , 推进系统到下一时刻得到 s_{k+1} ; 用更新后的状态作为新的初始条件, 返回 **step 2** 继续迭代与滚动优化.

step 4: 当达到规划终点或满足收敛/停止条件时, 输出最终控制序列与对应轨迹.

2.3 DiffMPC

DGAIL 可通过构建扩散判别器获得更稳定的隐式奖励信号 R_θ , 而 PSO-MPC 能够在非线性约束优化问题中高效求解近似最优解. 基于此, 可将两者集成为如下两阶段的 DiffMPC 框架, 以实现从专家示范到在线控制的闭环优化. 具体来说, DiffMPC 先收集专家轨迹, 使用扩散增强判别器为 SAC 策略网络提供隐式奖励, 策略网络训练完成后策略被冻结, 作为在线控制阶段的参考生成器使用. 在线控制时, 冻结的策略网络在每个时刻给出参考动作, PSO 在预测时域内搜索控制序列, 使其在满足约束的同时最优; 最终只执行全局最优序列的第一个控制量, 然后滚动到下一时刻继续此过程.

离线训练: 首先, 收集专家示范数据 $\tau_E = \{(s_t^E, a_t^E)\}_{t=1}^T$, 训练并利用扩散增强的对抗判别器 θ 估计某一 (s, a) 来源于专家行为的可能性. 该判别器的 (14) 构成了 DGAIL 模块的核心, 可生成替代奖励信号 R_θ . 使用学得的奖励函数, 在 SAC 框架下训练随机策略网络 $\pi_\phi(a|s)$, 并优化 (4). 一旦训练完成, 策略 $\pi_\phi(a|s)$ 被冻结并重新用于后续过程.

在线控制: 策略网络 $\pi_\phi(a|s)$ 被作为参考生成器, 用于 PSO-MPC 控制器中引导控制优化. 在每一个时间步, PSO 用于求解约束轨迹优化问题. 根据规则 (19)-(20) 迭代更新粒子的速度和位置, 并由个体适应度与全局适应度共同引导. 在优化完成后, 仅执行全局最优粒子对应的第一个控制动作, 并在后续时间步重复这一滚动优化过程.

对于系统约束, 离线学习阶段与在线控制阶段进行了协同处理. 在离线训练阶段, 专家演示数据由已满足物理与控制约束的控制器生成, 其动作分布天然位于可行域内, 从而将约束信息以隐式方式编码于 τ_E 中. 在线优化阶段则对系统约束进行显式处理: 一方面, 在粒子群更新后通过将 u 与 Δu 投影, 对 u_k 进行可行化修正; 另一方面, 在适应度函数中对 u 与 y 约束的违反程度施加惩罚项, 引导粒子搜索逐步回归可行域. 因此, 即使参考动作在当前状态下不可行, PSO-MPC 仍能够在约束条件下完成搜索, 并

仅执行全局最优控制序列的首个控制量, 实现滚动优化与闭环控制. 完整过程见**算法 2**.

算法 2 DiffMPC.

step 1: 初始化专家轨迹 τ_E , 策略参数 ϕ , 扩散判别器参数 θ , 学习率 η_ϕ, η_θ , 及**算法 1** 的其他组件.

step 2: 收集智能体轨迹 $\tau_j \sim \pi_\phi$; 批量采样 $\{(s_j^\phi, a_j^\phi)\}_{j=1}^L \sim \tau_j, \{(s_j^E, a_j^E)\}_{j=1}^L \sim \tau_E$; 采样时间步 $\{t_j\}_{j=1}^L \sim \mu(1, T)$, 以及采样噪声 $\{\epsilon_j\}_{j=1}^L \sim \mathcal{N}(0, \mathbf{I})$.

step 3: 计算预测噪声 $\hat{\epsilon}_j = \epsilon_\theta(s_j, a_j, t_j)$; 根据式 (14) 计算判别器 D_θ 输出, 并根据式 (15) 计算损失函数 \mathcal{L}_D ; 更新判别器参数: $\theta \leftarrow \theta + \eta_\theta \nabla \mathcal{L}_D$.

step 4: 根据 (16) 计算扩散奖励 $R_\theta(s, a)$; 使用 η_ϕ 与 R_θ 更新 SAC 策略 π_ϕ , 并更新熵系数 α .

step 5: 重复 step 2–step 4, 直至迭代上限或收敛.

step 6: 冻结策略网络 π_ϕ ; 利用 π_ϕ 获取参考动作 a^ϕ , 指导 PSO-MPC 更新; 返回**算法 1** 的 step 2.

3 仿真实例

为验证所提出控制算法的有效性和实用性, 在 MuJoCo 物理引擎中的 Hopper、HalfCheetah、Ant 和 Walker2D 等基准环境中进行实验, 这些对象的参数设置与文献^[14]中一致, 涵盖了不同复杂度与维度的多种机器人运动控制任务.

专家示范: 在每个环境中使用由 Stable-Baselines3 实现的 DDPG 算法^[15-16]来训练高性能智能体. 每个环境都收集 8 条成功的轨迹, 每条轨迹包含 1000 个 (s^E, a^E) 数据, 作为 DiffMP 框架的原始监督信号输入, 但在评估过程中不可见.

实现细节: 在 DGAIL 时间嵌入层和线性层注入随机噪声, 以增强其泛化能力. 判别器由三个全连接层组成, 隐藏层维度为 [512, 1024, 512], 采用 Mish 激活函数. 扩散过程根据任务复杂度设置为 20 至 50 步, 网络更新采用 Adam 优化器^[17], 学习率为 3×10^{-5} . 策略网络初始熵系数设为 0.2, 并在训练过程中自动调节. SAC 使用 Adam 优化器, 初始学习率为 0.001, 并在每 20 个 epoch 后减半. 控制部分采用 PSO-MPC 方案, 其中: $H = 15$, ω 从 [0.9, 0.4] 线性退火, $c_1 = c_2 = 1.5$; 取 $\beta_0 = 10, \beta_1 = 0, \beta_2 = 0$ 以突出模仿引导的动作选择而非人工设计的代价项. 所有实验均在随机种子 [0, 1, 2, 3, 4] 下进行, 在每个基准环境中运行五次, 并记录平均结果.

3.1 参考基准

本文提出的 DiffMPC 与四种代表性的 RL/IRL 算法进行比较: (1) BC: 一种监督学习方法, 通过训练状态-动作对直接模仿专家动作; (2) GAIL: 基于

生成对抗框架的模仿学习方法, 通过训练判别器区分专家与智能体的行为; (3) SAC: 最大熵强化学习算法, 这里采用仅基于专家示范的离线版本; (4) DiffAIL: 一种最新的对抗模仿学习方法, 用基于扩散的重构损失替代了标准判别器损失, 将扩散模型融入对抗模仿学习框架中. 这些基准方法涵盖了无模型强化学习与对抗模仿学习两类范式, 可提供与 DiffMPC 全面的对比.

3.2 实验结果

每个对象与环境交互 1000 步, 并收集累积奖励, 结果见表 1 与图 2. 每个任务均在五个不同的随机种子下进行训练, 由于 BC 无法与环境交互, 其表现以水平线形式表示. 与传统模仿学习基准算法 (BC、GAIL 和 DiffAIL) 相比, DiffMPC 实现了显著更高的累积奖励. 值得注意的是, DiffMPC 在所有任务中都生成了更稳定、更陡峭的奖励累积曲线, 凸显了其在长时域控制中的有效性. 图 2 表明, DiffMPC 在四个任务中均以较大优势稳定超越 BC 与 GAIL, 尤其是在高维环境 (Ant 和 Walker2D) 中, 单纯的行为克隆由于分布偏移往往会出现崩溃现象. 与同样使用 DM 但缺乏显式轨迹优化的 DiffAIL 相比, DiffMPC 控制性能更优, 验证了将基于策略引导的 PSO 融入控制环路的优势. 此外, DiffMPC 在某些环境中 (Ant) 达到了甚至略微超过专家水平的奖励, 并在未直接访问真实奖励函数的情况下, 与离线 SAC 具有竞争性表现. DiffMPC 在 HalfCheetah 与 Ant 环境中性能弱于 SAC, 源于任务特性与算法假设之间的匹配差异. 一方面, 该类任务具有高维连续动作空间与频繁接触切换等特点, 对动力学预测精度与长时域一致性要求较高, 而 MPC 在有限时域和实时计算限制下难以充分刻画复杂动力学, 预测误差易在规划过程中累积并影响控制质量. 另一方面, DiffMPC 所依赖的扩散判别奖励侧重于对专家行为分布的逼近, 其优化目标与环境真实回报并非完全一致, 在分布外状态下容易产生保守解, 限制性能上限. 相较之下, SAC 直接以环境回报为优化目标, 通过无模型策略与价值函数的联合学习避免了动力学建模误差的

表1 基于学习的控制算法在四种环境仿真奖励

任务	Hopper	HalfCheetah	Ant	Walker2D
Expert	3402	4463	4228	6717
BC	1405±457	2168±624	1920±932	1228±788
GAIL	3021±258	3402±122	3075±154	3213±527
DiffAIL	3280±98	4390±67	3614±94	5868±104
SAC	3319±193	5460±545	4996±125	4441±384
DiffMPC	3462±44	4502±83	3885±73	6092±86

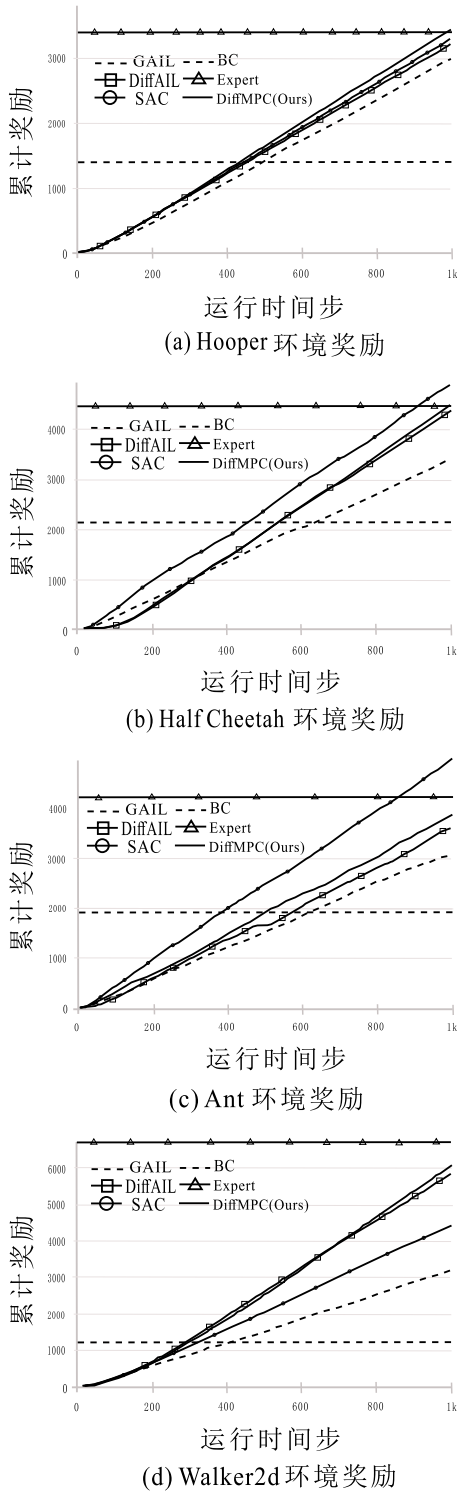


图2 四个环境下六种控制算法的仿真奖励累积

影响。这表明, 基于扩散的奖励替换与 PSO-MPC 规划的结合, 为无模型学习和对抗模仿学习范式提供了一种有效替代方案。

3.3 高效性

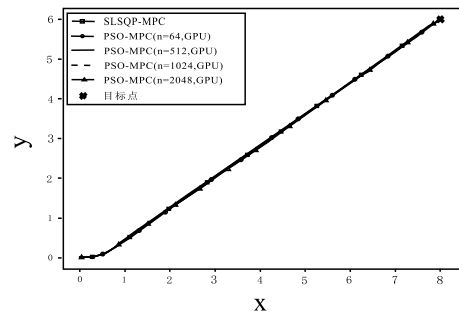
为了评估 DiffMPC 框架对 PSO 配置的敏感性, 通过改变粒子数量与优化迭代次数进行了一系列消融实验。四个环境中 (随机种子固定为 0) 各自评估 (粒子数/迭代次数) 的五种配置: 100/200、200/200、300/100、300/200和400/300。如表 2 所示, 对

于 Hopper 和 HalfCheetah 这类相对简单的环境, 中等配置 (如 200/200 或 300/100) 已经能在方差较小的情况下达到接近专家水平的性能: 例如, 粒子数从 200 提高到 400 或迭代次数从 200 增加到 300 仅带来边际提升。相比之下, 在 Walker2D 和 Ant 等更复杂的环境中, 性能受益于更高的计算开销: 例如, 300/200 情形的性能到 400/300 时的性能提升显著, 表明高维控制任务需要更充分的搜索以找到最优轨迹。

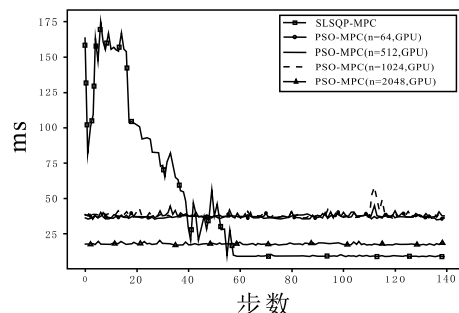
表2 粒子数量与迭代次数对控制性能的影响

粒子数/迭代次数	Hopper	HalfCheetah	Ant	Walker2D
Expert	3402	4463	4228	6717
100/200	3447±70	3136±256	-	-
200/200	3440±58	4395±103	-	-
300/100	3450±50	-	-	-
300/200	3455±67	4447±62	-	5843±113
400/300	3462±44	4502±83	3885±73	6092±86

此外, 本节基于 Unicycle 模型比较了 PSO-MPC 与主流梯度型约束优化器 SLSQP 的控制效果与控制效率。如图 3 所示, 不同粒子规模的 PSO-MPC 均能够生成与基于 SLSQP 的传统 MPC 几乎一致的系运动轨迹, 最终均准确到达目标位置。这表明, 在所考虑的非线性系统中, PSO-MPC 的控制精度和收敛性能并未因采用采样优化而产生明显退化, 在计算效率方面也展示了 PSO-MPC 相对于传统梯度型 MPC 求解器的显著优势: SLSQP-MPC 在控制初期由于非线性程度较高、梯度搜索迭代次数较



(a) 轨迹预测性能



(b) 每步求解时间对比

图3 PSO-MPC 求解效率对比

多,单步优化时间明显偏大,最高可达百毫秒量级,且存在较大波动:随着系统逐渐接近目标,依赖 warm-start 的梯度法求解时间才逐步下降.相比之下,基于 GPU 并行评估的 PSO-MPC 在整个控制过程中保持了稳定且较低的单步求解时间,其计算开销对系统状态变化不敏感,几乎不随控制阶段而波动.

这些结果表明,任务复杂度应指导 PSO 参数的选择:简单环境允许在较少的粒子与迭代次数下高效完成,而更具挑战性的任务则需要更大的计算开销以达到专家级性能.因此,在实际部署 DiffMPC 时,应综合考虑控制质量与计算成本之间的权衡.

3.4 通用性

为了评估 DiffMPC 在专家示范之外的泛化能力,在训练中未使用过的随机种子环境中进行性能测试.如图4所示,在一组全新的环境种子上进行测试,并冻结原始判别器,从而在这些新场景中提供一致的奖励信号以训练新策略.

如表3所示,DiffMPC 在所有任务和随机种子下均实现了较高的累积奖励,即便在 Walker2D 和 Ant 等具有挑战性的环境中依然如此.其性能保持稳定,并接近训练过程中观察到的水平,表明学习到的奖励模型与策略能够很好地泛化到新的状态分布和目标配置.这些结果突出了 DiffMPC 在分布外场景下的鲁棒性,这既得益于扩散模型所学习到的结构化先验,也得益于 PSO 优化所带来的稳定性.

3.5 安全性

本文方法在框架设计上将 MPC 的约束处理能力与模仿学习的高效策略搜索能力相结合,从而在保证控制性能的同时显式引入安全与可行性约束.为验证所提出方法在约束条件下的有效性,在四个典型环境中做了大量实验,以对比分析不同方法在受限动作空间中的控制行为,本节仅展示 HalfCheetah 环境下受限动作空间中的控制行为.

在实验设置中,对控制输入施加双重约束:一方面要求动作始终处于给定的幅值约束范围内,另一方面限制控制输入的变化率,以刻画实际工程系统中的执行器饱和与平滑性要求.与传统强化学习任务侧重于累计回报不同,DiffMPC 将评估重点从"奖励最大化"转向"任务可行性",即是否能够在约束条件下稳定完成 1000 个仿真步长的控制任务.

如图5所示,DiffMPC 与传统强化学习方法(专家策略)均能够完成控制任务,说明在基本任务可达性方面二者具有一致性,图中各分量代表的物理意义可参考文献中实验说明^[10].然而,从动作序列的分

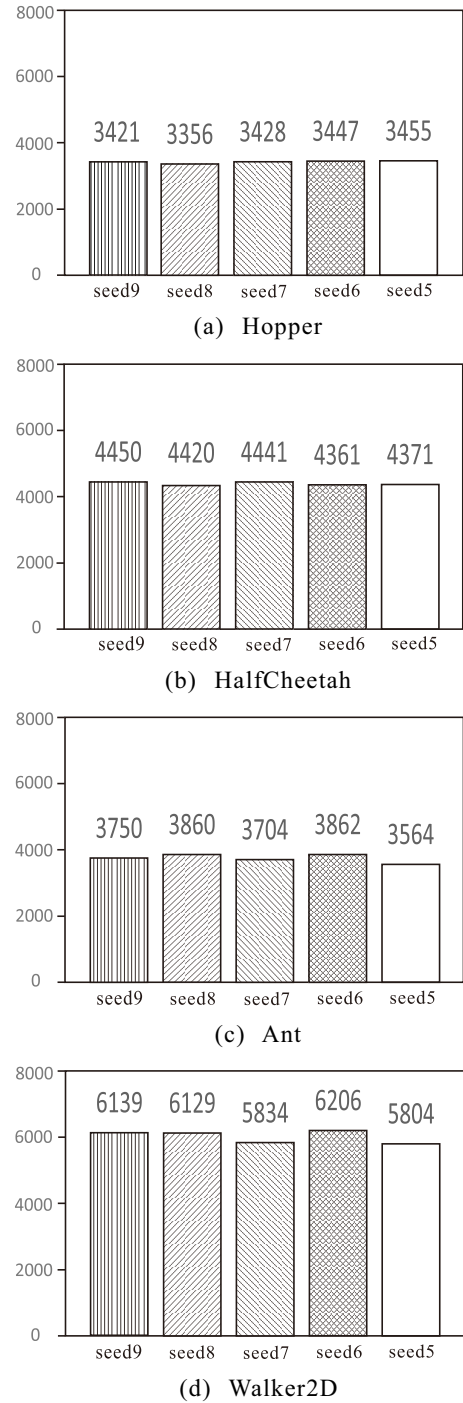


图4 四个环境下仿真通用性验证

表3 在新环境中的泛化性能

Env.	Trained Env.	Unseen Env.	Loss
Hopper	3462±44	3421.4±65	1.2%
HalfCheetah	4502±83	4408±47	2%
Ant	3885±73	3748±184	3.5%
Walker2D	6092±86	6022±218	1.1%

布特性可以观察到显著差异:相比传统强化学习方法,本文方法生成的控制输入在时间维度上更加平滑,且触碰动作约束边界的次数明显更少.这一现象在其它三个并行实验环境中都有类似的发现,表明所提出的方法能够在不依赖额外奖励的情况下,自

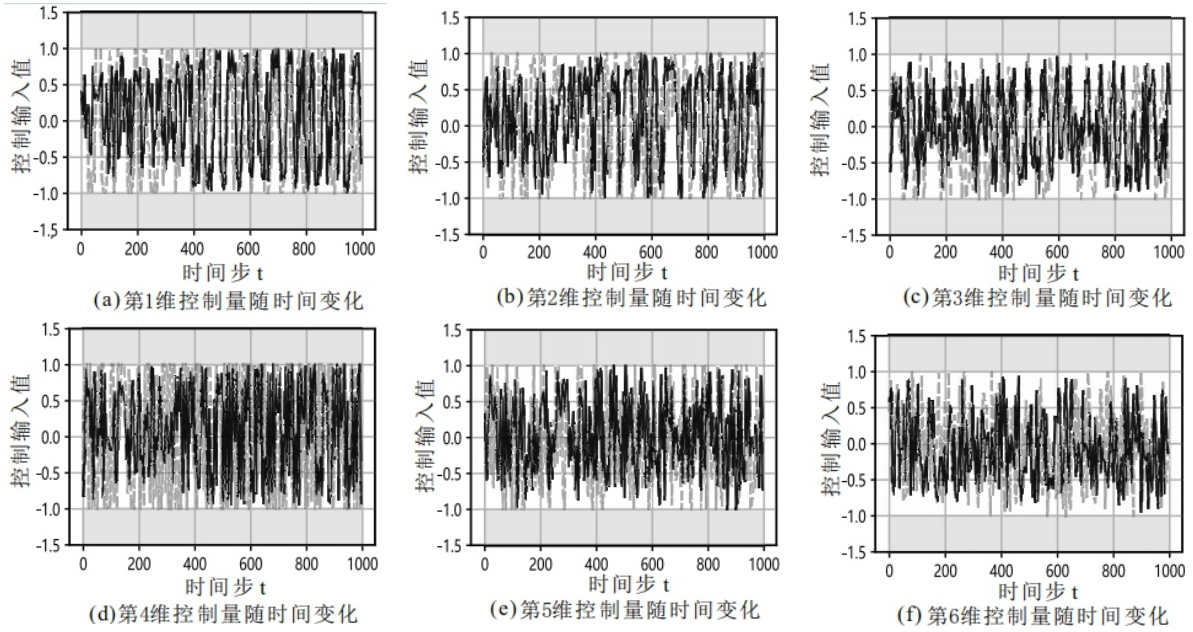


图5 HalfCheetah 系统的控制输入演化 (实线为 DiffMPC, 虚线为专家策略)

然地引导策略在可行域内部演化。

需要指出的是, 该约束行为可通过 MPC 代价函数中约束相关惩罚项进行灵活调节. 在实验中, HalfCheetah 任务对动作约束的要求相对宽松, 而在有些任务中 (如 Ant) 由于其维度更高、耦合更强, 对约束满足性要求更为严格, 因此动作序列分布更加集中. 这说明, DiffMPC 能够根据任务约束强度自适应调整控制行为, 为工程场景中具有强约束、高安全要求的复杂系统控制提供了良好的实践基础。

4 结论

提出了一种新型智能 MPC 框架, 该框架将基于扩散模型的对抗模仿学习与 PSO 增强的 MPC 相结合. 为克服传统 MPC 中建模非线性动力学与设计精确奖励函数的挑战, 引入了一种由专家示范生成的学习型奖励替代信号. 通过利用扩散模型的生成能力, 本方法提升了模仿学习过程的稳定性与表达能力, 奖励信号进一步用于引导 PSO-MPC 控制器, 有效应对复杂控制任务中高维优化的难题. 该两阶段框架首先通过基于扩散的对抗模仿学习获取奖励模型, 然后训练策略, 为实时规划提供强有力的先验, 从而将无模型模仿学习的灵活性与基于模型控制的安全性和结构性结合在一起。

在四个 MuJoCo 基准环境上的大量实验结果表明, 本方法在累积奖励上持续优于标准模仿学习基线方法. 此外, 它在此前未见的环境状态与目标上展现出较强的泛化能力, 并通过自适应 PSO 配置在控制性能与计算成本之间保持了合理的权衡. 这些结果凸显了基于扩散的奖励替代与预测优化结合的潜

力, 能够在具有挑战性且奖励稀疏的环境中实现稳健且具备泛化能力的控制。

参考文献 (References)

- [1] 朱建勇, 张琳, 陆荣秀, 等. 基于 Laguerre 的多组分稀土萃取分布式模型预测控制[J]. *控制与决策*, 2025, 40(3): 1005-1014.
(Zhu J Y, Zhang L, Lu X R, et al. Distributed model predictive control for multi-component rare earth extraction based on Laguerre[J]. *Control and Decision*, 2025, 40(3): 1005-1014.)
- [2] Mohseni F, Frisk E, Nielsen L. Distributed cooperative MPC for autonomous driving in different traffic scenarios[J]. *IEEE Transactions on Intelligent Vehicles*, 2021, 6(2): 299-309.
- [3] 丁博文, 付东飞, 金志豪, 等. 基于改进 Koopman 算子在线预估器的海洋浮体路径跟踪预测控制算法[J]. *控制与决策*, 2025, 40(3): 863-870.
(Ding B W, Fu D F, Jin Z H, et al. Path-following predictive control of marine floating body based on online predictor of Koopman operator[J]. *Control and Decision*, 2025, 40(3): 863-870.)
- [4] Schwenzer M, Ay M, Bergs T, et al. Review on model predictive control: An engineering perspective[J]. *The International Journal of Advanced Manufacturing Technology*, 2021, 117(5): 1327-1349.
- [5] Wang G M, Jia Q S, Qiao J F, et al. Deep learning-based model predictive control for continuous stirred-tank reactor system[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2021, 32(8): 3643-3652.
- [6] Nagabandi A, Kahn G, Fearing R S, et al. Neural network dynamics for model-based deep reinforcement

- learning with model-free fine-tuning[C]. IEEE International Conference on Robotics and Automation. Brisbane, 2018: 7559-7566.
- [7] Lin M, Sun Z Q, Xia Y Q, et al. Reinforcement learning-based model predictive control for discrete-time systems[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2024, 35(3): 3312-3324.
- [8] Ho J, Ermon S. Generative adversarial imitation learning[C]. *Advances in Neural Information Processing Systems*. Red Hook, 2016: 4565-4573.
- [9] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[C]. *Advances in Neural Information Processing Systems*. Red Hook, 2020: 6840-6851.
- [10] Wang B Z, Wu G Q, Pang T, et al. DiffAIL: Diffusion adversarial imitation learning[C]. *Proceedings of the AAAI Conference on Artificial Intelligence*. Vancouver, 2024: 15447-15455.
- [11] Puterman M L. *Markov decision processes: Discrete stochastic dynamic programming*[M]. Hoboken: John Wiley & Sons, 2014.
- [12] Chen M H, Hsieh P C, Lai C M, et al. Diffusion-reward adversarial imitation learning[C]. *Advances in Neural Information Processing Systems 37*. Vancouver, 2024: 95456-95487.
- [13] 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*. Stockholm, 2018: 1861-1870.
- [14] Todorov E, Erez T, Tassa Y. MuJoCo: A physics engine for model-based control[C]. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura-Algarve, 2012: 5026-5033.
- [15] Raffin A, Hill A, Gleave A, et al. Stable-baselines3: Reliable reinforcement learning implementations[J]. *Journal of Machine Learning Research*, 2021, 22(268): 1-8.
- [16] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]. *International Conference on Machine Learning*. Beijing, 2014: 387-395.
- [17] Kingma D P, Ba J. Adam: A method for stochastic optimization[J/OL]. 2014, arXiv: 1412.6980,.

作者简介

任凯楠 (2001-), 男, 硕士生, 主要研究方向为数据驱动的建模、预测控制, E-mail: 919001418@qq.com;

付东飞 (1984-), 男, 副教授, 博士, 主要研究方向为预测控制理论及其应用, E-mail: fudongfei@ouc.edu.cn;

金志豪 (2001-), 男, 硕士生, 主要研究方向为强化学习、风机预测控制, E-mail: 21230911006@stu.ouc.edu.cn;

黎明 (1975-), 男, 教授, 博士, 主要研究方向为智能信号处理与智能控制、海洋测控技术, E-mail: limingneu@ouc.edu.cn.