Software Engineering and Applications 软件工程与应用, 2024, 13(5), 704-719
Published Online October 2024 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2024.135072

Hans汉斯

基于去噪概率扩散模型的平均场多智能体强化学习算法

单国强[1]，缪霏阳[2]，张子胤[2]，李大鹏[2]

[1]南京邮电大学波特兰学院，江苏 南京
[2]南京邮电大学通信与信息工程学院，江苏 南京

收稿日期：2024年9月11日；录用日期：2024年10月22日；发布日期：2024年10月31日

摘 要

为了解决基于平均场的多智能体强化学习(M³-UCRL)算法中的环境动力学模型对下一时刻状态预测不精确和策略学习样本过少的问题。本文利用了去噪概率扩散模型(Denoising Diffusion Probabilistic Models, DDPM)的数据生成能力，提出了一种基于DDPM的平均场多智能体强化学习(DDPM-M³RL)算法。该算法将环境模型的生成表述为去噪问题，利用DDPM算法，提高了环境模型对下一时刻状态预测的精确度，也为后续的策略学习提供了充足的样本数据，提高了策略模型的收敛速度。实验结果表明，该算法可以有效提高环境动力学模型对下一时刻状态预测的精确度，根据环境动力学模型生成的状态转移数据可以为策略学习提供充足的学习样本，有效提高了导航策略的性能和稳定性。

关键词

多智能体强化学习，去噪概率扩散模型，平均场控制，策略学习

Mean Field Multi-Agent Reinforcement Learning Algorithm Based on Denoising Diffusion Probability Models

Guoqiang Shan[1], Feiyang Miao[2], Ziyin Zhang[2], Dapeng Li[2]

[1]Portland Institute, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu
[2]College of Telecommunications & Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu

Received: Sep. 11th, 2024; accepted: Oct. 22nd, 2024; published: Oct. 31st, 2024

文章引用: 单国强, 缪霏阳, 张子胤, 李大鹏. 基于去噪概率扩散模型的平均场多智能体强化学习算法[J]. 软件工程与应用, 2024, 13(5): 704-719. DOI: 10.12677/sea.2024.135072

Abstract

To solve the problems of inaccurate prediction of the next state by the environment dynamics model and too few samples for policy learning in the mean field based multi-agent reinforcement learning (M^3-UCRL) algorithm, this paper takes advantage of the data generation capability of denoising diffusion probability models (DDPM) and proposes a mean field multi-agent reinforcement learning (DDPM-M^3RL) algorithm based on DDPM. The algorithm formulates the generation of the environment model as a denoising problem. By using the DDPM algorithm, the accuracy of the environment model's prediction of the next state is improved, and sufficient sample data is provided for subsequent policy learning, which improves the convergence speed of the policy model. Experimental results show that the algorithm can effectively improve the accuracy of the environment dynamics model's prediction of the next state, and the state transition data generated by the environment dynamics model can provide sufficient learning samples for policy learning, which effectively improves the performance and stability of the navigation strategy.

Keywords

Multi-Agent Reinforcement Learning, DDPM, Mean-Field Control, Policy Learning

1. 引言

由于在机器人编队、分布式控制、资源管理、协同决策系统和数据挖掘等方面的潜在和实际应用，多智能体强化学习在过去几年中引起了广泛关注[1]-[3]。但随着任务场景越来越复杂，智能体数量的增加，带来了状态空间和动作空间的维度指数级增加，使得分析和计算越来越复杂和困难[4]。目前，借由平均场控制(Mean-Field Control, MFC)近似，可以对多智能体系统进行一定程度的化简[5]。在 MFC 中，我们用智能体的分布状态来描述系统状态，从而可以只关注一个代表智能体[6]。虽然 MFC 显著降低了算法在训练时的状态空间维度，但是多智能体强化学习的挑战仍有很多。

根据环境模型是否已知，强化学习可以分为无模型强化学习(Model-free Reinforcement Learning, MfRL)和模型化强化学习(Model-based Reinforcement Learning, MbRL)两大类[7]。在无模型强化学习算法中，由于环境模型是未知的，因此需要与真实环境进行大量交互来取得足够数量的训练样本，但这一过程中必然有大量无效和错误的动作产生，这需要消耗大量的时间和财力。而在真实环境中训练还有损坏智能系统的风险。另外，由于时间和财力的限制，训练样本的数量也难以保证，智能体无法从少量训练样本中提取足够的信息进行准确策略更新，从而导致最终的策略难以令人满意。

相比之下，模型化强化学习方法会先让智能体与环境交互采集数据，通过这些数据对环境精准建模后，智能体便可直接与环境模型交互生成所需训练样本，这意味着无需与真实环境互动就可以进行策略的学习和规划，不仅能解决在真实环境中训练损坏智能系统的风险，还能显著提高了样本数据的利用效率。

在文献[8]中，作者提出了基于平均场模型的多智能体强化学习算法(Model-Based Multi-Agent Mean-Field Upper-Confidence RL algorithm, M^3-UCRL)，该算法将平均场控制与基于模型的强化学习进行了结合，

解决了多智能体状态空间维度太大和训练样本利用率低的问题。但是该算法所建立的环境动力学模型是基于多层感知机的神经网络，随着任务场景日渐复杂，环境模型的精确度难以保证。为此，研究人员提出了一系列减小模型误差、提高环境模型准确性的方法，如贝叶斯网络、高斯过程等。这些工作在各自应用领域已经取得较好成果，但是面向大规模动态环境，如何学得通用的、高效的环境模型，仍是该领域的研究重点。

扩散(Diffusion)模型的概念最早在 2015 年的文献[9]中被提出。现在已经成为一类强大的生成模型，近年来引起了广泛的关注。这些模型所采用的去噪框架可以有效地逆转多步噪声过程来生成新的数据[10]。与变分自动编码器(Variational Autoencoders, VAE) [11]以及生成对抗网络(Generative Adversarial Networks, GAN) [12]这些早期的生成模型相比，扩散模型在生成高质量样本方面的能力表现得十分突出且拥有者足够的稳定性。因此，他们在包括计算机视觉[13]，自然语言处理[14] [15]，音频生成[16] [17]等多个领域取得了显著的进步和巨大的成功，推动生成式人工智能(Artificial Intelligence Generated Content, AIGC)技术取得了突破性发展。

本文的主要贡献和研究如下。

1) 提出一种基于去噪概率扩散模型的平均场多智能体强化学习算法(DDPM-M^{3}RL)。为多无人设备合作导航提供了一种新的解决方案。

2) 基于 DDPM 算法，将环境模型的生成表述为去噪问题，在训练后有效提高了环境模型的精确度。此外，在训练得到稳定的环境模型后，智能体可与环境模型直接进行交互，从而廉价地获得后续训练所需的样本数据。

3) 设计了无人车集群合作导航任务实验，仿真结果表明，本文的算法可以准确建模环境动力学模型，减少导航策略收敛的回合数，并让智能体在任务中获得更高期望奖励。

2. 系统模型与问题描述

2.1. 系统模型

强化学习通常被建模为具有完全可观察状态空间的马尔可夫决策过程(Markov Decision Process, MDP)，表示为 $M=(S,A,F,R,\gamma)$，其中 S 是状态空间，是动作空间，F 是具有离散时间系统动态特性的状态转移函数，即给定动作 $a_t \in A$ 时，状态 $s_{t+1}=F(s_t,a_t)$。

$R(s_t,a_t)$ 定义了奖励函数，$\gamma \in (0,1]$ 是未来奖励的折扣因子。我们考虑具有时间范围 H，紧凑状态空间 $S \subseteq R^p$ 和紧凑动作空间 $A \subseteq R^q$ 的情景平均场控制(MFC)问题，这些空间对于系统中的所有智能体都是通用的。在标准 N 个智能体设置中，系统在时间 $h \in \{0,...,H-1\}$ 中用各个智能体的状态 $\left(s_h^{(1)},...,s_h^{(N)}\right) \in S$ 来描述，它随着智能体 N 的数量呈指数增长。在 MFC 中，我们假设所有智能体都是相同的，并且总体是渐近无限的，即 $N \to \infty$，因此，智能体的状态可以用平均场分布来描述：

$$\mu_{t,h}(s)=\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{n}\delta\left\{s_{t,h}^{i}=s\right\} \tag{1}$$

其中 $\mu_{t,h}$ 属于 S 上的概率测度 $P(S)$ 的空间。由于 MFC 中相同代理的假设，我们可以关注群体中与代理分布交互的代表性代理，而不是单个代理和交互。

系统动力学模型。在每个回合 t，代表智能体会从一组可接受的策略选择一个策略配置 $\pi_t=(\pi_{t,0},...,\pi_{t,H-1})$。在每个时间 h，代表智能体选择一个动作 $a_{t,h}=\pi_{t,h}(s_{t,h},\mu_{t,h})$。然后环境返回奖励 $r(s_{t,h},a_{t,h},\mu_{t,h})$，同时智能体观察新的环境状态 $s_{t,h+1}$ 和平均场分布 $\mu_{t,h+1}$。

在 MFC 中，系统的动态通常由 McKean-Vlasov 类型的随机过程给出，并且取决于智能体的状态、动

作和平均场分布：

$$s_{h+1} = f\left(s_h, a_h, \mu_h\right) + \omega_h \tag{2}$$

其中 ω_h 是独立同分布的加性噪声向量。这与标准的基于单智能体模型的 RL 是不同的，其中的动力学仅取决于智能体的动作和状态。关键的是，由于这项工作的重点是在 MFC 中学习，我们假设真实的动态是未知的，目标是探索空间并通过多个片段了解环境模型 F。为此，代表智能体依赖于在第 t 回合中与真实系统的交互收集到的随机观测值 $D_t = \left\{\left(\left(s_{t,h}, a_{t,h}, \mu_{t,h}\right), s_{t,h+1}\right)\right\}_{h=0}^{H-1}$。

最后，在每一轮之后，我们假设整个系统都被重置，并且智能体的初始状态 s_0 是从已知的初始分布 μ_0 中得出的，即 $s_0 \sim \mu_0$ 在每一轮中都保持相同。

平均场流。由于在 MFC 中，所有智能体都是相同的，在共同的环境中交互，并遵循相同的策略 $\pi_t = \left(\pi_{t,0}, ..., \pi_{t,H-1}\right)$，因此后续的平均场分布满足以下的平均场流特性：

$$\mu_{h+1}\left(ds'\right) = \int_{s \in S} \mu_h\left(ds\right) P\left(s_{h+1} \in ds'\right) \tag{3}$$

Table 1. Main parameters
表 1. 主要参数

符号	含义
S	状态空间
A	动作空间
F	状态转移函数
R	奖励函数
U	状态分布空间
p	状态空间维度
q	动作空间维度
T	DDPM 去噪时间步
s_h	第 h 个时间步，代表智能体与真实环境交互时的状态
μ_h	第 h 个时间步，真实环境中系统状态分布
π_h	第 h 个时间步，代表智能体与真实环境交互时的可行策略
a_h	第 h 个时间步，代表智能体与真实环境交互时执行的动作
r_h	第 h 个时间步，获得的来自真实环境的奖励
$\tilde{s}_h$	第 h 个时间步，代表智能体与建立的环境模型交互时的状态
$\tilde{\mu}_h$	第 h 个时间步，建立的环境模型中系统状态分布
$\tilde{a}_h$	第 h 个时间步，代表智能体与建立的环境模型交互时执行的动作
ω_h	真实环境中的加性噪声
$\tilde{\omega}_h$	建立的环境模型中的加性噪声
x^T	第 T 个时间步的噪声分布

我们用 $\Phi\left(\mu_{t,h}, \pi_{t,h}, f\right)$ 来表示公式(1)中的转换函数，则上式可简化为 $\mu_{h+1} = \Phi\left(\mu_h, \pi_h, f\right)$。

评价指标。当给定策略 $\pi = \left(\pi_1, ..., \pi_{H-1}\right)$ 时，代表智能体的表现通过预期累积奖励来衡量。其中期望取代了转换中的噪声和初始分布 $s_0 \sim \mu_0$。通过考虑代表智能体的观点，目标是发现一个全局最优策略 $\pi*$，最大化预期总奖励，即：

$$J(\pi)=E\left[\sum_{h=0}^{H-1}R(s_h,a_h,\mu_h)\right] \tag{4}$$

其中：$a_h=\pi_h(s_h,\mu_h)$，$s_{h+1}=f(s_h,a_h,\mu_h)+\omega_h$，$\mu_{h+1}=\Phi(\mu_h,\pi_h,f)$。

每回合结束后，代表智能体通过选择累计奖励最大的策略来进行策略优化，即：

$$\pi^*\in\arg\max_{\pi}J(\pi) \tag{5}$$

本文主要参数见表 1 所示。

2.2. 问题描述

本文所介绍的算法首先对环境动力学模型进行建模，表示为 f^{δ}，δ 为环境动力学模型的参数。在使用 DDPM 进行训练的时候，我们将求解环境动力学模型 f^{δ} 转化为求解去噪过程中的已参数化的噪声模型 ε_θ，文献[13]提出了简化的学习 ε_θ 的损失函数，它是变分下界(Evidence Lower Bound, ELBO)的加权版本。即我们的优化目标为：

$$L_{\min}(\theta)=\min\left\{E_{x^0,\varepsilon,t}\left[\left\|\varepsilon-\varepsilon_\theta\left(\sqrt{\bar{\alpha}^t}x^0+\sqrt{1-\bar{\alpha}^t}\varepsilon,t\right)\right\|^2\right]\right\} \tag{6}$$

当噪声模型 ε_θ 训练完成后，我们即可使用无分类器引导采样的方法得到环境模型 f^{δ}。之后，在每个训练回合 t 中，智能体可直接与利用 DDPM 训练得到的环境动力学模型 f^{δ} 以及当前可采用的策略 π_t 来进行模拟，获得下一时刻的状态，累积奖励等信息。当该回合结束时，代表智能体会选择可以使累积奖励达到最大值的策略来进行策略更新，即解决以下问题：

$$\pi_t=\arg\max_{\pi,f}E\left[\sum_{h=0}^{H-1}r(\tilde{s}_{t,h},\tilde{a}_{t,h},\tilde{\mu}_{t,h})\right] \tag{7}$$

其中：$\tilde{a}_{t,h}=\pi_{t,h}(\tilde{s}_{t,h},\tilde{\mu}_{t,h})$，$\tilde{s}_{t,h+1}=F(\tilde{s}_{t,h},\tilde{a}_{t,h},\tilde{\mu}_{t,h})+\tilde{\omega}_{t,h}$，$\tilde{\mu}_{t,h+1}=\Phi(\tilde{\mu}_{t,h},\pi_{t,h},\tilde{f}^{\delta})$。

3. 基于去噪概率扩散模型的平均场多智能体强化学习算法

本节将对基于去噪概率扩散模型的平均场多智能体强化学习算法(DDPM-M^{3}RL)进行详细介绍。本算法是对基于平均场的多智能体强化学习(M^3-UCRL)算法[8]的改进，原算法的系统动力学模型仅依靠由多层感知机进行训练，当智能体数量增加时，其模型精度有很大的下降。本算法利用了 DDPM 在数据生成方面的优势，通过加噪—去噪的机制，有效提高了环境动力学模型的精度，提高了智能体学习策略的效率和准确度。本节将首先介绍去噪扩散模型的基本理论，然后介绍基于 DDPM 来学习环境动力学模型的过程，最后介绍如何利用学习到的环境动力学模型来进行策略优化。

根据强化学习的马尔可夫性质，在解决实际问题的时候，我们认为系统下一时刻的状态仅与当前状态有关，这极大简化了系统模型的复杂度。在利用 MFC 对多智能体系统进行化简后，我们可以不用关注每个智能体的具体行为而将精力聚焦在代表智能体上。本文首先让智能体与真实环境进行交互，获得准确数据，然后利用 DDPM 学习环境模型，当学习的环境模型达到预期后，我们让代表智能体直接与其进行交互，获得足够的样本数据来进行策略优化。算法流程如图 1 所示。

3.1. 去噪概率扩散模型基本理论

3.1.1. 去噪概率扩散模型

去噪扩散概率模型可分为前向过程(也称前向扩散过程、加噪过程等)和逆向过程(也称逆向扩散过程、

去噪过程等)两部分。如图 2 所示，前向过程是一个逐步加噪的过程，目的是使得原有的样本数据分布转换为一个简单的标准高斯分布。逆向过程是去噪的过程，从标准高斯分布中进行采样，每一步去除一个很小的高斯噪声，逐步贴近真实数据分布，进而得到其真实数据分布中的样本，从而达到生成数据的目的。

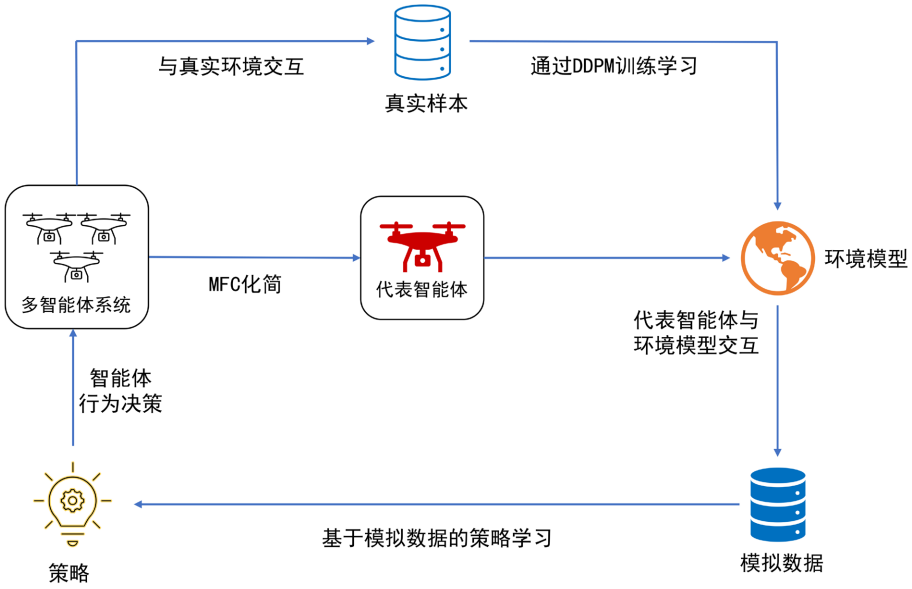

Figure 1. DDPM-M^{3}RL algorithm flow chart
图 1. DDPM-M^{3}RL 算法流程图

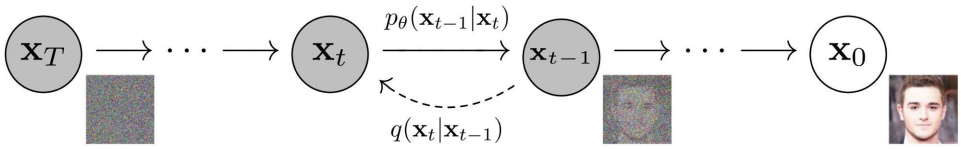

Figure 2. DDPM basic structure
图 2. DDPM 基本结构

假设真实数据 x^0 是从一个潜在分布 $q\left(x^0\right)$ 中采样的结果，DDPM [13]利用一个参数化的扩散过程，表示为 $p_\theta\left(x^0\right)=\int p\left(x^T\right)\prod_{t=1}^{T} p_\theta\left(x^{t-1}\mid x^t\right)dx^{1:T}$，对纯噪声 $x^T=N\left(0,\mathrm{I}\right)$ 如何去噪到真实数据 x^0 进行建模。扩散过程的每一步用 x^T 表示，T 表示总步数。值得注意的是，扩散过程和强化学习都涉及时间步长。因此，我们将扩散步长记为上标，强化学习时间步长记为下标。序列 $x^{T:0}$ 被定义为具有学习到的高斯转移的马尔可夫链，其特征为 $p_\theta\left(x^{t-1}\mid x^t\right)=N\left(\mu_\theta\left(x^t,t\right),\Sigma\left(x^t,t\right)\right)$。如果过程反转为 $x^{0:T}$，则每一步由前向转移 $q\left(x^t\mid x^{t-1}\right)$ 定义，其表达式为根据方差矩阵 $\beta^{1:T}$ 向数据中添加高斯噪声：

$$x^t=\sqrt{\alpha^t}x^{t-1}+\sqrt{1-\alpha^t}\varepsilon^t \tag{8}$$

其中 $\alpha^{t-1}=1-\beta^t$，$\varepsilon^t\sim N\left(0,\mathrm{I}\right)$。根据式(8)，我们可以得到 x^0 到 x^t 的直接映射：

$$x^t=\sqrt{\bar{\alpha}^t}x^0+\sqrt{1-\bar{\alpha}^t}\varepsilon\left(x^t,t\right) \tag{9}$$

其中 $\bar{\alpha}^t=\prod_1^T\alpha^i$，根据贝叶斯定理以及 x^t 与 x^0 的关系，我们有：

$$q\left(x^{t-1}\mid x^t,x^0\right)=N\left(\frac{1}{\sqrt{\alpha^t}}\left(x^t-\frac{\beta^t}{\sqrt{1-\bar{\alpha}^t}}\varepsilon\left(x^t,t\right)\right),\beta^t\mathrm{I}\right) \tag{10}$$

公式(10)允许我们从高斯噪声中采样 x^t，并逐步去噪，直到得到 x^0。然而，噪声 $\varepsilon\left(x^t,t\right)$ 是未知的。为了解决这个问题，使用了参数化网络 ε_θ 来预测噪声。文献[13]提出了以下简化的学习 ε_θ 的损失函数，它是变分下界(ELBO)的加权版本：

$$L(\theta)=E_{x^0,\varepsilon,t}\left[\left\|\varepsilon-\varepsilon_\theta\left(\sqrt{\bar{\alpha}^t}x^0+\sqrt{1-\bar{\alpha}^t}\varepsilon,t\right)\right\|^2\right] \tag{11}$$

其中 ε 是从高斯分布 $N(0,\mathrm{I})$ 中采样的。

3.1.2. 扩散模型的引导采样方法

基于扩散模型的引导采样(guidance sampling)考虑从条件分布 $p(x\mid y)$ 进行采样，其中 y 是生成样本的期望属性。引导的两个主要类别是分类器引导和无分类器引导。图 3 为扩散模型引导采样示意图。

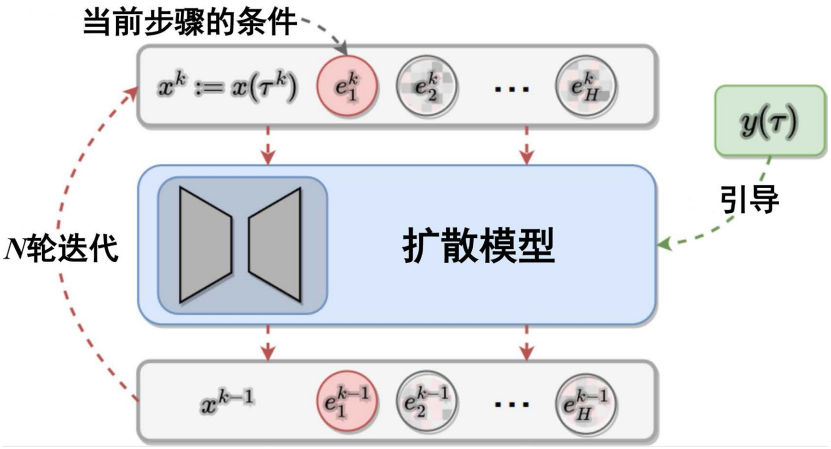

Figure 3. Schematic diagram of diffusion model guided sampling
图 3. 扩散模型引导采样示意图

分类器引导(Classifier guidance)。分类器引导采样依赖于一个可微的分类器模型 $p_\Phi(y\mid x)$。具体来说，由于需要对每个去噪步骤进行指导，因此分类器模型 $p\left(y\mid x^t\right)$ 通过 x 和对应的标签 y 的噪声样本进行训练。条件逆过程可以写为：

$$p_{\theta,\Phi}\left(x^{t-1}\mid x^t,y\right)=Zp_\theta\left(x^{t-1}\mid x^t\right)p_\theta\left(y\mid x^{t-1}\right) \tag{12}$$

式中 Z 为归一化因子。文献[18]对公式(11)进行了高斯分布的近似：

$$p_{\theta,\Phi}\left(x^{t-1}\mid x^t,y\right)=N\left(\mu^t+\omega\Sigma^t g,\Sigma^t\right) \tag{13}$$

其中 $g=\nabla_{x^t}\log p_\theta\left(y\mid x^t\right)|_{x^t=\mu^t}$，$\omega$ 是控制条件强度的引导尺度。μ^t 和 Σ^t 分别是公式(10)的均值和协方差矩阵。

无分类器引导(Classifier-free guidance, CFG)。无分类器引导采样依赖于额外的条件噪声模型 $\varepsilon_\theta\left(x^t,y,t\right)$。在实际中，有条件模型和无条件模型共享同一组参数，无条件模型通过设置 y 为空值来表示。文献[19]提出公式(11)中的噪声学习目标为 $p\left(x^t\right)$ 的一个尺度得分函数，即 $\sigma^t=\sqrt{\beta_t}$ 和 $\varepsilon\left(x^t,t\right)=-\sigma^t\nabla_{x^t}\log p\left(x^t\right)$。根据贝叶斯定理，我们有：

$$\nabla_{x^t}\log p\left(y\mid x^t\right)=-1/\sigma^t\left(\varepsilon\left(x^t,y,t\right)-\varepsilon\left(x^t,t\right)\right) \tag{14}$$

根据公式(12)，可得引导噪声预测器为 $\bar{\varepsilon}\left(x^t,y,t\right)=\varepsilon_\theta\left(x^t,t\right)-\omega\sigma^t\nabla_{x^t}\log p\left(y\mid x^t\right)$。将得分函数替换为

噪声模型预测，分类器引导采样中使用的噪声可以写为：

$$\hat{\varepsilon}_{\omega}\left(x^{t},y,t\right)=\omega\varepsilon_{\theta}\left(x^{t},y,t\right)+\left(1-\omega\right)\varepsilon_{\theta}\left(x^{t},t\right) \tag{15}$$

3.2. 基于 CFG-DDPM 及 MFC 的环境动力学模型

环境动力学模型对于基于模型的强化学习算法来说至关重要，只有环境动力学模型能够准确描述环境状态，智能体与之交互才能得到正确的状态数据，从而准确地指导策略学习。为了提高环境动力学模型的精确度，将无分类器引导去噪概率扩散模型(Classifier-free guidance Denoising Diffusion Probabilistic Models, CFG-DDPM)与基于模型且用 MFC 化简的多智能体强化学习相结合。我们将环境动力学模型表示为 $P\left(s_{k+1}\mid s_{k},a_{k},\mu_{k}\right)$，它指在当前的 k 时刻，已知智能体当前状态 s_{k} 和平均场状态 μ_{k}，在当前策略下，智能体根据策略执行动作 a_{k} 后，环境进入下一状态 s_{k+1} 的概率。在 CFG-DDPM 与环境模型结合后，我们的目标变为：已知 s_{k},a_{k},μ_{k}，正确预测出下一时刻的状态 s_{k+1}。在训练过程中，扩散模型通常有 T 个时间步来完成加噪/去噪过程，因此，在加噪过程中，我们将智能体与真实环境互动得到的下一时刻状态 s_{k+1} 作为 DDPM 中的 x_{k}^{0}，将 s_{k},a_{k},μ_{k} 与时间步 t 联合编码为新的向量 t_{k}，对公式(9)中高斯噪声的分布进行更新，即：

$$x_{k}^{t}=\sqrt{\bar{\alpha}^{t}}x_{k}^{0}+\sqrt{1-\bar{\alpha}^{t}}\varepsilon\left(x_{k}^{t},t_{k}\right) \tag{16}$$

去噪过程中，同样利用 t_{k} 对公式(10)中高斯噪声的分布进行更新，即：

$$q\left(x_{k}^{t-1}\mid x_{k}^{t},x_{k}^{0}\right)=N\left(\frac{1}{\sqrt{\alpha^{t}}}\left(x^{t}-\frac{\beta^{t}}{\sqrt{1-\bar{\alpha}^{t}}}\varepsilon\left(x_{k}^{t},t_{k}\right)\right),\beta^{t}\mathrm{I}\right) \tag{17}$$

通过对噪声 ε_{θ} 的损失函数 $L\left(\theta\right)$ 进行优化，ε_{θ} 将逐渐接近真实噪声。

在完成噪声模型的学习后，使用无分类器引导采样方法进行采样，将 $\left(s_{k},a_{k},\mu_{k}\right)$ 即公式(15)中代表约束条件变量 y 作为输入，随机生成的高斯噪声经过 T 步去噪过程即可采样得到环境下一时刻状态 s_{k+1}。

3.3. 策略优化

本算法采用梯度下降的方法对策略函数进行优化。当环境动力学模型经过 CFG-DDPM 训练趋于稳定后，我们将该环境模型记为 F^{θ}。在每个回合 t 中，使用 MFC 化简得到的代表智能体选择当前策略 $\pi_{t}=\left(\pi_{t,0},\ldots,\pi_{t,H-1}\right)$ 中可达到最高可能的累积回报的策略直接与环境模型 F^{θ} 进行交互，得到下一时刻的状态 s_{t+1}，代表智能体通过解决如下问题来进行策略优化：

$$\pi_{t}=\arg\max_{\pi,f}E\left[\sum_{h=0}^{H-1}r\left(\tilde{s}_{t,h},\tilde{a}_{t,h},\tilde{\mu}_{t,h}\right)\right] \tag{18}$$

其中 $\tilde{a}_{t,h}=\pi_{t,h}\left(\tilde{s}_{t,h},\tilde{\mu}_{t,h}\right)$，$\tilde{s}_{t,h+1}=F\left(\tilde{s}_{t,h},\tilde{a}_{t,h},\tilde{\mu}_{t,h}\right)+\tilde{\omega}_{t,h}$，$\tilde{\mu}_{t,h+1}=\Phi\left(\tilde{\mu}_{t,h},\pi_{t,h},\tilde{f}\right)$。

然后将策略函数和期望的累计奖励参数化，记为 π_{θ} 和 $J\left(\theta\right)$，使用梯度下降算法对策略的参数 θ 进行优化，更新得到在下一轮中进行使用的新策略。

3.4. DDPM-M^{3}RL 算法概述

本算法是对基于平均场的多智能体强化学习(M^3-UCRL)算法的改进，利用扩散模型对环境动力学模型进行建模。通过扩散模型加噪/去噪的过程，训练得到了期望去除的噪声参数，再通过无分类器的引导采样，完成了在已知当前环境状态和动作后对下一时刻环境状态的准确预测。这不仅减少了智能体与真实环境交互的需求，降低了时间和经济成本，也为智能体在策略学习中提供了足量的样本数据，提高了算法的性能。基于 DDPM 的动力学模型的训练算法如算法 1 所示，采样算法如算法 2 所示，DDPM-M^{3}RL 算法的伪代码如算法 3 所示。

算法 1：基于去噪概率扩散模型的动力学模型训练算法

repeat

$x_0 \sim q(x_0)$；

$t \sim \text{Uniform}(\{1,...,T\})$；

$\varepsilon \sim N(0,\text{I})$；

采用梯度下降算法优化：

$\nabla_\theta \left\| \varepsilon - \varepsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, t\right)\right\|^2$；

until 收敛

算法 2：基于去噪概率扩散模型的动力学模型采样算法

$x^T \sim N(0,\text{I})$

for $t = T,...,1$ **do**

$\text{z} \sim N(0,\text{I})$ if $t > 1$, else $\text{z} = 0$；

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t,t) + \sigma_t \text{z}\right);$$

end for

return x_0

算法 3：基于去噪概率扩散模型的平均场多智能体强化学习算法

输入：基于 DDPM 的动力学模型，时间跨度 H，奖励函数 $r(s_{t,h}, a_{t,h}, \mu_{t,h})$，初始状态 $s_0 \sim \mu_0$

for $t = 1,2,...$ **do**

根据当前的环境动力学模型及奖励函数选择策略

$\pi_t = (\pi_{t,0},...,\pi_{t,h-1})$；

使用算法 1 对环境动力学模型进行训练；

for $h = 0,2,...,H-1$ **do**

使用算法 2 采样得到 $s_{t,h+1}$；

$a_{t,h} = \pi_{t,h}(s_{t,h}, \mu_{t,h}) + \omega_{t,h}$；

$\mu_{t,h+1} = \Phi(\mu_{t,h}, \pi_{t,h}, f)$；

end for

用新的观测值 $\{(s_{t,h}, a_{t,h}, \mu_{t,h}), s_{t,h+1}\}_{h=0}^{H-1}$ 更新智能体模型；

重置系统 $\mu_{t+1,0} \leftarrow \mu_0$ 以及 $s_{t+1,0} \sim \mu_{t+1,0}$；

end for

4. 仿真与评估

为了验证本文所提的 DDPM-M^{3}RL 算法的性能，本节将选用无人车集群合作导航任务为应用场景，使用 OpenAI 旗下多智能体强化学习实验平台 MPE 中的合作导航场景作为仿真环境，并与使用了多层感知机拟合环境模型的 M^3-UCRL 算法进行对比。

4.1. 实验设置

4.1.1. 实验平台

操作系统：Ubuntu22.04；CPU：Intel i5-12490f；GPU：Nvidia RTX 3060ti；Python 版本：3.10。

4.1.2. 场景设置

我们将场景设置为 2 km × 2 km 的矩形场地。场景中共 15 辆无人车，每辆车对应一个目的地。每训练回合的初始化时会给每辆车随机安排起始位置和目的地。要求这 15 辆无人车同时启动，在不离开边界，不相互碰撞的情况下，在尽可能短的时间内到达自己的目的地。实验一共设置了 1000 个回合，每个回合设置了 30 个时间步。图 4 为场景的二维化展示，圆点代表导航目的地，圆圈代表无人车。

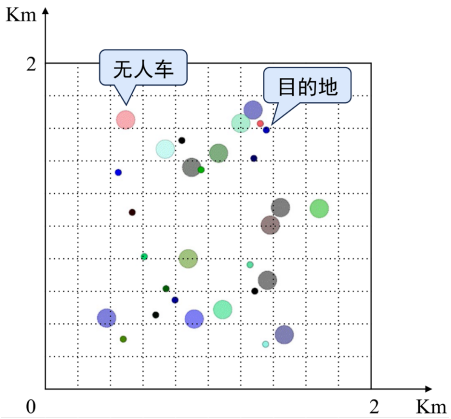

Figure 4. Schematic diagram of a 2D grid-based cooperative navigation scenario
图 4. 二维网格化合作导航场景示意图

4.1.3. 状态空间、动作空间、奖励函数设置

根据强化学习的性质，我们将无人车集群合作导航任务建模成一个马尔可夫决策过程。状态空间 S 包含了场地空间状态和无人车自身的状态。其中场地空间状态包含每辆无人车与其对应目的地的距离 d_{dst}，与其他无人车的距离 d_{obj}，与边界的距离 d_{edge} 以及是否到达目的地；无人车自身状态信息包含无人车当前的位置坐标 (x,y) 和当前速度 (v_x,v_y)。其中，无人车 i 与无人车 j 之间的距离为 $d_{obj}^{i,j}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$，无人车 i 与其对应目的地 (X_i,Y_i) 之间的距离为 $d_{dst}=\sqrt{(x_i-X_i)^2+(y_i-Y_i)^2}$。

对于动作空间，我们选定 4 种速度 $\{0\text{ m/s},10\text{ m/s},20\text{ m/s},40\text{ m/s}\}$ 和 8 种方向 $\left\{0,\frac{\pi}{4},\frac{\pi}{2},\frac{3}{4}\pi,\pi,\frac{5}{4}\pi,\frac{3}{2}\pi,\frac{7}{4}\pi\right\}$ 组成共计 25 维的离散向量空间。

在设计奖励函数时，需要对无人车和目的地之间的距离，无人车之间的距离，无人车和边界之间的距离进行整体把控。为了鼓励无人车往目的地靠近，我们将无人车与其目的地之间距离平方的相反数作为基准奖励，当与目的地较远时，无人车对目的地有更高的趋向性，当与目的地较近时，无人车趋向于微调位置。当无人车到达目的地(与目的地距离 10 m 以内)后，将得到正 20 分的奖励，若无人车之间发生碰撞(两辆无人车距离 5 m 以内)或行驶至边界外(与边界距离 10 m 以内)，则会得到负 15 分的惩罚，具体奖励函数如下：

$$R=\begin{cases}-d_{des}^2+20 & d_{des}\le 10\\ -d_{des}^2-15 & d_{des}\le 5\\ -d_{des}^2-15 & d_{des}\le 10\\ -d_{tar}^2 & otherwise\end{cases} \tag{19}$$

4.1.4. 网络参数设置

在本次实验中，对照组 M^3-UCRL 算法由全连接的多层感知机组成的概率集成神经网络拟合环境动

力学模型。策略函数 $\pi(\cdot)$ 采用全连接神经网络表示，如表 2，表 3 所示。

Table 2. Environmental dynamics model network parameter settings
表 2. 环境动力学模型网络参数设置

网络参数	设置
集成网络数量	5
每个网络隐藏层数量	2
每个网络隐藏层节点数	64
每个网络隐藏层激活函数	*SiLU*
每个网络优化器	*Adam*
初始学习率	0.001

Table 3. Policy network parameter settings
表 3. 策略网络参数设置

网络参数	设置
隐藏层数量	4
隐藏层节点数	128
隐藏层激活函数	*SiLU*
优化器	*Adam*
初始学习率	0.001

Table 4. DDPM network parameter settings
表 4. DDPM 网络参数设置

网络参数	设置
加噪/去噪总时间步	1000
集成网络数量	10
隐藏层数量	2
隐藏层节点数	128
隐藏层激活函数	*SiLU*
网络优化器	*Adam*
初始学习率	0.001

在 DDPM-M^{3}RL 算法中，不直接拟合环境动力学模型，而是拟合加噪过程中的噪声 ε_θ，训练完成后，通过对采样得到的随机高斯噪声进行多步去噪，得到环境动力学模型。将 (s_t, a_t, μ_t) 作为条件进行引导采样后，可以得到 s_{t+1} 的数据。DDPM 网络参数如表 4 所示。

需要注意的是，想要进行条件引导采样，需在表 4 所示的网络结构前增加一层用于将 (s_t, a_t, μ_t) 与时间步 t 联合编码的编码层。

策略网络中增加了一个额外的辅助策略网络，用于减少策略网络的认知不确定性和偶然不确定性。具体如表 5，表 6 所示。

Table 5. DDPM policy network parameter settings
表 5. DDPM 策略网络参数设置

网络参数	设置
隐藏层数量	6
隐藏层节点数	128
隐藏层激活函数	*SiLU*
优化器	*Adam*
输出节点激活函数	无
初始学习率	0.001

Table 6. Auxiliary strategy network parameter settings
表 6. 辅助策略网络参数设置

网络参数	设置
隐藏层数量	2
隐藏层节点数	64
隐藏层激活函数	*SiLU*
优化器	*Adam*
输出节点激活函数	无
初始学习率	0.001

4.2. 仿真结果评估

4.2.1. 环境动力学模型建模精确性

图 5 展示了 M^3-UCRL 算法环境模型的训练损失，大约在 450 回合后，环境模型的训练损失趋于稳定，图中曲线为 5 个集成网络损失的均值；图 6 展示了 DDPM-M^{3}RL 算法中噪声模型的训练损失，在 350 回合后噪声模型损失趋于稳定。

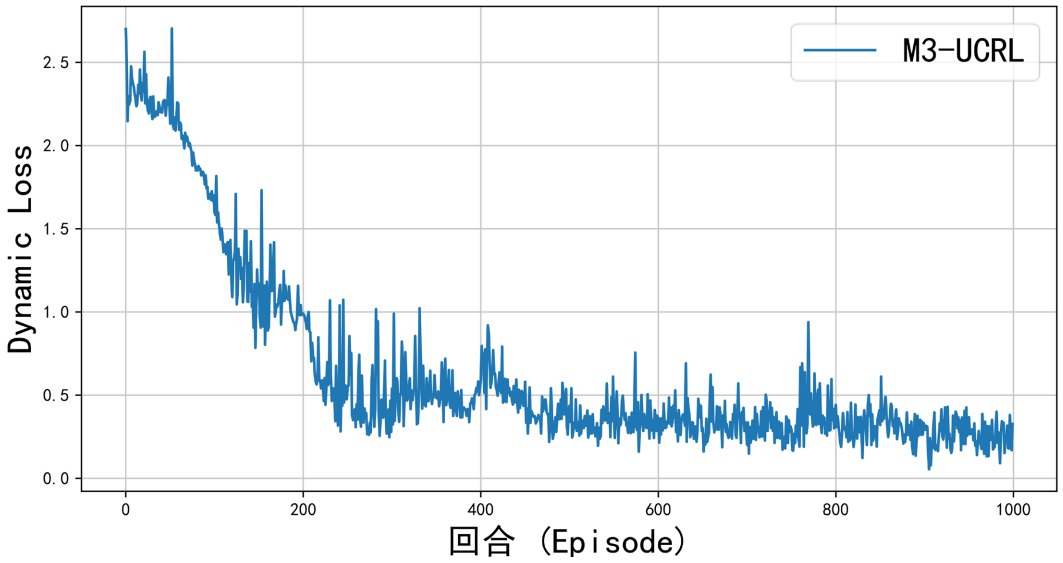

Figure 5. M^3-UCRL algorithm environment model training loss
图 5. M^3-UCRL 算法环境模型训练损失

由于两种算法的环境模型并不相同，无法直接通过训练损失比较算法学习到的环境模型对下一时刻状态预测的准确性，固选取均方差进行对比。我们抽取了 200 组真实环境的样本，并让环境模型预测下一时刻的状态，计算真实样本和环境模型预测的下一时刻状态之间的均方差，结果如图 7 所示。

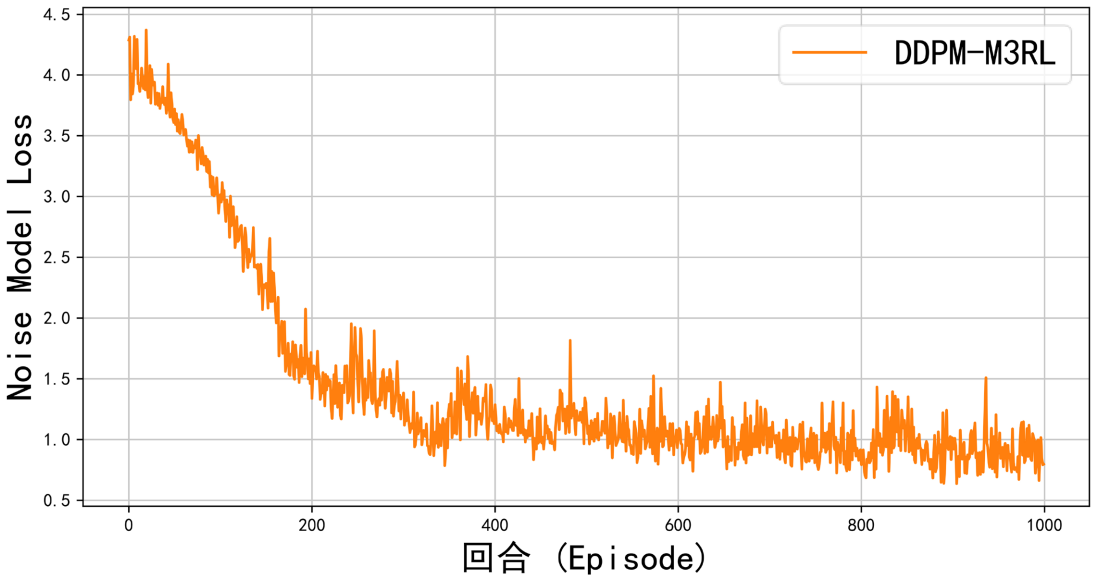

Figure 6. DDPM-M^{3}RL algorithm noise model training loss
图 6. DDPM-M^{3}RL 算法噪声模型训练损失

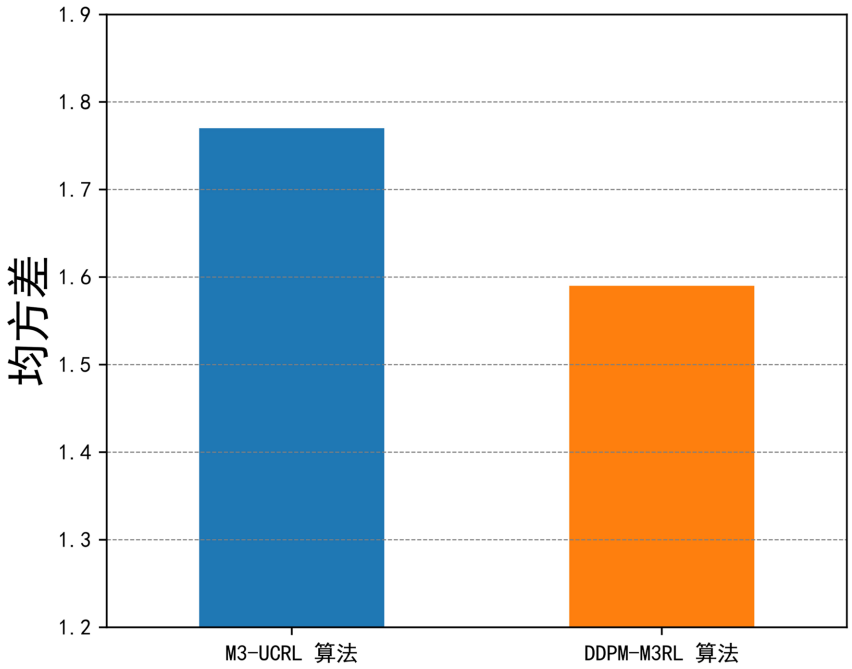

Figure 7. Comparison of mean square error between real samples and predicted samples
图 7. 真实样本与预测样本的均方差对比图

从图中可以看出，当两种算法的环境模型训练稳定后，M^3-UCRL 算法的预测数据与真实数据的均方差约为 1.77，而 DDPM-M^{3}RL 算法的预测数据与真实数据的均方差约为 1.59。不难看出，相较于 M^3-

UCRL 算法，DDPM-M^{3}RL 算法所训练出来的环境模型更稳定，更接近于真实环境，有更高的精确度。

4.2.2. 期望累积奖励

强化学习算法中，我们会对智能体做出的行为进行评价，当智能体做出我们期望的动作时，我们会给它奖励，相反，如果智能体做出了我们不希望它做的动作时，我们会给它惩罚。一般来说，当智能体与环境交互完成时，它所获得的奖励大小就是它是否按照我们的期望做出行为的直观体现。即，智能体获得的期望累积奖励越大，代表指导智能体行动的策略越好，算法的性能越强。

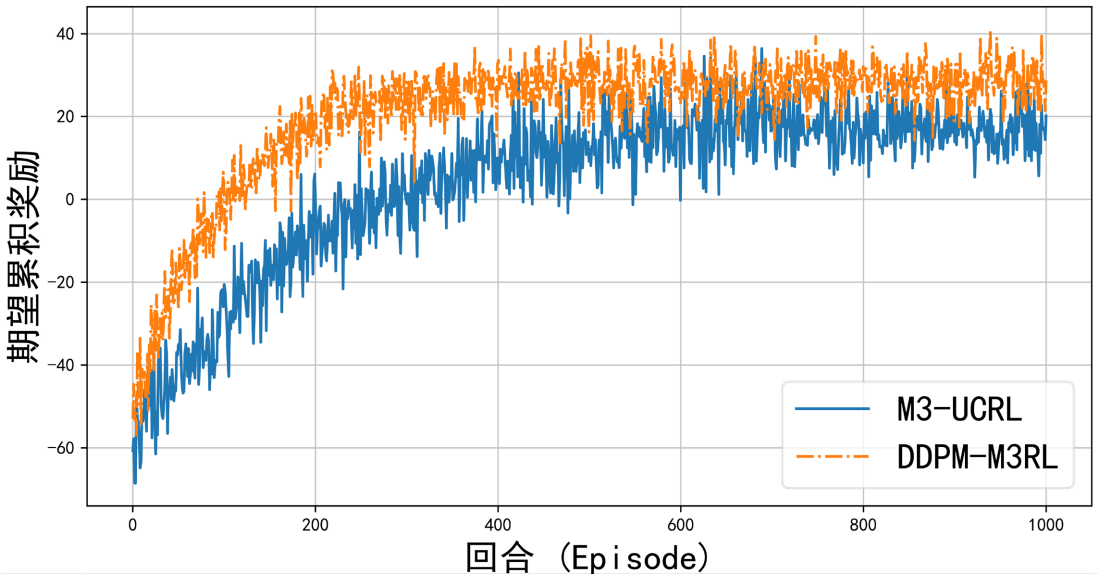

Figure 8. Comparison chart of expected cumulative rewards
图 8. 期望累积奖励对比图

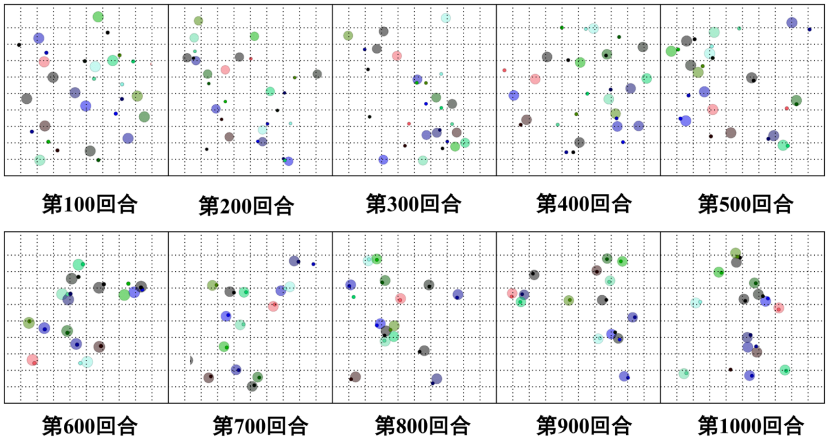

Figure 9. M3-UCRL algorithm location information diagram
图 9. M3-UCRL 算法位置信息图

M^3-UCRL 算法和 DDPM-M^{3}RL 算法均采用了 Dyna 算法框架[20]，这是一种在线强化学习算法，迭代地对环境模型和策略进行训练，最终得到最优策略。由图 8 可见，M^3-UCRL 算法获得的期望累积奖励在 600 回合左右趋于稳定，数值约为 19；而 DDPM-M^{3}RL 算法获得的期望累积奖励在 400 回合左右便趋于稳定，数值约为 25。不难看出 DDPM-M^{3}RL 算法收敛得更快且获得的奖励更高。另外，从图中可以看

出 DDPM-M^{3}RL 算法的奖励值的波动比 M^3-UCRL 算法更小，说明 DDPM-M^{3}RL 算法的稳定性更佳。

另外，为了跟踪 M^3-UCRL 算法和 DDPM-M^{3}RL 算法在策略学习中的学习效果，每 100 回合我们会记录下当前策略下的无人车的行动状态。具体位置信息如图 9，图 10 所示。

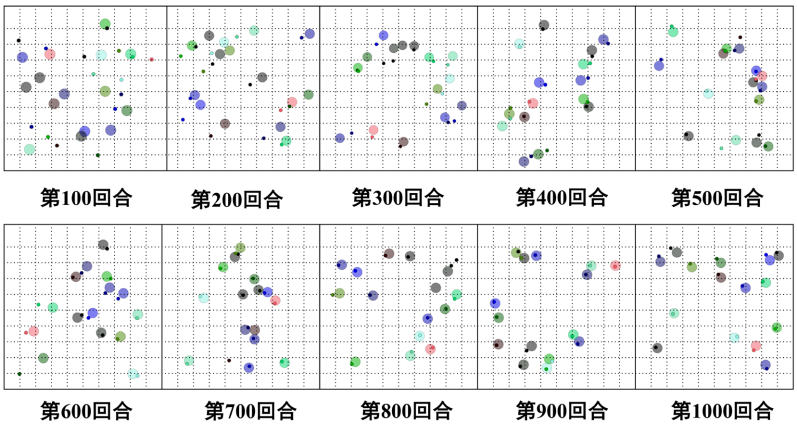

Figure 10. DDPM-M^{3}RL algorithm location information diagram
图 10. DDPM-M^{3}RL 算法位置信息图

综上所述，相较于 M^3-UCRL 算法，DDPM-M^{3}RL 算法的环境模型精确度和稳定性更好，策略收敛速度更快，期望奖励更高，整体性能更优。

5. 结束语

本文提出了一种基于去噪扩散模型的平均场多智能体强化学习算法，旨在解决基于模型的多智能体强化学习中环境模型建模难，开销大的挑战。在将多智能体系统通过 MFC 化简后，该算法利用了 DDPM 在数据生成方面的性能，对环境模型进行建模。智能体可以通过这种方式直接与环境模型进行交互，在节省了时间和经济成本的同时获得了大量优质数据用于后续的策略更新。我们在无人车集群合作导航应用场景下进行了仿真实验，结果表明 DDPM-M^{3}RL 算法拟合的环境模型拥有更高的准确性，在策略学习阶段收敛所需回合数更少，获得的累积期望奖励也更高，性能更加优越。

致　谢

感谢南京邮电大学波特兰学院对本文撰写的帮助，感谢南京邮电大学李大鹏教授对本文的指导，感谢上海交通大学房宇辰博士对本文创新点提出的建议。感谢张子胤和王肖同学对论文的帮助。

基金项目

国家自然科学基金资助项目(No.62371245)：基于图深度学习的无线网络优化关键技术研究。

参考文献

[1] Bin, W., Kerong, B., Yixue, H. and Mingjiu, Z. (2024) SQMCR: Stackelberg Q-Learning-Based Multi-Hop Cooperative Routing Algorithm for Underwater Wireless Sensor Networks. *IEEE Access*, **12**, 56179-56195. https://doi.org/10.1109/access.2024.3391386

[2] Shi, D., Li, L., Ohtsuki, T., Pan, M., Han, Z. and Poor, H.V. (2022) Make Smart Decisions Faster: Deciding D2D Resource Allocation via Stackelberg Game Guided Multi-Agent Deep Reinforcement Learning. *IEEE Transactions on Mobile Computing*, **21**, 4426-4438. https://doi.org/10.1109/tmc.2021.3085206

[3] Zhou, Z. and Xu, H. (2021) Decentralized Optimal Multi-Agent System Tracking Control Using Mean Field Games with

Heterogeneous Agent. 2021 *IEEE Conference on Control Technology and Applications* (*CCTA*), San Diego, 9-11 August 2021, 97-102. https://doi.org/10.1109/ccta48906.2021.9659203

[4] Hernandez-Leal, P., Kartal, B. and Taylor, M.E. (2019) A Survey and Critique of Multiagent Deep Reinforcement Learning. *Autonomous Agents and Multi-Agent Systems*, **33**, 750-797. https://doi.org/10.1007/s10458-019-09421-1

[5] Gu, H., Guo, X., Wei, X. and Xu, R. (2021) Mean-Field Controls with Q-Learning for Cooperative MARL: Convergence and Complexity Analysis. *SIAM Journal on Mathematics of Data Science*, **3**, 1168-1196. https://doi.org/10.1137/20m1360700

[6] Huang, M., Caines, P.E. and Malhame, R.P. (2007) Large-Population Cost-Coupled LQG Problems with Nonuniform Agents: Individual-Mass Behavior and Decentralized ε-Nash Equilibria. *IEEE Transactions on Automatic Control*, **52**, 1560-1571. https://doi.org/10.1109/tac.2007.904450

[7] Wang, T., Bao, X., Clavera, I., *et al*. (2019) Benchmarking Model-Based Reinforcement Learning. arXiv: 1907.02057.

[8] Pasztor, B., Bogunovic, I. and Krause, A. (2021) Efficient Model-Based Multi-Agent Mean-Field Reinforcement Learning. arXiv: 2107.04050.

[9] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., *et al*. (2015) Deep Unsupervised Learning Using Nonequilibrium Thermodynamics. *Proceedings of the* 32*nd International Conference on International Conference on Machine Learning*, Lille, 6-11 July 2015, 2256-2265.

[10] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S. and Poole, B. (2021) Score-Based Generative Modeling through Stochastic Differential Equations. arXiv: 2011.13456. https://doi.org/10.48550/arXiv.2011.13456

[11] Kingma, D.P. and Welling, M. (2013) Auto-Encoding Variational Bayes. arXiv: 1312.6114. https://doi.org/10.48550/arXiv.1312.6114

[12] Goodfellow, I., Abadie, J.P., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., *et al*. (2014) Generative Adversarial Nets. *Proceedings of the* 27*th International Conference on Neural Information Processing Systems*, Montreal, 8-13 December 2014, 2672-2680.

[13] Ho, J., Jain, A. and Abbeel, P. (2020) Denoising Diffusion Probabilistic Models. arXiv: 2006.11239. https://doi.org/10.48550/arXiv.2006.11239

[14] Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R. and Van Gool, L. (2022) Repaint: Inpainting Using Denoising Diffusion Probabilistic Models. 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, 18-24 June 2022, 11451-11461. https://doi.org/10.1109/cvpr52688.2022.01117

[15] Austin, J., Johnson, D.D., Ho, J., Tarlow, D. and Berg, R.V.D. (2021) Structured Denoising Diffusion Models in Discrete Statespaces. *Advances in Neural Information Processing Systems*, **34**, 17981-17993.

[16] Lee, J. and Han, S. (2021) Nuwave: A Diffusion Probabilistic Model for Neural Audio Upsampling. arXiv: 2104.02321. https://doi.org/10.48550/arXiv.2104.02321

[17] Kong, Z., Ping, W., Huang, J., Zhao, K. and Catanzaro, B. (2020) Diffwave: A Versatile Diffusion Model for Audio Synthesis. arXiv: 2009.09761. https://doi.org/10.48550/arXiv.2009.09761

[18] Dhariwal, P. and Nichol, A. (2021) Diffusion Models Beat GANs on Image Synthesis. *Advances in Neural Information Processing Systems*, **34**, 8780-8794.

[19] Ho, J. and Salimans, T. (2022) Classifier-Free Diffusion Guidance. arXiv: 2207.12598. https://doi.org/10.48550/arXiv.2207.12598

[20] Sutton, R.S. (1991) Dyna, an Integrated Architecture for Learning, Planning, and Reacting. *ACM SIGART Bulletin*, **2**, 160-163. https://doi.org/10.1145/122344.122377