近端策略优化的城市环境多智能体协作对抗方法

米广铭[1]，张辉[1,2]，张菁[1,2]，卓力[1,2]
（1.北京工业大学信息科学技术学院，北京 100124；2.北京工业大学计算智能与智能系统北京市重点实验室，北京 100124）

摘　要：城市环境由于其地理空间的复杂性及动态变化性，往往会令指挥系统变得低效且短视。针对该问题，提出了一种近端策略优化城市环境的多智能体协作对抗方法。首先，在建立完善的城市对抗环境的基础上，使用近端策略优化的演员-评论员网络算法进行求解；其次，针对多对一的评论网络采用嵌入方法来解决空间维度不同的异构智能体决策评价问题；再次，在近端策略优化的基础上，增加了自适应采样来辅助策略的更新；最后，对演员网络进行权重继承操作以帮助智能体迅速接管相应的任务。实验结果表明，相较于其他方法，所提方法的奖励回报提高了22.67%，收敛速度加快了8.14%，不仅可以满足城市环境下多个智能体协作对抗的决策，还能够兼容多异构智能体的协作对抗。

关键词：深度强化学习；多智能体；协作对抗；近端策略优化；城市环境
中图分类号：TP3
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025049

Multi-agent cooperative confrontation with proximal policy optimization in urban environments

MI Guangming[1], ZHANG Hui[1,2], ZHANG Jing[1,2], ZHUO Li[1,2]
1. School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China
2. Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing University of Technology, Beijing 100124, China

Abstract: To address the issue that urban environments often make command systems inefficient and inflexible due to their geospatial complexity and dynamic changes, a multi-agent cooperative confrontation method with proximal policy optimization for urban environments was proposed. First, on the basis of establishing a comprehensive urban confrontation environment, the AC (actor-critic) network with proximal policy optimization was used to solve the problem. Then, aiming at the multi-to-one critic network, an embedding method was adopted to address the issue of evaluating the decision-making of heterogeneous agents with different spatial dimensions. Furthermore, adaptive sampling was added to assist in the updating of proximal policy optimization. Finally, the weights of the actor network were inherited to help agents quickly take over the corresponding tasks. Experimental results show that the proposed method improves 22.67% reward and 8.14% convergence rate compared to other methods, which not only meets the decision-making of multiple agents' cooperative confrontation in urban environments, but also is compatible with the cooperative confrontation of multiple heterogeneous agents.

Keywords: deep reinforcement learning, multi-agent, cooperative confrontation, proximal policy optimization, urban environment

收稿日期：2024–10–29；修回日期：2025–02–20
通信作者：张菁，zhj@bjut.edu.cn
基金项目：北京市自然科学基金资助项目（No.L247025）
Foundation Item：The Beijing Natural Science Foundation (No.L247025)

0　引言

现代复杂环境下的决策支持系统作为关键决策流程的核心组成部分，其功能涵盖方案生成、多维度评估及优化选择。决策者需要综合分析实时态势信息、资源约束条件以及潜在变量因素，在多重限制条件下实现动态平衡[1-2]。随着应用场景的不断演进，系统面临的动态多变环境呈现出显著增加的复杂性特征，这对决策支持方法和技术体系提出了新的要求[3]。同时，智能化系统在各个领域内快速发展，应用范围日益增加[4-5]。因此，探索智能化信息处理、分析方法和技术，提升指挥系统的决策能力和效能，已成为当前亟待突破的研究方向[6]。

传统的指挥系统决策工作已经取得了一定进展。例如，文献[7]研究了基于线性加权任务效能函数在多因素任务分配问题中的应用求解；文献[8]考虑到不同武器和目标适配性，建立了一种基于混沌初始化的动态高斯变异虫群算法，能够求解大规模的武器目标分配问题；文献[9]提出了一种自适应的大领域搜索算法用以求解传感器−武器−目标分配问题，并对求解过程进行了线性近似，获得了问题模型的最广范围。尽管现有算法计算效率高、结果稳定、适合有明确规则的问题，并且对数据需求较少，但是无法处理长期决策问题，难以应对高维状态空间且缺乏自学习能力。

深度强化学习（DRL, deep reinforcement learning）由于结合了深度学习的感知能力和强化学习的决策技术，使机器可以在复杂的环境中进行决策和学习[10-11]。近些年，学者们开始探索更多样化的DRL算法和应用。例如，文献[12]首次提出了深度Q网络，是一种能够直接从原始像素输入学习控制策略的算法；文献[13]将DRL的应用从离散动作空间扩展到连续动作空间，提出了深度确定性策略梯度，为机器人控制的研究奠定了基础；为了解决无模型强化学习样本效率低下的问题，文献[14]通过将模型嵌入概率强化学习框架中，提出了一种基于离线策略模型的DRL算法，证明了短期虚拟推理的可靠性。文献[15]研究了近端策略优化（PPO, proximal policy optimization）技术在合作多智能体领域中的应用，提出了多智能体近端策略优化（MAPPO, multi-agent proximal policy optimization），并在多项基础实验中取得了良好结果。因此，在资源可变、背景环境复杂的情况下，将DRL技术用于指挥系统具有划时代的意义和作用。相比传统方法，DRL能够在复杂动态环境中自我调整和优化策略，更适合解决长期和不确定性问题。如文献[16]针对多种异构网络干扰场景，提出了一种基于多智能体强化学习的小区范围扩展偏置动态优化算法以应对无线网络的高吞吐量要求；针对有/无人机协同平台，文献[17]提出了一种基于DRL的协同目标搜索及轨迹规划算法；文献[18]基于动态环境，使用DRL技术解决目标分配问题，有效地实现了大规模目标分配方案的动态生成；文献[19]针对海上舰船防空反导导弹目标分配问题，提出了一种融合注意力机制的DRL算法，使舰船能够完成高效的导弹目标分配。此外还有许多科研人员在其他多种环境下使用DRL技术进行指挥决策。

现有DRL研究鲜有提及以城市环境为背景的智能体决策问题。城市环境由于其自身的复杂性、动态变化性以及空间限制性[20]，为多智能体的协作对抗提供了独特的挑战和应用环境，智能体不仅需要处理大量异质信息，还要在与其他同/异构智能体的互动中做出迅速且准确的决策。

同构智能体由于具备相似的结构和能力，可以共享相同的算法和策略，因而训练相对简单。而异构智能体状态和动作空间均比较复杂，往往会造成训练的不稳定以及不收敛。对于异构智能体的训练学习，文献[21]提出了异构智能体强化学习算法及异构智能体镜像学习框架，并通过实验证明了其具有更高的有效性和稳定性；文献[22]则讨论了异构智能体的博弈问题，并提出了一种混合Q学习网络架构；为了提高智能体的高精度车辆跟踪性能，文献[23]通过将异构智能体转换到同构智能体，提出了一种基于图的分布式学习控制法，使其能够克服异构智能体的影响。上述方案尽管在处理异构智能体问题上展现了显著的效果，但其对算力的需求庞大，同时资源占用也相对较高，且大多独立于现有的基础网络架构进行，环境针对性较强。这种高资源消耗和算力需求对实际应用带来了较大的挑战，因此亟须一种能够融入基础网络架构且低算力要求的解决方案，从而更高效地兼顾同/异构智能体的复杂性，在资源受限的环境中实现可持续的应用发展。

为此，本文面向城市环境对异构智能体问题进行求解，提出了一种近端策略优化的城市环境多智

能体协作对抗方法，致力于探索智能体在城市环境中的协作对抗过程及其在指挥通信领域的应用价值。主要贡献如下。

1) 面向城市作战环境，设计了包含完整环境参数及约束的多智能体，提供了一个较全面的智能体对抗环境。对环境中智能体的各项参数，以及环境基础参数进行详细设计，以满足智能体移动、攻击等基本指令，同时增加建筑物占领情况、基于时间窗口的环境变化等功能，提高了拟真性。

2) 提出一种近端策略优化演员-评论员（AC, actor-critic）网络算法，使用多对一的评论员网络结构完成多个评论员对单个智能体的行为评论过程，并使用嵌入方法解决异构智能体的空间差异问题。帮助评论网络更深入地理解智能体之间的复杂交互关系，从而更合理地评估智能体的行为价值。

3) 联合在线策略（on-policy）模型与离线策略（off-policy）模型，设计了自适应经验采样方法来优化近端策略优化AC网络模型的经验回放，将优秀的智能体的演员策略权重参数继承给其他智能体的演员，辅助智能体达到更优的训练效果，从而使智能体能够快速接管相应任务，达到缩短训练回合的目的。

1　系统模型

1.1　系统任务描述

近端策略优化的城市环境多智能体协作对抗系统旨在复杂多变城区作战环境下，协同多个智能体，赢得目标对抗任务。所提系统框架如图1所示，包括评论网络、演员网络、近端策略优化及城市环境四部分。评论网络包含多个独立的评论员，评论员接收多个智能体的动作序列$\boldsymbol{A}=\{a_t^1,a_t^2,\cdots,a_t^{N-1},a_t^N\}$以及与之对应智能体的状态序列$\boldsymbol{S}=\{s_t^1,s_t^2,\cdots,s_t^{N-1},s_t^N\}$作为输入，输出对于当前演员的价值评估$Q(s_t,a_t)$，并通过近端策略优化算法将经验价值反馈至各智能体对应的演员网络；演员网络中演员1~N对经验价值进行处理后，将个体动作a_t^n分别输出给评论网络及城市环境，城市环境响应接收到的动作，输出状态s_t^n至演员网络，并将这些状态所导致的奖励R_t反馈给评论网络，形成一个闭环的系统。

宏观意义上，由AC网络及其对应的策略优化算法所引导的城市环境组成了环境模型，多个存在协作对抗的同/异构智能体组成了智能体模型。智能体模型对环境观测，遵照环境约束做出相应决策，从而完成智能体各自的目标，这个过程同样形成了一个闭环系统。

1.2　智能体模型设置

智能体模型中的多个智能体被设置为两组，分别表述为智能体组α和智能体组β，定义如下

$$\alpha=\left\{\alpha_1^{\text{type}},\alpha_2^{\text{type}},\cdots,\alpha_{m-1}^{\text{type}},\alpha_m^{\text{type}}\right\} \tag{1}$$

$$\beta=\left\{\beta_1^{\text{type}},\beta_2^{\text{type}},\cdots,\beta_{n-1}^{\text{type}},\beta_n^{\text{type}}\right\} \tag{2}$$

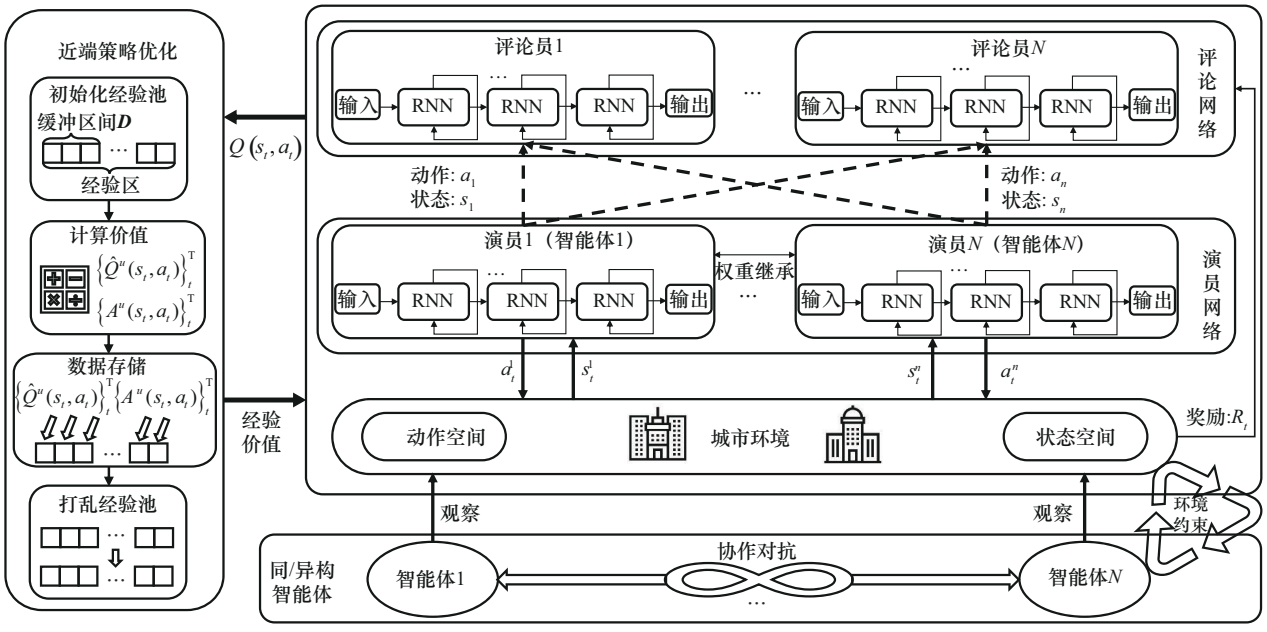


图1　近端策略优化的城市环境多智能体协作对抗系统框架

其中，type 表示单位型号，type=0 表示普通单位，type=1 表示机动单位；m 与 n 分别表示智能体组 α、β 中智能体的编号，本文定义 m=n。

α 的核心行动为占领目标单位，行动策略自主生成，其会根据对方位置、环境情况等参数生成相应决策并行动；β 的核心行动为摧毁 α 的全部成员，保护目标单位不被占领，其依据既定策略 π_ε 行动。既定策略 π_ε 即基于价值-距离的贪婪策略：当 β 成员 n' 行动时，其在攻击范围内遇到一个或多个 α 成员后，攻击价值最高的 α 成员 m'，否则继续巡逻。具体定义如下

$$\forall \begin{cases} n' \in [0,n] \\ m' \in [0,m] \end{cases}, \pi_\varepsilon = \begin{cases} \arg\max\limits_{m'} \left\{ \text{value}_{m'} \right\}, \exists d_{n'm'} \leqslant 2 \\ -1, \text{其他} \end{cases} \tag{3}$$

其中，$\text{value}_{m'}$ 为 α 成员 m' 的价值；$d_{n'm'}$ 为 α 与 β 成员在本文设立的城市环境下的欧几里得距离，计算式为

$$d_{n'm'} = \sqrt{(x_{m'} - x_{n'})^2 + (y_{m'} - y_{n'})^2} \tag{4}$$

其中，x 和 y 分别为智能体成员的横坐标和纵坐标，在真实世界坐标下，由大地主题解算方法[24]求解横纵坐标对应的经纬度。

由于智能体的type不同，因此其性能参数也有所区别，故需对相关智能体模型参数进行定义。对于任意智能体 i 在抵御智能体 j 进攻时的防御能力 $\varphi_{ij}(h)$ 定义如下

$$\varphi_{ij}\left(h_{\mathrm{i}}\right) = \begin{cases} \boldsymbol{K}_{ij}^{\varphi}\varphi', h_i \geqslant \hat{h} \\ \boldsymbol{K}_{ij}^{\varphi}\varphi' + \left(\varphi_{\max} - \varphi'\right)\dfrac{\hat{h} - h_i}{\hat{h}}, h_i < \hat{h} \end{cases} \tag{5}$$

其中，$\varphi_{\max}$ 为最大防御力，φ' 为标准参量，用来定义基准防御力；h_i 为智能体 i 的生命值，$\hat{h}$ 为生命阈值，当生命值 h_i 高于阈值时，智能体 i 的防御力为一常量，当生命值 h_i 低于阈值时，智能体 i 的防御力线性升高，逼近 $\varphi_{\max}$；$\boldsymbol{K}_{ij}^{\varphi}$ 为防御系数矩阵 $\boldsymbol{K}^{\varphi}$ 中智能体 i 在抵御智能体 j 进攻时的防御系数，$\boldsymbol{K}^{\varphi}$ 是大小为 $m \times n$ 的实数矩阵，写为

$$\boldsymbol{K}^{\varphi} = \begin{bmatrix} k_{11}^{\varphi} & k_{12}^{\varphi} & \cdots & k_{1n}^{\varphi} \\ k_{21}^{\varphi} & k_{22}^{\varphi} & \cdots & k_{2n}^{\varphi} \\ \vdots & \vdots & & \vdots \\ k_{m1}^{\varphi} & k_{m2}^{\varphi} & \cdots & k_{mn}^{\varphi} \end{bmatrix} \tag{6}$$

其中，k_{mn}^{φ} 表示 α 中的成员 m 抵御 β 中的成员 n 进攻时的防御系数。为简化取值流程，规定同一type的智能体成员在各个智能体组的位置相同（下文将默认该条件），因此 $k_{mn}^{\varphi} = k_{nm}^{\varphi}$；若不同type的智能体成员在各个智能体组的位置不同，则需要对防御系数矩阵 $\boldsymbol{K}^{\varphi}$ 进行翻折取反操作并建立新的矩阵集。

任意智能体 i 打击智能体 j 时的攻击能力 δ_{ij} 为

$$\delta_{ij} = \delta' \boldsymbol{K}_{ij}^{\delta} \tag{7}$$

其中，δ' 为标准参量，用来定义基准攻击能力；$\boldsymbol{K}_{ij}^{\delta}$ 为攻击系数矩阵 $\boldsymbol{K}^{\delta}$ 中智能体 i 对智能体 j 的攻击能力。在此状态下，将环境中的掩体作为临时智能体成员分别加入智能体组 α 与 β，$\boldsymbol{K}^{\delta}$ 是一大小为 $(m+1)\times(n+1)$ 的实数矩阵，表示为

$$\boldsymbol{K}^{\delta} = \begin{bmatrix} k_{11}^{\delta} & k_{12}^{\delta} & \cdots & k_{1n}^{\delta} & k_{1(n+1)}^{\delta} \\ k_{21}^{\delta} & k_{22}^{\delta} & \cdots & k_{2n}^{\delta} & k_{2(n+1)}^{\delta} \\ \vdots & \vdots & & \vdots & \vdots \\ k_{m1}^{\delta} & k_{m2}^{\delta} & \cdots & k_{mn}^{\delta} & k_{m(n+1)}^{\delta} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{8}$$

其中，k_{mn}^{δ} 为 α 中的成员 m 对 β 中的成员 n 的攻击系数，$k_{mn}^{\delta} = k_{nm}^{\delta}$。

令 $p_{\max}$ 为最大命中率，f 为超参数，用来控制递减速率，则基准命中概率 $\hat{p}$ 为

$$\hat{p} = p_{\max} \mathrm{e}^{-f d_{ij}} \tag{9}$$

任意智能体 i 打击智能体 j 时的攻击命中概率 p_{ij} 为

$$p_{ij}\left(d_{ij}, r\right) = \hat{p}\left(d_{ij}\right) r, r \in [0.8,1] \tag{10}$$

其中，r 表示随机因素，用来模拟真实世界下意外情况的发生。

综上，任意智能体 i 对智能体 j 在非驻扎情况下的杀伤力 ξ_{ij} 为

$$\xi_{ij} = \delta_{(i-1)(j-1)} p_{ij} - \gamma \varphi_{ji} \tag{11}$$

其中，γ 为权重系数。

智能体 i 的生命迭代过程为

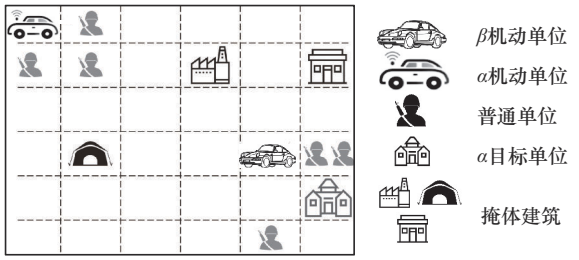
$$h_j = h_j - \gamma'\left(\delta_{(i-1)(j-1)} p_{ij} - \gamma \varphi_{ji}\right) = h_j - \gamma' \xi_{ij} \tag{12}$$

其中，γ' 为权重系数。

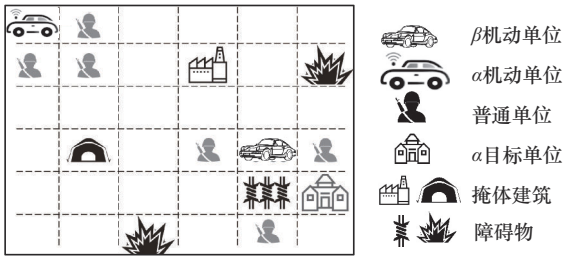
1.3　环境模型设置

在真实情况下，环境会随事态发展而发生改变，指挥系统的决策也会随之发生改变。例如，对于视野遮挡、位置信息模糊等复杂因素，可以将其统一视作场景随时间状态推移的未知变化。为此，环境模型被设置为2种不同时间状态，用来模拟真实的对抗情况。城市环境如图2所示，图2(a)是多智能体最开始捕获的位置地图，在智能体推断行为到第4步的时候，改变地图为图2(b)。该过程可以

模拟出现实世界中城市战场环境不断变化的特点，通过增加智能体修正环境感知，探索未知状态及既定策略的过程，如智能体在该环境下的观测范围为2格，则占领或攻击范围为1格。



(a) 对抗初期



(b) 对抗后期

图2　城市环境

在对抗初期，如图2(a)所示，城市环境下存在大量的掩体，其可以为普通单位提供一定的保护，但过多的驻扎单位导致掩体的保护降低，因此城市环境中掩体o所提供的保护$\varphi_o^p(l)$为

$$\varphi_o^p(l)=\frac{\varphi_o'}{1+cl^2} \tag{13}$$

其中，φ_o'为标准参量，用来定义基准防御能力；c为权重系数，控制下降速率；l为掩体内智能体个数，当智能体个数增多时，掩体o所提供的保护趋于饱和。

因此，驻扎于掩体内的智能体i在抵御智能体j进攻时的防御能力φ_{ij}^p为

$$\varphi_{ij}^p=\varphi_{ij}\left(h_i\right)+r\varphi_o^p(l) \tag{14}$$

但掩体并非坚不可摧，当掩体被摧毁时，驻扎在其中的普通单位也会同时消失，因此对智能体来说，长时间驻扎于掩体并不是一个很好的选择。

城市环境会随着时间的推移发生改变，在对抗初期，如图2(b)所示，部分掩体被摧毁形成障碍物，使原本可通行的道路受阻，进而导致智能体先前的策略失效。因此，面对多智能体与城市环境交互过程的复杂多变性，设计适应性强的AC网络算法对提升对抗赢率、发挥指挥系统决策效能尤为重要。

2　近端策略优化的AC网络

为了获得城市环境下智能体协作对抗问题的最优解，本文采用MAPPO框架进行求解，设计了一种基于近端策略优化的AC网络。该方案包含N个智能体，每个智能体有着自己一一对应的演员π与评论员v，其均采用循环神经网络（RNN, recurrent neural network）[25]作为训练网络。演员作为智能体大脑，指导其做出相应决策，其策略参数记为θ，旧策略参数记为θ'。各个演员在算法执行阶段会根据对抗情况有条件地选择复刻其他智能体演员，即权重继承；评论员则根据智能体决策做出质量评价，网络参数记为ρ，旧网络参数记为ρ'。每个智能体对应的评论员在评价自己所对应的智能体的同时，也评价其他多个智能体，形成多对一的评论网络。

近端策略优化的AC网络算法模型如图3所示，该算法模型包含2个阶段，即集中式训练与分布式执行。在集中式训练阶段，智能体首先与环境进行交互，将观测到的局部状态信息o传输至本地演员。然后，本地演员会根据该状态输出一个概率分布$p(a|o)$，并从动作空间中抽样选择一个动作a。接下来，智能体在城市环境下做出相应动作，从而使环境状态更新并获得下一个时间步t的观测信息与奖励r。之后，智能体将动作信息、状态信息等参数上传至评论员，评论员据此计算每个智能体的优势函数$\hat{A}$。每个时间步结束后，将交互流程中的全部参数均存入经验缓存D中，包括状态、动作、奖励、下一状态等信息。最后，基于优势函数，演员通过优化目标函数对演员网络进行更新，评论员网络则根据环境反馈的真实奖励优化其预测的状态值，使其更准确地评估当前状态的预期回报。

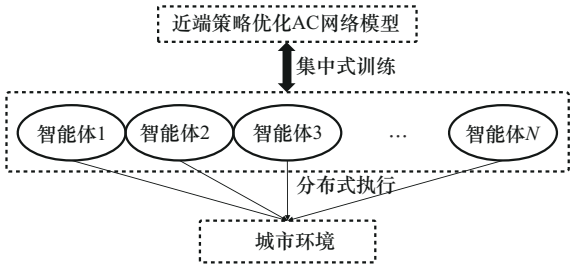


图3　近端策略优化的AC网络算法模型

在分布式执行阶段，每个智能体基于自己局部的观测信息做出决策，但由于权重继承的存在，智能体之间仍会产生决策级的关联，同时智能体还会根据其他智能体的状态优化其决策，保证了智能体间的一致性和协作能力，并非完全独立。因此在该算法模型的执行阶段，演员网络仍然独立运行，而不再需要评论网络的参与，确保了智能体可以在实际环境中不依赖全局信息进行高效决策，使其能够实现基于局部信息完成全局目标。

近端策略优化的AC网络算法如算法1所示。

算法1　近端策略优化的AC网络算法

1) 初始化评论员v、演员π、策略参数θ、旧策略参数θ'、评论网络参数ρ、旧网络参数ρ'及学习率和其他超参数

2) for episode=1, 2, 3, ⋯, k

3) 初始化全部参数

4) for t=0, 1, 2, 3, ⋯, T

5) for agent(n) in N

6) 智能体根据$\pi(o_t|\theta)$生成动作概率分布$p(a|o_t)$，并做出相应动作a_t

7) 智能体的评论员根据$v(o_t, a_t|\rho)$获得对应奖励r_t及全局状态s_{t+1}

8) end for

9) 生成智能体与城市环境的交互迹τ

$$\tau_n = \left\{o_t, r_t, a_t, o_{t+1}\right\}_{t=0}^{T}$$

10) end for

11) 计算智能体n在全部时间步的优势函数

$$\hat{A}_t\left(s_t, a_t\right)$$

12) $\left(o_t, a_t, r_t, o_{t+1}, \hat{A}_t\left(s_t, a_t\right)\right) \rightarrow \boldsymbol{D}$

13) for l=1, 2, 3, ⋯, L

14) 对每个智能体根据自适应策略从经验缓存$\boldsymbol{D}$中截取一段长度为b的小批次经验集

15) for c in b

16) 计算loss(θ)，并对θ进行梯度上升的更新

17) 计算loss(ρ)，并对ρ进行梯度上升的更新

18) end for

19) end for

20) 更新策略参数$\theta' \leftarrow \theta$

21) 更新评论网络参数$\rho' \leftarrow \rho$

22) end for

算法1中，步骤1)～步骤3)为初始化阶段，用来设置训练回合数；步骤4)～步骤10)收集智能体通过当前策略与城市环境的交互迹τ；步骤11)～步骤12)计算$\hat{A}$并将相关信息存入经验缓存$\boldsymbol{D}$中；步骤13)～步骤22)为更新演员的策略参数及评论员的网络参数的迭代过程。

由算法1流程可以看出，相较于传统的MAPPO算法，本文提出的近端策略优化的AC网络算法针对异构智能体优化设计了评论网络，同时在经验回放过程中采用了自适应策略，在分布式执行阶段构建了依据对抗条件选择性复刻其他演员的智能体权重继承方案。下面将从这3个部分内容展开进一步的论述。

2.1　多对一评论网络

评论网络负责评估多智能体环境中的状态或状态-动作对的价值，帮助策略网络进行优化。每一个演员（智能体）都拥有对应的评论员，同时该评论员还会对其他智能体的行为做出评价，这使评论网络能够在多智能体环境中充分利用全局信息，评估各智能体的行为如何影响整体系统的性能。这些评论员基于不同的策略和观点对智能体的行为进行评价，从而帮助智能体改进其决策。进一步地，在复杂的多智能体博弈环境中，评论出现冲突较为常见，本文采用优先级判别的思路根据评论员的历史评估的准确性以及当前状态的相关性对每个评论员的反馈进行加权，权重较高的评论员的反馈会对智能体的行为学习产生较大的影响，在产生冲突时也往往采用这类高优先级的评论员策略。这种方式可以在合作任务中很好地引导智能体学习协调行动，实现全局目标。此外，在进行实际应用的部署时，评论员数量往往是固定的，且网络结构可以通过剪枝和参数共享等技术进一步优化，因此实际运行时的计算负担可以在人为控制内，避免造成资源浪费。

2.1.1　RNN评论网络

所提算法使用RNN作为评论网络，其结构如图4所示。RNN是一种用于处理序列数据的神经网络，具有记忆能力，能够保留之前输入的信息，并用于当前时刻的计算，具体表示如下

$$h_t = \tanh\left(W_h h_{t-1} + W_x x_t + b_h\right) \tag{15}$$

其中，h_t是时间步t的隐藏状态，W_h是隐藏层的权

重，W_x是输入层的权重，x_t是时间步t的输入，b_h是偏置项，tanh()是激活函数。

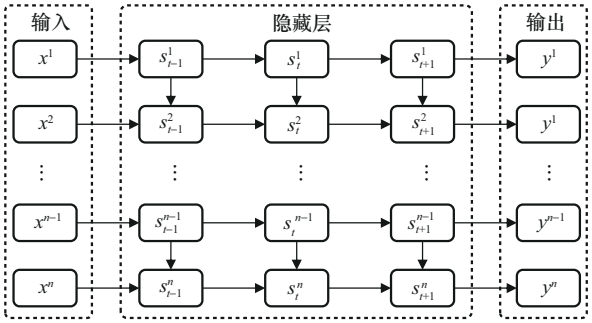


图4　RNN结构

RNN的输出为

$$y_t = \text{Sigmoid}\left(W_o h_t + b_o\right) \tag{16}$$

其中，y_t是时间步t输出，W_o是输出层权重，b_o是输出层偏置项，Sigmoid()是输出层的激活函数。

与之对应，每个智能体通常只能观测到环境的局部状态，不能完全了解整个系统的全局状态。这种情况下，智能体无法基于当前时刻的单独观测直接做出准确的决策。而RNN的隐藏状态能够在时间步之间累积信息，结合过去的观测信息，通过记忆和信息整合，构建对全局状态的更好估计。

同时，智能体的动作往往与先前的状态和行为紧密相关，因此存在较强的时序依赖。而状态价值不仅仅依赖当前时刻的观测，还依赖之前的状态变化。因此RNN通过其循环结构，能够将过去的隐藏状态h_{t-1}传递到当前时刻t，使评论网络在计算当前状态价值时可以考虑之前多个时间步的状态和行为，捕捉到这种时序相关性。

2.1.2　异构智能体问题优化设计

在处理异构问题时，由于异构智能体拥有不同的状态空间、动作空间、奖励函数等特性，因此为训练带来了更多的挑战。在集中式训练领域中，目前大多数工作均是通过修正奖励函数或增加条件约束来限制异构智能体的行为模式[26]，但这种方法往往会造成迭代缓慢、计算资源浪费等问题。

针对上述问题，本文将嵌入方法加入评论网络，嵌入方法可以通过并行计算进一步加速，且在高维空间情况下的加速效果更优。异构智能体可能具有不同的感知能力和动作空间，将这些离散的感知和动作通过全连接神经网络嵌入一个共享的连续向量空间中，如图5所示。这种方法能够使智能体之间的状态、动作进行有效的对比和协作。每个智能体有对应的状态空间S和动作空间A。对于智能体i，其状态为$s_i \in S_i$，动作为$a_i \in A_i$，智能体需要通过其策略π_θ来选择合适的动作以最大化其回报。

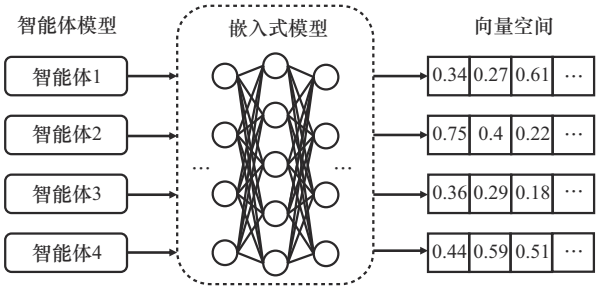


图5　嵌入方法示意

在收集不同智能体的交互数据（包括状态、动作等信息）后，将每个智能体的状态空间S_i嵌入相同的向量空间中，映射为相同的向量，表示为

$$z_i = \eta_i\left(s_i\right), z_i \in \mathbb{R}^d \tag{17}$$

其中，η_i为嵌入函数，z_i为嵌入后的状态向量，d为向量维度。

同样地，对于每个智能体的动作空间，嵌入后的动作向量为

$$u_i = \eta_i\left(a_i\right), u_i \in \mathbb{R}^d \tag{18}$$

因此，每个智能体的策略可以基于嵌入后的状态z_i进行表示，即通过嵌入后的状态选择嵌入后的动作。这样，每个智能体的策略可以表示为

$$\pi_i\left(s_i\right) = \arg\max_{a_i} Q\left(z_i, u_i\right) \tag{19}$$

其中，$Q(z_i,u_i)$是基于嵌入空间中的状态和动作的价值函数，它评估当前状态z_i和动作u_i的潜在收益。

同样地，嵌入相同向量空间后的特征可以作为RNN的输入，更新隐藏状态h_t为

$$h_t = \text{RNN}\left(h_{t-1}, \left\{\left[z_i, u_i\right]\right\}_0^N\right) \tag{20}$$

通过将异构智能体的状态、动作等信息嵌入连续向量空间中，能够使评论网络更好地理解和处理智能体之间的复杂交互关系，更合理地对智能体行为做出价值估计。

2.2　近端策略优化算法

近端策略优化算法[27]是一种策略优化类的强化学习算法，其核心思想是在更新策略时，限制策略的变化幅度，提升算法迭代过程中的模型稳定性。

在强化学习领域，策略梯度方法的目标是直接优化策略π，以最大化累积期望奖励$\mathbb{E}_{\pi}\left[\sum_t r_t\right]$。策略梯度的一般形式如下

$$\mathbb{E}_{\tau\sim\pi}\left[\sum_t \nabla_\theta \log \pi_\theta \hat{A}_t\right] \tag{21}$$

其中，$\log\pi_\theta$是策略的对数概率，$\hat{A}_t$的计算式为

$$\hat{A}_t = \sum_{i=1}^{n}\left(Q(z_i,u_i) - V(s_t)\right) \tag{22}$$

其中，$V(s_t)$为全局状态值函数，其表示在状态s_t下，所有智能体按照当前策略执行的预期回报。

策略更新容易发生过大的更新步长，导致新策略与旧策略之间的差异过大，进而引发策略崩溃或训练不稳定的问题。因此通过定义剪切目标函数来限制新策略π_θ和旧策略$\pi_{\theta'}$的比率rate来控制策略的变化幅度。

$$\text{rate}(\theta) = \frac{\pi_\theta}{\pi_{\theta'}} \tag{23}$$

剪切目标函数为

$$L^{\text{CLIP}} = \mathbb{E}_t\left[\min\left(\text{rate}(\theta)\hat{A}_t, C\right)\right] \tag{24}$$

其中，$\text{rate}(\theta)\hat{A}$表示常规的策略梯度目标；$C$为裁剪函数，表示对rate($\theta$)进行裁剪，以限制其在[1–$\varepsilon$, 1+$\varepsilon$]范围内，防止更新幅度过大，具体定义为

$$C = \text{CLIP}\left(\text{rate}(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_t \tag{25}$$

为了更好地辅助策略优化更新，将最新生成的on-policy数据保留，主要用于策略更新的依据。首先初始化检验缓存，再将过去经验存储在一个经验缓存中并进行打乱操作，允许在一定概率下从经验缓存中进行采样，以辅助策略更新，加快学习速度。

一种采样方法是根据检验缓存中数据的生成时间进行加权，优先选择较新的数据进行采样，使用一种指数衰减机制对经验进行采样，较旧的经验将赋予较低的权重。从检验缓存重采样的第i条经验的概率P_i^t为

$$P_i^t = \frac{\mathrm{e}^{-\alpha\left(t_c - t_i\right)}}{\sum_{j=1}^{N}\mathrm{e}^{-\alpha\left(t_c - t_j\right)}} \tag{26}$$

其中，t_c是当前时间步，t_i是第i条经验的生成时间步，α是控制采样偏向的超参数，N是检验缓存中总的经验条数。

另一种采样方法是引入优先级经验回放。此机制根据经验的时序差分误差（TD-Error, temporal difference error）来决定经验的重要性，优先采样误差较大的经验以加速学习，从检验缓存重采样的第i条经验的概率P_i^s为

$$P_i^s = \frac{\left|\vartheta_i\right|^\beta}{\sum_{j=1}^{N}\left|\vartheta_i\right|^\beta} \tag{27}$$

其中，β是控制优先级采样程度的超参数，ϑ_i是第i条经验的TD-Error或优势函数值，TD-Error表示为

$$\vartheta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{28}$$

其中，ϑ_t是在时间步t的TD-Error，r_{t+1}是在时间步t得到的即时奖励，γ是折扣因子，$V(s)$是不同状态下的预期回报。

将2种采样方法融合后，若经验较新，则它更可能被选中；若经验的TD-Error大，同样也更容易被选中。第i条经验的自适应采样概率P_i为

$$P_i = \frac{\lambda_1 P_i^t \lambda_2 P_i^s}{\sum_{j=1}^{N}\lambda_1 P_j^t \lambda_2 P_j^s} \tag{29}$$

其中，λ_1用来控制时间加权在采样中的重要性，λ_2用来控制优先级在采样中的重要性。

综上所述，本文采用了一种基于重要性和时序差分误差的自适应采样优化策略，在实际应用时可以通过调整样本大小来控制计算资源的消耗，从而减少冗余计算。对策略进行更新时采用on-policy的近端策略优化算法，同时使用off-policy的经验缓存进行辅助，从而完成更高效的策略更新。

2.3　权重继承的演员网络

与评论网络所对应的演员网络同样采用了RNN作为网络模型，RNN允许演员网络将当前状态与之前的观测关联起来，通过记忆历史信息来估计更准确的策略，从而在动态和复杂的多智能体环境中做出更好的决策。同时，多智能体环境中的决策通常是连续的，RNN能够将这些时间依赖特征融入策略决策中，提高决策的连贯性。

权重继承是指当一个智能体消失时，其他仍然活跃的智能体可以接收并使用其策略参数，以此继续优化决策。在这种情况下，权重继承的动机是为了减少重新学习的时间，同时传递策略知识，形成协作，以避免策略网络因智能体的消失而丢失重要

的信息。为了保证方法的可行性，必须保证权重继承是发生在同构智能体之间的。智能体在协同执行同样的任务，且环境对每个智能体的影响是类似的，权重继承将使部分剩余的智能体能快速适应并接管消失的智能体的任务。

对于智能体 i，其策略参数为 θ_i，当智能体 k 消失时，权重的智能体 i 的新策略为

$$\theta_i = \alpha'\theta_k + (1-\alpha')\theta_i \tag{30}$$

其中，$\alpha'\in[0,1]$ 是一个权重继承系数，用来控制继承策略的比例。若 $\alpha'\fallingdotseq 1$，表示完全继承消失智能体的策略权重；若 $\alpha'\fallingdotseq 0$，表示保持自己原有的策略权重。这个过程同样可以表示为深度为 l 的神经网络中前 $l(\alpha')$ 层的共享操作。

在继承权重后，策略参数 θ_i 需要在新的环境或局部任务变化下进行策略微调，可以通过优化目标函数来实现，表示为

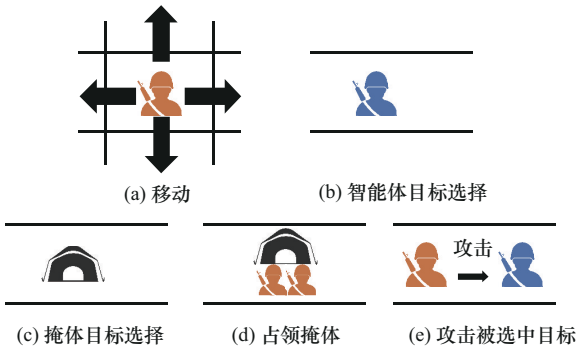
$$\theta_i \leftarrow \theta_i - \eta_{\text{rate}}\nabla_{\theta_i}\ell\left(\theta_i\right) \tag{31}$$

其中，η_{rate} 是学习率，$\ell\left(\theta_i\right)$ 是演员网络的损失函数，即 PPO 的剪切目标函数。通过微调，智能体可以在继承的基础上进一步适应新的策略需求。

3　城市环境多智能体协作对抗

3.1　多智能体协作对抗的动作空间

在城市环境下，多个异构智能体需要根据环境特点及位置特点进行策略生成，从而完成占领目标单位的任务。其动作空间是离散的，如图 6 所示。进一步地，环境对智能体的动作约束如下：type=0 的普通单位无法攻击掩体，也无法将掩体作为攻击目标；type=1 的机动单位则无法占领掩体；且所有智能体只能在设定范围内行动，不能超出边界。



(a) 移动　(b) 智能体目标选择　(c) 掩体目标选择　(d) 占领掩体　(e) 攻击被选中目标

图 6　多智能体协作对抗的动作空间

3.2　多智能体协作对抗的状态空间

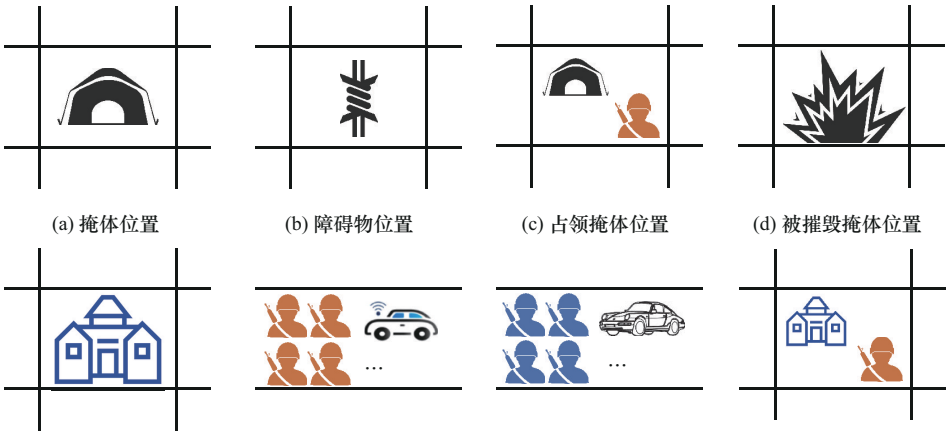
多智能体协作对抗的状态空间分为 2 个区域，一个是代表全局状态的环境状态空间，其状态及含义如图 7 所示。另一个是代表局部状态的智能体状态空间，其状态及含义如图 8 所示。

3.3　多智能体协作对抗的奖励函数

奖励函数是衡量智能体在某个状态下采取某个动作后所获得的回报值，是学习过程中的核心组成部分。奖励函数直接影响智能体的行为学习，它指导智能体优化策略以长期最大化累积的回报。正奖励鼓励行为的重复，而负奖励则抑制某种行为。在训练时，奖励值会在基础奖励上适当进行缩放调整来控制样本的平衡性。多智能体协作对抗的奖励函数如表 1 所示。

3.4　城市环境下多智能体协作对抗

在城市环境中的多智能体协作对抗过程中，智



(a) 掩体位置　(b) 障碍物位置　(c) 占领掩体位置　(d) 被摧毁掩体位置

(e) 目标位置　(f) α组存活情况　(g) β组存活情况　(h) 目标位置占领情况

图 7　环境状态空间

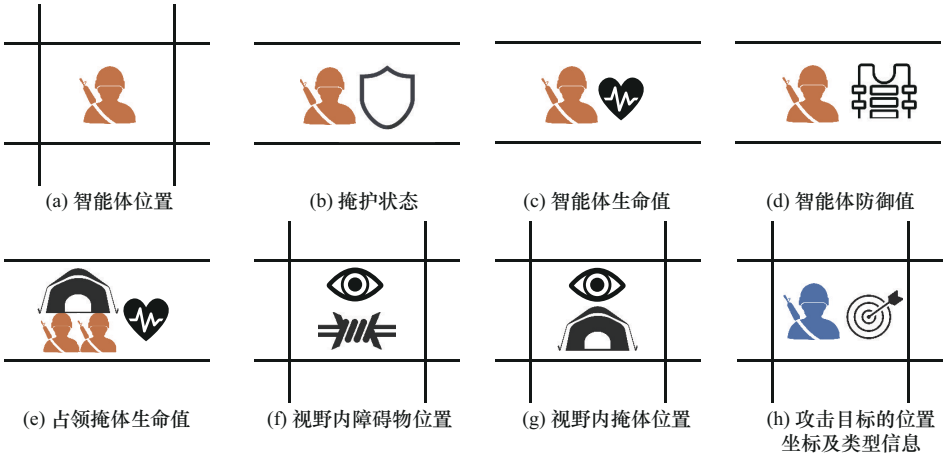


图 8　智能体状态空间

表 1　　**多智能体协作对抗的奖励函数**

对象	分值	说明
移动	–10	智能体每移动一次扣 10 分
击中	$12.5+k\xi$	智能体每击中对方增加基础分数 12.5 分以及 k 倍杀伤力 ξ 分，其中 $k\in[0,1]$
击倒	200/500	击倒普通单位加 200 分，智能体成功击倒机动单位加 500 分
被击倒	–150 /–600	普通单位被击倒扣 150 分，机动单位被击倒扣 600 分
占领掩体	$-100-\lambda_i$	当有 2 个智能体占领同一掩体时扣 100 分，之后每增加 i 个在此基础上扣 λ_i 分，$\lambda\in[100,150]$
占领目标单位	R_{score}	R_{score} 根据目标智能体数量进行动态调整，应高于全部目标智能体被击倒时的总分

能体的行动涉及多个层次的决策与策略调整，如图 9 所示。首先，智能体需要感知环境，包括对方的位置、障碍物、建筑物等信息。由于城市环境的复杂性和视野的受限，单个智能体的感知能力通常不足，因此多智能体通过共享情报来补充感知盲区。α 与 β 的对抗中，需要保证自身存活率的同时以最快速度占领目标单位，这个过程可以通过火力集中、分散行动等多种可能的方式执行。智能体间协作依赖于行动的配合，如利用建筑物提升自身防御同时吸引对方。在这一过程中，智能体不仅需要自主调整动作，还需要确保团队间的实时协作，尤其是在面对突然的环境变化时，智能体之间通过共享局部信息保持进攻和防守的持续性，当某个智能体消失后，其他智能体能够继续他的工作与任务。

智能体需要在复杂的动态环境中相互协作，灵活应对对方的行动，并通过不断的学习与调整实现作战目标，优化进攻路径，提高存活率。这种学习过程不仅优化了单个智能体的决策能力，也提升了整个团队的协作效率。

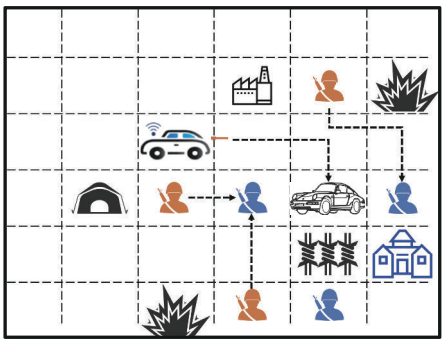


图 9　在城市环境中的多智能体协作对抗示意

4　实验

本节通过实验验证本文所述近端策略优化的城市环境多智能体协作对抗方法的有效性，共设置了 6 组实验，城市环境下的收敛性实验，验证城市环境的训练可靠性；参数共享实验和算法对比实验，验证算法 1 在城市环境下的可靠性及优越性；异构智能体奖励收敛性实验、异构智能体价值损失实验以及异构智能体性能对比实验，证明本文对抗方法对异构智能体在城市环境下进行协作对抗的稳定性。实验中所述的具有具体收敛时间的实验，其智

能体均完成了对应的任务目标，而未收敛则代表智能体无法正确完成任务；对于收敛的智能体实验，其智能体的任务完成时间和损耗均是当前环境下的最优解。

多智能体协作对抗的实验参数如表2所示，本文对抗方法在计算机系统软件为Ubuntu 22.04、GPU硬件为NVIDIA GeForce RTX 3070 Ti的环境下训练运行，算法编译利用了基于Python3.7的PyTorch及其他数学组工具包。方法中嵌入空间大小为2种智能体动作空间的公倍数。该值属于经验参数，并没有具体的设置依据，本文对此进行了简单的实验，设置了6～20个不同大小的空间维度，结果表明空间维度并没有影响智能体的收敛性，但是可能与奖励值的微小波动有关，而更具体的结论和过程探讨会在未来工作中进行进一步研究。实验中，智能体的任务和行为模式参见1.2节，位置信息及环境信息参见1.3节。在后续智能体数量发生变化的实验中，智能体初始位置并未发生改变，依照从左到右、从上到下的原则依次重复放置智能体。

表2　多智能体协作对抗的实验参数

参数	值
学习率	0.000 5
折扣率	0.9
迭代轮数	5 000/5 500
每轮步数	2 000
经验缓存	14 000
批大小	256
激活函数	ReLU/ Sigmoid
动作增益	0.008
RNN隐藏层数量/神经元个数	2/64
RNN学习率	0.000 8
PPO算法限制率	0.3

4.1　城市环境下近端策略优化的收敛性

为了证明本文近端策略优化在城市环境多智能体协作对抗中的可实施性，实验分别设置2、4、6、8个智能体在城市环境下运行，每个组别均被设置为状态空间及动作空间相同的同构智能体，且参与对抗的另一方单位也设置为数量相同的4个组别。为了便于观察，对每个组别的奖励进行了缩放，以测试城市环境下的收敛性。城市环境下的收敛性如图10所示，4个组别的奖励值在约第4 000个回合后趋于稳定，且不同组别收敛轨迹具有高一致性。这是因为本文对抗方法通过统一策略网络参数，使新增智能体不需要重新构建独立模型，不仅能有效应对智能体数量增长带来的环境动态性增强问题，还能避免策略突变对系统的破坏，从而降低了系统复杂度。这也说明，智能体数量的增加不会影响本文对抗方法的稳定性，因而在城市环境下是可行的。

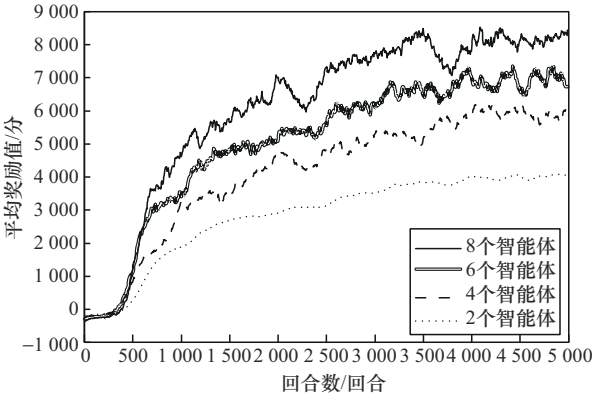


图10　城市环境下的收敛性

4.2　多智能体算法对比

为了验证本文所提出的近端策略优化的AC网络算法在城市环境下的稳定性，分别设置了智能体参数共享实验和算法对比实验。上述实验均采用4个智能体参与城市环境的协作对抗。

4.2.1　参数共享的影响

参数共享实验旨在检验智能体对应的演员网络参数的共享与否对算法的影响。实验结果如表3所示，在其他实验参数不变的情况下，由于共享演员网络参数的智能体能够共享经验和知识且冗余学习量少，计算资源利用率高，因此共享演员网络参数的智能体相较于多个独立演员网络参数的智能体，拥有更快的收敛速度（4 855回合）和更高的奖励回报（5 872分）。

4.2.2　和其他算法的对比

为了验证相比其他算法的优越性，本文对城市环境下协作对抗问题进行智能体求解，对比了以下3种DRL算法：iDQN[28]、CTDE-MADQN[29]以及MADDPG[30]，具体如下。

表3　参数共享的影响

智能体	奖励/分	平均收敛速度/回合
独立参数智能体1	5 840	4 900
独立参数智能体2	5 800	4 945
独立参数智能体3	5 828	4 940
独立参数智能体4	5 812	4 925
共享参数智能体	**5 872**	**4 855**

iDQN[28]通过增量学习提升效率和性能，为环境中每个智能体使用一个自主的Q-learning算法，从而使用环境作为智能体之间唯一的交互来源。

CTDE-MADQN[29]结合了集中训练与分散执行的思想。在训练过程中，所有智能体的状态、动作和奖励信息被集中处理，在执行过程中，各个智能体基于局部信息独立决策。

MADDPG[30]采用了集中训练和分散执行的框架，允许多个智能体在训练过程中共享全局信息，且每个智能体在执行阶段能够根据自己局部的观测信息独立采取动作。

本文算法和其他算法的对比如表4所示，本文算法的平均奖励回报为5 872分，收敛速度为4 855回合，综合性能优于对比算法。与iDQN相比，本文算法的平均奖励回报高出22.67%，且iDQN在5 500回合内未能收敛。这是因为iDQN中每个智能体独立学习，忽略了其他智能体的行为变化，在多智能体环境中容易出现非平稳性问题，容易陷入局部最优解，在大规模多智能体系统中难以有效应用；相比CTDE-MADQN，本文算法平均奖励回报高出4.10%，回合收敛速度加快0.52%，这是因为CTDE-MADQN在训练过程中需要集中化的信息，在智能体数量增加时扩展性差，且对部分观察不完全或高维状态空间的场景处理较为困难，但由于其算法结构简单，因此在使用相同硬件运行时略快于本文算法；相比MADDPG，本文算法平均奖励回报低0.95%，但收敛速度加快了8.14%，这是由于MADDPG主要用于解决多智能体环境中的协作和竞争问题，特别是在智能体之间的交互可能非常复杂的情况下处理连续动作空间的问题，同时其算法的复杂性导致其处理智能体任务时的效率较低。

表4　本文算法和其他算法的对比

算法	奖励/分	平均收敛速度/回合	最快收敛速度/min
iDQN	4 787	5 500（未收敛）	—
CTDE-MADQN	5 641	4 880	**903**
MADDPG	**5 928**	5 250	1 011
本文算法	5 872	**4 855**	992

综上，本文算法克服了iDQN的非平稳性和缺乏协调性问题，避免了MADQN在训练过程中对集中信息的高依赖性，且相较于MADDPG的效率较高。通过有效的信息共享和更高效的协作机制，实现了更稳定的收敛和更优的性能。

4.3　异构智能体对比

本文所述的城市环境包含了2种不同参数空间的异构智能体，设计了异构智能体奖励收敛性实验与价值损失实验，以及异构智能体性能对比实验。上述实验均采用8个智能体参与城市环境的协作对抗。

4.3.1　异构智能体奖励收敛性与价值损失

为了验证本文异构智能体在城市环境下运行的收敛情况，设置了4组分别包含1～4个异构智能体的奖励收敛性实验，并对各个组别进行了奖励缩放，结果如图11所示。在城市环境下，由于嵌入方法将异构智能体的空间投影到同一空间，在处理时能够达到有效的信息对比协作，因此包含单个或多个异构智能体的协作对抗在本文对抗方法的驱动下均可在3 500～5 000回合后收敛，显示了本文对抗方法对异构智能体参与仿真的包容性。此外，随着异构智能体的增加，所提方法仍能够正常收敛，甚至在存在3个异构智能体时，其收敛速度明显加快。这是因为本文对抗方法通过嵌入统一表征与对比协作机制，将异构智能体的状态/动作映射至共享潜在空间，从而消除异构建模差异。该机制不仅能够提高智能体的策略协同，还使系统在规模扩展中维持收敛效率，展现了对异构体增量部署的兼容性，也说明本文对抗方法在异构智能体数量增加的情况下仍能保持稳定性。

单/双异构智能体系统下评论网络价值损失情况如图12所示。无论是单异构智能体还是双异构智能体，评论网络的价值损失均在1 000回合后开始收敛，表明本文对抗方法准确预测智能体在给定

状态下的预期奖励回报，评估策略好坏，且策略已经在城市环境中达到相对理想的状态，可以视为多智能体系统在当前策略下对城市环境的价值理解达到了稳定。

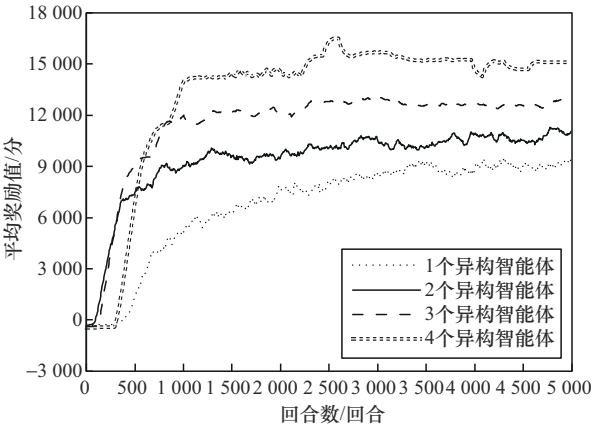


图11　异构智能体收敛性

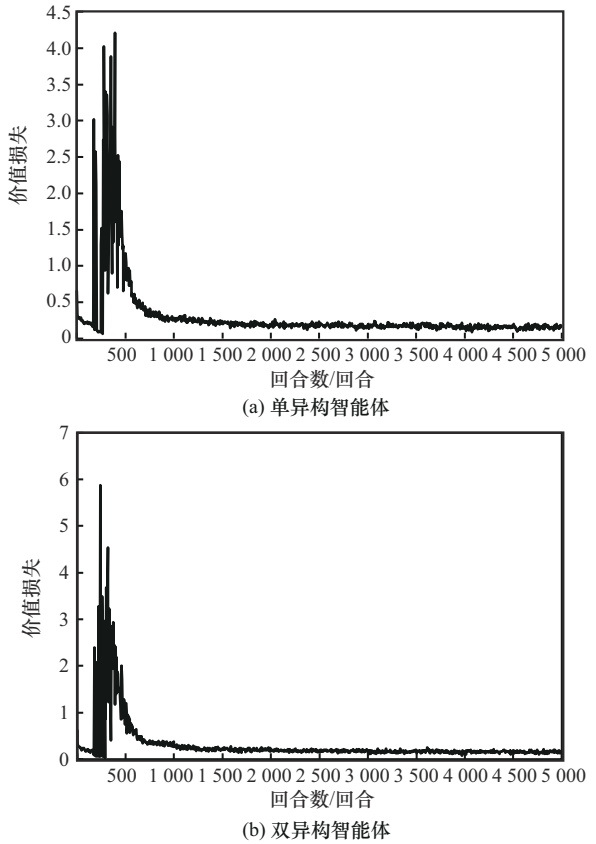


图12　单/双异构智能体系统下评论网络价值损失情况

4.3.2　异构智能体性能对比

异构智能体性能对比实验旨在验证本文对抗方法相较于仅使用强制约束来限制异构智能体行为模型的方法，在奖励回报和收敛速度上的优越性，结果如表5所示。在处理单异构智能体时，本文方法奖励（8 404分）相比强制约束方法奖励（6 975分）高20.49%，收敛速度（4 890回合）相比强制约束方法收敛速度（5 350回合）快9.41%；而在处理2个及2个以上的异构智能体时，强制约束方法未能使智能体正常收敛。

表5　异构智能体性能对比

异构智能体数量	本文方法		强制约束方法	
	奖励/分	收敛速度/回合	奖励/分	收敛速度/回合
1	8 404	4 890	6 975	5 350
2	11 057	4 920	未收敛	未收敛
3	13 076	3 600	未收敛	未收敛
4	15 103	4 620	未收敛	未收敛

在仅使用强制约束来限制智能体的行为空间时，例如当机动单位占领掩体时给予其过大的负奖励回报或强制其移出相应位置，这种方法使智能体由于超参数干扰的影响不能学习到正确的策略，因此无法获取其收敛时的标准奖励回报。强制约束方法下智能体的错误行为奖励如图13所示，智能体在3 000回合后因超参数强制影响导致生成错误的策略，造成奖励失效，结果表明，在仅使用参数约束方法时，异构智能体往往无法获得正确的奖励回报，在分布式执行的过程中仍会发生异常情况，例如普通单位攻击城市环境中的掩体，且收敛速度较慢，甚至无法收敛。

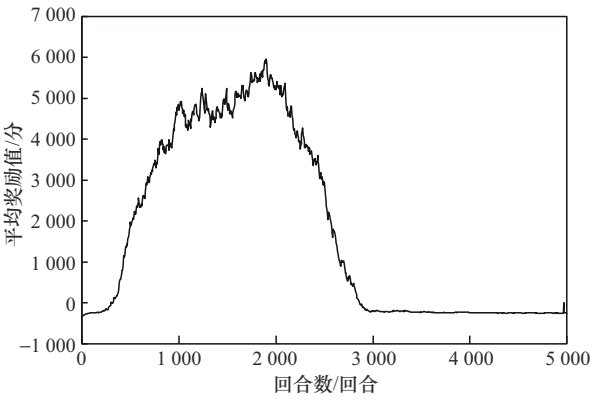


图13　强制约束方法下智能体的错误行为奖励

5　结束语

本文针对城市环境提出了一种近端策略优化的多智能体协作对抗方法。首先设计了包含复杂场景

环境参数及对应的环境约束多智能体城市作战环境，并对环境中智能体的各项参数及环境基础参数进行了详细设计；其次，基于该环境建立了近端策略优化的 AC 网络算法，该算法通过使用嵌入方法解决异构智能体的空间差异问题，对经验回放进行了优化，设计了自适应经验采样方法，并将权重继承方法融入了演员网络中，增加了智能体的迭代能力；最后，对本文方法进行实验，实验结果表明该方法可以完成多智能体在城市环境下的协作对抗任务，并且拥有较快的收敛速度和较高的奖励。

尽管本文方法在城市环境下表现出色，但目前模拟的城市场景仍然存在发展空间，例如视野遮挡、信号屏蔽等实际影响因素，为了保证算法能够在这种信号干扰下正常收敛，往往需要大量正确的对抗信息数据支持。同时下一步将在更复杂的动态场景中验证智能体的泛化能力以及异构智能体嵌入空间维度大小对结果的影响。此外还可以增强智能体异构性和数量，进行算法扩展。在极端动态环境中策略更新效率仍有提升空间。未来工作可以重点探索更多类型智能体的协作对抗、引入真实数据验证算法效果、提升长时任务决策能力、增强对抗场景的可扩展性以提高智能体的适应性和普适性、探索计算资源和网络带宽对算法的限制，并进一步扩大实验规模，探讨不同规模智能体系统下算法的表现，以验证其在大规模环境中的适应性和鲁棒性。同时引入通信优化技术以减轻大规模系统中的通信负担。

参考文献：

[1] 槐泽鹏, 龚旻, 陈克 . 未来战争形态发展研究[J]. 战术导弹技术, 2018, 38(1): 1-8.
HUAI Z P, GONG M, CHEN K. Study of future war form development[J]. Tactical Missile Technology, 2018, 38(1): 1-8.

[2] BERLIN R H. United states army world war II corps commanders: a composite biography[J]. The Journal of Military History, 1989, 53 (2): 147.

[3] 焦志强, 张杰勇, 姚佩阳, 等 . 指挥信息系统生成方案综合评估方法[J]. 控制与决策, 2022, 37(12): 3297-3306.
JIAO Z Q, ZHANG J Y, YAO P Y, et al. Comprehensive evaluation method for construction scheme of C4ISR system[J]. Control and Decision, 2022, 37(12): 3297-3306.

[4] 尹浩, 魏急波, 赵海涛, 等 . 面向有人/无人协同的智能通信与组网关键技术: 现状与趋势[J]. 通信学报, 2024, 45(1): 1-17.
YIN H, WEI J B, ZHAO H T, et al. Intelligent communication and networking key technologies for manned/unmanned cooperation: states-of-the-art and trends[J]. Journal on Communications, 2024, 45(1): 1-17.

[5] APOSPORIS P. A review of global and regional frameworks for the integration of an unmanned aircraft system in air traffic management[J]. Transportation Research Interdisciplinary Perspectives, 2024, 24: 101064.

[6] LI J R, WU G H, WANG L. A comprehensive survey of weapon target assignment problem: model, algorithm, and application[J]. Engineering Applications of Artificial Intelligence, 2024, 137: 109212.

[7] 穆磊, 陈建英, 屈小媚, 等 . 无线传感器及执行器网络中多因素任务分配问题研究[J]. 通信学报, 2017, 38(S1): 25-31.
MU L, CHEN J Y, QU X M, et al. Multifactor task allocation problem in wireless sensor and actuator networks[J]. Journal on Communications, 2017, 38(S1): 25-31.

[8] XU H, ZHANG A, BI W H, et al. Dynamic Gaussian mutation beetle swarm optimization method for large-scale weapon target assignment problems[J]. Applied Soft Computing, 2024, 162: 111798.

[9] WANG Y, WANG J P, HAO J K, et al. Efficient adaptive large neighborhood search for sensor – weapon – target assignment[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2024, 54(10): 6397-6409.

[10] GAO Q H, SCHWEIDTMANN A M. Deep reinforcement learning for process design: review and perspective[J]. Current Opinion in Chemical Engineering, 2024, 44: 101012.

[11] 杜丽娜, 卓力, 杨硕, 等 . 基于强化学习的移动视频流业务码率自适应算法研究进展[J]. 通信学报, 2021, 42(9): 205-217.
DU L N, ZHUO L, YANG S, et al. Survey on reinforcement learning based adaptive bit rate algorithm for mobile video streaming services[J]. Journal on Communications, 2021, 42(9): 205-217.

[12] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[J]. arXiv Preprint, arXiv: 1312.5602, 2013.

[13] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. arXiv Preprint, arXiv: 1509.02971, 2015.

[14] TAN X Y, QU C, XIONG J W, et al. Model-based off-policy deep reinforcement learning with model-embedding[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2024, 8(4): 2974-2986.

[15] YU C, VELU A, VINITSKY E, et al. The surprising effectiveness of PPO in cooperative multi-agent games[C]//Proceedings of the 36th International Conference on Neural Information Processing Systems. New York: ACM Press, 2022: 24611-24624.

[16] 张铖, 朱家烨, 刘泽宁, 等 . 基于多智能体强化学习的异构网络 CRE 偏置动态优化算法[J]. 通信学报, 2023, 44(12): 86-98.
ZHANG C, ZHU J Y, LIU Z N, et al. Multi-agent reinforcement learning based dynamic optimization algorithm of CRE offset for heterogeneous networks[J]. Journal on Communications, 2023, 44(12): 86-98.

[17] 卢卓, 吴启晖, 周福辉 . 有人机/无人机智能协同目标搜索和轨迹规划算法[J]. 通信学报, 2024, 45(1): 31-40.
LU Z, WU Q H, ZHOU F H. Algorithm for intelligent collaborative target search and trajectory planning of MAV/UAV[J]. Journal on Communications, 2024, 45(1): 31-40.

[18] 马悦, 吴琳, 许霄 . 基于多智能体强化学习的协同目标分配[J]. 系统工程与电子技术, 2023, 45(9): 2793-2801.
MA Y, WU L, XU X. Collaborative goal assignment based on multi-agent reinforcement learning[J]. Journal of Systems Engineering and Electronics, 2023, 45(9): 2793-2801.

[19] 肖友刚, 金升成, 毛晓, 等. 基于深度强化学习的舰船导弹目标分配方法[J]. 控制理论与应用, 2024, 41(6): 990-998.
XIAO Y G, JIN S C, MAO X, et al. Missile-target assignment method of naval ship based on deep reinforcement learning[J]. Control Theory & Applications, 2024, 41(6): 990-998.

[20] 樊延平, 宋畅, 薛中兴. 陆军城市巷战作战运用构想创新研究[J]. 国防科技, 2019, 40(5): 122-126.
FAN Y P, SONG C, XUE Z X. Research on the opertation conception innovation for army urban street combat[J]. National Defense Technology, 2019, 40(5): 122-126.

[21] ZHONG Y, KUBA J G, FENG X, et al. Heterogeneous-agent reinforcement learning[J]. Journal of Machine Learning Research, 2024, 25 (32): 1-67

[22] YU X Y, LIN Y F, WANG X S, et al. GHQ: grouped hybrid Q-learning for cooperative heterogeneous multi-agent reinforcement learning[J]. Complex & Intelligent Systems, 2024, 10(4): 5261-5280.

[23] MENG D Y, ZHANG J Y. Distributed learning control for heterogeneous linear multi-agent networks[J]. Automatica, 2024, 169: 111838.

[24] DUCHNOWSKI R, WYSZKOWSKA P. Robust procedures in processing measurements in geodesy and surveying: a review[J]. Measurement Science and Technology, 2024, 35(5): 052002.

[25] MA Z F, ZHANG H, LIU J. DB-RNN: an RNN for precipitation nowcasting deblurring[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2024, 17: 5026-5041.

[26] NGUYEN T T, NGUYEN N D, NAHAVANDI S. Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications[J]. IEEE Transactions on Cybernetics, 2020, 50(9): 3826-3839.

[27] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J]. arXiv Preprint, arXiv: 1707.06347, 2017.

[28] SUI D, XU W P, ZHANG K. Study on the resolution of multi-aircraft flight conflicts based on an IDQN[J]. Chinese Journal of Aeronautics, 2022, 35(2): 195-213.

[29] SUN G X, WANG X M, JIANG R, et al. Beamforming and resource allocation in multi-cell OFDMA systems based on deep transfer reinforcement learning[C]//Proceedings of the 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring). Piscataway: IEEE Press, 2022: 1-6.

[30] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM Press, 2017: 6382-6393.

[作者简介]

米广铭（1999– ），男，回族，北京人，北京工业大学博士生，主要研究方向为人工智能。

张辉（1982– ），男，河南周口人，博士，北京工业大学副教授、硕士生导师，主要研究方向为视觉信息处理、人工智能等。

张菁（1975– ），女，广东梅州人，博士，北京工业大学教授、博士生导师，主要研究方向为图像/视频处理、人工智能等。

卓力（1971– ），女，江苏徐州人，博士，北京工业大学教授、博士生导师，主要研究方向为图像/视频处理、人工智能等。