文章编号:2095-6134(2024)02-0231-10

基于不确定度的多智能体信用分配方法*

杨光开[1,2],陈皓[1,2],张茗奕[1],尹奇跃[1,2],黄凯奇[1,2,3†]

(1 中国科学院自动化研究所智能系统与工程研究中心,北京 100190;2 中国科学院大学人工智能学院,北京 100049;
3 中国科学院脑科学与智能技术卓越创新中心,上海 200031)

(2022 年 3 月 18 日 收稿; 2022 年 4 月 26 日 收修改稿)

Yang G K, Chen H, Zhang M Y, et al. Uncertainty-based credit assignment for cooperative multi-agent reinforcement learning[J]. Journal of University of Chinese Academy of Sciences, 2024, 41(2): 231-240. DOI: 10.7523/j.ucas.2022.047.

摘 要 近年来,部分可观测条件下多智能体协同受到广泛关注。中心化训练分布式执行作为处理这类任务的通用范式面临信用分配这一核心问题。值分解是该范式中的代表性方法,通过混合网络将联合状态动作值函数分解为多个局部观察动作值函数以实现信用分配,在很多问题中表现很好。然而这些方法维持对混合网络参数的单一点估计,因缺乏不确定度表示而难以有效应对环境中的随机因素导致只能收敛到次优策略。为缓解这一问题,对混合网络进行贝叶斯分析,提出一种基于不确定度的多智能体信用分配方法,通过显式地量化参数的不确定度来指导信用分配。考虑到智能体之间复杂的交互,利用贝叶斯超网络隐式地建模参数任意复杂的后验分布,以避免先验地指定分布类型而陷于局部最优解。在星际争霸微操环境中的多个地图上与代表性算法的性能进行对比与分析,验证了算法的有效性。

关键词 多智能体协同;深度强化学习;信用分配;贝叶斯超网络

中图分类号:TP183 **文献标志码**:A **DOI**:10.7523/j.ucas.2022.047

Uncertainty-based credit assignment for cooperative multi-agent reinforcement learning

YANG Guangkai[1,2], CHEN Hao[1,2], ZHANG Mingyi[1], YIN Qiyue[1,2], HUANG Kaiqi[1,2,3]

(1 *Center for Research on Intelligence System and Engineering, Institute of Automation, Chinese Academy of Sciences, Beijing* 100190, *China*; 2 *School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing* 100049, *China*; 3 *Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai* 200031, *China*)

Abstract In recent years, multi-agent cooperation under partially observable conditions has attracted extensive attention. As a general paradigm to deal with such tasks, centralized training with decentralized execution faces the core problem of credit assignment. Value decomposition is a representative method within this paradigm. Through the mixing network, the joint state action-value

* 国家自然科学基金(61876181)、北京市科技创新计划(Z19110000119043)、中国科学院先导科技专项(QYZDB-SSWJSC006)和中国科学院青年创新促进会项目资助

† 通信作者,E-mail: kqhuang@nlpr.ia.ac.cn

function is decomposed into multiple local observation action-value functions to realize credit assignment, which performs well in many problems. However, the single point estimation of the mixing network parameters maintained by these methods lacks the representation of uncertainty and is thus difficult to effectively deal with the random factors in the environment, resulting in convergence to the suboptimal strategy. To alleviate this problem, this paper performs Bayesian analysis on the mixing network and proposes a method based on uncertainty for multi-agent credit assignment, which guides the credit assignment by explicitly quantifying the uncertainty of parameters. Considering the complex interactions among agents, this paper utilizes the Bayesian hypernetwork to implicitly model the arbitrary complex posterior distribution of the mixing network parameters, to avoid falling into the local optima by specifying the distribution type a priori. This paper compares and analyzes the performance of representative algorithms on multiple maps in StarCraft multi-agent challenge (SMAC) and verifies the effectiveness of the proposed algorithm.

Keywords multi-agent cooperation; deep reinforcement learning; credit assignment; Bayesian hypernetwork

现实世界中很多复杂任务可以自然地建模为多智能体合作博弈问题如无人驾驶车辆协同[1]、传感器网络[2]和无人机群控制[3]等。近年来,多智能体强化学习作为决策智能领域内一直以来的研究热点在很多任务中取得了巨大成功,在各类游戏(如 DOTA2[4]、星际争霸[5]和王者荣耀[6])中展现出超凡的性能。相比于单智能体强化学习,在多智能体强化学习场景中,智能体不仅需要与环境交互,还需要和其他智能体进行策略交互。此时,环境反馈的奖励信号取决于所有智能体动作组成的联合动作而不是单个智能体的动作。联合动作空间随智能体数目呈指数级增长,中心化训练中心化执行的学习范式将所有智能体视为一个中心智能体,因而受到扩展性限制,无法应用于智能体数目较多的场景中[7]。部分可观测限制与通信的约束使得智能体分布式决策成为必要[8]。然而,由于策略交互导致智能体之间策略相互影响,并且在训练过程中智能体策略不断发生变化,使得每个智能体学习不稳定,其学习目标动态漂移。因此分布式训练分布式执行范式面临环境非平稳挑战[9],难以处理复杂合作任务。为应对这些挑战,中心化训练分布式执行作为上述两种范式的结合被提出来,并逐渐成为多智能体强化学习领域中广泛通用的学习范式,在一系列复杂问题中取得了令人满意的成果[8,10-14]。基于这一范式,在中心化训练阶段,学习算法能够访问环境的全局状态信息作为额外的学习信号,同时所有智能体形式上被视为一个中心化智能体以确定学习目标。而在分布式执行阶段,由于部分可观测限制,每个智能体无法访问环境的全局状态信息,只能基于自己的局部观测信息依赖于自己的策略选择动作实现分布式决策。

在多智能体合作博弈中,智能体与环境交互时,环境只返回单一奖励信号作为对智能体联合动作的反馈。如何合理地实现信用分配,即如何合理地分配奖励信号以促进每个智能体的学习,是中心化训练分布式执行的核心问题[11]。值分解方法作为该范式中代表性的方法,通过混合网络将全局的联合状态动作值函数以确定的形式分解为每个智能体的局部观测动作值函数,来实现多智能体的信用分配[8]。这些方法基于超网络维持对混合网络参数的单一点估计,这种估计缺乏不确定度表示而难以有效应对环境中的随机因素导致只能收敛到次优策略[15-16]。为缓解这一问题,本文对混合网络进行贝叶斯分析,提出一种基于不确定度的多智能体信用分配方法,通过显式地量化参数的不确定度来指导信用分配。考虑到智能体之间复杂的交互,利用贝叶斯超网络[17-18]隐式地建模参数任意复杂的后验分布以避免先验地指定分布类型而陷于局部最优解。

本文贡献包括如下两方面内容:1)基于值分解框架,对混合网络进行贝叶斯分析,提出基于不确定度的多智能体信用分配方法,显式地量化混合网络参数的不确定度以更好应对环境中的随机因素;2)复杂的协同任务中智能体之间交互十分复杂,为实现信用分配带来极大挑战,利用贝叶斯超网络隐式地建模网络参数任意复杂的后验分布

以避免先验指定的简单分布类型导致局部最优,提升了方法的扩展性。

1　问题背景与相关工作

1.1　问题定义

现实世界中的多智能体协同任务大多是部分可观测的,智能体只能基于自己的局部观测决策。本文研究对象为部分可观测条件下的完全合作的多智能体协同任务,这类任务常用分布式部分可观测马尔科夫决策过程(decentralized partially observable Markov decision process, Dec-POMDP)[19]进行数学建模。Dec-POMDP 定义为一个多元组 $G=(n,S,U,r,\gamma,P,Z,O)$,其中 S 为环境的状态空间。在每个时间步,每个智能体 $a\in A=\{1,\cdots,n\}$,同时选择一个动作 $u^a\in U_a$,组合得到所有智能体的联合动作 $u=(u^1,\cdots,u^n)$,$u\in U\equiv U_1\times\cdots\times U_n$ 与环境交互,环境根据状态转移概率 $P(s'\mid s,u):S\times U\times S\to[0,1]$ 转移到新状态。所有智能体共享奖励函数 $r(s,u):S\times U\to R$,其中 γ 为奖励折扣因子。为确保部分可观测约束,为每个智能体指定相应的观测函数 $O(s,a):S\times A\to Z$,从状态 s 中采样出观测 z。每个智能体 a 拥有自己的动作观测历史 $\tau^a\in T\equiv(Z\times U)^*$,并利用它得到自己的随机策略 $\pi^a(u^a\mid\tau^a):T\times U\to[0,1]$。累积折扣回报定义为 $R_t=\sum_{i=0}^{\infty}\gamma^i r_{t+i}$,学习算法的目标在于找到能最大化累积奖励的联合策略 π^*。

1.2　信用分配

多智能体独立学习如独立 Q 学习(independent Q learning, IQL)[9]为了实现信用分配直接将环境反馈的奖励在所有智能体之间共享或者取平均分配。这种方式简单直接,扩展性强,适用于很多问题。然而实验[20]发现共享奖励无法给每个智能体分配准确的奖励信号,会导致“懒惰智能体”[21]现象,智能体学习不够充分,难以应用于复杂的协同任务中。反事实多智能体策略梯度方法(counterfactual multi-agent policy gradient, COMA)[11]通过差异奖励为每个智能体显式地分配奖励信号,利用中心化评论家估计智能体动作的反事实优势值实现信用分配,但由于需要对奖励基准值做出单独估计因而难以在需要复杂合作策略的困难任务中发挥作用。SQDDPG(Shapley Q-value deep deterministic policy gradient)[22]则利用合作博弈[23]从理论上指定了信用分配的框架,基于智能体的边际贡献夏普利值(Shapley value)[24]显式地分配奖励信号,保证了分配的公平性。该方法假设了合作博弈本身的凸性,但对于实际问题而言这一假设过强且很难验证,在实际问题中无法取得令人满意的效果。显式方法能直接指定每个智能体获得的奖励大小,具备一定的可解释性,但智能体之间交互行为的复杂性导致显式分配极为困难。

现有的相关方法多采用值分解方法隐式地实现多智能体信用分配。值分解网络(value-decomposition network, VDN)[21]将联合状态动作值函数分解为多个智能体观测动作值函数的线性加和,而 QMIX[8]则利用超网络生成参数得到混合网络并在单调性约束下对联合动作值函数进行分解。MAVEN[13]是为了克服 QMIX 结构性约束导致的探索限制而提出的分层强化学习方法来实现深度探索,但因为计算开销增加导致实际效果欠佳。QTRAN[25]则是通过对全局状态动作值函数的变换消除了结构性约束但也由于计算复杂度增加而在实际中表现不好。LICA(learning implicit credit assignment)[26]利用策略分解来解决信用分配,消除了 QMIX 等值分解方法的结构性约束但样本效率较低,难以高效地进行训练。COPPO(coordinated proximal policy optimization)[27]将单智能体强化学习中的 PPO(proximal policy optimization)[28]算法拓展到多智能体场景中实现了动态信用分配,但由于 PPO 本身依赖于算法的具体实现,算法效果方差比较大。另外一些工作结合元强化学习方法来处理信用分配问题[29],但面临优化上的挑战[30]。此外 MMD-MIX[31]将值分解与分布性强化学习(distributional reinforcement learning, DRL)[32]结合,显式建模联合状态动作值函数的不确定度以应对环境中的随机性,但实际中效果不稳定。

1.3　值分解框架

值分解作为中心化训练分布式执行的代表性方法在很多问题中取得了良好效果。QTRAN 中提出的个体-全局最优(individual-global-max, IGM)条件定义了联合策略和分布式策略之间的最优一致性,利用该一致性可以在线性时间内得到最优的分布式策略。IGM 条件可以表示为

$$\underset{u}{\operatorname{argmax}} Q_{\text{tot}}(\tau,u)=\begin{pmatrix}\operatorname{argmax}_{u^1} Q_1(\tau^1,u^1)\\ \vdots\\ \operatorname{argmax}_{u^n} Q_N(\tau^n,u^n)\end{pmatrix}. \tag{1}$$

其中：$Q_{\text{tot}}(\tau,u)$ 为联合动作值函数，$Q_i(\tau^i,u^i)$ 为第 i 个智能体的动作值函数。

VDN 将联合动作值函数 $Q_{\text{tot}}(\tau,u)$ 分解为每个智能体 a 动作值函数 $Q_a(\tau^a,u^a;\theta^a)$ 的简单加和，通过如下所示的加性约束满足了 IGM 条件：

$$Q_{\text{tot}}(\tau,u)=\sum_{i=1}^{n} Q_i(\tau^i,u^i;\theta^i). \tag{2}$$

QMIX 是 VDN 的改进版本，利用神经网络来分解联合动作值函数，实现更加复杂的信用分配策略。QMIX 利用单调性约束来满足 IGM 条件：

$$\frac{\partial Q_{\text{tot}}(\tau,u)}{\partial Q_i(\tau^i,u^i)}\geqslant 0,\ i=1,\cdots,n. \tag{3}$$

在执行时对混合网络的参数取绝对值即可实现此单调性约束。

然而，当前的值分解如 QMIX 维持对混合网络参数的单一点估计，缺乏不确定度表示而难以处理环境中的随机因素，在复杂的协同任务中无法取得令人满意的结果。

1.4 贝叶斯神经网络和贝叶斯超网络

为建立安全人工智能和保持探索，维持一定的不确定度在实际问题中不可避免。例如，在医学领域，计算机诊断系统应该对诊断结果保持某种程度的不确定性以防做出过于确信的判断而导致误诊。除此之外，对于机器学习而言，普通的前向神经网络容易过拟合[16]。在监督学习和强化学习问题中，这些网络往往无法应对训练数据的随机因素(如噪声)而对类别标签或者动作选择做出不准确的预测，维持网络参数不确定度有助于缓解这一问题。为此，Blundell 等[16] 提出 Bayes-by-backprop 来捕捉神经网络参数的不确定度。直观地说，贝叶斯神经网络中的参数不再是单个的值，而是从某个隐式后验分布中采样得到的样本。参数的不确定度被对应的后验分布方差量化。贝叶斯神经网络的这一特性使得它被广泛应用于很多领域：在连续学习中，为防止网络参数过拟合于旧任务的训练数据，贝叶斯神经网络被用来缓解“灾难性遗忘”[33]。在多臂老虎机(multi-armed bandit)问题中，贝叶斯神经网络被用于更好的平衡策略的探索与利用问题[16]。Lipton 等[34] 利用贝叶斯神经网络来提升算法在任务导向型对话问题中的探索能力。在强化学习中，贝叶斯神经网络用于捕捉智能体对动作值函数估计的不确定度以指导探索。在实际问题中，真实的后验分布往往无法求出，只能通过变分近似得到。贝叶斯神经网络的学习依赖于变分自由能[35]，通过最小化变分近似和后验分布之间的 KL 散度来实现：

$$\begin{aligned}\boldsymbol{\theta}^* &= \operatorname{argmin}_{\boldsymbol{\theta}}\mathrm{KL}[q(\boldsymbol{w}\mid\boldsymbol{\theta})\,\|\,p(\boldsymbol{w}\mid D)]=\\ &\operatorname{argmin}_{\boldsymbol{\theta}}\{\mathrm{KL}[q(\boldsymbol{w}\mid\boldsymbol{\theta})\,\|\,p(\boldsymbol{w})]-\\ &\mathrm{E}_{q(\boldsymbol{w}\mid\boldsymbol{\theta})}[\log p(D\mid\boldsymbol{w})]\}.\end{aligned} \tag{4}$$

其中：$\boldsymbol{\theta}$ 为待学习的参数；$q(\boldsymbol{w}\mid\boldsymbol{\theta})$ 为参数 $\boldsymbol{w}$ 的变分近似；$p(\boldsymbol{w}\mid D)$ 为参数 $\boldsymbol{w}$ 的真实后验分布；$p(\boldsymbol{w})$ 为参数 $\boldsymbol{w}$ 的先验分布，借助领域知识预定义；D 为训练数据。

在利用贝叶斯神经网络捕捉参数参数的不确定度时，变分近似往往局限于预定义的分布类型，如单峰高斯分布。然而现实问题十分复杂，单峰高斯分布等预定义的简单分布类型无法建模更为复杂的分布类型如多峰分布、混合高斯分布等。为此，Pawlowski 等[18] 进一步提出贝叶斯超网络，该网络对一个简单的噪声分布进行变换，可以隐式建模出参数任意复杂的后验分布，极大提升了方法的灵活性和扩展性，在实际中得到广泛应用。

2 基于不确定度的信用分配方法

本节详细介绍提出的基于不确定度的多智能体信用分配方法(uncertainty-based credit assignment, UCA)，利用贝叶斯超网络隐式地建模混合网络参数的后验分布，显式地度量参数或者信用分配的不确定度。

2.1 贝叶斯混合网络

在单智能体强化学习中，不确定度包含两部分：策略网络、动作值函数网络或者值函数网络参数的不确定度和累积奖励的不确定度[36]。参数不确定度可以通过对参数注入噪声实现[37]，或者通过贝叶斯推断获得对参数的后验分布。分布性强化学习通过维持对动作值函数的后验分布来表示累积奖励的不确定度。但是在中心化训练分布式执行范式中，除了智能体策略网络等参数以及累积奖励不确定度外，还包括信用分配的不确定度，即值分解方法 QMIX 中混合网络参数的不确定度。MMD-MIX 将值分解方法和分布性强化学习相结合量化多智能体强化学习中联合动作值函

数的不确定度。然而当前的值分解方法完全忽略了对信用分配不确定度的表示,因此无法有效应对环境中的随机因素,智能体难以进一步学习更合理的信用分配方法,导致最终只能得到次优策略。本文显式地考虑混合网络参数的不确定度,即信用分配的不确定度。

QMIX 算法通过超网络基于每个时刻环境的全局状态信息生成并取绝对值得到混合网络参数,对参数维持单一点估计。为表示参数的不确定度,提出贝叶斯混合网络替换原有的混合网络。类似于贝叶斯神经网络,贝叶斯混合网络参数通过从建模出的后验分布中采样得到,后验分布的方差显式地度量参数的不确定度。

隐式分布的概率密度很难写出,但是采样很方便,确保对近似期望和对应梯度的简单计算。最著名的隐式分布来源于对抗神经网络,其中生成器将简单的噪声样本变换为高精度的图像。

如图 1 所示,隐式后验分布通过由 2 个部分构成的贝叶斯超网络隐式建模:预编码器 $g_{\psi}(\boldsymbol{\xi})$ 和参数生成器 $f_{\rho}(\boldsymbol{z},\boldsymbol{s}_t)$。首先,从给定的噪声分布中采样出噪声样本,预编码器对噪声样本 $\boldsymbol{\xi}$ 进行预编码得到隐变量 $\boldsymbol{z}$。然后将隐变量 $\boldsymbol{z}$ 与环境的全局状态信息 $\boldsymbol{s}_t$ 拼接得到 $[\boldsymbol{z},\boldsymbol{s}_t]$,输入参数生成器,最后对结果取绝对值得到贝叶斯混合网络的参数。由于取绝对值保证了单调性约束,因此该方法依然满足 IGM 条件。概括地说,整个过程即是贝叶斯超网络将简单的噪声分布逐渐变换为需要的后验分布,但是在生成参数时依然充分利用环境的全局状态信息。在实验中,本文选取噪声分布为多维高斯分布,其中每一维相互独立,$\boldsymbol{\xi} \sim \boldsymbol{N}(0,0.05)$。

2.2　贝叶斯超网络学习

前面介绍了用变分近似来学习贝叶斯神经网络如公式(4)。该式包含 2 项,第 1 项为正则化项,确保变分近似不能偏离参数的先验分布太远,而第 2 项则是误差重建项,通过对数似然度实现。在实际中,往往为了处理简单,参数的变分近似被指定为特殊的分布类型如单峰高斯分布等。然而因为任务的高复杂度,这些预定义的简单分布类型难以建模任意复杂的分布类型。因此,Pawlowski 等[18]提出贝叶斯超网络来建模复杂的分布类型。此时变分近似 $q(\boldsymbol{w}\mid\boldsymbol{\theta})$ 由神经网络表示,无法写出具体的分布形式,难以直接求 KL 散度。Jiang 等[38]提出近似任意 2 个分布 KL 散度的核估计方法

$$\mathrm{KL}[q(\boldsymbol{w}\mid\boldsymbol{\theta})\,\|\,p(\boldsymbol{w})] = \frac{d}{N}\sum_{i=1}^{N}\log\frac{\min_j\|\boldsymbol{w}_q^i-\boldsymbol{w}_p^j\|}{\min_{j\neq i}\|\boldsymbol{w}_q^i-\boldsymbol{w}_q^j\|}+\log\frac{m}{N-1}. \tag{5}$$

其中:$\boldsymbol{w}_q$ 和 $\boldsymbol{w}_p$ 分别是变分近似和先验分布的样本;d,N,m 分别为参数的维度、变分近似样本的个数以及先验分布样本的个数。该 KL 散度近似公式类似于最近邻距离的比率。在实验中,本文选取先验分布为每一维相互独立的多维高斯分布,$\boldsymbol{w}_p \sim \boldsymbol{N}(0,0.5)$。考虑到先验分布每维的独立性,本文将参数也独立地考虑,取 $d=1$。

2.3　总体损失函数

根据公式(4),本文算法框架的损失函数包含两部分:KL 散度正则化项和对数似然度误差

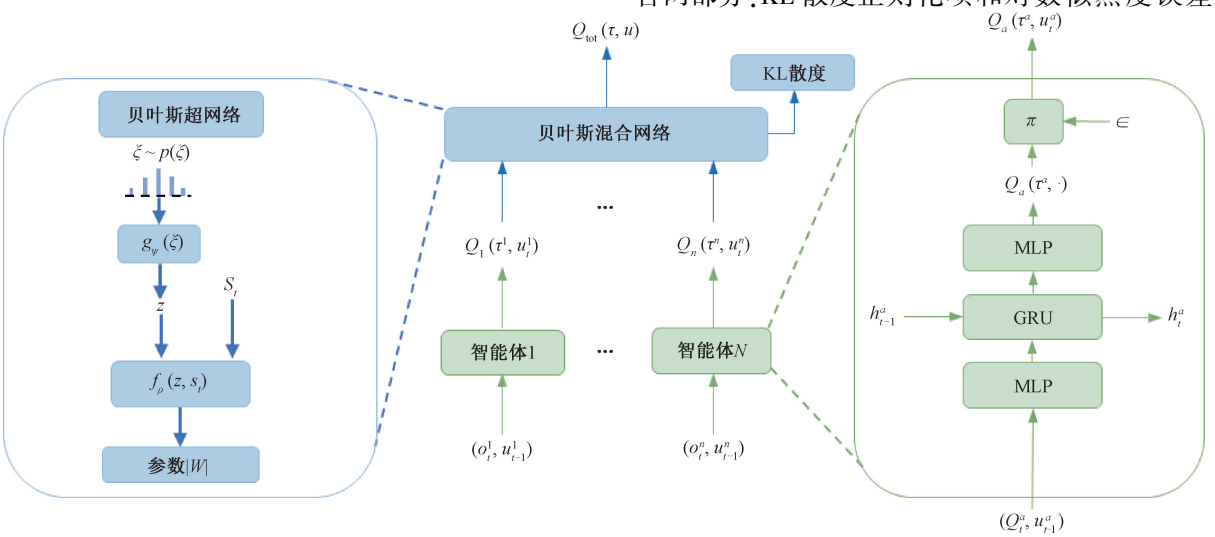

图 1　UCA 算法的网络结构

Fig. 1　Network Structure of UCA algorithm

项。其中第 1 项由公式(5)给出近似,而第 2 项被 DQN 算法中的时序差分项(TD-Error)给出。因此本文算法框架的总体损失函数如下式所示:

$$L_{\text{tot}}(\boldsymbol{\varphi}) = L_{\text{TD}}(\boldsymbol{\varphi}) + \lambda_{\text{KL}} \times L_{\text{KL}}(\boldsymbol{\theta}). \quad (6)$$

其中,λ_{KL} 是平衡损失函数的温度系数。

$$L_{\text{TD}}(\boldsymbol{\varphi}) = \sum_{i=1}^{b} [(y_i^{\text{tot}} - Q_{\text{tot}}(\tau, u; \boldsymbol{\varphi}))^2]. \quad (7)$$

$$y^{\text{tot}} = r + \gamma \max_{u'} Q_{\text{tot}}(\tau', u'; \boldsymbol{\varphi}^-). \quad (8)$$

其中:b 是从经验回放池中采样得到的一个批次样本的数量;$\boldsymbol{\varphi}$ 是所有需要学习的参数,包括贝叶斯超网络的参数;$\boldsymbol{\varphi}^-$ 是对应的目标网络的参数;$\boldsymbol{\theta} = [\boldsymbol{\psi}, \boldsymbol{\rho}]$ 为贝叶斯超网络的参数。基于该总体损失函数,算法端到端地进行训练。

3 实验设计与结果分析

本节在复杂的多智能体协同任务上开展充分的实验以评估本文提出的算法 UCA 的性能,并与领域内代表性算法进行对比来证明本文方法的有效性。

3.1 测试环境

星际争霸Ⅱ作为世界范围内享有盛誉的实时策略游戏,提供了丰富的同构和异构单元,这些单元之间可以涌现出极其复杂的合作行为。因此本文选取星际争霸微操环境(StarCraft multi-agent challenge, SMAC[39],简称为星际微操环境)进行算法性能测试。微操是指对每个单元进行分别控制以击败对手。在星际微操环境中,一个由多种类型的友方单元组成的联盟与另一组敌方单元构成的团队进行对抗,友方单元受分布式智能体控制,而敌方单元受内置的启发式规则控制。针对一场战役,联盟内部的智能体需要彼此合作才能打败敌方团队从而获取胜利。图 2 展示了在地图 MMM2 上的对战画面。根据任务的困难程度,SMAC 环境将地图分为简单、困难和超级困难 3 个等级。表 1 给出了本文选择的用于算法性能测试的不同难度等级的地图。其中,狂热者、跳虫是近战单元。掠夺者、海军陆战队员、巨像、追猎者是远程单元。医疗舰不具备攻击能力,但可以治疗友方单元。这些地图包含了异构、非对称、大动作空间和微操技巧等挑战。本文的星际争霸版本为 SC2.4.10(B75689)[26]。

为保证部分可观测性,每个智能体的视野局限在一定的范围内,范围之外的信息智能体无法获取。具体而言,智能体能在视野范围内获得的信息包括范围内联盟和敌方的属性:(距离,相对 x,相对 y,血量,护盾,单元类型)。仅仅基于智能体的局部观测信息,无法区分其他智能体是处于视野范围外还是已经死亡。智能体只能基于获得的部分观测信息决策。环境的全局状态信息需要联合所有智能体的局部观测得到。在 SMAC 环境中,智能体拥有离散的动作空间,包括朝上下左右 4 个方向移动、攻击某个目标、停止和不执行任何动作。其中,不执行任何动作单独适用于阵亡的智能体。SMAC 环境通过禁用星际争霸原有的移动-攻击指令,将移动和攻击指令彻底分离开,使得智能体必须在每个时间步都进行动作决策,导致决策更加复杂。

图 2 地图 MMM2 的对战画面

Fig. 2 Combat scenario of map MMM2

表 1 实验中选定的 SMAC 地图

Table 1 SMAC maps used in our experiments

地图名称	友方单元	敌方单元	地图类型	难度等级
1c3s5z	1 名巨像, 3 名追猎者, 5 名狂热者	1 名巨像, 3 名追猎者, 5 名狂热者	异构, 对称	简单
8m_vs_9m	8 名海军陆战队员	9 名海军陆战队员	同构, 非对称	简单
10m_vs_11m	10 名海军陆战队员	11 名海军陆战队员	同构, 非对称	简单
2c_vs_64zg	2 名巨像	64 只跳虫	同构, 非对称	困难
MMM2	1 艘医疗舰 2 名掠夺者 7 名海军陆战队员	1 艘医疗舰 3 名掠夺者 8 名海军陆战队员	同构, 非对称	超级困难
27m_vs_30m	27 名海军陆战队员	30 名海军陆战队员	同构, 非对称	超级困难

SMAC 环境有两种奖励设定:稠密奖励和稀疏奖励。其中稀疏奖励设定中,环境在每局结束后才会根据胜负反馈奖励信号,其他时刻不存在奖励信号。而在稠密奖励设定下,联盟在每个时间步都会接收到环境返回的全局奖励。本文选择稠密奖励设定:对敌方的伤害值减去友方伤害值的一半,获胜之后反馈奖励值为友方智能体剩余血量总和加额外的 200 点奖励值。

3.2　训练和测试细节

本文选择测试胜率为评估指标。对于每个算法,在选定的所有地图上都使用 5 个不同的随机种子运行 200 万个时间步进行实验并画出测试胜率-时间步曲线图,且用阴影表示了 25% ~75% 分位数的结果,如图 3 所示。

本文对比的基线算法包括经典的值分解算法 QMIX[8]、VDN[21]、QTRAN[25]、MAVEN[13] 和 IQL[9],如表 2 所示。本文实验基于 Pymarl 框架[8],同时为了排除某些技巧[40]的影响以保证对比的公平性,所有算法的超参数完全按照 Pymarl 中的设置。对于 UCA 算法特有的超参数:32 个样本估计 KL 散度,温度系数 λ_{KL} = 0.12,噪声样本的维度设置为 2;预编码器有 2 层隐藏层,分别具有 32 个单元和 16 个单元;中间激活函数为 ReLU。参数生成器的结构和原始 QMIX[8] 的超网

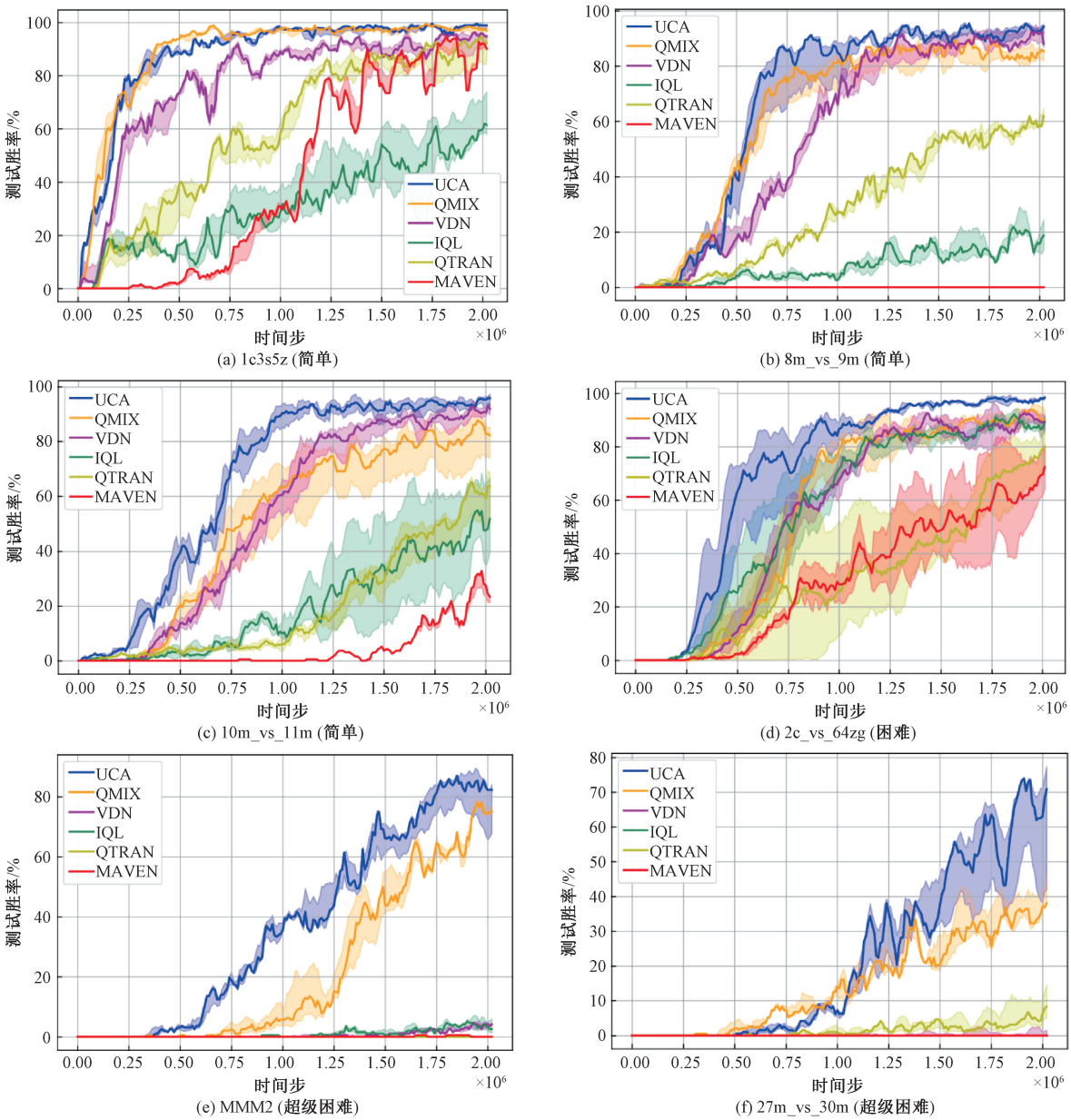

图 3　UCA 算法在 SMAC 上的实验结果

Fig. 3　Experimental results of UCA algorithm on SMAC

表 2 算法测试胜率的中值性能

Table 2 Median performance of the test win percentage %

地图名称	UCA	QMIX	VDN	IQL	QTRAN	MAVEN
1c3s5z	**99**	97	97	69	94	92
8m_vs_9m	**94**	86	89	22	65	0
10m_vs_11m	**96**	84	91	56	62	23
2c_vs_64zg	**98**	90	89	89	80	72
MMM2	**82**	73	4	7	0	0
27m_vs_30m	**71**	38	0	0	8	0

络结构保持相同:生成混合网络权重参数的超网络由 64 个具有 ReLU 非线性单元的单个隐藏层组成,而生成偏置参数的超网络由 32 个单元的单个隐藏层组成。所有神经网络都使用 RMSprop 优化器,学习率设定为 0.000 5。测试时,每 1 万个时间步评估记录算法的测试胜率,并且为了排除异常值影响,本文选择比较算法胜率的中位数而不是均值。

3.3 实验结果

本文将 UCA 算法与 QMIX[8]、VDN[21]、IQL[9]、QTRAN[25]和 MAVEN[13]在星际争霸微操环境中选定的 6 个地图上进行对比,具体实验胜率结果如表 2 所示。其中:1c3s5z,8 m_vs_9 m,10 m_vs_11 m 是简单地图;2c_vs_64zg 为困难地图;27 m_vs_30 m 和 MMM2 是超级困难地图,需要学到集火、"放风筝"等某些特定的微操技巧才有可能取得对局的胜利。

从图 3 的胜率曲线和表 2 的胜率数值结果可以发现,UCA 算法在上述所有地图上都取得了超过现有代表性算法的性能,学到了最好的合作策略,尤其是在困难和超级困难地图上提升十分明显。在简单地图上:对于 1c3s5z,除 IQL[9],其他算法都学到了不错的合作策略,而 UCA 虽然一开始由于分布方差不为零使得算法具有一定的探索行为而相比之下收敛较慢,但是最终仍然取得了最好的效果。对于 8 m_vs_9 m 和 10 m_vs_11 m 而言,虽然都是简单地图,但对抗的非对称性给学习算法带来很大挑战,UCA 算法相比于其他任何算法都更快地收敛到了最好的合作策略。在困难地图 2c_vs_64zg 中,2 个巨像之间协同,联合动作空间较小,因此除 QTRAN[25]和 MAVEN[13]由于计算开销过大而效果较差之外,其他算法都能取得不错的效果,然而 UCA 算法则进一步更快收敛到更好的合作策略。对于超级困难地图,MMM2 地图中智能体的动作空间不相同,异构性为协同带来很大挑战。而 27 m_vs_30 m 地图中对抗的非对称性和大联合动作空间导致很多算法完全无法学到有效的合作策略。智能体之间需要非常精妙的合作才能在超级困难地图上获胜。相比之下 QMIX[8]虽然有一定的效果,但因为缺失了对混合网络参数不确定的考量,而使得网络发生过拟合,算法无法学习到更好的信用分配方法,导致最终只能得到次优合作策略。而 UCA 算法尤其是在超级困难地图上取得了非常好的效果,有了极大的提升。

4 总结与展望

本文对值分解中的混合网络进行贝叶斯分析,提出一种基于不确定度的多智能体信用分配方法,通过对网络参数做出不确定度估计以指导信用分配来更好地应对环境中的随机因素。为了显式地量化混合网络参数的不确定度,同时考虑到智能体之间交互行为的复杂性,避免先验地指定分布类型而陷于局部最优解,利用贝叶斯超网络将简单的噪声分布变换为足够复杂的分布类型以隐式建模参数的后验分布,并从中采样得到贝叶斯混合网络参数来实现信用分配。实验结果表明,本文提出的 UCA 算法在星际争霸微操环境中的多个地图上均取得了超过现有代表性算法的性能,尤其是在困难和超级困难地图上提升明显,充分证明了 UCA 算法的有效性。未来值得进一步探索的问题包括在存在大量噪声的环境中以及更为复杂的异步异构的多智能体协同任务[41]上实现合理的信用分配等。

参考文献

[1] Bhalla S, Ganapathi Subramanian S, Crowley M. Deep multi agent reinforcement learning for autonomous driving[M]// Advances in Artificial Intelligence. Cham: Springer International Publishing, 2020: 67-78. DOI: 10.1007/978-3-030-47358-7_7.

[2] Ye D Y, Zhang M J, Yang Y. A multi-agent framework for packet routing in wireless sensor networks[J]. Sensors

(Basel Switzerland), 2015, 15(5): 10026-10047. DOI: 10.3390/s150510026.

[3] Hüttenrauch M, Šošić A, Neumann G. Guided deep reinforcement learning for swarm systems [EB/OL]. arXiv: 1709.06011 (2017-09-18) [2022-04-15]. https://arxiv.org/abs/1709.06011.

[4] Berner C, Brockman G, Chan B, et al. Dota 2 with large scale deep reinforcement learning [EB/OL]. arXiv: 1912.06680 (2019-12-13) [2022-04-15]. https://arxiv.org/abs/1912.06680.

[5] Vinyals O, Ewalds T, Bartunov S, et al. StarCraft II: a new challenge for reinforcement learning [EB/OL]. arXiv: 1708.04782 (2017-08-16) [2022-04-15]. https://arxiv.org/abs/1708.04782.

[6] Ye D H, Chen G B, Zhang W, et al. Towards playing full moba games with deep reinforcement learning [EB/OL]. arXiv:2011.12692 (2020-12-31) [2022-04-15]. https://arxiv.org/abs/2011.12692.

[7] Gupta J K, Egorov M, Kochenderfer M. Cooperative multi-agent control using deep reinforcement learning [M] // Autonomous Agents and Multiagent Systems. Cham: Springer International Publishing, 2017: 66-83. DOI: 10.1007/978-3-319-71682-4_5.

[8] Rashid T, Samvelyan M, Witt C S D, et al. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning [EB/OL]. arXiv: 1803.11485 (2018-06-06) [2022-04-15]. https://arxiv.org/abs/1803.11485.

[9] Tan M. Multi-agent reinforcement learning: independent vs. cooperative agents[M]//Machine Learning Proceedings 1993. Amsterdam: Elsevier, 1993: 330-337. DOI:10.1016/b978-1-55860-307-3.50049-6.

[10] Du Y L, Han Lei, Fang M, et al. LIIR: learning individual intrinsic reward in multi-agent reinforcement learning[C/OL] // Advances in Neural Information Processing Systems, Cambridge, MIT Press, 2019: 4405-4416. (2021-06-15) [2022-04-18]. https://dl.acm.org/doi/10.5555/3454287.3454683.

[11] Foerster J, Farquhar G, Afouras T, et al. Counterfactual multi-agent policy gradients [EB/OL]. arXiv: 1705.08926 (2017-12-14) [2022-04-15]. https://arxiv.org/abs/1705.08926.

[12] Kraemer L, Banerjee B. Multi-agent reinforcement learning as a rehearsal for decentralized planning[J]. Neurocomputing, 2016, 190: 82-94. DOI:10.1016/j.neucom.2016.01.031.

[13] Mahajan A, Rashid T, Samvelyan M, et al. MAVEN: multi-agent variational exploration [EB/OL]. arXiv: 1910.07483v2 (2020-01-20) [2022-04-15]. https://arxiv.org/abs/1910.07483v2.

[14] Oliehoek F A, Spaan M T J, Vlassis N. Optimal and approximate q-value functions for decentralized POMDPs[J]. Journal of Artificial Intelligence Research, 2008, 32: 289-353. DOI:10.1613/jair.2447.

[15] Wang S C, Li B. Implicit posterior sampling reinforcement learning for continuous control [M] // Neural Information Processing. Cham: Springer International Publishing, 2020: 452-460. DOI:10.1007/978-3-030-63833-7_38.

[16] Blundell C, Cornebise J, Kavukcuoglu K, et al. Weight uncertainty in neural network [EB/OL]. arXiv: 1505.05424 (2015-05-21) [2022-04-18]. https://arxiv.org/abs/1505.05424.

[17] Krueger D, Huang C W, Islam R, et al. Bayesian hypernetworks [EB/OL]. arXiv: 1710.04759 (2018-04-24) [2022-04-15]. https://arxiv.org/abs/1710.04759.

[18] Pawlowski N, Brock A, Lee M C H, et al. Implicit weight uncertainty in neural networks [EB/OL]. arXiv: 1711.01297 (2018-05-25) [2022-04-15]. https://arxiv.org/abs/1711.01297.

[19] Oliehoek F A, Amato C. A concise introduction to decentralized POMDPs [M]. Cham: Springer International Publishing, 2016. DOI:10.1007/978-3-319-28929-8.

[20] Wolpert D H, Tumer K. Optimal payoff functions for members of collectives [M] // Modeling Complexity in Economic and Social Systems. WORLD SCIENTIFIC, 2002: 355-369. DOI:10.1142/9789812777263_0020.

[21] Sunehag P, Lever G, Gruslys A, et al. Value-decomposition networks for cooperative multi-agent learning [EB/OL]. arXiv: 1706.05296 (2017-06-16) [2022-04-15]. https://arxiv.org/abs/1706.05296.

[22] Wang J H, Zhang Y, Kim T K, et al. Shapley q-value: a local reward approach to solve global reward games [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(5): 7285-7292. DOI: 10.1609/aaai.v34i05.6220.

[23] Chalkiadakis G, Elkind E, Wooldridge M. Computational aspects of cooperative game theory[J]. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2011, 5(6): 1-168. DOI:10.2200/s00355ed1v01y201107aim016.

[24] Shapely L S. A value for n-person games [J]. Annals of mathematics studies, 1953, 2: 307-318. DOI: 10.7249/P0295.

[25] Son K, Kim D, Kang W J, et al. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning [EB/OL]. arXiv: 1905.05408v1 (2019-05-14) [2022-04-15]. https://arxiv.org/abs/1905.05408v1.

[26] Zhou M, Liu Z Y, Sui P W, et al. Learning implicit credit assignment for cooperative multi-agent reinforcement learning [EB/OL]. arXiv: 2007.02529 (2020-10-22) [2022-04-15]. https://arxiv.org/abs/2007.02529.

[27] Wu Z F, Yu C, Ye D H, et al. Coordinated proximal policy optimization [EB/OL]. arXiv: 2111.04051 (2021-11-07) [2022-04-15]. https://arxiv.org/abs/2111.04051.

[28] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy

optimization algorithms [EB/OL]. arXiv: 1707.06347 (2017-08-28) [2022-04-15]. https://arxiv.org/abs/1707.06347.

[29] Shao J Z, Zhang H C, Jiang Y C, et al. Credit assignment with meta-policy gradient for multi-agent reinforcement learning [EB/OL]. arXiv: 2102.12957 (2021-02-24) [2022-04-15]. https://arxiv.org/abs/2102.12957.

[30] Xu Z W, Hasselt H, Silver D. Meta-gradient reinforcement learning[C/OL]//Advances in International Conference on Neural Information Processing Systems, Cambridge, MIT Press, 2018: 2396-2407. (2018-12-03) [2022-04-18]. https://dl.acm.org/doi/10.5555/3327144.3327166.

[31] Xu Z W, Li D P, Bai Y P, et al. MMD-MIX: value function factorisation with maximum mean discrepancy for cooperative multi-agent reinforcement learning[C]//2021 International Joint Conference on Neural Networks (IJCNN). July 18-22, 2021, Shenzhen, China. IEEE, 2021: 1-7. DOI: 10.1109/IJCNN52387.2021.9533636.

[32] Bellemare M G, Dabney W, Munos R. A distributional perspective on reinforcement learning [EB/OL]. arXiv: 1707.06887 (2017-07-21) [2022-04-15]. https://arxiv.org/abs/1707.06887.

[33] Ahn H, Lee D, Cha S, et al. Uncertainty-based continual learning with adaptive regularization [EB/OL]. arXiv: 1905.11614 (2019-11-14) [2022-04-18]. https://arxiv.org/abs/1905.11614.

[34] Lipton Z C, Li X J, Gao J F, et al. BBQ-networks: efficient exploration in deep reinforcement learning for task-oriented dialogue systems [EB/OL]. arXiv: 1608.05081 (2017-11-23) [2022-04-15]. https://arxiv.org/abs/1608.05081.

[35] Hinton G E, van Camp D. Keeping the neural networks simple by minimizing the description length of the weights[C]//Proceedings of the sixth annual conference on Computational learning theory - COLT '93. July 26-28, 1993. Santa Cruz, California, USA. New York: ACM Press, 1993: 5-13. DOI: 10.1145/168304.168306.

[36] Moerland T M, Broekens J, Jonker C. Efficient exploration with double uncertain value networks [EB/OL]. arXiv: 1711.10789 (2017-11-29) [2022-04-15]. https://arxiv.org/abs/1711.10789.

[37] Fortunato M, Azar M G, Piot B, et al. Noisy networks for exploration [EB/OL]. arXiv: 1706.10295 (2019-07-09) [2022-04-15]. https://arxiv.org/abs/1706.10295.

[38] Jiang B, Xu T Y, Wong W H. Approximate bayesian computation with kullback-leibler divergence as data discrepancy[C/OL]//Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. JMLR: Volume 84. 2018: 1711-1721. (2018-03-31) [2022-04-18]. http://proceedings.mlr.press/v84/jiang18a/jiang18a.pdf.

[39] Samvelyan M, Rashid T, de Witt C S, et al. The StarCraft multi-agent challenge [EB/OL]. arXiv: 1902.04043 (2019-12-09) [2022-04-18]. https://arxiv.org/abs/1902.04043.

[40] Hu J, Wu H, Harding S A, et al. RIIT: Rethinking the importance of implementation tricks in multi-agent reinforcement learning [EB/OL]. arXiv: 2102.03479 (2022-01-01) [2022-04-15]. https://arxiv.org/abs/2102.03479.

[41] Yao M, Yin Q Y, Yu T T, et al. The partially observable asynchronous multi-agent cooperation challenge [EB/OL]//arXiv: 2112.03809 (2021-12-07) [2022-04-15]. https://arxiv.org/abs/2112.03809.