计算机科学
COMPUTER SCIENCE

# 基于状态估计的值分解方法

熊丽琴, 曹雷, 陈希亮, 赖俊

引用本文

熊丽琴, 曹雷, 陈希亮, 赖俊.基于状态估计的值分解方法[J]. 计算机科学, 2023, 50(8): 202-208.

XIONG Liqin, CAO Lei, CHEN Xiliang, LAI Jun. Value Factorization Method Based on State Estimation [J]. Computer Science, 2023, 50(8): 202-208.

---

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220500270

# 基于状态估计的值分解方法

熊丽琴　曹　雷　陈希亮　赖　俊
陆军工程大学指挥控制工程学院　南京 210007
(x18779557924@126.com)

**摘　要**　值分解方法是一种流行的解决合作多智能体深度强化学习问题的方法,其核心是基于IGM(Individual-Global-Max)原则将联合值函数表示为个体值函数的某种组合。该方法中,智能体仅根据基于局部观察的个体值函数选择动作,这导致智能体无法有效地利用全局状态信息学习策略。尽管许多值分解算法已经采用了注意力机制、超网络等手段来提取全局状态的特征以加权个体值函数,从而间接地利用全局信息来指导智能体训练,但这种利用非常有限。在复杂环境中,智能体仍旧难以学到有效策略,学习效率较差。为提高智能体策略学习能力,提出了一种基于状态估计的多智能体深度强化学习值分解方法——SE-VF(Value Factorization based on State Estimation),该方法引入状态估计网络来提取全局状态的特征并得到评估全局状态优劣的状态值,然后将状态损失值作为损失函数的一部分来更新智能体网络的参数,从而优化智能体的策略选择过程。实验结果表明,在星际争霸2微观管理任务测试平台的多个场景中,SE-VF的表现比QMIX等基线更好。

**关键词**:状态估计;值分解;多智能体强化学习;深度强化学习
**中图法分类号**　TP181

## Value Factorization Method Based on State Estimation

XIONG Liqin,CAO Lei,CHEN Xiliang and LAI Jun
College of Command and Control Engineering,Army Engineering University,Nanjing 210007,China

**Abstract**　Value factorization is a popular method to solve cooperative multi-agent deep reinforcement learning problems,which factorizes joint value function into individual value functions according to IGM principle. In this method,agents select actions only according to individual value functions based on local observation,which leads to agents cannot effectively use global information to learn strategy. Although many value factorization algorithms extract the features of global state to weight individual value functions by many approaches,including attention mechanism,super network,and et al,so as to indirectly utilize global information to train agents,but this utilization is pretty limited. In a complex environment,it is difficult for agents to learn effective strategies and their learning efficiency is poor. In order to improve agents' policy learning ability,an optimized value factorization method based on state estimation(SE-VF) is put forward,which introduces a state network to extract the features of global state and get a state value,and then take state loss value as part of the loss function to update agents network parameters,so as to optimize the strategy selection process of agents. Experimental results show that SE-VF performs better than QMIX and other baselines in multiple scenarios of the StarCraft 2 micromanagement mission test platform.

**Keywords**　State estimation,Value factorization,Multi-agent reinforcement learning,Deep reinforcement learning

## 1　引言

近年来,随着智能化技术的发展,深度强化学习(Deep Reinforcement Learning,DRL)取得了重大突破。从AlphaGo[1]打败围棋世界冠军李世石到AlphaStar[2]玩转星际争霸2达到大师级水平,都展示出深度强化学习的强大能力。然而,在多智能体环境中,深度强化学习面临着巨大挑战[3]。为更好地应对这些挑战,多智能体深度强化学习(Multi-Agent Deep Reinforcement Learning,MADRL)应运而生。

MADRL结合了多智能体系统(Multi-Agent System,MAS)和DRL,旨在有效解决多智能体协同问题[4]。合作场景下的多智能体协同通常涉及不同数量/类型的智能体的分工合作,解决该问题的一种典型的MADRL方法是将全局共享的联合值函数分解为基于个体观察的个体值函数,并利用联合值函数计算误差来训练网络,从而使智能体学到一个最优联合动作策略。这类方法被称为值分解方法

到稿日期:2022-05-30　返修日期:2022-09-05
基金项目:国家自然科学基金(61806221)
This work was supported by the National Natural Science Foundation of China(61806221).
通信作者:曹雷(caolei_nj2022@126.com)

(Value Factorization，VF)。

VF的训练模式为集中训练与分散执行(Centralized Training and Decentralized Execution，CTDE)[5-6]，其核心是利用全局共享的联合值函数优化智能体网络，并根据基于单个智能体局部观察和动作的个体值函数贪婪地选择动作。相比完全集中训练的方法[7]，值分解方法有效缓解了环境的部分可观测和状态-动作空间维度爆炸问题，具有较强的可扩展性。与独立学习[8]的方法相比，它又能很好地应对环境的非平稳性以及多智能体协作问题，具有更好的稳定性。由于具有这些显著优势，值分解方法颇受研究者们的关注。

VDN(Value Decomposition Networks)是最早被提出的值分解方法，它将联合值函数表示为等于个体值函数的简单和[5]。该方法完全没有利用全局状态信息，因此在复杂场景中表现较差。而后，Rashid 等[6]提出了一种改进算法——QMIX，它利用带有全局信息的权重网络得到一组非负权重，并将该权重向量与个体值函数混合得到联合值函数。QMIX间接地利用了全局状态信息来指导智能体选择动作，在星际争霸2环境中的表现非常突出。WQMIX[9]，AIQMIX[10]，DQMIX[11]和SMIX[12]等是基于QMIX提出的优化算法，它们以与QMIX相同的方式利用全局状态信息来训练智能体。QTRAN[13]，QTRAN++[14]等根据基于全局状态信息的权重向量计算出真实联合动作值，并利用易分解的联合动作值来模仿真实联合动作值以优化智能体的策略。Qatten[15]，AVD-Net[16]，QPLEX[17]和REFIL[18]等采用注意力机制[19]来提取全局状态信息的特征，并得到一组权重向量，从而提高了联合值函数的表征能力。MMD-MIX[20]提出了一种以全局状态信息为输入的MMD(Maximum Mean Discrepancy)混合网络，该网络将分布式强化学习和值分解方法相结合，以适应情况的随机性。此外，研究者们还通过引入通信机制使智能体之间实现信息共享，从而得到更完整的状态信息来训练网络[21-23]。这些方法在避免出现动作空间维度爆炸问题的同时均尽可能地获取并利用了全局状态信息，充分体现了训练时有效利用全局状态信息对实现多智能体协同的重要性。

然而，上述方法基本都是通过提取全局状态信息的特征来加权个体值函数，对全局状态信息的利用比较单一、有限。为此，本文提出了一种基于状态估计的值分解方法SE-VF。该方法采用状态网络对访问过的状态进行近似估计以获得对应的状态值，并计算状态值误差来优化全局损失函数以稳定训练，为解决全局状态信息利用问题提供了一种新视角。在具有挑战性的星际争霸2微观管理任务平台上对SE-VF进行评估，实验结果表明，所提方法在多个场景中的表现均优于QMIX等基线方法。

## 2 多智能体深度强化学习值分解方法

本节主要介绍了多智能体深度强化学习值分解方法的建模方法、训练模式和经典算法。

### 2.1 去中心化部分可观测马尔可夫决策过程

完全合作的多智能体强化学习任务可以由分布式部分可观测马尔可夫决策过程(Decentralized Partially Observable Markov Decision Processes，Dec-POMDP)来描述，其形式化表示为 $G=\langle n,S,U,P,r,Z,O,\gamma\rangle$[24]。其中，$n$ 表示初始时智能体的个数；$s\in S$ 描述了环境的真实状态。在每个时间步内，每个智能体 $i$ 都选择一个动作 $u^i\in U$，从而形成联合动作集 $u\in U\equiv U^n$。当智能体在状态 $s$ 下执行联合动作 $u$ 时，环境将依据状态转移概率函数 $P(s'|s,u|)$：$S\times U\times S\to[0,1]$转移到状态 $s'$，并给所有智能体返回一个全局共享的奖励函数 $r(s,u)$：$S\times U\to\Re$。在训练过程中，每个智能体都有自己的观察值 $z\in Z$，由各自的观测函数 $O(s,u)$：$S\times U\to Z$ 来刻画；$\gamma\in(0,1)$是折扣因子。

在每个时间步内，每个智能体 $i$ 都有一个观察历史 $h^i\in H\equiv(Z\times U)^*$，并基于此生成随机策略 $\pi^i(u^i|h^i|)$：$H\times U\to[0,1]$，从而形成联合观察历史 $h$ 和联合策略 $\pi$。在联合策略的指导下，可以得到联合动作值函数 $Q^\pi(s_t,u_t)=E_{s_{t+1,\infty},u_{t+1,\infty}}[R_t|s_t,u_t|]$，其中，$R_t=\sum_{i=0}^{\infty}\gamma^i r_{t+i}$ 为累积折扣回报。

### 2.2 集中训练与分散执行(CTDE)

CTDE是一种流行的解决合作多智能体深度强化学习问题的训练模式[5-6]，它能够有效解决IQL(Independent Q Learning)和完全集中式学习无法解决的多智能体任务。

集中训练：训练期间，智能体之间的通信不受约束，可以访问其他智能体的动作观察历史 $h$ 和全局状态 $s$，并利用联合值函数 $Q^\pi(s_t,u_t)$训练智能体。

分散执行：执行期间，智能体间通信受限，每个智能体仅能访问自己的观察历史 $h^i$，并根据个体值函数 $Q^i(h^i,u^i)$选择动作。

### 2.3 价值分解网络(VDN)

VDN算法是由 Sunehag 等[5]提出的一种个体学习的可加值分解方法，其核心是将联合动作值函数 $Q_{\text{tot}}(h,u)$表示为个体动作值函数 $Q^i$ 的线性和，即：

$$Q_{\text{tot}}(h,u)=Q_{\text{tot}}((h^1,h^2,\cdots,h^n),(u^1,u^2,\cdots,u^n))=\sum_{i=1}^{n}Q^i(h^i,u^i;\theta^i)\tag{1}$$

价值分解网络的关键是利用表示个体动作值函数的深度神经网络反向传播联合动作值函数 $Q_{\text{tot}}(h,u)$的梯度，从联合奖励中学习最优线性值分解。在训练过程中，每个智能体都基于自己的局部观察得到 $Q^i$，并采用 $\varepsilon$-greedy 算法贪婪地选择动作，从而产生一个分散策略。然后，利用DQN的更新规则将 $Q(s,u;\theta)$替换成 $Q_{\text{tot}}(h,u)$，得到新的损失函数 $L_{\text{VDN}}(\theta)$，并通过最小化损失函数更新网络参数。$L_{\text{VDN}}(\theta)$定义为：

$$L_{\text{VDN}}(\theta)=\sum_{i=1}^{b}[(y_i^{\text{tot}}-Q_{\text{tot}}(h,u))^2]\tag{2}$$

### 2.4 QMIX

QMIX是基于VDN改进的一种经典值分解方法，其核心是将联合动作值函数 $Q_{\text{tot}}(h,u)$表示为仅基于局部观察的个体动作值函数 $Q^i$ 的复杂非线性组合。它通过对 $Q_{\text{tot}}(h,u)$和 $Q^i$ 实施单调性约束，使在 $Q_{\text{tot}}(h,u)$上进行 argmax 操作与在每个 $Q^i$ 上产生的动作一致，即当

$$\frac{\partial Q_{\text{tot}}(h,u)}{\partial Q^i}\geqslant 0,\forall i\in\{1,2,\cdots,n\}|\tag{3}$$

$$\frac{\partial Q_{\text{tot}}(h,u)}{\partial Q^i}\geqslant 0,\ \forall i\in\{1,2,\cdots,n\}| \tag{4}$$

有

$$\arg\max_u Q_{\text{tot}}(h,u)=\begin{pmatrix}\arg\max_{u^1} Q^1\\ \cdots\\ \arg\max_{u^n} Q^n\end{pmatrix} \tag{5}$$

QMIX 采用了 3 种神经网络：智能体网络、超网络和混合网络，整体架构如图 1 所示。智能体网络由 DRQN[25] 表示，以智能体 $i$ 的局部观察 $o_t^i$ 和当前时间步接收的最后一个动作 $u_{t-1}^i$ 为输入计算得到 $Q^i(h^i,u^i)$。超网络由一个单一线性层和绝对值激活函数组成，其将全局状态信息依次通过线性层和绝对值激活函数得到一组非负权重 $w$。混合网络是一个前馈神经网络，其将 $Q^i(h^i,u^i)$ 和非负权重 $w$ 混合得到 $Q_{\text{tot}}(h,u)$。

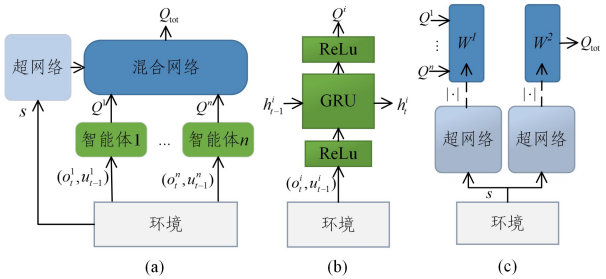



图 1 QMIX 整体架构图

Fig. 1 Overall architecture of QMIX

QMIX 通过最小化损失函数来训练智能体模型，其损失函数为：

$$L_{\text{QMIX}}(\theta)=\sum_{i=1}^{b}[(y_i^{\text{tot}}-Q_{\text{tot}}(h,u,s;\theta))^2] \tag{6}$$

其中，$b$ 为采样大小，$y^{\text{tot}}=r+\gamma\max Q_{\text{tot}}(h',u',s';\theta^-)$，$\theta^-$ 为目标网络参数。

QMIX 作为一种经典的值分解方法，在实验测试中取得了较好的效果。然而，由于具有单调性结构约束，QMIX 无法解决在非单调环境下求解最优策略的问题。因此，研究者们提出了一种加权QMIX 算法 WQMIX[9]（Weighted QMIX），在更新网络时引入权重对联合动作值的平方误差进行加权以得到最优联合值函数。该算法提出了两种加权方式，即理想中心加权（CWQMIX）和乐观加权（OWQMIX），权重函数定义如下：

$$(CWQMIX)\ w(s,u)=\begin{cases}1, & u=\arg\max_u Q(s,u)\\ \alpha, & \text{otherwise}\end{cases} \tag{7}$$

$$(OWQMIX)\ w(s,u)=\begin{cases}1, & Q_{\text{tot}}(s,u)<Q(s,u)\\ \alpha, & \text{otherwise}\end{cases} \tag{8}$$

则 WQMIX 的损失函数定义为：

$$L_{\text{WQMIX}}(\theta)=\sum_{i=1}^{b}[w(s,u)(y_i^{\text{tot}}-Q_{\text{tot}}(h,u))^2] \tag{9}$$

## 3 值分解方法中的状态估计

本节主要提出了一种基于状态估计的值分解方法——SE-VF，该方法旨在通过优化更新智能体网络的全局损失函数来提高智能体的策略学习能力。

### 3.1 状态估计方法

众所周知，多智能体深度强化学习值分解方法是一种有效的解决多智能体协同问题的方法。在训练过程中，智能体利用全局共享的联合值函数来训练智能体网络，并根据基于局部观察的个体值函数选择动作。这导致智能体无法有效地利用全局状态信息来学习策略。然而，在某些时刻，策略学习与智能体选择的动作无关，只和当前的全局状态有关。例如，在 MMM2 中，当攻击型智能体全被杀死时，治疗型智能体无论选择什么动作都无法避免最终的失败。因此，在训练时有效利用全局状态信息非常重要。尽管许多值分解算法都通过提取全局状态的特征来加权个体值函数，从而间接地利用全局信息来指导智能体学习，但这种利用非常有限。在复杂环境中，智能体仍然难以学到有效策略，学习效率较差。

为提高智能体的策略学习能力，本文提出了一种优化的多智能体深度强化学习值分解方法。该方法在原有值分解方法的基础上引入了一种状态估计网络，通过提取全局状态特征得到一个状态值以评估全局状态，并将状态损失值作为算法损失函数的一部分来更新智能体网络，从而优化指导智能体的策略选择。算法的整体框架如图 2 所示。

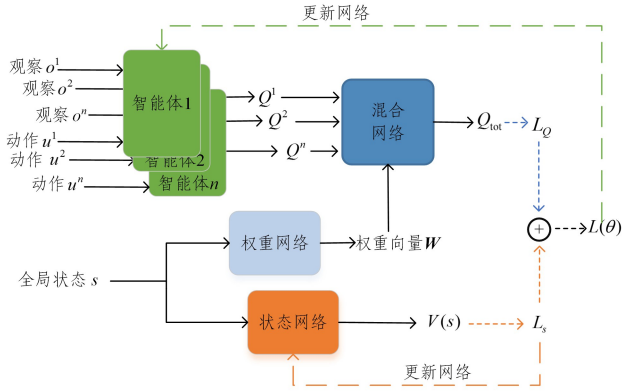



图 2 SE-VF 整体框架图

Fig. 2 Overall architecture of SE-VF

SE-VF 算法的整体架构包括 4 种网络：智能体网络、权重网络、混合网络以及状态网络。首先，智能体网络基于局部观察和上一步的动作得到个体值函数，权重网络基于全局状态得到权重向量，状态网络基于全局状态得到状态值函数。然后混合网络将个体值函数和权重向量混合得到联合值函数。基于联合值函数和状态值函数分别计算联合值损失 $L_Q$ 和状态值损失 $L_S$，从而得到全局损失函数 $L(\theta)$。最后分别利用 $L_S$ 和 $L(\theta)$更新状态网络参数和智能体网络参数。

智能体网络以智能体的局部观察和上一步的动作为输入，然后陆续通过线性激活函数、GRU 网络以及线性激活函数以得到个体值函数。它利用优化的全局损失函数 $L(\theta)$进行反向策略梯度迭代来更新网络参数。权重网络是按照原有的值分解算法得到权重向量，如 QMIX 中的超网络、Qatten 中的注意力机制等。混合网络将权重向量和个体值函数混合得到联合值函数。状态网络的结构设计与智能体网络类似，它以全局状态为输入，陆续通过线性激活函数、GRU 网络、线性激活函数以获取评估全局状态优劣的预测状态值，而后

根据状态损失函数 $L_s(\theta_s)$更新状态网络参数。

### 3.2 损失函数

SE-VF 的实现有两个关键点：一是利用由状态网络得到的状态值来评估状态，并使用状态损失函数更新状态网络参数；二是利用基于状态估计的全局损失函数来更新智能体网络参数，该全局损失函数由状态损失函数和联合值损失函数构成。本小节主要对这 3 种损失函数进行介绍。

状态损失函数 $L_s(\theta_s)$为预测状态值和目标状态值的均方差，用于更新状态网络参数，定义如下：

$$L_s(\theta_s)=\sum_{i=1}^{b}[(y_i^s-V_s(s;\theta_s))^2] \tag{10}$$

其中，$V_s(s;\theta_s)$表示通过状态网络得到的预测状态值，$y_i^s$ 表示状态网络的目标状态值，其定义如下：

$$y_i^s=r+\gamma V^*(s') \tag{11}$$

其中，$V^*(s')$表示在状态 $s'$下的最优状态值。

然而，由于全局状态维度较大，我们难以直接通过网络来拟合所有全局状态的最优状态值。因此，本文利用状态值函数 $V_\pi(s)$和状态动作值函数 $Q_\pi(s,u)$之间的关系来间接求解最优状态值。

在强化学习建模过程中，状态值函数 $V_\pi(s)$和状态动作值函数 $Q_\pi(s,u)$的定义如下：

$$V_\pi(s)=E_\pi[\sum_{t=0}^{\infty}\gamma^t r_{t+1}\mid s=s_t\mid] \tag{12}$$

$$Q_\pi(s,u)=E_\pi[\sum_{t=0}^{\infty}\gamma^t r_{t+1}\mid s=s_t,u=u_t\mid] \tag{13}$$

则状态值函数 $V_\pi(s)$与状态动作值函数 $Q_\pi(s,u)$之间的关系可表示为：

$$V_\pi(s)=E_{\pi\sim u}[Q_\pi(s,u)] \tag{14}$$

由此可知，最优状态值函数 $V^*(s)$与最优状态动作值函数 $Q_\pi^*(s,u)$的关系式为：

$$\begin{aligned}V^*(s)&=\max E_{\pi\sim u}[Q_\pi(s,u)]\\&=\max_u Q_\pi(s,u)\\&=Q_\pi^*(s,u)\\&\approx Q^*(h,u)\end{aligned} \tag{15}$$

因此，我们可以将状态网络的目标函数 $y_i^s$ 重新定义为：

$$y_i^s=r+\gamma V^*(s')=r+\gamma\max_{u'}Q_{\text{tot}}(h',u',s';\theta^-) \tag{16}$$

联合值损失函数 $L_Q(\theta_Q)$为预测联合动作值与目标联合动作值的均方差。在原始算法中，该损失函数被直接用于更新智能体网络参数。但在本文的方法中，该损失函数仅作为用于更新智能体网络的全局损失函数的一项，其定义如下：

$$L_Q(\theta_Q)=\sum_{i=1}^{b}[(y_i^{\text{tot}}-Q_{\text{tot}}(h,u,s;\theta_Q))^2] \tag{17}$$

其中，$y^{\text{tot}}=r+\gamma\max_{u'}Q_{\text{tot}}(h',u',s';\theta_Q^-)$，$\theta_Q^-$ 表示智能体目标网络的参数。

全局损失函数 $L(\theta)$由联合值损失函数 $L_Q(\theta_Q)$和状态损失函数 $L_s(\theta_s)$得到，用于更新智能体网络参数，从而指导智能体选择策略，其定义如下：

$$L(\theta)=L_Q(\theta_Q)+\lambda L_s(\theta_s) \tag{18}$$

其中，$\lambda$ 为加权系数。

全局损失函数不仅包含了联合值函数的估计误差，还考虑了状态值函数的估计误差。这种额外的考虑是有效且必要的，它从另一个角度间接利用全局状态信息来指导智能体选择策略，有利于智能体收敛到最优状态。尽管状态值函数$V(s)$通常可以由联合动作值函数 $Q_{\text{tot}}(s,u)$得到，但联合值函数是由基于局部观察的个体值函数加权得到，其确切表示应该为 $Q_{\text{tot}}(h,u)$。注意，$Q_{\text{tot}}(s,u)\approx Q_{\text{tot}}(h,u)$，这种近似在某些时刻可能会存在较大的误差。因此，引入状态网络以准确评估全局状态的优劣是必要的。算法的伪代码如算法 1 所示。

**算法 1** 基于状态估计的值分解方法

初始化经验池 D、智能体网络权重参数 $\theta_Q$、目标网络权重参数 $\theta_Q^-=\theta_Q$ 以及状态网络参数 $\theta_s$

1. For episode=1～M do：
2. For t=1～limited step do：
3. For i=1～N do：
4. 智能体 i 根据 $Q_{t-1}^i(o_{t-1}^i,u_{t-1}^i,\theta_Q)$贪婪地选择动作 $u_t^i$
5. 得到联合动作 $u_t$
6. End For
7. 执行联合动作 $u_t$ 后转移到下一步状态 $s_{t+1}$，并获得环境奖励 $r_t$
8. 存储经验样本$(s_t,o_t,u_t,r_t,s_{t+1})$到经验池 D 中
9. 从经验池 D 中随机采样小批量存储样本$(s_t,o_t,u_t,r_t,s_{t+1})$
10. 根据样本得到联合值函数 $Q_{\text{tot}}(h,u)$和状态值函数 V(s)
11. 由式(10)、式(17)和式(18)分别计算状态损失函数 $L_s(\theta_s)$、联合值损失函数 $L_Q(\theta_Q)$和全局损失函数 $L(\theta)$
12. 使用梯度下降法更新每个智能体的网络模型参数 $\theta_Q$ 和状态网络参数 $\theta_s$
13. 每隔 K 步更新智能体目标网络参数 $\theta_Q^-=\theta_Q$
14. End For
15. End For

## 4 实验

### 4.1 设置

本节以 StarCraft Multi-Agent Challenge(SMAC)环境为测试平台评估 SE-VF 在星际争霸 2 微观管理任务中的性能，并与多种经典的值分解方法(如 QMIX，VDN，IQL 等)进行对比分析。训练过程中，我方单元由 SE-VF 训练的智能体控制，敌方单元由内置的 AI 控制。所有的训练参数均与 SMAC 中的 QMIX 相同。

SE-VF 主要包括 SE-VDN 和 SE-QMIX，它们分别针对 VDN 和 QMIX 而提出。分别在 4 个地图(8m，5m_vs_6m，3s_vs_5z 和 MMM2)上对 SE-VDN 和 SE-QMIX 进行评估，并与 IQL，VDN，QMIX 等基线方法进行对比分析。其中，8m 属于简单场景，5m_vs_6m 和 3s_vs_5z 属于困难场景，MMM2 属于超难场景。地图的特点如表 1 所列。

表 1 测试地图特点分析

Table 1 Features of 8m，5m_vs_6m，3s_vs_5z and MMM2

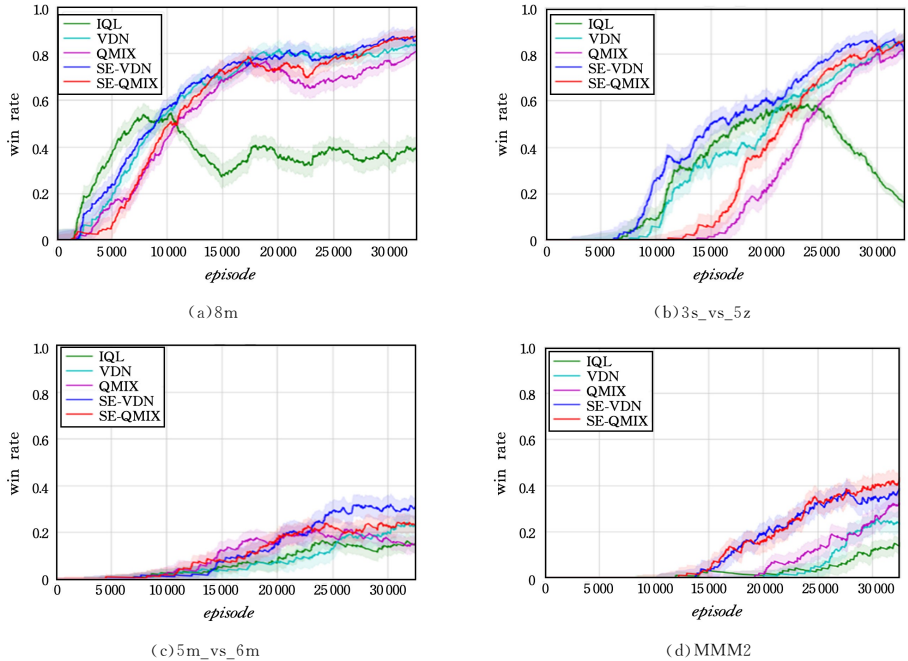
| 地图名 | 难易程度 | 作战单元组成 | 对抗模式 |
|---|---|---|---|
| 8m | 简单 | 同构 | 对称 |
| 3s_vs_5z | 难 | 同构 | 非对称 |
| 5m_vs_6m | 难 | 同构 | 非对称 |
| MMM2 | 超难 | 异构 | 非对称 |

### 4.2 验证实验

在本组实验中，SE-VF 中状态损失函数的权重 $\lambda=1$，其他参数均与 QMIX 保持一致。

首先，在地图 8m 上评估了 SE-QMIX 和 SE-VDN 的性能，训练结果如图 3(a)所示。实验结果表明，SE-VF 可以很好地完成任务，并且最终保持较高的胜率。从胜率曲线中还可以发现，无论是最终胜率还是稳定性，QMIX 在简单场景中的表现都略差于 VDN。在引入状态网络之后，SE-QMIX 的性能虽然较 QMIX 有所提升，但依旧略差于 VDN，而 SE-VDN 较 VDN 也并没有明显的提升。这主要是因为在简单场景中智能体不需要过多考虑全局状态信息就可以有效学习策略。IQL 没有考虑到多智能体强化学习中存在的环境非平稳性问题，导致算法的稳定性较差，在训练过程中出现越学越差的情况。

而后，分别在地图 3s_vs_5z 和 5m_vs_6m 上评估了 SE-QMIX 和 SE-VDN 的性能，训练结果如图 3(b)－图 3(c)所示。3s_vs_5z 是一种同构不对称的困难场景，我方智能体要想取胜，必须学会与敌人保持距离以分散敌方兵力，然后包夹敌人并逐个消灭它们的策略。这对智能体的协同能力提出了较高要求。实验结果显示，SE-VF 在不对称困难场景中具有较好的表现，其性能较原始算法均有较大提升。SE-QMIX 的学习效率明显比 QMIX 更好，SE-VDN 的学习效率也比 VDN 明显要好。这恰好说明通过引入状态网络来利用全局状态信息以优化策略选择过程是可行且有效的。5m_vs_6m 也是一种异构不对称的困难场景，在该场景中我方不存在兵种优势，并且需要以少胜多，因此这是一个具有挑战性的场景。实验结果显示，SE-VDN 的性能较 VDN 有明显提升，而 SE-QMIX 在前期与 QMIX 没有明显区别，但后期的胜率明显比 QMIX 高。在地图 5m_vs_6m 中，QMIX 前期的表现略优于 VDN，但后期的训练不稳定，出现胜率下降的趋势，这说明利用基于全局状态信息的权重加权个体值函数虽然可以提升算法在复杂场景中的性能，但效果并不理想。而 SE-VDN 的训练效果明显优于 QMIX 也说明引入状态网络可以更好地利用全局状态信息。




(a)8m (b)3s_vs_5z

(c)5m_vs_6m (d)MMM2

图 3 训练过程中各算法的胜率

Fig. 3 Win rates of SE-VF and baselines during training

最后，在超难场景(MMM2)中评估了 SE-QMIX 和 SE-VDN 的性能，训练结果如图 3(d)所示。MMM2 是一种异构不对称场景，包括医疗运输机、劫掠者和陆战队员 3 种单元。在 MMM2 场景中取胜主要依赖 3 种单元的协作，即医疗运输机吸引敌方火力并治疗我方受伤单元，劫掠者突破敌人防御并收割残血单元，而陆战队员提供防御并攻击敌方医疗运输机。因此，能否充分考虑全局状态信息对赢得战斗具有很大的影响。实验结果也表明，引入状态网络后的 SE-VF 表现非常突出，其训练时的胜率远高于原始算法。除此之外，QMIX 的胜率较 VDN 略有提升，SE-QMIX 和 SE-VDN 的表现相当，其胜率均明显高于 QMIX。这正说明使用状态网络优化全局损失函数的方法比使用超网络加权个体值函数的方法更能有效地利用全局状态信息，提升算法的性能。

综上所述，无论在异构还是同构、简单还是困难的场景中，SE-VF 都可以较好地工作，其整体性能均优于基线算法；而且，相较于使用超网络以加权个体值函数的方法，使用状态网络以优化全局损失函数的方法更能有效地利用全局状态信息，优化指导智能体的策略选择，从而提高算法在复杂场景中的学习效率和适应性。

为更直观地说明各算法在所有场景中的性能，我们利用训练好的模型进行 5000 轮测试，并得到统计胜率。统计胜率指测试时胜利的轮数与总测试轮数的比值，最大为 1。测试结果如表 2 所列，对每个场景中测试胜率最高的值进行加粗表示。

表 2 SE-VF 和基线的测试胜率

Table 2 Test win rates of SE-VF and baseline

| Test Win Rate | SE-QMIX | SE-VDN | QMIX | VDN | IQL |
|---|---|---|---|---|---|
| 8m | 0.9714 | 0.9872 | 0.9890 | **0.9898** | 0.6184 |
| 3s_vs_5z | **0.9916** | 0.9828 | 0.9876 | 0.9902 | 0.1964 |
| 5m_vs_6m | 0.4998 | **0.5910** | 0.3002 | 0.5856 | 0.3076 |
| MMM2 | **0.7356** | 0.6620 | 0.7008 | 0.5050 | 0.3730 |

测试结果显示,SE-VF 几乎在所有场景中都具有最好的表现,尤其在 5m_vs_6m 和 MMM2 这两个难度较大的场景中,SE-VF 的最终测试胜率显著优于同等情况下的原始算法。

### 4.3 消融实验

本组实验将状态损失函数的权重参数 $\lambda$ 设置为不同的值,并分别在较难的场景中测试 SE-VF 的性能,以进一步分析状态网络的有效性。考虑到智能体是基于个体值函数选择动作,全局损失函数中联合值损失函数应该更重要,而状态损失函数只是作为一个辅助项,如果状态损失函数占比过大,训练可能不太稳定。因此,我们将 $\lambda$ 的值分别设置为 0,0.2,0.5,0.8 和 1.0。

首先,在超难场景 MMM2 中评估了 SE-QMIX 的性能,结果如图 4 所示。实验结果表明,随着 $\lambda$ 值的增大,算法的胜率确实有显著的提升。当 $\lambda$ 为 0.2,0.5 时,由于全局损失函数中状态损失值的比重较小,即联合值损失函数起主导作用,此时算法的表现几乎没有提升。然而,当 $\lambda$ 为 0.8,1.0 时,状态损失值占比较大,此时算法的学习效率大幅提升。而且,与 $\lambda=0.8$ 时的表现相比,$\lambda=1.0$ 时算法的表现明显更好。这说明引入状态网络确实能有效地提升算法的性能。

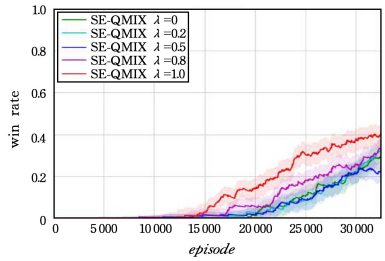



图 4 具有不同 $\lambda$ 值的 SE-QMIX 在 MMM2 中的胜率

Fig. 4 Win rates of SE-QMIX with different $\lambda$ in MMM2

而后,在地图 5m_vs_6m 中评估了 SE-VDN 的性能,结果如图 5 所示。实验结果与 SE-QMIX 在 MMM2 中的类似,随着 $\lambda$ 值的增大,算法的学习效率有明显的提升。这也进一步证明了引入状态网络以提升算法在复杂环境中的性能是可行且有效的,且适用于多种值分解算法。

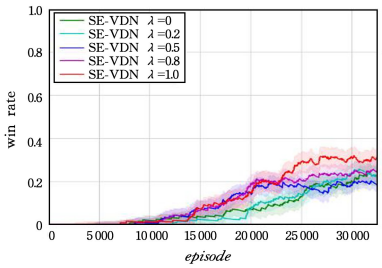



图 5 具有不同 $\lambda$ 值的 SE-VDN 在 5m_vs_6m 中的胜率

Fig. 5 Win rates of SE-VDN with different $\lambda$ in 5m_vs_6m

**结束语** 本文提出了一种基于状态估计的多智能体深度强化学习值分解方法,它为如何在训练时有效利用全局信息以提升算法性能提供了一种新的思路。首先分析了在训练时利用全局状态信息的重要性,并讨论了引入状态网络的可行性,然后利用状态网络提取全局状态信息的特征并得到对应状态值,最后根据基于状态损失值的全局损失函数来更新智能体网络参数。在 SMAC 环境中的实验结果表明,SE-VF 在多种场景中都具有较好表现,其性能均优于原始经典算法。

## 参 考 文 献

[1] SILVER D,HUANG A,MADDISON C J,et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature,2016,529(7587):484-489.

[2] MNIH V,KAVUKCUOGLU K,SILVER D,et al. Human-level control through deep reinforcement learning[J]. Nature,2015,518(7540):529-533.

[3] LI Y,XU F,XIE G Q,et al. Survey of development and application of multi-agent technology[J]. Computer Engineering and Applications,2018,54(9):13-21.

[4] SUN Y,CAO L,CHEN X L,et al. Overview of multi-agent deep reinforcement learning[J]. Computer engineering and Application,2020,56(5):13-24.

[5] SUNEHAG P,LEVER G,GRUSLYS A,et al. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based on Team Reward[C]// Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems. 2018:2085-2087.

[6] RASHID T,SAMVELYAN M,SCHROEDER C,et al. Qmix:Monotonic value function factorisation for deep multi-agent reinforcement learning[C]// International Conference on Machine Learning. 2018:4295-4304.

[7] FOERSTER J,FARQUHAR G,AFOURAS T,et al. Counterfactual multi-agent policy gradients[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2018:2974-2982.

[8] TAMPUU A,MATIISEN T,KODELJA D,et al. Multiagent cooperation and competition with deep reinforcement learning[J]. PloS one,2017,12(4):e0172395.

[9] RASHID T,FARQUHAR G,PENG B,et al. Weighted QMIX:Expanding Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning[C]// Advances in Neural Information Processing Systems. 2020:10199-10210.

[10] IQBAL S,WITT C S D,PENG B,et al. AI-QMIX:Attention and Imagination for Dynamic Multi-Agent Reinforcement Learning[J]. arXiv:2006.04222,2020.

[11] ZHAO J,YANG M,HU X,et al. DQMIX:A Distributional Perspective on Multi-Agent Reinforcement Learning[J]. arXiv:2202.10134,2022.

[12] YAO X,WEN C,WANG Y,et al. SMIX(λ):Enhancing Centralized Value Functions for Cooperative Multi-Agent Reinforcement Learning[J]. IEEE Transactions on Neural Networks and Learning Systems,2021,6:1-12.

[13] SON K,KIM D,KANG W J,et al. Qtran:Learning to factorize

with transformation for cooperative multi-agent reinforcement learning[C]// International Conference on Machine Learning. 2019:5887-5896.

[14] SON K, AHN S, REYES R D, et al. QTRAN++: Improved Value Transformation for Cooperative Multi-Agent Reinforcement Learning[J]. arXiv:2006.12010, 2020.

[15] YANG Y, HAO J, LIAO B, et al. Qatten: A general framework for cooperative multiagent reinforcement learning[J]. arXiv: 2002.03939, 2020.

[16] ZHANG Y, MA H, WANG Y. AVD-Net: Attention Value Decomposition Network For Deep Multi-Agent Reinforcement Learning[C]// 2020 25th International Conference on Pattern Recognition(ICPR). 2021:7810-7816.

[17] WANG J, REN Z, LIU T, et al. QPLEX: Duplex Dueling Multi-Agent Q-Learning[J]. arXiv:2008.01062, 2020.

[18] IQBAL S, DE WITT C A S, PENG B, et al. Randomized Entity-wise Factorization for Multi-Agent Reinforcement Learning [C]// International Conference on Machine Learning. 2021: 4596-4606.

[19] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]// Advances in Neural Information Processing Systems. 2017:5998-6008.

[20] XU Z, LI D, BAI Y, 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). 2021:1-7.

[21] FOERSTER J N, ASSAEL Y M, DE FREITAS N, et al. Learning to communicate with Deep multi-agent reinforcement learning[C]// Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016:2145-2153.

[22] WU B, YANG X, SUN C, et al. Learning Effective Value Function Factorization via Attentional Communication[C]// 2020 IEEE International Conference on Systems, Man, and Cybernetics(SMC). 2020:629-634.

[23] ZHOU H, LAN T, AGGARWAL V. Value Functions Factorization with Latent State Information Sharing in Decentralized Multi-Agent Policy Gradients[J]. arXiv:2201.01247, 2022.

[24] OLIEHOEK F A, SPAAN M T, VLASSIS N. Optimal and Approximate Q-value Functions for Decentralized POMDPs[J]. Journal of Artificial Intelligence Research, 2008, 32:289-353.

[25] HAUSKNECHT M, STONE P. Deep recurrent Q-learning for partially observable mdps[C]// 2015 AAAI Fall Symposium Series. 2015:29-37.

**XIONG Liqin**, born in 1997, postgraduate. Her main research interests include multi-agent deep reinforcement and intelligent command and control.

**CAO Lei**, born in 1965, Ph.D, professor, Ph.D supervisor. His main research interests include machine learning, command information system and intelligent decision making.

(责任编辑:何杨)