Computer Science and Application 计算机科学与应用, 2024, 14(4), 201-212
Published Online April 2024 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2024.144090

Hans汉斯

基于值分布的多智能体强化学习方法

韩明志，李　宁，王　超

北京信息科技大学计算机学院，北京

收稿日期：2024年3月18日；录用日期：2024年4月16日；发布日期：2024年4月23日

摘　要

近年来，多智能体强化学习随着深度学习技术的发展和算法研究的深入，成为人工智能领域的研究热点。特别是在处理复杂的决策问题和环境中，多智能体系统展现出其独特的优势。本文介绍了一种基于值分布的多智能体强化学习算法，旨在通过改进算法结构和学习机制，提升多智能体协作中的性能和稳定性。首先，本文深入分析了强化学习中的值分布概念，并探讨了其在多智能体系统中的应用挑战和潜在价值。随后，提出了CvM-MIX算法，该算法通过结合值分布强化学习和值分解技术，有效地提高了对环境随机性的适应能力，并采用了一种改进的基于权重优先级的经验回放机制，进一步优化了学习过程。通过在星际争霸II多智能体挑战赛(SMAC)平台进行的一系列实验，验证了CvM-MIX算法相较于传统算法在性能和稳定性上的优势。实验结果显示，CvM-MIX算法在多种对抗模式下均表现出更快的收敛速度和更高的胜率，尤其是在复杂场景中的表现尤为突出。

关键词

深度强化学习，多智能体强化学习，值分布

Multi-Agent Reinforcement Learning Method Based on Value Distribution

Mingzhi Han, Ning Li, Chao Wang

School of Computing, Beijing Information Science and Technology University, Beijing

Received: Mar. 18th, 2024; accepted: Apr. 16th, 2024; published: Apr. 23rd, 2024

Abstract

In recent years, multi-agent reinforcement learning has become a research hotspot in the field of artificial intelligence with the development of deep learning technology and the deepening of algorithm research. Especially in dealing with complex decision-making problems and environments, multi-agent systems demonstrate their unique advantages. This article introduces a mul-

文章引用: 韩明志, 李宁, 王超. 基于值分布的多智能体强化学习方法[J]. 计算机科学与应用, 2024, 14(4): 201-212.
DOI: 10.12677/csa.2024.144090

ti-agent reinforcement learning algorithm based on value distribution, aiming to improve the performance and stability of multi-agent collaboration by improving the algorithm structure and learning mechanism. Firstly, this article provides an in-depth analysis of the concept of value distribution in reinforcement learning, and explores its application challenges and potential value in multi-agent systems. Subsequently, the CvM MIX algorithm was proposed, which effectively improved its adaptability to environmental randomness by combining value distribution reinforcement learning and value decomposition techniques. An improved weight priority based experience replay mechanism was adopted to further optimize the learning process. Through a series of experiments conducted on the StarCraft II Multi Agent Challenge (SMAC) platform, the performance and stability advantages of the CvM MIX algorithm compared to traditional algorithms were verified. The experimental results show that the CvM MIX algorithm exhibits faster convergence speed and higher win rate in various adversarial modes, especially in complex scenes.

Keywords

Deep Reinforcement Learning, Multi-Agent Reinforcement Learning, Value Distribution

1. 引言

近年来，深度强化学习[1] (Deep Reinforcement Learning, DRL)作为深度学习[2] (Deep Learning, DL)和强化学习[3] (Reinforcement Learning, RL)的结合，在序贯决策问题[4]中发挥着越来越重要的作用。在强化学习中，智能体根据行动模式、当前状态和以及相应的反馈与环境进行实时交互。强化学习的目的是从观察状态到主体行为的映射，从而最大限度地提高从环境中获得的预期回报。而深度强化学习则是将强化学习对策略的决策能力和深度学习的感智能力相结合，利用深度神经网络的强大表征能力来解决复杂的决策问题。这种思维使得强化学习在许多应用场景中都有显著的效果，例如：雅达利游戏 2600 [5]、机器人控制[6]和纸牌游戏[7] [8]等。但随着深度学习技术的不断发展以及研究环境的不断复杂化，强化学习算法研究领域逐渐从单智能体领域转向多智能体领域，因此多智能体深度强化学习[9] (Multi-Agent Deep Reinforcement Learning, MADRL)成为逐渐成为研究热点。

当前研究人员为了获取更好的训练结果和更稳定的学习过程，提出了一系列具有分布视角的值分布强化学习算法。为了通过分布来模拟值函数，当前业界主流的解决方案是使用值分布强化学习方法[10]。值分布强化学习并没有使用期望标量的平均值，而是通过利用分类分布[10]或者分位数函数[11]来对回报上的分布进行预测。当前，值分布强化学习方法主要集中于单智能体强化学习领域，不能直接应用于基于价值的多智能体强化学习方法中。其中包含以下几个方面原因：(1) 在基于价值的 MARL 中，个体智能体的值分布 Q 值应当整合到联合动作的全局分布 Q 值当中；(2) 应确保联合动作的全局分布 Q 值与个体智能体的值分布 Q 值行为选择的单调一致性，即满足 Individual-Global-Max [12] (IGM)原则。

目前强化学习算法普遍存在两点问题。首先，强化学习算法普遍存在 Q 值高估——过估计问题。该问题首先在 Q 学习算法中被发现，它是现有大多数基于价值的强化学习算法原型。Hasselt 等证明了任何类型的估计错误都会引起向上的偏差，无论该错误是由系统噪声、值函数逼近还是其他问题引起的。由于时序差分学习中 Q 值使用后续状态的 Q 值进行更新，因此过估计偏差将通过时序差分算法进一步传播和扩大。而在后续提出的确定性环境以及基于 Actor-Critic 框架的深度确定性策略梯度方法中也都存在过

估计问题。一方面，由于真正的 Q 值是未知的，任何算法都会不可避免的引入估计偏差和方差；另一方面，值函数逼近同样不可避免造成误差。因此不确定的过估计可能会导致出现智能体的次优动作被过高估计，从而导致次优策略。其次，附加信息丢失。当前主流的值函数分解方法只专注于估计联合动作值函数的期望，而忽略了完整回报分布中包含的附加信息。而这些附加信息在 Lyle 等提出的文献里已经被证实有利于策略学习。在实际应用中，了解完整的回报分布，包括可能存在的极端值，对于制定鲁棒的策略和优化长期回报至关重要。例如：在面对高风险环境时，两个动作可能存在相同的期望回报，其中一个回报分布更为集中，另一个则波动更大，此时仅依赖于单纯地期望回报将无法正确区分这两种情况，进而导致次优策略。

当前基于值分布的智能体强化学习相关研究较少，据我们所知，最早的一项工作是 C51 算法，其核心思想是预测一系列固定数量的离散价值上的概率分布，而不是预测一个期望的累计回报。其中多智能体强化学习的一项工作是 DFAC，该模型从值分布角度建模个体和全局 Q 值，将回归分布分解为两部分：确定性部分(即期望值)和随机部分(均值为零)。DFAC 则进行了强力的假设：通过将全局价值函数的期望可以对个体价值函数的期望来拟合，而这个在实践中并不一定成立，该实验以两个智能体系统为例，其中全局价值函数与个体价值函数的关系为：$Z_{tot}(s,a)=Relu(Z_1+Z_2)$。另一项工作是 MCMARL [13]，该文针对 DFAC 提出的全局价值函数分布由两部分个体价值分布期望拟合的观点进行反驳，认为在个体遵循不同价值分布且具有相同的期望时，全局价值的期望有可能不同。该模型提出了新的值分布多智能体强化学习框架，通过混合分类分布来建模个体 Q 值分布和全局 Q 值分布并参数化值函数，定义加权、偏差、卷积、投影和函数等五种基本操作，实现分布变换和多重分布的组合。

针对于以上两点问题，同时为了使多智能体强化学习获取更好的结果和更稳定的学习过程，本文提出了一种基于值分布的强化学习算法。直观的说，与一般的强化学习算法相比，本章方法的贡献如下：

(1) 本章实验提出了 CvM-MIX 算法，进行对分类分布函数的参数化建模，其结合了值分布强化学习算法和值分解算法，捕捉回报完整信息，同时引入自适应价值原子间隔，提升从而提高对环境状态随机性的适应能力。

(2) 提出了基于 Cramér-von Mises 距离的混合网络，利用 CvM 在分布差异计算上的优异性能代替 KL 散度计算分类分部间差异。

(3) 基于权重优先级的经验回放方法，通过引入重要性抽样权重，有效提高了经验回放的效率。

2. 相关背景研究

2.1. 强化学习

强化学习是一种通过与环境交互、学习状态到行为的映射关系，学习实现特定目标的最优策略，从而获取最大积累期望汇报的方法。强化学习定义了两种价值函数用于表示智能体策略效用。其中一种是状态价值函数 $V_\pi(s)$ 表示从状态 s 开始，遵循当前策略 π 时所获得的预期回报，公式如下：

$$V_\pi(s)=E_\pi\left[G_t \mid S_t=s\right] \tag{1}$$

$$G_t=R_{t+1}+\gamma R_{t+2}+\gamma^2 R_{t+3}+\cdots=\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \tag{2}$$

将式(1)带入式(2)化简可得：

$$V_\pi(s)=E_\pi\left[R_{t+1}+\gamma V(S_{t+1}) \mid S_t=s\right] \tag{3}$$

而另一种则是状态行为价值函数 $Q(s,a)$，表示针对当前状态 s 执行某一具体行为 a 后，继续执行策略 π 所获得的期望回报；也表示遵循策略 π 时，对当前状态 s 执行行为 a 的价值大小，公式如下：

$$Q_\pi(s,a)=E_\pi\left[G_t \mid S_t=s,A_t=a\right] \tag{4}$$

将式(2)带入式(4)化简可得：

$$Q_\pi(s,a)=E_\pi\left[R_{t+1}+\gamma Q\left(S_{t+1},A_{t+1}\right) \mid S_t=s,A_t=a\right] \tag{5}$$

其中式(5)为 $t+1$ 时刻到 T 时刻的累计回报，R 为及时奖励回报，γ 为衰减因子。根据贝尔曼最优方程[14]可得：

$$V^*(s)=\max Q^*(s,a)=R_s^a+\gamma\sum_{s'\in S}P_{ss'}^aV^*(s') \tag{6}$$

2.2. 部分可观测多智能体马尔可夫决策过程

相比于一般的马尔可夫决策过程[15]，本文将多目标多智能体强化学习环境建模为部分可观测多智能体马尔可夫决策过程[16] (Dec-POMDP)。部分可观测多智能体马尔可夫决策过程是一种为多智能体环境设计的决策框架，用于处理每个智能体只能部分观测到整个环境状态的情况。在 Dec-POMDP 中，每个智能体根据自身的局部观测做出决策，而这些决策又会影响整个系统的动态。Dec-POMDP 可以定义为一个八元组：

$$M=(I,S,A,T,R,\Omega,O,\gamma) \tag{7}$$

式(7)中，I 为智能体数量，S 为状态空间；A 为联合动作空间；R 为回报函数；Ω 是观测空间，是系统中所有智能体观测到的环境信息集合；$\gamma\in[0,1]$ 为折扣因子，表示对未来奖励的不确定性。S 的演化是基于转移概率 $T(s,a,s'):S\times a\to\delta(s)$，同马尔可夫决策过程定义的一样，这表示给定联合动作 a 和当前状态 s，下一个状态将为 s'的概率。在每个时间步长中，所有智能体都收到一个观察概率 $o_i\in\Omega_i$，由联合观察概率 $O(o,s',a)=P(o\mid s',a)$ 给出。对于每个智能体 i，可以将其在迭代 t 时的局部观测历史定义为 $\vec{o}_t^{(i)}=\left(o_1^{(i)},\cdots,o_t^{(i)}\right)$。

在马尔可夫决策过程中，目标是通过选择一个最优的联合策略来最大化预期回报。但是在当前情况下，策略的执行需要从本地观察到操作的映射，因此，智能体 i 的本地策略为 $\pi^{(i)}:\vec{o}_t^{(i)}\to A^{(i)}$，最大预期回报如下：

$$J_i(\pi_i)=E\left[\sum_{t=0}^{n}\gamma^t r_i\left(S_t,A_{1,t},\cdots,A_{n,t}\right)\right] \tag{8}$$

2.3. CTDE 范式与值分解算法

CTDE 范式[17] (Centralized Training with Decentralized Execution, CTDE)是指集中式训练、分散式执行范式。具体来说，CTDE 旨在结合 IL 和 CL 的优点，CTDE 范式在集中式训练阶段，通过在训练阶段无限制地获取所有智能体的信息来学习策略，智能体可以访问全局状态信息，其中包括联合轨迹、共享全局奖励、策略和值函数，即使用一个全局价值函数或策略来指导所有智能体的学习过程。在分散式执行阶段，每个智能体只能依靠自身接受的局部观测信息为条件执行动作。CTDE 范式通过集中式学习来利用环境中的全局信息优化整体性能，而通过分散式执行来确保实际应用中的可行性和灵活性。多智能体系统的仿真训练通常不受实际硬件条件的限制，这意味着智能体间的通信不会受到阻碍，进一步促进了 CTDE 架构在实践中的应用。正是由于这些显著的优点，CTDE 成为了多智能体强化学习领域的一个标志性学习框架。

MADDPG [18]算法基于 CTDE 框架，训练阶段通过全局状态信息增强的 Critic 网络指导各个智能体的 Actor 网络进行训练，而在执行阶段，每个智能体以自身局部观察的 Actor 执行动作，此外，该算法通

过整合其他智能体策略信息提高了系统的整体鲁棒性。VDN [19]算法主要思路是智能体根据自身对整体的贡献优化各自的目标函数，通过引入价值分解网络来解决智能体之间的信用分配问题，其将多智能体系统的联合价值函数拆分为各个智能体价值函数的简单加和，从而更好地映射每个智能体对整体贡献的反映：

$$Q_{tot}\left(\tau,a\right)=\sum_{i=1}^{n}Q_i\left(\tau_i,a_i\right) \tag{9}$$

在简单任务上 VDN 算法十分高效，但是面对复杂环境的多智能体强化学习问题时，VDN 的训练效果则不尽人意，其原因在于 VDN 仅通过简单的线性加和方式将联合价值函数进行分解，从而导致智能体价值网络拟合效果受限。QMIX [20]算法旨在解决 VDN 算法中对于联合价值函数分解条件的严格限制，通过设计一种混合网络结构并引入超网络，确保了各个智能体的价值函数与整体价值函数之间在最优解上保持单调一致，联合动作价值函数如式(10)：

$$\arg\max Q_{tot}\left(\tau,a\right)\begin{cases}\arg\max Q\left(\tau_i,a_i\right)\\ \vdots\\ \arg\max Q\left(\tau_n,a_n\right)\end{cases} \tag{10}$$

QATTEN [21]算法则从理论上出发，推导如何将联合价值函数分解为局部值的过程，并采用多头注意力机制的网络以近似联合动作价值。

2.4. 基于值分布的多智能体强化学习

QR-MIX [22]和 RMIX [23]利用值分布强化学习提高其在 SMAC 环境中的性能表现。QR-MIX 是 QMIX 算法的一种变体，他将混合网络建模为 IQN，不同于将联合分布分解为个体效用分布，它将效用分解为狄拉克分布；另一方面，RMIX 也是 QMIX 的一种变体，其利用条件风险价值度量(CVaR)替换了混合网络的输入和输出。C51 算法通过预测一系列固定支持点上的返回值分布学习策略，但是由于预设了固定的支持点来近似价值分布，限制了算法捕捉真实价值分布的能力，若真实回报分布不适合这些预设的支持点，C51 算法可能无法准确估计价值分布。

3. 模型方法

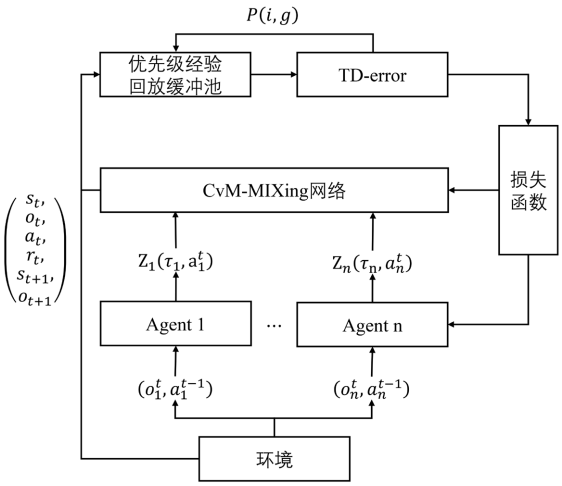


Figure 1. The overall architecture of CvM-MIX
图 1. CvM-MIX 算法整体流程

本文对多智能体深度强化学习算法的预期回报进行分布建模，引入值分布思想，提出了一种基于值分布的多智能体强化学习算法；同时在此基础上结合 Cramér-von Mises 距离[24]衡量 TD-error；最后引入了基于权重优先级经验回放机制。整体模型如图 1 所示。

3.1. 分类分布函数的参数化建模

基于价值的多智能体强化学习算法通常直接建模智能体状态-动作对的期望回报，具体可以表示为：

$$Q_\pi(s,a)=E_\pi\left[R_{t+1}+\gamma Q(S_{t+1},A_{t+1})\mid S_t=s,A_t=a\right] \tag{11}$$

式(11)中，$Q_\pi(s,a)$ 表示用于评估当前状态时刻状态－动作对 (s,a) 所计算的最大期望回报，R_{t+1} 为在状态 s 下采取动作 a 后获取的及时奖励，γ 是折扣因子。在使用基于价值的强化学习方法解决序贯决策问题中，$Q_\pi(s,a)$ 计算准确与否将最大程度上决定算法在任务环境中的表现。然而式(11)中却忽略了智能体与环境交互时内在随机性对期望回报的影响，该方法未能捕捉到回报分布的完整信息，$Q_\pi(s,a)$ 仅利用了完整分布信息中的平均期望信息，忽略了环境的随机性和不确定性，而平均期望在面对极端数值时易于受到影响，从而导致算法鲁棒性受到影响。除此之外，式(11)中针对预期回报采取最大化等行为均属于自举行为，因此将不可避免产生预期回报高估现象。基于以上问题，本文将采用值分布的概念，强调掌握预期回报分布特征的重要性，不再仅针对期望值进行建模，而是对预期回报分布特征进行建模，不仅仅考虑预期回报的平均值，同时还考虑回报的整体分布。本文将预期回报记作 $Z_\pi(s,a)$，通过概率回报分布来刻画智能体的值函数：

$$Z_\pi(s,a)=\sum_{t=0}^{T}\gamma^t R(s_t,a_t) \tag{12}$$

使用概率回报分 $Z_\pi(s,a)$ 等价值函数的预期回报 $Q_\pi(s,a)$，从而可以推出值分布的贝尔曼算子：

$$TZ_\pi(s,a)=R(s,a)+\gamma Z(s',a') \tag{13}$$

式(13)中，$Z_\pi(s,a)$ 不再表示单一的期望信息，而是一个随机变量的完整分布。

针对随机变量的分布函数 $Z_\pi(s,a)$ 采取 C51 的分类分布建模，公式如式(14)：

$$Z(i)=v_{\min}+i\cdot\Delta z \tag{14}$$

$$\Delta z=\frac{v_{\max}-v_{\min}}{N-1} \tag{15}$$

式(15)中，$v_{\max}$ 为价值分布的最大值(本文中 $v_{\max}=20$)，$v_{\min}$ 为价值分布的最小值(本文中 $v_{\min}=-10$)，N 为价值原子的总数(本文中 $N=15$)，Δz 是一个固定值，为相邻价值原子之间的间隔。

但是由于在不同的环境或不同状态下，价值分布的不确定性是不同的，使用固定间隔可能导致对高不确定性区域的过拟合；同时环境可能存在动态变化，而固定间隔 Δz 无法适应这种动态变化从而导致学习效率和性能下降，因此作为固定值的 Δz 不足以精确捕捉不同环境中所有的价值分布特性，本文在 C51 的分类分布建模基础上提出了可调节价值原子间隔，公式可表示为：

$$\Delta z_i=\frac{v_{\max}-v_{\min}}{N-1}\cdot f(\sigma_i) \tag{16}$$

式(16)中，$f(\sigma_i)=1+\alpha$，Δz_i 为价值分布在第 i 个区间的间隔调整值，$a\in(0,1)$ 是调节系数。此调节系数的目的在于通过 α 动态调整 Δz_i，可以灵活根据具体任务的需要调整价值分布的粒度，以捕获价值分布的细微差异，从而优化模型性能。

由于对每个智能体采用分类分布来参数化其 Q 值分布，这种分布具有很高的灵活性，可以近似任何形状的分布。假设智能体分布集合表示为 $[z]=\{z_1,z_2,\cdots,z_M\}$，其中 $z_i\in[z_1,z_M]$，分布在价值分布预定义

范围$[v_{\max}, v_{\min}]$中，x为遵循分类分布的离散随机变量。为了可以将分布可以近似任何形状，本文在基于MCMARL文章定义的5个基本操作基础上对部分操作进行了改进：

操作1. 加权。加权操作类似对于标量变量进行缩放操作，加权系数W用于以ω缩放离散随机变量：

$$WX = \omega x_j \quad w.p. \quad p_j \tag{17}$$

操作2. 偏置。偏置操作类似对于标量变量进行平移操作，偏置操作B将一个离散随机变量沿着b方向平移：

$$BX = x_j + \omega \quad w.p. \quad p_j \tag{18}$$

操作3. 卷积。卷积操作用于组合两个离散随机变量X_1和X_2，假定二者共享相同的原子间隔Δz_i：

$$Conv(X_1, X_2) = x_j^* \quad w.p. \quad p_j^* \tag{19}$$

操作4. 投影。投影操作目的是将随机变量分布x_j映射到原子$\hat{x}_k$，定义的投影操作如下：

$$\Phi_{\hat{x}_k} X = \hat{x}_k \quad w.p. \quad \sum_j \left[1 - \frac{\left| x_j{}_{v_{\min}}^{v_{\max}} - \hat{x}_k \right|}{\Delta z} \right] p_j \tag{20}$$

操作5. Relu函数。由于MCMARL中对函数定义过于宽泛，本文仅针对混合网络中使用到的Relu函数操作F_{Relu}进行定义：

$$F_{Relu} X = \max(0, x_j) \quad w.p. \quad p_j \tag{21}$$

3.2. 引入基于Cramér-von Mises距离的混合网络

Cramér-von Mises距离用于测量两个分布函数之间整体的平方差异的积分，因此对分布的形状非常敏感，例如：尾部行为和多峰性，这对于理解和优化具有复杂返回分布的环境特别重要。与KL散度相比，它是一种更加平滑的度量方式，不仅仅利用分布的均值或者方差，同时对异常值的敏感性较低，更适合处理分布的形状差异。因此，本文选择使用Cramér-von Mises距离代替KL散度用于评估分布差异。

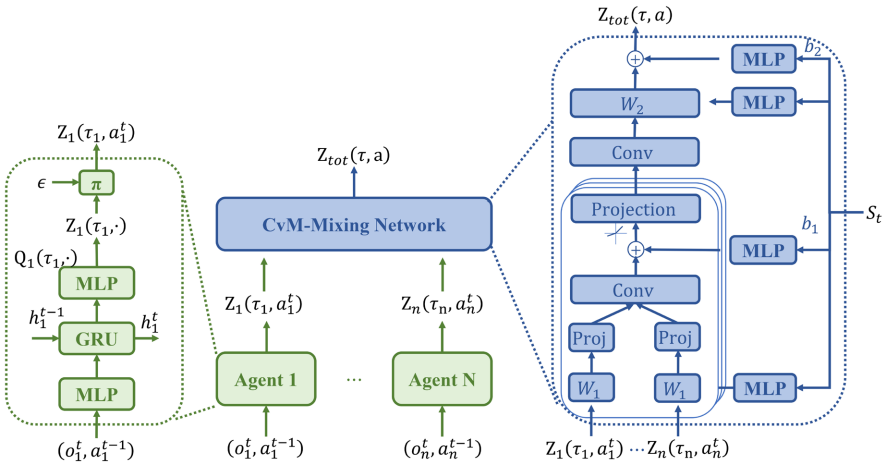


Figure 2. The overall architecture of CvM-MIX
图2. CvM-MIX的整体结构

给定状态s和动作a，令$Z_{tot}(s, a; \theta)$代表当前网络参数θ下的联合价值分布，$Z_{tot}(s, a; \theta')$代表基于贝尔曼目标的联合价值分布，其中θ'是目标网络的参数，基于以上假设，Cramér-von Mises距离的TD-error可以表示为：

$$\text{TD-error}_{CvM}(\theta)=\int_{-\infty}^{+\infty}\left[Z_{tot}(s,a;\theta')-Z_{tot}(s,a;\theta)\right]^2 \mathrm{d}x \tag{22}$$

但是考虑到本文借鉴 C51 算法，属于离散情况下处理离散化价值分布，因此将公式量化为：

$$\text{TD-error}_{CvM}(\theta)=\sum_{i=1}^{N}\left[Z_{tot}^{i}(s,a;\theta')-Z_{tot}^{i}(s,a;\theta)\right]^2 \tag{23}$$

算法整体结构如图 2 所示。

3.3. 值分布 IGM

在协同对抗任务中，基于值函数分解的多智能体强化学习算法需要遵从 IGM 原则，为确保联合贪婪动作值函数和个体贪婪行为选择的一致性，因此满足 IGM 原则的值分布分解对于分解回归分布是非常必要的，公式如下：

$$argmaxE\left[Z(\tau,a)\right]=\begin{cases} argmax_{u1}E\left[Z_1(\tau_1,\mathrm{a}_1)\right] \\ \cdots \\ argmax_{uK}E\left[Z_K(\tau_K,\mathrm{a}_K)\right] \end{cases} \tag{24}$$

3.4. 基于权重优先级的经验回放

在强化学习领域，传统的经验回放机制是一种常用的数据利用策略，旨在通过存储智能体与环境进行交互获取经验，并在后续的学习过程中重复利用这些经验达到学习最优策略并最大化其累计奖励的目的，最终提高学习效率。然而，此方法假设所有经验均等重要，忽略了不同经验对学习过程的影响程度存在显著差异事实。这种非差异化处理方式可能存在样本经验提取效率低下从而导致学习效率不佳等问题，尤其是在复杂的多智能体强化学习系统中，某些关键经验的缺失可能对整体学习进程具有决定性影响。

具体而言，使用时间差分误差作为评估经验价值的标准是一种有效的方法。TD 误差的大小反映了当前智能体策略预测的 Q 值与目标 Q 值之间的差异，即当前策略对于给定经验的适应程度。通过将 TD 误差转化为经验的优先级采样概率 P_i 并存处于经验回访池中，可以精确指导经验的抽取和利用。在该机制下，TD 误差较小的经验表示智能体已较好地掌握了相应应对策略，即这部分经验无需再被频繁采样进行训练。相反，当某个经验的估计 Q 值与目标 Q 值存在较大偏差时，表明该经验目前对于智能体相对稀缺，其代表了智能体目前策略中潜在改进点。因此，这类经验应当优先且频繁应用于训练，以帮助智能体更好适应并应对未来可能遭遇的类似情况。

本实验选取经验的 TD-error 作为评估经验价值的指标，使用 TD-error 定义优先级的程度，即 TD-error 数值越大，代表预测精度还需要继续提升，则该样本越需要被学习，从而意味着优先级越高。由于本文选择使用 Cramér-von Mises 距离代替 KL 散度用于评估分布差异，因此 TD-error 公式如下所示：

$$\text{TD-error}_{CvM}(\theta)=\sum_{i=1}^{N}\left[Z_{tot}^{i}(s,a;\theta')-Z_{tot}^{i}(s,a;\theta)\right]^2 \tag{25}$$

为了避免使用贪婪 TD-error 优先级造成采样多样性缺失，从而使系统陷入过拟合状态，因此引入了一种随机经验采样方法，定义经验采样的概率为：

$$P(i,g)=\frac{{P_{i,g}}^{\alpha}}{\sum {P_{i,k}}^{\alpha}}+\delta \tag{26}$$

其中，优先级 $P_{i,g}=\dfrac{1}{Rank(i,g)}$，为智能体 i 的第 g 个经验样本在经验回放缓冲区的排列位置。参数 α

控制优先级的使用程度，当 α= 0 时，为均匀采样；当 α = 1 时，为贪婪策略采样。由于采用优先级经验采样策略，因此为避免部分经验采样概率过低导致选中概率趋近于 0 的情况，对经验采样概率添加常量 $\delta \in (0,1)$，从而确保每条经验均具备 δ 的采样概率。

但是，由于采样策略更倾向于对高 TD-error 的经验进行采样，这将改变状态的访问频率，进而导致神经网络的训练过程易于震荡，甚至会严重发散。因此为解决该问题，在权重变化计算中采取重要性抽样权重，公式如下：

$$\omega_{i,g} = \left(\frac{1}{N} \frac{1}{P(i,g)} \frac{1}{f_{(i,g)}} \right)^{\beta} \tag{27}$$

其中，$\omega_{i,g}$ 为智能体 i 的第 g 个经验样本的重要性采样权重，N 是回放缓冲区的大小，$P(i,g)$ 是采样智能体 i 的第 g 个经验样本，$f_{(i,g)}$ 为智能体 i 的第 g 个经验样本频率的倒数，用于衡量经验在经验回放池中的稀缺性，经验被采集次数越少则意味着该经验对于当前策略较为罕见，则被认为越稀缺，因此应赋予更高权重。参数 β 用于控制矫正使用的程度，当 β = 0 时，重要性抽样权重采取完全不使用重要性采样；当 β = 1 时，要性抽样权重采取正常重要性采样，因此在训练临近结束时，应令 β 趋近于 1。

4. 实验环境

为了全面评估我们算法的性能和鲁棒性，我们在多种具备离散动作状态的场景中进行了详尽的实验。这些实验是在《星际争霸 2》平台上的星际争霸多智能体挑战赛(以下简称 SMAC)中进行的。在 SMAC，我们选取了多种对抗模式，并训练了红方的多个单位以便它们能够有效地对抗拥有固定策略的电脑对手，此部分实验的核心目的在于验证我们的算法是否能够有效学习策略。

本文采用的实验平台配置：CPU 为 Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz，GPU 为 Tesla V100-SXM2-32GB GPU，操作系统为 CentOS Linux release 7.9.2009 (Core)，深度学习框架使用 Pytorch 1.8.2。

本文选取了默认 AI 难度——等级 7。具体对战场景包含“3 m”、“8 m”、“2s3z”和“3s5z”，具体信息见表 1。

Table 1. System resulting data of standard experiment
表 1. 具体对战场景

地图	我方单位	AI 单位	类型
3 m	3 名海军陆战队员	3 名海军陆战队员	同构、对称
8 m	8 名海军陆战队员	8 名海军陆战队员	同构、对称
2s3z	2 名追踪者 3 名狂热者	2 名追踪者 3 名狂热者	异构、对称
3s5z	3 名追踪者 5 名狂热者	2 名追踪者 3 名狂热者	异构、对称

实验中各个算法具体超参数如表 2 所示。

Table 2. System resulting data of standard experiment
表 2. 具体对战场景

超参数	
Steps	2e6
最大 ϵ 概率	1
最小 ϵ 概率	0.05

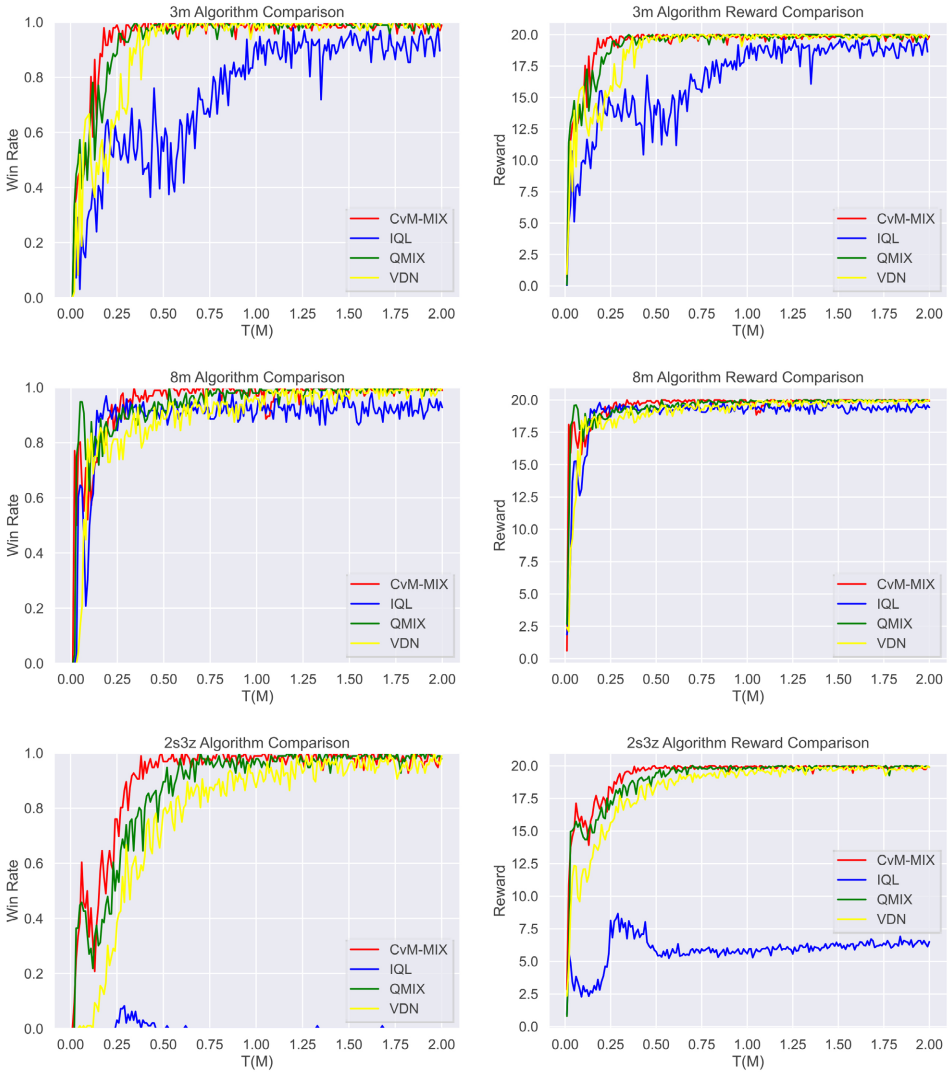
续表

ϵ 概率衰减步长	5e4
经验回放池容量	5e3
Batch size	32
更新步数	50
学习率	5e-4
最大单回合步数	50

5. 结果分析

为验证 CvM-MIX 的各种性能，本文在星际争霸 2 多智能体挑战赛环境中进行了训练并测试了模型，将结果与文献提出的 VDN 算法模型和文献提出的 QMIX 算法模型以及 IQL 算法模型进行对比。本文各算法模型均运行 3 次并取最终测试结果的平均值。

对比实验



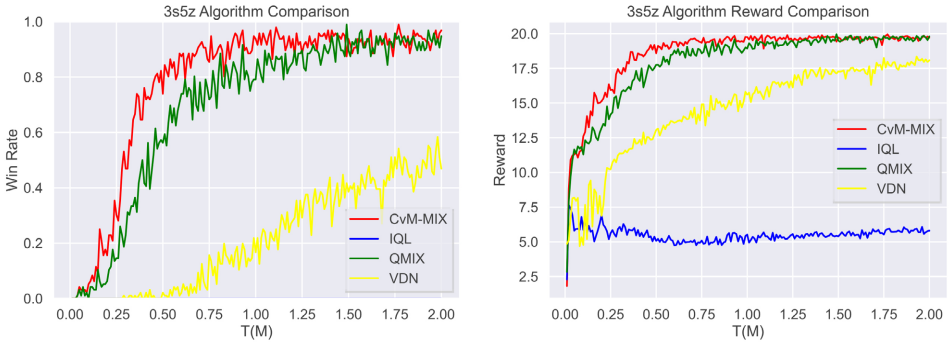


Figure 3. The winning rate curve and reward return curve of each algorithm in testing
图 3. 各算法在测试中的胜率曲线以及奖励回报曲线

图 3 中的数据显示出了 CvM-MIX 算法几乎在所有场景中均表现出色。在 3 m 和 8 m 场景中，IQL 算法性能提升较慢，最终性能表现相对较差。各算法在收敛的稳定性上，本文所提出的 CvM-MIX 算法在中后期表现始终稳定，其余算法均存在一定程度的波动，这是由于 CvM-MIX 算法采用了分类分布函数的参数化建模，充分利用了分类分布建模联合动作函数完整的建立了对局信息，提高了算法的稳定性。同时，CvM-MIX 胜率提升速度最快，并最先趋于收敛，因此，CvM-MIX 表现出了相比基线算法更加出色的性能。

其次在相对较为复杂的 2s3z 和 3s5z 地图表现上，四种算法差距进一步拉大。IQL 算法未能收敛，同时训练曲线表现较差，因此性能存在较大差距；QMIX 算法尽管成功收敛，但是收敛时间相比较于 CvM-MIX 算法较慢；CvM-MIX 算法在 3s5z 地图上表现出了更为出色的收敛速度和性能，同时训练曲线相较于 QMIX、VDN 算法而言波动较为平稳。

奖励回报曲线图表现上，CvM-MIX 算法在所有场景中，无论是收敛速度、稳定性以及性能表现方面均优于其他算法，展示了 CvM-MIX 算法的性能优势。

6. 结束语

针对传统多智能体强化学习算法的局限性，本文针对性地提出了基于值分布的多智能体强化学习算法 CvM-MIX，并将提出的算法与传统算法 VDN、QMIX 和 IQL 进行对比，结果表明，本文提出的算法可以在获取完整价值分布的同时，提升模型性能的表现，验证了算法的有效性。

参考文献

[1] Li, Y. (2017) Deep Reinforcement Learning: An Overview. arXiv preprint arXiv:1701.07274.

[2] LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep Learning. *Nature*, **521**, 436-444. https://doi.org/10.1038/nature14539

[3] Kaelbling, L.P., Littman, M.L. and Moore, A.W. (1996) Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, **4**, 237-285. https://doi.org/10.1613/jair.301

[4] Barto, A.G., Sutton, R.S. and Watkins, C. (1989) Learning and Sequential Decision Making. University of Massachusetts, Amherst.

[5] Fedus, W., Ghosh, D., Martin, J.D., *et al*. (2020) On Catastrophic Interference in Atari 2600 Games. arXiv preprint arXiv:2002.12499.

[6] Conrad, S., Teichmann, J., Auth, P., *et al*. (2024) 3D-Printed Digital Pneumatic Logic for the Control of Soft Robotic Actuators. *Science Robotics*, **9**, eadh4060. https://doi.org/10.1126/scirobotics.adh4060

[7] Brown, N. and Sandholm, T. (2018) Superhuman AI for Heads-up No-Limit Poker: Libratus Beats Top Professionals. *Science*, **359**, 418-424. https://doi.org/10.1126/science.aao1733

[8] Brown, N. and Sandholm, T. (2019) Superhuman AI for Multiplayer Poker. *Science*, **365**, 885-890.

https://doi.org/10.1126/science.aay2400

[9] Da Silva, F.L. and Costa, A.H.R. (2019) A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems. *Journal of Artificial Intelligence Research*, **64**, 645-703. https://doi.org/10.1613/jair.1.11396

[10] Bellemare, M.G., Dabney, W. and Munos, R. (2017) A Distributional Perspective on Reinforcement Learning. *Proceedings of the* 34*th International Conference on Machine Learning*, Sydney, 6-11 August 2017, 449-458.

[11] Sun, W.F., Lee, C.K. and Lee, C.Y. (2021) DFAC Framework: Factorizing the Value Function via Quantile Mixture for Multi-Agent Distributional Q-Learning. *Proceedings of the* 38*th International Conference on Machine Learning*, 18-24 July 2021, 9945-9954.

[12] Hong, Y., Jin, Y. and Tang, Y. (2022) Rethinking Individual Global Max in Cooperative Multi-Agent Reinforcement Learning. *Advances in Neural Information Processing Systems*, **35**, 32438-32449.

[13] Zhao, J., Yang, M., Zhao, Y., *et al*. (2023) MCMARL: Parameterizing Value Function via Mixture of Categorical Distributions for Multi-Agent Reinforcement Learning. *IEEE Transactions on Games*, 1-10. https://doi.org/10.1109/TG.2023.3310150

[14] Kappen, H.J. (2011) Optimal Control Theory and the Linear Bellman Equation. In: Barber, D., Cemgil, A.T. and Chiappa, S., Eds., *Bayesian Time Series Models*, Cambridge University Press, Cambridge, 363-387. https://doi.org/10.1017/CBO9780511984679.018

[15] Filar, J. and Vrieze, K. (2012) Competitive Markov Decision Processes. Springer Science & Business Media, Berlin.

[16] Guicheng, S. and Yang, W. (2022) Review on Dec-POMDP Model for Marl Algorithms. In: Jain, L.C., Kountchev, R., Hu, B. and Kountcheva, R., Eds., *Smart Communications, Smart Communications, Intelligent Algorithms and Interactive Methods*, Springer, Singapore, 29-35. https://doi.org/10.1007/978-981-16-5164-9_5

[17] Zhou, Y., Liu, S., Qing, Y., *et al*. (2023) Is Centralized Training with Decentralized Execution Framework Centralized Enough for MARL? arXiv preprint arXiv:2305.17352.

[18] Lowe, R., Wu, Y.I., Tamar, A., *et al*. (2017) Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 6382-6393.

[19] Sunehag, P., Lever, G., Gruslys, A., *et al*. (2017) Value-Decomposition Networks for Cooperative Multi-Agent Learning. arXiv preprint arXiv:1706.05296.

[20] Rashid, T., Samvelyan, M., De Witt, C.S., *et al*. (2020) Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *The Journal of Machine Learning Research*, **21**, 7234-7284.

[21] Yang, Y., Hao, J., Liao, B., *et al*. (2020) Qatten: A General Framework for Cooperative Multiagent Reinforcement Learning. arXiv preprint arXiv:2002.03939.

[22] Hu, J., Harding, S.A., Wu, H., *et al*. (2020) QR-MIX: Distributional Value Function Factorisation for Cooperative Multi-Agent Reinforcement Learning. arXiv preprint arXiv:2009.04197.

[23] Qiu, W., Wang, X., Yu, R., *et al*. (2021) RMIX: Learning Risk-Sensitive Policies for Cooperative Reinforcement Learning Agents. *Advances in Neural Information Processing Systems*, **34**, 23049-23062.

[24] Darling, D.A. (1957) The Kolmogorov-Smirnov, Cramer-von Mises Tests. *The Annals of Mathematical Statistics*, **28**, 823-838. https://doi.org/10.1214/aoms/1177706788