SEA：面向多智能体强化学习的空间显式架构

李大鹏，徐志伟，张斌，樊国梁自动化研究所，中国科学院人工智能学院，中国科学院大学中国北京 lidapeng2020@ia.ac.cn，xuzhiwei2019@ia.ac.cn，zhangbin2020@ia.ac.cn，guoliang.fan@ia.ac.cn

arXiv:2304.12532v1 [cs.MA] 25 Apr 2023

摘要—**空间信息在各个领域都至关重要。如何根据智能体的空间位置进行显式建模，对于多智能体问题也非常重要，尤其是在智能体数量不断变化且规模庞大的情况下。受计算机视觉中点云任务的启发，本文提出了一种用于多智能体强化学习的空间信息提取结构。智能体可以通过空间显式的编码器-解码器结构，有效共享邻域信息和全局信息。我们的方法遵循集中式训练与分散式执行（CTDE）范式。此外，我们的结构经过少量修改即可应用于多种现有主流强化学习算法，并能够处理智能体数量可变的问题。在多个多智能体场景中的实验表明，通过加入我们的空间显式架构，现有方法可以取得令人信服的结果。**

I. 引言

近年来，多智能体强化学习（多智能体强化学习）取得了诸多进展 [1]–[4]。尽管上述成果取得了成功，但很少有多智能体强化学习算法能够成功应用于现实世界的应用场景。其中一个重要原因是，在某些环境中，由于智能体数量庞大，问题会变得极其复杂。如果我们简单地使用独立的单智能体强化学习算法，而不考虑其他智能体的信息，通常很难形成协作 [5]。然而，使用集中式方法控制所有智能体会导致状态和动作空间呈指数级爆炸 [6]。此外，许多场景中的智能体数量会在回合之间以及回合内部不断变化。因此，可以总结出两个主要问题：a) 如何有效利用或结合全局和局部信息。b) 当智能体规模增加时，如何避免维度灾难。

为了解决上述问题，一些研究 [7]–[10] 尝试仅考虑邻近智能体的信息，而不是所有其他智能体的信息，从而简化问题。然而，这种邻近近似会丢失全局信息，并可能导致全局协作能力下降。此外，其中大多数方法只是简单地计算邻接矩阵，忽略了不同距离下邻居影响力的变化。

一些方法使用特定的网络结构，在执行或训练期间共享智能体之间的信息。例如，CommNet [11] 使用平均池化汇总所有智能体的信息，以促进智能体之间的通信。然而，简单的平均操作会丢失每个智能体的特定信息。因此，BicNet [12] 利用循环神经网络（RNN）在智能体之间传递信息，以避免上述缺陷。然而，智能体的输入顺序会影响 RNN 的输出，因此，当智能体的顺序和数量不固定时，此类方法会变得不稳定。

多智能体强化学习中的另一分支是网格方法 [13]。与计算机视觉中的分割任务类似，Grid- Net [13] 将连续环境划分为离散网格空间，并为每个网格提供预测结果。通过将全局状态输入编码器-解码器架构中的卷积网络，可以抽象出所有智能体的联合空间表示。然而，这种网格结构极大地限制了其准确性和实用性。

上述示例表明，利用智能体的空间位置以及智能体之间的信息共享，对于解决多智能体问题具有重要意义，但二者都面临各自独特的挑战。因此，设计一种能够兼顾智能体空间信息并有效提取智能体之间信息的通用框架，是一个亟待解决的问题。我们发现，点云分类问题与大规模多智能体问题之间存在显著相似性。具体而言，二者都包含具有空间拓扑的点（或智能体），且每个点都包含多个特征。每个点的预测都基于其他局部点或所有点的信息。受点云问题中一系列经典工作的启发 [14], [15] ，我们提出了一种通用的**空**间**显**式**架**构（**SEA**），将每个智能体视为具有特征的坐标点，并通过编码器-解码器结构学习联合空间表示。首先，编码器根据智能体的位置将智能体集合划分为相互重叠的局部簇。然后，从每个小簇中提取局部特征。通过多层级提取，智能体的感受野不断扩大。因此，更高层级的簇包含全局特征。在解码器中，根据每个簇的位置重建其特征。因此，输出特征同时包含局部和全局信息。SEA 的优势可总结如下：

- SEA 可以促进局部和全局信息共享，并有效利用空间信息。

- SEA 可以处理大规模以及智能体数量变化的问题。
- SEA 的输出结果不受智能体输入顺序的影响。

在本文中，我们将 SEA 与当前主流的多智能体强化学习算法相结合。在连续观测和离散观测场景中的实验结果表明，引入 SEA 可以显著提升当前主流强化学习算法的性能。

II. 相关工作

本文主要涉及两类学术文献：多智能体强化学习和空间显式学习。

A. 多智能体强化学习

在多智能体强化学习问题中，最简单的训练方法是独立训练和执行每个智能体，并将其他智能体视为环境的一部分 [5]。尽管这种完全去中心化方法易于实现且计算成本较低，但由于每个智能体没有考虑其他智能体的动态变化，因此具有不稳定性。与之相反的方法也称为完全中心化方法，它使用中心化模型基于全局信息为所有智能体做出决策。然而，随着智能体数量的增加，状态空间和决策空间将呈指数级增长。对于中心化方法，现有研究主要设计特定的策略结构，包括使用循环神经网络和注意力模型实现智能体之间的信息共享，或直接输入全局状态，通过多个头为每个智能体生成动作。

除了上述两种极端方法之外，另一种折中方案是集中训练分散执行（CTDE）。在集中训练分散执行的设置下，智能体可以在训练期间共享信息，同时在执行期间仅依据局部观测做出决策。这种方法受到了广泛研究，主要包括集中式策略梯度方法和价值分解方法。在集中式策略梯度方法中，每个智能体由一个中心化评论家和一个去中心化执行者组成。中心化评论家可以访问所有智能体的全局信息。经典的集中式策略梯度工作包括 MADDPG [16]、COMA [17], 和 MAPPO [18], [19]。价值函数分解方法将联合状态动作价值函数分解为各个智能体的独立价值函数。例如，VDN [20] 将联合Q值表示为各个独立Q值之和，而 QMIX [21] 使用参数化混合网络来计算联合Q值。由于 QMIX 表现优异，研究人员提出了一系列变体 [22]–[24] 来解决各种问题。

B. 空间显式学习

空间信息已广泛应用于社交网络和生态学等多个领域。许多研究表明，显式考虑空间关系的模型表现优于未考虑空间关系的模型 [25]。在多智能体领域，也有许多工作 [7], [8], [10], [26] 通过考虑邻近智能体的信息来实现局部协作。这些研究表明，引入空间信息可以提升模型的表现。由于只考虑邻域，大多数此类方法都能处理智能体数量不固定的情况。然而，上述方法只关注局部信息，却忽略了全局共享。

我们的研究受到其他关注空间信息的领域启发，例如计算机视觉中的点云分割问题。点云分割中最具挑战性的问题，是如何处理数量庞大且变化多样、带有特征的空间点；多智能体强化学习场景中也存在这一问题。此外，在点云分割和多智能体问题中，都需要权衡全局和局部特征。对于点云分割问题，PointNet [14] 使用简单的最大池化，只能获取全局信息。其改进版本 PointNet++ [15]则尝试采用多层级结构来处理这种权衡。

III. 背景

A. 问题形式化

强化学习可以视为智能体通过与环境持续交互，逐步学习以获得最大回报的过程。本文中的协作多智能体问题可形式化为一个元组 $G = \langle S, A, N, O, \mathcal{T}, Z, r_i, \gamma \rangle$。$s \in S$是环境的真实全局状态。在每个离散时间步 t，每个智能体$i \in \mathcal{N} := \{1, \ldots, N\}$ 都会选择一个动作 $a_i \in A_i$，其中$A = A_1 \times \cdots \times A_N$是联合动作空间。 $\mathcal{T}(s'|s, \boldsymbol{a})$: $S \times A \times S \to P(S)$是状态转移函数，其中 $\boldsymbol{a} = \{a_1, \ldots, a_N\}$ 是联合动作。在部分可观测马尔可夫决策过程中，环境的全局状态无法获取，每个智能体 i 只能通过观测函数$Z(s, i)$: $S \times \mathcal{N} \to O$获取其个体观测 $o_i \in O$。 $r_i(s, a_i)$: $S \times A_i \to \mathbb{R}$ 是每个智能体 i的奖励函数。回报 $R_i = \sum_{t=0}^{T} \gamma^t r_i^t(s, a_i)$定义为折扣奖励和，其中 $\gamma \in [0, 1]$是折扣因子。每个智能体 i 的目标是学习策略 $\pi_i(a_i|o_i)$: $O_i \to P(A_i)$，以最大化其期望回报。

B. 中心化策略梯度方法

在单智能体任务中，基于价值的强化学习方法通常需要评估最优累计奖励，并据此确定最优策略。然而，在基于策略的方法中，策略函数会被直接优化，而价值函数的评估则是可选的、次要的。策略梯度方法始终包含一个参数化策略函数，通常称为“演员”。演员的参数可以通过梯度下降直接优化，以找到最优策略。策略梯度可以通过以下公式计算：

$$g = \mathbb{E}\left[\sum_{t=0}^{\infty} \Psi_t \nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right)\right], \tag{1}$$

其中， Ψ_t 可以由不同算法以不同方式表示。除了以优势演员

评论家网络（A2C）为代表的早期工作外，策略梯度还分为两个分支，即基于信赖域的方法和确定性策略梯度方法。与完全分散式和完全中心化方法相比，集中训练分散执行（CTDE）作为一种折中方法，已逐渐成为更受欢迎的多智能体训练范式。在集中训练阶段，智能体可以获取其他所有智能体的观测或策略等信息，但在分散执行过程中，只能根据自身的局部观测选择适当的动作。因此，基于CTDE 的方法可以缓解环境的不稳定性，同时也适用于部分可观测多智能体任务。中心化策略梯度方法包括MAA2C、MADDPG 和 MAPPO，它们将单智能体策略梯度方法应用于多智能体问题，并遵循 CTDE 范式。借助演员-评论家框架，这类方法可以在执行时使用演员模型进行决策，并在训练过程中通过评论家网络评估所有智能体的联合动作。

集中训练分散执行方法近年来在多智能体领域取得了进展。然而，这类方法没有考虑智能体之间的空间关系，而是将位置信息作为常规特征，并在不将其与其他特征区分开的情况下输入模型进行学习。这些隐式方法忽略了大量空间信息以及智能体之间的关系。

IV. 方法

A. 点视角下的空间表征

在本章中，我们首先简要介绍点云分割与多智能体协作之间的共性，然后从点视角出发考察智能体，从而建立点云与多智能体问题之间的联系。因此，我们可以自然地利用点云领域中的许多结构来解决多智能体问题中的空间表示问题。

点云分割的目的是对大量点集进行分类。对于具有 $x_i \in \mathbb{R}^d$ 的无序点集 $\{x_1, x_2, \ldots, x_N\}$，PointNet 使用最大池化操作将整个点集聚合为一个向量：

$$f(x_1, x_2, \ldots, x_N) = \phi(\mathrm{MAX_{i=1,\ldots,N}}\{\beta(x_i)\}), \qquad (2)$$

其中， ϕ 和 β 是多层感知机网络。尽管这一简单操作具有一些良好特性，但它会丢失不同尺度的局部特征信息。因此，PointNet++ 提出了一种多尺度特征表示，可以提取不同尺度下的全局和局部信息。

同样，在大多数多智能体环境中，智能体都有其地理位置 [27]。因此，在多智能体问题中构建空间模型，以提取智能体之间的空间关系，是一个实际问题。然而，大多数算法都忽略了这些信息，并将位置视为普通特征，这会导致大量重要信息丢失。在这种情况下，我们希望模型具备以下优良属性：(a) 模型对输入排列不变；(b) 模型结构能够处理数量可变的智能体；(c) 模型能够提取全局和局部信息。因此，这个问题与点云分割问题非常相似。具体而言，两者都有大量具有空间拓扑的点，且每个点都有自己的特征。对每个点的预测都需要同时考虑邻域和全局点。因此，我们可以将每个智能体视为具有空间坐标的特征点。所有特征智能体的坐标位置为 $\{x_1, \ldots, x_N\}$，智能体的局部观测特征为 $\{o_1, \ldots, o_N\}$

受 PointNet++ 启发，我们使用编码器-解码器结构作为空间提取器。该结构通过多层下采样生成具有不同视觉范围的高维聚类中心特征点，然后通过多层上采样为每个智能体插值出包含局部和全局信息的新特征。在编码器中，我们首先使用最远点采样 [15]选择 N_l 个智能体作为聚类的质心，并在第 l层进行分组，其中 $l \in \{1, 2\}$是下采样层的数量。 $N \geq N_1 \geq N_2$，且N 是输入智能体的数量。然后，为每个聚类质心划分最近的 K个智能体作为其邻域。需要注意的是，每个智能体可以被划分到多个聚类中。

对于每个聚类，我们可以使用 PointNet（如公式 2 所示）提取每个聚类中的特征。对于一个聚类中的点子集坐标$\{x_1, \ldots, x_n\}$及其局部观测$\{o_1, \ldots, o_n\}$，其中 n是该聚类中智能体的数量，其公共特征可提取如下：

$$f(o_1, \ldots, o_n) = \phi(\mathrm{MAX_{i=1,\ldots,n}}\{\beta(o_i)\}). \qquad (3)$$

假设 x_j 是该簇的中心点；该簇的输出 f_j 和中心点的坐标 x_j将构成更高层级的新点。使用MLP层 β 和 ϕ 的原因在于，整个簇包含的信息多于单个点，因此我们需要使用一个层来增加特征向量的维度，以避免在分组过程中丢失过多信息。通过重复这一上采样过程，我们可以获得多层级的点子集，因此，更高层级的点具有更大的视觉范围。由于最大值运算是一种对称函数，输出与输入智能体的顺序无关。

在解码器中，计算层 $l-1$ 中的智能体与层 l 中其最近的 k 个智能体之间的逆距离，将其作为插值权重，并通过插值生成层 $l-1$ 中 N_{l-1} 个智能体的新特征。插值过程可表示如下：

$$h_j = \frac{\sum_{i=1}^{k} w_i(x_j) f_i}{\sum_{i=1}^{k} w_i(x_j)} \quad w_i(x) = \frac{1}{d(x, x_i)^p}, \qquad (4)$$

其中， d 是距离度量。

*B. MARL*的空间显式架构

与将这种结构端到端地应用于计算机视觉中的分割任务不同，我们将这种编码器-解码器结构用作中间插件，以提取空间表示。因此，该结构的输出维度与输入维度

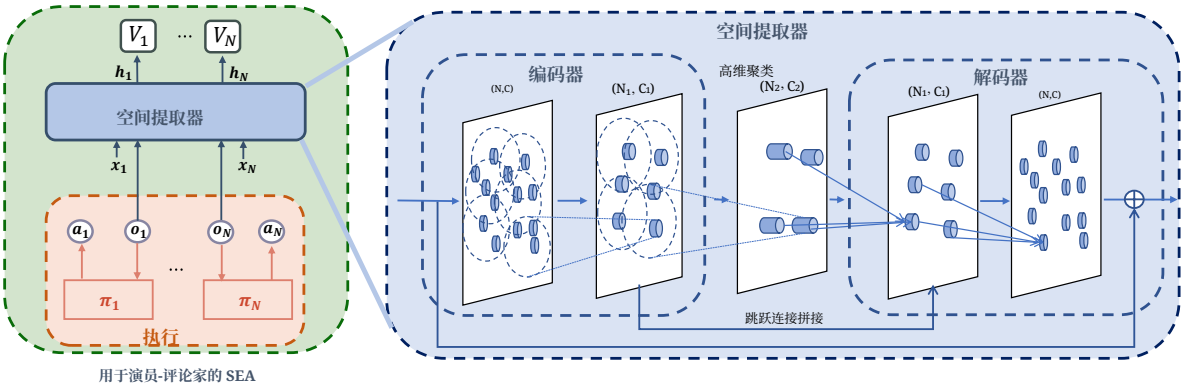


图1。SEA-AC 的总体设计。左侧是 Actor-Critic 结构中 SEA 的示意图，右侧是 SEA 结构的框架。

设置为相同，这使其更具通用性，也更容易迁移到不同的多智能体强化学习算法中。我们还添加了一个额外的残差结构。此外，考虑到在强化学习中使用批归一化层会带来不稳定性，我们移除了批归一化层。整个架构如图 1 右侧所示。

接下来，我们将介绍如何将这一空间显式架构与现有的多智能体框架相结合，包括 A2C、PPO 和 DDPG。

a) SEA-A2C：为了满足集中训练分散执行（CTDE）的设置，我们不在演员中使用额外信息。如图1左侧所示，在评论家网络中，通过设置智能体的坐标 x_i及其对应的观测 o_i，将其构造为一个特征点 $\{x_i, o_i\}$。将同一帧中的所有特征点输入空间提取器，每个智能体都可以利用局部和全局信息获得新的隐藏特征，

$$\boldsymbol{h} = \{h_1, \ldots, h_N\} = \mathrm{SE}(x_1, \ldots, x_N, o_1, \ldots, o_N), \quad (5)$$

然后将每个智能体的新的隐藏特征 h_i 输入多层感知机，以获得每个智能体当前的状态价值。我们将这一结构称为 SEA-A2C。

b) SEA-PPO：PPO 算法与 A2C 类似，因为它同样计算状态价值函数，但它具有梯度裁剪功能，并且可以针对相同的轨迹进行多个训练轮次。由于我们只改变评论家网络的输入，因此评论家网络的损失函数可以表示如下：

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} (\max[(V_\theta(h_i) - \hat{R})^2, (\mathrm{clip}(V_\theta(h_i), V_\theta(h_i) - \varepsilon, V_\theta(h_i) + \varepsilon) - \bar{R})^2]). \quad (6)$$

c) SEA-DDPG：由于输入是观测和动作，SEA-DDPG 学习的是动作价值函数，因此每个智能体的隐藏状态可以如下构成：

$$\boldsymbol{h} = \mathrm{SE}(x_1, \ldots, x_N, o_1, \ldots, o_N, a_1, \ldots, a_N), \quad (7)$$

策略梯度可计算如下：

$$\nabla J(\pi_i) = \mathbb{E}_{\mathcal{D}}[\nabla \pi_i(\tau_i) \nabla_{a_i} Q_i(h_i)|_{a_i = \pi(\tau_i)}], \quad (8)$$

$Q_i(h_i)$ 表示动作价值函数。

SEA 的优势在于，它不仅能够基于智能体之间的空间位置实现特征表示，还能够处理智能体数量变化或规模较大的问题。

d) 聚类设置和死亡掩码：在许多多智能体场景中，有两个重要特性：一个回合内智能体的数量会不断变化（由于智能体死亡，其中大多数情况下数量会不断减少），或者不同场景中的智能体规模差异很大。这种变化会给我们的算法带来两个主要问题：如何在训练中处理这些死亡智能体，以及如何针对不同规模的任务设置每一层中合适数量的聚类。对于第一个问题，我们只需在编码器的第一层滤除那些死亡智能体，否则这些死亡智能体也会被划分到簇中并影响结果。为了解决第二个问题，我们简单地将簇的大小设置为与每个场景中的智能体规模成比例。对于包含 n 个智能体的场景，第 1 层和第 2 层中的聚类数量可以计算如下：

$$N_1 = \lfloor \sqrt{n} \rfloor \\ N_2 = \lfloor \sqrt{N_1} \rfloor. \quad (9)$$

V. 实验

为了评估SEA在不同智能体规模下的能力，我们在四个多智能体环境中评估SEA，包括协作导航 [28], 战斗、战场 和*Gather* [27]。这些场景中的智能体数量从几个到数百个不等，既包括离散观测场景，也包括连续观测场景。

通过这些实验，我们希望回答以下问题：

问题1：SEA方法能为现有方法带来多大程度的改进?

问题2：与其他同样在智能体之间交互信息的算法（例如注意力机制）相比，它的表现如何?

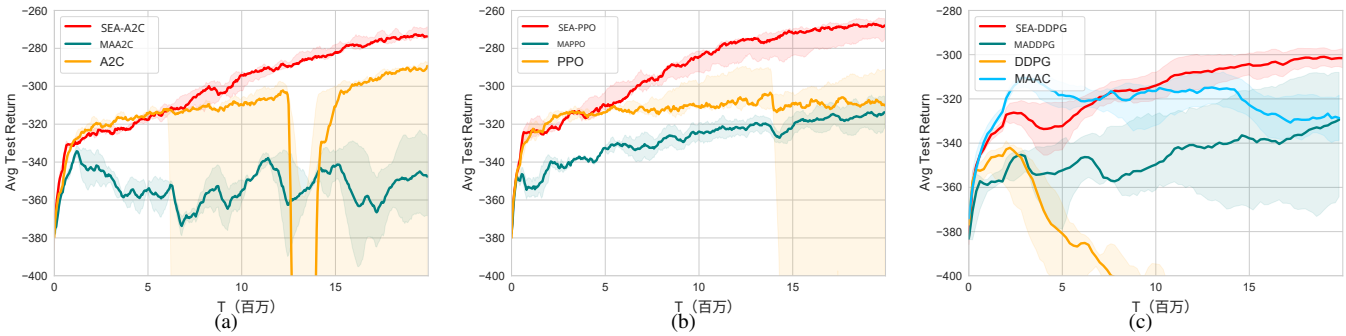


图2. 不同算法在协作导航上的智能体平均测试回报。

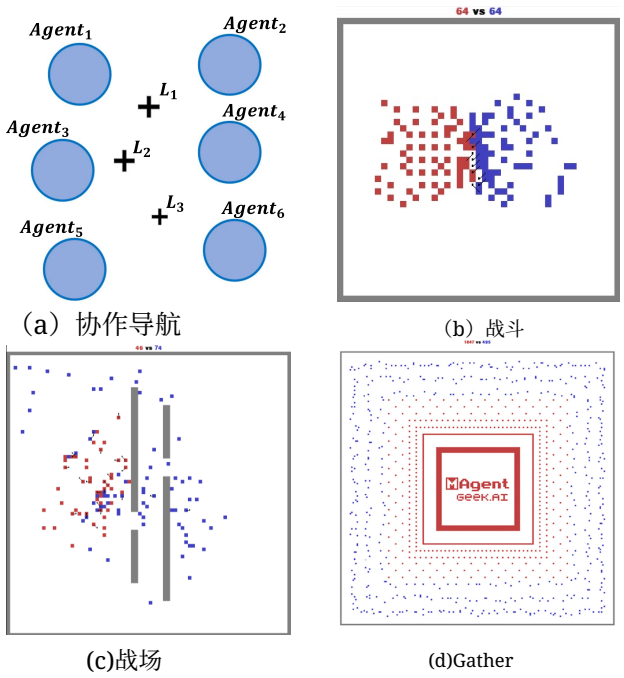


图3. 我们进行实验的四个环境的可视化。(a)协作导航(b)战斗(c)战场 (d)*Gather*。

问题3：局部和全局信息如何影响 SEA 的性能?

A. 环境描述

我们在四个多智能体环境中测试了我们的算法，包括协作导航、战斗、战场和*Gather*。所有环境的可视化结果见图3，下面将详细介绍我们测试的每个环境。**协作导航** 协作导航 是一个来自多智能体粒子环境的任务。该任务涉及连续二维环境中的粒子和地标。最大回合长度为25，即经过25步后回合终止。所有智能体都会获得自身的速度、位置，以及相对于其他所有智能体和地标的相对位置。**战斗** 战斗 场景包含两支队伍，即红队和蓝队，每支队伍有121个智能体。地图大小为 80×80。

智能体会缓慢恢复生命值，因此最好迅速消灭敌方智能体。智能体初始拥有10点生命值，每次攻击减少2点生命值，并且每步恢复0.1点生命值。当智能体攻击对手智能体时，会获得0.2的奖励；死亡时会受到-0.1的惩罚。此外，每次攻击动作都会受到-0.1的惩罚。

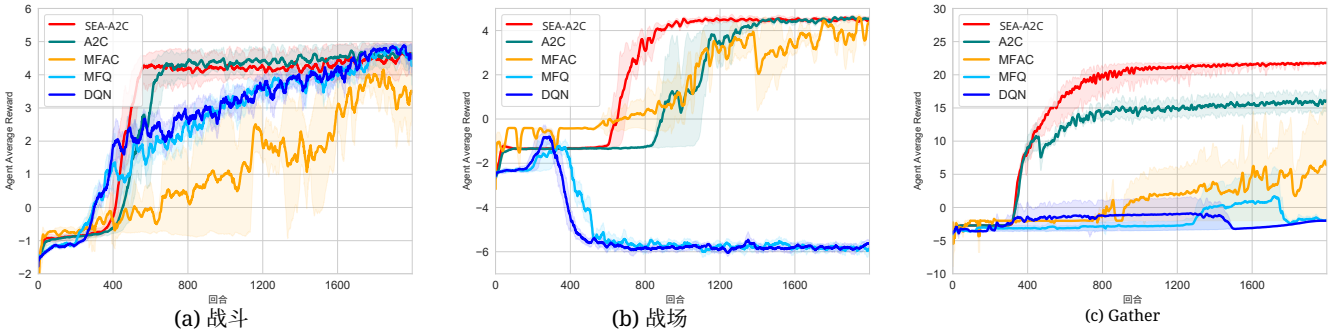
战场 战斗 和战场 的设置基本相同，但战场 中有一些固定障碍物，使任务变得更加困难。地图大小为 80×80，每支队伍中有 74 个智能体。

Gather 该环境中有数百个智能体和数千块食物。智能体通过吃食物获得奖励。每块食物在受到 5 次攻击后会被吸收。由于食物有限，智能体可能会相互竞争，以垄断食物。

B. 协作导航的结果

在协作导航中，有 N 个智能体和 L 个地标。每个智能体都可以观察其他智能体和地标的相对位置，并根据任一智能体与各地标之间的距离获得集体奖励，同时在彼此发生碰撞时受到惩罚。因此，每个智能体都应学会以更少的碰撞占据更多地标。然而，原始设定 $N = 3$, $L = 3$ 过于简单，难以区分不同模型之间的性能。因此，我们设置了$N = 6$, $L = 3$智能体需要学习如何与其他智能体共享地标。

为回答**问题1**，我们将 SEA 与不同的经典算法相结合，包括 A2C、PPO 和 DDPG。结果如图 2 所示。每个实验均使用五个随机种子进行，阴影区域由所有随机种子的最小值和最大值围成，实线表示所有种子的平均值（所有实验均相同）。结果表明，在不进行任何其他改动的情况下，仅添加空间提取组件，就能使所有算法相较于其原始算法获得显著提升。如图 2(a) 所示，A2C 相当不稳定，在训练阶段，某些随机种子的性能会大幅下降；MAA2C 的评论家网络中包含全局状态，但其表现并未超越 A2C，这可能是因为 MPE 中的全局状态只是将所有智能体的观测简单拼接在一起，从而导致大量冗余



(a) 战斗 (b) 战场 (c) Gather

图 4. 不同算法在战斗、战场 和*Gather*上的智能体平均测试回报。

信息。SEA-A2C 更加稳定，其性能超越了 A2C 和 MAA2C，这意味着 SEA 能够在智能体之间实现更好的信息共享。如图 2(b) 所示，与 A2C 中的情况类似，但 MAPPO 的表现优于 MAA2C；MAPPO 的结果接近 PPO，且更加稳定。最终结果表明，SEA-PPO 相较于 PPO 和 MAPPO 具有显著优势。与上述两种同策略方法不同，在图 2(c) 中，DDPG 的表现不如 MADDPG，而 SEA-DDPG 相较于 MADDPG 和 DDPG 也有很大提升。由于 SEA-DDPG 能够在智能体之间实现信息共享，为了进行公平比较，我们还评估了 MAAC。MAAC 使用注意力机制在智能体之间共享信息，同时也是一种异策略方法。图 2(c) 的结果表明，MAAC 能够更快地取得较高分数，但其表现仍未超越 SEA-DDPG，这可以回答**问题2**。

C. 战斗和战场上的结果

在战斗中，两支队伍需要与对手队伍进行战斗，每支队伍有 N 个智能体。每支队伍的规模可以设置为不同大小。每个智能体可以在每个时间步移动或攻击，并会根据其表现获得奖励。具体而言，每个智能体在攻击对手智能体时会获得奖励，死亡时会受到惩罚。

我们将预训练DQN模型设为固定的对手策略，并在64个智能体的设置下训练所有模型2000个回合。Battle中的主要结果如图4(a)所示。SEA-A2C比所有其他基线方法收敛更快；MFQ和DQN的表现相近。尽管MFAC在开始阶段学习速度更快，但其训练曲线不稳定，最终奖励低于其他方法。由于Battle场景相当简单，各算法之间的差异较小。

由于战场中存在一些障碍物，因此战场比战斗更加复杂，这使得智能体学习更加困难。由于地图更大，我们将74个智能体设为主要实验配置。图4(b)中的结果表明，在更复杂的场景中，SEA-A2C的收敛速度快于所有其他方法，并且比其他方法更加稳定。DQN和MFQ起初获得了相对较高的分数。然而，由于那些更加激进的智能体或位于群体前方的智能体会最先死亡，它们的样本规模会更小，因此模型更容易学习到被动策略，即停留不动或躲避，最终收敛到低得多的奖励。MFAC在初期表现良好，但训练过程不稳定，收敛速度也较慢。

*D. Gather*中的结果

在*Gather*中，有数百个智能体和数千个食物块。智能体通过进食获得奖励。食物需要经过五次攻击才能被吸收。食物数量有限，因此智能体可以与其他智能体合作以更有效地获取食物，或攻击其他智能体来垄断食物。

a) 主要结果： *Gather* 中的主要结果如图4(c)所示。这里我们采用495个智能体和1847个食物的默认设置，但将时间步数量限制在较小范围内，以便智能体能更多地专注于吸收食物。SEA-AC收敛到远高于其他基线方法的奖励，并实现了更加稳定的训练曲线。

为了进一步探究 SEA-A2C 能在奖励方面取得如此优势的原因，我们运行了 10 场测试游戏，对 SEA-A2C 和其他基线方法进行评估，每场游戏展开 200 步。表 I 展示了各算法的详细信息，包括总奖励、死亡数量、吸收的食物以及吸收量与死亡数之比。对于 DQN 和 A2C 等独立方法，它们会尽可能获取食物，并杀死其他智能体以垄断食物；或者变得更加消极，不与其他智能体争夺食物。因此，智能体死亡数与吸收的食物之间存在权衡。SEA-A2C 的吸收量与死亡数之比最高，且几乎吸收了所有食物，这证明 SEA-A2C 能够有效平衡这种权衡。A2C 和 MFAC 能在有限时间内吸收约 70% 的食物，但需要牺牲更多智能体；MFQ 和 DQN 则走向另一个极端，过于害怕杀戮，因此只能吸收少量食物。

b) 案例研究与可视化： 为了展示实验结果中SEA-A2C优势的具体来源，我们在图5中将测试结果的过程进行了可视化。如实验所示，在与初始设置相同的规模下，由于智能体在学习过程中共享全局和局部信息，SEA-A2C更倾向于

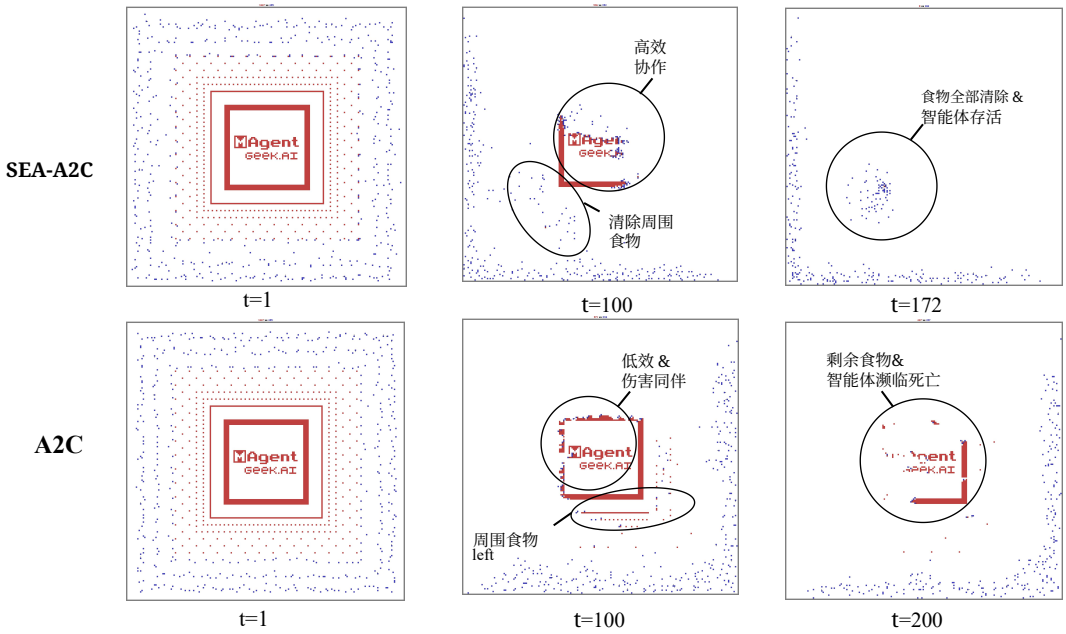


图 5。SEA-A2C 和 A2C 在 InGather场景中的分析与可视化。时间步的最大限制为 200。从左到右分别是每种算法从回合开始、中段到结束的时间线。

表 I 收集结果的详细信息

算法	奖励	吸收	死亡数	比例
SEA-A2C	**10911**	**1845**(99%)	217	**8.50**
A2C	7679	1302(70%)	272	4.78
MFAC	7150	1361(73%)	288	4.72
MFQ	3101	538(29%)	**86**	6.25
DQN	766	364(20 %)	97	3.75

与其他智能体合作，而不是相互攻击。当 t=100 时，大量SEA-A2C智能体在中心进行高效的协作采矿，外围智能体则清理角落处散落的食物。使用A2C算法在地图中心采集食物的智能体因想要垄断食物而相互攻击，导致效率低下。SEA-A2C算法在第172步时提前完成了所有食物收集任务，但A2C算法没有完成

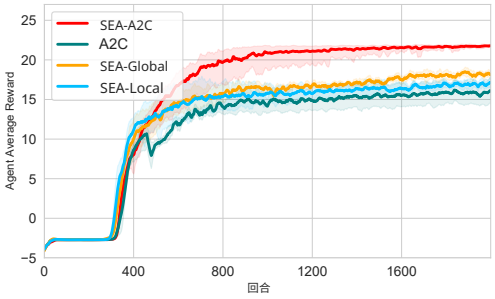


图 6. Gather 上的消融研究。

收集食物直至第 200 步，而中间的大多数智能体都因互相残杀而死亡。由于我们遵循集中训练分散执行（CTDE）设置，即每个智能体在执行期间不共享信息，且只有局部观测，边缘上的智能体在视野中看不到食物信息，因此无论是 SEA-A2C 还是 A2C算法，它们都只能在边缘徘徊。

c) 消融研究：由于 SEA 能够提取智能体之间的局部和全局信息，本节中我们进行了一项消融研究，进一步探究局部和全局信息对 SEA 性能的影响（**Q3**）。我们设计了两个变体，分别只提取局部信息或全局信息。在 SEA-Global 中，我们将所有点归为一个簇，因此该簇包含所有点的聚合信息。在 SEA-Local 中，我们移除了高层分组，使 SEA 只在附近的智能体之间共享信息。图 6 的结果表明，全局和局部信息都能提升 A2C 的表现，但提升幅度很小。与此同时获取全局和局部特征时，模型性能则能得到大幅提升。

VI. 结论与未来工作

本文提出了一种在集中训练分散执行范式下用于多智能体强化学习的空间显式框架，称为 SEA。我们将 SEA 与多种多智能体算法相结合，并在多个环境中评估其性能。结果表明，在训练过程中，SEA 能够有效地在智能体之间共享局部和全局隐藏信息，尤其是在智能体规模较大且任务复杂时。据我们所知，SEA 是首项将计算机视觉中的点云问题与多智能体决策问题相结合的研究。未来，我们计划进一步探讨 SEA 在非集中训练分散执行设置下的性能。

参考文献

[1] C. Berner、G. Brockman、B. Chan、V. Cheung、P. Debiak、C. Dennison、D. Farhi、Q. Fischer、S. Hashme、C. Hesse、R. J´ozefowicz、S. Gray、C. Olsson、J. Pachocki、M. Petrov、H. P. de Oliveira Pinto、J. Raiman、T. Salimans、J. Schlatter、J. Schneider、S. Sidor、I. Sutskever、J. Tang、F. Wolski 和 S. Zhang，《使用大规模深度强化学习进行 Dota 2》，CoRR，第 abs/1912.06680 卷，2019 年。[2] O. Vinyals、I. Babuschkin、W. M. Czarnecki、M. Mathieu、A. Dudzik、J. Chung、D. H. Choi、R. Powell、T. Ewalds、P. Georgiev、J. Oh、D. Hor- gan、M. Kroiss、I. Danihelka、A. Huang、L. Sifre、T. Cai、J. P. Agapiou、M. Jaderberg、A. S. Vezhnevets、R. Leblond、T. Pohlen、V. Dalibard、D. Budden、Y. Sulsky、J. Molloy、T. L. Paine、C.. G¨ulc.ehre、Z. Wang、T. Pfaff、Y. Wu、R. Ring、D. Yogatama、D. W¨unsch、K. McKinney、O. Smith、T. Schaul、T. P. Lillicrap、K. Kavukcuoglu、D. Hassabis、C. Apps 和 D. Silver，《使用多智能体强化学习达到星际争霸 II 大师级水平》，Nat.，第 575 卷，第 354 页，2019 年 7782，350 no. pp. – 。[3] T. Chu、J. Wang、L. Codec`a 和 Z. Li，《用于大规模交通信号控制的多智能体深度强化学习》，IEEE Trans. Intell. Transp. Syst.，第 21 卷，第 3 期，第 1086–1095 页，2020 年。[4] S. Shalev-Shwartz、S. Shammah 和 A. Shashua，《面向自动驾驶的安全多智能体强化学习》，CoRR，第 abs/1610.03295 卷，2016 年。[5] M. Lanctot、V. F. Zambaldi、A. Gruslys、A. Lazaridou、K. Tuyls、J. P´erolat、D. Silver 和 T. Graepel，《多智能体强化学习的统一博弈论方法》，载于《神经信息处理系统进展 30：2017 年神经信息处理系统年度会议》，2017 年 12 月 4–9 日，美国加利福尼亚州长滩，I. Guyon、U. von Luxburg、S. Bengio、H. M. Wallach、R. Fergus、S. V. N. Vishwanathan 和 R. Garnett 编，2017 年，第 4190–4203 页。[6] L. Bus.oniu、R. Babu˘ska 和 B. De Schutter，《多智能体强化学习：概述》。柏林、海德堡：Springer Berlin Heidelberg，2010 年，第 183–221 页。[7] Y. Yang、R. Luo、M. Li、M. Zhou、W. Zhang 和 J. Wang，《均值场多智能体强化学习》，载于《第 35 届国际机器学习会议论文集》，ICML 2018，Stock- holmsm¨assan，瑞典斯德哥尔摩，2018 年 7 月 10–15 日，《机器学习研究论文集》，J. G. Dy 和 A. Krause 编，第 80 卷，PMLR，2018 年，第 5567–5576 页。[8] H. Mao、W. Liu、J. Hao、J. Luo、D. Li、Z. Zhang、J. Wang 和 Z. Xiao，《邻域认知一致的多智能体强化学习》，载于《第三十四届 AAAI人工智能会议》，AAAI 2020，《第三十二届人工智能创新应用会议》，IAAI 2020，《第十届 AAAI人工智能教育进展研讨会》，EAAI 2020，美国纽约，2020 年 2 月 7–12 日。AAAI Press，2020 年，第 7219–7226 页。[9] T. Zhang、Q. Ye、J. Bian、G. Xie 和 T. Liu，《MFVFD：一种用于合作任务和非合作任务的多智能体 Q学习方法》，载于《第三十届国际人工智能联合会议论文集》，IJCAI 2021，线上会议/加拿大蒙特利尔，2021 年 8 月 19–27 日，Z. Zhou 编。ijcai.org，2021 年，第 500–506 页。[10] J. Jiang、C. Dun 和 Z. Lu，《用于多智能体协作的图卷积强化学习》，CoRR，第 abs/1810.09202 卷，2018 年。[11] S. Sukhbaatar、A. Szlam 和 R. Fergus，《通过反向传播学习多智能体通信》，载于《神经信息处理系统进展 29：2016 年神经信息处理系统年度会议》，2016 年 12 月 5–10 日，西班牙巴塞罗那，D. D. Lee、M. Sugiyama、U. von Luxburg、I. Guyon 和 R. Garnett 编，2016 年，第 2244–2252 页。[12] P. Peng、Q. Yuan、Y. Wen、Y. Yang、Z. Tang、H. Long 和 J. Wang，《用于学习进行星际争霸战斗游戏的多智能体双向协调网络》，CoRR，第 abs/1703.10069 卷，2017 年。[13] L. Han、P. Sun、Y. Du、J. Xiong、Q. Wang、X. Sun、H. Liu 和 T. Zhang，《电子游戏人工智能中的多智能体强化学习网格级控制》，载于《第 36 届国际机器学习会议论文集》，ICML 2019，2019 年 6 月 9–15 日，美国加利福尼亚州长滩，《机器学习研究论文集》，K. Chaudhuri 和 R. Salakhutdinov 编，第 97 卷。PMLR，2019 年，第 2576–2585 页。[14] C. R. Qi、H. Su、K. Mo 和 L. J. Guibas，《PointNet：点集上的三维分类与分割深度学习》，CoRR，第 abs/1612.00593 卷，2016 年。[15] C. R. Qi、L. Yi、H. Su 和 L. J. Guibas，《PointNet++：度量空间中点集的深度层次特征学习》，载于《神经信息处理系统进展》信息处理系统*30*：神经信息处理系统年度会议，*2017*年*12*月*4*日至*9*日，美国加利福尼亚州长滩，I. Guyon、U. von Luxburg、S. Bengio、H. M. Wallach、R. Fergus、S. V. N. Vishwanathan 和 R. Garnett 编，2017年，第5099–5108页。[16] R. Lowe、Y. Wu、A. Tamar、J. Harb、P. Abbeel 和 I. Mordatch，“混合协作竞争环境中的多智能体演员-评论家”，载于神经信息处理系统进展*30*：神经信息处理系统年度会议*2017*，*2017*年*12*月*4*日至*9*日，美国加利福尼亚州长滩，I. Guyon、U. von Luxburg、S. Bengio、H. M. Wallach、R. Fergus、S. V. N. Vishwanathan 和 R. Garnett 编，2017年，第6379–6390页。[17] J. N. Foerster、G. Farquhar、T. Afouras、N. Nardelli 和 S. Whiteson，“反事实多智能体策略梯度”，载于第三十二届*AAAI*人工智能会议、第三十届创新人工智能应用会议（*IAAI-18*）以及第八届*AAAI*人工智能教育进展研讨会（*EAAI-18*）论文集，美国路易斯安那州新奥尔良，*2018*年*2*月*2*日至*7*日，S. A. McIlraith 和 K. Q. Weinberger 编。AAAI出版社，2018年，第2974–2982页。[18] C. Yu、A. Velu、E. Vinitsky、Y. Wang、A. M. Bayen 和 Y. Wu，“MAPPO在协作多智能体博弈中的惊人有效性”，*CoRR*，第abs/2103.01955卷，2021年。[19] B. Zhang、Y. Bai、Z. Xu、D. Li 和 G. Fan，“通过超图神经网络高效生成多智能体电子游戏中的协作策略”，*CoRR*，第abs/2203.03265卷，2022年。[在线]。可访问：https://doi.org/10 48550 2203 03265 。/arXiv。。P. Sunehag、G. Lever、A. Gruslys、W. M. Czarnecki、V. F. Zambaldi、M. Jaderberg、M. Lanctot、N. Sonnerat、J. Z. Leibo、K. Tuyls 和 T. Graepel，“用于协作多智能体学习的价值分解网络”，*CoRR*，第abs/1706.05296卷，2017年。[20] T. Rashid、M. Samvelyan、C. S. de Witt、G. Farquhar、J. N. Foerster 和 S. Whiteson，“QMIX：用于深度多智能体强化学习的单调价值函数分解”，载于第*35*届国际机器学习会议，*ICML 2018*，瑞典斯德哥尔摩¨，*2018*年*7*月*10*日至*15*日，机器学习研究论文集，J. G. Dy 和 A. Krause 编，第80卷。PMLR，2018年，第4292–4301页。[21] Z. Xu、Y. Bai、D. Li、B. Zhang 和 G. Fan，“SIDE：我推断想要学习的状态”，*CoRR*，第abs/2105.06228卷，2021年。 [在线]。可访问：https://arxiv.org/abs/2105 06228 。Z. Xu、Y. Bai、B. Zhang、D. Li 和 G. Fan，“HAVEN：具有双重协调机制的分层协作多智能体强化学习”，*CoRR*，第abs/2110.07246卷，2021年。 [在线]。可访问：https://arxiv.org/abs/2110 07246 。Z. Xu、D. Li、B. Zhang、Y. Zhan、Y. Bai 和 G. Fan，“将前瞻与想象相结合：基于模型的协作多智能体强化学习”，*CoRR*，第abs/2204.09418卷，2022年。 [在线]。可访问：https://doi.org/10 48550 2204 09418 。/arXiv。。B. Yan、K. Janowicz、G. Mai 和 R. Zhu，“用于地理知识图谱摘要的空间显式强化学习模型”，*Trans. GIS*，第23卷，第3期，第620–640页，2019年。[26] Z. Cao、X. Ma、M. Shi 和 Z. Zhao，“受信息素启发的大规模多智能体强化学习通信框架”，载于人工神经网络与机器学习——*ICANN 2022*——第*31*届国际人工神经网络会议，英国布里斯托尔，*2022*年*9*月*6*日至*9*日，论文集，第二部分，计算机科学讲义系列，E. Pimenidis、P. P. Angelov、C. Jayne、A. Papaleonidas 和 M. Aydin 编，第13530卷。Springer，2022年，第75–86页。[27] L. Zheng、J. Yang、H. Cai、M. Zhou、W. Zhang、J. Wang 和 Y. Yu，“Magent：面向人工集体智能的多智能体强化学习平台”，载于第三十二届*AAAI*人工智能会议、第三十届创新人工智能应用会议（*IAAI-18*）以及第八届*AAAI*人工智能教育进展研讨会（*EAAI-18*）论文集，美国路易斯安那州新奥尔良，*2018*年*2*月*2*日至*7*日，S. A. McIlraith 和 K. Q. Weinberger 编。AAAI出版社，2018年，第8222–8223页。[28] R. Lowe、Y. Wu、A. Tamar、J. Harb、P. Abbeel 和 I. Mordatch，“混合协作竞争环境中的多智能体演员-评论家”，神经信息处理系统（*NIPS*），2017年。