TransfQMix：利用多智能体强化学习问题图结构的 Transformer

Matteo Gallici KEMLG研究组，加泰罗尼亚理工大学，巴塞罗那，西班牙
gallici@cs.upc.edu

Mario Martin KEMLG研究组，加泰罗尼亚理工大学，巴塞罗那，西班牙
mmartin@cs.upc.edu

Ivan Masmitja 海洋科学研究所（ICM），CSIC，巴塞罗那，西班牙
masmitja@icm.csic.es

摘要
协调是多智能体强化学习（MARL）中最具挑战性的方面之一。原因之一是智能体通常彼此独立地选择动作。为了使独立策略组合后能够涌现出协作策略，近期研究主要关注使用集中式函数（CF）学习每个智能体对团队奖励的贡献。然而，智能体和集中式函数所接收的环境表示结构通常被忽视。我们发现，用于描述协调问题的特征可以表示为潜在图结构的顶点特征。在此，我们提出 TransfQMix，这是一种利用 Transformer 发掘这一潜在结构并学习更优协作策略的新方法。我们的 Transformer 智能体针对可观测实体的状态进行图推理。我们的 Transformer Q 混合器从一个更大的图中学习单调混合函数，该图包含智能体的内部状态和外部状态。TransfQMix 的设计可实现完全迁移，这意味着相同的参数可用于控制和训练规模更大或更小的智能体团队。这使得诸如迁移学习、零样本迁移和课程学习等有前景的方法能够在 MARL 中得到应用，从而节省训练时间并导出通用策略。我们报告了 TransfQMix 在 Spread 和星际争霸 II 环境中的性能。在这两种设置下，它都优于最先进的 Q 学习模型，并展现出解决其他方法无法解决的问题的能力。

关键词
多智能体强化学习、Transformer、协调图、迁移学习

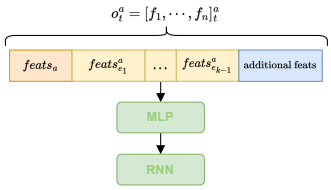
ACM 引用格式：
Matteo Gallici、Mario Martin 和 Ivan Masmitja。2023年。TransfQMix：利用多智能体强化学习问题的图结构的 Transformer。载于预印本版本，已被接收：《第 *22* 届自主智能体与多智能体系统国际会议论文集》（*AAMAS 2023*），英国伦敦，*2023* 年 *5* 月 *29* 日至 *6* 月 *2* 日，IFAAMAS，9 页。

1 引言
为了求解合作式多智能体问题，智能体以协调的方式行动至关重要。深度强化学习（RL）已成功应用于众多多智能体优化任务 [6, 8, 12]。然而，当我们尝试使用强化学习来学习协作策略时，会面临诸多挑战。由于通信约束，部署中央控制器并不切实际。即使允许通信，观测空间和动作空间的规模过大也会造成维数灾难，因此不宜使用单个执行器。因此，智能体应彼此独立地选择动作。为了让独立策略相结合后涌现出协调策略，最先进的多智能体强化学习（MARL）模型会使用一个或多个集中式函数（CFs），以学习智能体的动作对团队目标的贡献。集中式函数能够根据全局团队奖励优化智能体参数。训练完成后，仍可自主部署这些函数，因为每个智能体都负责选择自己的行为。这种方法称为集中训练、分散执行 （CTDE）范式 [4, 7]。

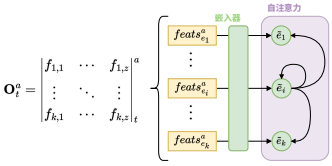
近年来，大多数研究都聚焦于集中训练、分散执行中的混合函数。价值分解网络（VDN） [21], QMix [18], 和 QTran [20] 等方法扩展了传统的 Q 学习算法 [28]，引入中央网络来（学习如何）将智能体的动作价值投影到联合动作的 Q 值上。多智能体深度确定性策略梯度（MADDPG） [11] 和多智能体近端策略优化（MAPPO） [30], 等演员-评论家模型，允许评论家网络在训练期间访问全局观测。较新的方法，如深度隐式协调图（DICG） [9] 和 QPlex [26]，则通过采用多头自注意力和图神经网络改进了混合函数。尽管如此，单个智能体通常仍采用简单的结构，即使用循环神经网络（RNN），并将观测向量输入其中；这些观测向量是由各种类型的特征拼接而成的大型向量（见图 1a）。这种拼接会导致一项关键信息丢失：尽管许多特征对应不同的实体（例如地图上的位置），但它们实际上属于完全相同的类型。

我们的工作表明，观测空间的结构，以及用于部署智能体和混合函数（CF）的架构，在解决复杂协调任务时发挥着重要作用。我们认为，观测向量主要包含潜在图结构的顶点特征；当我们重新审视这些特征如何输入神经网络时，这种结构便会显现。因此，我们不再将许多特征串联起来，生成描述智能体所观测到的世界 状态的向量，而是对一组特征进行泛化，并用它们描述智能体（或混合函数）所观测到的实体状态。我们的方法如图 1b 所示。在这一过程中，我们不会加入任何额外信息

预印本版本，已被接收：第 *22* 届自主智能体与多智能体系统国际会议论文集（*AAMAS 2023*），*A. Ricci*、*W. Yeoh*、*N. Agmon*、*B. An*（编），*2023* 年 *5* 月 *29* 日至 *6* 月 *2* 日，英国伦敦。© 2023年国际自主智能体与多智能体系统基金会（www.ifaamas.org）。版权所有。



(a) 传统方法



(b) 我们的图方法

图 1：传统观测向量与我们的图方法。在传统方法 (a) 中，时间步 t 下智能体 a 的观测向量由与其自身、其他 $k-1$ 实体以及其他元素（例如先前动作）相关的特征拼接而成。在我们的方法 (b) 中，我们仅保留为所有实体定义的 z 特征，以生成协调图的顶点；该图的边通过自注意力机制学习得到。

。相反，有时我们需要移除并非所有观测实体都能获取的数据。这种方法有几个优点：（i）我们可以使用嵌入式前馈网络的同一组权重处理相同的顶点特征，从而降低特征空间的复杂度；（ii）我们可以利用自注意力机制学习潜在协调图的边。具体而言，我们采用 Transformer [24]，事实证明，这种基于图的架构在自然语言处理 [27]、计算机视觉 [29] 领域表现有效，甚至可用于开发通用型智能体 [19]。

我们的 Transformer 智能体在处理特定时间步观测到的实体图后采样动作。我们的 Transformer Q 混合器从一个更大的图中学习单调混合函数，该图包含智能体的内部状态和外部状态。鉴于强化学习问题中存在很强的时间依赖性，我们在智能体和混合器中都加入了循环机制，使前一时间步的嵌入能够影响当前时间步的图推理。由此得到的模型 TransfQMix 具有完全可迁移的优势，这意味着相同的参数可用于控制和训练规模更大或更小的团队。这是因为网络权重构成了一种注意力机制，其不受所应用的顶点数量影响。相反，每当引入新的实体时，传统模型都必须重新训练，因为拼接向量的维度会发生变化，因此必须重新调整网络权重。TransfQMix 的完全可迁移性使得迁移学习、零样本迁移和课程学习成为可能，这些都是迈向更通用的多智能体强化学习模型的关键步骤。

我们在 Spread任务的多个场景中 [11]，以及星际争霸 II (SC2) [25]中最难的地图上测试了 TransfQMix。TransfQMix 在这两种环境中的表现都优于最先进的 Q 学习模型，并且能够解决其他模型无法处理的问题，总体而言，它展现出更快的收敛速度，并能学得更好的协调策略。

本文的贡献如下：

(1) 我们为合作型多智能体强化学习形式化提出了一种新范式，即将协调任务重新视为图嵌入任务。(2) 我们提出了一种新方法 TransfQMix，利用 Transformer 发挥协调图的作用，其表现优于最先进的 Q 学习方法。(3) 我们引入了一种基于图的循环机制，将时间依赖性纳入 Transformer 智能体和混合器中。(4) 我们设计的 TransfQMix 能够处理规模各异的实体图。这使我们得以获得一种更通用的方法，可用于在多智能体强化学习中实现零样本迁移、迁移学习和课程学习。

2 相关工作

近期最先进的方法采用集中训练、分散执行（CTDE）范式解决多智能体强化学习问题 [4, 7]。CTDE 方法已成功应用于基于策略和基于价值的方法 [9, 30, 31]。本文聚焦于采用 CTDE 的基于价值的方法。

在多智能体 Q 学习中有效实现集中训练、分散执行的一个必要条件是：对联合动作进行贪心采样，必须等价于从各个智能体中分别贪心采样动作 [25]。这一原则称为个体—全局最大值（IGM） [20]。VDN 是最早使用集中训练、分散执行将 Q 学习扩展到多智能体强化学习的方法之一 [21]。它采用一个无参数的混合函数，将联合动作的 Q_{tot} 计算为各个智能体动作价值之和。尽管符合 IGM，这种混合函数过于简单，无法有效建模各智能体对 Q_{tot} [18]的贡献。

QMix [18] 表明，要满足 IGM，只需混合函数相对于个体动作价值具有单调性。因此，VDN 的求和函数被多层感知机（MLP）取代。这个混合网络 能够学习复杂的非线性投影，将多个动作价值映射到 Q_{tot}上。其权重由一组以状态为条件的超网络生成 s，并通过绝对值激活函数确保为正。我们提出的方法是 QMix 的改进版本。具体而言，TransfQMix 同样学习一个以 s 为条件的单调混合函数，用于根据个体动作价值生成 Q_{tot}。不过，TransfQMix 在 Transformer 的应用上更为复杂。

先前的方法尝试改进 QMix。OWQMix 和 CWQMix [17] 采用加权机制来学习非单调的混合函数，从而更加重视表现更好的联合动作。QTran [20] 学习了 Q_{tot} 的因子分解，该分解同样不受单调性约束，但它是通过多个多层感知机实现的。QPlex [26] 提出了一种对决结构，在遵循 IGM原则的同时学习非单调的混合函数。值得注意的是，QPlex 和 TransfQMix 一样采用了多头注意力，但仅用于其集中式对决网络的一个子集。所有这些方法都使用了循环神经网络智能体和大型拼接观测向量。尽管在简单的理论框架中展现出显著优势，但放宽单调性约束是否有利于对复杂问题进行建模，仍存在争议

复杂问题 [26]。我们对 QMix 的改进侧重于合作博弈的表示和网络架构，而非单调性。

Transformer 已成功应用于单智能体强化学习 [16], , 但需要修改架构。对于多智能体问题而言，这些调整并非必要，因为将其表示为图问题会更加自然。DeepMind 的通用智能体（Gato） [19] 采用标准的 Transformer，可以解决多种强化学习任务，但尚未在多智能体场景中进行测试。此外，Gato 并非通过强化学习训练，而是采用监督式方法训练。在一种称为通用策略分解Transformer（UPDET） [5], 的方法中，Transformer 被应用于部分 SC2 任务。UPDET 采用了 QMix 框架，但用 Transformer 替换了循环神经网络智能体，并使用了一种策略解耦系统：基于实体的动作（尤其是在 SC2 中攻击特定敌人的 Q 值）由该实体的 Transformer 嵌入生成。该模型在这部分 SC2 任务中表现良好，但文中并未说明如何将其应用于其他多智能体强化学习问题。此外，作者证明，在不采用解耦方法时，使用循环神经网络的 QMix 表现优于使用 Transformer 的 QMix。由于策略解耦并不适用于许多场景，UPDET 似乎只对非常特定的问题有效。

我们的方法将通用框架形式化，表明即使策略解耦并不适用，使用 Transformer 也具有明显优势。TransfQMix 也在中央混合器中采用了 Transformer，而 UPDET 则沿用 QMix 的多层感知机。因此，TransfQMix 是一种完全可迁移的方法。相比之下，UPDET 只能部分迁移，因为每次将智能体应用于新任务时，都必须重新训练混合网络。TransfQMix 采用了与 UP-DET 提出的方法类似的循环图方法。不过，TransfQMix 通过直接从隐藏状态中采样未解耦的动作，更充分地利用了隐藏状态。此外，TransfQMix 也在混合网络中采用了这一循环机制。据我们所知，这是首个在混合函数中引入时间条件化的方法。

[1] 采用了与我们类似的基于实体的图方法，在多智能体强化学习中探索了零样本迁移、迁移学习和课程学习。然而，该方法仅适用于通信问题，而 TransfQMix 旨在成为一种通用的多智能体强化学习方法。

3 背景

合作多智能体任务被形式化为去中心化部分可观测马尔可夫决策过程（Dec-POMDP） [13]。元组 $G = S, U, P, r, Z, O, H, n, \gamma$ 描述了智能体 $a \in A \equiv \{1, \ldots, n\}$，它们在每个时间步从隐藏状态 $h^a \in H$ 中选择一个动作 $u^a \in U$，从而形成联合动作 $\mathbf{u} \in \mathbf{U} \equiv U^n$。这会根据状态转移函数 $P(s' \mid s, \mathbf{u}) : S \times \mathbf{U} \times S \to [0, 1]$使环境发生状态转移，其中 $s \in S$是环境的真实状态。所有智能体共享相同的奖励函数 $r(s, \mathbf{u}) : S \times \mathbf{U} \to \mathbb{R}$，且 $\gamma \in [0, 1)$ 是折扣因子。智能体只能获取环境的部分观测 $z \in Z$，这些观测依据观测函数 $O(s, a) : S \times A \to Z$生成。每个智能体都有一个动作-观测历史 $\tau^a \in T \equiv (Z \times U)^*$，并以此为条件采用随机策略 $\pi^a(u^a \mid \tau^a) : T \times U \to [0, 1]$。联合策略 π 具有联合动作价值函数：$Q^\pi(s_t, \mathbf{u}_t) = \mathbb{E}_{s_{t+1:\infty}, \mathbf{u}_{t+1:\infty}}[R_t \mid s_t, \mathbf{u}_t]$，其中 $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$是折扣回报。

为了找到最优联合动作价值函数 $Q^*(s, \boldsymbol{u}) = r(s, \boldsymbol{u}) + \gamma \mathbb{E}_{s'}[\max_{\boldsymbol{u}'} Q^*(s', \boldsymbol{u}')]$, , 我们使用 Q 学习 [28]，并采用由 θ [23] 参数化的深度神经网络来最小化期望时序差分误差 [26]:

$$\mathcal{L}(\theta) = \mathbb{E}_{(\boldsymbol{\tau}, \boldsymbol{u}, r, \boldsymbol{\tau}') \in D}\left[\left(r + \gamma V\left(\boldsymbol{\tau}'; \theta^-\right) - Q(\boldsymbol{\tau}, \boldsymbol{u}; \theta)\right)^2\right] \tag{1}$$

其中，$V(\boldsymbol{\tau}'; \theta^-) = \max_{\boldsymbol{u}'} Q(\boldsymbol{\tau}', \boldsymbol{u}'; \theta^-)$ 是时序差分目标的一步期望未来回报，θ^- 是目标网络的参数，并将定期使用 θ更新。我们使用缓冲区 D存储转移元组（$\boldsymbol{\tau}, \boldsymbol{u}, r, \boldsymbol{\tau}'$），其中 r是智能体在联合动作-观测历史 $\boldsymbol{\tau}$ 下执行动作 $\boldsymbol{u}$ 并转移至 $\boldsymbol{\tau}'$所获得的奖励。

我们采用单调集中训练分散执行学习范式 [4, 7, 18, 21]。执行过程是分散的，这意味着每个智能体的学习策略仅以其自身的动作-观测历史 τ^a为条件。训练期间，中央混合网络可以访问环境的全局状态 s 以及智能体的隐藏状态 H，以将个体动作价值投影到联合动作的 Q_{tot} 上，并在公式（1）中用于端到端训练模型。对混合函数施加的单调约束与 QMix 所形式化的相同：

$$\frac{\partial Q_{tot}}{\partial Q_a} \geq 0, \forall a \in A \tag{2}$$

这确保了遵循 IGM原则。

我们方法中的神经网络是 Transformer [24], , 大量使用注意力机制 [2]。具体而言，我们使用 Transformer 通过多头自注意力（MHSA） [10, 15, 24]来处理图$^{n \times h}$ 。给定一个嵌入图矩阵$\mathbf{X}^{n \times h}$，其中包含由 h维向量表示的顶点，Transformer 会计算一组查询$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q$、键$\mathbf{K} = \mathbf{X}\mathbf{W}^K$和值$\mathbf{V} = \mathbf{X}\mathbf{W}^V$，其中$\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$是三个不同的参数化矩阵，其维度为 $h \times k$。然后按如下方式计算自注意力：

$$\text{Self-Attention}(\mathbf{X}) = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{n}}\right)\mathbf{V} \tag{3}$$

Transformer 使用 m 个注意力模块并行处理，然后将所有输出拼接起来，并通过最终的$\mathbf{W}^O$ 前馈层将其投影回 h维向量：

$$\begin{aligned} \text{MultiHeadSelfAttn}(\mathbf{X}) &= \mathit{Concat}\left(\text{head}\,1, \cdots, \text{head}_m\right)\mathbf{W}^O \\ \text{where } \mathit{head}_i &= \text{Attention}\left(\mathbf{X}\mathbf{W}_i^Q, \mathbf{X}\mathbf{W}_i^K, \mathbf{X}\mathbf{W}_i^V\right). \end{aligned} \tag{4}$$

4 方法

4.1 图观测与状态

我们的方法重新思考了如何将合作问题呈现给神经网络。为简单起见，我们假设智能体在每个时间步 t观测到 k 个实体，其中 k是环境中的实体总数。在我们的方法中，每个实体由一组 z 特征定义。由于环境具有部分可观测性，不同智能体观测到的特征值可能不同。

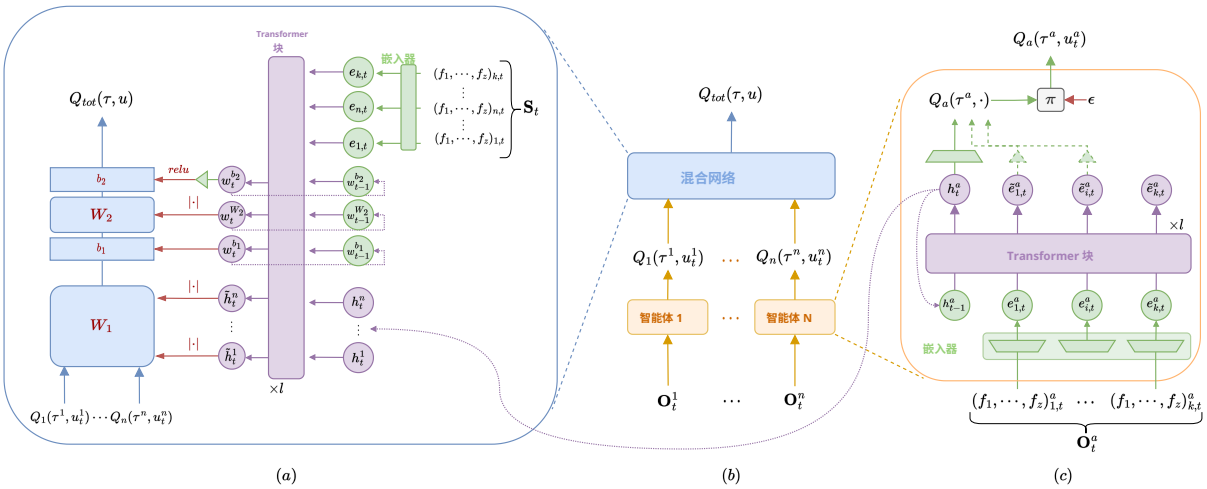


图 2：（a）Transformer 混合器。（b）TransfQMix 总体架构。（c）Transformer 智能体。紫色虚线表示循环连接。绿色组件是简单的前馈层（嵌入器和标量投影器），绿色圆圈是嵌入顶点。紫色圆圈是变换后的顶点。绿色虚线组件表示动作解耦机制。

因此，

$$ent^a_{i,t} = [f_1, \cdots, f_z]^a_{i,t} \tag{5}$$

定义了智能体 a 在时间步 t观察到的实体 i。我们将传统的观测向量替换为维度为 $k \times z$ 的观测矩阵，其中包含智能体 a 在 t观察到的所有 k 实体：

$$\mathbf{O}^a_t = \begin{bmatrix} ent_1 \\ \vdots \\ ent_k \end{bmatrix}^a_t = \begin{bmatrix} f_{1,1} & \cdots & f_{1,z} \\ \vdots & \ddots & \vdots \\ f_{k,1} & \cdots & f_{k,z} \end{bmatrix}^a_t \tag{6}$$

这种结构使智能体能够使用形状为 $z \times h$的参数化矩阵 **嵌入** 的同一组权重来处理同类型特征，其中 h是嵌入维度。由顶点嵌入 $[e1, \cdots \ e_k]^{a\top}_t$构成的矩阵 $\mathbf{E}^a_t = \mathbf{O}^a_t$**嵌入**^{a}将由 Transformer 进一步处理。请注意，**嵌入**^{a}与 k无关。相比之下，RNN 智能体所用的编码前馈层约有 $k \times z \times h$个参数。因此，我们的方法在实体数量方面具有更好的可扩展性和可迁移性。

我们所研究的协作环境中的观测向量 [11, 20, 25]已经包含了隐式的矩阵结构，或者只需进行极少修改即可采用这种结构。相对地图位置、速度、剩余生命值等特征通常会为所有实体定义，然后拼接到同一个向量中，因此可以很容易地重新理解为观测矩阵的顶点特征。另一方面，智能体的上一个动作的独热编码或智能体 ID 的独热编码等特征则需要进行额外处理。此外，由于在我们的方法中，同类型特征由相同的权重处理，我们会丢失拼接向量中隐含的位置信息。实际上，传统编码器可以学习到某些特定向量位置中的特征与某些特定实体相关，因此会区别于其他特征对其进行处理。

在初步研究中，我们发现可以通过使用两个额外的二元特征来弥补这些缺陷。第一个特征 IS_SELF 用于指示所描述的实体是否为该观测矩阵所属的智能体：

$$f^a_{i,\texttt{IS_SELF}} = \begin{cases} 1, & \text{if } i = a \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

对于$ent^a_{a,\ t}$ 该特征取值为 1；对于所有其他实体，取值为 0。IS_SELF 可以看作是对智能体 ID 独热编码的一种重新适配，这种编码通常用于最先进的模型中 [18, 26, 30]。第二个特征用于指示所描述的实体是否为合作智能体：

$$f^a_{i,\texttt{IS_AGENT}} = \begin{cases} 1, & \text{if } i \in A \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

这样便可以区别对待队友的顶点特征与其他顶点的特征。尽管最先进的方法并不总是包含这一特征，但我们认为这并未使用额外数据，因为这些信息已通过向量位置隐式编码。

我们将智能体的观测以相同方式重新表述，并应用于全局状态。通常，状态定义为与实体相关的一组“真实”特征向量（即智能体无法部分观测到的特征），和/或所有智能体观测的拼接。在我们的方法中，我们将状态定义为一个维度为 $k \times z$ 的状态矩阵 $\mathbf{S}_t$：

$$\mathbf{S}_t = \begin{bmatrix} ent_1 \\ \vdots \\ ent_k \end{bmatrix}_t = \begin{bmatrix} f_{1,1} & \cdots & f_{1,z} \\ \vdots & \ddots & \vdots \\ f_{k,1} & \cdots & f_{k,z} \end{bmatrix}_t \tag{9}$$

它从全局视角定义了所有实体的顶点特征。为简洁起见，在记号中我们假设在S 和O中使用相同的 z 特征。不过，我们也可以使用不同的特征。例如，在S 中加入 IS_SELF 并无意义，因为这些特征并非相对于任何智能体定义的，而且

实际上，在我们的实验中，我们确实从S中排除了 IS_SELF。在我们考虑的环境中，状态向量呈现出一种结构，可以轻松地重塑为公式 (9) 中的形式。至于O，我们可以通过参数化矩阵$\mathbf{Emb}^s$并行处理相同的特征类型，从而获得嵌入顶点，这些顶点还可以进一步由 Transformer 处理，即$\mathbf{E}_t = [e1, \cdots e_k]_t^\top = \mathbf{S}_t\mathbf{Emb}^s$。

4.2 Transformer 智能体

我们的 Transformer 智能体以嵌入顶点 E $[e_1, \quad e_k]_t^{a\top}$ h_{t-1}^a $_t^a =$ $\cdots$

和一个隐藏向量作为输入，该向量与任意向量 e_i^a大小相同，并在回合开始时填充为 0。最终的输入矩阵为 $\mathbf{X}_t^a = [h_{t-1}^a, e_{1,t}^a, \cdots e_k^a, {}_t]^\top$。Transformer 块的˜输出为 $l\,\mathbf{X}_t^a = \text{MultiHeadSelfAttn}(\mathbf{X}_t^a)$，这是一个精化图，其中所有顶点都根据它们对其他顶点的注意力进行了调整。具体而言，$h_t^a = \tilde{h}_{t-1}^a$ 可以视为根据实体新状态所获得的注意力，对智能体隐藏状态进行的变换。与自然语言处理中的方法类似，在自然语言处理中，第一个词元（ [CLS] ，位于 BERT [3]中）的变换被认为能够编码整个句子，我们认为 h_t^a编码了智能体的整体协作推理。因此，我们使用维度为 $h \times u$ 的前馈层 $\mathbf{W}^u$，从 h_t^a 中采样智能体的动作价值，其中 u为动作数量：$Q_a(\tau^a, \cdot) = h_t^a\mathbf{W}^u$。最后，我们将 h_t^a传递到下一时间步，使智能体能够递归地更新其协作推理。当某些智能体的动作与观测到的某些实体直接相关时（例如，在星际争霸 II 中的“攻击敌人 i”），我们的 Transformer 智能体会使用一种与 [5]中提出的方法类似的解耦机制。具体而言，与实体相关的动作的动作价值由相应的实体嵌入导出。在这种情况下，会使用一个维度为 $h \times 1$ 的额外前馈矩阵 $\mathbf{W}$ ˆu。例如，攻击敌人 i 的 Q 值采样为 $\tilde{e}_{i,t}^a\mathbf{W}$ ˆu。随后，将与实体无关的动作和与实体相关的动作的 Q 值拼接起来，得到 $Q_a(\tau^a, \cdot)$。

4.3 Transformer 混合器

与 QMix 完全相同，TransfQMix 使用一个多层感知机，将单个智能体采样动作的 Q 值 Q_A映射为联合采样动作的 Q 值Q_{tot}。形式化地：

$$Q_{tot} = (Q_A^{(1\times n)}\mathbf{W_1}^{(n\times h)} + \mathbf{b_1}^{(1\times h)})\mathbf{W_2}^{(h\times 1)} + \mathbf{b_2}^{(1\times 1)} \tag{10}$$

其中，$\mathbf{W_1}$、$\mathbf{b_1}$ 和$\mathbf{W_2}$、$\mathbf{b_2}$ 分别是隐藏层和输出层的权重与偏置。我们在方括号中明确标出了公式 10 的维度，以表明只有三个值与之相关：n，即智能体的数量；h，即隐藏维度；以及 1，它表示 Q_{tot}是标量。这说明，为了构建多层感知机混合器，我们需要 $n + 2$ 个大小为 h 的向量，外加一个标量。

QMix 使用 4 个 MLP 超网络生成这些向量。我们提出使用 Transformer 的输出生成混合器的 MLP 权重。我们的 Transformer 混合器的输入图为：

$$\mathbf{X}_t = \left[h_t^1, \cdots, h_t^n, w_{t-1}^{\mathbf{b}_1}, w_{t-1}^{\mathbf{W}_2}, w_{t-1}^{\mathbf{b}_2}, e_{1,t}, \cdots, e_{k,t}\right]^\top \tag{11}$$

其中，h_t^1、$\cdots$ h_t^n是智能体的 n隐藏状态；$w\mathbf{b}_{t-1}^{\ 1}$、$w\mathbf{W}_{t-1}^{\ 2}$、$w\mathbf{b}_{t-1}^{\ 2}$是三个循环向量，在回合开始时均以 0 填充；$e_{1、t}$、$\cdots$ $e_{k、t}$是嵌入状态，即 $\mathbf{E}_t =$ $\mathbf{S}_t\mathbf{Emb}^s$。输出是一个矩阵 ˜$\mathbf{X}_t = \text{MultiHeadSelfAttn}(\mathbf{X}_t)$，其中包含由多头自注意力机制变换后的 $\mathbf{X}_t$的相同顶点。具体而言，$\tilde{h}_t^1$、$\cdots$ $\tilde{h}_t^n$是智能体的协调推理，它们借助全局信息得到增强，而这些信息是智能体原本无法获取的，即其他智能体的隐藏状态以及环境的真实状态。这些 n 经过细化的向量用于构建 $\mathbf{W_1}$。因此，$Q_A\mathbf{W_1}$ 是将个体 Q 值 Q_A重新投影到智能体隐藏状态的变换上。请注意，个体 Q 值恰恰是智能体根据 h_t^1、$\cdots$ h_t^n 生成（或经其条件化）的。这意味着 Transformer 混合器的主要目标是组合并细化各智能体独立的推理，使其能够体现团队协同。

循环向量的变换后嵌入，$w\mathbf{b}_t^{\ 1} = \tilde{w}\mathbf{b}_{t-1}^{\ 1}$、$w\mathbf{W}_t^{\ 2} = \tilde{w}\mathbf{W}_{t-1}^{\ 2}$、$w\mathbf{b}_t^{\ 2} = \tilde{w}\mathbf{b}_{t-1}^{\ 2}$ 分别用于生成$\mathbf{b_1}$、$\mathbf{W_2}$、$\mathbf{b_2}$。由于$\mathbf{b_2}$ 是一个标量，因此我们在$\mathbf{b}_t^{\ 2}w$上应用了一个尺寸为 $h \times 1$ 的附加参数化矩阵。我们采用循环机制有两个原因：（i）确保 Transformer 混合器完全不受环境中实体数量的影响；（ii）根据问题的 MDP 形式化定义，在集中式训练中纳入时间依赖性。我们认为，Q_{tot} 高度依赖先前状态，因此这种依赖应在混合网络中显式编码。这一循环过程使混合器能够在不同时间步提供更一致的目标，从而使训练更加稳定。

我们采用 QMix 所述的相同策略来满足单调性约束。具体而言，我们对权重$\mathbf{W_1}$ 和$\mathbf{W_2}$ 应用绝对值激活函数，并对$\mathbf{b_2}$应用同样的处理。

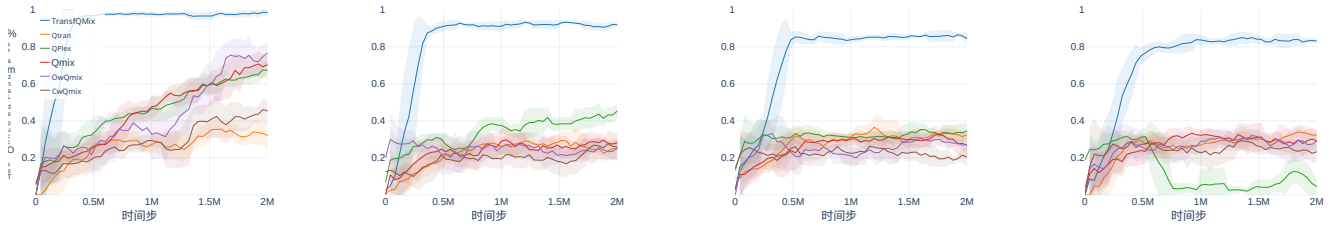
5 实验设置

5.1 Spread

在 Spread 环境中，[11], 智能体的目标是尽可能靠近 n 地标所处的随机位置，同时避免彼此碰撞。智能体可以朝四个方向移动，也可以保持静止。最优策略是让每个智能体各自占据一个地标，从而实现完美的空间分布。由于每个智能体都必须预判其他智能体将占据哪个目标，并前往剩余目标，因此需要进行稳健的协作推理。

全局奖励是每个地标到任一智能体的最小距离的负值。此外，还会增加一项惩罚智能体之间碰撞的奖励。需要注意的是，[11] 实现的原始奖励函数受一个冗余因子的影响，即乘以了 $2n$。之后，PettingZoo [22] 消除了这一冗余；本文使用的正是该奖励函数。

智能体 a 在 Spread 中的观测空间由一个向量组成，其中包含自身的速度和绝对位置，以及所有其他智能体和地标的相对位置。为了将其转换为观测矩阵，我们只保留相对位置，这些位置是 a所观测到的所有实体共有的特征。因此，每个观测到的实体都由 $ent_{i,t}^a = [pos_x, pos_y, \text{IS_SELF}, \text{IS_AGENT}]_{i,t}^a$定义，其中 pos_x和pos_y 分别是实体 i 相对于 a 在水平轴和垂直轴上的相对位置。



(a) 3 个智能体，3 个地标 **(b) 4 个智能体，4 个地标** **(c) 5 个智能体，5 个地标** **(d) 6 个智能体，6 个地标**

图 3：Spread环境中的对比结果。

Spread 的状态空间由所有智能体观测的拼接组成。同样在这种情况下，我们保留所有实体共有的特征，即绝对位置和速度。在最终的状态矩阵中，实体由 $ent_{i,t} = [\hat{}pos_x, \hat{}pos_y, v_x, v_y, \text{IS_AGENT}]_{i,t}$ 定义，其中 $\hat{}pos_x$ 和 $\hat{}pos_y$ 是实体 i 的绝对位置，而 v_x 和 v_y 是其速度（地标的速度为 0）。

Spread的标准报告指标是全局奖励。然而，该指标缺乏参考价值，因为它是一个难以解读的值，并且会随着n 的变化而改变取值范围。因此，我们提出了一项新指标：回合结束时被占据的地标所占的百分比（POL）。计算POL时，我们统计距离某个智能体小于预设阈值的地标数量，再除以地标总数。POL是一个更有参考价值的指标，因为它评估了智能体分布是否合理。此外，当 n 发生变化时，该指标的取值范围仍保持不变（0, 1）。我们发现，当距离阈值设为0.3时，POL与奖励函数的相关系数为0.95，这意味着我们所展示的数据仍可与以往研究进行比较。

5.2 星际争霸 II

该环境采用星际争霸II学习环境 [25]，其中提供了一系列基于知名即时战略游戏《星际争霸 II》[1]的微观管理任务。每项任务都由一个独特的战斗场景组成：一组智能体各自控制一个单位，与由《星际争霸》游戏中央人工智能指挥的一支军队交战。要赢得游戏，智能体必须制定协同行动序列，以便集中注意力攻击特定敌方单位。我们报告了文献中被认为最具挑战性的 8 项任务在星际争霸 II 中的结果 [26, 30]：5m_vs_6m、8m_vs_9m、27m_vs_30m、5s10z、3s5z_vs_3s6z、6h_vs_8z、MMM2 和 corridor。

星际争霸 II 的智能体 a 观测向量由智能体视野范围内友军和敌军的特征串联而成。这些特征包括实体相对于 a的位置、距离、生命值、护盾状态，以及实体类型的独热编码（类型可以是陆战队员、掠夺者、追猎者等）。这种结构已定义了一个观测矩阵，只需添加 IS_SELF 和 IS_AGENT 特征即可用于 TransfQMix。不过，TransfQMix 无法使用原始星际争霸 II 观测向量中的一些额外特征，这些额外特征包括可用动作和先前动作的独热编码，以及地图边界的表示。

我们的 Transformer 混合器可以直接使用星际争霸 II 的原始状态向量作为输入，该向量也是为所有 k 实体定义的特征的拼接。这些特征与观测向量中的特征相同，但以全局视角定义，即相对于地图中心的位置。另一方面，TransfQMix 不使用一个额外特征——所有智能体采取的动作，因为它与图状态方法不兼容。

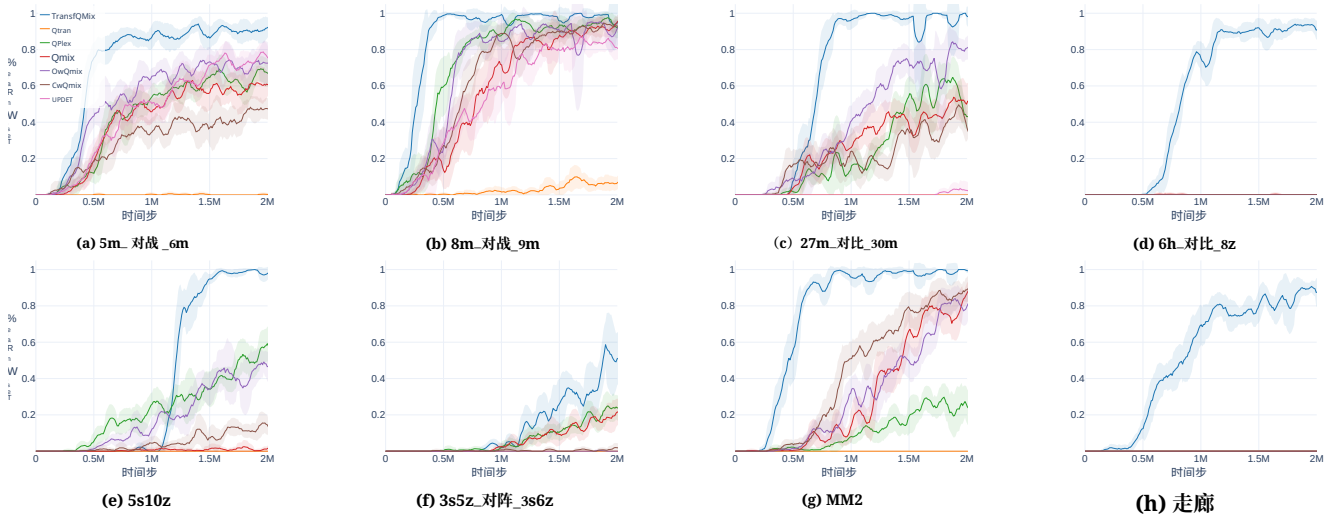
第 4.2 节所述的解耦技术用于 TransfQMix 和 UPDET，即攻击敌方 i 的 Q值由 i的 Transformer 嵌入决定。在适用情况下，对包含治疗另一智能体的动作也采用相同的过程。

5.3 算法

我们的代码库基于 pymarl [4, 18] 构建，可在以下地址获取：https://github.com/mttga/pymarl_transformers。代码库包含 TransfQMix、我们为 Spread 和星际争霸 II 编写的封装，以及本文方法所比较的算法的原始实现：QMix、QTran、QPlex、OW-QMix、CW-QMix 和 UPDET。对于所有比较方法，我们均采用其原始实现中报告的相同超参数。在所有实验中，我们保持每种方法的参数不变。请注意，在所有实验中，我们都使用了参数共享 技术，即所有智能体共享相同的权重。多项研究 [14, 18, 30]已证明该技术非常有益。

我们在星际争霸 II 的 5m_vs_6m 任务中对 TransfQMix 进行了微调，并在所有其他设置（包括 Spread任务）中使用相同的参数。具体而言，Transformer 智能体和混合器的隐藏嵌入维度均设为 32，注意力头设为 4 个，Transformer 块设为 2 个，因此两个网络的参数总数均为 ~ $50k$。所有 Transformer 架构（包括 UPDET）的学习配置都使用 *Adam* 优化器，学习率为 0。001，并使用 λ 为 0。6 来计算双延迟（时序差分）目标值。这一设置不同于最先进的基于 RNN 的模型所采用的设置（*RMSProp* 优化器、0。0005 的学习率，以及时序差分目标的 λ为 0）。不过，我们发现，TransfQMix 的最优学习配置并不适用于其他模型；换言之，这些模型采用各自原有的学习设置时表现更好。所有方法共享一些参数，例如缓冲区大小（5000 回合）、批次大小（32 回合）、目标网络更新间隔（200 回合），以及 epsilon 衰减的退火时间（$100k$ 时间步）。

[1]星际争霸 II 是暴雪娱乐的商标™



(a) 5m_ 对战 _6m　(b) 8m_对战_9m　(c) 27m_对比_30m　(d) 6h_对比_8z

(e) 5s10z　(f) 3s5z_对阵_3s6z　(g) MM2　(h) 走廊

图4：星际争霸 II 环境中的比较结果。

6 结果与讨论

6.1 主要结果

Spread任务中的多智能体强化学习方法通常使用 $n = 3$报告性能。我们将 n增加到 6，以分析这些方法的可扩展性。图 3 展示了各方法在不同场景中训练时，POL 的变化情况。每经过 $40k$ 个时间步，我们就通过运行 30 个独立回合来计算 POL；每个回合中，各智能体均采用贪心的分散式动作选择。在包含 3 个智能体的标准任务中，最先进方法学到了良好的策略，平均覆盖了 ~80% 的地标，QTran 和 CW-QMix 除外（POL 为 ~50%）。然而，它们的表现并未显著优于 QMix。在包含 4 个或 5 个智能体的任务中，唯一能够击败 QMix 的最先进方法是 QPlex（POL 为 50%），但其表现非常不稳定 $n = 6$。相比之下，TransfQMix 在所有场景中的表现都显著优于 QMix 和其他方法，仅用 $500k$ 个时间步，POL 就稳定在接近 90% 的水平。请注意，在 Spread任务中，每种 n 对应的最优策略都相同（即每个智能体占据一个地标）。只有在团队规模较小时，最先进方法才能学会这一策略。另一方面，TransfQMix 展现出更强的智能体—团队规模不变性，在所有场景中都取得了相近的结果。

图4展示了所有方法在《星际争霸 II》最难任务中的结果。报告的指标是在训练期间每 100 个回合进行一次贪心动作采样时，获胜游戏所占的平均百分比。UPDET 的结果仅针对包含陆战队员的任务报告，因为该方法的原始实现不支持其他场景。值得注意的是，UPDET 的表现并不优于基于 RNN 的模型，并且在 27m vs 30m_ _任务中失败了，这表明采用策略解耦的 Transformer 智能体并不一定能带来明显优势。相反，我们对 Transformer 更精巧的运用在每项任务中都显著优于其他模型，并且始终能够击败《星际争霸 II》的中央人工智能，即使是在以往方法无法赢得任何一场游戏的场景中也是如此。TransfQMix 也展现了其有效性

表 1：Spread 中零样本迁移的结果。

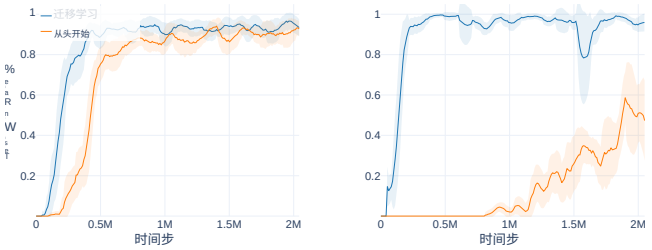
	POL 场景			
模型	3v3	4v4	5v5	6v6
TransfQMix（3 对 3）	**0.98**	0.88	0.8	0.75
TransfQMix（4 对 4）	0.96	**0.93**	**0.9**	0.86
TransfQMix (5v5)	0.88	0.85	0.82	0.82
TransfQMix (6v6)	0.91	0.88	0.85	0.84
TransfQMix（课程学习）	0.88	0.88	0.87	**0.87**
最先进水平	0.76	0.45	0.36	0.33

在实体数量较多的环境中，例如走廊地图（该地图对应 6 个狂热者对战 24 个跳虫，共 30 个实体）以及 27m_vs_30m（57 个实体）。其他方法需要根据实体数量增加参数，而 TransfQMix 的网络在所有任务中的规模几乎相同。这表明，TransfQMix 的架构或许足够通用，无需进行结构调整即可应对各种问题。

6.2 迁移学习

我们通过将特定 Spread 任务中训练得到的网络应用于其他任务，测试了 TransfQMix 的零样本能力。表 1 展示了 TransfQMix 在不同 n的场景中使用 n 个智能体训练后，经过 1000 个回合计算得出的平均 POL。作为基准，文中报告了在每个特定任务中训练的最先进模型所取得的最佳 POL。我们还列出了采用课程学习（CL）方法训练的 TransfQMix 的表现，该方法让智能体在规模逐步扩大的团队中进行协作。具体而言，我们让智能体分别组成 3、4、5 和 6 人团队，每个团队训练 $500k$ 个时间步。

总体而言，除了通过课程学习（CL）训练的智能体在所有场景中的表现都相近外，每个网络都展现出出色的零样本迁移能力，但在规模更大的团队中表现较差。



(a) 8m_vs_9m 对 5m_vs_6m　　(b) 5s10z 到 3s5z_vs_3s6z

图 5：在两个 SC2任务中，迁移学习与从头训练的对比。

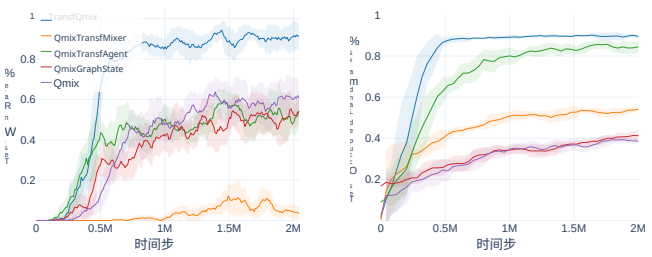
从这个意义上说，课程学习（CL）似乎是利用 TransfQMix 获得通用协作策略的一种有前景的方法。令人意外的是，可迁移性最强的策略是在 4v4 任务中学得的。这可能是因为该场景足够复杂，需要学习强有力的协作策略，却又没有复杂到会导致不稳定或拖慢学习过程。最后，值得注意的是，TransfQMix 的所有零样本迁移表现都优于在各种场景中训练的最先进方法。

在不同情境下使用 TransfQMix 的唯一限制是，顶点特征空间必须相同。在星际争霸 II 环境中，这一点并不总能得到保证，因为单位类型的独热编码特征取决于场景中的单位类型总数。尽管如此，我们仍可以在实体类型相同的地图中使用相同的网络。图 5 展示了分别在 8m_vs_9m 和 5s10z 场景中训练的智能体，经过微调后在 5m_vs_6m 和 3s5z_vs_3s6z 任务中取得的结果。

在这两种情况下，我们都是将较简单设置中学到的协作策略迁移到更困难的任务中，这意味着我们正在进行最小课程学习。可以看出，与从头开始训练相比，微调帮助 TransfQMix 学到了显著更好的策略（图 5b），或更快地收敛（图 5a）。图中的初始峰值对应零样本表现，随后进入下降阶段，此时权重针对新任务进行了快速调整。

总之，结果表明 TransfQMix 具有在不同场景之间迁移知识的良好潜力，也展示了迁移和课程学习如何助力解决复杂的多智能体强化学习任务。

6.3 消融



(a) 星际争霸 II：5m_vs_6m　　(b) Spread

图 6：消融研究。

有人可能会认为，TransfQMix 在 Spread 中取得的结果无法与先前方法相比较，因为我们使用的状态观测矩阵不同于原始状态向量。为检验这一观点，我们在 Spread 中使用状态矩阵的展平版本运行了 QMix。所有 Spread任务的平均测试 POL 如图 6b 所示。可以看出，使用图状态的 QMix（QMixGraphState）与使用原始状态向量的 QMix 相比，性能并无显著差异。同一图还显示，将 QMix混合器的超网络替换为 Transformer 混合器后，性能有所提升（QMixTransformerMixer）。这表明，若要从基于图的状态中获益，就应使用基于图的网络，例如 Transformer。我们还给出了 Transformer智能体与传统 QMix超网络结合时所取得的结果。与基于循环神经网络的原始框架相比，该框架在协调方面明显更胜一筹，但稳定性和性能均不及 TransfQMix。

在星际争霸 II 5m_vs_6m 任务中也进行了相同的消融研究（图 6a）。在这一场景中，仅使用 Transformer 智能体和混合器无法提升 QMix 的性能，这表明我们需要在智能体和混合器网络中都使用 Transformer，才能利用观测和状态的图结构。

7 结论

本文提出了一种新颖的、基于图的多智能体强化学习问题形式化方法，能够以更自然的方式描述协作问题。我们介绍了 TransfQMix，这是一种基于 Transformer 的方法，利用这种结构增强 QMix 中智能体与混合器的协作推理。TransfQMix 展现出强大的学习能力，在无需针对特定任务调整超参数的情况下，便在最具挑战性的星际争霸 II 和 Spread 任务中取得了优异表现。与以往试图改进 QMix 的方法不同，TransfQMix 并不侧重于单调性约束或学习过程的其他方面。这表明，要改进多智能体强化学习方法，就需要更加重视神经网络架构和环境表示。

将 TransfQMix 应用于迁移学习、零样本迁移和课程学习取得了令人鼓舞的结果。在未来的研究中，我们计划将多个任务纳入单一学习流程，以探索该方法的泛化能力。例如，我们计划训练同一组智能体解决所有 SC2 任务。此外，我们还希望研究在不同多智能体强化学习领域之间迁移协作策略的可行性。最后，我们希望更深入地考察多头自注意力对协作推理的影响。

致谢

本项目获得了欧盟地平线2020研究与创新计划资助，资助依据为玛丽·斯克沃多夫斯卡-居里资助协议 No 893089。本工作感谢“SeveroOchoa卓越中心”认证（CEX2019-000928-S）。我们衷心感谢 David and Lucile Packard 基金会。

参考文献

[1] Akshat Agarwal、Sumit Kumar、Katia Sycara 和 Michael Lewis。2020。在多智能体团队中学习可迁移的协作行为。载于第*19*届国际自主智能体与多智能体系统会议论文集（奥克兰，新西兰）（*AAMAS '20*）。1741–1743。[2] Dzmitry Bahdanau、Kyung Hyun Cho 和 Yoshua Bengio。2015。通过联合学习对齐与翻译实现神经机器翻译。第3届国际学习表征会议，ICLR 2015；会议日期：2015年5月7日至2015年5月9日。[3] Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019。BERT：用于语言理解的深度双向 Transformer 预训练。载于*2019*年计算语言学协会北美分会会议论文集：人类语言技术，第*1*卷（长篇与短篇论文）。计算语言学协会，明尼阿波利斯，明尼苏达州，4171 4186 10 18653 1 19 1423–。https://doi.org/。/v/N - [4]

Jakob Foerster、Gregory Farquhar、Triantafyllos Afouras、Nantas Nardelli 和 Shi- mon Whiteson。2018。反事实多智能体策略梯度。载于*AAAI*人工智能会议论文集，第32卷。[5] Siyi Hu、Fengda Zhu、Xiaojun Chang 和 Xiaodan Liang。2020。UPDeT：通过 Transformer 策略解耦实现通用多智能体强化学习。载于国际学习表征会议。[6] Maximilian Hüttenrauch、Sosic Adrian、Gerhard Neumann 等。2019。群体系统的深度强化学习。机器学习研究期刊20，54（2019），1–31。[7] Landon Kraemer 和 Bikramjit Banerjee。2016。将多智能体强化学习作为去中心化规划的演练。神经计算190（2016），82–94。[8] Sergey Levine、Chelsea Finn、Trevor Darrell 和 Pieter Abbeel。2016。深度视觉运动策略的端到端训练。机器学习研究期刊17，1（2016），1334–1373。[9] Sheng Li、Jayesh K Gupta、Peter Morales、Ross Allen 和 Mykel J Kochenderfer。2021。用于多智能体强化学习的深度隐式协调图。载于第*20*届国际自主智能体与多智能体系统会议论文集。764–772。[10] Zhouhan Li n、Minwei Feng、Cicero Nogueira dos Santos、Mo Yu、Bing Xiang、Bowen Zhou 和 Yoshua Bengio。2017。一种结构化自注意力句子嵌入。载于国际学习表征会议。[11] Ryan Lowe、Yi I Wu、Aviv Tamar、Jean Harb、OpenAI Pieter Abbeel 和 Igor Mordatch。2017。混合合作竞争环境中的多智能体演员-评论家。神经信息处理系统进展30（2017）。[12] Volodymyr Mnih、Koray Kavukcuoglu、David Silver、Andrei A Rusu、Joel Veness、Marc G Bellemare、Alex Graves、Martin Riedmiller、Andreas K Fidjeland、Georg Ostrovski 等。2015。通过深度强化学习实现人类水平控制。《自然》518，7540（2015），529–533。[13] Frans A Oliehoek 和 Christopher Amato。2016。去中心化*POMDP*简明导论。施普林格。[14] Georgios Papoudakis、Filippos Christianos、Lukas Schäfer 和 Stefano V Albrecht。2021。在合作任务中对多智能体深度强化学习算法进行基准测试。载于第三十五届神经信息处理系统会议数据集与基准赛道（第一轮）。[15] Ankur Parikh、Oscar Täckström、Dipanjan Das 和 Jakob Uszkoreit。2016。一种用于自然语言推理的可分解注意力模型。载于*2016*年自然语言处理实证方法会议论文集。计算语言学协会，奥斯汀，德克萨斯州，2249–2255。[16] Emilio Parisotto、Francis Song、Jack Rae、Razvan Pascanu、Caglar Gulcehre、Sid- dhant Jayakumar、Max Jaderberg、Raphael Lopez Kaufman、Aidan Clark、Seb Noury 等。2020。《稳定用于强化学习的变换器》。载于国际机器学习大会。PMLR，7487–7498。[17] Tabish Rashid、Gregory Farquhar、Bei Peng 和 Shimon Whiteson。2020。《加权 QMIX：扩展用于深度多智能体强化学习的单调价值函数分解》。神经信息处理系统进展 33（2020），10199–10210。[18] Tabish Rashid、Mikayel Samvelyan、Christian Schroeder、Gregory Farquhar、Jakob Foerster 和 Shimon Whiteson。2018。《QMix：用于深度多智能体强化学习的单调价值函数分解》。载于国际机器学习大会。PMLR，4295–4304。[19] Scott Reed、Konrad Zolna、Emilio Parisotto、Sergio Gomez Colmenarejo、Alexander Novikov、Gabriel Barth-Maron、Mai Gimenez、Yury Sulsky、Jackie Kay、Jost Tobias Springenberg 等。2022。《通用型智能体》。*arXiv* 预印本 *arXiv:2205.06175*（2022）。[20] Kyunghwan Son、Daewoo Kim、Wan Ju Kang、David Earl Hostallero 和 Yung Yi。2019。《Qtran：通过变换学习分解，以实现协作式多智能体强化学习》。载于国际机器学习大会。PMLR，5887–5896。[21] Peter Sunehag、Guy Lever、Audrunas Gruslys、Wojciech Marian Czarnecki、Vinicius Zambaldi、Max Jaderberg、Marc Lanctot、Nicolas Sonnerat、Joel Z Leibo、Karl Tuyls 等。2018。《基于团队奖励的协作式多智能体学习价值分解网络》。载于第 *17* 届国际自主智能体与多智能体系统会议论文集（*AAMAS 2018*），第 3 卷。2085–2087。[22] J Terry、Benjamin Black、Nathaniel Grammel、Mario Jayakumar、Ananth Hari、Ryan Sullivan、Luis S Santos、Clemens Dieffendahl、Caroline Horsch、Rodrigo Perez-Vicente 等。2021。《PettingZoo：用于多智能体强化学习的 Gym》。神经信息处理系统进展 34（2021），15032–15043。[23] Hado Van Hasselt、Arthur Guez 和 David Silver。2016。《结合双重 Q学习的深度强化学习》。载于*AAAI*人工智能会议论文集，第 30 卷。[24] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017。《注意力就是一切》。神经信息处理系统进展 30（2017）。[25] Oriol Vinyals、Timo Ewalds、Sergey Bartunov、Petko Georgiev、Alexander Sasha Vezhnevets、Michelle Yeo、Alireza Makhzani、Heinrich Küttler、John Agapiou、Julian Schrittwieser 等。2017。《星际争霸 II：强化学习的新挑战》。*arXiv* 预印本 *arXiv:1708.04782*（2017）。[26] Jianhao Wang、Zhizhou Ren、Terry Liu、Yang Yu 和 Chongjie Zhang。2021。《QPLEX：双重对决多智能体 Q 学习》。载于第 *9* 届国际学习表征会议（*ICLR 2021*），线上会议，奥地利，*2021*年*5*月*3*日至*7*日。[27] Qiang Wang、Bei Li、Tong Xiao、Jingbo Zhu、Changliang Li、Derek F Wong 和 Lidia S Chao。2019。《面向机器翻译的深度变换器模型学习》。载于计算语言学协会第 *57* 届年会论文集。1810–1822。[28] Christopher JCH Watkins 和 Peter Dayan。1992。《Q 学习》。机器学习 8，3（1992），279–292。[29] Bichen Wu、Chenfeng Xu、Xiaoliang Dai、Alvin Wan、Peizhao Zhang、Zhicheng Yan、Masayoshi Tomizuka、Joseph Gonzalez、Kurt Keutzer 和 Peter Vajda。2020。《视觉变换器：用于计算机视觉的基于词元的图像表示与处理》。*arXiv* 预印本 *arXiv:2006.03677*（2020）。[30] Chao Yu、Akash Velu、Eugene Vinitsky、Yu Wang、Alexandre Bayen 和 Yi Wu。2021。《近端策略优化在协作式多智能体游戏中的惊人效果》。*arXiv* 预印本 *arXiv:2103.01955*（2021）。[31] Meng Zhou、Ziyu Liu、Pengwei Sui、Yixuan Li 和 Yuk Ying Chung。2020。《为协作式多智能体强化学习学习隐式信用分配》。神经信息处理系统进展 33（2020），11853–11864。

TransfQMix：利用多智能体强化学习问题图结构的 Transformer（补充材料）

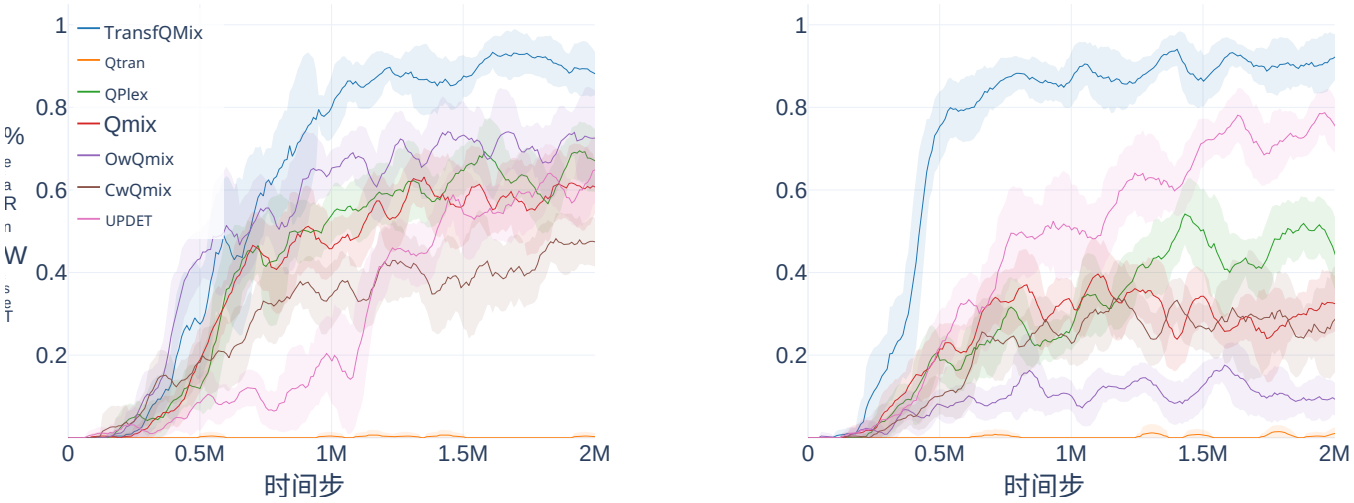
Matteo Gallici KEMLG研究组，加泰罗尼亚理工大学。西班牙巴塞罗那 gallici@cs.upc.edu

Mario Martin，KEMLG研究组，加泰罗尼亚理工大学，西班牙巴塞罗那，mmartin@cs.upc.edu

Ivan Masmitja，海洋科学研究所（ICM），CSIC，西班牙巴塞罗那，masmitja@icm.csic.es

ACM 参考格式：
Matteo Gallici、Mario Martin 和 Ivan Masmitja。2023年。TransfQMix：利用多智能体强化学习问题图结构的 Transformer（补充材料）。载于预印本版本，已被 第 *22* 届自主智能体与多智能体系统国际会议（*AAMAS 2023*），英国伦敦，*2023* 年 *5* 月 *29* 日至 *6* 月 *2* 日论文集，IFAAMAS，共 6 页。

预印本版本，已被以下会议接收：第 *22* 届自主智能体与多智能体系统国际会议论文集（*AAMAS 2023*），*A. Ricci*、*W. Yeoh*、*N. Agmon*、*B. An*（编），*2023* 年 *5* 月 *29* 日至 *6* 月 *2* 日，英国伦敦。© 2023年国际自主智能体与多智能体系统基金会（www.ifaamas.org）。保留所有权利。



(a) 原始优化参数。 **(b) TransfQMix 优化参数。**

图 1：在星际争霸 II 5m_vs_6m 任务中，使用最先进模型通常采用的优化参数（*RMSProp* 优化器、0.0005 学习率，以及用于时序差分的 λ 为 0）和 TransfQMix 的最优优化参数（*Adam* 优化器、0.001 学习率，以及用于时序差分的 λ 为 0.6）所获得的结果。最先进模型无法从 TransfQMix 使用的优化方案中获益。同时，即使使用最先进方法的配置进行训练，TransfQMix 的表现仍优于最先进方法。UPDET 是唯一能从 TransfQMix 的优化器配置中获益的方法，这表明这些参数适用于训练 Transformer 网络。

表 1：TransfQMix 的参数。与 Transformer 相关的参数由 Transformer 智能体和 Transformer 混合器共享。

参数	值	描述
缓冲区大小	5000	用于训练的最近保存回合数
批次大小	32	用于训练的批次大小
更新间隔	200	用于更新目标网络的回合间隔
优化器	Adam	优化器
学习率	0.001	学习率
TD-Lambda	0.6	用于计算 TD 目标的 Lambda
嵌入维度	32	嵌入维度 h
注意力头	4	每个 Transformer 块中的自注意力头
Transformer 块	2	Transformer 层数
Dropout	0	Transformer 块中的 Dropout 百分比
可学习参数	$\sim 50k$	单个网络（混合器或智能体）的可学习参数

表 2：TransfQMix 与其他最先进模型的参数数量（智能体网络和混合器网络）对比。报告了 Spread 3v3、6v6 和 星际争霸 II 27m_vs_30m 中的参数数量，以考察其与环境实体数量之间的关系。TransfQMix 的参数数量不会随实体数量变化。相反，其他方法的参数数量会随实体数量成比例增加，导致其在 星际争霸 II 的 27m_vs_30m 任务中网络过于庞大。总体而言，TransfQMix 是比其他方法更轻量的模型（但在 Spread 3v3 和 6v6 中，QMix 除外）。

模型	智能体	混合器
TransfQMix	50k	50k
QMix	27k	18k
QPlex	27k	251k
O-CWQMix	27k	179k

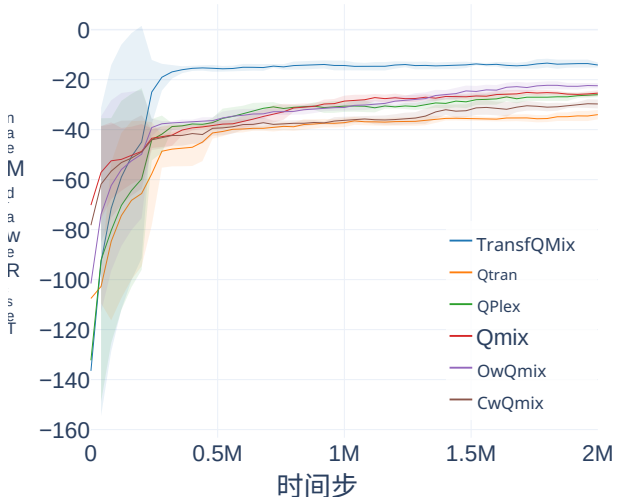
(a) Spread 3v3

模型	智能体	混合器
TransfQMix	50k	50k
QMix	28k	56k
QPlex	28k	597k
O-CWQMix	28k	301k

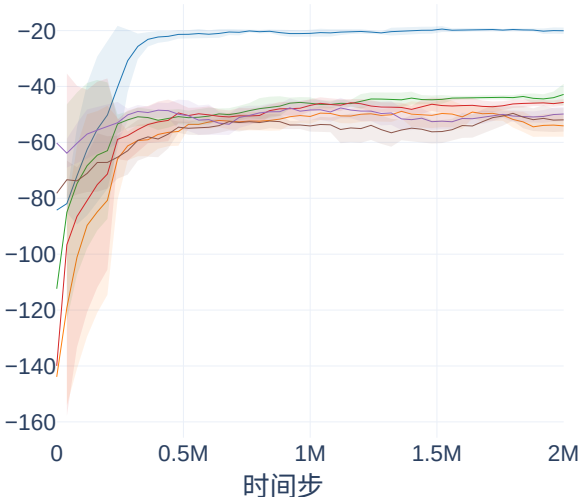
(b) Spread 6v6

模型	智能体	混合器
TransfQMix	50k	50k
QMix	49k	283k
QPlex	49k	3184k
O-CWQMix	49k	1021k

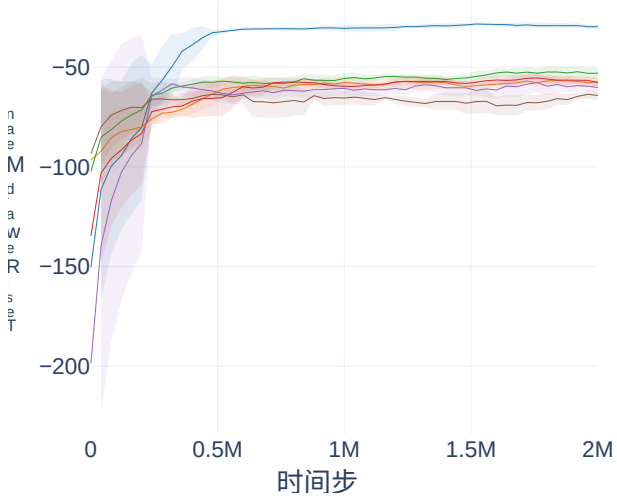
(c) 星际争霸 II 27m_对阵_30m



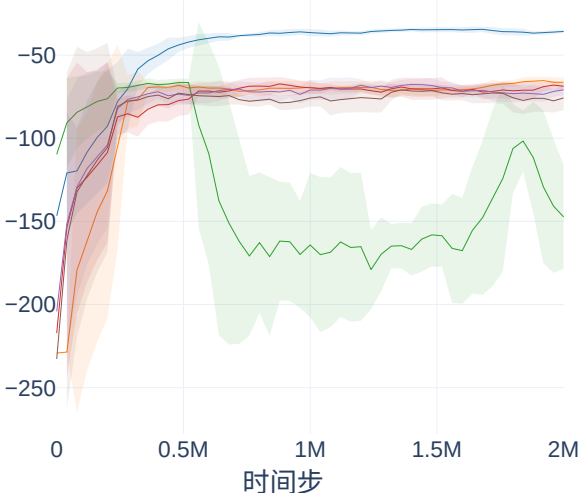
(a) 3 个智能体，3 个地标



(b) 4 个智能体，4 个地标

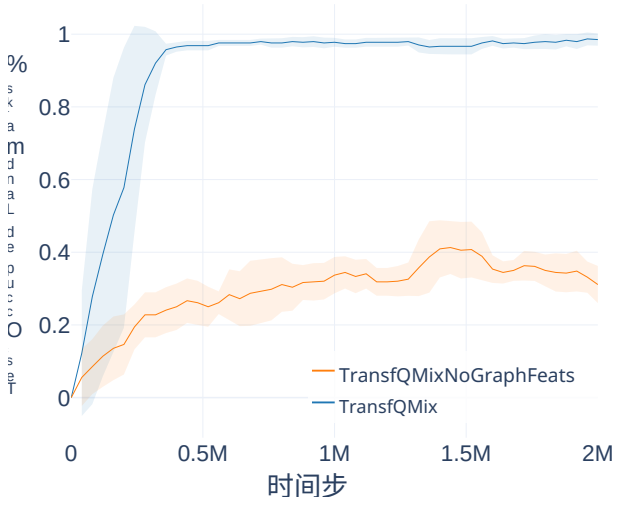


(c) 5 个智能体，5 个地标

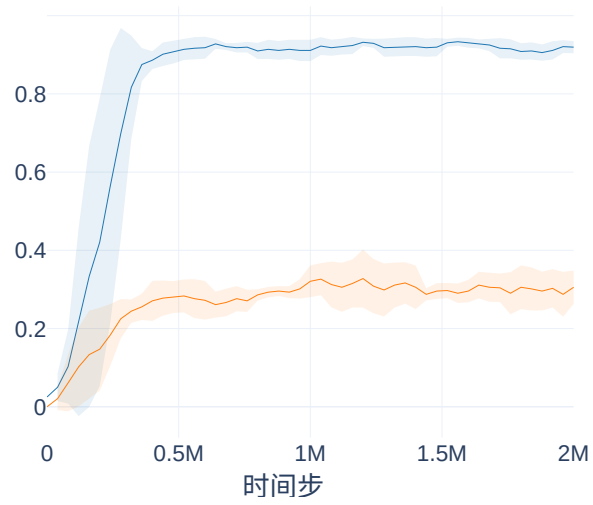


(d) 6 个智能体，6 个地标

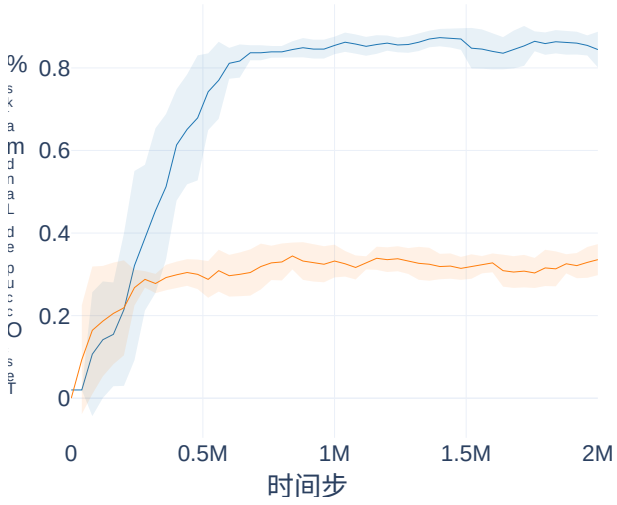
图 2：Spread 中训练期间采用贪心动作选择时的平均奖励。全局奖励为从每个地标到任意智能体的最小距离的负值。我们使用了 PettingZoo 奖励，该奖励相对于原始奖励与 $1/2n$ 成正比。结果与论文中基于 POL 的结果成正比。



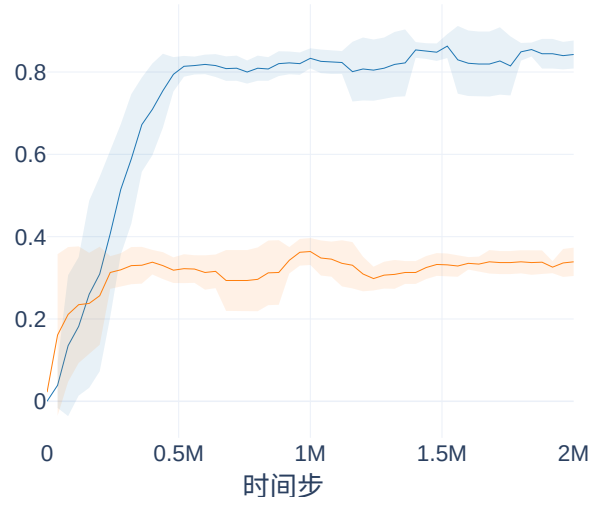
(a) 3 个智能体，3 个地标



(b) 4 个智能体，4 个地标

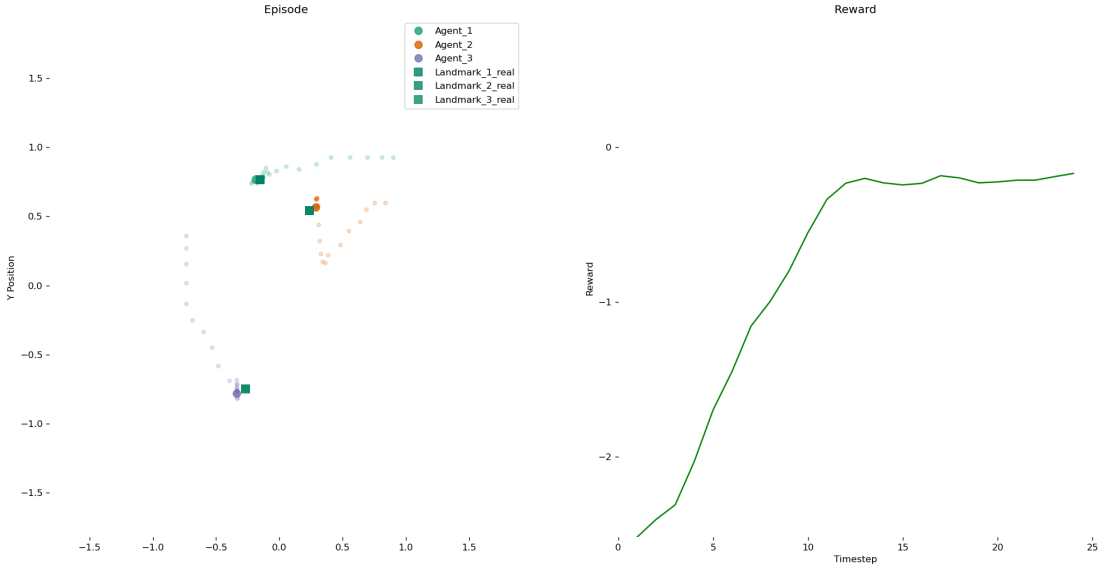


(c) 5 个智能体，5 个地标

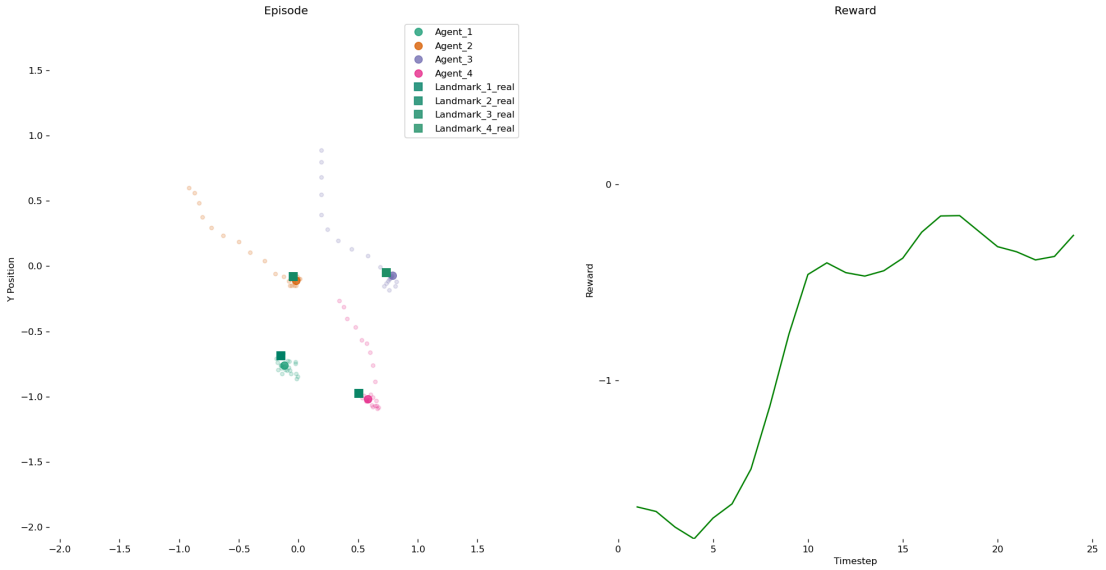


(d) 6 个智能体，6 个地标

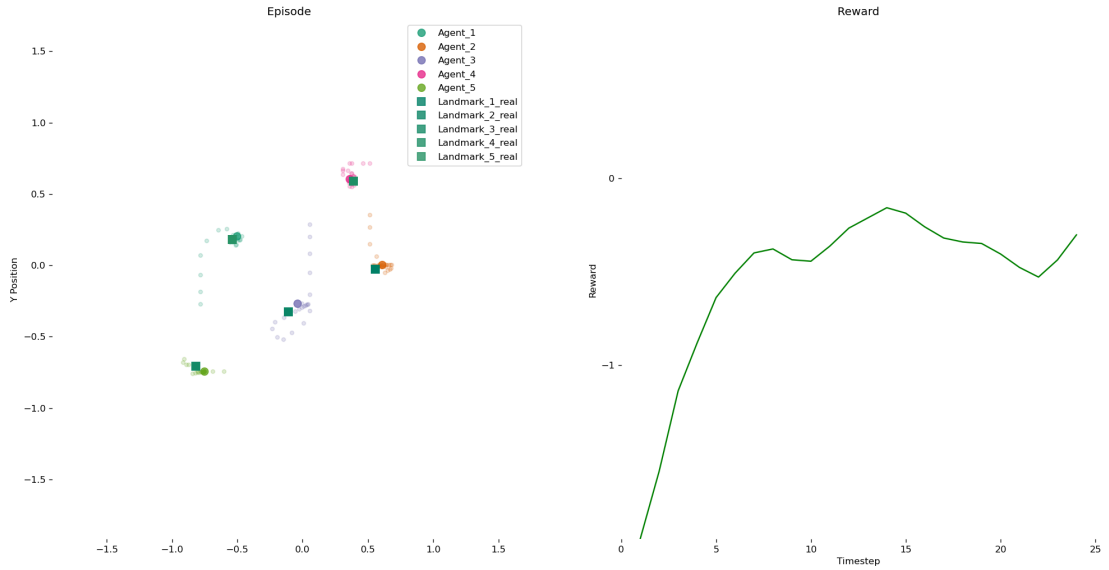
图 3：在训练 TransfQMix 期间，POL 在 Spread任务中执行贪心动作时使用 IS_SELF 和 IS_AGENT 顶点特征（TransfQMix），以及不使用这些特征（TransfQMixNoGraphFeats）的情况。尽管这两个二元特征非常简单，却能让 Transformer 推断哪些实体嵌入相对于当前智能体，哪些相对于其他队友。这似乎对于利用自注意力生成一致的协调图极为重要。



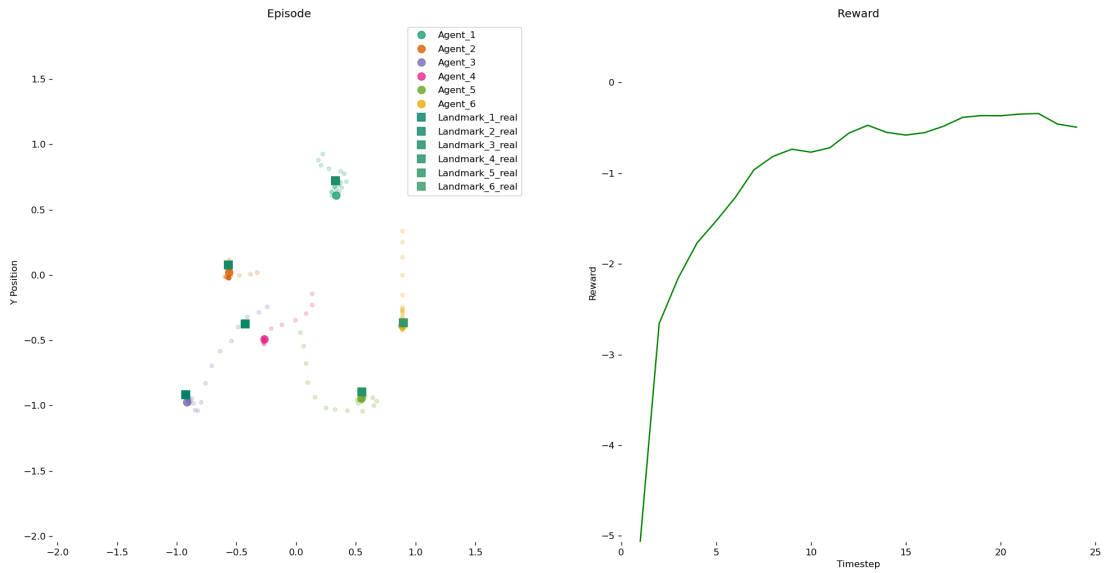
(a) 3 个智能体，3 个地标



(b) 4 个智能体，4 个地标



(c) 5个智能体，5个地标



(d) 6个智能体，6个地标

图4：使用在4对4场景中训练的TransfQMix，在Spread任务中学习到的策略示例。平滑圆圈表示智能体的轨迹。实心圆表示回合结束时智能体的位置。右侧图中的绿色线条表示回合期间全局奖励的变化。