星际争霸2：强化学习新挑战

> SC2LE——星际争霸2学习环境，是一种基于星际争霸2游戏的强化学习环境。是DeepMind和暴雪公司联合进行研究开发的。本文可以看作是对《迈向通用人工智能：星际争霸2人工智能研究环境SC2LE完全入门指南》的扩充，详细讲解了DeepMind发布的StarCraft II Paper。有关代码的GitHub是PYSC2，Blizzard的公布的API是s2client-proto。

概述

星际争霸2等即时策略游戏游戏代表了比以往大多数工作中考虑的更具有挑战性的一类问题，**挑战主要体现在以下几个方面：**

- 它是多智体多玩家交互的问题。
- 由于地图仅仅能够局部观测，信息具有不完整性。
- 巨大的动作空间，玩家可以进行多种动作，这些动作涉及数百个单元的选择和控制。
- 巨大的状态空间，必须仅从原始输入特征面独立地进行观察。
- 此外，由于游戏的回报需要涉及数千个步骤，激励的长期性推迟了策略的信贷分配。

我们描述了对星际争霸2的观察、操作和奖励规范，并提供了一个基于python的开放源码接口，用于与游戏引擎通信。除了主要的游戏地图之外，我们还提供了一套专注于星际争霸II游戏的不同元素的迷你游戏。对于主要的游戏地图，我们还提供了来自人类专家玩家的游戏回放数据的数据集。我们给出了从这些数据中训练出来的神经网络的初始基线结果来预测游戏结果和玩家的行为。最后，我们给出了应用于星际争霸II领域的典型深度控制-强化学习因子的初始基线结果。在小型游戏中，这些代理商学习达到一个水平的游戏，可以与一个新手相比。然而，当训练在主要的游戏时，这些代理人不能做出明显的进步。因此，SC2LE为解释深层强化学习算法和体系结构提供了一个新的、具有挑战性的环境。

简介

深度学习RL（Reinforce Learning）目前进展得如火如荼，它提供了使用神经网络进行非线性函数逼近的强大工具包，语音识别、计算机视觉或自然语言处理等领域的也是进展迅速。强化学习在Atari游戏、围棋游戏、三维虚拟环境和模拟机器人领域都取得了显著的成功。

本文介绍了SC2LE（星际争霸2学习环境），一个新的、更具挑战性的领域强化学习，基于星际争霸2游戏。星际争霸是一种实时策略(RTS)游戏，它将快节奏的微动作与高层次的计划和执行相结合。星际争霸是开创性和持久的电子竞技，拥有高水平职业运动员。因此，击败人类顶级玩家成为一个有意义的、可衡量的长期目标。

星际争霸2的特性决定了它是一个具有挑战性的新领域：

- 它是一个多主体的问题：在这个问题中，多个参与者争夺影响力和资源。
- 它也是一个多代理问题：每个玩家控制数百个单位，需要协作以达成一个共同的目标。
- 这是一个不完整信息的游戏：地图只能通过本地摄像头部分观察到，为了让玩家整合信息，必须主动移动摄像头；“战争迷雾的存在，使玩家无法观察地图上未被访问的区域，有必要积极探索地图以确定对手的状态。
- 行动空间广阔多样：玩家通过点击鼠标界面，在大约108种可能性的组合空间中选择动作。
- 动态的行动空间：随着游戏的进行，合法行为的集合也会发生变化。
- 游戏通常持续成千上万的框架和行动,玩家很早之前决定的后果可能无法被及时体现。导致在游戏信贷分配存在丰富的挑战和探索（Exploitation and Exploration）。

SC2LE提供了一种**更加直观的星际争霸2强化学习接口**：

- 观测和动作是用低分辨率的特征网格来定义的。
- 奖励是基于星际争霸2引擎对电脑对手的分数。
- 除了完整的游戏地图外，还提供了一些简单的迷你游戏。
- （未来的版本将扩展星际争霸2的完整挑战界面：观察和操作将暴露RGB像素；代理商将根据多玩家游戏的最终输赢结果进行排名；评估将被限制在完整的游戏地图用于竞争的人类游戏等等。）

SC2LE提供了一个基于人类玩家的游戏回放的**大型数据集**：当人们玩游戏时，这个数据集将增加到数百万次。我们认为，接口和这个数据集的结合将提供一个有用的基准测试，不仅可以测试现有的和新的RL算法，而且还可以测试感知、记忆和注意力、序列预测和建模不确定性的有趣方面，这些都是机器学习研究的活跃领域。

在最初版本的星际争霸中，已经存在了一些强化学习的环境。但是，它们关注的是新版本的星际争霸2；观察和操作基于UI端，而不是编程端。目前最好的人工星际机器人，基于内置人工智能或之前的环境研究，甚至可以被业余选手打败。相对简陋的开发基础，加之星际争霸有趣的游戏属性和庞大的玩家基础，使它成为探索深度强化学习算法的理想研究环境。

相关工作

计算机游戏为评估和比较标准化任务的不同学习和规划方法提供了一个引人注目的解决方案，也是人工智能研究的一个重要的挑战来源。**计算机游戏对人工智能**

领域的研究有很多优点：

- 他们有明确的客观的成功衡量标准。
- 电脑游戏通常输出丰富的观测数据流，这是深度神经网络的理想输入。
- 它们被外部定义，使人类玩起来困难而有趣。这确保了挑战本身不会被研究者调整，从而使正在开发的算法更容易地解决问题。
- 游戏被设计成可以在任何地方运行，具有相同的界面和游戏机制，使它更容易与其他研究人员精确地共享一个挑战。
- 在某些情况下，会有一群狂热的人类玩家存在，这就有可能对高技能的个人进行基准测试。
- 由于游戏是模拟的，它们可以被精确地控制，并在规模上运行。

也许游戏驱动强化学习研究的最好例子是街机学习环境(ALE)，它允许用Atari视频游戏进行简单和可复制的实验。这种标准化的任务集对人工智能最近的研究来说是一个不可思议的福音。在这种环境下的游戏分数可以通过出版物和算法进行比较，允许直接测量和比较。ALE是AI丰富的视频游戏基准测试传统中的一个突出例子，包括超级马里奥、吃豆人小姐、毁灭战士、玄幻竞技场以及通用视频游戏框架和竞争。

SC2LE

> SC2LE是DeepMind针对与星际争霸2开发的环境，包含三个子组件：一个Linux星际争霸2二进制文件、星际争霸II API和PySC2。PySC2是一个Python环境，它封装了StarCraft II API，以简化智能体们和StarCraft II之间的交互。PySC2定义了一个操作和观察规范，包括一个随机代理和一些脚本化代理作为示例。它还包括一些迷你游戏作为挑战和可视化工具，以了解代理可以看到和做什么。

完整的游戏描述和奖励机制

在星际争霸2的1v1游戏中，两个对手在地图上生成，地图包含资源和其他元素，如坡道、瓶颈和岛屿。要赢得比赛，玩家必须：积累资源(矿物质和瓦斯)、构建生产建筑、组建一支军队、清除对手所有的建筑。一个游戏可能持续几分钟或者一个小时，在游戏中所采取的早期行动(例如，建造哪些建筑物和单位)具有长期的一致性。玩家有不完整的信息，因为他们只能看到地图上有单位的部分。如果他们想了解对手的策略并对其做出反应，他们就必须派遣部队去侦察。之后会描述，动作空间也是非常独特，具有挑战性。

尽管未来可能会考虑2V2、3V3等更加复杂的情况，在这里我们关注1v1格式，最流行的形式的竞争星际争霸。

星际争霸2包含了一个内置的人工智能引擎，它基于一套手工制定的规则，并且有10级难度（较为强大的人工智能引擎可能会通过获得额外的资源或特权的视觉欺骗）。不幸的是，它们都是照本宣科的，策略空间较为狭窄。因此，它们很容易被利用，这意味着人类很快就会对它们失去兴趣。尽管如此，对于我们在第4和5节中所研究的基线这样的纯学习方法来说，它们是一个合理的第一个挑战；它们的表现比随机要好得多，运行速度非常快，几乎不需要计算，并且提供了一致的基线来进行比较。

我们定义**两个不同的奖励结构**：三元奖励结构1/0/−1在游戏的最后分别表示胜利、平局和失败以及暴雪的分数。三元的输赢分数是我们真正关心的奖励。暴雪分数是玩家在游戏胜利时，屏幕上看到的分数。虽然玩家只能在游戏结束时看到这个分数，但是我们提供了游戏中每一步运行的暴雪分数，这样分数的变化可以作为强化学习的奖励。它是计算被研究的当前资源和升级的总和，以及当前存在和正在建设的单元和建筑物。这意味着玩家的累计奖励随着挖掘的资源增加而增加，当失去单位/建筑时减少，所有其他的行为(训练单位、建筑和研究)都不会影响它。暴雪的分数不是零和的，因为它是以玩家为中心的，它比三元奖励信号要少得多，它在一定程度上与赢或输有关。

观测

星际争霸2使用了一个游戏引擎，可以用3D渲染图形。在使用模拟整个环境的底层游戏引擎时，星际争霸II API目前不渲染RGB像素。相反，它生成了一组“特性层”，这些“特性层”从人类游戏中看到的RGB图像中抽象出来，同时保持星际争霸II的核心空间和图形概念(参见图2)。因此,功能层的主要观测是集呈现的N×M像素(其中N和M是可配置的,但在我们的实验中我们总是使用N = M),其中每一层代表特定的游戏,例如:单元类型、生命值、所有者,或可见性。其中一些(例如，命中点，高度图)是标量，而另一些(例如，可见性，单位类型，所有者)是分类的。有两组特征层:minimap是对整个世界状态的粗略表示，屏幕是对屏幕上的播放器的一个分区的详细的视图，其中大多数操作都是执行的。屏幕和小地图都有一些特性(例如，所有者或可见性)，而其他的(例如，单位类型和点击点)只存在于屏幕上。有关所提供的所有观察结果的完整描述，请参阅环境文档6。

除了屏幕和小地图，游戏的人机界面提供了各种非空间观察。这包括收集的气体和矿物的数量，当前可用的操作集(取决于游戏环境，例如，选择了哪些单位)，关于选定单位的详细信息，构建队列和运输工具中的单位。这些观察结果也由PySC2公开，并且在环境文档中有完整的描述。

在整个游戏中，屏幕是用一个高分辨率的全3D透视相机渲染的。这就导致了复杂的观测结果:随着屏幕上的单位变得越来越“高”，单位变得越来越小，而且在屏幕后面可以看到比前面更多的世界房地产。为了简化这一点，特征层是通过使用自顶向下正投影的摄像机呈现的。这意味着一个特征层中的每个像素对应的世界房地产数量是完全相同的，因此，无论在什么位置，所有单元的大小都是相同的。不幸的是，这也意味着特征层呈现并不完全符合人们的预期。经纪人在前面看的多一些，在后面看的少一些。这确实意味着人类在回放中所做的一些行为无法被完全展现。

在未来的版本中，我们将公开一个渲染的API，允许代理从RGB像素播放。这将允许我们研究从原始像素学习与从特征层学习的效果，并与人类游戏进行更密切的比较。与此同时，我们使用特性层来验证代理是否有严重缺陷。虽然游戏经验显然是改变我们发现解决N,M≥64足以让人类玩家选择和单独控制小狗等小单元。我们鼓励读者尝试使用pysc2。

行动

我们设计了环境操作空间来尽可能地模拟人机界面，同时保留了其他RL环境中使用的一些约定，如Atari[4]。图3显示了由播放器和代理生成的动作的简短序列。

游戏中的许多基本动作都是复合动作。例如，要在地图上移动一个选定的单位，玩家必须首先选择按m移动它，然后可能选择按shift来排队动作，然后点击屏幕上的一个点或小地图来执行动作。我们没有让代理生成那3个键/鼠标按按顺序，而是将其作为一个原子复合函数操作:move screen(queued, screen) 。

更正式地说，操作 a 表示为函数标识符a0和函数标识符所需的参数序列的组合a1、a2.....例如，考虑选择多个单位通过绘制一个矩形。我们目标的行为就变成了select rect(select add,(x1,y1),(x2,y2)) 。选择add的第一个参数是二进制的。其他参数是定义坐标的整数——它们的允许范围与观测结果的分辨率相同。这个操作以[select rect，[[select add]，[x1,y1]，[x2,y2]]的形式反馈给环境。为了表示完整的操作空间，我们定义了近300个动作函数标识符，这些标识符包含13种可能的参数类型(从bi- nary到在离散2D屏幕上指定一个点)。有关通过PySC2提供的操作的更详细规范和描述，请参阅环境文档。

在《星际争霸》中，并不是所有的动作在每个游戏状态下都可用。例如，移动命令只在选择单元时可用。人类玩家可以在屏幕上的“命令卡”中看到哪些动作是可用的。类似地，我们通过每个步骤向代理提供的观察结果提供了可用操作的列表。采取不可用的行动被认为是错误的，因此代理人应该过滤他们的行动选择，以便只采取合法的行动。

人类通常每分钟做30到300个动作(APM)，随着玩家技能的提高，这个数字大致会增加，职业玩家的APM通常会超过500个。在我们所有的RL实验中，我们每8个游戏帧进行表演，大约相当于180个APM，这对于中级玩家来说是一个合理的选择。

我们相信这些早期的设计选择使我们的环境成为开发复杂的RL代理的一个很有前途的测试平台。具体来说，基于神经网络的agent具有固定大小的特征层输入空间和类人动作空间。这与最近的其他作品形成了鲜明的对比，在这些作品中，游戏是在单位单元的基础上被访问的，动作是单独指定给每个单元的。虽然这两种接口风格都有优点，但是PySC2提供了以下优点:

- 从人类的回放中学习变得更加简单。
- 我们不需要每分钟不现实的/超人类的行动来给每个单元单独发出指令。
- 这款游戏的设计初衷就是通过这个UI来玩，而在战略高层决策、管理你的经济和控制军队之间的平衡使得这款游戏更加有趣。

迷你游戏介绍

为了独立地研究游戏的元素，并为完整的游戏提供更细致的步骤，我们构建了几个迷你游戏。这些场景都集中在小地图上，这些小地图的构建目的是测试具有清晰奖励结构的动作和/或游戏机制的子集。不像整个游戏的奖励只是赢/输/平，迷你游戏的奖励结构可以奖励特定的行为。

我们鼓励社区使用强大的星际争霸地图编辑器来构建修改或新的迷你游戏。这不仅仅是设计一个更小范围的挑战领域。它允许与其他研究人员共享相同的设置和评估，并获得直接可比的评估分数。受限的操作集、自定义奖励函数和/或时间限制直接定义在生成的SC2Map文件中，易于共享。因此，我们鼓励用户使用这种定义新任务的方法，而不是在代理端定制。

我们发布的7款迷你游戏如下：

- 移动到瞭望塔:代理有一个海军陆战队员，每次到达一个瞭望塔得到+1。这个映射是一个带有简单贪婪算法的单元测试。
- 收集矿物碎片:特工从两个海军陆战队员开始，必须选择并移动他们去收集散布在地图上的矿物碎片。它移动单位的效率越高，得分就越高。
- 找到并击败小狗:特工从3名陆战队员开始，必须探索地图找到并击败小狗。这需要移动相机和高效的探索。
- 击败蟑螂:特工从9名陆战队员开始，必须打败4名蟑螂。每次击败所有的蟑螂，它就会得到5名海军陆战队增援和4名新蟑螂的重生。奖励+ 10 /罗奇和

–1 /海洋死亡。它能维持的海军陆战队越多，就能打败越多的蟑螂。
- 击败小狗和恶霸:和击败恶霸一样，除了对手有蛇眼和恶霸，当被杀死时每个奖励+5。这需要不同的策略，因为敌人的能力不同。
- 收集矿物和天然气:代理人在有限的基础上开始，在有限的时间内收集的总资源会得到奖励。一个成功的代理必须建立更多的工人，并扩大其资源收集率。
- 建立海军陆战队:特工从一个有限的基地开始，以建立海军陆战队为奖励。它必须建立工人，收集资源，建立补给库，建立兵营，然后训练海军陆战队。操作空间被限制为完成此目标所需的最小操作集。

每个迷你游戏都是有时间限制的，在mini_games.md的官方文档中有详细描述。

原始 API

星际争霸2有一个原始API，类似于Broodwar API (BWAPI)。在本例中，观察结果是地图上所有可见单元的列表，以及属性(单元类型、所有者、坐标、健康状态等)，但没有任何可视组件。战争迷雾仍然存在，但是没有摄像头，所以你可以同时看到所有可见的单位。这是一种更简单、更精确的表征，但它与人类对游戏的感知并不相符。为了与人类进行比较，这被认为是“欺骗”，因为它提供了重要的附加信息。

使用原始API，操作控制单元或者单元标识符单独的单元组。在发布行动之前，不需要选择个人或单位组。这允许比人机界面更精确的操作，从而产生了通过此API进行超人类行为的可能性。

虽然我们没有使用原始API进行机器学习研究，但是它作为发布的一部分被包含，以支持社区可能会发现有用的其他用例(例如脚本化的代理和可视化)。

性能

我们可以比实时运行环境更快。观察结果的呈现速度取决于几个因素:地图的复杂性、屏幕分辨率、每个动作的非呈现帧数和线程数。

对于复杂地图(如全梯形地图)，计算主要以仿真速度为主。减少动作的次数，允许更少的渲染帧，会减少计算量，但是收益递减很快就会开始，这意味着每次动作超过8步的收益很少。只要花很少的时间就可以做出更高的分辨率。在并行线程中运行更多的实例可以很好地扩展。

对于更简单的地图(例如收集矿物碎片)，世界模拟非常快，因此呈现观察结果占

主导地位。在这种情况下，增加每个动作的帧数和减少分辨率会有很大的影响。这时，瓶颈就变成了Python解释器，用一个解释器就可以抵消大约4个线程的性能提升。

分辨率为64×64的速度和代理8帧/行动,梯子的单线程的速度从200 - 700不等的游戏地图每墙上时钟的第二个步骤,这是比实时快一个数量级以上。确切的速度取决于多个因素，包括:游戏的阶段，游戏中的单位数量，以及它运行的计算机。在CollectMineralShards上同样的设置允许在壁钟每秒运行1600-2000步。

强化学习：基线代理

> 这个部分提供了一些基准测试结果，这些结果有助于校准地图的难度，并且已经建立的RL算法可以学习有用的策略，至少在小型游戏中是这样，但是仍然存在许多挑战。我们另外提供了两名人类玩家的得分:一名DeepMind游戏测试员（新手级别）和一名星际争霸特级（专业级别）。

学习算法

我们的强化学习代理的核心在于一个以θ为参数的深度神经网络，它定义了一个政策π_θ。在时间戳t时刻，代理接收观测S_t，选择一个行动的概率$\pi_\theta(a_t|s_t)$，然后收到奖励r_t环境，γ为奖励折扣系数。代理的目标是最大限度地增加返回$G_t = \sum_{k=0}^{\infty} = \gamma^k r_{t+k+1}$，。尽管为了不丧失通用型，我们也应该考虑到之前的状态，但是在这里我们假设策略仅以观测S_t为条件。

该策略的参数是通过使用 Mnih 等人所定义的 Asynchronous Advantage Actor Critic（A3C）来学习的，该方法在不同的环境中产生了最先进的结果。这是一种政策梯度方法，在$\mathbb{E}[G_t]$上进行近似梯度上升。

A3C梯度定义如下:

$$\underbrace{(G_t - v_\theta(s_t))\nabla_\theta log\pi_\theta(a_t|s_t)}_{PolicyGradient} + \underbrace{\beta(G_t - v_\theta(s_t))\nabla_\theta v_\theta(s_t)}_{ValueEstimationGradient} + \underbrace{\eta \sum_{k=1}^{a} \pi_\theta(a|s) log\pi_\theta(a|s)}_{EntropyRegularization}$$

符号意义：$v_\theta(s)$是估价函数对有同一个神经网络返回的返回值$\mathbb{E}[G_t|s_t = s]$的估计；我们利用的是 n 步之后的返回值，即$G_t = \sum_{k=0}^{\infty} = \gamma^k r_{t+k+1} + \gamma^n v_\theta(s_{t+n})$；A3C的最后一项是针对于大熵制定的，这一项鼓励了探索（exploration），参数β和η用来权衡各种损失成分的重要性。

策略表示

API将操作把 a 动作当成是一个包含各种动作标识符和参数的嵌入列表。因为所有参数包括屏幕上像素坐标和小地图上是离散的，对策略$\pi_\theta(a|s)$朴素的参数化方法需要数以百万计的值以确定 a 上的联合分布，即使是较低的空间分辨率。相反，我们建议用自回归的方式来表示策略，使用链式法则:

$$\pi_\theta(a|s) = \prod_{l=0}^{L} \pi(a^l|a^{<l}, s)$$

这种表示方法更加简单，因为它将选择一个完整的动作a的问题转换为每个参数a^l的一系列决策问题。根据函数标识符a0的不同，所需的参数L的数量是不同的。有些操作(例如，no-op操作)不需要任何参数，而其他操作(例如，move screen(x, y))需要。

可以使用任意的参数排列来定义应用链式法则的顺序。然而，在哪里点击屏幕的决定自然取决于游戏中点击的目的，即调用的函数的标识符。在我们的大多数实验中，我们发现独立建模子动作已经足够了，然而，我们还探索了自回归策略，其中顺序应该被固定。所以，我们首先要决定函数标识符，然后确定所有的范畴参数和像素坐标（如果有需要）。

为了确保我们的代理永远不会选择不可用的操作，我们屏蔽了a^0的函数标识符选择，以便只对适当的子集进行抽样，模拟玩家在UI上随机单击按钮的方式。我们通过掩盖动作和重命名a^0上的概率分布来实现。

代理架构

SC2LE的几种架构如下。

输入预处理。

所有的基线代理都对输入特性层进行相同的预处理。我们所有功能层包含分类值嵌入到一个连续空间相当于使用一个炎热的编码通道尺寸1×1卷积。我们也用对数变换重新调整数值特征，因为它们中的一些，如击中点或矿物，可能会达到相当高的数值。

Atari-net代理。

第一个基线是对Atari基准和DeepMind实验室环境成功使用的架构的轻微调整。它处理屏幕和小地图特征层，其卷积网络有两个层，两个图层，分别有16、32个

滤镜，大小分别为8、4和步数分别为4、2。利用带tanh非线性的线性层对非空间特征向量进行处理。结果被连接并通过一个ReLU激活的线性层发送。最后，结果向量被用作线性层的输入，这些线性层分别作为输出操作函数的策略。对于空间操作(坐标)，我们独立地建模策略来选择（离散的）x和y坐标。

完全卷积代理。

完全卷积网络代理通过一系列保留分辨率的卷积层直接预测空间操作。我们提出的网络没有跨步，在每一层都使用了padding，从而保持了输入空间信息的分辨率。为了简单起见，我们假设屏幕和小地图输入具有相同的分辨率。我们通过屏幕和小地图观察通过单独的双层卷积网络与16日32过滤器的大小5×5,分别为3×3。状态表示是由屏幕和小地图网络输出以及广播向量统计沿着通道维的串联形成的。为了计算分类（非空间）操作的基线和策略，状态表示首先通过包含256个单元和ReLU激活的完全连接层，然后是完全连接的线性层。最后,政策空间卷积操作是获得使用1×1的状态表示一个输出通道。

完全卷积LSTM代理。

以上两个基线都是前馈架构，因此没有内存。虽然这对于一些任务来说已经足够了，但是我们不能期望它对于星际争霸的复杂性足够了。这里我们介绍了一个基于卷积LSTM的基线体系结构。我们遵循上面描述的完全卷积代理的管道，在小地图和屏幕特征通道与非空间特征连接之后，简单地添加一个卷积LSTM模块。

随机代理。

我们使用两个随机基线。随即策略是一个代理，它在所有有效的操作中都是随机选择的，这突出了在非常大的操作空间中偶遇成功事件的困难。随机搜索基准通过采取许多独立的、随机初始化的策略网络（采用较低的softmax温度，可诱导接近确定性的行为）来工作，对每一集进行20次评估，并保持平均得分最高的那一集。这是互补的，因为它在政策空间而不是行动空间中取样。

结果

> 在A3C中，我们在网络的K = 40前步或接收到终端信号后，切割轨迹并进行反向传播。优化过程使用共享RMSProp运行64个异步线程。对于每种方法，我们进行了100次实验，每次都使用随机采样的超参数。学习速率是采样来自表单(10–5、10–3)区间。学习率线性退火从抽样值到一半的初始速度为所有代理人。我们使用一个独立的熵罚款10–3的操作功能和每个action函数参数。我们每8个游戏步骤就有一个固定的速率，这相当于每秒3个动作或

> 180个APM。

全部游戏

对于完整游戏的实验，我们选择了用于排名网络游戏和专业比赛的深海礁石学习环境梯子地图。在人族对战人族的比赛中，特工与最简单的内置人工智能对抗。最高游戏时长设定为30分钟，之后宣布平局，剧集结束。

实验结果如图5所示。不出所料，没有一个训练有素的三元奖励的代理能够为整个游戏开发一个可行的策略。最成功的一个基于完全卷积架构的没有内存的代理，通过使用人族的能力来提升并将建筑移出攻击范围，从而避免了持续的损失。

接受过暴雪评分训练的经纪人会采取一些琐碎的策略，只会避免分散工作人员对采矿的注意力，从而保持分数的稳定提升。因此，大多数代理的得分集中到简单地保留最初的挖掘过程，而不需要构建更多的单元或结构（在下面的迷你游戏中也观察到了这种行为）。

迷你游戏

我们在每个迷你游戏上都训练了我们的基线代理。总的来说，完全卷积代理在非人类基线中的表现最好。有点令人惊讶的是，Atari-net代理在涉及到战斗的小型游戏中似乎是一个相当强大的竞争者——即找到并打败虫族、打败蟑螂、收集矿物和瓦斯的游戏中。只有最好的卷积剂学会通过生产更多的工作单元并将其分配给开采来增加初始资源收益。

除了最简单的移动到瞭望塔外，所有的代理在与特级大师玩家进行比较时都表现不佳，这只需要良好的机制和反应——人工代理应该擅长的是什么。然而，在某些游戏中，比如打败虫族，我们的游戏代理与DeepMind游戏测试员相比表现不错。

利用回放进行监督式学习

> 游戏回放是专业玩家和业余玩家都使用的重要资源，他们可以学习新的策略，发现游戏中的关键错误，或者只是把观看别人的游戏作为一种娱乐方式。重放在星际争霸中尤其重要，因为它隐藏了很多信息：“战争之雾”会隐藏所有敌人的单位，除非他们在你的视野之内。因此，在职业球员中，回顾和分析他们所玩的每一场比赛是标准做法，即使他们赢了。使用监督数据被用于星际争霸2中，虽然它不是用来训练基本动作的策略，而是用来发现构

> 建命令。星际争霸2提供了收集和学习大量不断增长的人类回放的机会。同时，更多的游戏会定期添加到这个集合中，因为一个相对稳定的玩家池会玩新的游戏。
>
> 从回放学习应该有助于引导或补充强化学习。在国际标准化组织中，它还可以作为序列建模或内存体系结构处理长期相关性的基准。事实上，要理解一个游戏的展开过程，一个人必须有效地跨多个时间步骤整合信息。此外，由于部分可观测性，重放还可用于研究不确定性模型。最后，比较结果/动作预测的性能可能有助于指导神经体系结构的设计，使其在该领域具有适当的归纳偏差。
>
> 在本节的其余部分中，我们使用第4节中描述的体系结构提供基线，但是使用一组800K游戏来学习值函数（即，从游戏观察中预测游戏的赢家），以及一个策略（即，从游戏观察中预测所采取的行动）。游戏包含了星际争霸2中所有可能的对手戏。，我们不限制代理人玩单一种族。
>
> 我们总结了一些最有趣的数据：通过比赛评分(MMR)来衡量玩家的技能水平——休闲玩家、高端业余玩家、专业人士；平均每分钟行动数(APM)是153，平均MMR是3789；回放没有经过过滤，而是使用了所有在暴雪战网上进行的“排名”联赛比赛84场。只有不到1%回放是来自顶级玩家的大师级重放；我们还展示了根据人类玩家的使用频率排序的动作分布——在43%的时间里，最频繁的动作是移动相机，动作分布是一个重尾分布，有一些常用的动作（移动相机，选择矩形，攻击屏幕）和大量不常用的动作（建造一个工程舱）。
>
> 我们训练双头网络来预测游戏结果（1代表赢，0代表输或平局），以及玩家在每一步所采取的行动。共享网络的主体使得有必要平衡这两个损失函数的权重，但它也允许价值和政策预测相互告知。在监督训练中，我们并没有将平局作为一个单独的游戏结果类，因为数据集中的平局数量非常少。

值预测

预测游戏的结果是一项具有挑战性的任务。即使是职业星际争霸2的评论员也经常无法预测赢家，尽管他们可以完全进入游戏状态。能够准确预测游戏结果的价值函数是可取的，因为它们可以用来减轻从稀疏奖励中学习的挑战。从给定的状态出发，一个经过良好训练的价值函数可以表明，在看到游戏结果之前很久，哪些邻国是值得进入的。

我们的监督学习设置从第4节中描述的简单的基线体系结构开始:Atari-net和

FullyConv。网络没有考虑到以前的观察，即。他们从单个框架中预测结果(这显然不是最优)。此外，观察结果不包括任何特权信息：一个代理必须根据它在任何给定的时间步骤（例如，启用战争迷雾）所能看到的内容来进行价值预测。因此，如果对手成功地秘密生产了许多单位，这些单位对特工建立的军队非常有效，它可能会错误地认为自己的地位比实际更强大。

第4节中提出的网络独立地产生操作标识符及其参数。然而，预测屏幕上某一点的准确性可以通过调整基地行动来提高，例如，建立一个额外的基地而不是移动军队。因此，除了Atari-net和FullyConv体系结构之外，我们还有arFullyConv，它使用4.2节中引入的自回归策略引入，即使用函数标识符a0和先前采样的参数a<l来在当前的参数al上建模策略。

在《星际争霸2》中，所有可能的比赛中，网络都经过了200k梯度下降的训练。我们用小批量的64个观察数据进行训练，这些观察数据是随机从所有的回放中得到的。与RL的设置相一致的是，观察数据的采样采用8阶乘数法。资源-全局屏幕和小地图是64×64。每个观察包括屏幕和小地图空间特征层以及玩家数据，如食物帽和人类玩家在屏幕上看到的收集的矿物质的数量。我们使用了90%的回放作为训练集，以及从剩下的10%的回放中提取的0.5M帧的固定测试集。随着训练的进行，代理性能将根据测试集进行持续评估。

图8显示了训练步长的平均准确率以及训练模型的准确率作为游戏时间的函数。一个随机的基线会在大约50%的情况下纠正，因为比赛在所有的比赛组合中都是平衡的，而且平手非常罕见。随着训练的进行，FullyConv体系结构的准确率达到了64%。而且，随着游戏的发展，价值预测变得更加准确，这反映了之前在星际争霸I上的工作成果。

策略预测

经过训练以预测值的同一个网络有一个单独的输出，用于预测用户发出的操作。我们有时将网络的这一部分称为策略，因为它可以很容易地部署到游戏中。

人们可以采用许多方法来训练网络，以模仿人类的行为。在这里，我们使用一个简单的方法，直接连接到第4节中的RL工作。

当训练我们的策略时，我们在8帧的固定步长乘子上采样观察。我们将每8个框架内发布的第一个行动作为政策的学习目标。如果在此期间没有采取任何行动，我们将目标定为“no-op”，即。，一种没有效果的特殊行为。

当人类在玩星际争霸2时，在任何给定的时间里，只有一部分可能的动作是可用的。例如，“建造海军陆战队”只在当前选择兵营的情况下启用。网络不应该学习

如何避免非法行为，因为这些信息是现成的。因此，在训练中，我们过滤掉了人类玩家无法使用的动作。为此，我们对过去8个帧的所有可用操作进行联合，并应用掩码将所有不可用操作的概率设置为接近零。
注意，正如前面提到的，我们训练策略来玩所有可能的对弈。因此，原则上，代理人可以参加任何比赛。然而，为了与第4节所研究的强化学习代理保持一致，我们报告了单个人族与人族对弈的游戏内度量。

表2显示了不同体系结构在预测动作标识符、屏幕和最小ap参数方面的准确性方面的执行情况。正如预期的那样，FullyConv和arFullyConv体系结构在空间参数方面都表现得更好。同样，arFullyConv架构out——执行FullyConv，可能是因为它知道参数将用于哪个操作标识符。

当我们将经过监督学习训练的策略直接插入游戏中时，它可以产生更多的单位，并且作为观察回放数据的函数更好地发挥作用。它的表现也超过了在更简单的buildmarine迷你游戏第4部分中训练的所有特工，后者有一个受限的操作空间，尽管监管政策是在玩一个无限制的、完整的1v1游戏。这些结果表明，监督模仿学习可能是引导星际争霸II玩家的一个很有前途的方向。未来的工作应该着眼于通过直接培训和加强我们真正关心的目标的学习来改进初始化的模仿政策。,游戏的结果。

结论和未来的工作

本文将星际争霸II作为深入强化学习研究的新挑战。我们提供了免费的Python界面来玩游戏的详细信息，以及通过暴雪官方收集的排名游戏的人工回放数据。在这个初始版本中，我们描述了策略和价值网络的人类重放数据上的监督学习结果。我们还描述了7个迷你游戏和完整游戏中直接基线RL代理的结果。

我们认为迷你游戏主要是单元测试。也就是说，如果RL代理有机会在整个游戏中取得成功，那么它应该能够相对轻松地在这些游戏中实现人类水平的性能。构建额外的迷你游戏可能是有益的，但我们将完整的游戏——基于最终结果进行评估——作为最有趣的问题，并希望首先也是最重要的是鼓励研究，从而找到解决方案。

虽然在一些迷你游戏上的表现接近于专家的人类游戏，但我们发现，正如预期的那样，目前最先进的基线代理无法学会在整个游戏中战胜最简单的内置AI。我们提供的环境提出了一个挑战，对于现成的基线算法来说，它既是规范的、外部定义的，也是完全难以解决的。

这个版本简化了游戏的几个方面，因为它是由人类玩的：这些观察结果在提交给

代理之前经过预处理，操作空间已经被简化为更容易被RL代理使用，而不是使用键盘和鼠标点击设置由人类使用；它是在固定步中运行的，这样代理就可以在每个时间步长（而不是实时）中计算它们需要的时间。完整的游戏只允许对内置的人工智能玩。然而，我们认为真正的挑战是构建能够在自己的地盘上扮演最好的人类玩家的代理，即RGB像素观测和严格的时间限制。因此，未来的发行版可能会放松上面的简化，并允许自玩，使我们朝着训练代理的目标前进，人类认为这些代理是公平的对手。