上海大学未来技术学院 | 上海大学人工智能研究院
SCHOOL OF FUTURE TECHNOLOGY, SHANGHAI UNIVERSITY | INSTITUTE OF ARTIFICIAL INTELLIGENCE, SHANGHAI UNIVERSITY

人工智能导论

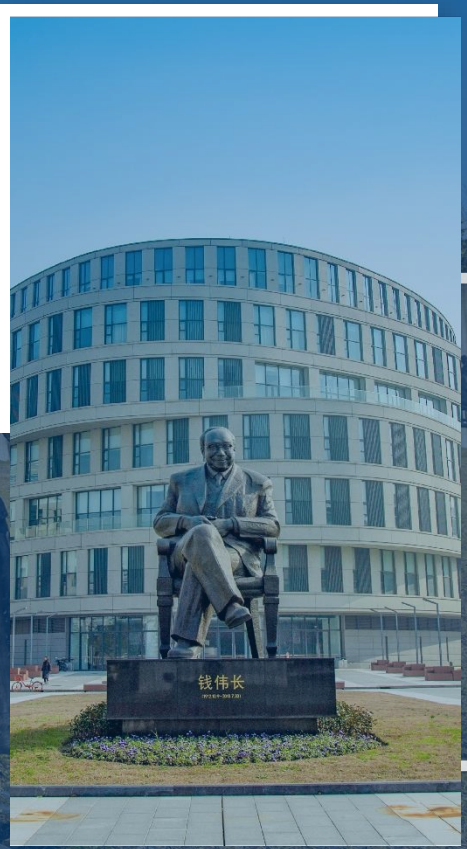
——第5课：人工智能在无人集群的应用

（博弈对抗）

叶林奇

未来技术学院（人工智能研究院）

2023冬季学期



提纲

一、博弈论

二、智能博弈

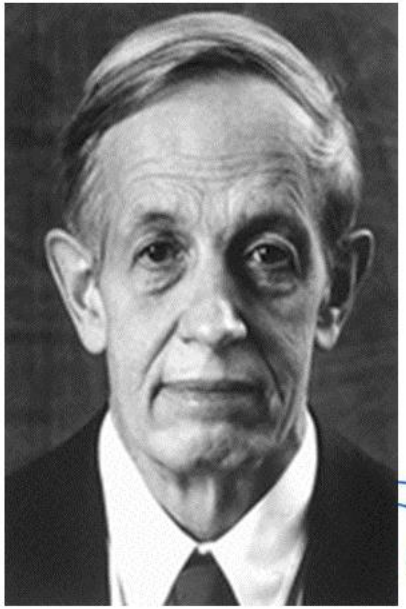
三、AlphaStar和DeepNash

四、踢足球、躲猫猫、追逃、空战

博弈论

诺贝尔奖：“博弈论”的全胜

1994年



约翰·纳什(JOHN NASH)，美国人(1928-2015)，由于他与另外两位数学家在非合作博弈的均衡分析理论方面做出了开创性的贡献，对博弈论和经济学产生了重大影响，而获得1994年诺贝尔经济奖。

纳什均衡

(完全信息静态博弈)

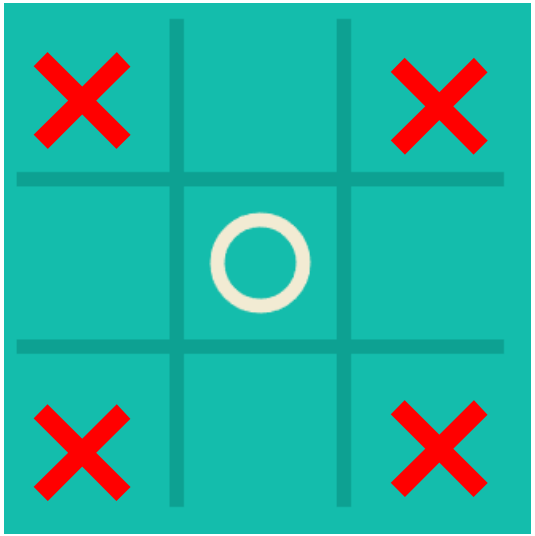
博弈论

- Game Theory : 游戏理论、赛局论、对策论、博弈论。
- Game : 游戏、赛局、博弈。

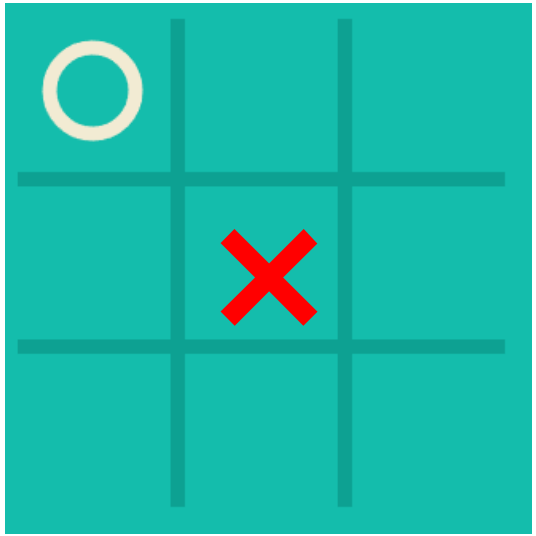
博弈论

1913年，**策梅洛定理**：对于一个两人的完全信息游戏，一定存在一个策略，要么先手一定获胜，要么后手一定获胜，要么双方一定平局。（数数游戏）

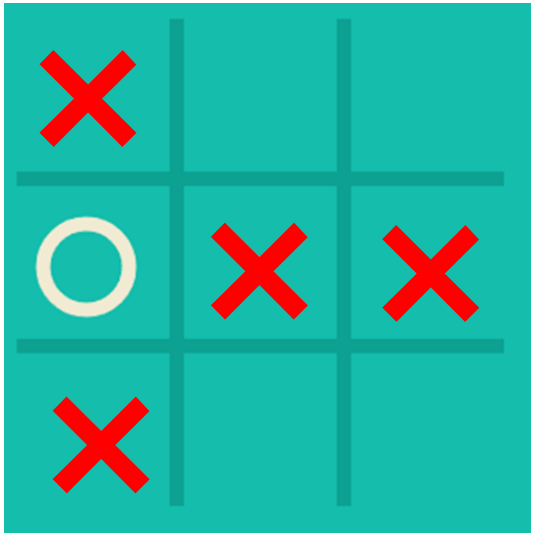
博弈论



下中必下角



下角必下中



下边占旁边

博弈论

- 纳什只有二十七页的博士论文《非合作性博弈》(Non-cooperative Games)，奠定了博弈论最重要的数学基础。

- 在这论文里，纳什大大发展了由冯·诺伊曼(John von Neumann)所创立的博弈论。

ANNALS OF MATHEMATICS
Vol. 54, No. 2, September, 1951

NON-COOPERATIVE GAMES

JOHN NASH

(Received October 11, 1950)

Introduction

Von Neumann and Morgenstern have developed a very fruitful theory of two-person zero-sum games in their book *Theory of Games and Economic Behavior*. This book also contains a theory of n-person games of a type which we would call cooperative. This theory is based on an analysis of the interrelationships of the various coalitions which can be formed by the players of the game.

Our theory, in contradistinction, is based on the *absence* of coalitions in that it is assumed that each participant acts independently, without collaboration or communication with any of the others.

The notion of an *equilibrium point* is the basic ingredient in our theory. This notion yields a generalization of the concept of the solution of a two-person zero-sum game. It turns out that the set of equilibrium points of a two-person zero-sum game is simply the set of all pairs of opposing "good strategies."

In the immediately following sections we shall define equilibrium points and prove that a finite non-cooperative game always has at least one equilibrium point. We shall also introduce the notions of solvability and strong solvability of a non-cooperative game and prove a theorem on the geometrical structure of the set of equilibrium points of a solvable game.

As an example of the application of our theory we include a solution of a simplified three person poker game.

Formal Definitions and Terminology

In this section we define the basic concepts of this paper and set up standard terminology and notation. Important definitions will be preceded by a subtitle indicating the concept defined. The non-cooperative idea will be implicit, rather than explicit, below.

Finite Game:

For us an *n-person game* will be a set of n *players*, or *positions*, each with an associated finite set of *pure strategies*; and corresponding to each player, i, a *payoff function*, p_i, which maps the set of all n-tuples of pure strategies into the real numbers. When we use the term *n-tuple* we shall always mean a set of n items, with each item associated with a different player.

Mixed Strategy, s_i:

A *mixed strategy* of player i will be a collection of non-negative numbers which have unit sum and are in one to one correspondence with his pure strategies. We write $s_i = \sum_\alpha c_{i\alpha}\pi_{i\alpha}$ with $c_{i\alpha} \geqq 0$ and $\sum_\alpha c_{i\alpha} = 1$ to represent such a mixed strategy, where the $\pi_{i\alpha}$'s are the pure strategies of player i. We regard the s_i's as points in a simplex whose vertices are the $\pi_{i\alpha}$'s. This simplex may be re-

博弈论

- 博弈论尝试为决策者之间的冲突与合作建立数学模型。
- 它研究每一个决策者将如何根据其他对手的策略，去作出最有利自己的策略。

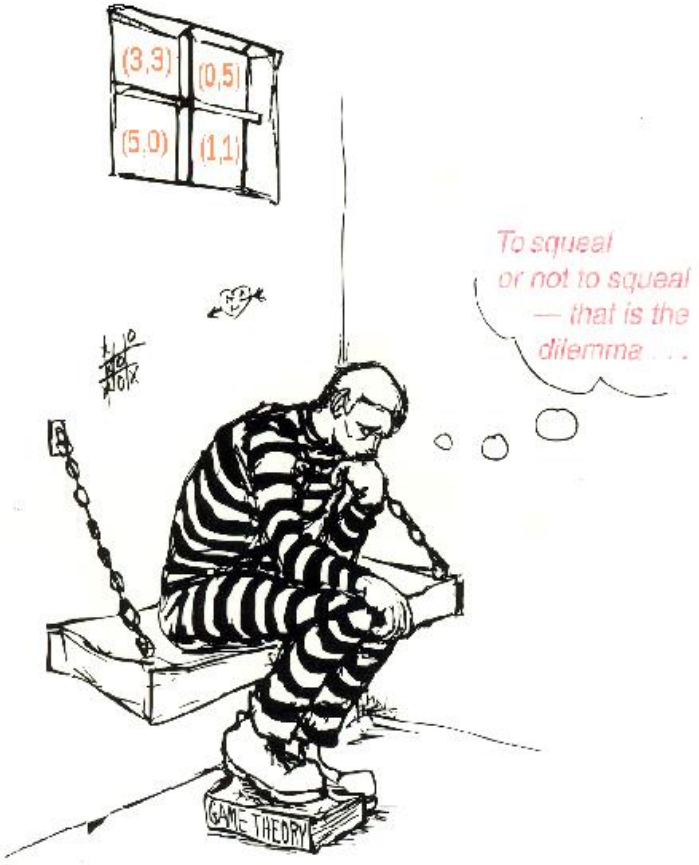
博弈论

1950年，**纳什定理**:

只要参与博弈的几方策略都是有限的，那么就一定存在一种平衡状态，大家都会采用这种平衡策略，而没有单方面改变策略的动力。这种平衡状态就叫做纳什均衡。

策梅洛定理其实是纳什定理的一个推论。

囚徒困境 (Prisoner's Dilemma)



- 甲与乙被警方以藏械罪名拘捕。警方怀疑他们正准备持械行劫。两人被单独囚禁和盘问。
- 如果二人都承认意图行劫，每人将被判入狱三年。
- 如果他们都不承认，则各判入狱一年。
- 如果一人否认而另一人承认，并且愿意作证，那否认者将被判入狱五年，而承认者则可获释放。

囚徒困境

		乙	
		认罪	不认罪
甲	认罪	-3, -3	0, -5
	不认罪	-5, 0	-1, -1

甲：(认，不认) > (不认，不认) > (认，认) > (不认，认)

乙：(不认，认) > (不认，不认) > (认，认) > (认，不认)

博弈的解 (Solution of a game)

- 对每一个博弈，我们都希望知道每个参与者将如何决策。
- 所有参与者的最后决策便构成博弈的解 (solution of a game) 。
- 试找出「囚徒困境」的解。

假设甲认罪，乙应该如何决策?

		乙	
		认罪	不认罪
甲	认罪	-3, -3	0, -5
	不认罪	5, 0	-1, -1

假设甲不认罪，那么乙应该怎样做?

		乙	
		认罪	不认罪
甲	认罪	~~-3, -3~~	~~0, -5~~
	不认罪	-5, 0	-1, -1

- 无论其他对手怎样选择，这个策略给某参与者带来的得益，都比任何其他策略为高。
- 乙和甲的上策都是认罪。
- 若乙与甲都是理性的，则他们都会选择认罪。

博弈的解(结果)

		乙	
		认罪	不认罪
甲	认罪	-3, -3	0, -5
	不认罪	-5, 0	-1, -1

博弈论

- 如果每个参与者都有一个上策，则他们都会选择其上策，我们因而得知博弈的结果。

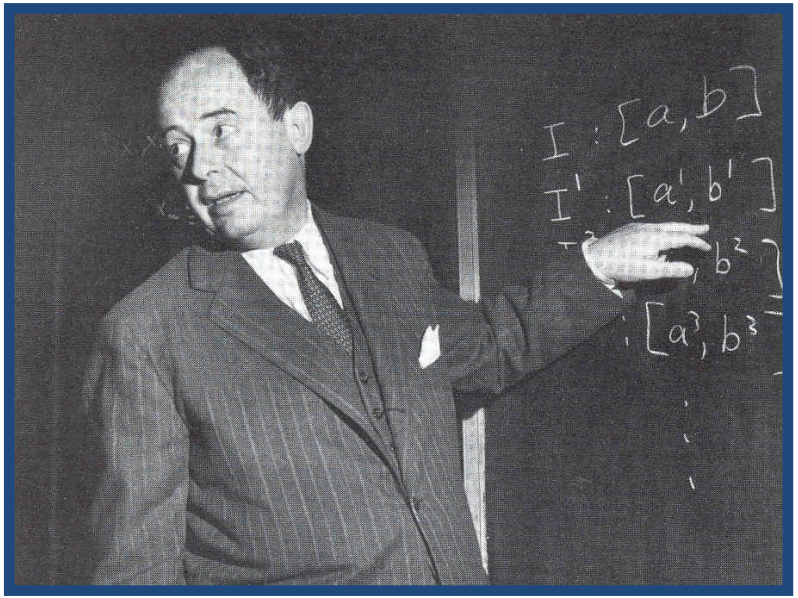
- 如果每个参与者都选择其上策，则这个策略组合称为「上策均衡」。

博弈论

- 处于「上策均衡」时，则每个参与者都不会改变自己的策略。
- 如果一个博弈存在「上策均衡」，那么每个参与者将依从这个均衡去作出选择，我们因此可以推断出他们的行为。

博弈论

- 冯·诺伊曼证明了对一类特殊的二人博弈，「零和博弈」(zero-sum game)，上策均衡必定存在。
- 可是对大部份的博弈，上策均衡都不存在。



纳什为何获得诺贝尔经济学奖?

- 在纳什的博士论文里，他引入了现称为混合纳什均衡 (mixed Nash equilibrium)的概念。它是一种比上策均衡更一般的解。

- 纳什证明，一般地，每个博弈都最少要有一个混合纳什均衡。

纳什的诺贝尔得奖定理

每个 n 人非合作博弈都至少有一个混合纳什均衡。

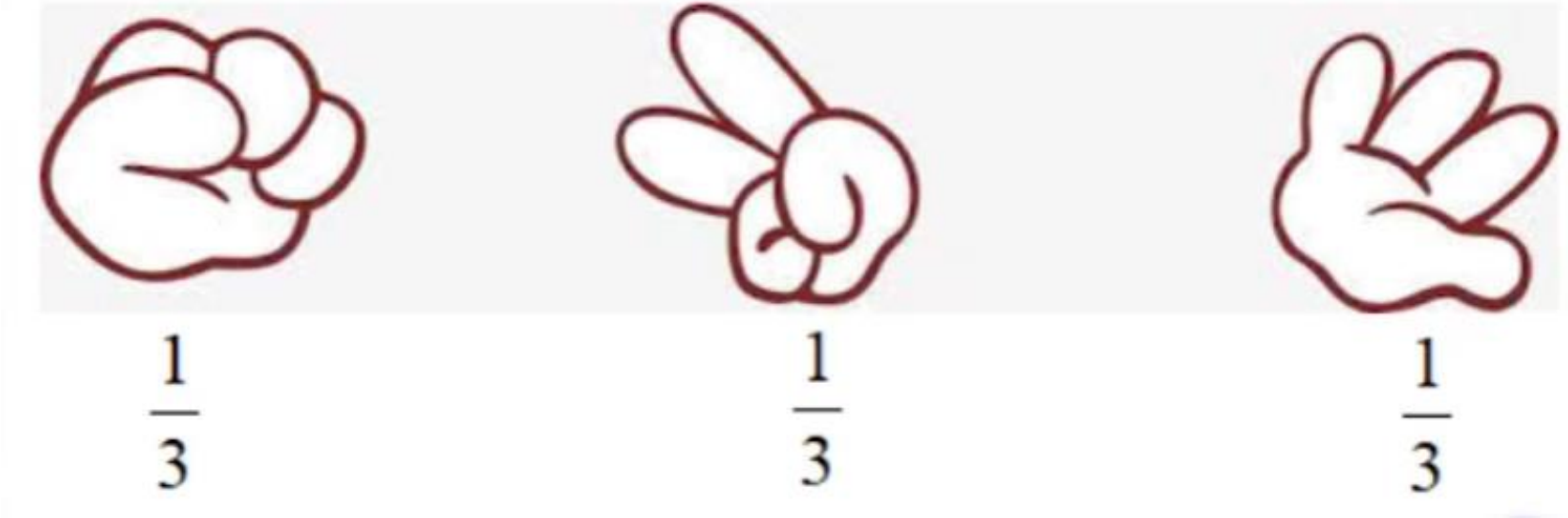
Every finite n-player non-cooperative game has a mixed Nash equilibrium.

冯 • 诺伊曼证明：

每一个二人「零和博弈」(zero-sum game)，都有一个「混合上策均衡」。

博弈论

石头剪刀布不存在纯策略纳什均衡——双方都固定不变的策略集合
石头剪刀布存在混合策略纳什均衡——以固定的频率在几种策略之间切换让自己的收益最大化。

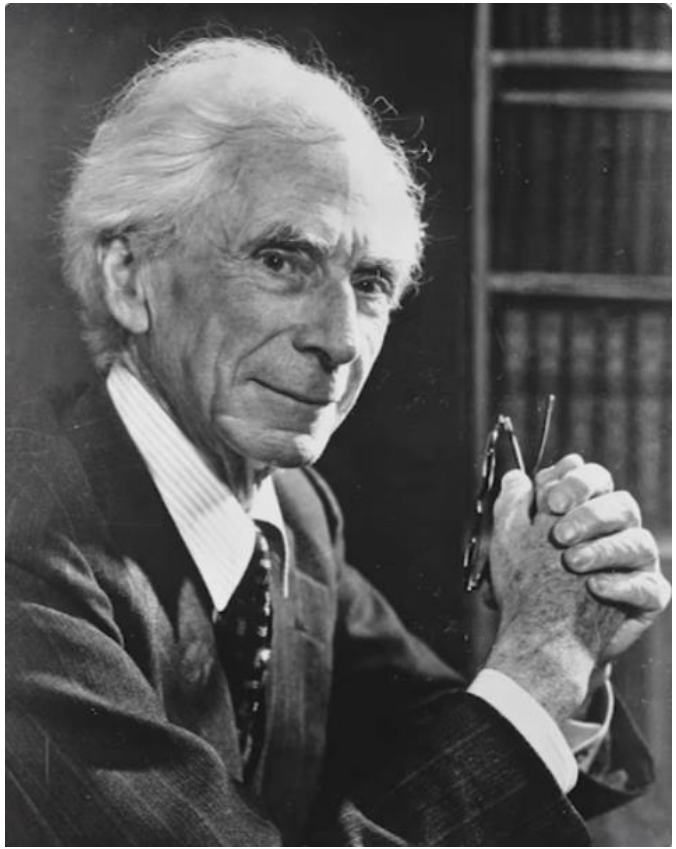


博弈论

- 一个策略组合称为「纳什均衡」，如果所有参与者都不会独自改变他们已选择的策略。
- 当处于「上策均衡」时，每一个参与者都不会再改变自己的策略。
- 当处于「纳什均衡」时，每一个参与者都不会独自改变自己的策略。
- 因此「上策均衡」必定是「纳什均衡」。

博弈论

胆小鬼博弈，1959，罗素：
两个人在一条车道上相对着开车，每个人都可以随时打方向盘驶出车道最先驶出车道的人就会被对方嘲笑为胆小鬼而一直在车道上狂飙的人就被称为英雄。



胆小鬼博弈

	B示弱	B死磕
A示弱	2，2	1，3
A死磕	3，1	-10，-10

	B 示弱	B 死磕
A 示弱	2，2	1，3
A 死磕	3，1	-10，-10

	B 示弱	B 死磕
A 示弱	2，2	1，3
A 死磕	3，1	-10，-10

博弈论

混合策略纳什均衡:纳什均衡点，包括纯策略纳什均衡和混合策略纳什均衡，绝大多数情况下都是奇数个。

胆小鬼博弈中至少还有一个混合策略纳什均衡。

	B 示弱	B 死磕
A 示弱	2，2	1，3
A 死磕	3，1	-10，-10

	B 示弱	B 死磕
A 示弱	2，2	1，3
A 死磕	3，1	-10，-10

胆小鬼博弈的混合策略纳什均衡

频率和收益	B示弱y	B死磕1-y
A示弱x	xy，（2，2）	x(1-y)，（1，3）
A死磕1-x	y(1-x)，（3，1）	(1-x)(1-y)，（-10，-10）

$$E(A)=(11-12y)x+13y-10$$

B调整y，让E(A)与x无关，所以y=11/12

$$E(B)=(11-12x)y+13x-10$$

A调整x，让E(B)与y无关，所以x=11/12

拍卖陷阱

- 拍卖20元纸币这20元是真币，但是没有任何收藏价值
- 1元起拍，每次加价至少1元
- 出价最高的朋友会获得这20元
- 出价最高和出价次高的朋友都需要按照你的出价付款

商业竞争

教育内卷

你来，我培训你

你不来，我培训你对手

——补习班广告

参考资料

李永乐老师：生活中的博弈论
【中科院科学公开课S02EP29】

吴端伟博士
香港大学数学系

《博弈高手——浅论约翰·纳殊的诺贝尔奖得奖理论》
https://hkumath.hku.hk/~ntw/Chinese_Public_lecture.ppt

提纲

一、博弈论

二、智能博弈

三、AlphaStar和DeepNash

四、踢足球、躲猫猫、追逃、空战

智能博弈

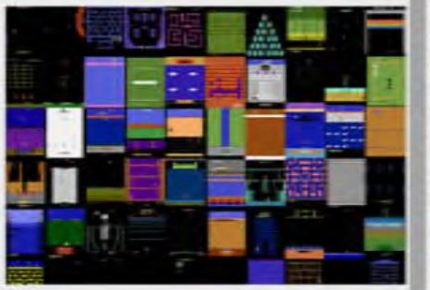
“深蓝”
国际象棋AI

特点：

- 战胜世界象棋冠军
- 基于枚举法，每秒检索1亿到2亿个棋局

1997年

Atari 视频游戏AI



特点：

- 在49个视频游戏中的得分均超过人类高级玩家
- 直接将游戏画面作为信息输入

2015年

智能博弈

Alpha AI
空战系统

特点：

- 在模拟器中击败人类飞行员
- 遗传模糊树
- 基于演进式规则的推理系统

AlphGo
围棋AI

特点：

- 打败世界围棋冠军
- 深度强化学习
- 蒙特卡洛搜索

先知
兵棋AI

特点：

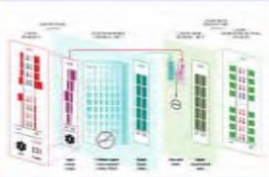
- 全国兵棋大赛中以7:1成绩击败人类前8强选手
- 知识规则、不确定推理

AlphZero
围棋AI

特点：

- 从零开始自学国际象棋、将棋和围棋，打败了世界顶尖程序
- 自博弈强化学习

DeepStack
德州扑克AI



特点：

- 击败职业玩家
- 两人无限注德州扑克
- 基于神经网络的后悔值估计

Libratus
德州扑克AI

特点：

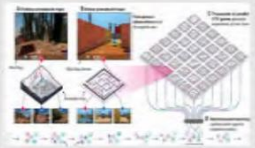
- 击败职业玩家
- 两人无限注德州扑克
- 子博弈嵌套求解

2016年

2017年

智能博弈

雷神之锤3
夺旗游戏AI



特点：

- 超越人类玩家水平
- 多智能体强化学习，种群训练

Muzero
通用棋类AI

特点：

- 游戏规则未知，超人性能与AlphaZero相匹敌。
- 基于模型的强化学习方法

Pluribus
多人德州扑克AI

特点：

- 顶尖六人无限注德州扑克AI
- MCCFR、权重衰减、动态剪枝

OpenAI Five
Dota2 AI

特点：

- 在Dota2游戏中打败业余队伍
- 强化学习，自博弈

捉迷藏AI

特点：

- 智能体们掌握了四种基本的游戏策略。
- 自博弈和近端策略优化

战颅
墨子兵棋AI

特点：

- 在第三届全国兵棋推演大赛人机挑中战胜参赛的11位人类选手。

2018年

2019年

智能博弈

Suphx
麻将 AI

特点：

- 第一个击败大部分人类顶尖玩家的麻将AI
- 监督学习、强化学习、在线策略优化

绝悟- WeKick
Google 足球 AI

特点：

- 使用人类玩家数据监督学习
- 端到端
- 标记宏观策略和微观操作

AI 星际指挥官
星际争霸 AI

特点：

- 以2:0 的成绩战胜全国冠军和总决赛冠军、最强人族选手

绝悟- SL
王者荣耀 AI

特点：

- 使用人类玩家数据监督学习
- 端到端
- 标记宏观策略和微观操作

AlphStar
星际争霸 AI

特点：

- 打败了职业星际玩家
- 多智能体强化学习方法、模仿学习、图卷积神经网络

绝悟
王者荣耀AI

特点：

- 支持5v5 对战，操控不同英雄达到人类顶尖水平
- 近端策略优化、控制解耦、注意力机制

2020年

智能博弈

DouZero
斗地主AI

特点：

- 达到专家水平
- 不需要人类知识
- 蒙特卡罗与强化学习结合、自博弈

GameBreaker
DARPA 项目

特点：

- 将人工智能应用到开放世界视频游戏中
- 探索对游戏最不稳定的新功能、战术和规则修改

AlphaDogFight
挑战赛

特点：

苍鹭系统公司的AI算法在虚拟空战中5:0压倒性优势击败了F16飞行教官

2021年

DeepNash

特点：

DeepMind用AI模拟足球比赛，实现1v1实物对战和3v3仿真对战

Football

特点：

DeepMind AI首次在西洋陆军棋达到人类专家水平，胜率84%

2022年

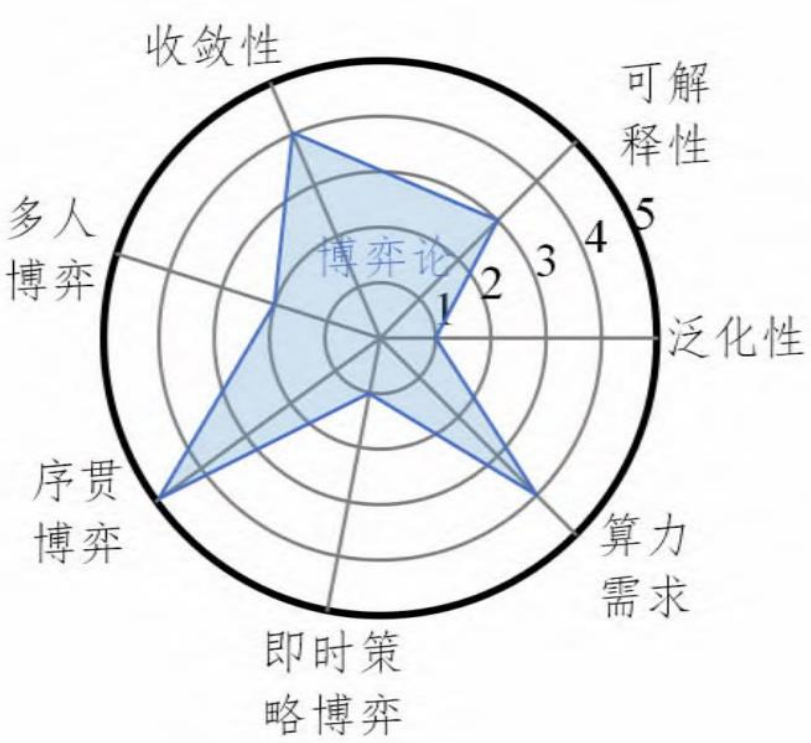
典型大规模对抗空间博弈场景

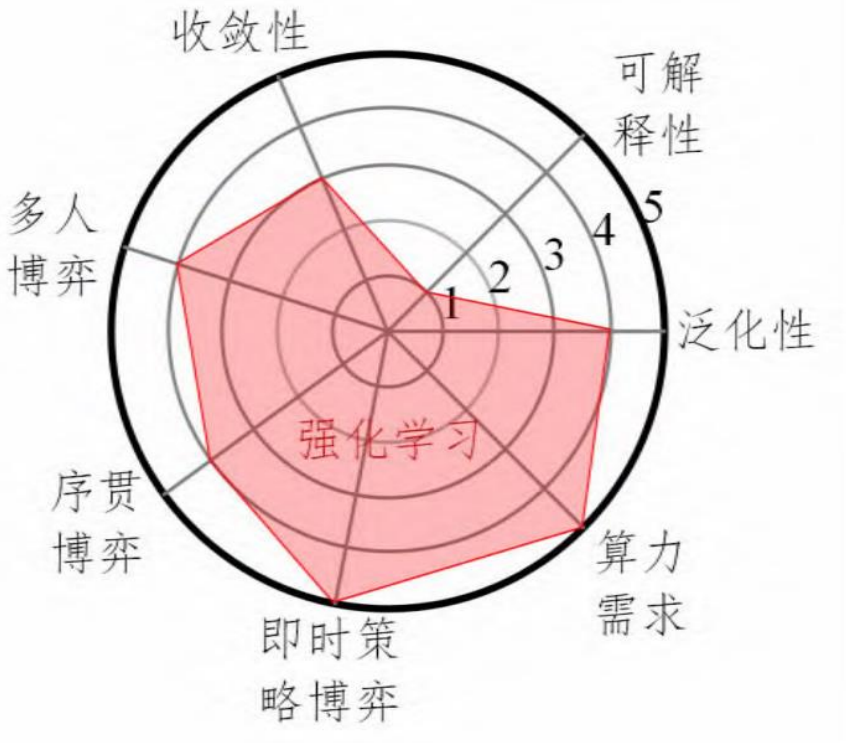
博弈场景	状态空间	备注
桥牌	10^{67}	—
斗地主	10^{83}	—
德州扑克	10^{160}	两人无限注
围棋	10^{170}	—
王者荣耀	10^{600}	1 对 1 场景
兵棋推演	10^{793}	城镇居民地想定[38]
星际争霸	10^{1685}	128×128 地图，仅考虑 400 个单元的位置

典型大规模对抗空间博弈场景

游戏/兵棋	状态空间	动作空间	决策数量	胜利条件	回报值设置	战争迷雾	观察信息	对手建模	想定设计
《Go》	中等	中等	中等	数子法/数目法	简单	无	简单	中等	固定
《星际争霸 II》	复杂	复杂	较多	单任务目标	中等	有	中等	中等	变化较小
《Dota 2》	复杂	复杂	较多	单任务目标	中等	有	中等	中等	固定
《CMANO》	非常复杂	非常复杂	巨大	多任务目标	复杂	有	复杂	复杂	变化较大
《智戎•未来指挥官》	非常复杂	非常复杂	巨大	多任务目标/积分	复杂	有	复杂	复杂	变化较大
《王者荣耀》	复杂	复杂	较多	单任务目标	中等	有	中等	中等	固定
《战争游戏：红龙》	非常复杂	非常复杂	巨大	多任务目标	复杂	有	复杂	复杂	变化较大
《MaCA》	中等	中等	中等	积分	简单	有	中等	中等	固定

博弈论与强化学习方法的特性对比





强化学习求解大规模智能博弈算力需求

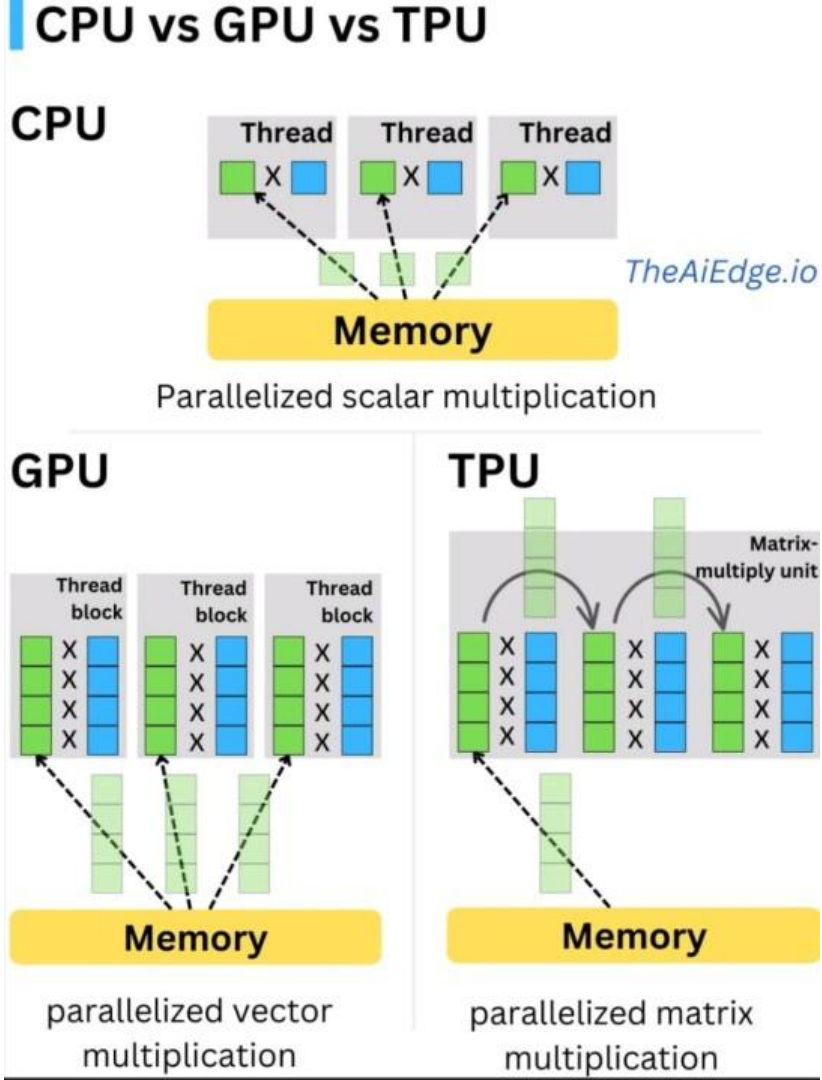
复杂博弈	硬件资源	训练时间/d	采样数据
AlphaStar[19]	16 TPUs	14	200 年游戏数据量
OpenAI Five (Dota2)[83]	128000 CPUs, 100 GPU	—	每天 180 年游戏数据量
Hide-and-Seek[26]	—	—	3.8 亿回合
Deepstack[24]	35000 CPUs, 320 GPUs	14	—
DeltaDou[91]	68 CPUs	60	—

Difference Between

CPU

GPU

TPU



参考资料

[1]袁唯淋,罗俊仁,陆丽娜,et al.智能博弈对抗方法:博弈论与强化学习综合视角对比分析[J].计算机科学, 2022, 49(8):14.DOI:10.11896/jsjkx.220200174.

[2]孙宇祥,彭益辉,李斌,et al.智能博弈综述:游戏AI对作战推演的启示[J].智能科学与技术学报, 2022(004-002).DOI:10.11959/j.issn.2096-6652.202209.

提纲

一、博弈论

二、智能博弈

三、AlphaStar和DeepNash

四、踢足球、躲猫猫、追逃、空战

AlphaStar和DeepNash

Deepmind

Article

Grandmaster level in StarCraft II using multi-agent reinforcement learning

2019，《Nature》

AlphaStar

https://doi.org/10.1038/s41586-019-1724-z
Received: 30 August 2019
Accepted: 10 October 2019
Published online: 30 October 2019

Oriol Vinyals[1,3]*, Igor Babuschkin[1,3], Wojciech M. Czarnecki[1,3], Michaël Mathieu[1,3], Andrew Dudzik[1,3], Junyoung Chung[1,3], David H. Choi[1,3], Richard Powell[1,3], Timo Ewalds[1,3], Petko Georgiev[1,3], Junhyuk Oh[1,3], Dan Horgan[1,3], Manuel Kroiss[1,3], Ivo Danihelka[1,3], Aja Huang[1,3], Laurent Sifre[1,3], Trevor Cai[1,3], John P. Agapiou[1,3], Max Jaderberg[1], Alexander S. Vezhnevets[1], Rémi Leblond[1], Tobias Pohlen[1], Valentin Dalibard[1], David Budden[1], Yury Sulsky[1], James Molloy[1], Tom L. Paine[1], Caglar Gulcehre[1], Ziyu Wang[1], Tobias Pfaff[1], Yuhuai Wu[1], Roman Ring[1], Dani Yogatama[1], Dario Wünsch[2], Katrina McKinney[1], Oliver Smith[1], Tom Schaul[1], Timothy Lillicrap[1], Koray Kavukcuoglu[1], Demis Hassabis[1], Chris Apps[1,3] & David Silver[1,3]*

2022，《Science》

DeepNash

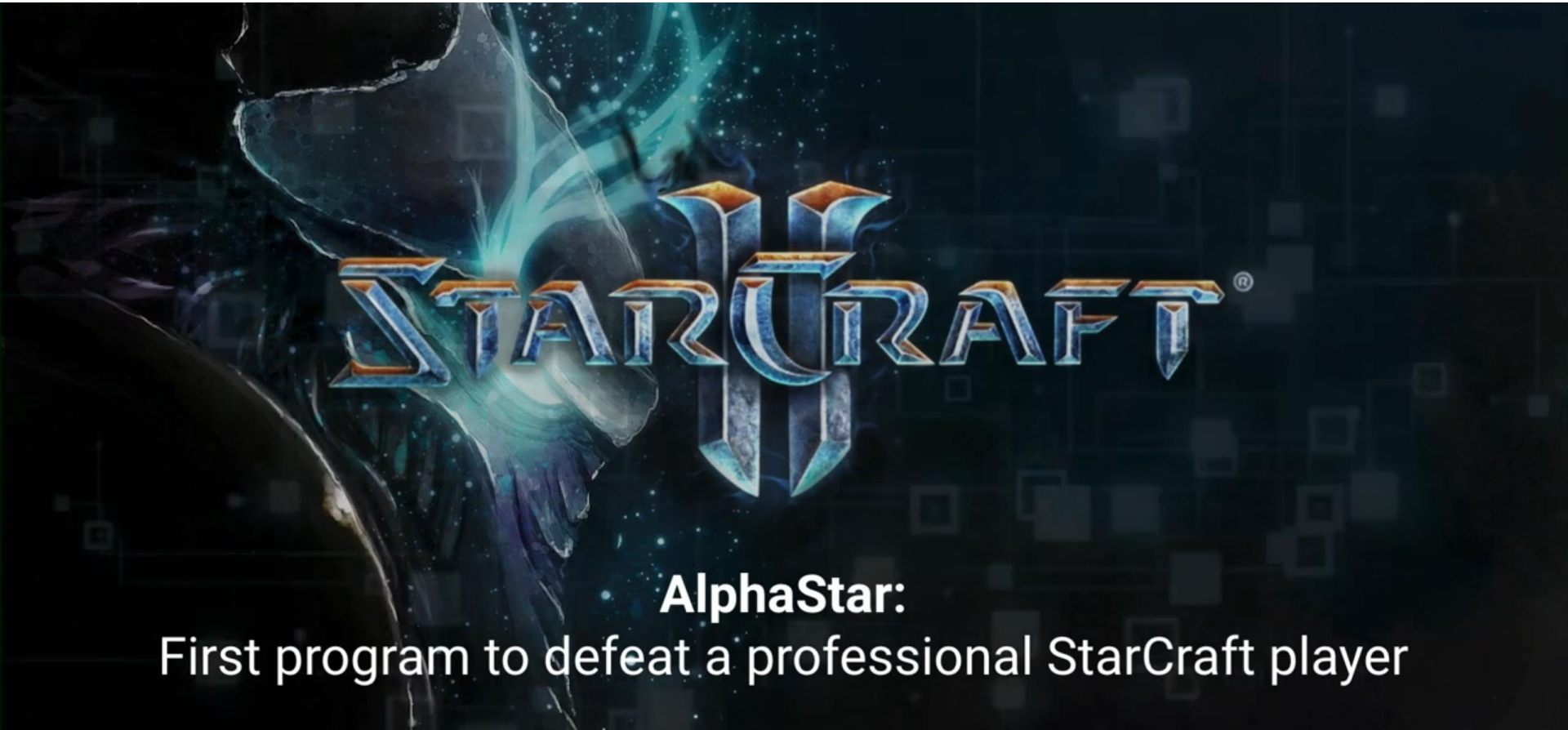
RESEARCH

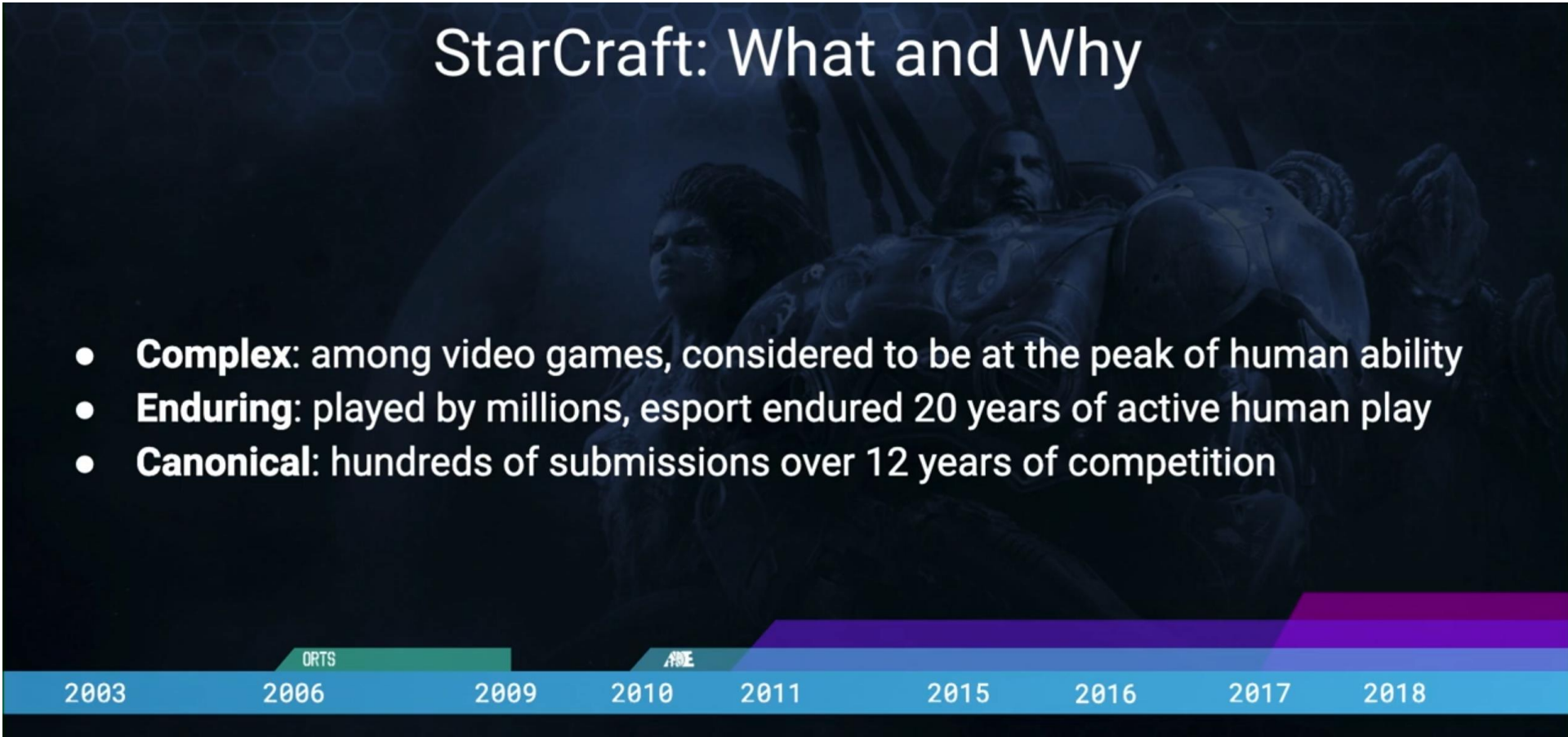
MACHINE LEARNING

Mastering the game of Stratego with model-free multiagent reinforcement learning

Julien Perolat*†, Bart De Vylder*†, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer‡, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean-Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Remi Munos, David Silver, Satinder Singh, Demis Hassabis, Karl Tuyls*†

AlphaStar和DeepNash



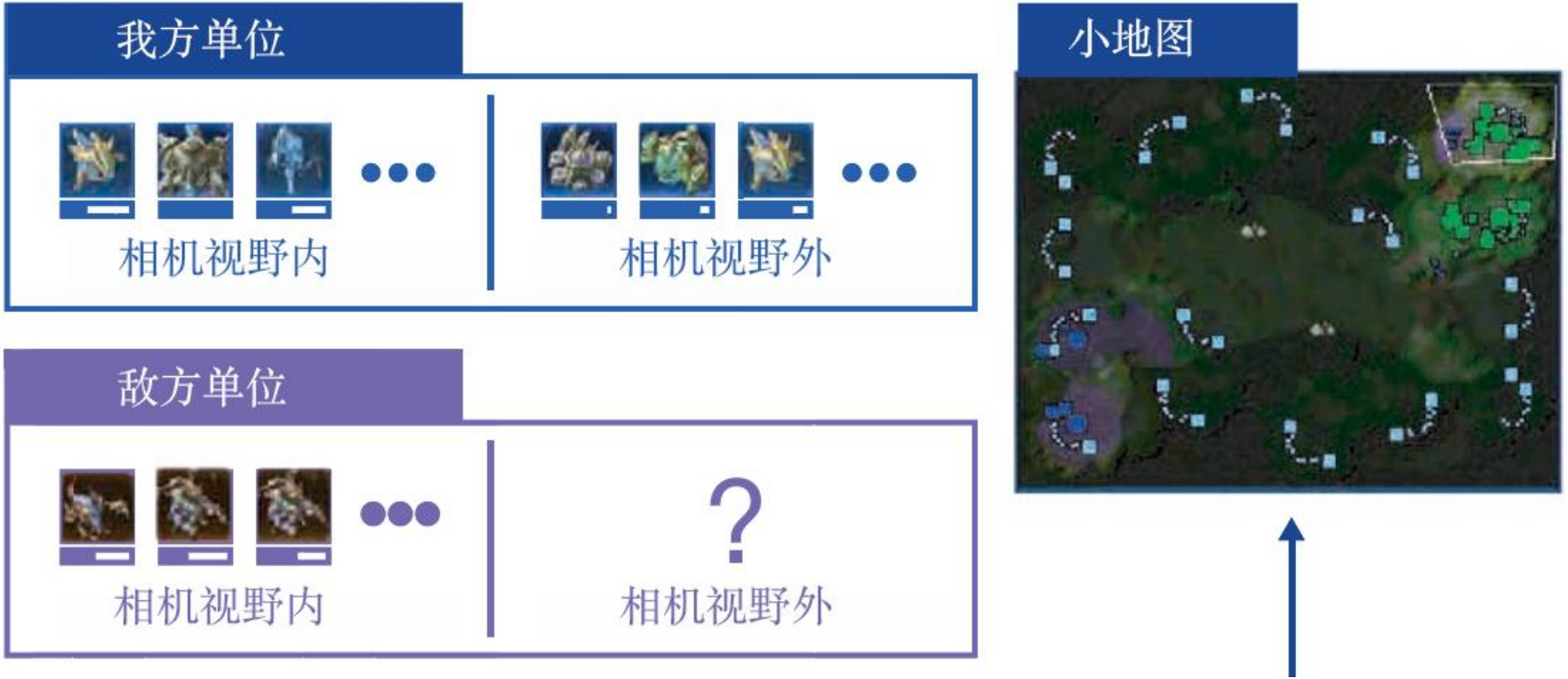

StarCraft: What and Why
Complex: among video games, considered to be at the peak of human ability
Enduring: played by millions, esport endured 20 years of active human play
Canonical: hundreds of submissions over 12 years of competition
ORTS
2003
2006
2009
2010
2011
2015
2016
2017
2018

The Challenge of StarCraft for AI

- Large action space: simultaneous control of hundreds of units
  - 10^{26} actions
- Long planning horizon
  - 15,000 steps until the game is won/lost
- Impoverished information
  - Fog-of-war: only see opponent units within range of own units
  - Active camera view: can only view/control units that are on-screen
- Game theoretical challenges
  - Cyclic strategies: discovered by pro players over 20 years
  - Compounded by hard exploration, human alignment

AlphaStar和DeepNash

AlphaStar将《星际争霸II》的环境状态分为4部分，分别为实体（entities）信息、地图（map）信息、玩家数据（player data）信息、游戏统计（game statistics）信息



AlphaStar和DeepNash

- 第一部分——————实体信息，例如当前时刻环境中有什么建筑、兵种等，并且我们将每一个实体的属性信息使用向量表示。例如对于一个建筑，其当前时刻的向量中包含此建筑的血量、等级、位置以及冷却时间等信息。所以对于当前帧的全部实体信息，环境会给神经网络 N个长度为 K的向量，分别表示此刻智能体能够看见的 N个实体的具体信息（向量信息）。

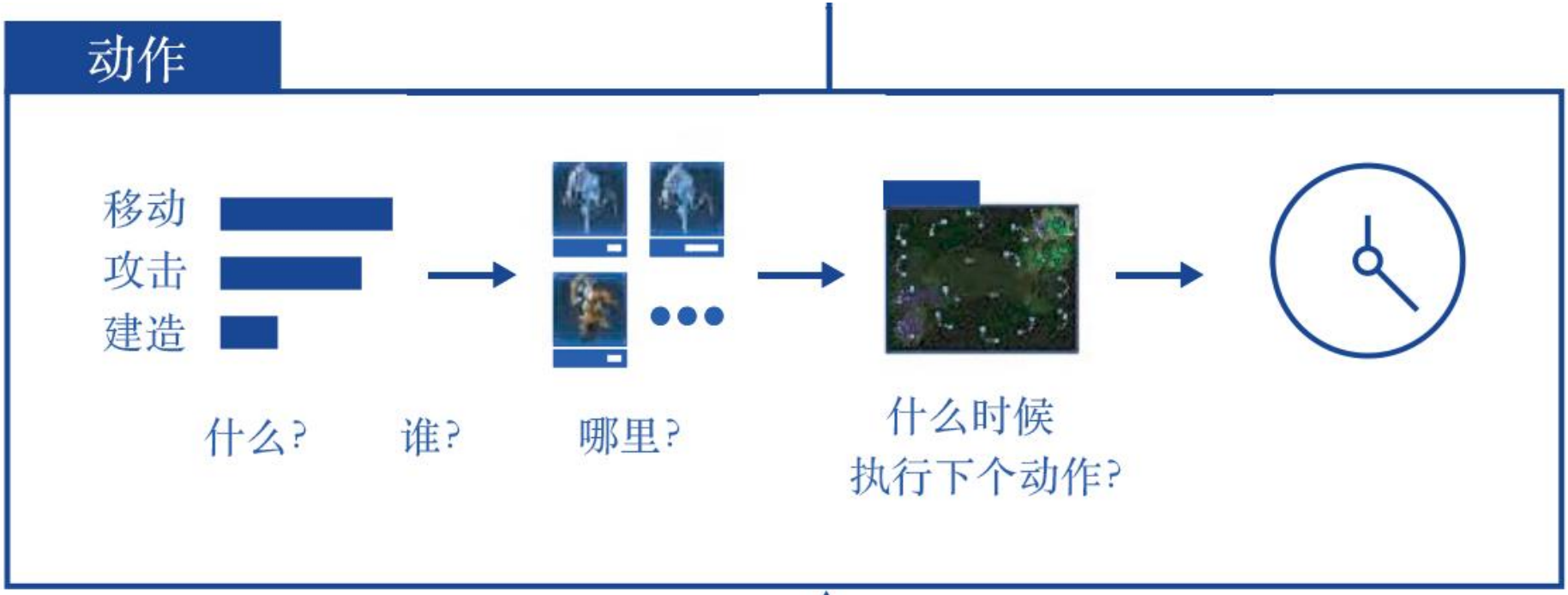
- 第二部分——————地图信息，这部分比较好理解，即将地图中的信息以矩阵的形式输入神经网络中，来表示当前状态全局地图的信息（向量信息或图像信息）。

- 第三部分——————玩家数据信息，也就是当前状态下，玩家的等级和种族等信息（标量信息）。

- 第四部分——————游戏统计信息，视野的位置（小窗口的位置，区别于第二部分的全局地图信息），还有当前游戏的开始时间等信息（标量信息）。

AlphaStar和DeepNash

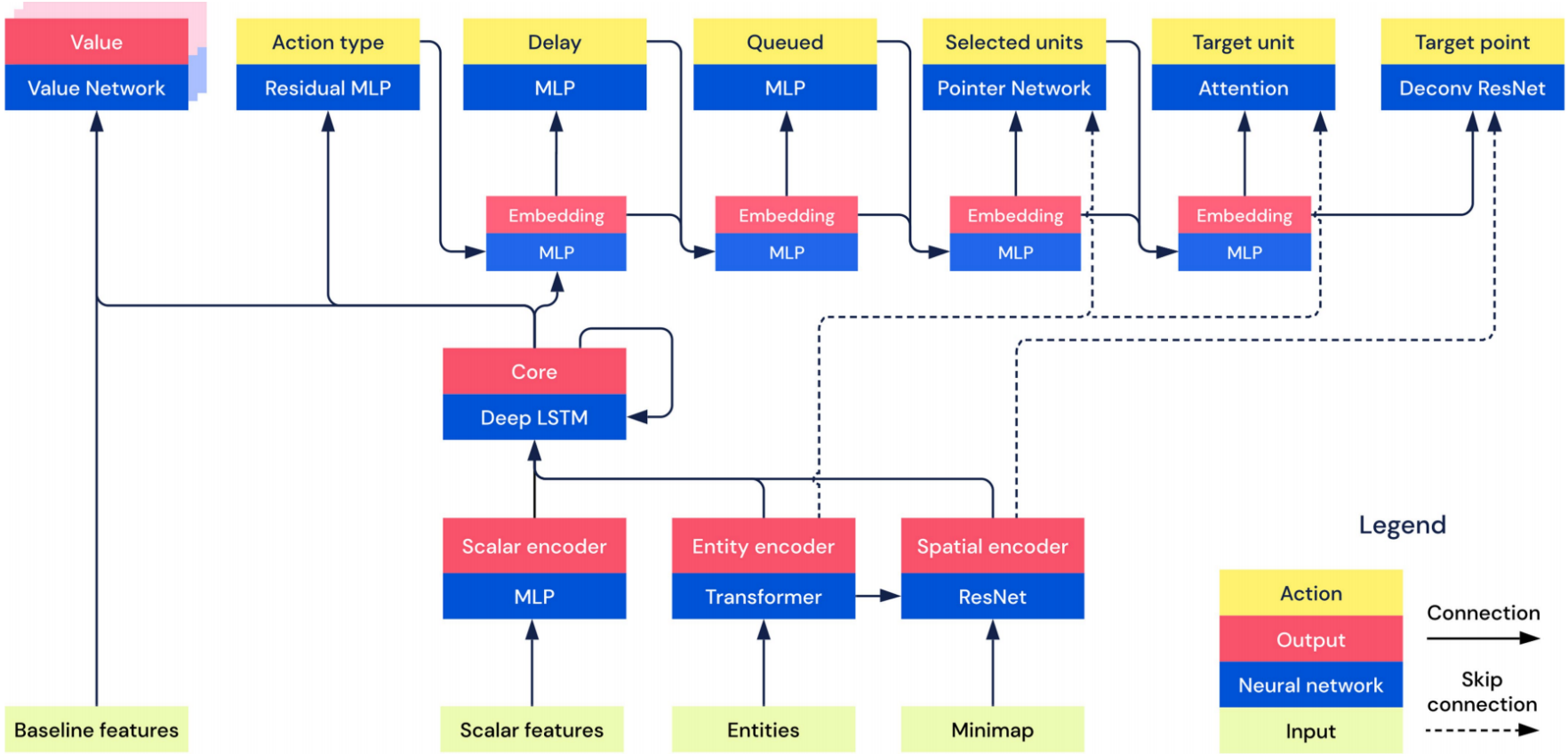
AlphaStar的动作信息主要分为6个部分，分别为动作类型（action type）、选中的单元（selected units）、目标（target）、执行动作的队列（queued）、是否重复（repeat）以及延时（delay），各个部分间是有关联的。



AlphaStar和DeepNash

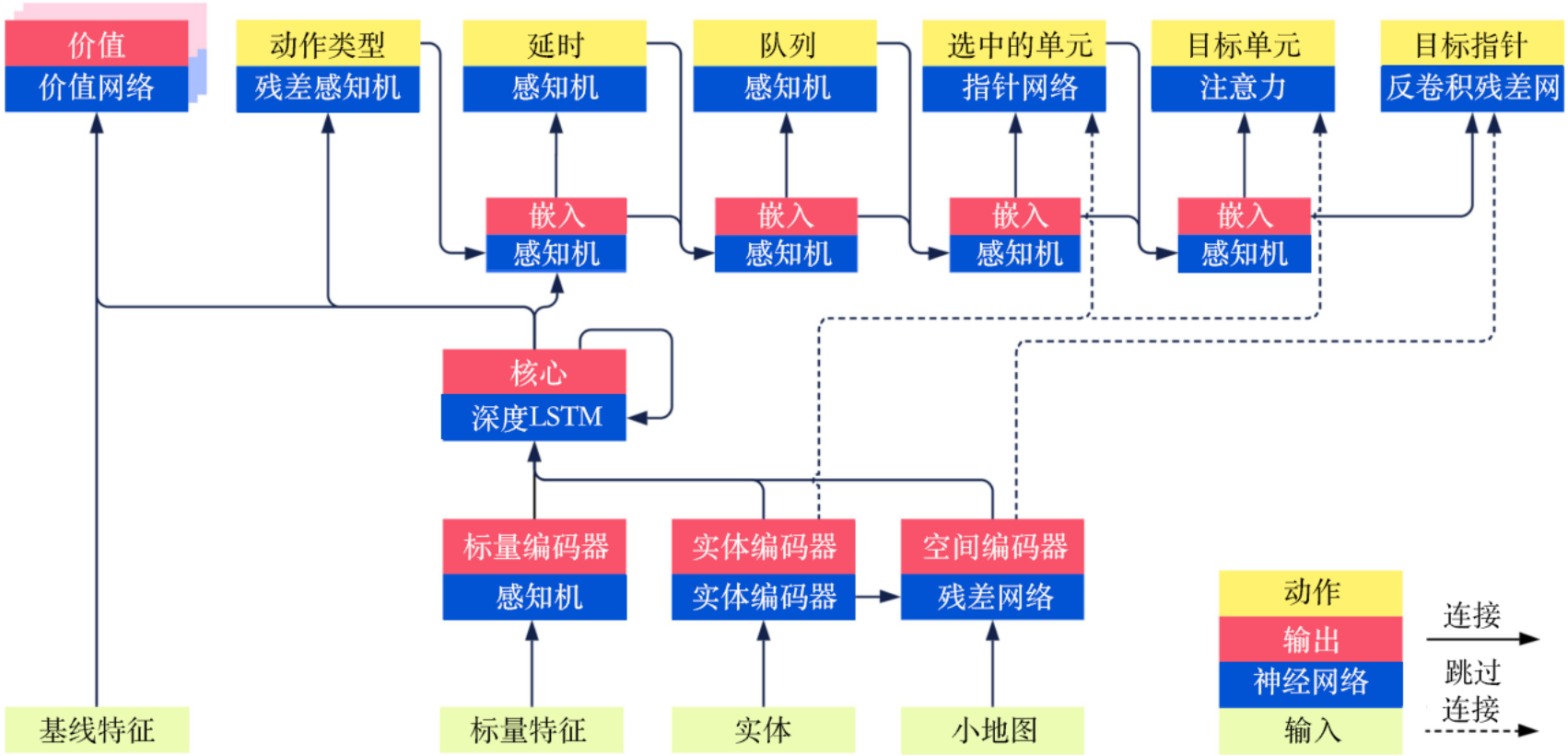
- 第一部分——动作类型，即下一次要进行的动作的类型是移动小兵、升级建筑还是移动小窗口的位置等。
- 第二部分——选中的单元，承接第一部分，例如我们要进行的动作类型是移动小兵，那么我们就应该选择具体移动哪一个小兵。
- 第三部分——目标，承接第二部分，我们移动小兵A后，是要去地图的某一个位置还是去攻击对手的哪一个目标等，即选择目的地或攻击的对象。
- 第四部分——执行动作的队列，即是否立即执行动作，对于小兵A，是到达目的地后直接进行攻击还是原地待命。
- 第五部分——是否重复，如果需要小兵A持续攻击，那么就不需要再通过网络计算得到下一个动作，直接重复上一个动作即可。
- 第六部分——延时，即等候多久后再接收网络的输入，可以理解为一个操作的延迟。

AlphaStar和DeepNash



Extended Data Fig. 3 | Overview of the architecture of AlphaStar. A detailed description is provided in the Supplementary Data, Detailed Architecture.

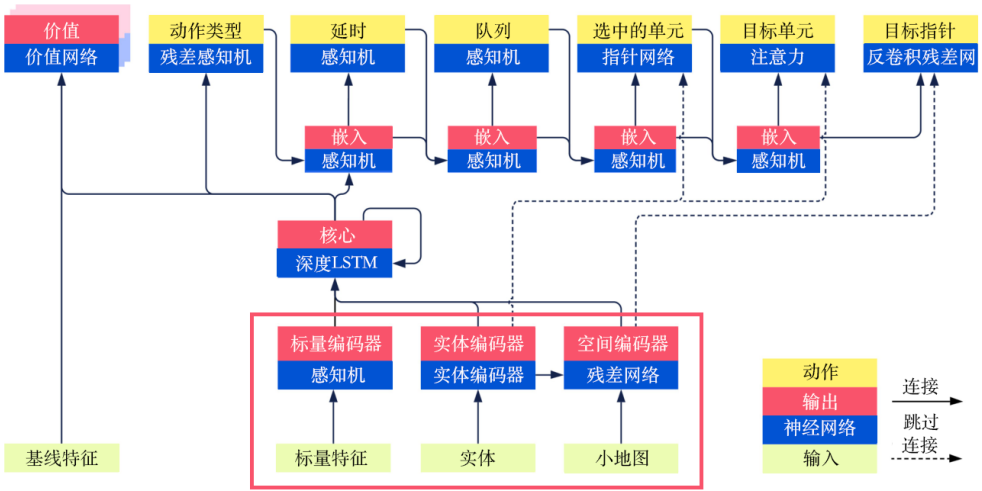
AlphaStar和DeepNash



AlphaStar和DeepNash

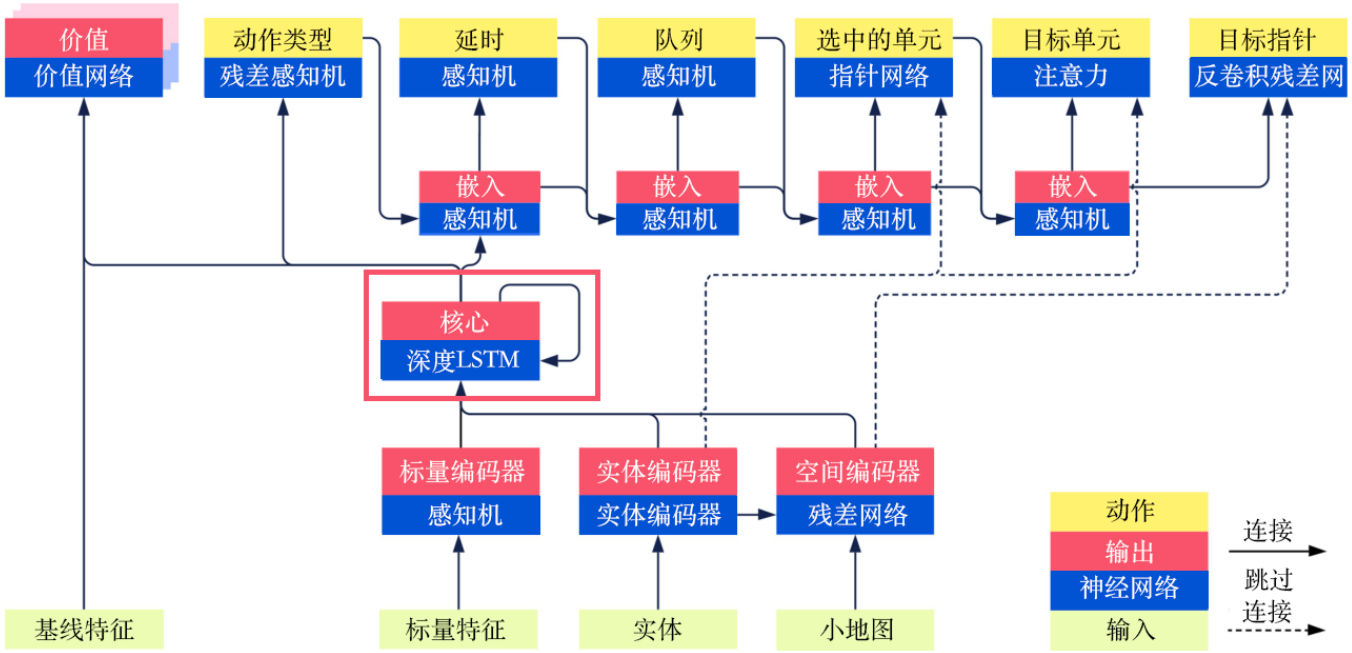
模型的输入部分主要有3个部分：标量特征（scalar features)，例如前面描述的玩家等级以及小窗口的位置等信息；实体（entities)，是向量，即前面所叙述的一个建筑或一个小兵的当前所有的属性信息；小地图（minimap），即图像数据。

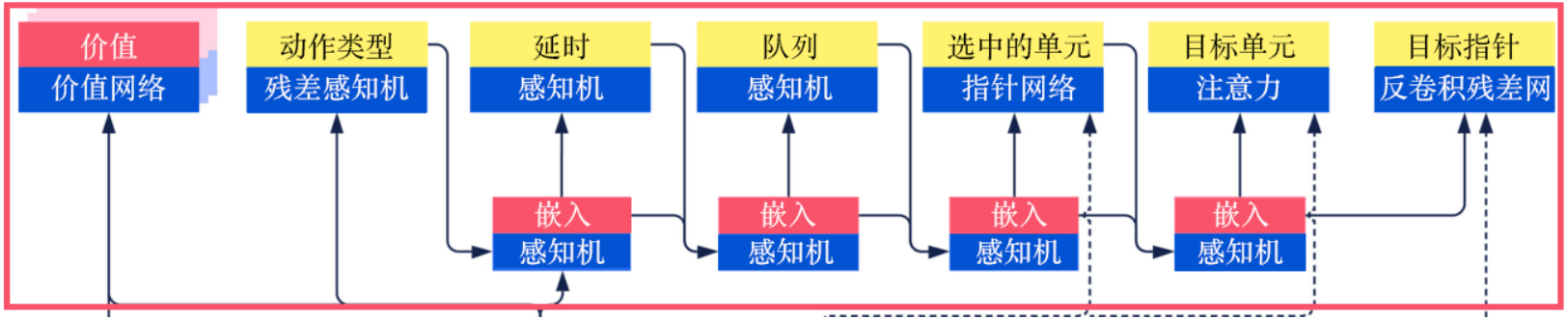
- 对于标量特征，使用多层感知机（multilayer perceptron，MLP），就可以得到对应的向量，可以认为是一个嵌入过程。
- 对于实体，使用自然语言处理中常用的Transformer架构作为编码器（encoder）。
- 对于小地图，使用图像中常用的ResNet架构作为编码器，得到一个定长的向量。



AlphaStar和DeepNash

中间过程比较简单，即通过一个深度长短期记忆网络模块融合3种当前状态下的嵌入并进行下一时刻的输出，并且将该输出分别送入价值网络（value network）、残差多层感知机（residual MLP）以及动作类型的后续的多层感知机中。



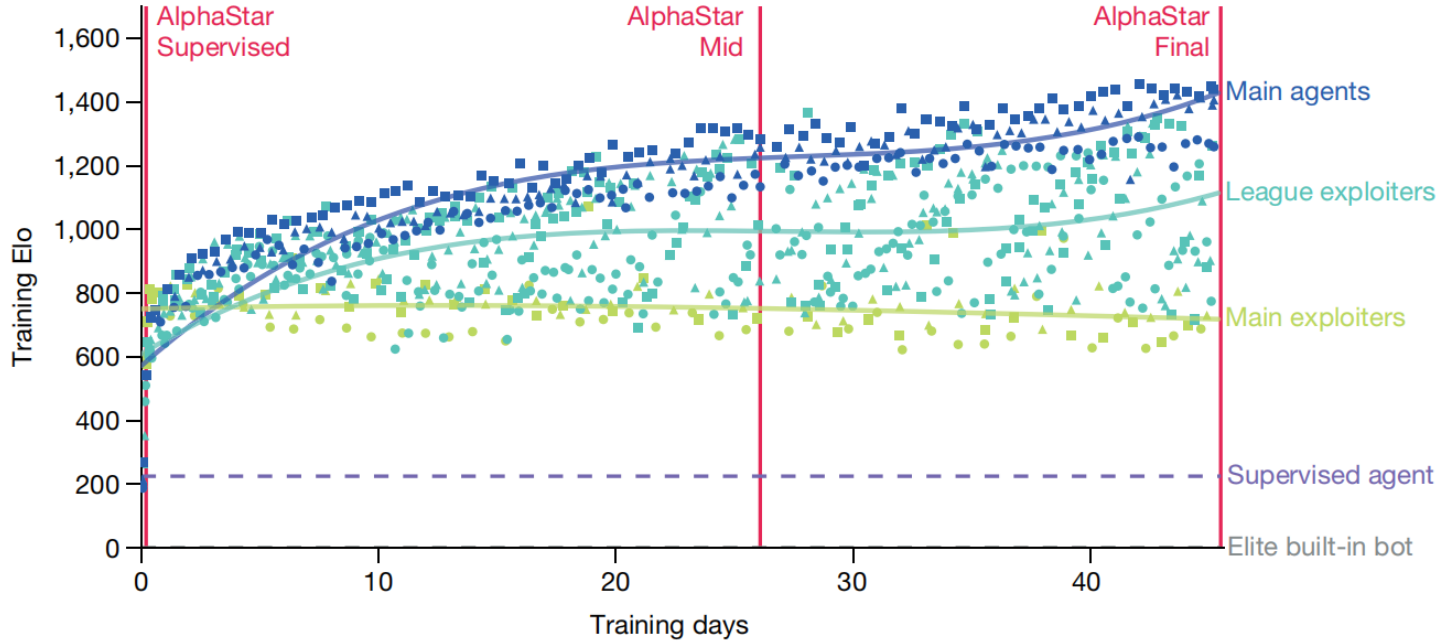


输出的动作:

- 首先是动作类型：使用深度长短期记忆网络的嵌入向量作为输入，使用残差多层感知机得到动作类型的Softmax激活函数的输出结果，并将其传给下一个子模型进行嵌入。
- 然后是延时：将动作类型嵌入的结果以及深度长短期记忆网络的结果一起输入多层感知机后得到结果，并传给下一个子模型进行嵌入。
- 接下来是执行动作的队列：将延时的结果以及嵌入的结果一起输入多层感知机后得到结果，并传给下一个子模型进行嵌入。
- 然后是选中的单元：将队列的结果、嵌入的结果以及实体编码后的全部结果（非平均的结果）一起送入指针网络（pointer network）中得到结果，并传给下一个子模型进行嵌入。这里的指针网络的输入是一个序列，输出是另外一个序列，并且输出序列的元素来自输入的序列。其主要用于自然语言处理中，在这里很适合我们选中的单元的计算。
- 接着是目标单元（target unit）和目标指针（target point）两者二选一，对于目标单元，使用注意力（attention）机制得到最优的动作作用的一个对象；对于目标区域，使用反卷积残差网络，将嵌入的向量反卷积为地图的大小，从而执行目标移动到某一点的对应动作。

AlphaStar和DeepNash

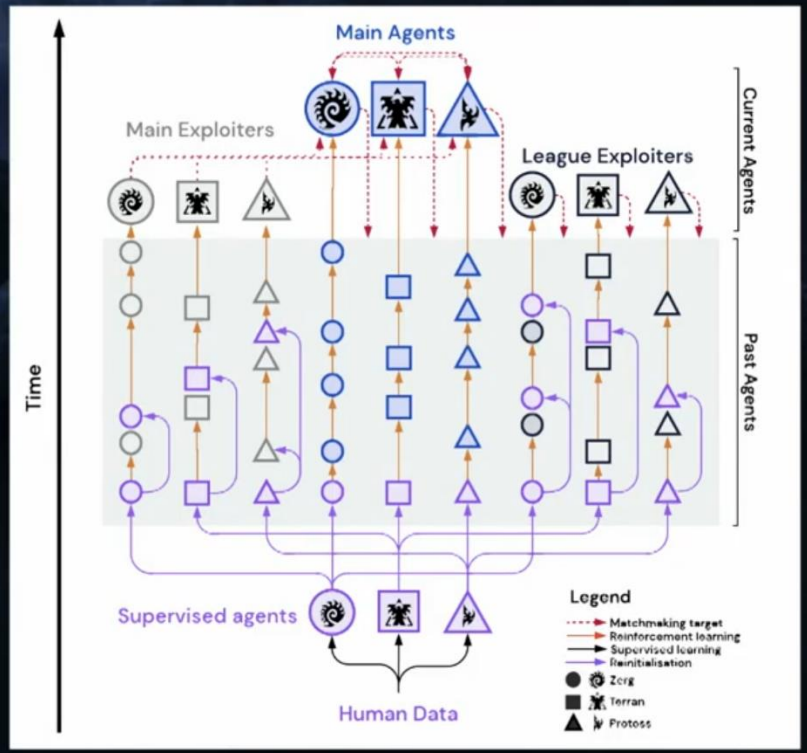
对于上面复杂的模型，AlphaStar究竟如何进行训练呢？总结下来一共分为4个部分，即监督学习（主要是解决训练的初始化问题）、强化学习、模仿学习（配合强化学习）以及多智能体学习或自学习（面向对战的具体问题）

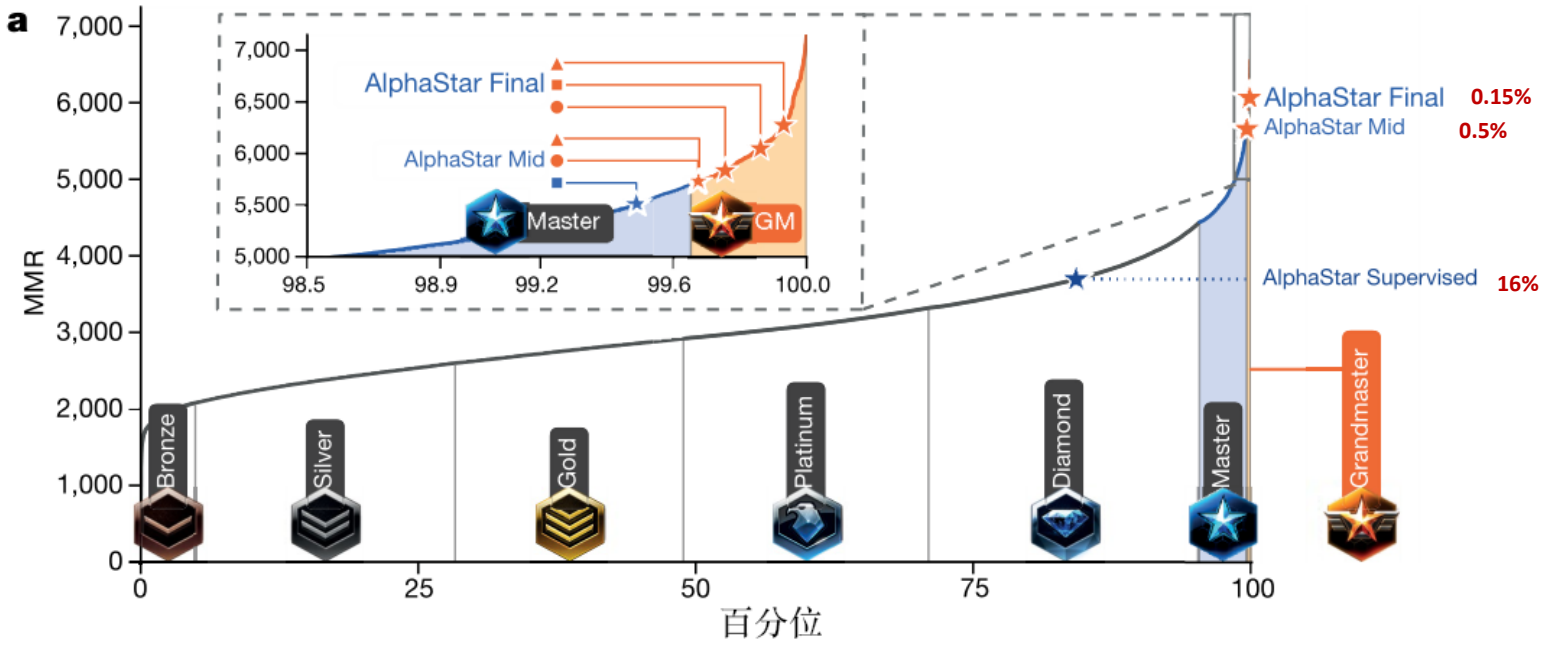


AlphaStar和DeepNash

AlphaStar League

- Prioritised fictitious play
  - Play opponents in proportion to their win-rate against agent
  - Ensures the agent does not forget old strategies
- Exploiters discover counter-strategies
  - By specialising against main agents or against whole league
  - Thereby making main agents more robust
- Initialised by supervised learning
  - Explores human-like behaviours



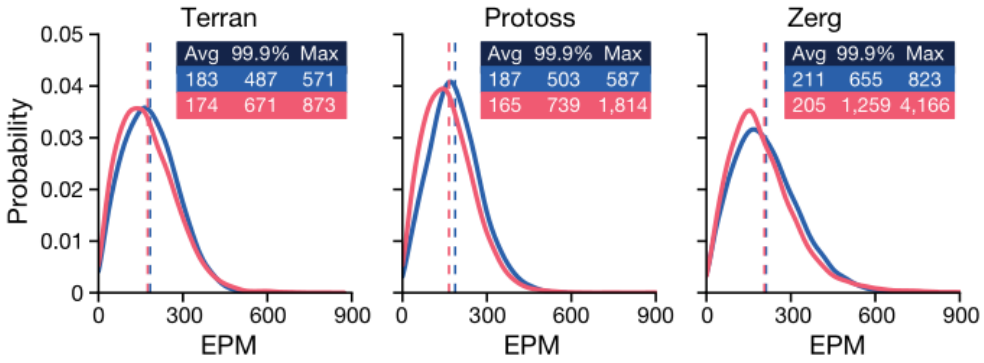


b

Opponent race

AlphaStar race				
	6,275 99.93% 25/30	6,196 99.91% 11/14	– – 4/4	6,297 99.94% 10/12
	6,048 99.86% 18/30	5,991 99.83% 4/8	6,209 99.92% 4/7	5,971 99.82% 10/15
	5,835 99.76% 18/30	5,755 99.7% 8/14	5,531 99.51% 5/10	6,500 99.96% 5/6

c



AlphaStar和DeepNash

关于AlphaStar的总结如下:

1. AlphaStar设计了一个高度可融合图像、文本、标量等信息的神经网络架构，并且对于网络设计使用了自回归（autoregressive）技巧，从而解耦了结构化的动作空间。
2. 其融合了模仿学习和监督学习的内容，例如人类统计量 Z的计算方法。
3. 其拥有复杂的深度强化学习方法以及超复杂的训练策略。
4. 其完整模型的端到端训练过程需要大量的计算资源。对于此，原文表述如下：每个智能体使用32个第三代张量处理单元（tensor processing unit，TPUs）进行了44天的训练；在训练期间，创建了近900个不同的游戏玩家。

DeepNash

AI 在此前尚未掌握的经典棋类游戏 Stratego（西洋陆军棋）中，表现出了人类专家级一般的水准——以97%的最低胜率击败了其他 AI 机器人；在 Gravon 平台上与人类专业玩家对弈，取得了84%的总胜率，在年初至今和历史排行榜上都排在前三名。

AlphaStar和DeepNash

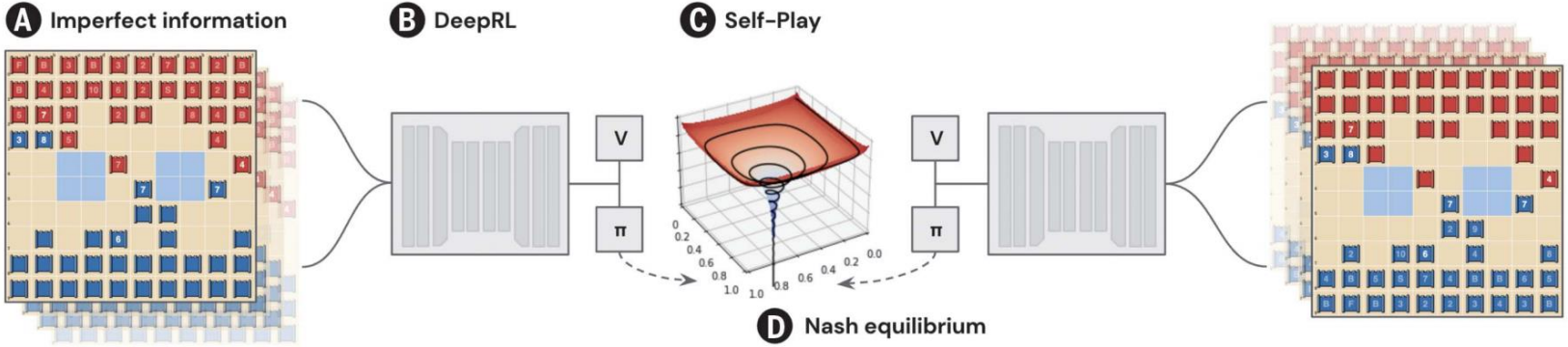
- 在 Stratego 中，双方各有代表元帅（Marshal）、将军（General）、上校（Colonel）、中校（Major）、上尉（Captain）、中尉（Lieutenant）、士官（Sergeant）、除雷兵（Miner）、斥侯（Scout）、间谍（Spy）、地雷（Bomb）、军旗（Flag）的棋子。
- 具体游戏规则为：两方将所有己棋竖立、以正面朝后的方式排布，然后轮流移动一枚己棋；可以将棋子沿纵横方向移动一格至空格或敌棋处，但需要维持正面朝后；如果一方棋子到达敌棋处，便将两棋公开，一般胜方这一棋子会被放回原位且正面继续朝后，输方这一棋子则被移除游戏。

AlphaStar和DeepNash

- Stratego 诞生于 1947 年，与中国陆军棋不同，其军衔、棋子数量较多，棋盘设计较为简单，没有铁路、行营，也没有裁判，当两方棋子相遇后，才会揭开来判断大小。二者的相同之处，都是以夺得对方军旗或消灭所有可移动的棋子为胜利标志。
- Stratego 是一种不完全信息游戏。与之相反，国际象棋、跳棋、日本将棋和围棋可看作完全信息博弈，因为双方完全清楚游戏规则，当前局面对方可能的下法等信息。
- 而且，Stratego 具有非常复杂的结构，其博弈树具有10^535种可能的状态，比无限德州扑克（10^164）和围棋（10^360）都要多。
- 另外，在特定情况下，Stratego 玩家需要在游戏开始时推理出多于10^66对可能的排布，而在德州扑克中，这一数字仅为 10^6；完全信息游戏则没有这一阶段，相对更为简单。

AlphaStar和DeepNash

由于Stratego的博弈树复杂性如此之大，DeepNash无法采用其他AI在玩游戏时用的蒙特卡洛树搜索。DeepNash采用的，是一种新的博弈论算法思想——正则化纳什动态规划（Regularized Nash Dynamic，R-NaD）。它引导着DeepNash，让它的学习行为朝着纳什均衡的方向发展。



Replicator dynamics: $\frac{d}{d\tau}\pi_\tau^i(a^i) = \pi_\tau^i(a^i)\left[Q_{\pi_\tau}^i(a^i) - \sum_{b^i}\pi_\tau^i(b^i)Q_{\pi_\tau}^i(b^i)\right]$

Reward transformation: $r^i(\pi^i, \pi^{-i}, a^i, a^{-i}) = r^i(a^i, a^{-i}) - \eta\log\left(\frac{\pi^i(a^i)}{\pi_{\text{reg}}^i(a^i)}\right) + \eta\log\left(\frac{\pi^{-i}(a^{-i})}{\pi_{\text{reg}}^{-i}(a^{-i})}\right)$

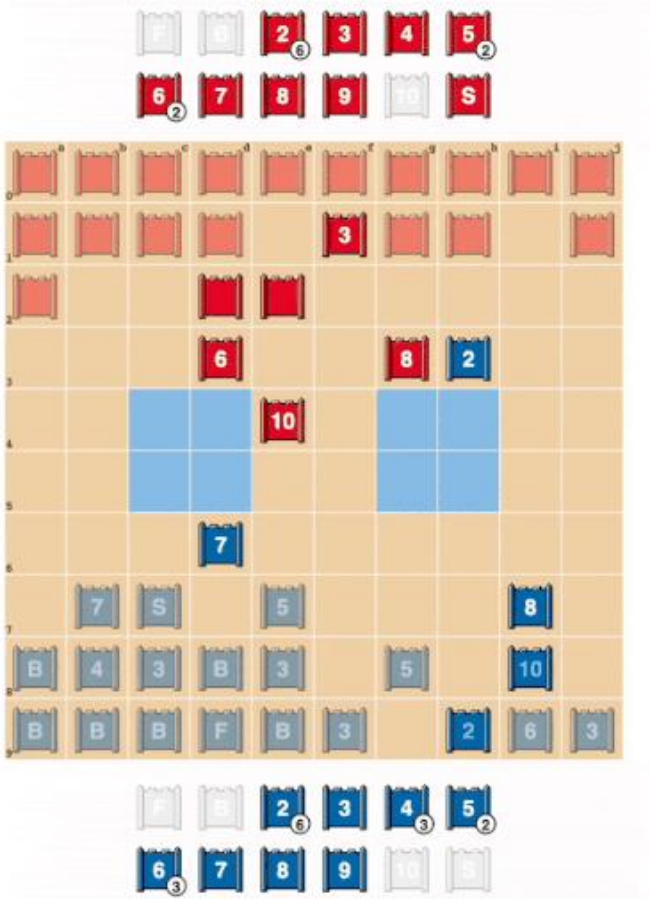
Fig. 2. Overview of R-NaD. (**A**) Overview of the R-NaD approach at scale underlying DeepNash, which allows for learning to play the imperfect information game Stratego. (**B** to **D**) R-NaD learns a policy represented by a deep neural network (B) through self-play from scratch (C) and aims at converging to a Nash equilibrium (D). The approach relies on two core ideas to reach convergence: replicator dynamics and reward transformation. Their equations are shown for illustrative purposes in their simplest form.

AlphaStar和DeepNash

博弈论之父冯·诺伊曼：现实生活中充满“虚张声势”、“欺骗的小伎俩”以及“猜测别人会认为我打算做什么”。

DeepNash的两种唬人技巧：主动吓唬（positive bluffing）与被动吓唬（negative bluffing）。

所谓**主动吓唬**，就是假装自己的棋子等级很高，威慑对手。简单来说就是「虚张声势」。
在扑克中，优秀的玩家会玩心理战，即使在我方弱势的情况下，也要让对方形成威慑。
DeepNash也学会了这种虚张声势的策略——**被动吓唬**（negative bluffing）。
也就是我们常说的「扮猪吃老虎」：将自己等级高的棋子伪装成等级低的棋子，等到对方上当后再一举拿下。



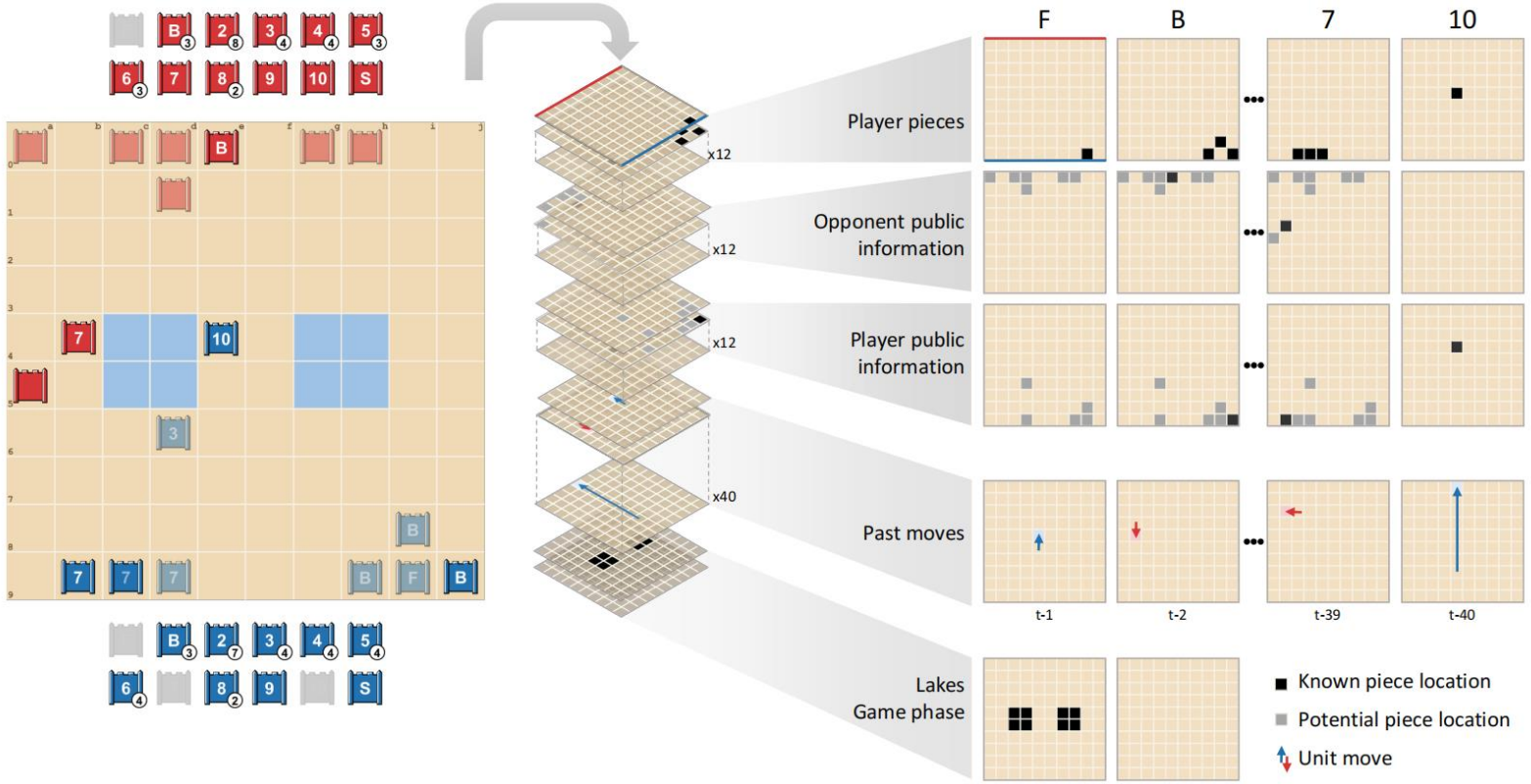


Figure S2: The input of the neural network is a single tensor encoding the position of pieces, the currently known information of both opponent and own pieces (whether a piece moved or was revealed), a limited move history and the position of the lakes.

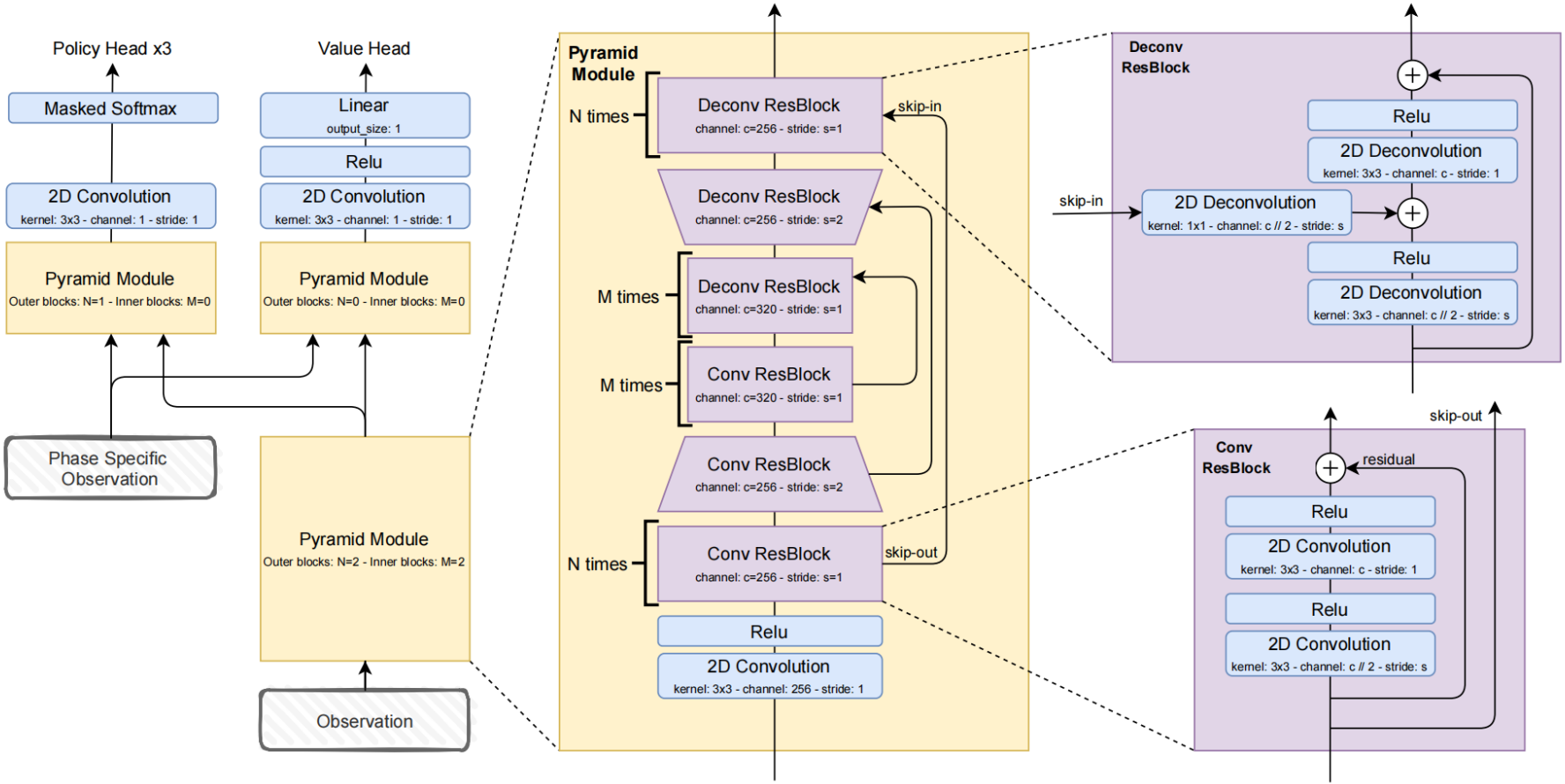


Figure S3: Network implementation details. When applying striding, residual connections are also processed by a convolution layer with 1×1 kernel (hidden for clarity).

提纲

一、博弈论

二、智能博弈

三、AlphaStar和DeepNash

四、踢足球、躲猫猫、追逃、空战

踢足球

Deepmind

2022
3v3仿真

SCIENCE ROBOTICS | RESEARCH ARTICLE

ARTIFICIAL INTELLIGENCE

From motor control to team play in simulated humanoid football

Siqi Liu*†, Guy Lever*†, Zhe Wang†, Josh Merel, S. M. Ali Eslami, Daniel Hennes, Wojciech M. Czarnecki, Yuval Tassa, Shayegan Omidshafiei‡, Abbas Abdolmaleki, Noah Y. Siegel, Leonard Hasenclever, Luke Marris, Saran Tunyasuvunakool, H. Francis Song, Markus Wulfmeier, Paul Muller, Tuomas Haarnoja, Brendan Tracey, Karl Tuyls, Thore Graepel§, Nicolas Heess†

2023
1v1实物

DeepMind

2023-4-27

Learning Agile Soccer Skills for a Bipedal Robot with Deep Reinforcement Learning

Tuomas Haarnoja[*,1], Ben Moran[*,1], Guy Lever[*,1], Sandy H. Huang[*,1], Dhruva Tirumala[1], Markus Wulfmeier[1], Jan Humplik[1], Saran Tunyasuvunakool[1], Noah Y. Siegel[1], Roland Hafner[1], Michael Bloesch[1], Kristian Hartikainen[2,4], Arunkumar Byravan[1], Leonard Hasenclever[1], Yuval Tassa[1], Fereshteh Sadeghi[3,4], Nathan Batchelor[1], Federico Casarini[1], Stefano Saliceti[1], Charles Game[1], Neil Sreendra, Kushal Patel, Marlon Gwira, Andrea Huber[1], Nicole Hurley[1], Francesco Nori[1], Raia Hadsell[1] and Nicolas Heess[1]

[*]Equal contributions, [1]DeepMind, [2]University of Oxford, [3]Google, [4]Work done at DeepMind

踢足球

DeepMind将 AI 技术应用于模拟踢足球之中，并将研究论文发表于Science Robotics上，，其研究了在模拟足球场的环境下控制多智能体自身的运动和长距离运动决策的集成方法，通过强化学习算法构建出多时间、多空间、多主体下的 AI 足球比赛。

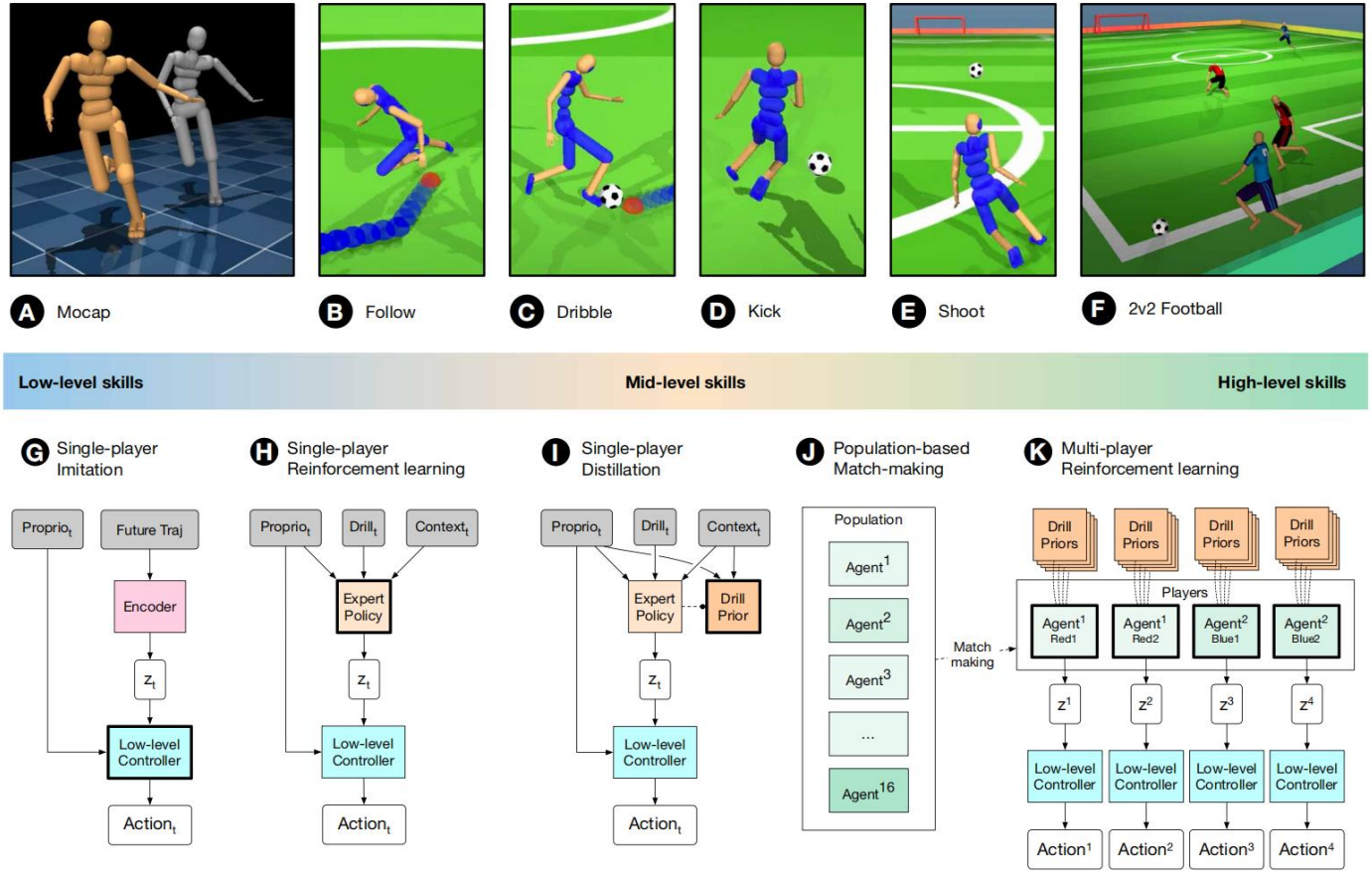


Fig. 1. Overview of the proposed learning framework. Components shown in bold were optimized during their corresponding stage and transferred to the subsequent stages. (**A** and **G**) An example frame of the motion capture behavior (gray) and the low-level controller optimized to reproduce matching behaviors (beige). (**B** to **E** and **H** and **I**) Illustrations of the four mid-level drill tasks and their corresponding training procedures to obtain per-task expert policy and subsequently per-task, skill prior. (**F**, **J**, and **K**) An example frame of a 2v2 football match with the two teams sampled from the population of agents. The agents were optimized end to end by RL, reusing the low-level controller while regularized toward pretrained skill priors. Proprio$_t$ and Context$_t$ denote the proprioceptive and game context (e.g., ball position) observations, and Drill$_t$ denotes the drill-specific observation (e.g., desired ball position in the dribble drill task). Future Traj denotes the future trajectory. See Materials and Methods and section S1.4 for more details of each stage.

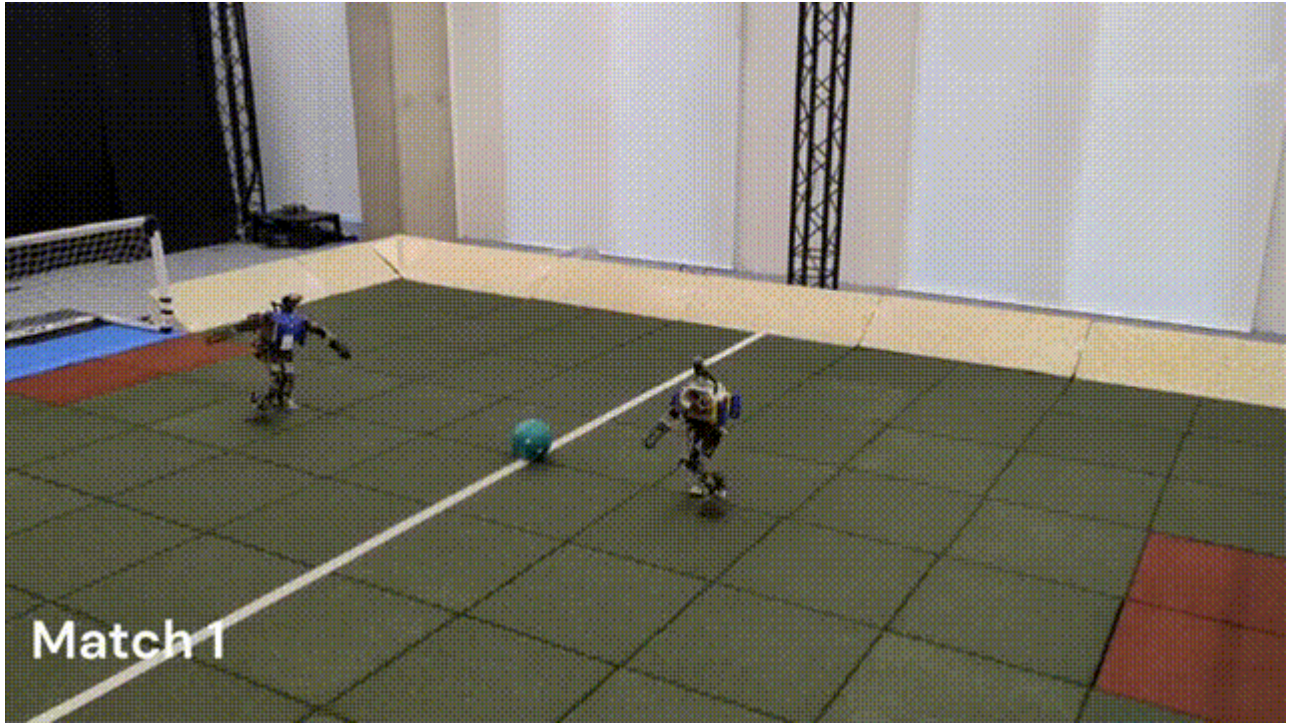
首先，为了诱导 AI 球员产生最初的运动行为，项目组创建了人类动作行为片段的运动原始模块，能够根据抽象的运动指令产生瞬时的仿人类运动，自动生成原始动作片段中不存在的动作序列。

然后，为了训练 AI 球员进行长距离的运动（奔跑等），算法利用在单次的足球训练任务中预先训练过的运动模型，训练多智能体进行中等水平的足球运动，由此产生的技能被表示为可重复使用的足球技能，可以进一步随机产生与足球训练相关的不同行为，将行为正规化。

最后，项目组还用注意力感知算法模拟出 AI 球员增量训练的全过程，将球员不断转移至球员数量更多的比赛中进一步训练。

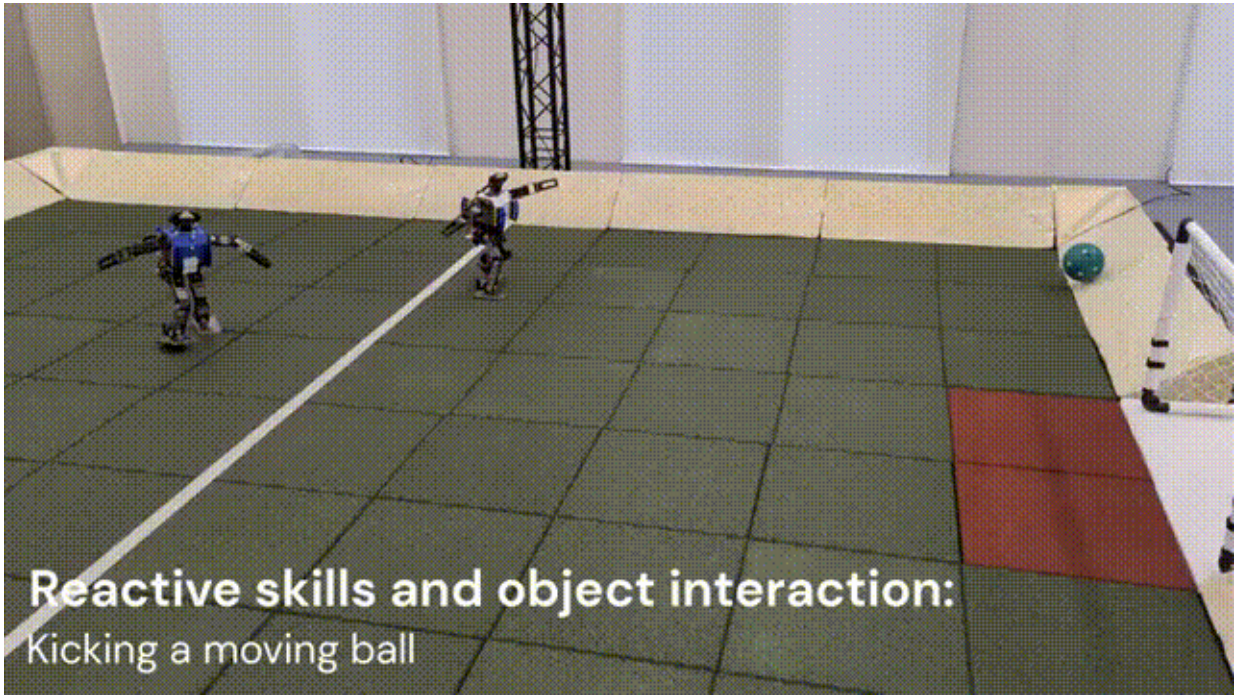
踢足球

DeepMind研究团队通过使用深度强化学习训练了一个具有20个驱动关节的微型仿生机器人OP3，在为其设置了多种单一行为策略后，OP3可以逐渐掌握在动态环境中组合自身行为应对复杂情况的能力，例如，两个OP3就可以进行简单的一对一足球比赛：



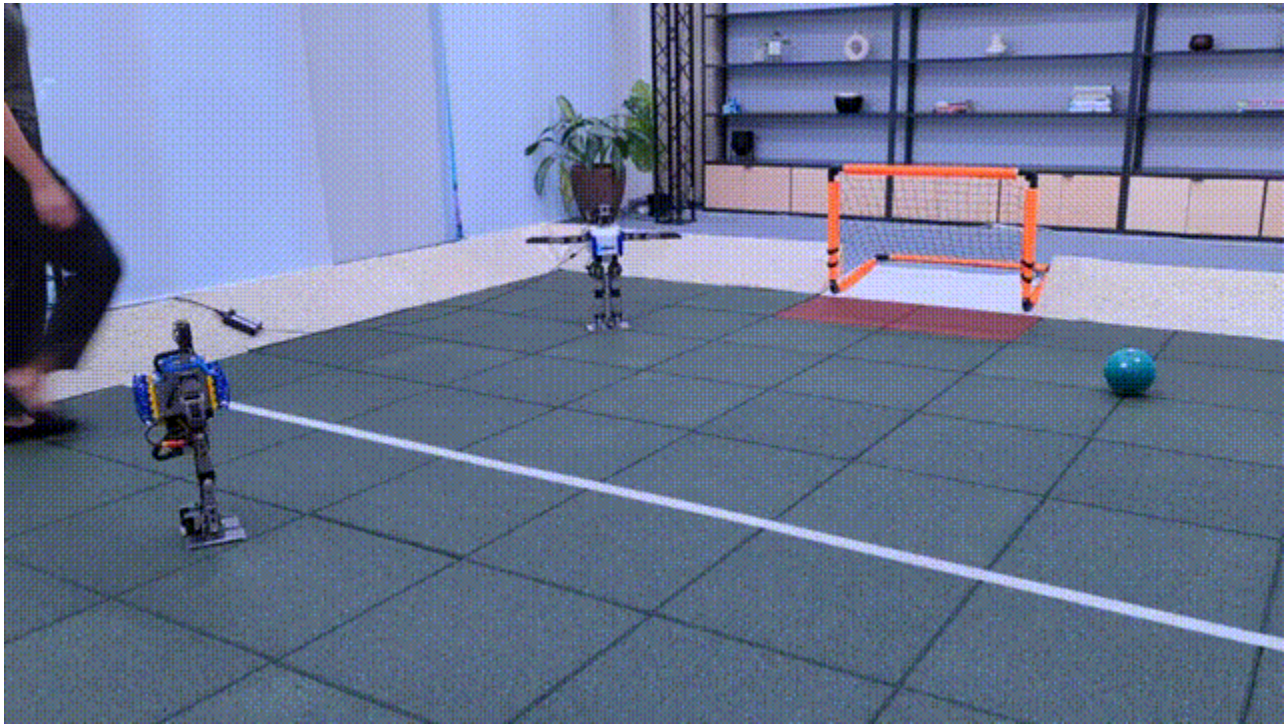
踢足球

在对OP3进行训练时，研究团队首先对其训练单一的行为技能，然后使其通过自我博弈方式端到端的组合这些单一技能，通过这种方式产生的组合行为展现出了惊人的运动潜能。**例如OP3可以流畅地完成行走、转身、运球、射门等复杂足球行为。**



踢足球

除了上述专业足球动作，研究团队还着重考虑了**OP3对外界环境的适应能力，例如快速跌倒恢复动作**，如下图所示，如果将运动中的OP3直接推到，它能够快速平稳的重新站立，展现出了较强的环境适应能力。



踢足球

研究团队首先在一个定制的足球仿真环境中训练了智能体，然后将策略转移到对应的真实环境中，该环境由一个长5米、宽4米的足球场构成，其中设置了两个球门，每个球门的开口宽度为0.8米。**在仿真环境和真实环境中，足球场的周围都设置有坡道，确保球保持在边界内。**真实的足球场上铺有橡胶地板砖，以增加机器人与地面的摩擦力。

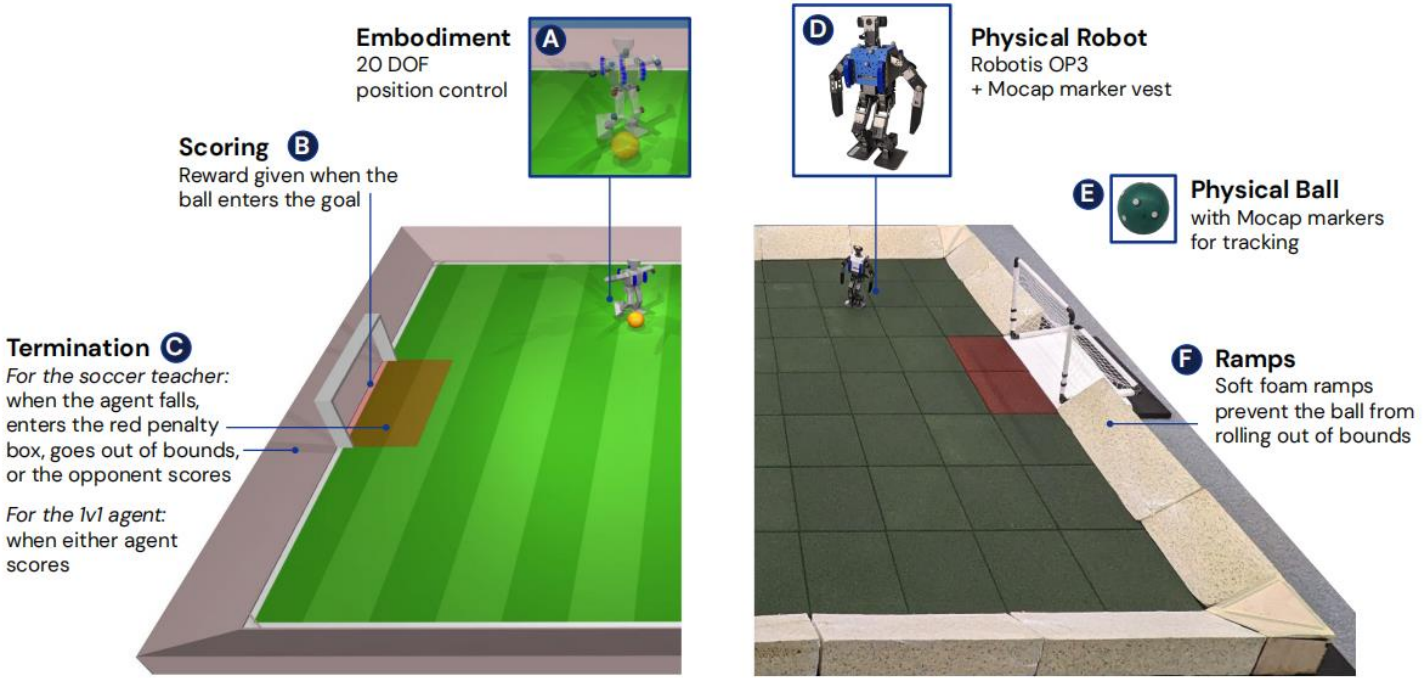


Figure 1 | We created matching simulated (left) and real (right) soccer environments. The pitch is 5 m long by 4 m wide. The real environment was also equipped with a motion capture (mocap) system for tracking the two robots and the ball.

踢足球

如果仅仅对智能体的目标函数进行简单的稀疏奖励训练，很难实现复杂的行为组合效果，因而本文通过将整体训练过程分为两个阶段来分步实现

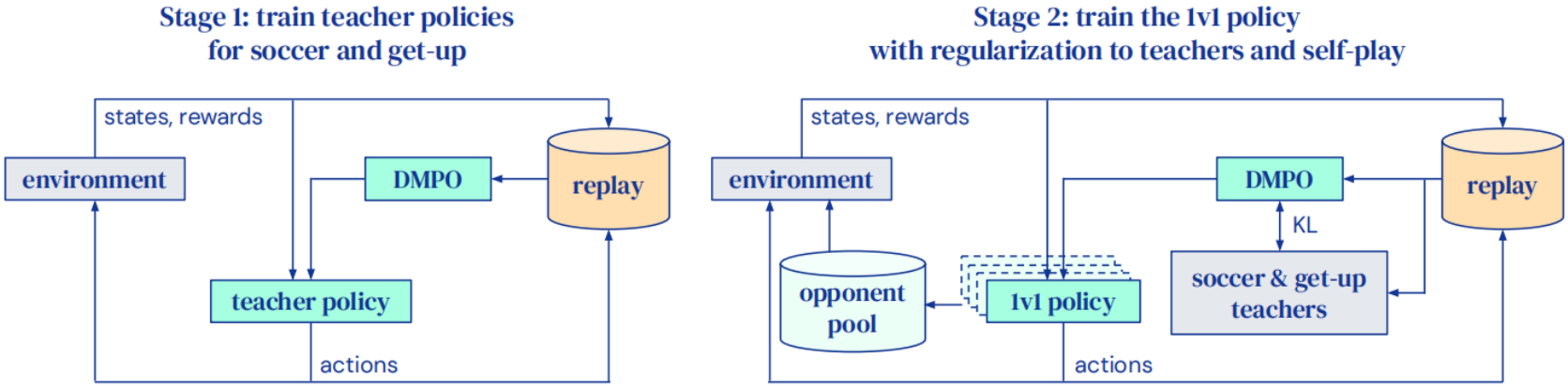


Figure 3 | We trained soccer agents in two stages. In the first stage (left), we train a separate soccer teacher and get-up teacher (Section 3.2.1). In the second stage (right), we distill these two teachers into a single agent that can both get up from the ground and play soccer (Section 3.2.2). The second stage also incorporates self-play: the opponent is uniformly randomly sampled from a pool that consists of policy snapshots from earlier in training. We found that this two-stage approach leads to qualitatively better behavior and improved sim-to-real transfer, compared to training an agent from scratch for the 1v1 soccer task.

踢足球

- 在第一阶段，研究团队首先训练了一个教师模型，教师模型主要使用两个特定技能进行训练，**即从地面起身（getting up from the ground）和进攻对手得分（scoring goals）。**需要注意的是，在训练进攻对手得分任务时，智能体必须处于站立状态，**如果没有对该条件进行限制，智能体会陷入一个局部最小值陷阱，即在地面上滚动来将球送入球门，而不是通过行走运球和射门，这是强化学习训练中常见的问题。**

- 在第二阶段，研究团队使用第一阶段训练得到的教师模型来指导智能体学习如何有效地对抗越来越强的对手。这里作者采用了**自我博弈**的形式，**即对手是从智能体的先前训练版本中随机采样得到的。**这是一种自动课程学习的方式，对手的强度随着智能体的改进而增加。**此外，为了提高后续策略迁移的泛化能力，作者在智能体训练过程中加入了域随机化、随机扰动和传感器噪声等增强手段。**

躲猫猫

蓝色小人（躲藏者）需要利用各种办法隐藏自己，避免进入红色小人（搜索者）的视野。每局游戏刚开始，搜索者会被锁定，给躲藏者一点时间。
在已经进行了 **4.81亿次**的“躲猫猫”游戏中，人工智能控制的小人不断开发出新玩法....

Published as a conference paper at ICLR 2020

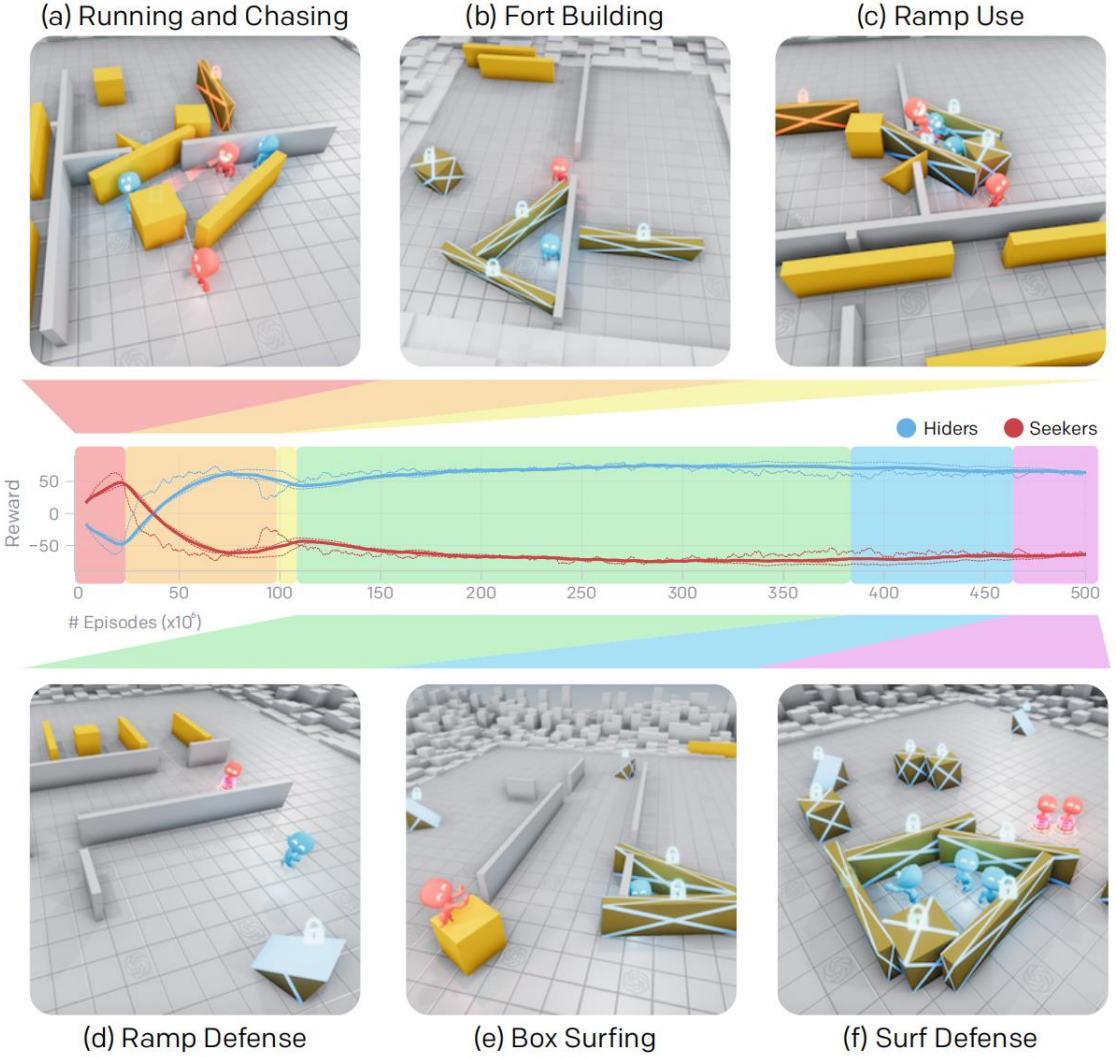
EMERGENT TOOL USE FROM MULTI-AGENT AUTOCURRICULA

Bowen Baker* OpenAI bowen@openai.com
Ingmar Kanitscheider* OpenAI ingmar@openai.com
Todor Markov* OpenAI todor@openai.com
Yi Wu* OpenAI jxwuyi@openai.com
Glenn Powell* OpenAI glenn@openai.com
Bob McGrew* OpenAI bmcgrew@openai.com
Igor Mordatch*† Google Brain imordatch@google.com

躲猫猫

- 通过几亿次简单的躲猫猫游戏，两支相互对立的 AI 智能体（agent）团队找到了复杂的游戏策略，其中甚至有工具的使用和团队协作。
- 测试结果表明，两支团队通过竞争模式进行自我改进的速度，远远超过任何单一智能体的进化速度。
- 研究人员认为，这些初步结果表明，通过简单的游戏规则、多智能体竞争和标准的大规模强化学习算法，可以刺激智能体在没有监督的情况下学习复杂的策略和技能，这是进化为更复杂人工智能的一个很好的方式。
- “我们没有告诉隐藏者或搜寻者要跑到盒子附近或利用盒子当做工具，”论文作者之一 Bowen Baker 说，“但通过竞争模式，它们为彼此创造了新的任务，使得另一个团队不得不适应。”

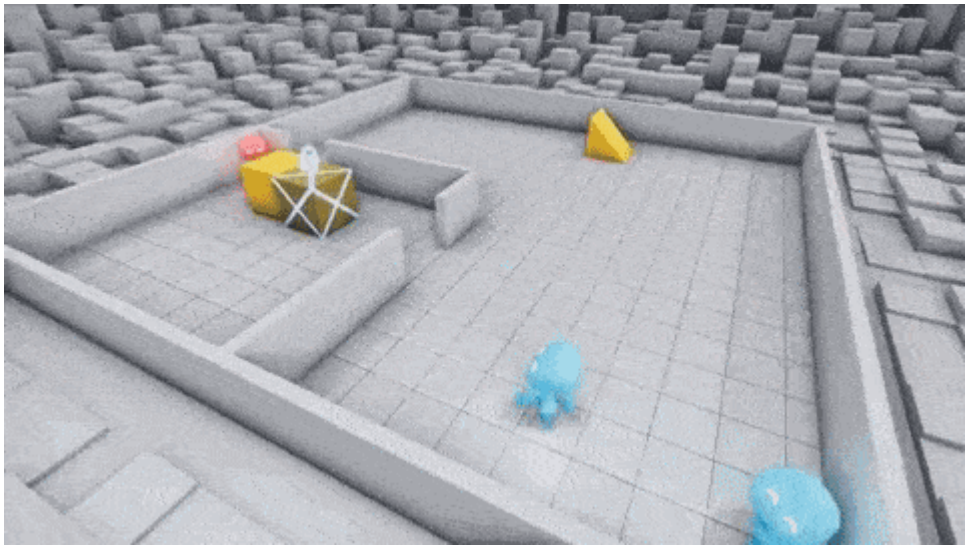
躲猫猫


(a) Running and Chasing
(b) Fort Building
(c) Ramp Use
Hiders
Seekers
Reward
50
0
−50
0
50
100
150
200
250
300
350
400
450
500
Episodes (x10^6)
(d) Ramp Defense
(e) Box Surfing
(f) Surf Defense

躲猫猫

第一阶段：追逐和隐藏

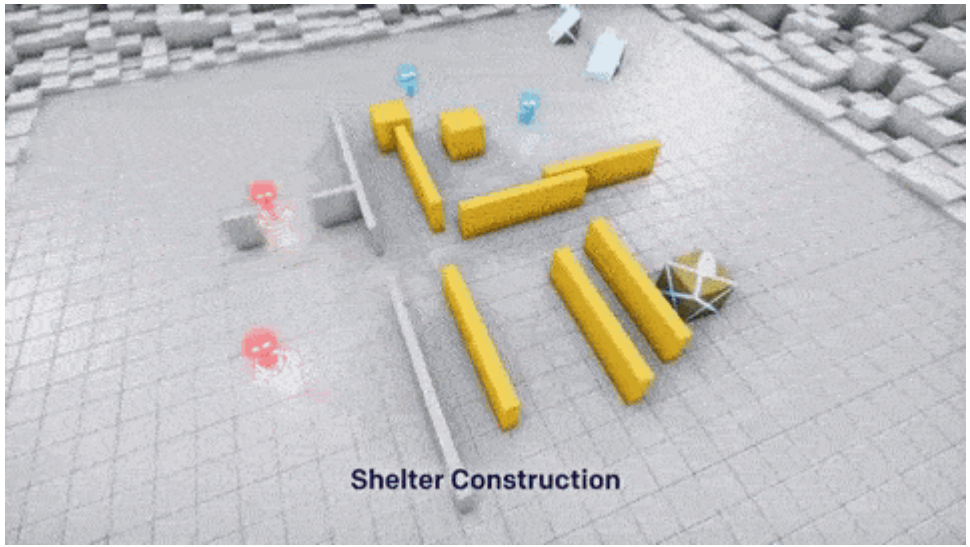
最初，AI什么都不会。但是在进行了数百万次“躲猫猫”之后，AI控制的小人发现了游戏的基本原理：在地图上互相追逐和躲藏。



躲猫猫

第二阶段：把自己藏起来

AI小人可以把游戏里的道具方块“锁定”（不能移动），而另一方无法解锁。又经过数百万次的练习，蓝色小人（躲藏者）学会了在适当的区域建造庇护所，这样红色小人（搜索者）就看不到它们。

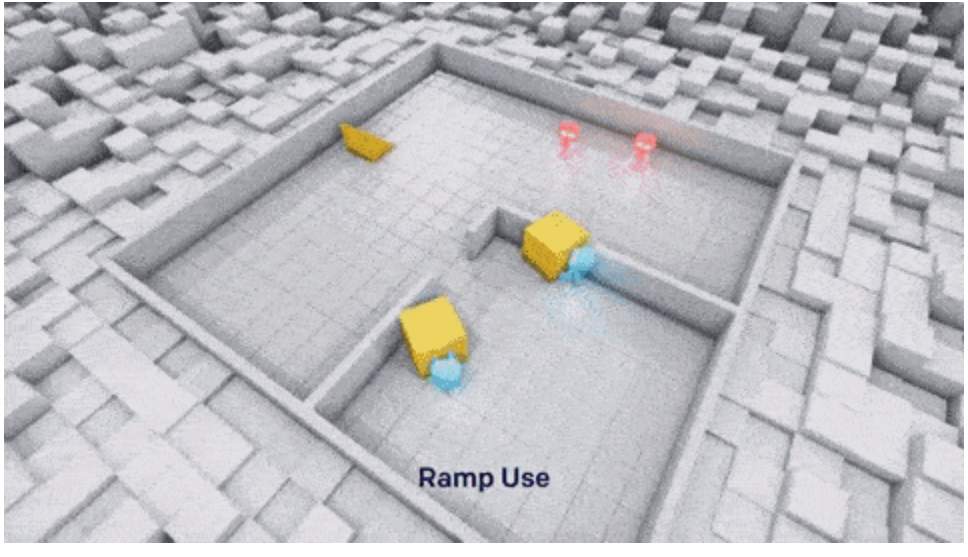


躲猫猫

第三阶段：突破掩体

又经过数百万次游戏，红色小人找到了一种办法：它们把斜坡方块拖到合适位置，然后爬上斜坡，找到蓝色小人。

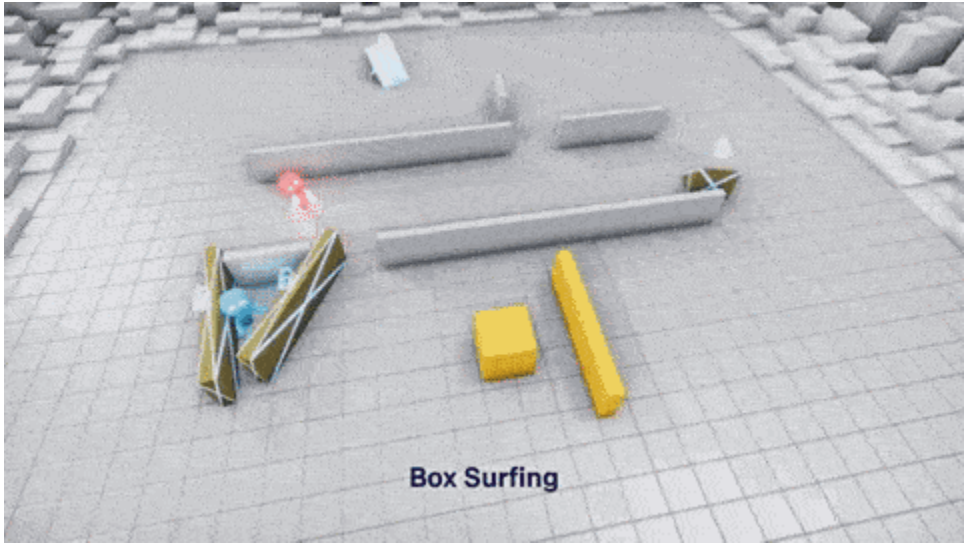
蓝色小人也不是吃素的，它们学会了反击：把斜坡方块锁定在适当位置，让红色小人无法移动它们。



躲猫猫

第四阶段：冲浪玩法

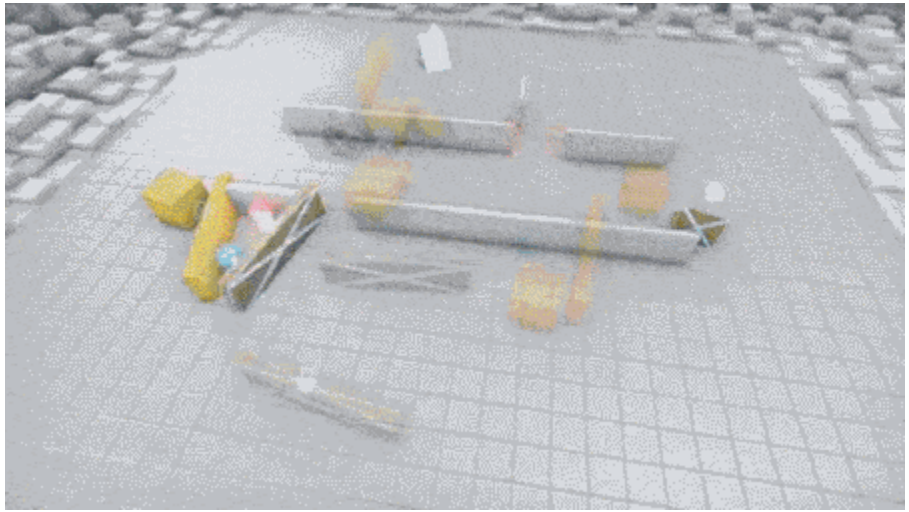
这个阶段有点超出预期：红色小人竟然学会了站在箱子上，使其像“冲浪板”那样滑行，并最终跳进庇护所。



躲猫猫

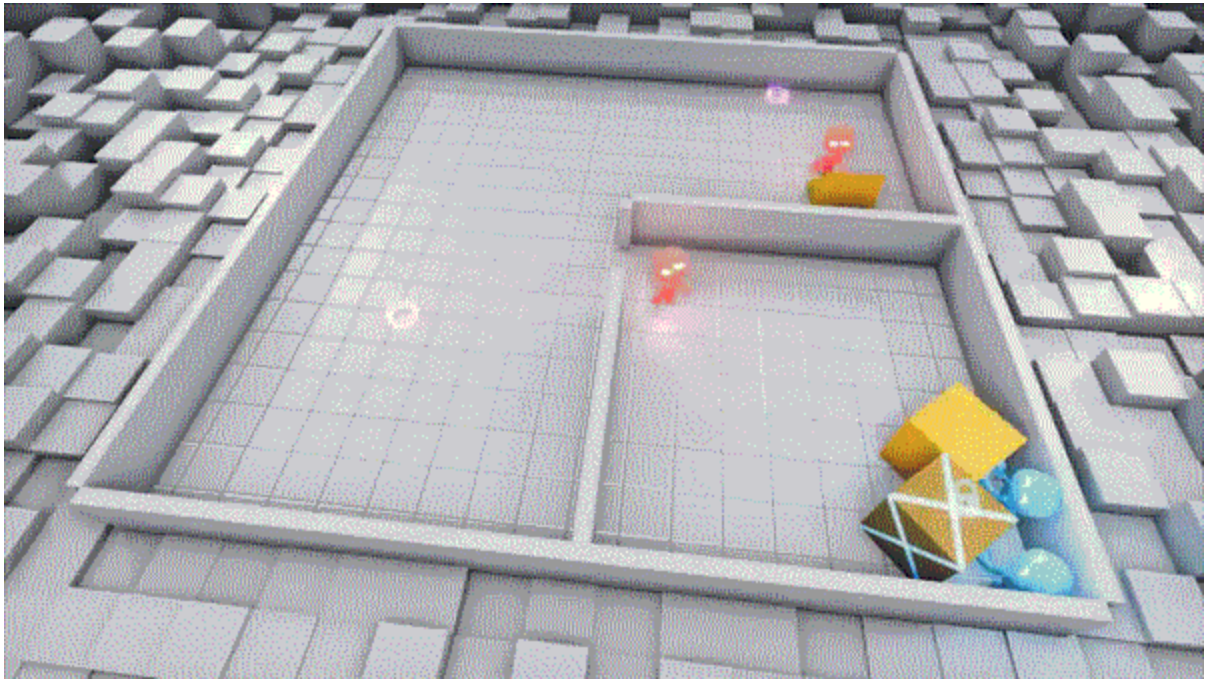
第五阶段：防止冲浪

无数次的失败之后，蓝色小人有了对策：它们锁定周围的一切，使红色小人没有任何可使用的工具。



躲猫猫

最新的进展，是红色小人学会了一招：以合适的角度推着斜坡方块奔跑，然后就可以让自己飞起来！



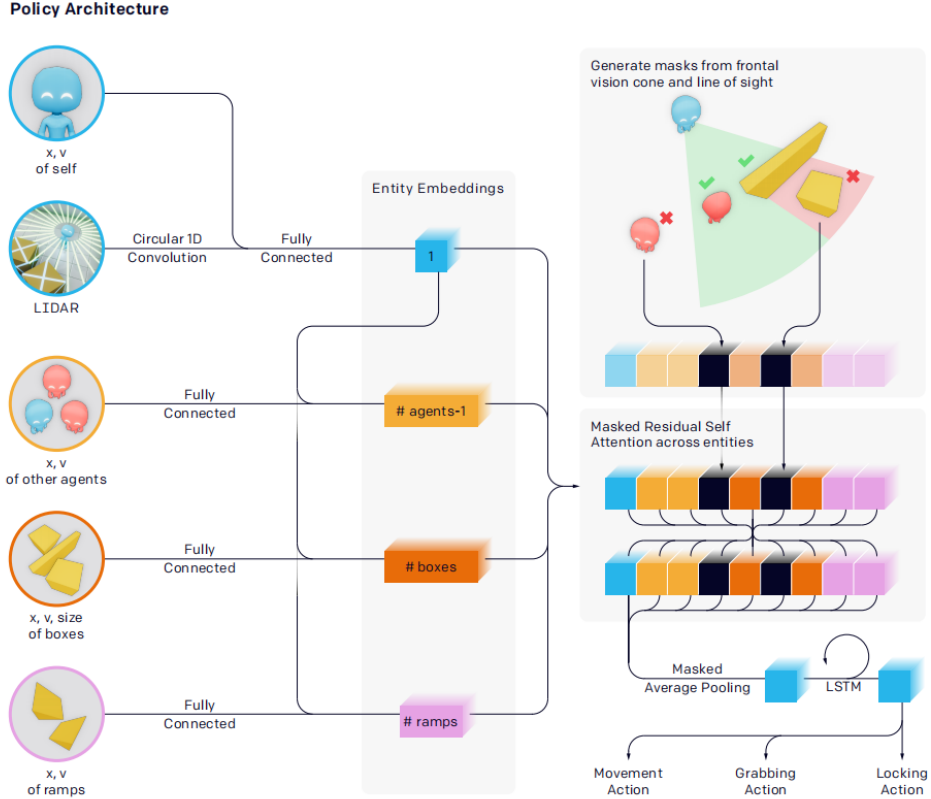


Figure 2: Agent Policy Architecture. All entities are embedded with fully connected layers with shared weights across entity types, e.g. all box entities are encoded with the same function. The policy is ego-centric so there is only one embedding of "self" and $(\#\text{agents} - 1)$ embeddings of other agents. Embeddings are then concatenated and processed with masked residual self-attention and pooled into a fixed sized vector (all of which admits a variable number of entities). x and v stand for state (position and orientation) and velocity.

追逃

高源博士与林天麟教授开发异构机器人团队围捕系统。

该论文旨在开发一个更加强健和智能的异构机器人团队围捕系统，以应对现代异构多机器人系统中不断增长的异构性和不同类型机器人数量。

Asymmetric Self-Play-Enabled Intelligent Heterogeneous Multirobot Catching System Using Deep Multiagent Reinforcement Learning

Yuan Gao, *Member, IEEE*, Junfeng Chen, Xi Chen, Chongyang Wang, Junjie Hu, *Member, IEEE*, Fuqin Deng, *Member, IEEE*, and Tin Lun Lam, *Senior Member, IEEE*

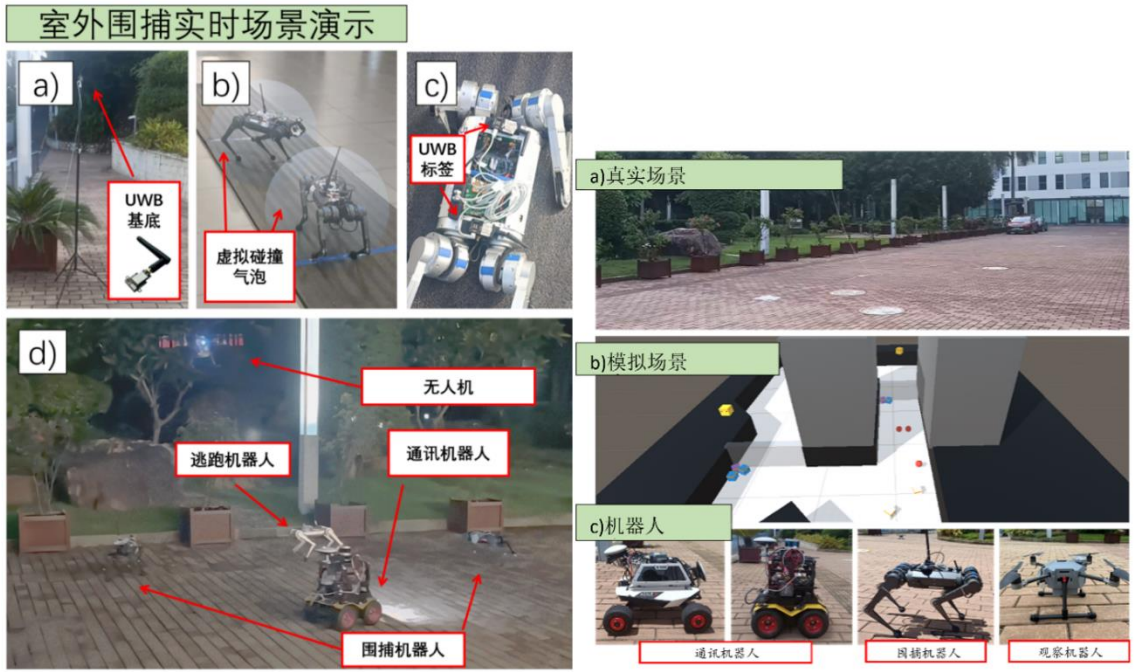
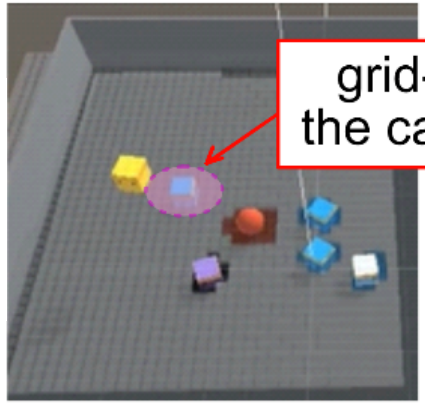
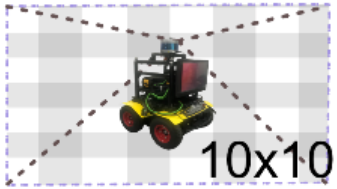


Fig. 3. Illustration of each kind of robot in the catching team and their corresponding primary type of perception in both simulation and the real world. The communication robot receives a small grid-sensor observation and a vector observation (e.g., locations and velocities). The catcher robots and observer robots receive, respectively, a smaller and larger grid-sensor as inputs. If the observer robots are present in the catching team, they send the location of the runner robots to the catcher robots.



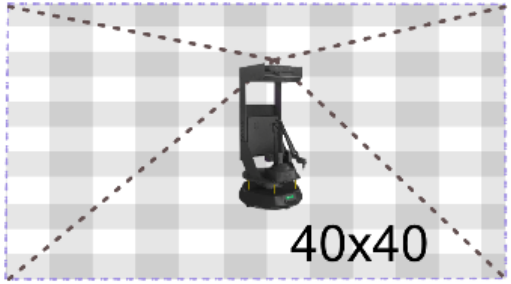
Examples of grid-sensor in simulation



+

Pose n_w ... | Pose n_t ... | Pose n_r ...

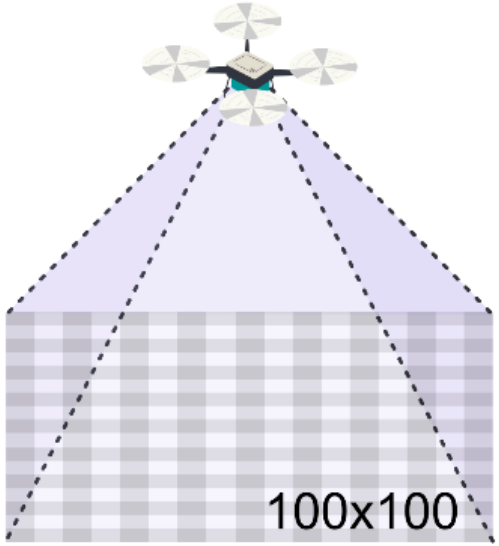
small grid-sensor observations and vector observations



+

Pose n_r ...

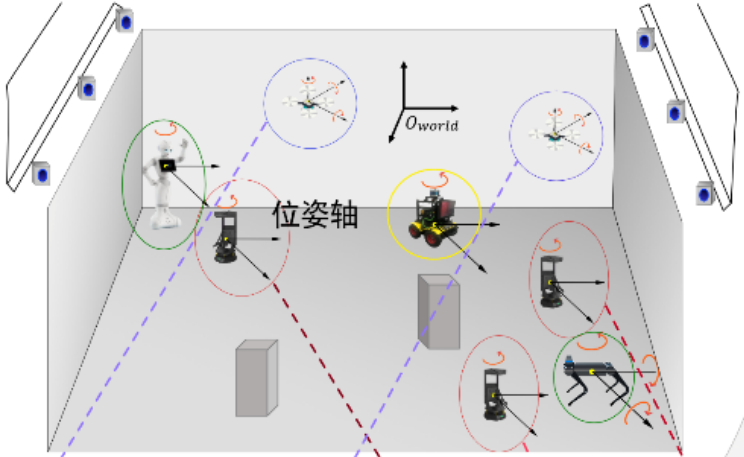
grid-sensor observations and vector observations

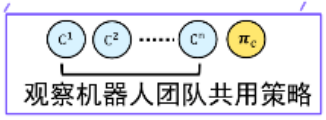


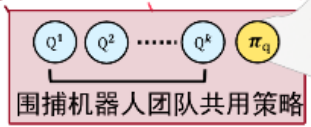
large grid-sensor observations

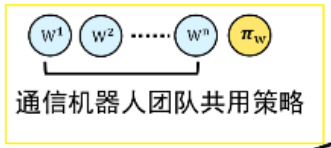
追逃

文章提出了一种基于多智能体强化学习的用于解决异构机器人团队合作博弈的框架，通过融合不对称自我博弈和课程学习技术来实现异构机器人团队之间的合作行为。这个框架使异构多机器人系统在真实世界约束条件下进行复杂捕捉变成可能。

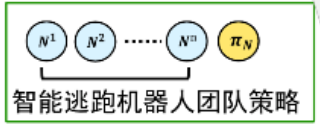


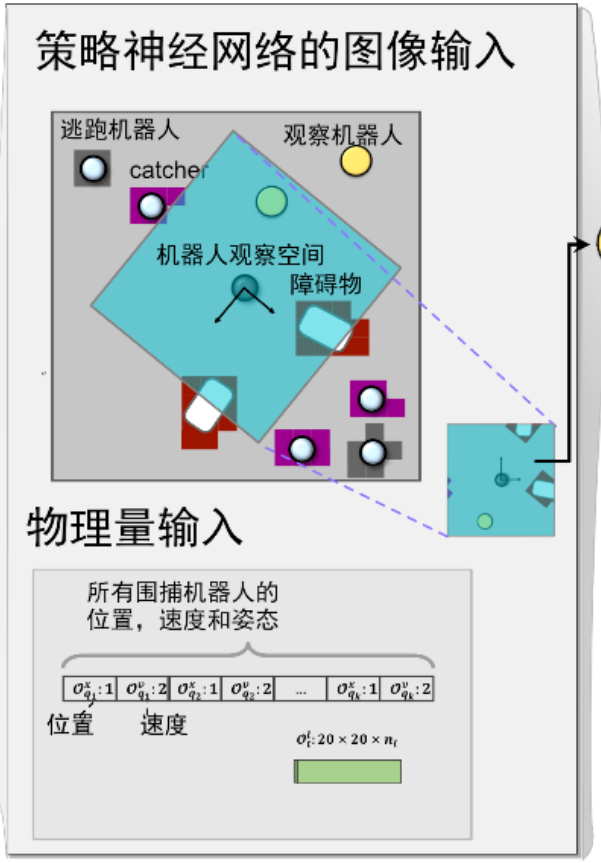






非对称博弈





团队奖励分配

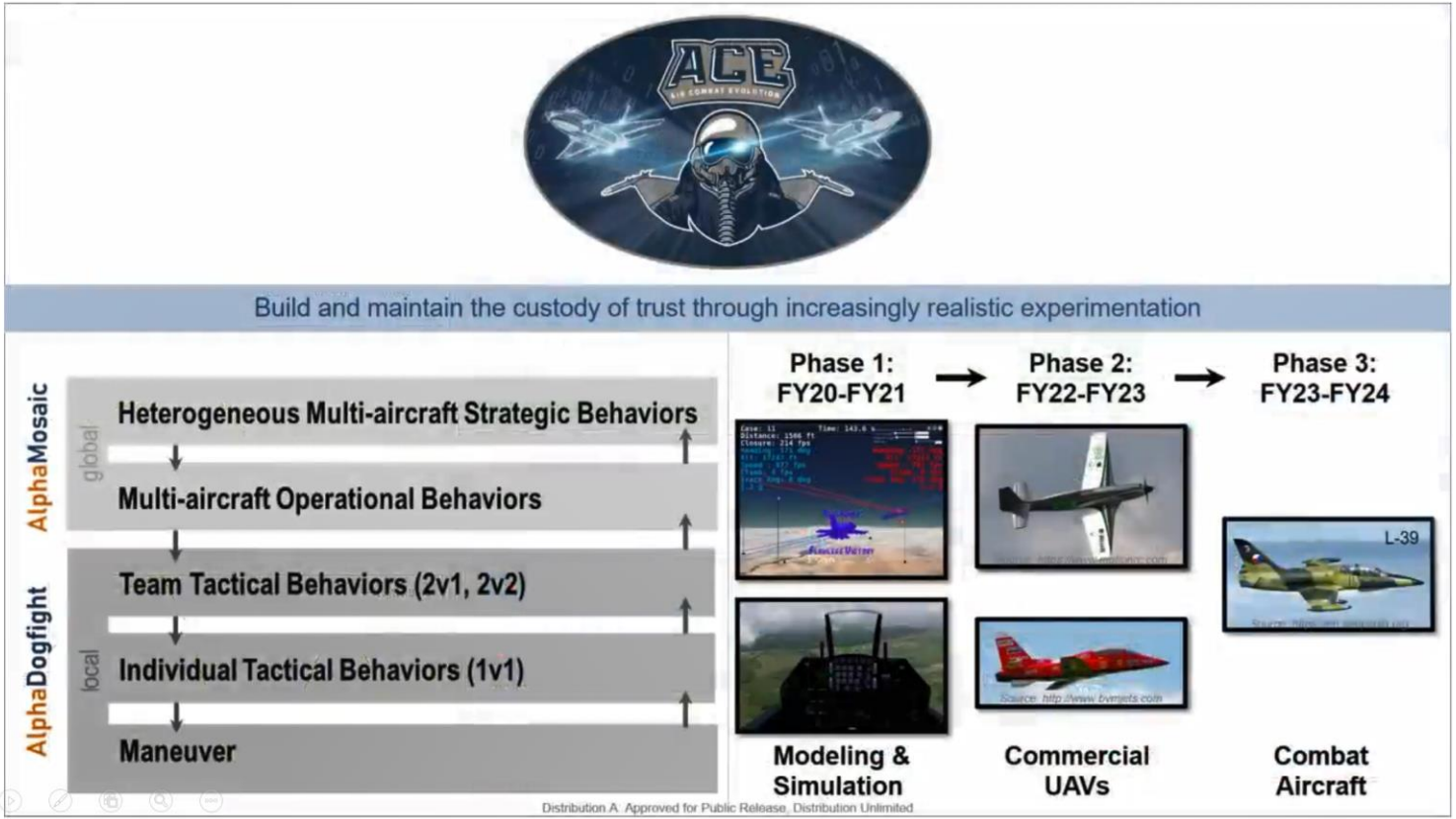
$$D^a = r(s, \mathbf{u}) - r(s, (\mathbf{u}^{-a}, c^a))$$

外界奖励

R_w, R_t, R_q

空战

DARPA ACE Program Makes Strides in Phase 1



美国人工智能空战发展路线

2016

阿尔法空战系统

特点

- 在模拟器中击败人类飞行员
- 遗传模糊树
- 基于演进式规则的推理系统

2019.3

Skyborg验证机

特点

- 人工智能空中对抗试验平台
- 原型机正逐步具备作战能力

2020.8

阿尔法狗斗

特点

- 近距自主空中格斗项目
- 深度强化学习
- 在仿真器中大比分战胜人类

2020.11

真机–智能机半仿真对抗

特点

- 技术模拟产生半仿真智能机
- 智能机通过增强现实显示在头盔中
- 配备战术AI系统

2023

ACE第三阶段真机验证

AI“飞行员”驾驶一架完整的喷气式飞机（L-39教练机）进行实际空战缠斗。

Flying in Simulator

Watch DARPA's AI vs. Human in Virtual F-16 Aerial Dogfight (FINALS)

谢谢大家