

Melting Pot 调研报告

"A Suite of Test Scenarios for MARL"

胡逸同, 2023/11/09

目录

1. 目录
2. 概述
 1. *Substrates & Scenarios*
 2. Evaluation Metrics
3. Contest @NeurIPS 23
 1. 任务
 2. Evaluation Metric
 3. Allelopathic Harvests
 1. 动作空间
 2. 策略 (直觉)
 4. Clean Up
 1. 动作空间
 2. 策略 (直觉)
 5. Territory: Rooms
 1. 动作空间

概述

Melting Pot 由 DeepMind 设计/开发，它：

- 关注 MARL 算法在**社会情境**（agents 彼此熟悉 or 陌生）做**互动任务**（**混合**合作/竞争/欺骗等）的**泛化**能力
- 集成 50+ 经典游戏（substrate），并为每个 substrate 设计了不同的 scenario（共计 256+）
- 为 MARL 算法提供了 **benchmark**

具体解释：

1. *Substrates & Scenarios*
2. Evaluation Metrics

Substrates & Scenarios

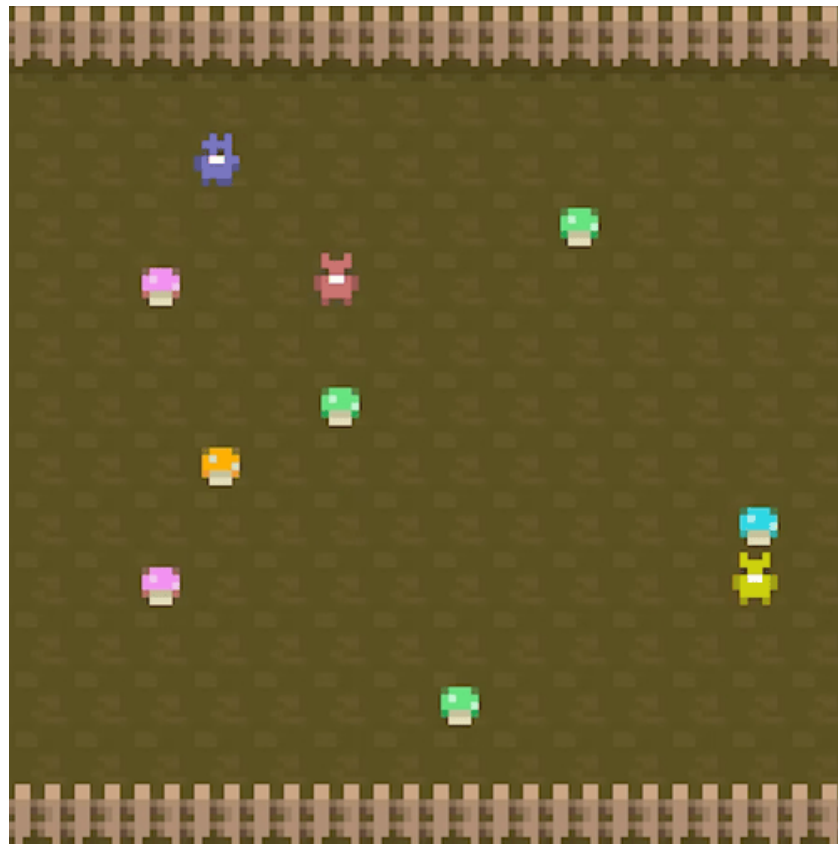
substrate 是世界的物理部分（空间布局、物体的位置/运动方式、物理规则等）

A substrate is an N -player **partially observable** general-sum Markov game (Leibo et al., 2021).

“*scenario* 是 *substrate* 参数化后的实例”:

A test scenario is a Markov game parameterized by a substrate factory F , a role configuration \mathbf{r} , the focal-population size $N \leq |\mathbf{r}|$, and a background population of held-out bots g_F .

- *background population* 是游戏中的 NPC，它们不参与训练，只用于评估
- *focal population* 是参与训练的 agents

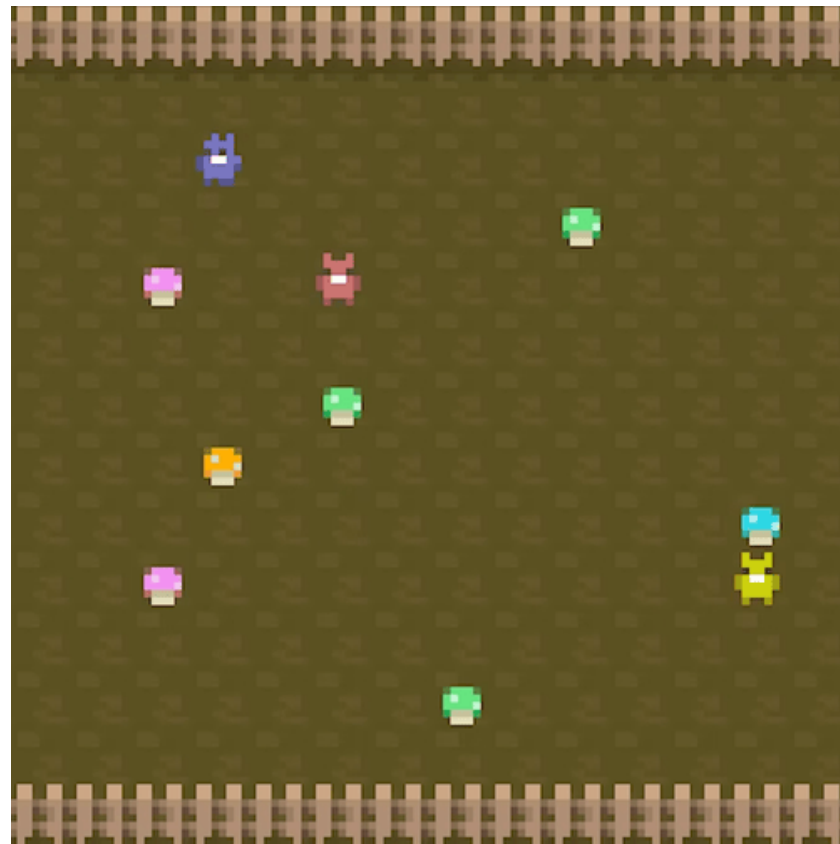


Visualized Substrates

Evaluation Metrics

- *focal_player_returns*
- *focal_per_capita_return*
- *background_player_returns*
- *background_per_capita_return*

详见论文



Visualized Substrates

Contest @NeurIPS 23

🕒 Round 1: 2 days left

NeurIPS 2023



Melting Pot Challenge

Multi-Agent Dynamics & Mixed-Motive Cooperation

🏆 \$10,000 Cash Prize Pool

+

💻 \$50,000 Compute Budget

By  Alcrowd &  Cooperative AI Foundation

👁️ 23.2k


👤 633

👥 117

🚀 532

❤️ 39

Share



任务

比赛只关注以下 4 个 *substrates* 和各自对应的 *scenarios*:

Substrate	Scenarios	Sum
allelopathic_harvest__open	allelopathic_harvest__open_0-2	3
clean_up	clean_up_2-8	7
prisoners_dilemma_in_the_matrix__arena	prisoners_dilemma_in_the_matrix__arena_0-5	6
territory__rooms	territory__rooms_0-3	4

Evaluation Metric

- 参赛者提交 4 组群体 (agents) , 分别对应 4 个 substrate。
- 对于每个 substrate, 在其 scenario 中评估 agents, 计算 *focal_per_capita_return*.
 - 对各 scenario 的得分做平均, 得到该 substrate 的得分。
- 为了在不同 scenario 间保持评估的一致性, 基于 MP2.0 的 baseline 范围, 对 *focal_per_capita_return* 做标准化:

$$\text{Normalized Focal Per Capita Return} = \frac{\text{Focal Per Capita Return} - \text{Min Baseline Score}}{\text{Max Baseline Score} - \text{Min Baseline Score}}$$

其中:

- Focal Per Capita Return 是参赛者提交的 agents 在特定 substrate 中的 *focal_per_capita_return*。
- Max/Min Baseline Score 是 baseline 在该 substrate 所有 scenario 中的最高/低得分。

Note: 参赛者的得分可以超出 [0-1] (可能比 baseline 更好/更差)

- 对 4 个 substrate 的得分做平均, 得到参赛者的最终得分 (排名) 。

Allelopathic Harvests

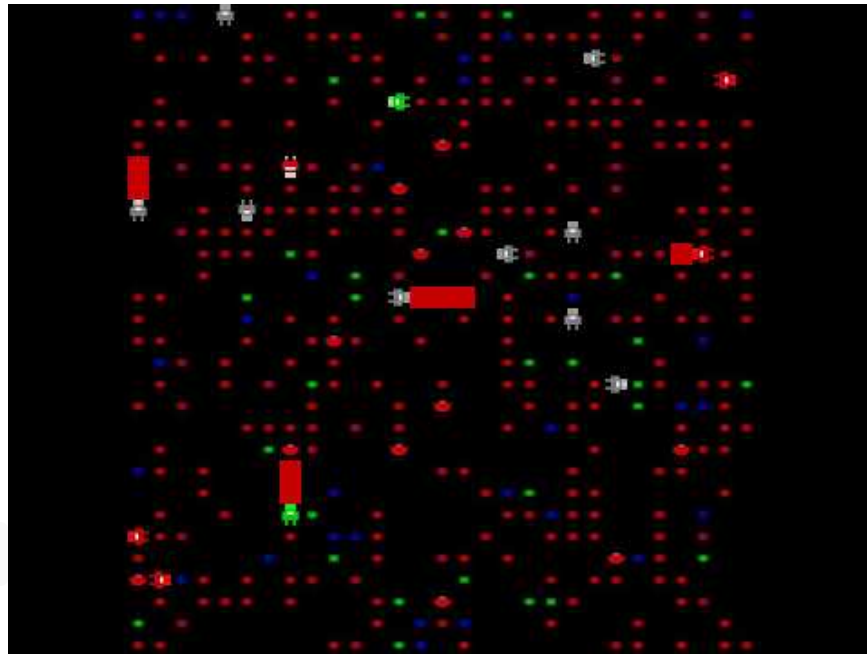
简介

1. 有 3 种浆果：红、绿、蓝，agent 可以种任何果
2. 某浆果占比越多，其生成速度越快，二者线性正相关
3. agent 可以吃任何果（浆果成熟后）获得奖励
4. agent 可以杀任何其他 agents

Rreward

对于每个 agent

Event	Rreward
吃红果	2
吃绿 or 蓝果	1
被杀	-10



动作空间

```
a = ['W', 'A', 'S', 'D', 'Q', 'E', '1', '2', '3', 'SPACE']
```

按键

动作

``W`` / ``A`` / ``S`` / ``D``

移动（碰到果自动吃）

``Q`` / ``E``

左/右转

``1`` / ``2`` / ``3``

种下红/绿/蓝果

``SPACE``

发射 zapper (AOE) , 第一次索敌（会标记上X）, 第二次击杀

``TAB``

换人

1. ``W`` / ``A`` / ``S`` / ``D`` & ``SPACE`` & ``1`` / ``2`` / ``3`` 可能会被其他 agents 阻挡
2. 被击杀的 agent 会在随机位置复活

策略 (直觉)

1. start-up problem
2. free-rider problem
3. 尽量避免击杀，扣分太多
4. 守着自己的果园，不要去别人的果园？

Clean Up

简介

1. 7 人游戏，目标是收集苹果
2. 果园中苹果的生长速度与附近河流的清洁度成反比
3. 河流以恒定速度积累污染，污染超过阈值后，苹果停止生长
4. agent 可以清理河流 (AOE)
5. agent 可以击杀其他 agents (AOE)，被击杀的 agent 50 步后复活

TODO

谁的 50 步？击杀者的 50 步？击杀者的 50 步后，被击杀者复活在哪里？
收苹果的 reward 是多少？



Reward (对于每个 agent)

Event	Rreward
收苹果	?

动作空间

```
a = ['W', 'A', 'S', 'D', 'Q', 'E', '1', '2']
```

按键

动作

`W` / `A` / `S` / `D`

移动（碰到果自动收）

`Q` / `E`

左/右转

`1`

一击必杀 agent (AOE)

`2`

清理河流 (AOE)

`TAB`

换人

策略（直觉）

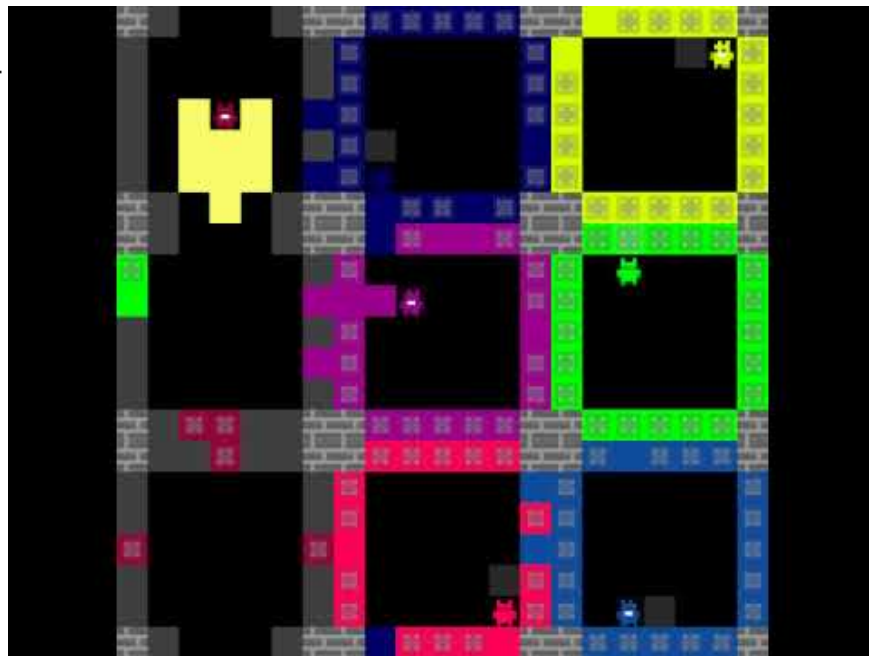
1. public good provision problem
2. 清理河流工作量很大，但是收益是共享的
3. 主要是合作，要合理分配清理和收果
4. 避免击杀：地方大，人多，击杀的收益不大

Territory: Rooms

1. 认领领地，碰到墙自动认领资源，100 timesteps 后资源激活，以每时间步 0.01 的比率随机向领取资源的玩家提供奖励
2. agent 可以拆墙 (AOE)，拆墙后墙消失，墙上的资源消失
3. agent 可以击杀其他 agents (AOE)，被击杀的 agent 永久消失

Reward (对于每个 agent)

Event	Rreward
资源被激活	每时间步 0.01 的比率随机向资源所属 agent 提供奖励
被击杀	清零



动作空间

```
a = ['W', 'A', 'S', 'D', 'Q', 'E', 'SPACE']
```

按键

动作

`W` / `A` / `S` / `D`

移动

`Q` / `E`

左/右转

`SPACE`

攻击 (AOE) , 类似 allenlopathic_harvest, 两步操作

`TAB`

换人

策略（直觉）

1. 一旦有人拆墙，就会引发连锁反应，大家都会拆墙
2. 拆墙后，资源被激活，但是资源是随机分配的，所以拆墙后，资源可能被其他人拿走

运行指南

1. 环境设置
2. 动手体验游戏
3. Training, Evaluation and Visualization

环境设置

```
conda activate mpc_main \  
&& cd ~/Projects/Melting-Pot-Contest-2023 \  
&& substrate=(pd_arena al_harvest clean_up territory_rooms) \  
&& path=(./results/torch/${substrate[0]}/PPO_meltingpot_a0cbb_00000_0_2023-10-31_14-20-46 \  
./results/torch/${substrate[1]}/PPO_meltingpot_b833f_00000_0_2023-10-31_14-50-03 \  
./results/torch/${substrate[2]}/PPO_meltingpot_086de_00000_0_2023-10-31_15-06-37 \  
./results/torch/${substrate[3]}/PPO_meltingpot_18d19_00000_0_2023-10-31_15-07-04) \  
&& n=0
```

n	Substrate
0	pd_arena
1	al_harvest
2	clean_up
3	territory_rooms

动手体验游戏

```
# allelopathic_harvest__open
```

```
python meltingpot/human_players/play_allelopathic_harvest.py
```

```
# clean_up
```

```
python meltingpot/human_players/play_clean_up.py
```

```
# prisoners_dilemma_in_the_matrix__arena
```

```
python meltingpot/human_players/play_anything_in_the_matrix.py --level_name=prisoners_dilemma_in_the_matrix__arena
```

```
# territory__rooms
```

```
python meltingpot/human_players/play_territory.py
```

Training, Evaluation and Visualization

1. Training (baseline)

```
python baselines/train/run_ray_train.py --exp=${substrate[$n]}  
# python baselines/train/run_ray_train.py --exp=${substrate[$n]} --as-test=True
```

Note: 默认使用远程 Ray 服务器

2. Evaluation

```
python baselines/evaluation/evaluate.py \  
--config_dir=${path[$n]} \  
--policies_dir=${path[$n]}/checkpoint_000001/policies
```

BUG: 无法创建场景视频 (未解决)

```
--create_videos=True --video_dir='./results/videos'  
# 报错:  
# OpenCV: FFMPEG: tag 0x30397076/'vp90' is not supported with codec id 167 and format 'webm / WebM'
```

3. Visualization

```
python baselines/train/render_models.py \  
--config_dir=${path[$n]} \  
--policies_dir=${path[$n]}/checkpoint_000001/policies
```

References

1. [AlCrowd | Melting Pot Contest \(MPC\)](#)
2. [GitHub | MPC Bassline](#)
3. [Github | MP 2.0](#)
4. [arXiv | MP 2.0 Tech. Report](#)

Thank you!

胡逸同, 2023/11/09