

Zero-Shot Coordination in Overcooked-AI

胡逸同, 2024/01/25

目录

1. Overcooked-AI

1. 简介
2. 历史遗留问题
3. 改进方向

2. Zero-Shot Coordination Baselines

1. On the Utility of Learning about Humans for Human-AI Coordination (HARL)
[^UtilityLearningHumans2019]
2. Fictitious Co-Play (FCP) [^strouseCollaboratingHumansHuman2021]
3. Trajectory Diversity (TrajeDi) [^lupuTrajectoryDiversityZeroShot2021]
4. Maximum Entropy PBT (MEP) [^zhaoMaximumEntropyPopulationBased2023]
5. Hidden-Utility Self-Play (HSP) [^yuLearningZeroShotCooperation]
6. PECAN [^louPECANLeveragingPolicy2023]
7. Cooperative Open-ended LEarning (COLE) [^CooperativeOpenendedLearning2023]

Overcooked-AI

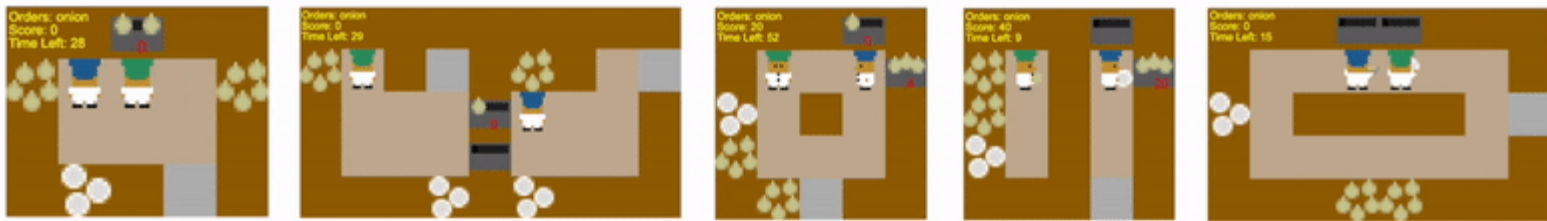
简介

Overcooked-AI 是由 UC Baerkeley CHAI 团队开发的 benchmark 环境，旨在通过 Overcooked 游戏，评估各算法在 **human-AI 完全合作** 任务中的性能。

在 Overcooked-AI 中，2 个玩家需要**合作完成** `取食材-移动食材-入锅-装盘-上菜` 一系列任务，获得团队得分。Agents 需要学习地图导航、物体交互和上菜，同时注意与伙伴的协调，属于 common-payoff game。

环境：

- 2 agents, agents pair = $[A_0, A_1]$, $A_i \in [AI, Human]$
- 5 种不同的布局，各有不同的地形和物体分布
- 可交互物体 = [洋葱，盘子，锅，台面，上菜区]，环境会无限生成洋葱和盘子



Agents:

- 动作空间 = [上、下、左、右移动, 啥也不干, 交互]

交互: 对个物体有不同的操作, 例如: 将洋葱入锅, 拿盘子盛锅里的汤, 将盛好汤的盘子放在上菜区, 或把洋葱/盘子暂存至台面

- 可完全观测环境 (MDP), 或泛化到可部分观测环境 (POMDP)

任务:

- 将 3 个洋葱放入锅中 - 煮 20 timesteps - 将汤装入盘子 - 将盘子放在上菜区
- **上菜才能得分**, 有时间限制
- 只有团队得分, 无个人得分

Press enter to begin!

Human (Green Hat) v.s. AI (Blue Hat) in Coordination Ring

布局:



不同的布局要求不同的协作策略，从左到右，分别是：

1. Cramped Room：提供低级协调挑战，因空间限制，代理很容易相撞。
2. Asymmetric Advantages：测试玩家是否可以选择发挥自身优势的高级策略。
3. Coordination Ring：玩家必须协调，才能在布局的左下角与右上角之间移动。
4. Forced Coordination：消除了碰撞协调问题，强迫玩家发展高级联合策略，因为单个玩家无法独自上菜。
5. Counter Circuit：涉及隐式的协调策略，洋葱经柜台传递至锅中，而不是绕道携带。

使用 Multi-Agent MDP 表达游戏过程 [1]:

A multi-agent MDP is defined by a tuple $\langle S, \alpha, \{A_{i \in \alpha}\}, \mathcal{T}, R \rangle$:

- S is a finite set of states, and $R : S \rightarrow \mathbb{R}$ is a real-valued reward function.
- α is a finite set of agents.
- A_i is the finite set of actions available to agent i .
- $\mathcal{T} : S \times A_1 \times \cdots \times A_n \times S \rightarrow [0, 1]$ is a transition function that determines the next state given all of the agents' actions.

-
1. Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., & Dragan, A. (2019). On the Utility of Learning about Humans for Human-AI Coordination. Advances in Neural Information Processing Systems, 32.
https://proceedings.neurips.cc/paper_files/paper/2019/hash/f5b1b89d98b7286673128a5fb112cb9a-Abstract.html 


历史遗留问题

早期的 Overcooked-AI 组件版本依赖复杂，且不向下兼容。

截至 2023 年，CHAI 团队已经将上述组件合并发布至 Overcooked-AI 仓库。与 ``neurips2019`` 版本相比，当前版本的套件有大量优化，包括 ``human_aware_rl`` 引入 Ray 作为分布式训练框架，``overcooked_ai_py`` 在游戏中加入新动作 ``煮食材``，``overcooked_demo`` 可一键更新 ``overcooked_ai_py`` 版本并在 Web 演示游戏，以及更丰富的文档和用例。

然而，目前所调查的相关工作均使用 ``neurips2019`` ^[1] 版本实现自己的算法，且不被当前版本的套件兼容。

因此，目前将以 ``neurips2019`` 版本为基础复现相关工作。

-
1. Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., & Dragan, A. (2019). On the Utility of Learning about Humans for Human-AI Coordination. Advances in Neural Information Processing Systems, 32.
https://proceedings.neurips.cc/paper_files/paper/2019/hash/f5b1b89d98b7286673128a5fb112cb9a-Abstract.html 



改进方向

后期，为方便 zero-shot coordination (ZSC) 研究者享受现代 benchmark env 的特性，可以尝试：

1. 将 `neurips2019` 版的模型转换为兼容当前版本的格式，或
2. 使用当前版本 Overcooked-AI 或 Melting Pot ^{[1] [2]} 复现相关工作

Melting Pot 是 DeepMind 团队提出的更现代的 MARL benchmark 环境，其同样实现了 Overcooked 游戏，可被视为 Overcooked-AI 的超集。Melting Pot 聚焦于社会情境下的多智能体互动，具有更丰富的特性，例如：更精细的环境设置（可调的观测视窗）、更多的互动任务（游戏）、更多的玩家数量，以及更合理的评估指标，有望成为 ZSC 研究的新标杆。

今后的有关 MARL 的工作可以考虑使用 Melting Pot 作为 simulator。

1. Agapiou, J. P., Vezhnevets, A. S., Duéñez-Guzmán, E. A., Matyas, J., Mao, Y., Sunehag, P., Köster, R., Madhushani, U., Kopparapu, K., Comanescu, R., Strouse, D. J., Johanson, M. B., Singh, S., Haas, J., Mordatch, I., Mobbs, D., & Leibo, J. Z. (2023). Melting Pot 2.0 (arXiv:2211.13746). arXiv. <https://doi.org/10.48550/arXiv.2211.13746> 
2. Hu Y. (2023). Melting Pot Research Report. <https://yitong-hu.metattri.com/melting-pot-contest-2023/> 

Zero-Shot Coordination Baselines

Zero-Shot Coordination: 与没见过的伙伴（人或 AI）协作
以下工作大部分使用 Overcooked-AI 评估算法性能

On the Utility of Learning about Humans for Human-AI Coordination (HARL) [1]

2019 年, Self-play (SP) 和 population-based training (PBT) 是两种常用的 MARL 训练策略, 用于训练与人类协作的 agents。

本文认为, SP 和 PBT agent 将假设其伙伴是最优的或者与自己相似的 (而人类的行为不是最优的且难以预测), 这会导致 agent 更适合跟自身而非人类协作, 将人类数据或模型纳入训练过程将改进 human-AI coordination 的性能。因此, 本文设计了 Overcooked-AI 环境, 并提出了:

- Behavior Cloning model (BC)、Proxy human model H_{Proxy}

二者都是使用人类数据训练的动作分类 (预测) 器, BC 是参与训练 agent 的伙伴; 而 H_{Proxy} 作为 ground truth, 用于评估 agent 的性能, 二者关系类似于 训练集 和 测试集

- 2 类与人类协作的 agent 模型
 - 不使用人类数据: Self-Play (SP)、Population-Based Training (PBT)、规划方法
 - 使用人类数据: PPO with human model PPO_{BC} 、规划方法

-
1. Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., & Dragan, A. (2019). On the Utility of Learning about Humans for Human-AI Coordination. Advances in Neural Information Processing Systems, 32.

Method

Self-Play: paly with self in each iteration, using PPO.

Population-Based Training (PBT) : play with n agents in each iteration, using PPO

- 种群规模 $n = 3$ (本文中) , 每个 agent 与 SP agent 结构相同, 只是伙伴从自己变成了不同的 agents
- PBT 算法可简述为: 初始化 n agents, 两两配对训练, 最差的 agent 变异 (进化) , 细节如下:

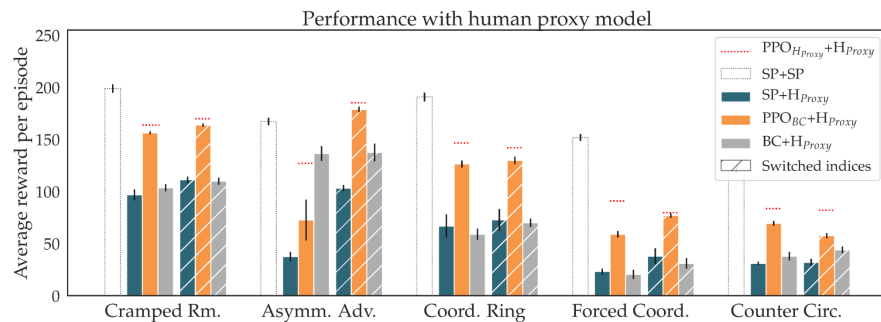
```
while not converged:
    for i in range(n):
        for j in range(i+1, n):
            train(agent[i]) # training agent_i using PPO and agent_j is embedded into the environment
        performance[i] = eval(agent[i])
    worst_agent = get_worst(performance)
    agent[worst_agent] = mutate(agent[worst_agent])
```

PPO_{BC}: play with BC in each iteration, using PPO

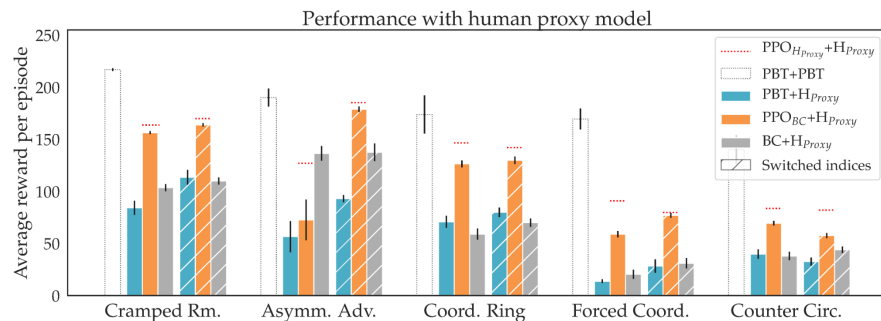
1. 使用人类游戏数据训练行为克隆 (behavior cloning) 模型 BC
 - 分类任务
 - 使用 cross-entropy loss
2. BC 作为环境的一部分, 使用 PPO 作为策略梯度算法, 训练agent

Evaluation

AI- H_{Proxy} Play



(a) Comparison with agents trained in self-play.



(b) Comparison with agents trained via PBT.

AI-Human Play

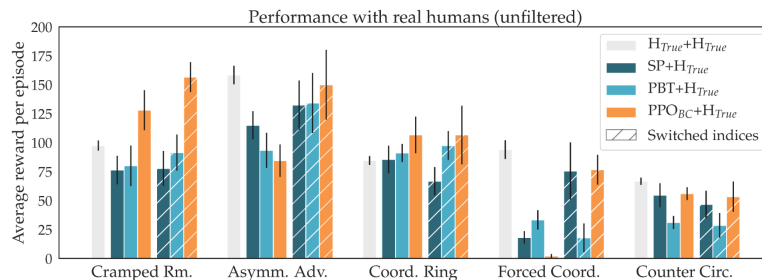


Figure 12: The results are mostly similar to those in Figure 6, with the exception of larger standard errors introduced by the non-cooperative trajectories. The reported standard errors are across the human participants for each agent type.

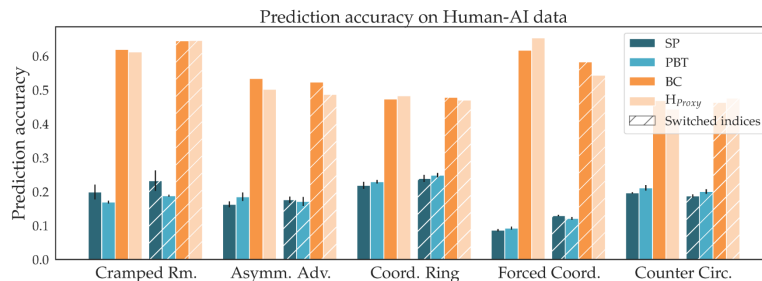


Figure 13: Accuracy of various models when used to predict human behavior in all of the human-AI trajectories. The standard errors for DRL are across the 5 training seeds, while for the human models we only use 1 seed. For each seed, we perform 100 evaluation runs.


Fictitious Co-Play (FCP) [1]

Motivation:

1. Self Play (SP) 或 Population Play (PP) , 产生的 agent 过度适应他们的训练伙伴, 难以推广到人类
2. HARL 提出的 PPO_{BC} (本文称为 BCP) 涉及到收集大量人类数据, 繁重而昂贵
3. 与 novel partner 的合作需要处理对称问题, 比如: 二人相遇时的避让策略, 同左 or 同右?
4. 与人类合作需要迅速理解并适应他们的个人优势、劣势和偏好
5. 好的 agent 应该能够和各水平的伙伴合作, 而不是只能和最优伙伴合作

Contribution:

- 提出 Fictitious Co-Play (FCP) 来训练能够与人类进行 zero-shot 协调的 agent
- 证明 FCP agent 在与各种 agents 进行 zero-shot 协调时, 比 SP、PP 和 BCP 的表现更好
- 证明 FCP 在任务得分和人类偏好方面都明显优于 BCP 的 SOTA

-
1. Strouse, D., McKee, K., Botvinick, M., Hughes, E., & Everett, R. (2021). Collaborating with Humans without Human Data. Advances in Neural Information Processing Systems, 34, 14502–14515.
<https://proceedings.neurips.cc/paper/2021/hash/797134c3e42371bb4979a462eb2f042a-Abstract.html> 

Method

Stage 1: 独立训练 n 个 SP agents, 保存各阶段的 checkpoint 至 pool (代表不同水平)

Stage 2: 与 pool 中 agents 配对训练 FCP agent

为了推广 FCP, 使其能够接受视觉 observation, 本文未采用 PPO, 而是设计了强化学习算法: 使用 V-MPO 算法, 结合 ResNet 和 LSTM 构建所有 agent (stage 1 & 2), 在分布式环境并行训练。

“For our reinforcement learning agents, we use the V-MPO [65] algorithm along with a ResNet [26] plus LSTM [29] architecture which we found led to optimal behavior across all layouts. Agents are trained using a distributed set of environments running in parallel [17], each sampling two agents from the training population to play together every episode.” (Strouse 等, 2021, p. 4) (pdf)

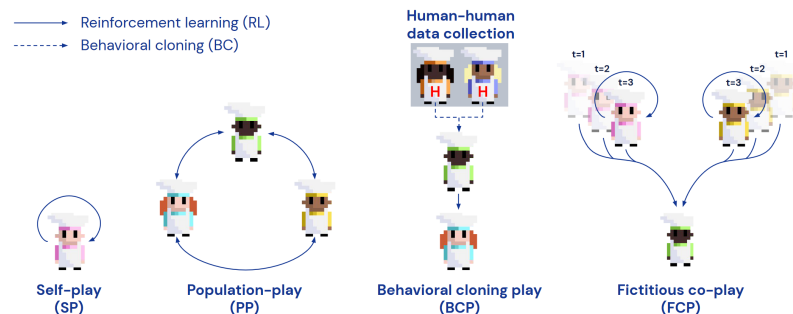
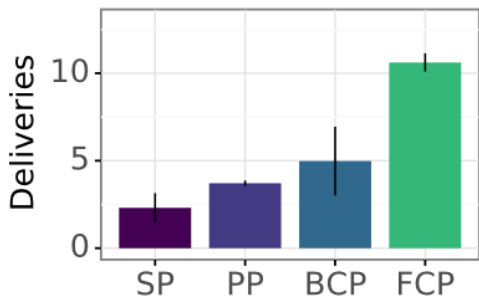


Figure 2: The four agent training methods we evaluate in this work. **Self-play (SP)** where an agent learns with itself, **population-play (PP)** where a population of agents are co-trained together, and **behavioral cloning play (BCP)** where data from human games is used to create a behaviorally cloned agent with which an RL agent is then trained. In our method, **Fictitious Co-Play (FCP)**, N self-play agents are trained independently and checkpointed throughout training. An agent is then trained to best respond to the entire population of SP agents and their checkpoints.

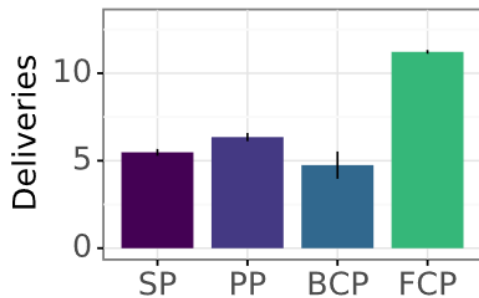
Evaluation

本文使用 3 类 agent, 与 FCP 和 baselines 配对玩游戏, 比较上菜次数 (Deliveries) :

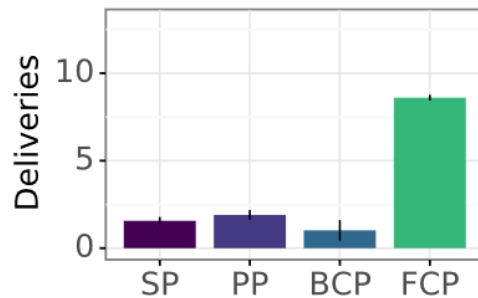
- Proxy human H_{Proxy}
- SP agent (as skillfull partner)
- 随机初始化的策略 agent (as low-skill partner)



(a) With H_{proxy} .



(b) With diverse SP agents.



(c) With random agents.

Figure 5: **Agent-agent collaborative evaluation:** Performance of each agent when partnered with each of the held-out populations (Section 4.1) in episodes of length $T = 540$. Importantly, FCP scores higher than all baselines with a variety of test partners. Error bars represent standard deviation over five random training seeds. Plots aggregate data across kitchen layouts; results calculated by individual layout can be found in Appendix C.2.

消融实验

- FCP: pool 中 agents 结构相同, seed 不同, Stage 2 使用过去 checkpoints.
- FCP_{-T} : 相比于 FCP, 不使用过去 checkpoints (未收敛的 agents), 用于测试过程中生成的 checkpoints 的重要性.
- FCP_{+A} : 相比于 FCP, agents 结构不同, 用于测试不同的结构会不会带来更好的多样性.
- $\text{FCP}_{-T,+A}$: 相比于 FCP_{+A} , 不使用过去 checkpoints, 用于测试不同结构能否替代过程中生成的 checkpoints.

Partner	FCP	FCP_{-T}	FCP_{+A}	$\text{FCP}_{-T,+A}$
H_{proxy}	10.6 ± 0.5	4.7 ± 0.4	9.9 ± 0.6	7.0 ± 0.8
Diverse SP	11.2 ± 0.1	6.9 ± 0.1	11.1 ± 0.4	8.6 ± 0.4
Random	8.6 ± 0.2	1.0 ± 0.1	8.4 ± 0.4	3.2 ± 0.5

Table 1: **Ablation results:** Performance of each variation of FCP – training with past partner checkpoints (T for time) and adding partner variation in architecture (A). Scores are mean deliveries with standard deviation over 5 random seeds. Notably, we find that the inclusion of past checkpoints is essential for strong performance ($\text{FCP} > \text{FCP}_{-T}$), and additionally including architectural variation does not improve performance ($\text{FCP} \approx \text{FCP}_{+A}$). However, architectural variation is better than no variation, improving performance when past checkpoints are not available ($\text{FCP}_{-T,+A} > \text{FCP}_{-T}$).


然而：

- FCP 不仅耗费时间，而且容易出现研究者的偏见，可能会对创建的 agent 的行为产生负面影响。
- 对于更加复杂的游戏，FCP 可能需要更大的 pool，这可能是不切实际的。

Trajectory Diversity (TrajeDi) [1]

TBD

MEP 传承了 TrajeDi 的思想，并达到新 SOTA。

-
1. Lupu, A., Cui, B., Hu, H., & Foerster, J. (2021). Trajectory Diversity for Zero-Shot Coordination. Proceedings of the 38th International Conference on Machine Learning, 7204–7213.
<https://proceedings.mlr.press/v139/lupu21a.html> 

Maximum Entropy PBT (MEP) [1]


TL;DR

竞争环境下，SP 和 PBT 效果较好，但在与人类合作的环境下，二者会训练出过于 specific 的策略。

一种解决思路是，引入人类数据辅助训练，但数据收集成本较高；

另一种思路是提高参与训练的 agents 的多样性：

- **diverse set of policies**: 例如 TrajeDi 优化 trajectory 间的 JS 散度从而达到 diverse 的目标，FCP 则使用随机种子或不同的 checkpoints；
- **domain randomization**: some features of the environment are changed randomly during training to make the policy robust to that feature, 本文的方法可被视为 domain randomization。同时，本文采用最大熵强化学习（MERL），相比于一般的强化学习，MERL 则需要最大化 return + 熵，这样会使得策略更具有**探索性**并且具有更强的**鲁棒性**。

-
1. Zhao, R., Song, J., Yuan, Y., Hu, H., Gao, Y., Wu, Y., Sun, Z., & Yang, W. (2023). Maximum Entropy Population-Based Training for Zero-Shot Human-AI Coordination. Proceedings of the AAAI Conference on Artificial Intelligence, 37, 6145–6153. <https://doi.org/10.1609/aaai.v37i5.25758> 

Method

与 FCP 类似, MEP 也是两阶段法: 首先训练一个 maximum entropy population, 然后通过 population 训练一个 robust agent.

本文借鉴最大熵强化学习的思想修改了训练的目标函数, 涉及两个概念: **Population Diversity & Entropy**:

Population Diversity: 首先需要 population 中 agents 自身的策略更有探索性, 同时也需要两两 agents 的策略差异更大.

$$\text{PD} \left(\left\{ \pi^{(1)}, \pi^{(2)}, \dots, \pi^{(n)} \right\}, s_t \right) := \frac{1}{n} \sum_{i=1}^n \mathcal{H} \left(\pi^{(i)} (\cdot | s_t) \right) + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n D_{\text{KL}} \left(\pi^{(i)} (\cdot | s_t), \pi^{(j)} (\cdot | s_t) \right)$$

where KL-divergence (D_{KL}) and entropy (\mathcal{H}) are defined as follows:

$$D_{\text{KL}} \left(\pi^{(i)} (\cdot | s_t), \pi^{(j)} (\cdot | s_t) \right) = \sum_{a \in \mathcal{A}} \pi^{(i)} (a_t | s_t) \log \frac{\pi^{(i)} (a_t | s_t)}{\pi^{(j)} (a_t | s_t)}$$

$$\mathcal{H} \left(\pi^{(i)} (\cdot | s_t) \right) = - \sum_{a \in \mathcal{A}} \pi^{(i)} (a_t | s_t) \log \pi^{(i)} (a_t | s_t)$$

Population Entropy: 因为 PD 计算复杂度过高, 并且 KL 散度是 unbounded 的, 可能会有收敛性的问题, 因此本文提出了 PE (population mean policy 的熵), 其具有线性复杂度并且是 bounded 的, 作为 PD 的 surrogate loss。文中也证明了 PE 是 PD 的 lower bound, 因此可以作为 surrogate loss。

$$\text{PE} \left(\left\{ \pi^{(1)}, \pi^{(2)}, \dots, \pi^{(n)} \right\}, s_t \right) := \mathcal{H}(\bar{\pi}(\cdot | s_t)), \text{ where } \bar{\pi}(a_t | s_t) := \frac{1}{n} \sum_{i=1}^n \pi^{(i)}(a_t | s_t)$$

为了训练出能 cooperate well 又 mutually distinct 的 strategy, 本文在目标函数中引入 PE 分量, 同时也引入了 hyperparameter α 来控制 PE 的权重, 作为 **MEP training objective**:

$$J(\bar{\pi}) = \sum_t \mathbb{E}_{(s_t, a_t) \sim \bar{\pi}} [R(s_t, a_t) + \alpha \mathcal{H}(\bar{\pi}(\cdot | s_t))]$$

Stage 1: train a maximum entropy population

1. 随机从 population 中采一个 agent
2. 然后优化该 agent 的策略
3. 重复步骤 1 - 2, 直到 $J(\bar{\pi})$ 收敛。

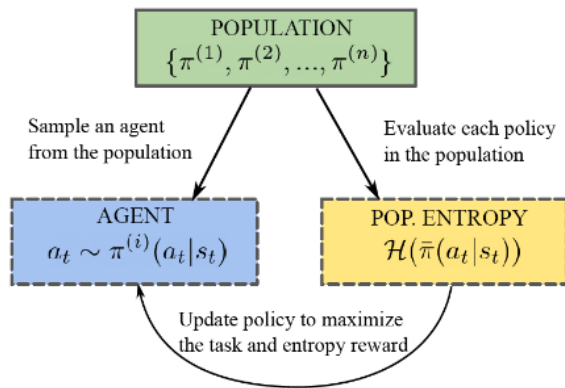


Figure 1: Maximum Entropy Population: We train each agent in the population to maximize its task reward as well as the population entropy reward to attain a maximum entropy population.

Algorithm 1: Maximum Entropy Population

while not converged do

Sample agent from population:

$$\pi^{(i)} \sim \{\pi^{(1)}, \pi^{(2)}, \dots, \pi^{(n)}\}$$

for $t \leftarrow 1$ **to** *steps_per_episode* **do**

Sample action $a_t \sim \pi^{(i)}(a_t | s_t)$.

Step environment $s_{t+1} \sim p(s_{t+1} | s_t, a_t)$.

Calculate the population entropy reward and combine it with the task reward:

$$r = r(s_t, a_t) - \alpha \log(\bar{\pi}(a_t | s_t))$$

Update policy $\pi^{(i)}$ **to maximize** $\mathbb{E}_\tau [r]$.

$r(s_t, a_t)$ 的获取是由采得的 agent 以及他的 copy 作为 partner 得到的, 相当于 SP

Stage 2: Training a robust agent (MEP Agent) paired with MEPooulation

本文没有直接对 MEpopulation 做 uniformly sample 来获得伙伴 agent 与 MEP agent 配对训练，而是使用了 learning progress-based prioritized sampling (LPPS) 来选择伙伴。LPPS 会选择 learning progress 最大的伙伴，这样可以使得 MEP agent 更具有探索性。

对于具体的 LPPS 方法，本文未采用 maximize average（最大化对 population 中所有 partner 的表现的平均值），因为 MEP agent 可能会学到与最容易合作的伙伴合作的策略，而放弃了难以合作的。因本文用 ranked-based 优先级采样让 MEP agent 优先跟难以合作的伙伴配对训练：

$$p(\pi^{(i)}) = \frac{\text{rank} \left(1/\mathbb{E}_\tau \left[\sum_t R(s_t, a_t^{(A)}, a_t^{(i)}) \right] \right)^\beta}{\sum_{j=1}^n \text{rank} \left(1/\mathbb{E}_\tau \left[\sum_t R(s_t, a_t^{(A)}, a_t^{(j)}) \right] \right)^\beta}$$

优先级采样是 smooth approximation of maximize minimum（极端情况下只和最难合作的进行训练就是 maximize minimum 了），当 population 足够多时，会有 partner agent 的策略与人类策略 ϵ -close，文中也证明了 human-ai coordination 的一些下界的性质。

Evaluation

AI- H_{Proxy} Play

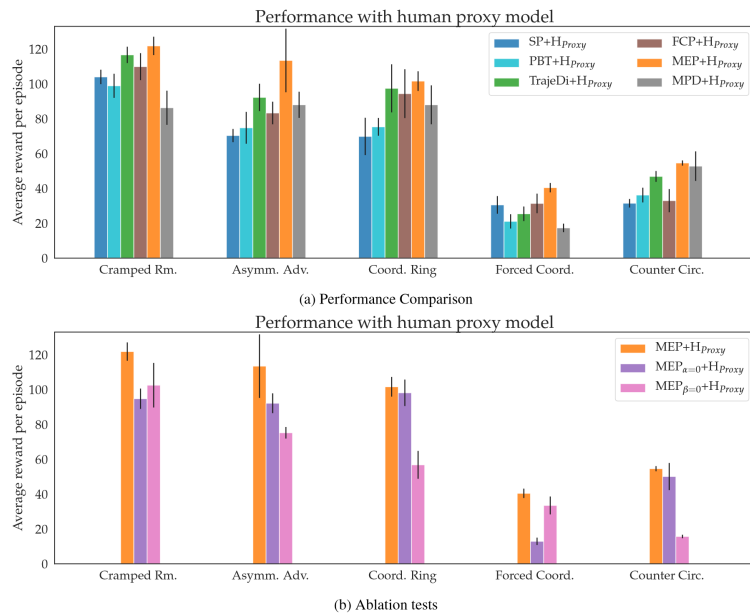


Figure 4: Performance comparison and ablation test: Average episode rewards over 400 timestep (1 min) trajectories for different methods, with standard error over 5 different random seeds, paired with the proxy human H_{Proxy} . Figure (a) shows the performance comparison among MEP and other methods including SP, PBT, TrajeDi, FCP, and MPD. Figure (b) shows the ablation tests, where we use $MEP_{\alpha=0}$ to denote MEP without PE reward and use $MEP_{\beta=0}$ to denote MEP without prioritized sampling. For more detailed experimental results, please refer to the figures in Appendix E.

Hidden-Utility Self-Play (HSP) [\[1\]](#)

TBD

1. Yu, C., Gao, J., Liu, W., Xu, B., Tang, H., Yang, J., Wang, Y., & Wu, Y. (n.d.). Learning Zero-Shot Cooperation with Humans, Assuming Humans Are Biased. [\[2\]](#)

PECAN [1]

Policy Ensemble Context-Aware zero-shot human-AI coordination

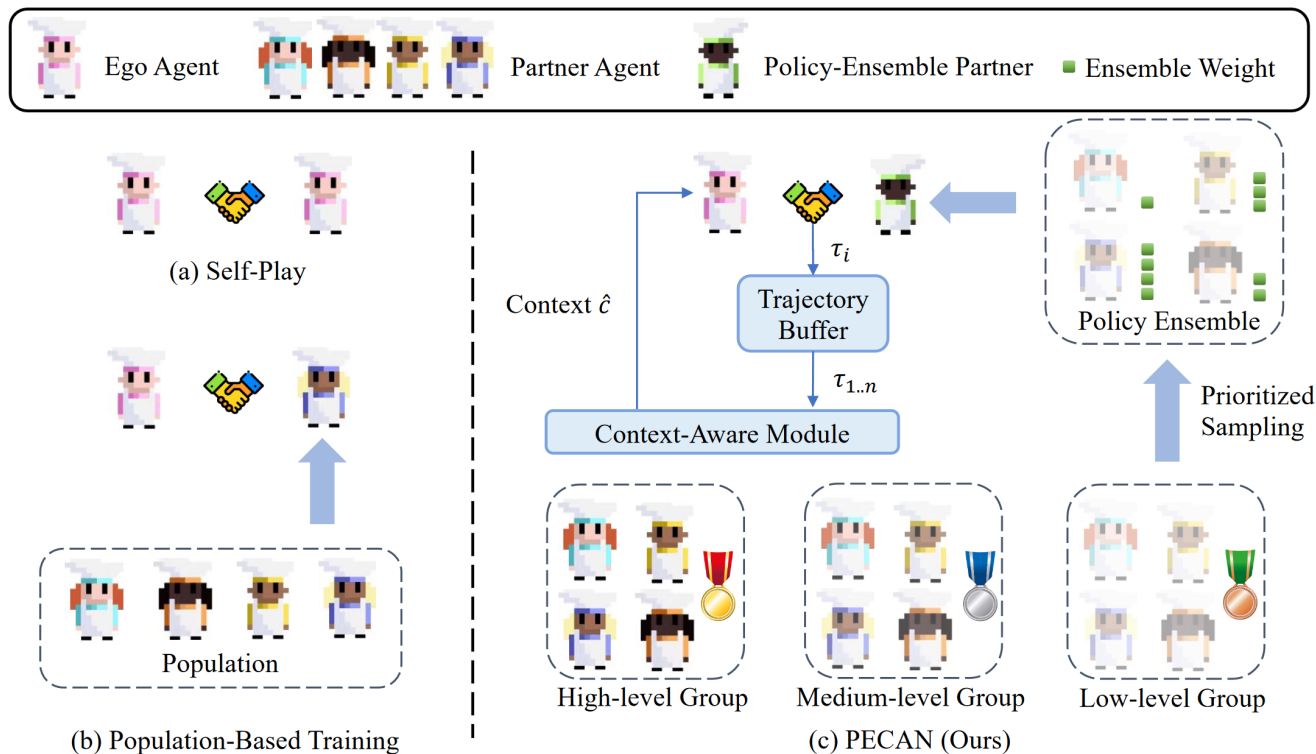


Figure 1: (a) Self-play training (SP). The ego agent is trained with a copy of itself. **(b) Population-based training (PBT).** The ego agent is trained with a population of partners. A partner is sampled at each iteration to cooperate with the ego agent. **(c)**

Cooperative Open-ended LEarning (COLE) [1]

Method

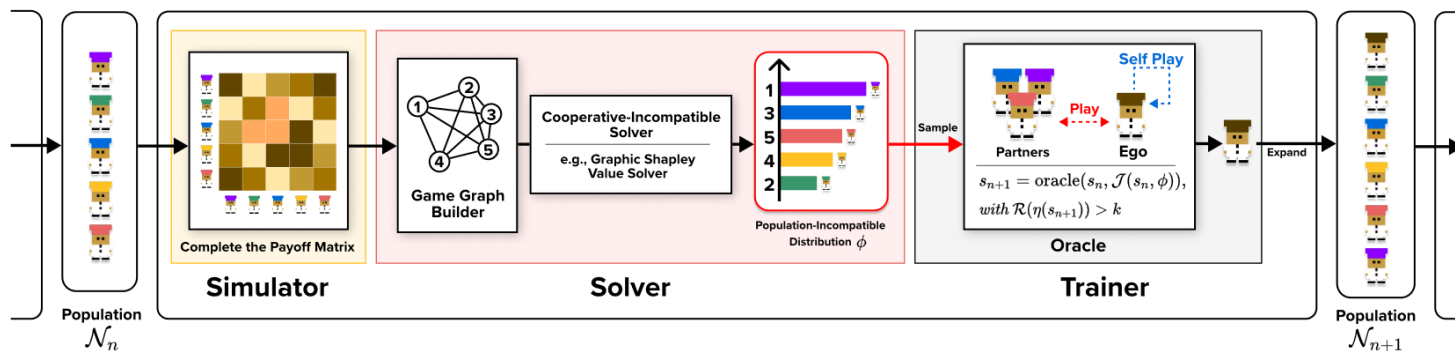


Figure 3. An overview of one generation in COLE framework: The solver derives the cooperative incompatible distribution ϕ using a cooperative incompatibility solver, which can be any algorithm that evaluates cooperative contribution. The trainer then approximates the relaxed best response by optimizing individual and cooperative compatible objectives. The oracle’s training data is generated using partners selected based on the cooperative incompatibility distribution and the agent’s strategy. Finally, the approximated strategy s_{n+1} is added to the population, and the next generation begins.

1. Li, Y., Zhang, S., Sun, J., Du, Y., Wen, Y., Wang, X., & Pan, W. (2023). Cooperative Open-ended Learning Framework for Zero-Shot Coordination. Proceedings of the 40th International Conference on Machine Learning, 20470–20484. <https://proceedings.mlr.press/v202/li23au.html>

Evaluation

AI- H_{Proxy} Play

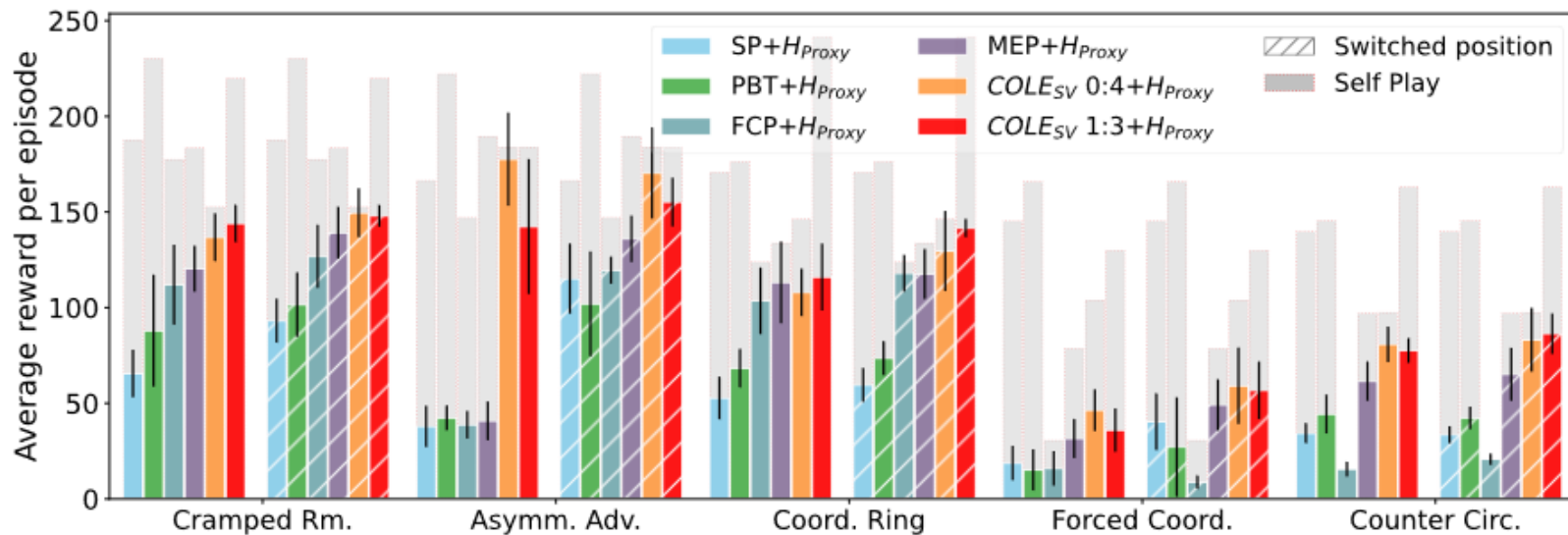


Figure 5. Performance with middle-level partners. The performance of COLE_{SV} with middle-level partners is presented in terms of mean episode rewards over 400 timesteps trajectories for differ-

AI-AI Play

Table 1. Performance with expert partners. Mean episode rewards over 1 min trajectories for baselines and COLE_{SV} with ratio 0:4, 1:3. Each column represents a different expert group, in which the result is the mean reward for each model playing with all others.

LAYOUT	RATIO	BASELINES				COLEs
		SP	PBT	FCP	MEP	
CRAMPED RM.	0:4	153.00	198.50	199.83	178.83	169.76
	1:3	165.67	209.83	207.17	196.83	212.80
ASYMM.ADV.	0:4	108.17	164.83	175.50	179.83	182.80
	1:3	108.17	161.50	172.17	179.83	178.80
COORD. RING	0:4	132.00	106.83	142.67	130.67	118.08
	1:3	133.33	158.83	144.00	124.67	166.32
FORCED COORD.	0:4	58.33	61.33	50.50	79.33	46.40
	1:3	61.50	70.33	62.33	38.00	86.40
COUNTER CIRC.	0:4	44.17	48.33	60.33	21.33	90.72
	1:3	65.67	64.00	46.50	76.67	105.84

Thank you!

胡逸同, 2024/01/25