Maximum Entropy Population-Based Training for Zero-Shot Human-AI Coordination

Rui Zhao[1*], **Jinming Song**[1], **Yufeng Yuan**[1],
Haifeng Hu[1], **Yang Gao**[2], **Yi Wu**[2], **Zhongqian Sun**[1], **Wei Yang**[1]

[1] Tencent AI Lab
[2] Tsinghua University

Abstract

We study the problem of training a Reinforcement Learning (RL) agent that is collaborative with humans without using human data. Although such agents can be obtained through self-play training, they can suffer significantly from the distributional shift when paired with unencountered partners, such as humans. In this paper, we propose Maximum Entropy Population-based training (MEP) to mitigate such distributional shift. In MEP, agents in the population are trained with our derived Population Entropy bonus to promote the pairwise diversity between agents and the individual diversity of agents themselves. After obtaining this diversified population, a common best agent is trained by paring with agents in this population via prioritized sampling, where the prioritization is dynamically adjusted based on the training progress. We demonstrate the effectiveness of our method MEP, with comparison to Self-Play PPO (SP), Population-Based Training (PBT), Trajectory Diversity (TrajeDi), and Fictitious Co-Play (FCP) in both matrix game and Overcooked game environments, with partners being human proxy models and real humans. A supplementary video showing experimental results is available at https://youtu.be/Xh-FKD0AAKE.

Introduction

Deep Reinforcement Learning (RL) has gained many successes against humans in competitive games, such as Go (Silver et al. 2017), Dota (OpenAI 2019), and StarCraft (Vinyals et al. 2019). However, it remains a challenge to build AI agents that can coordinate and collaborate with humans that the agents have not encountered during training (Kleiman-Weiner et al. 2016; Lerer and Peysakhovich 2017; Carroll et al. 2019; Shum et al. 2019; Hu et al. 2020; Knott et al. 2021). This challenging problem, namely zero-shot human-AI coordination, is particularly important for real-world applications, such as cooperative games (Carroll et al. 2019), communicative agents (Foerster et al. 2016), self-driving vehicles (Resnick et al. 2018), and assistant robots (Akkaya et al. 2019), because it removes the onerous and expensive step of involving human or human data in AI training. Thus, studying this problem could potentially make our ultimate goal of building AI systems that can assist humans and augment our capabilities (Engelbart 1962; Carter and Nielsen 2017) more achievable.

An efficient scheme for training AI agents in collaborative or competitive settings is through self-play reinforcement learning (Tesauro 1994; Silver et al. 2017). Due to its training paradigm, self-play-trained agents are very specialized since they only encounter their own policies during training and assume their partners will behave in a particular way. Therefore, those agents can suffer significantly from distributional shift when paired with humans. For example, in the Overcooked game, the self-play-trained agents only use a specific pot and ignore the other pots while humans use all pots. As a consequence, the AI agent ends up waiting unproductively for the human to deliver a soup from the specific pot, even though the human has instead decided to fill up the other pots (Carroll et al. 2019).

In this paper, we propose a robust and efficient approach *Maximum Entropy Population-based training* (MEP), to train agents for zero-shot human-AI coordination based on the advances in maximum entropy RL (Haarnoja et al. 2018b), diversity (Eysenbach et al. 2019), and Multi-agent RL (Lowe et al. 2017; Foerster et al. 2018). To encourage the diversity and explorability of policies of the individual agent in the population, we utilize the maximum entropy objectives (Ziebart et al. 2008; Toussaint 2009; Ziebart 2010; Rawlik, Toussaint, and Vijayakumar 2013; Fox, Pakman, and Tishby 2015; Haarnoja et al. 2017, 2018b; Zhao, Sun, and Tresp 2019) for the individual policies. To acquire diverse and distinguishable behaviors (Eysenbach et al. 2019) between agents in the population, we further utilize the average Kullback–Leibler (KL) divergence between all agent pairs in the population to promote pairwise diversity. We define this combination of individual diversity and pairwise diversity as *Population Diversity* (PD) and derive a safe and computationally efficient surrogate objective *Population Entropy* (PE), which is the lower bound of the original PD objective with linear runtime complexity. Analogous to maximum entropy RL training, each agent in the population is rewarded to maximize the centralized population entropy. With this diverse population, we train a best response agent by pairing it with the agents sampled from this population with a prioritization scheme based on the difficulty to collaborate with (Schaul et al. 2016; Vinyals et al. 2019; Han et al. 2020). By doing so, this newly trained AI agent encounters

[*] Correspondence to: Rui Zhao {rui.zhao.ml@gmail.com}.

a diverse set of strategies and could have better generalization (Pan and Yang 2009; Tobin et al. 2017; Akkaya et al. 2019).

The contributions of this paper are three-fold. First, based on the novel population diversity objective that considers both individual diversity and pairwise diversity for agents in the population, we derive a safe and computationally efficient surrogate objective, the population entropy, which is the lower bound of the population diversity objective. Secondly, we propose the maximum entropy population-based training framework, which comprises training a diverse population and then training a robust AI agent using this population. Last but not least, we evaluate our method and other state-of-the-art methods on the Overcooked game environment (Ghost Town Games 2016), with both human proxy models and real humans.

Preliminaries

Markov Decision Process: A two-player Markov Decision Process (MDP) is defined by a tuple $\mathcal{M} = \langle \mathcal{S}, \{\mathcal{A}^{(i)}\}, \mathcal{P}, \gamma, R \rangle$ (Boutilier 1996), where $\mathcal{S}$ is a set of states; $\mathcal{A}^{(i)}$ is a set of the i-th agent's actions, where $i \in [1, 2]$; $\mathcal{P}$ is the transition dynamics that maps the current state and all agents' actions to the next state; γ is the discount factor; R is the reward function. The i-th agent's policy is $\pi^{(i)}$. A trajectory is denoted by τ. The shared objective is to maximize the expected sum rewards, which is $\mathbb{E}_\tau \left[\sum_t R(s_t, a_t)\right]$, where $a_t = (a_t^{(1)}, a_t^{(2)})$. We can extend the objective to infinite horizon problems by the discount factor γ to ensure that the sum of expected rewards is finite. In the perspective of a single agent, the other agent can be treated as a part of the environment. In this case, we can reduce the process to the partially observable MDP (POMDP) for that agent.

AI Agent, Population, and Human: In the case of human-AI coordination, we have a two-player MDP, in which one player is human, and the other is AI. Throughout this paper, we use the phrase *AI agent* to explicitly denote the agent that plays the AI role in human-AI coordination. The *population* of agents is used to train the AI agent to make it capable of cooperating with different partner agents. The *human* policy is represented as $\pi^{(H)}$ and a model of the human policy is $\hat{\pi}^{(H)}$. The AI agent is denoted as $\pi^{(A)}$.

Environment: We use the Overcooked environment (Carroll et al. 2019) as the human-AI coordination testbed, see Figure 2. In the Overcooked game, it naturally requires coordination and collaboration between the two players to have a high score. The players are tasked to cook soups.

Maximum Entropy RL: Standard reinforcement learning maximizes the expected sum of rewards $\mathbb{E}_\tau \left[\sum_t R(s_t, a_t)\right]$, At the beginning of learning, almost all actions have equal probability. After some training, some actions have a higher probability in the direction of accumulating more rewards. Subsequently, the entropy of the policy is reduced over time during training (Mnih et al. 2016). while maximum entropy RL augments the standard RL objective with the expected entropy of the policy (Ziebart 2010; Haarnoja et al. 2018b), which incentives the agent to select the non-dominate actions. The maximum entropy RL objective is defined as:

$$J(\pi) = \sum_t \mathbb{E}_{(s_t, a_t)\sim\pi} \left[R(s_t, a_t) + \alpha \mathcal{H}(\pi(\,\cdot\,|s_t)) \right], \quad (1)$$

where parameter α adjusts the relative importance of the entropy bonus against the reward. The maximum entropy RL objective has several advantages. First, the policy favors more exploration and mitigates the issue of early convergence (Haarnoja et al. 2017; Schulman, Chen, and Abbeel 2017). Secondly, the policy can capture multiple modes of near-optimal behaviors and has better robustness (Haarnoja et al. 2018a, 2019).

Method

In this section, we first define the *Population Diversity* objective, which includes average individual policy entropy and average pairwise difference among policies. Secondly, we derive its safe and computationally efficient lower bound, *Population Entropy*, as the surrogate objective for optimization. Thirdly, we illustrate the *Maximum Entropy Population-based training* framework, which comprises training a maximum entropy population and training a robust AI agent via prioritized sampling.

Population Diversity

Motivated by maximum entropy RL, we want to make the policies in the population exploratory and diverse. First, by utilizing the maximum entropy bonus, we encourage each policy itself to be exploratory and multi-modal. Secondly, to encourage the policies $\{\pi^{(1)}, \pi^{(2)}, ..., \pi^{(n)}\}$ in the population to be complementary and mutually different, we utilize the KL divergence of each policy pair in the population as part of our objective. Formally, we define the *Population Diversity* (PD) as a combination of the average entropy of each agent's policy and the average KL-divergence between each agent pair in the population. Mathematically,

$$\text{PD}(\{\pi^{(1)}, \pi^{(2)}, ..., \pi^{(n)}\}, s_t) := \frac{1}{n} \sum_{i=1}^{n} \mathcal{H}(\pi^{(i)}(\,\cdot\,|s_t)) \quad (2)$$

$$+ \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} D_{\text{KL}}(\pi^{(i)}(\,\cdot\,|s_t), \pi^{(j)}(\,\cdot\,|s_t)),$$

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

$$D_{\text{KL}}(\pi^{(i)}(\,\cdot\,|s_t), \pi^{(j)}(\,\cdot\,|s_t)) = \quad (3)$$

$$\sum_{a\in\mathcal{A}} \pi^{(i)}(a_t|s_t) \log \frac{\pi^{(i)}(a_t|s_t)}{\pi^{(j)}(a_t|s_t)}, \quad (4)$$

$$\mathcal{H}(\pi^{(i)}(\,\cdot\,|s_t)) = -\sum_{a\in\mathcal{A}} \pi^{(i)}(a_t|s_t) \log \pi^{(i)}(a_t|s_t). \quad (5)$$

Although the PD objective not only captures a single agent's explorability but also encourages agents' policies to be mutually distinct, evaluating this objective requires a quadratic runtime complexity of $O(n^2)$, where n is the population size. Besides, as the KL-divergence is unbounded, optimizing this objective as part of the reward function may lead to convergence issues.

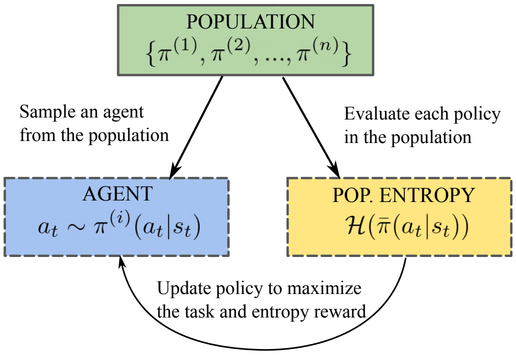

POPULATION
$\{\pi^{(1)}, \pi^{(2)}, ..., \pi^{(n)}\}$
Sample an agent from the population
Evaluate each policy in the population
AGENT
$a_t \sim \pi^{(i)}(a_t|s_t)$
POP. ENTROPY
$\mathcal{H}(\bar{\pi}(a_t|s_t))$
Update policy to maximize the task and entropy reward

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.

Population Entropy

To improve the stability and the runtime complexity of the PD objective, we derive a bounded and efficient surrogate objective *Population Entropy* (PE) for optimization, which is defined as the entropy of the mean policies of the population. Mathematically,

$$\text{PE}(\{\pi^{(1)}, \pi^{(2)}, ..., \pi^{(n)}\}, s_t) := \mathcal{H}(\bar{\pi}(\cdot|s_t)), \tag{6}$$
$$\text{where } \bar{\pi}(a_t|s_t) := \frac{1}{n}\sum_{i=1}^{n} \pi^{(i)}(a_t|s_t).$$

PE serves as a lower bound of the PD objective.

Theorem 1. *Let the population diversity be defined as Equation (2). Let the population entropy be defined as Equation (6). Then, we have*

$$PD(\{\pi^{(1)}, \pi^{(2)}, ..., \pi^{(n)}\}, s_t) \tag{7}$$
$$\geq PE(\{\pi^{(1)}, \pi^{(2)}, ..., \pi^{(n)}\}, s_t), \tag{8}$$

where n is the population size. Proof. See Appendix A. □

The PE objective serves as a lower bound for the PD objective, which requires only a linear runtime complexity $O(n)$. Moreover, when defined on categorical distribution, the PE objective is bounded, which makes it desirable to be optimized as part of the reward function. Therefore, we use the derived PE objective for optimization.

Training a Maximum Entropy Population

With the PE objective, we can train a population of agents, which can cooperate well with each other with mutually distinct strategies. Therefore, similar to the objective in maximum-entropy RL, we define the objective for MEP training as follows:

$$J(\bar{\pi}) = \sum_t \mathbb{E}_{(s_t,a_t)\sim\bar{\pi}} \left[R(s_t, a_t) + \alpha\mathcal{H}(\bar{\pi}(\cdot|s_t))\right], \tag{9}$$

where $\bar{\pi}$ is the mean policy of the population and α determines the relative weight of the population entropy term with respect to the task reward. As $\bar{\pi}(a_t|s_t)$ can be written as $\frac{1}{n}\sum_{i=1}^{n} \pi^{(i)}(a_t|s_t)$, to optimize Equation (9), we can uniformly sample agents from the population to maximize the augmented reward function $R(s_t, a_t) - \alpha \log \bar{\pi}(a_t|s_t)$. The task reward is related to the agent and its partner agent, which is a copied version of itself playing the partner role in our case. When the centralized PE reward is calculated, it considers all the agents in the population. We summarize the method of training a maximum entropy population in Algorithm 1 and Figure 1.

Algorithm 1: Maximum Entropy Population

```
while not converged do
    Sample agent from population:
      $\pi^{(i)} \sim \{\pi^{(1)}, \pi^{(2)}, ..., \pi^{(n)}\}$
    for $t \leftarrow 1$ to $steps\_per\_episode$ do
        Sample action $a_t \sim \pi^{(i)}(a_t|s_t)$.
        Step environment $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$.
        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 $\mathbb{E}_\tau [r]$.
```

After having the maximum entropy population, we utilize this diverse set of agents to train a robust AI agent that can be readily paired with any player, including real human. The intuition behind MEP is that the AI agent should be more robust when paired with a group of diversified partners during training than trained only via only self-play. In the extreme case, when the AI agent can coordinate well with an infinite set of different partners, it can also collaborate well with the human players. In a more realistic sense, the more diverse the population is, the more likely the population covers most of the human behaviors in the training set. Subsequently, the final AI agent should be less "panicked" when facing "abnormal" human actions.

Training a Robust Agent via Prioritized Sampling

With the maximum entropy population, training this robust agent is still non-trivial. If we train the robust agent (A) by pairing it with i-th agent uniformly sampled from the population, the resulting policy gradient is:

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_\tau \left[\sum_t \nabla_\theta \log \pi_\theta^{(A)}(a_t^{(A)}|s_t) \sum_t R(s_t, a_t^{(A)}, a_t^{(i)})\right],$$

where n is the population size. This "maximize average" paradigm may lead to an agent (A) that exploits the easy-to-collaborate partners as pairing with them will inevitably lead to much higher return, and therefore, ignores the hard-to-collaborate partners, which is orthogonal to our intention of training an robust agent. However, motivated by Vinyals et al. (Vinyals et al. 2019), we can mitigate this issue by skewing the sampling distribution as follows:

$$p(\pi^{(i)}) \propto 1/\mathbb{E}_\tau \left[\sum_t R(s_t, a_t^{(A)}, a_t^{(i)})\right],$$

We assign a higher priority to the agents that are relatively hard to collaborate with. By doing so, we change the

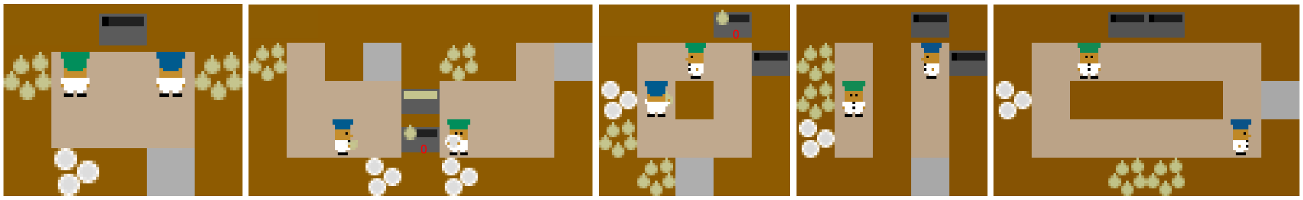


Figure 2: Overcooked environment: From left to right, the layouts are *Cramped Room*, *Asymmetric Advantages*, *Coordination Ring*, *Forced Coordination*, and *Counter Circuit*.

”maximize average” paradigm to a smooth approximation of ”maximize minimal” paradigm to mitigate the issue of over exploitation of easy-to-collaborate partners. In the extreme case, at each optimization step, if we always choose the hardest agent in the population to train the AI agent, we optimize a performance lower bound of the cooperation between the AI agent and any agent in the population. Mathematically,

$$\pi^{(A)} = \arg\max \min_{i\in\{1,\ldots,n\}} J(\pi^{(A)}, \pi^{(i)}), \tag{10}$$

where $J(\pi^{(A)}, \pi^{(i)})$ denotes the expected return achieved by $\pi^{(A)}$ and $\pi^{(i)}$ collaborating with each other. For more detail on the performance lower bound, Equation (10), see Appendixi B. We derive the performance connection between two pairs of agents, $(\pi^{(A)}, \pi^{(i)})$ and $(\pi^{(A)}, \pi^{(j)})$, when the partner agent $\pi^{(i)}$ in the first pair is ϵ-close to the other partner agent $\pi^{(j)}$ in the second pair, see Appendix C. If the population we used for training is diverse and representative enough such that we can find an agent that is ϵ-close to the human player's policy, then we have a performance lower bound of human-AI coordination.

In practice, we follow the rank-based approach as proposed by Schaul et al. (Schaul et al. 2016) in consideration for stability. The probability of the i-th agent to be sampled is:

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

where $\text{rank}(\cdot)$ is the ranking function ranging from 1 to n and β is a hyper-parameter for adjusting the strength of the prioritization.

Experiments

Environment: To evaluate the proposed method, we first use a toy environment, the matrix game (Lupu et al. 2021), see Figure 3, and then use the Overcooked environment (Carroll et al. 2019), see Figure 2. The Overcooked game naturally requires human-AI coordination to achieve a high score. The players are tasked to cook the onion soups as fast as possible. The relevant objects are onions, plates, and soups. Players are required to place three onions in a pot, cook them for 20 timesteps, put the cooked soup on a plate, and serve the soup. Afterward, the players receive a reward of 20. The six actions are up, down, left, right, no-operation, and interact. There are five different layouts, see Figure 2, and each exhibits a unique challenge.

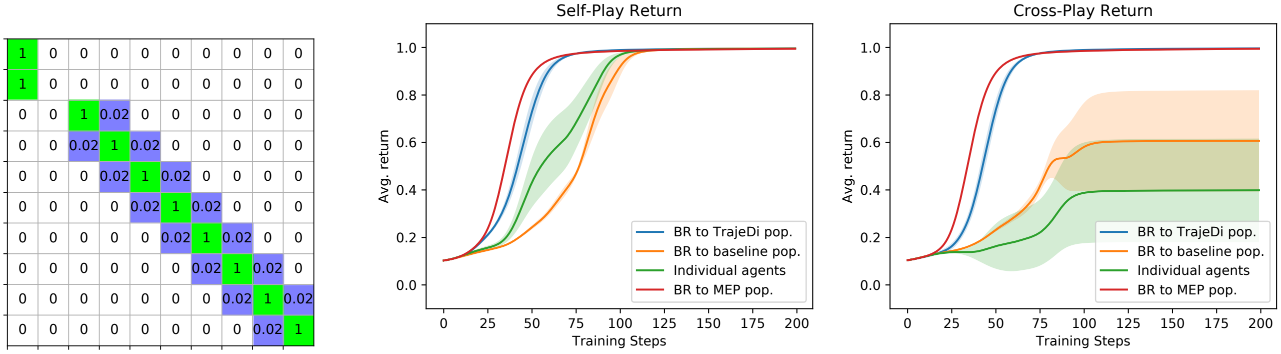
Experiments: First, we train the population using the PE bonus and investigate the effect of the entropy weight α. Secondly, we use the learned maximum entropy population to train the AI agent with the learning progress-based prioritized sampling and report the performance. In the ablation study, we show the effectiveness of both population entropy and prioritized sampling. We also show the comparison between MEP and Maximum Population Diversity (MPD), which maximizes the population diversity objective, see Section . We compare our method with other methods, including Self-Play (SP) Proximal Policy Optimization (Schulman et al. 2017; Carroll et al. 2019), Population Based Training (PBT) (Jaderberg et al. 2017; Carroll et al. 2019), Trajectory Diversity (TrajeDi) (Lupu et al. 2021), Fictitious Co-Play (FCP) (Strouse et al. 2021), and MPD. To evaluate these methods, we use the protocol proposed by Carroll et al. (Carroll et al. 2019), in which a human proxy model, H_{Proxy}, is used for evaluation. The human proxy model is trained through behavior cloning (Bain and Sammut 1999) on the collected human data. Furthermore, we conduct a user study using Amazon Mechanical Turk (AMT), in which we deploy our models through web interfaces and let real human players play with the AI agents. The experimental details are shown in Appendix D. Our code is available at https://github.com/ruizhaogit/maximum_entropy_population_based_training.

Question 1. *How does MEP perform in toy environments?*

In the single-step collaborative matrix game (Lupu et al. 2021), player 1 must select a row while player 2 chooses a column independently. Both agents get the reward associated with the intersection of their choices at the end of the game. We use the same evaluation protocol as proposed by Lupu et al. (Lupu et al. 2021). As shown in Figure 3, MEP converges faster than TrajeDi in both self-play return and cross-play return. An extensive hyper-parameter search for TrajeDi can be found in the figure of Appendix E.

Question 2. *Does PE reward increase the entropy of the population?*

To check whether the PE reward increases the entropy of the population, we investigate the effect of different values of α and record the entropy of the population that corresponds to the best reward during training in Table 1. As shown in Table 1, the population entropy with $\alpha > 0$ is generally greater than that with $\alpha = 0$ and the overall trend is the population entropy increases as α gets larger. This em-



1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	1	0.02	0	0	0	0	0	0
0	0	0.02	1	0.02	0	0	0	0	0
0	0	0	0.02	1	0.02	0	0	0	0
0	0	0	0	0.02	1	0.02	0	0	0
0	0	0	0	0	0.02	1	0.02	0	0
0	0	0	0	0	0	0.02	1	0.02	0
0	0	0	0	0	0	0	0.02	1	0.02
0	0	0	0	0	0	0	0	0.02	1

Figure 3: Performance comparison: Training and test performances in the matrix game. Shown are the results for Best Responses (BRs) to MEP agents, BRs to TrajeDi populations, BRs to baseline populations, and individual agents.

α	Cramped Rm.	Asymm. Adv.	Coord. Ring	Forced Coord.	Counter Circ.
0.000	0.971	1.120	0.878	0.970	0.988
0.001	1.031	1.051	0.907	0.858	1.152
0.005	0.949	1.075	0.901	0.889	1.038
0.010	1.057	1.139	0.840	1.079	1.151
0.020	1.029	1.074	0.947	1.093	1.171
0.030	1.134	1.203	1.028	0.957	1.715
0.040	1.194	1.353	1.122	1.460	1.791
0.050	1.127	1.364	0.996	1.703	1.791

Table 1: Population entropy with different α: In this table, α denotes the weight of the PE reward in Equation (9).

pirical finding verifies that PE reward effectively increases the entropy of the population.

Question 3. *What does an MEP population look like?*

To have an intuition of what a maximum entropy population looks like, we show the behavior of the agents in the supplementary video from 0:01 to 0:21. This video clip presents the population trained without and with the PE reward in the first and second row, respectively. In the first row, the blue and green agents move in a synchronized way most of the time, and the routines among the five agent pairs are similar. However, in the second row of the video clip, the agents' behavior trained with population entropy reward is more diverse. The movements of the blue agent and the green agent are less synchronized, and their routines are less predictable. For example, in the second agent pair, the blue agent throws the onion into a random spot or passes the first and second plates in a row, and in the fifth agent pair, the green agent behaves in a highly unexpected way.

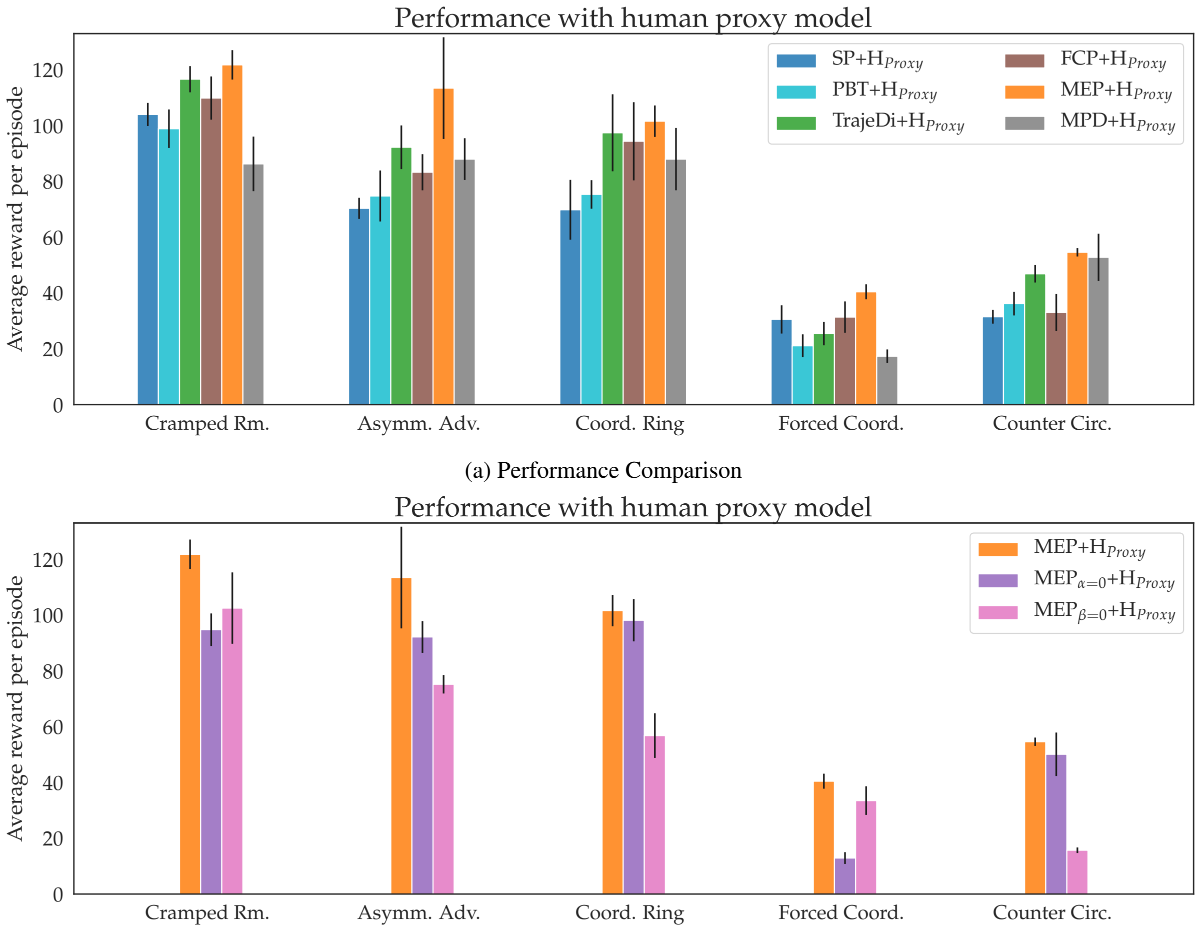
Question 4. *How does MEP compare to other methods?*

We pair each agent trained with SP, PBT, TrajeDi, FCP, MPD, and MEP, with the human proxy model H_{Proxy}, and evaluate the team performance in all the five layouts, as shown in Figure 2. Following the evaluation protocol proposed by Carroll et al. (Carroll et al. 2019), we use the cumulative rewards over a horizon of 400 timesteps as the proxy for coordination ability since good coordination between teammates is essential to achieve high scores in the Overcooked game. For all the results, we report the average reward per episode and the standard error across five different random seeds. As shown in Figure 4a, MEP outperforms other methods in all five layouts when paired with a human proxy model. Additionally, according to the ablation test shown in Figure 4b, both the population entropy reward and the prioritized sampling are necessary components for achieving the best performance.

We did a hyper-parameter search for TrajeDi on the discounting factor γ, see the figure in Appendix E, and report the best results, which use γ as 1.0 in Figure 4a. We also did a hyper-parameter search for FCP on the population size. By default, for TrajeDi, MPD, and MEP, we use a population size of 5. However, we use a population size of 10 for FCP in Figure 4a. The performance comparison between FCP with population size of 5 and 10 is shown in the figure in Appendix E. As the results show, a larger population size indeed improves the performance. However, MEP outperforms FCP with only half the population size, see Figure 4a.

Question 5. *How does MEP perform with real humans?*

For this human-AI coordination test, we use Amazon Mechanical Turk (AMT) and follow the same evaluation procedure proposed by Carroll et al. (Carroll et al. 2019). We evaluate TrajeDi, FCP, and MEP-trained AI agents and measure their average episode reward when the agents are paired with a real human player. We reuse the testing results of SP and PBT from the human-AI evaluation on AMT carried out by Carroll et al. (Carroll et al. 2019). These testing results are compatible because the evaluation procedure is the same



(a) Performance Comparison

(b) Ablation tests

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.

and uses a between-subjects design, meaning each user is only paired with a single AI agent. The results are presented in Figure 5. The chart in Figure 5 shows that, on average, across all five layouts, MEP outperforms other methods, including SP, PBT, FCP, and TrajeDi, and its performance is on par with the Human-Human coordination performance.

Question 6. *What does AI do when paired with humans?*

Here, we show and analyze some qualitative behaviors observed during the real human-AI coordination experiments, which are shown in the supplementary video from 0:22 to 2:27. From 0:24 to 0:44, we observe that in the Forced Coordination layout, the MEP-trained agent is more robust and less likely to get stuck during coordination than SP and PBT. Next, from 0:44 to 1:09, in the Asymmetric Advantage layout, the SP-trained and the PBT-trained agents only learned to put the onion into the pot and did not learn to deliver the onion soup, while the MEP-trained agent learned to put the onion into the pot and learned to deliver the onion soup when its human partner is busy. Similarly, from 1:09 to 1:29, in the Cramped Room layout, the SP-trained and PBT-trained agents only learned to use the plate to take the soup, whereas the MEP-trained agent additionally learned to carry the onion to the pot. Interestingly, from 1:29 to 1:56, in the Coordination Ring layout, the SP-trained and PBT-trained agent only learned to deliver the onion soup in one direction, while the MEP-trained agent learned to deliver the soup clockwise and counterclockwise, depending on where its human partner stands. Last but not least, from 2:01 to 2:26, in the Counter Circuit layout, the SP-trained and PBT-trained agents only learned to pass the onion over the “counter”.

Related Work

Deep reinforcement learning has gained many successes in competitive games, such as Go (Silver et al. 2017),

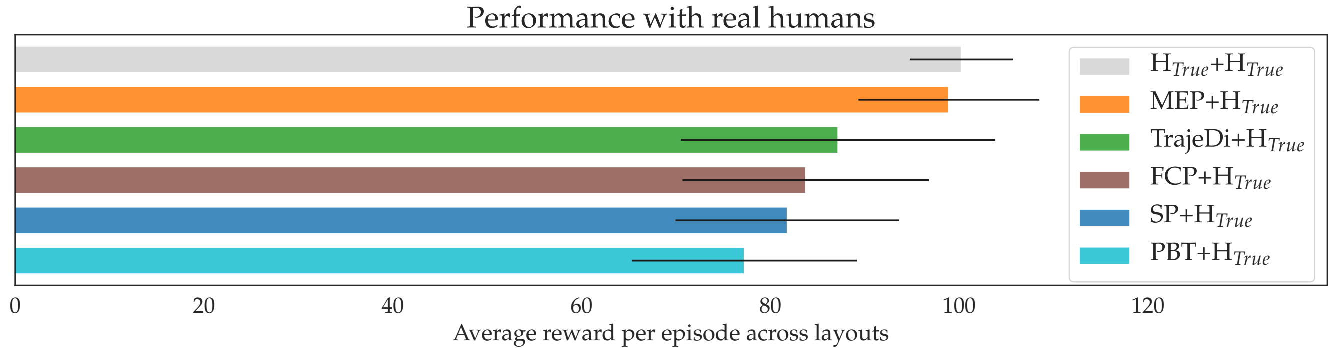


Figure 5: Performance with real humans

Dota (OpenAI 2019), Quake (Jaderberg et al. 2019), and StarCraft (Vinyals et al. 2019), where self-play (SP) and population-based training (PBT) have been leveraged to improve performance. However, the agents trained via SP or PBT tend to learn overly specific policies in collaborative environments (Carroll et al. 2019). Recent works (Lerer and Peysakhovich 2018; Tucker, Zhou, and Shah 2020; Carroll et al. 2019; Knott et al. 2021) tackle the collaboration problem using some behavioral data from the partner to select the equilibrium of the existing agents (Lerer and Peysakhovich 2018; Tucker, Zhou, and Shah 2020) or build and incorporate a human model into the training process (Carroll et al. 2019; Knott et al. 2021). In this work, we consider the zero-shot setting, where no behavioral data from the human partner is available during training (Hu et al. 2020).

From a Bayesian perspective, when we do not have a prior on what the human policies look like, we should train the AI agent to be robust and capable of collaborating with a diverse set of policies (Murphy 2012). One popular approach towards robust AI agents is through maximum entropy reinforcement learning (Ziebart et al. 2008; Ziebart 2010; Fox, Pakman, and Tishby 2015; Haarnoja et al. 2017, 2018b), and many previous works leverage it as a means of encouraging exploration (Schulman, Chen, and Abbeel 2017; Haarnoja et al. 2018b) or skill discovering (Eysenbach et al. 2019; Zhao et al. 2021). However, obtaining a diversified population through entropy maximization is still subjective to research. In Multi-agent Reinforcement Learning (MARL), a group of agents is trained to achieve a common goal by Centralized Training and Decentralized Execution (CTDE) (Lowe et al. 2017; Foerster et al. 2018).

The idea of MEP shares a common intuition with domain randomization, where some features of the environment are changed randomly during training to make the policy robust to that feature (Tobin et al. 2017; Peng et al. 2018; Tan et al. 2018; Akkaya et al. 2019; Tang et al. 2020). In general, MEP can be seen as a domain randomization technique, where the randomization is conducted over a set of partners' policies.

A recent related work – TrajeDi (Lupu et al. 2021) has a similar motivation and encourages the trajectories from different agents in the population to be distinct. TrajeDi directly optimizes the trajectory-level Jensen-Shannon divergence between policies as part of the policy loss, while our method trains the population with reward function augmented by population entropy on the action level. However, the variance of the evaluated gradient from TrajeDi could be unbounded due to its trajectory-level importance sampling part, while our formulation does not have importance sampling terms. Another recent work – FCP (Strouse et al. 2021) is closely related to our work. Strouse et al. (Strouse et al. 2021) show that with diversity induced by different checkpoints and different random seeds, the agent can generalize well in collaborative games. However, in our experiments, we find out that FCP requires a relatively large population to work well. Compared to FCP, MEP only uses half the population and works better.

There are also other population diversity-based methods, such as Diversity via Determinants (DvD) (Parker-Holder et al. 2020) and Diversity-Inducing Policy Gradient (DIPG) (Masood and Doshi-Velez 2019). DvD is based on the determinant of the kernel matrix, and DIPG is derived from Maximum Mean Discrepancy (MMD). These two methods are formulated for the single-agent setting, whereas MEP is designed for the multi-agent cooperative setting. In games with non-transitive dynamics where strategic cycles exist, e.g., Rock-Paper-Scissors, Policy-Space Response Oracle (PSRO)-based methods (Balduzzi et al. 2019; Perez-Nieves et al. 2021; Liu et al. 2021) provide solutions to learn diverse behaviors. In general, MEP is complementary to these previous works and is applicable to many human-AI coordination tasks.

Conclusion

This paper introduces Maximum Entropy Population-based training (MEP), a deep reinforcement learning method for robust human-AI coordination. With the derived population entropy reward encouraging diversity in policies and the learning progress-based prioritized sampling enhancing generalization to unencountered policies, the MEP-trained agents demonstrate more flexibility and robustness to various human strategies. Our result, which bridges maximum entropy RL and PBT, suggests that entropy maximization can be a promising avenue for achieving diversity and robustness in reinforcement learning. Combing MEP with other MARL algorithms could be meaningful directions for future work.

References

Akkaya, I.; Andrychowicz, M.; Chociej, M.; Litwin, M.; McGrew, B.; Petron, A.; Paino, A.; Plappert, M.; Powell, G.; Ribas, R.; Schneider, J.; Tezak, N.; Tworek, J.; Welinder, P.; Weng, L.; Yuan, Q.; Zaremba, W.; and Zhang, L. 2019. Solving Rubik's Cube with a Robot Hand. *arXiv preprint*.

Bain, M.; and Sammut, C. 1999. A Framework for Behavioural Cloning. In *Machine Intelligence 15, Intelligent Agents [St. Catherine's College, Oxford, July 1995]*, 103–129. Oxford, UK, UK: Oxford University. ISBN 0-19-853867-7.

Balduzzi, D.; Garnelo, M.; Bachrach, Y.; Czarnecki, W.; Perolat, J.; Jaderberg, M.; and Graepel, T. 2019. Open-ended learning in symmetric zero-sum games. In *International Conference on Machine Learning*, 434–443. PMLR.

Boutilier, C. 1996. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, 195–210. Morgan Kaufmann Publishers Inc.

Carroll, M.; Shah, R.; Ho, M. K.; Griffiths, T.; Seshia, S.; Abbeel, P.; and Dragan, A. 2019. On the Utility of Learning about Humans for Human-AI Coordination. In *Advances in Neural Information Processing Systems*, 5175–5186.

Carter, S.; and Nielsen, M. 2017. Using Artificial Intelligence to Augment Human Intelligence. *Distill*. https://distill.pub/2017/aia.

Engelbart, D. C. 1962. Augmenting human intellect: A conceptual framework. *Menlo Park, CA*.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2019. Diversity is All You Need: Learning Skills without a Reward Function. In *International Conference on Learning Representations*.

Foerster, J.; Assael, I. A.; De Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in neural information processing systems*, 2137–2145.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Fox, R.; Pakman, A.; and Tishby, N. 2015. Taming the noise in reinforcement learning via soft updates. *arXiv preprint arXiv:1512.08562*.

Ghost Town Games. 2016. Overcooked. https://store.steampowered.com/app/448510/Overcooked/. Accessed: 2016-08-03.

Haarnoja, T.; Ha, S.; Zhou, A.; Tan, J.; Tucker, G.; and Levine, S. 2019. Learning to Walk Via Deep Reinforcement Learning. In *Robotics: Science and Systems*.

Haarnoja, T.; Hartikainen, K.; Abbeel, P.; and Levine, S. 2018a. Latent Space Policies for Hierarchical Reinforcement Learning. *arXiv preprint arXiv:1804.02808*.

Haarnoja, T.; Tang, H.; Abbeel, P.; and Levine, S. 2017. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, 1352–1361. PMLR.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018b. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, 1861–1870. PMLR.

Han, L.; Xiong, J.; Sun, P.; Sun, X.; Fang, M.; Guo, Q.; Chen, Q.; Shi, T.; Yu, H.; and Zhang, Z. 2020. Tstarbot-x: An open-sourced and comprehensive study for efficient league training in starcraft ii full game. *arXiv preprint arXiv:2011.13729*.

Hu, H.; Lerer, A.; Peysakhovich, A.; and Foerster, J. 2020. "Other-Play" for Zero-Shot Coordination. In *International Conference on Machine Learning*, 4399–4410. PMLR.

Jaderberg, M.; Czarnecki, W. M.; Dunning, I.; Marris, L.; Lever, G.; Castañeda, A. G.; Beattie, C.; Rabinowitz, N. C.; Morcos, A. S.; Ruderman, A.; Sonnerat, N.; Green, T.; Deason, L.; Leibo, J. Z.; Silver, D.; Hassabis, D.; Kavukcuoglu, K.; and Graepel, T. 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443): 859–865.

Jaderberg, M.; Dalibard, V.; Osindero, S.; Czarnecki, W. M.; Donahue, J.; Razavi, A.; Vinyals, O.; Green, T.; Dunning, I.; Simonyan, K.; et al. 2017. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*.

Kleiman-Weiner, M.; Ho, M. K.; Austerweil, J. L.; Littman, M. L.; and Tenenbaum, J. B. 2016. Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. In *CogSci*.

Knott, P.; Carroll, M.; Devlin, S.; Ciosek, K.; Hofmann, K.; Dragan, A.; and Shah, R. 2021. Evaluating the Robustness of Collaborative Agents. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 1560–1562.

Lerer, A.; and Peysakhovich, A. 2017. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068*.

Lerer, A.; and Peysakhovich, A. 2018. Learning social conventions in markov games. *arXiv preprint arXiv:1806.10071*.

Liu, X.; Jia, H.; Wen, Y.; Yang, Y.; Hu, Y.; Chen, Y.; Fan, C.; and Hu, Z. 2021. Unifying Behavioral and Response Diversity for Open-ended Learning in Zero-sum Games. *arXiv preprint arXiv:2106.04958*.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 6379–6390.

Lupu, A.; Cui, B.; Hu, H.; and Foerster, J. 2021. Trajectory Diversity for Zero-Shot Coordination. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 7204–7213. PMLR.

Masood, M. A.; and Doshi-Velez, F. 2019. Diversity-inducing policy gradient: Using maximum mean discrepancy to find a set of diverse policies. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937. PMLR.

Murphy, K. P. 2012. *Machine Learning: A Probabilistic Perspective. Adaptive Computation and Machine Learning*. MIT press.

OpenAI. 2019. OpenAI Five Finals. https://openai.com/blog/openai-five-finals/. Accessed: 2019-03-26.

Pan, S. J.; and Yang, Q. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359.

Parker-Holder, J.; Pacchiano, A.; Choromanski, K. M.; and Roberts, S. J. 2020. Effective Diversity in Population Based Reinforcement Learning. *Advances in Neural Information Processing Systems*, 33.

Peng, X. B.; Andrychowicz, M.; Zaremba, W.; and Abbeel, P. 2018. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, 1–8. IEEE.

Perez-Nieves, N.; Yang, Y.; Slumbers, O.; Mguni, D. H.; Wen, Y.; and Wang, J. 2021. Modelling Behavioural Diversity for Learning in Open-Ended Games. In *International Conference on Machine Learning*, 8514–8524. PMLR.

Rawlik, K.; Toussaint, M.; and Vijayakumar, S. 2013. On stochastic optimal control and reinforcement learning by approximate inference. In *Twenty-third international joint conference on artificial intelligence*.

Resnick, C.; Kulikov, I.; Cho, K.; and Weston, J. 2018. Vehicle community strategies. *arXiv preprint arXiv:1804.07178*.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2016. Prioritized experience replay. In *International Conference on Learning Representations*.

Schulman, J.; Chen, X.; and Abbeel, P. 2017. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shum, M.; Kleiman-Weiner, M.; Littman, M. L.; and Tenenbaum, J. B. 2019. Theory of minds: Understanding behavior in groups through inverse planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6163–6170.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Strouse, D.; McKee, K.; Botvinick, M.; Hughes, E.; and Everett, R. 2021. Collaborating with Humans without Human Data. *Advances in Neural Information Processing Systems*, 34.

Tan, J.; Zhang, T.; Coumans, E.; Iscen, A.; Bai, Y.; Hafner, D.; Bohez, S.; and Vanhoucke, V. 2018. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*.

Tang, Z.; Yu, C.; Chen, B.; Xu, H.; Wang, X.; Fang, F.; Du, S. S.; Wang, Y.; and Wu, Y. 2020. Discovering Diverse Multi-Agent Strategic Behavior via Reward Randomization. In *International Conference on Learning Representations*.

Tesauro, G. 1994. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2): 215–219.

Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; and Abbeel, P. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 23–30. IEEE.

Toussaint, M. 2009. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th annual international conference on machine learning*, 1049–1056.

Tucker, M.; Zhou, Y.; and Shah, J. 2020. Adversarially Guided Self-Play for Adopting Social Conventions. *arXiv preprint arXiv:2001.05994*.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Zhao, R.; Gao, Y.; Abbeel, P.; Tresp, V.; and Xu, W. 2021. Mutual Information State Intrinsic Control. In *International Conference on Learning Representations*.

Zhao, R.; Sun, X.; and Tresp, V. 2019. Maximum Entropy-Regularized Multi-Goal Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, 7553–7562. PMLR.

Ziebart, B. D. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; Dey, A. K.; et al. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 8, 1433–1438.