Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit Nr. 3525485

Into the Minds of the Chefs

Using Theory of Mind for Robust Collaboration with Humans in Overcooked

Constantin Ruhdorfer

Course of Study:	Informatik
Examiner:	Prof. Dr. Andreas Bulling
Supervisor:	Matteo Bortolleto, M.Sc.
Commenced:	April 17, 2023
Completed:	October 17, 2023
CR-Classification:	I.7.2

Abstract

The ability to infer the believes, desires and preferences of other humans around us - referred to Theory of Mind - is crucial for effective human cooperation. In this work we investigate how this ability can facilitate cooperation among artificial agents, particularly in zero-shot cooperation scenarios where the partner might be human. While previous works had access to ground truth belief states of the other agents during training, we study Theory of Mind based collaboration in a multi-agent collaborative environment where no ground-truth belief states exists, namely Overcooked. We propose three auxiliary tasks that agents are trained with which in turn are inspired by Theory of Mind: Predicting (i) partner's next action, (ii) partner's next strategic goal and (iii) partner's neural state. Our research demonstrates that self-play agents trained with these auxiliary tasks exhibit improved competence in playing the game but tend to underperform when interacting with others, suggesting a tendency towards overspecialisation on oneself. To address this challenge, we add our auxiliary tasks to population based methods for training against diverse populations and show increased performance on several benchmarks, especially on the layout Asymmetric Advantages. Overall, our work shows the importance of explicitly modelling Theory of Mind for multi-agent cooperation.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

CNN Convolutional Neural Network. 31

FC Fully Connected. 31

FCP Fictitious Co-Play. 18

GRU Gated Recurrent Unit. 32

HSP Hidden-Utility Self-Play. 19

LN Layer Normalisation. 31

MDP Markov Decision Process. 22

MEP Maximum Entropy Population Based Training. 18

MSE Mean Squared Error. 25

MToM Machine Theory of Mind. 20

POMDP Partially observable Markov Decision Process. 22, 23

PPO Principal Policy Optimisation. 24

RL Reinforcement Learning. 13

SP Self-Play. 13

ToM Theory of Mind. 5

1 Introduction

During collaboration humans naturally perceive the current intentions, beliefs and desires of those surrounding them and react to them accordingly even from a young age. This ability of inferring others' mental states is referred to as Theory of Mind (ToM) [BJST17; BLF85; PW78; YDF08]. Imagine a couple is doing the dishes after dinner. If one partner takes a plate, cleans it in the sink, dries it off and hands it to the other, what were they thinking? And what do they expect of them? Clearly, they want them to put the plate away, an intention one can naturally infer from observing them alone. We humans draw this inference as we are able to reason not only about the raw motions this person is going through but instead, we are reasoning about their behaviour as a whole, including their current mental states. Moreover, this mostly occurs with little or incomplete verbal interaction. There is usually no explicit agreement in them putting the dish in the correct cabin and not in a drawer, the trash or back in the sink. Clearly, one is able to deduct these goals from non-verbal cues alone.

We infer others' mental states subconsciously, even without actively thinking about it [SC13]. Think about all the small daily coordination tasks you are going through with complete strangers, i.e. how you are deciding who exits the subway train first during rush hour, whether you move to the right or left when you and someone else are on a collision path or who moves through a door first when you and someone else want to pass through it at the same time from opposite sides. Moreover, realise how awkward it is if this kind of coordination fails and you have misread your opposite. Recall, the times where you and someone else try to pass each other on the same site and both of you come to an unexpected hold.

ToM is clearly crucial for human-human coordination (also see Langley et al. [LCCS22]). Naturally, we are therefore interested in investigating how ToM affects other agent to agent coordination. While studying ToM only recently became a focus in studying artificial agents in general [RPS+18], recent work also starts to focus on collaborative settings [FWCL21], as-well as AI-AI [SNB22; YFZ+21] and human-AI cooperation [CSHD20; KCD+21].

Human-AI collaboration is non-trivial to achieve and is considered a long-standing challenge for AI [AZI+18; DHB+20; KWB+04]. Approaches for human-AI coordination can be separated by whether human data is used to build an effective model of human

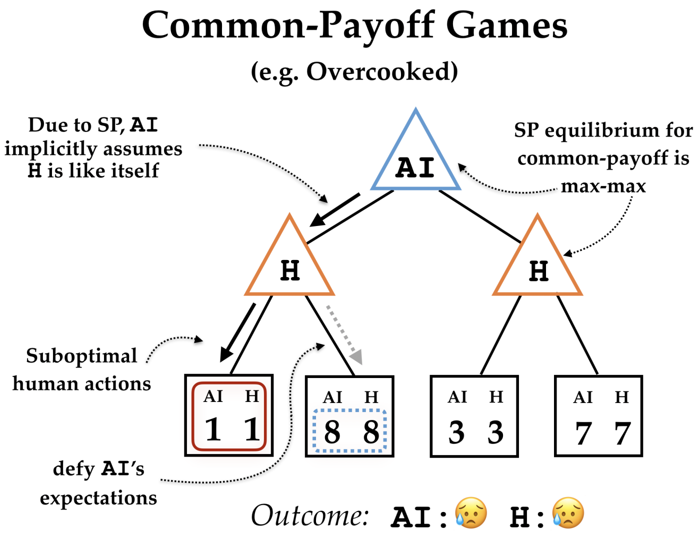

Figure 1.1: Common-Payoff games pose a special challenge for self-play based methods. Graphic taken from: [CSH+19].

behaviour [BSF+20; CSH+19; She16] or not [SMB+21; YGL+23]. Since human data is often hard to come by at appropriate scale, collaboration often needs to be achieved without making use of it. In these settings, artificial agents need to learn to cooperate with a novel human partner – known a zero-shot coordination [HLPF20; KZGR23].

While reinforcement learning (RL) and self-play (SP) [Tes94] have been at the core of many recent successes in the advancement of AI [HDG+19; SSS+17] – including playing Go [SHM+16; SSS+17], chess [SHS+17] or even many games at once [RZP+22] at the super-human level – these techniques fail to train policies that solve the zero-shot coordination problem in cooperative environments. SP agents often only learn to cooperate with *themselves*, producing policies that are not capable of cooperating well with novel partners, especially not with humans. This is mostly due to the nature of cooperative environments, as illustrated in Figure 1.1. Cooperative games belong to the class of *common-payoff games* in which both agents observe a shared reward that is based on a common objective. In such games, SP agents are biased to learn max-max policies, i.e. to learn which is the best move to make given that the other agent will also pick the best move. Note that this assumption does not always hold true for humans. Therefore, a more sound approach to human-AI interaction would be to find the best move given the move your partner is likely to play. This motivates building agents that are *robust* under zero-shot cooperation.

Given the non-trivial nature of human-AI cooperation, the role of Theory of Mind in human cooperation and the recent of emergence of Machine Theory of Mind, we propose the use of MToM to improve cooperation.

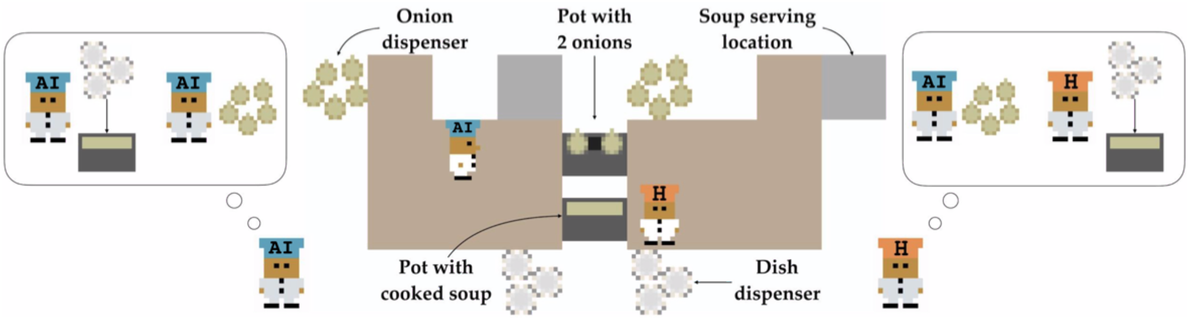

Figure 1.2: Overcooked-AI [CSH+19] is a challenging two-agent cooperation benchmark designed to study human-AI cooperation specifically. Graphic taken from: [CSH+19].

Our contributions are the following: (i) We design a self-supervised Theory of Mind mechanism in the popular human-AI cooperation environment Overcooked-AI [CSH+19] (see Figure 1.2), (ii) we propose three different belief supervision targets based on the abstractness of the belief representation, denoted *low-level*, *high-level* and *neural representation*, (iii) we add this mechanism to multiple state-of-the-art (SOTA) algorithms and baselines, provide detailed ablations studies into their performance and while doing so propose or review multiple evaluation methods that do not depend on human input. With our contributions we aim to answer four research questions to examine the capabilities of cooperative ToM agents with their non-ToM counterparts:

H1 *ToM-versions of agents have higher average validation reward when playing with different partners compared to their non-ToM counter parts.*

H2 *ToM-versions of agents have higher average validation reward when playing with strongly biased policies compared to their non-ToM counter parts.*

H3 *ToM-versions of agents have higher unit test success rate compared to their non-ToM counter parts.*

H4 *ToM-versions of agents achieve highest evaluation reward when playing with other ToM agents.*

Our work shows the importance of modelling Theory of Mind in multi-agent, collaborative settings. Specifically, we show that our Theory of Mind agents outperform strong baselines on several evaluation metrics, for instance when playing with each other or playing with scripted (i.e. strongly biased) policies. We additionally also show that our agents are able to perform well with models of human behaviour and generally in challenging scenarios as established through unit testing for robustness. Lastly, we provide evidence that our agents are capable of correctly modelling the type of cooperation partner they encounter.

2 A Motivating Example

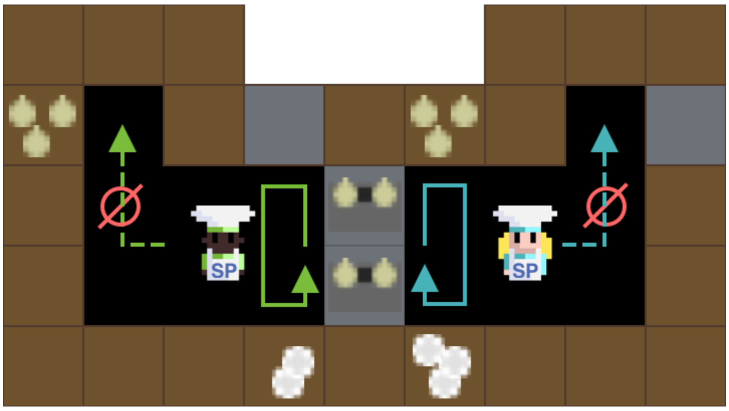

Figure 2.1: Robustness failure in the wild on Asymetric Advantages.

As a motivating example the kinds of problems our work aims to solve, we present an example that displays agents' failure to cooperate in the wild, see Figure 2.1. When training a self-play agent in Asymmetric Advantages layout, the agent learns a different policy depending on position that it will stubbornly stick to it. Since the right agent can access onions easier, it learns to only place onions into pots. Consequently, the left agent only learns to pick a plate and deliver the soup since the grey serving location is easier to access for it. This is highlighted in the Figure by the circling arrows. Both agents will never leave the middle of the layout (crossed-out arrows). Importantly, this behaviour achieves good performance *if two self-play agent are paired*. Conversely, if the right self-play agent is paired with a left agent that does not deliver soups, it fails to do so itself which results in a combined score of zero. The reason being that the self-play agent expects the other agent to deliver the soup as it has only encountered this behaviour during training. Thereby the self-play agent fails to solve the zero-shot coordination problem as it can not adapt to the behaviour of others. This is exactly the kind of robustness failures we address in this work.

3 Related Work

Our work connects the domains of human-AI collaboration, Machine Theory of Mind, AI robustness and safety, and multi-agent reinforcement learning through its unique approach of studying Theory of Mind as a robustness measure in a collaborative environment, namely Overcooked-AI.

3.1 Human-AI Collaboration

With the increasing deployment of AI systems in the real world, making sure that these systems are capable of collaboration has garnered increasing attention. This is especially true for settings in which the future cooperation partner is novel and of unknown nature as these most accurately resemble many real world scenarios such as autonomous driving or communication [KHA+16; LPB17; RKCW18; SSF16]. In the literature, these settings are discussed under two separate problem formulations depending on whether the agent is allowed to learn the policies of the other agent(s) or not during interaction. If so, it is referred to as the *ad-hoc coordination* and if not as the *zero shot coordination* problem [HLPF20; SKKR10; SR13]. In the latter case, one is challenged to create a fixed policy which then must be able to interact with new agents that were not encountered during training. Being able of only interacting with oneself and failing to interact with other novel partners motivates the study of this topic under the lens of AI robustness [KCD+21] and out-of-distribution generalization [KZGR23]. Under this view, failing to cooperate is a form of *over-fitting*.

A common approach in reinforcement learning is to train agents via self-play [Tes94]. While self-play can produce agents capable of achieving high scores with themselves, these agents typically rely on highly specialized conventions that do not work for agents they have not been trained with [BFC+20; CSH+19; KCD+21]. Taking this and the sub-optimal nature of human behaviour into account, Hu et al. [HLPF20] proposed *Other Play* as a method for zero-shot collaboration with sub-optimal partners in Hanabi [BFC+20]. This is one of a few works that explicitly take into account human properties [CSH+19; YGL+23] to improve human-AI coordination [SR13]. Human-AI coordination poses a challenge since humans can act sub-optimally according to their beliefs and

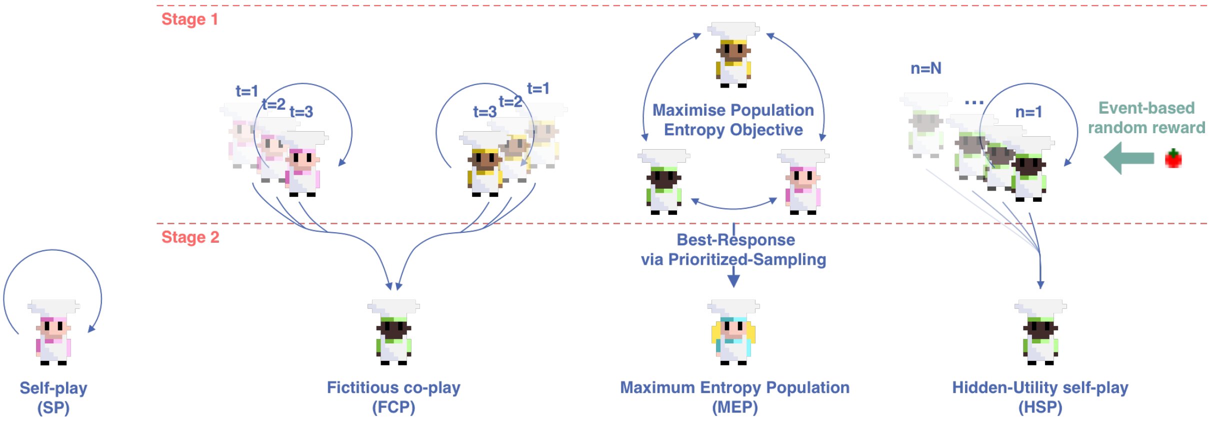
Stage 1
Stage 2
t=1
t=2
t=3
t=3
t=2
t=1
Maximise Population Entropy Objective
Best-Response via Prioritized-Sampling
n=N
...
n=1
Event-based random reward
Self-play (SP)
Fictitious co-play (FCP)
Maximum Entropy Population (MEP)
Hidden-Utility self-play (HSP)

Figure 3.1: Overview over methods in the literature for achieving zero-shot cooperation with humans in Overcooked-AI, namely Fictitious Co-Play [SMB+21], Maximum Entropy Population Based Training [ZSY+23], and Hidden Utility Self-Play [YGL+23]. Note that they all follow a two stage approach where first a population of diverse agents is obtained and then a best response to that is trained thereby capturing a diverse set of behaviours in its resulting policy.

biases [TK74]. Moreover, the conventions self-play and other naive algorithms develop often are unintelligible for human players and thus alternative methods need to be developed that deal with this issue.

Overcooked-AI [CSH+19] has been proposed as a challenging benchmark to study human-AI coordination, attracting a wealth of new research and publications [CMD20; FHZ+21; KCD+21; NGS+21; RMSM23; SMB+21; STSD22; YGL+23; ZSY+23]. In Overcooked-AI, agents are tasked with cooking a soup together, while making use of ingredient dispenser, pots, dishes and serving locations. The environment presents different layouts. Figure 1.2 shows a layout with only onions as the soup ingredient but other layouts support different ingredients and recipes. Upon the delivery of a finished soup both agents receive a shared reward, which makes Overcooked a common-payoff game.

Three approaches evaluated on Overcooked that are relevant to this work are Fictitious Co-Play [SMB+21], Maximum Entropy Population Based Training [ZSY+23], and Hidden Utility Self-Play [YGL+23] which we will describe in more detail below. An overview over these methods is presented in Figure 3.1.

3.1.1 Fictitious Co-Play

Fictitious Co-Play (FCP) [SMB+21] follows the insight that to cooperate well with a diverse set of others of differing skill level, agents should encounter diverse behaviour via a population of diverse agents with different skill levels. The approach is simple and can be summarized into two main stages. In the first step, a population of self-play agents initialised with varying random seeds is trained. Varying the random seeds is supposed to help generate different agents in the population. During training, three checkpoints are saved for each trained agent to simulate different skill level from the start, middle and end of training. These represent agents that are not or only slightly, somewhat and fully capable at the task. A best response agent is trained against the population, leading to an agent capable of adapting to agents with previously unseen behaviour.

This approach is simple but it has a few drawbacks. First, different random seeds do not guarantee substantially different behaviour in the trained policies. Second, since all policies in the population are trained via self-play they will likely result in agents with similar biases and assumptions that are specific to self-play. Thus, the best response agent will pick up on these behaviours even though they are likely harmful in zero-shot cooperation settings. Third, even if the population happens to include diverse behaviour it might still not cover human behaviour well thus having our agent learn to play against policies that are irrelevant at testing time.

3.1.2 Maximum Entropy Population Based Training

One approach that specifically addresses the first FCP drawback is Maximum Entropy Population Based Training (MEP) [ZSY+23]. MEP follows a similar two-step procedure for arriving at its final agent where first a population and then second a best response are trained. It differs from FCP in two important ways. MEP starts from the observation that the training population should be as diverse as possible. This is achieved by adding a entropy term to the policy objective, which encourages agents to take very different actions that still lead to (near) optimal results. Moreover, MEP introduces the idea of prioritized sampling for best response training. The best response agent is paired with an agent from the population based on how hard it is to collaborate with that specific agent, i.e. via prioritized sampling. This is dynamically adjusted during training. While addressing the first weakness of FCP, the last two still remain. The population is based on self-play agents alone and the behaviour of the agents in the population is unrelated to human behaviour.

3.1.3 Hidden Utility Self-Play

The core drawback of the previous approaches is that the behaviour of the agents in the population is only weakly related to human behaviour. Humans seldomly act as a reinforcement learning agent would. Humans pause and think, and tend to act according to their own preferences and biases (which are different between any two humans), often sub-optimally [CSH+19]. Ideally, we would like the population to reflect this. Following this line of thinking Yu et al. [YGL+23] studied human behaviour in Overcooked and realised that humans have event-based preferences that are unrelated to the game state. They for instance might prefer picking up a tomato over picking up an onion even though the onion would have higher expected reward. Based on this they introduced the Hidden-Utility Self-Play (HSP), which is also based on a two step training regime. To account for human preferences they constructed a population of biased self-play agents through giving additional reward to the agents for a special subset of game interactions. These subsets are different between agent pairs, leading to the agent having different preferences towards certain game mechanics. Since the pool of policies trained this way still will contain similar policies they furthermore filter their pool of policies down to the half that omits the most diverse behaviour through a simple greedy search. They fill the missing half with MEP agents which requires the training of both an MEP as well as an HSP population, thereby significantly increasing performance requirements.

With this setup HSP holds the current state-of-the-art performance in Overcooked for human-AI collaboration. Still, there are several drawbacks with this approach. First and foremost, HSP depends on creating biased SP agents via manually picked reward functions based on game events. While this might be possible for simple enviorments, it is unclear how such an approach could scale towards more complex scenario's. Secondly, HSP requires a lot of compute to train. The authors needed to train 36 biased self-play policies and 18 MEP policies for each of the 5 layouts they evaluated (for a total of $54 * 5 = 270$ agents) before then training one adaptive agent for each layout.

Our approach extends the ideas present in playing against diverse populations realized in FCP, MEP and HSP with training a best response Theory of Mind agent instead of standard best response training.

3.2 Machine Theory of Mind

We have already alluded to the importance of Theory of Mind for human collaboration. To recap, ToM refers to the ability of an agent to reason about the preferences, mental states and goals of other agents in a system [BJST17; BLF85; HS44; PW78; YDF08].

As this ability is innate to humans and their behavior, an increasing effort has recently also been placed into studying Theory of Mind with and in machines, in order to give them similar capabilities. [RPS+18] took a deep learning perspective and formulated Machine Theory of Mind (MToM) as a meta-learning task in which an observer is tasked with reasoning about the actions of agents in a grid-world setting. To do so, they design a ToM neural network, ToMnet, which has inspired others to include ToM modelling in their deep learning network architectures [BCC21; FWCL21; LZL+23; NNL+22; NNL+23; SNB22; YSP+22; ZZH+23]. Except for modelling ToM via deep learning other works take a Bayesian perspective [BJST17; KHA+16] or base their approach on partially observable Markov decision processes (POMDP) [DQGY10; HG18].

More specific to this work, MToM was also used to improve human-AI coordination, most notably in Hanabi [FWCL21]. Within Overcooked MToM was explored as an inductive biases for improving a human model [KCD+21] to be used during training. Their modeling of ToM essentially boils town to a handcrafted rule-based scripted agent that tries to act more human-like by for instance having some chance of stopping to simulate a "thinking" human in the environment etc.

While not in Overcooked, Yuan et al. [YFZ+21] developed and studied a Q-learning algorithm that proposes to learn to solve auxiliary ToM tasks during training for improving cooperation across two environments. Their algorithm took had both agents estimate belief and belief over belief over each other during training. To have the labels be stationary, they updated their agents in turn where switching was defined via a small probability. This alleviates the problem of non-stationary which is a common assumption in algorithms [PJB20].

3.3 Auxiliary Tasks in Reinforcement Learning

In auxiliary tasks are sometimes used to support the agents training and can be viewed as a special variant of adding supplementary cost functions [SMD+11]. The idea often being that the learning should be additionally guided through introducing additional objective. These objectives are often based on the environment. As an example Lample and Chaplot [LC17] tasked agents in first-person shooter games to detect whether and enemy was in sight and found this to increase game performance and Mirowski et al. [MPV+17] had agents perform depth prediction during maze solving. Clearly, detecting enemies makes the task of winning such a game much easier which is why this was introduced to guide training. Other examples include reward-prediction and pixel- and feature-control all proposed by Jaderberg et al. [JMC+17]. Note that these two, along with terminal prediction [KHT19], are notable examples of auxiliary tasks that are independent of the environment and thus do not need to be hand designed.

During collaboration we would guide our agents to pay attention to the mental states of those surrounding them. Thus, core to our idea is that our agents solves auxiliary Theory of Mind tasks during play such as predicting the other agents next action. We model this and the corresponding update on the literature from auxiliary task RL that is, we add these tasks as supplementary cost functions.

3.4 AI Robustness & Safety

Research in machine learning should not only care about developing agents that can achieve the highest score in some cooperation task with a human but we would also like these agents to be *safe*. Thus we should give serious consideration towards risks, especially such related to accidents [AOS+16; Bos14]. Since humans are not idealized rational agents [TK74] any cooperation agent must specifically also account for unexpected behaviour, thus requiring them to be *robust* under a wide distribution of human actions and policies. One strategy is value alignment [FGH+20], a technique that hopes to ensure that the artificial agents perspective match these of their human counterpart. Such an agent would then be denoted as being *aligned*. Some misalignment's have already been observed in research, for instance goal misgeneralization [LKS+22]. In this regard, ToM is a crucial capability to have in cooperation since it allows an agent to take into account the preferences, goals and desires of their partner before making a decision. Our work builds on this view as we propose and evaluate ToM as a robustness measure in Overcooked.

4 Method

4.1 Preliminaries

Following Sclar, Neubig, and Bisk [SNB22] we define the Theory of Mind Multi Agent Reinforcement Learning (ToM-MARL) paradigm as an extension to the usual multi agent setting in which actions are picked by each agent conditioned on their belief of the mental state of the other agents. We start by defining a N-player Markov Decision Process (MDP) as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}^{(i)}, \mathcal{T}, \mathcal{P}, \gamma, \mathcal{R} \rangle$ where $\mathcal{S}$ is a set of states in the environment, $\mathcal{A}^{(i)}$ the set of possible actions for any given player $i \in N$ where N is the number of players in the set of players $\mathcal{N}$, $\mathcal{T}$ is the transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A}^{(1)} \times ... \times \mathcal{A}^{(n)} \rightarrow \Delta(\mathcal{S})$ which represents the distribution over next states, $\mathcal{P}$ the initial state distribution, γ the discount factor and $\mathcal{R}$ the reward function producing a joint reward for all players in $\mathcal{N}$, specifically $\mathcal{R} : \mathcal{S} \times \mathcal{A}^{(1)} \times ... \times \mathcal{A}^{(n)} \rightarrow \mathbb{R}$. In this setting, agents pick an action according to their policy $\pi_i, i \in N$ where $\pi_i : \mathcal{S} \rightarrow \Delta(A^{(i)})$. Additionally, we define the *history* as the past interactions of all agents with the environment, captured by $(\mathcal{S} \times \mathcal{A}^{(1)} \times ... \times \mathcal{A}^{(n)})^t$.

We define an agent with no ToM capabilities as one that simply acts based on their policy and their current observation. That is, given their policy π_i they choose the next action at timestep t as $\pi_i(a_{(i,t)}|s_t)$. Agents that can reason over their own knowledge and thus are stateful are denoted as *zeroth order* ToM agents [FVHK08; HZ02]. These agents pick an action given their policy π_i and their hidden state $h_t^{(i)}$ accumulated over the episode, i.e. via $\pi_i(a_{(i,t)}|h_t^{(i)})$ where $h_t^{(i)}$ also includes a representation of the current state s_t. Since both are not reasoning about mental states of others, we will refer to both kinds of agent as having no ToM. ToM agents are defined as agents capable of estimating $h_t^{(j)}, j \neq i$. We believe this definition to be to restrictive and instead define a ToM agent as any agent that reasons over (future) actions, goals or mental states of others. Our definition is more general than the one of Sclar, Neubig, and Bisk [SNB22]. Especially since ours also allows reasoning over the mental states of others implicitly, for instance as was done by Rabinowitz et al. [RPS+18].

Notice that if all other policies except for the i-th are fixed, we can reduce this paradigm to a single-agent partially observable Markov Decision Process (POMDP), treating all other agents as part of the environment. In this setting, the task reduces to finding one optimal policy $\pi^{(i)}$ in this Partially observable Markov Decision Process (POMDP) as all others $\{\pi^{(j)}|i \neq j\}$ are given.

4.2 Assigning Mental States to Artificial Agents in Overcooked

In order to predict mental states of other artificial agents in Overcooked one has to define the notion of mental state of an artificial agent, as there is no ground truth available. Consider the example in Figure 4.1, where a yellow agent approaches a tomato. Suppose that we have the full trajectory of the agent given as:

$$\tau = [\texttt{right}, \texttt{right}, \texttt{right}, \texttt{interact}(\texttt{tomato})].$$

What is the mental state of the agent given τ and how can we represent it such that other agents can predict it? We propose three ways of representing mental states in Overcooked.

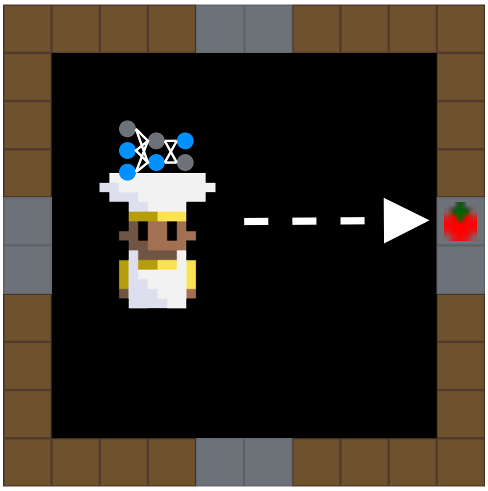

Figure 4.1: What is the mental state of this yellow agent given the trajectory outlined by the arrow? We propose three options: (i) Its next action (`right`), (ii) its strategic goal (`interact(tomato)`) or (iii) some activation vector in its policy network.

First, we pose that the mental state of the artificial agent corresponds to the low-level action it is about to perform. To this end recall that at each step the agent picks an action to perform, thereby *revealing their preference* about what to do given their current evaluation of the environment state. Consequently, we name this representation the *low-level representation*. Note, that this is akin to the theory of revealed preferences [Var06] in economics. Second, we propose that on a more abstract level the mental state of the agent is to pickup the tomato. We will from now on refer to as the *strategic representation*. This is inspired by the fact that humans seldomly reason about minute details of their motions but rather think about the goal they want to reach next. Since reasoning about the goals others is a classical ToM capability, we are interested in specifically testing this higher-level representation. Thirdly, we note that the actual state of the agent is captured in the activation's of its policy network, to which we have full access to during training. We thus suggest a *neural representation* of mental states where we task the agent with predicting the activations in a layer of the policy network of the other agent. This is different from how humans reason about each others' mental state as we never predict how the neurons of those around you will fire. This third approach resembles more closely the original formulation of ToM agents given by [SNB22]. Note that we do not argue that one of these representations is necessarily correct but instead we hypothesize that having an artificial agent reason about these will be helpful during cooperation.

4.3 Self-Supervised Belief Prediction in Overcooked

In this section we illustrate how mental state labels are computed during training. Reinforcement learning is often split into two stages (i) rollout and (ii) agent updating. During rollout, samples are collected into a dataset $\mathcal{D} = \{\tau^1, \tau^2 \ldots, \tau^D\} = \{(s_t^k, a_t^k, r_t^k)_{t=0}^T\}_{k=1}^D$ of length D which is then used to perform learning with. Note that it is trivial to expand this dataset after each episode, post-hoc, before any learning occurs. This makes it possible to compute the belief labels after each episode and obtain: $\mathcal{D} = (\bar{a}, \bar{g}, \bar{n}, s_t^k, a_t^k, r_t^k)_{t=0}^T$ where $\bar{a}, \bar{g}$ and $\bar{n}$ are belief labels at timestep t for other agents next action $\bar{a}$, their strategic goal $\bar{g}$ and their neural representation $\bar{n}$. For the low-level representation collecting the actions taken during the episode is sufficient to compute $\bar{a}$, the same holds for storing activation vectors from both policy networks for $\bar{n}$. When it comes to the strategic goal $\bar{g}$, we are inspired by the work of Yu et al. [YGL+23] and use environment events as our basis for strategic goal labels. Examples of these events include picking up an onion, picking up a tomato, delivering a soup etc. At any time step t the belief label then is the next game event triggered by the other agent. For example, the belief labels for the scene discussed above and shown in Figure 4.1 are: $\{(\texttt{r}, \texttt{i(t)}, \alpha^h_{t=0}), (\texttt{r}, \texttt{i(t)}, \alpha^h_{t=1}), (\texttt{r}, \texttt{i(t)}, \alpha^h_{t=2}), (\texttt{i}, \texttt{i(t)}, \alpha^h_{t=3})\}$ with $\texttt{r} = \texttt{right}$, $\texttt{i(t)} = \texttt{interact(tomato)}$ (a game event), and $\alpha^h_{t=0}$ an activation vector from the yellow agent's policy network.

4.4 Theory of Mind Agent Training

Because of its simplicity and recent successes we base our objective function on the Principal Policy Optimisation (PPO) [SWD+17] objective. Because solving ToM tasks during collaboration can be viewed through the lens of auxiliary task reinforcement learning, we are inspired by their methods for finding a joint objective function. In auxiliary task reinforcement learning, additional cost functions are often added to the reinforcement learning objective via a weighted sum of additional cost functions

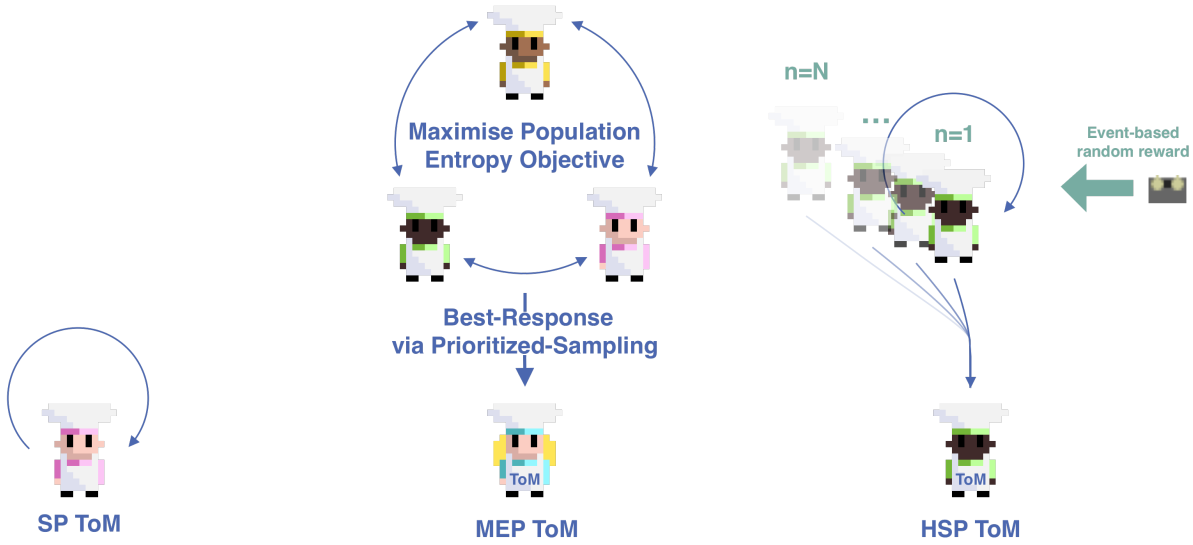
Maximise Population Entropy Objective
Best-Response via Prioritized-Sampling
n=N
...
n=1
Event-based random reward
ToM
ToM
SP ToM
MEP ToM
HSP ToM

Figure 4.2: Overview over our Theory of Mind agent training variants.

[SMD+11]. In this work we take the same approach. Assuming a suitable loss function for partner's next action prediction $L_{\bar{a}}$, partner's next strategic goal $L_{\bar{g}}$ and partner's neural state prediction $L_{\bar{n}}$ we define the total loss as a weighted sum, given weights $\lambda_{\bar{a}}$, $\lambda_{\bar{g}}$, $\lambda_{\bar{n}}$:

$$L_{\text{ToMPPO}} = L_{\text{PPO}} + \lambda_{\bar{a}} L_{\bar{a}} + \lambda_{\bar{g}} L_{\bar{g}} + \lambda_{\bar{n}} L_{\bar{n}}. \tag{4.1}$$

As other next action and other next strategic goal prediction are classification tasks, the respective natural choice for the loss function is cross-entropy

$$\mathcal{H}(p,q) = -\sum p(x)\text{log}q(x) \tag{4.2}$$

whereas for neural state prediction we choose Mean Squared Error (MSE) as it is formulated as a regression task, i.e.:

$$\text{MSE}(y,\hat{y}) = (y-\hat{y})^2. \tag{4.3}$$

Given that we base our implementation for theory of mind agent training on PPO, we combine the auxiliary losses with the outlined PPO objective. We first compute the losses for all the auxiliary objectives for mental state training, as introduced above. Given the parameters of the actor network θ_{a_j} for agent j we first compute the individual losses:

$$L_{\bar{g}_j} = \frac{1}{|\mathcal{D}|T} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^{T} \mathcal{H}(g_{\theta_{a_j}}(s_t), (\bar{g}_j)_t), \tag{4.4}$$

$$L_{\bar{a}_j} = \frac{1}{|\mathcal{D}|T} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^{T} \mathcal{H}(a_{\theta_{a_j}}(s_t), (\bar{a}_j)_t), \tag{4.5}$$

$$L_{\bar{n}_j} = \frac{1}{|\mathcal{D}|T} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^{T} \text{MSE}(n_{\theta_{a_j}}(s_t), (\bar{n}_j)_t). \tag{4.6}$$

In this formulation we assume $\mathcal{D}$ to be the dataset of trajectories where T is the time-steps per trajectory (in Overcooked $T = 400$ commonly). Additionally, we assume three functions that are additional heads of the policy network for computing the mental states of the other agent(s). Thereby at time-step t agent j computes $\hat{g}_j = g_{\theta_{a_j}}(s_t)$, $\hat{a}_j = a_{\theta_{a_j}}(s_t)$ and $\hat{n}_j = n_{\theta_{a_j}}(s_t)$. Given the auxiliary losses $L_{\bar{g}}$, $L_{\bar{a}}$ and $L_{\bar{n}}$, we then compute the clipped loss for PPO:

$$L_{\text{CLIP}_j}(\theta_{a_{j_k}}) = -\frac{1}{|\mathcal{D}|T} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^{T} \min\left(r_t(\theta_{a_j})\hat{A}_t, \text{clip}(r_t(\theta_{a_j}), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right) \tag{4.7}$$

where $r_t(\cdot)$ is the ratio of the probability under the new and old policy, ϵ a hyper-parameter and $\hat{A}_t$ is the estimation of the advantage function, i.e as described by Schulman et al. [SWD+17]:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}, \tag{4.8}$$

$$\text{where } \delta_t = r_t + \gamma V(s_{t+1})V(s_t), \tag{4.9}$$

In total we thus compute the update to obtain $\theta_{a_{j_{k+1}}}$ as:

$$L_{\text{ToMPPO}} = L_{\text{CLIP}} + \lambda_{\bar{a}} L_{\bar{a}} + \lambda_{\bar{g}} L_{\bar{g}} + \lambda_{\bar{n}} L_{\bar{n}} \tag{4.10}$$

$$\theta_{a_{j_{k+1}}} = \text{arg min}_{\theta_{a_{j_k}}} L_{\text{ToMPPO}}. \tag{4.11}$$

Conversely, we do not change the update to the value function update and stick to the original [SWD+17]. That is given the value function parameters $\theta_{v_{j_k}}$ we use:

$$\theta_{v_{j_{k+1}}} = \text{arg min}_{\theta_{v_{j_k}}} \frac{1}{|\mathcal{D}|T} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^{T} (V_{\theta_{v_{j_k}}}(s_t) - \hat{R}_t)^2 \tag{4.12}$$

Given this combined loss function we adopt two algorithms for training our ToM agent. The first algorithm is inspired by Yuan et al. [YFZ+21]. In their work they amended standard self-play with ToM auxiliary tasks. This showed better performance on several multi-agent benchmarks than other self-play techniques. Their work is different from ours in two important ways: (i) they are only interested in self-play performance and not zero-shot performance and (ii) they only evaluate on environments where ground truth beliefs exist and therefore do not need to compute labels from the dataset alone. Still, our resulting algorithm is consequently denoted *Self-Supervised Self-Play Theory of Mind Training* or *ToM Self-Play* for short. Recall that self-play fails to solve the zero-shot coordination problem due to its implicit assumption of playing with a partner that is similar to oneself. Therefore, while we would expect ToM Self-Play to outperform standard self-play we have to assume that it will also fail the zero-shot coordination problem for the same reasons. Thus, we further take inspiration from previous research on zero-shot coordination in Overcooked. Most successful methods, i.e. FCP, MEP and HSP, train an adaptive best response agent against a population of diverse agents to prepare their final agent for diverse behaviour. Consequently, we denote our algorithm as *Self-Supervised Adaptive Best Response Theory of Mind Training* or *Best Response ToM* for short.

4.4.1 Self-Supervised Self-Play Theory of Mind Training

We outline our Self-Play ToM training in Algorithm 4.1. In short, our algorithm trains two separate policies using PPO in self-play while both agents solve the auxiliary tasks of predicting the other agent's next action, strategic goal and neural representation.

This algorithm differs from standard self-play in three important ways. First, we introduce auxiliary losses that produce predictions about mental states of the other agents. The labels for these losses are computed on the fly in a self-supervised manner from the collected trajectories before adding it to the dataset $\mathcal{D}$. Second, note that we always only update one of the two agents at any given time. This is to avoid the problem of non-stationary as is also done by Yuan et al. [YFZ+21]. Essentially, if we were to update both agents at the same time the mental states labels would resemble a moving target which makes convergence for most learning algorithms impossible. We switch the agent being updated at random given the probability r (also inspired by [YFZ+21]). This way both agents predict the mental states of each other. Third, we set $\lambda_{\bar{a}}$, $\lambda_{\bar{g}}$, $\lambda_{\bar{n}}$ to 0 for the first ten epochs since it is not useful to predict the mental state of an agent that is not capable.

Algorithm 4.1 PPO Self-Supervised Self-Play Theory of Mind Training.

```
Require: pre-trained population $\mathcal{P}$, initial actor function parameters $\theta_{a_0}$ , initial value function parameters $\theta_{v_0}$, other neural state head $n(\cdot)$, other next action head $a(\cdot)$, other strategic goal head $g(\cdot)$, loss balancing factors $\lambda_{\bar{a}}$, $\lambda_{\bar{g}}$, $\lambda_{\bar{n}}$, agent indices $N = \{i|i \in 0, 1\}$, initial agent to update index $j \in N$, probability to switch updating $p_s$, number of iterations $i$, number of PPO epochs $k$, minibatch size $M$, $\epsilon$
  for 1, 2, ... in $i$ do
      $\mathcal{D} = \{\}$
      repeat
          repeat
              Agents sample actions $a_l$ according to their policy $\pi_{k_l} = \pi(\theta_{a_{k_l}})$
              Agents observe the next state $s_{t+1}$ and the reward $r_t$
              Agents add these to the trajectory $\tau$
          until game ends
          Compute $\{\bar{g}_l\}_{0:T}$, $\{\bar{a}_l\}_{0:T}$ and $\{\bar{n}_l\}_{0:T}$ as labels for agent $l$ from $\tau$
          Update $\mathcal{D}$ with new trajectory $\tau'$: $\{s_t, a_t, r_t, (\bar{g}_l)_t, (\bar{a}_l)_t, (\bar{n}_l)_t\}_{0:T}$
      until number of games reached per epoch
      Switch agent index $j$ with probability $p_s$
      Optimize surrogate $L_{\text{ToMPPO}}$ w.r.t. $\theta_{a_{j_k}}$ with $k$ epochs and minibatch size $M$
  end for
```

4.4.2 General Self-Supervised Adaptive Best Response Theory of Mind Training

To solve the known issues with self-play, we introduce our second and ultimately final Algorithm 4.2. The algorithm takes as input a population of pre-trained agents and produces a best response ToM agent trained on this population. Our ToM agent predicts the same mental states as the self-play variant but importantly the population does not posses any explicitly modeled theory of mind capabilities. This gives the best response theory of mind training the interesting property of being usable with many previously introduced SOTA method as they mostly differ in their way of generating and sampling from the pre-trained population. We thus can freely choose how to collect a diverse population of agents and also how to sample agents from it. In particular, we train our population using Hidden-Utility Self-Play and Maximum Entropy Population Based Training, which we will detail in the following. Note that we do not experiment with FCP as MEP is an improved version of FCP.

Algorithm 4.2 PPO Self-Supervised Adaptive Best Response Theory of Mind Training.

```
Require: pre-trained population $\mathcal{P}$, initial actor function parameters $\theta_{a_0}$ , initial value
  function parameters $\theta_{v_0}$, other neural state head $n(\cdot)$, other next action head $a(\cdot)$, other
  strategic goal head $g(\cdot)$, loss balancing factors $\lambda_{\bar{a}}$, $\lambda_{\bar{g}}$, $\lambda_{\bar{n}}$, agent indices $N = \{i|i \in 0, 1\}$,
  number of iterations $i$, number of PPO epochs $k$, minibatch size $M$, $\epsilon$
  for 1, 2, ... in $i$ do
      $\mathcal{D} = \{\}$
      repeat
          for $\pi_p$ ... in $\mathcal{P}$ do                                   // Pre-trained agents
              repeat
                  Agents ($\pi_p$, $\pi_k = \pi(\theta_{a_k})$) sample actions $a_t$ according to their policy
                  Agents observe the next state $s_{t+1}$ and the reward $r_t$
                  Agents add these to the trajectory $\tau$
              until game ends
              Compute $\{\bar{g}_l\}_{0:T}$, $\{\bar{a}_l\}_{0:T}$ and $\{\bar{n}_l\}_{0:T}$ as labels for agent $\pi_k$ from $\tau$
              Update $\mathcal{D}$ with new trajectory $\tau'$: $\{s_t, a_t, r_t, \bar{g}_t, \bar{a}_t, \bar{n}_t\}_{0:T}$
          end for
      until number of games reached per epoch
      Obtain $\theta_{a_{(k+1)}}$ by optimizing Eq. 4.10 w.r.t. $\theta_{a_k}$ with $k$ epochs and minibatch size
  $M$ using $\mathcal{D}$
  end for
```

4.4.3 Best Response ToM with Maximum Entropy Population Based Training

MEP trains a population of diverse agents based on a maximum-entropy objective [HZAL18]. The idea is to find a population of agents that are encouraged to take different actions from each other while still performing well. This is achieved by an additional entropy term in the objective. Following Zhao et al. [ZSY+23]:

$$J(\bar{\pi}) = \sum_t \mathbb{E}_{(s_t,a_t)\sim\bar{\pi}} \left[R(s_t, a_t) + \alpha \mathcal{H}_{\text{MEP}}(\bar{\pi}(\cdot|s_t))\right] \tag{4.13}$$

Here, $\mathcal{H}_{\text{MEP}}(\bar{\pi}(\cdot|s_t))$ refers to the population entropy (PE). Specifically,

$$\text{PE}(\{\pi^{(1)}, \pi^{(2)}, ..., \pi^{(n)}\}) := \mathcal{H}_{\text{MEP}}(\bar{\pi}(\cdot|s_t)), \text{ where } \bar{\pi}(a_t|s_t) = \frac{1}{n} := \sum_{i=1}^{n} \pi^{(i)}(a_t|s_t). \tag{4.14}$$

Within $J(\bar{\pi})$ the parameter α balances the task reward with the population entropy term. To optimise $J(\bar{\pi})$ agents are sampled uniformly from the population and tasked with self-play. Given a trained population Zhao et al. [ZSY+23] in turn then trains a best response agent by sampling the hardest to play against agents first using a process called Prioritized Sampling. This allegedly makes sure that the agent does not exploit weaker members of the population for high reward while not learning to play against harder opponents. For details please refer to the original work of Zhao et al. [ZSY+23]. We use the MEP implementation of Yu et al. [YGL+23] which performs no prioritized sampling as their results show that it is not necessary. This method will be referred to as *MEP ToM*.

4.4.4 Best Response ToM with Hidden-Utility Self-Play

While MEP is based on the observation that a population needs to be diverse and capable for obtaining a best response agent capable of zero-shot cooperation, Yu et al. [YGL+23] makes the important observation that for zero-shot human-AI cooperation the population should reflect human traits. In their work they make the observation that humans are acting non-optimally and are biased. They for instance might have a preference for tomatoes over onions. So, assuming humans are biased, they instead set out to optimize a hidden utility Markov game in which next to the task reward R_t an additional hidden reward R_ω exists. Based on this Yu et al. [YGL+23] introduce HSP by having to agents (π_a, π_ω) maximize the different rewards R_t and R_ω respectively where R_ω is only observable to π_ω. Yu et al. [YGL+23] argue that, in Overcooked, human preferences are event-centric and they thus design their reward function as linear functions over Overcooked game events. Specifically, $\mathcal{R} = \{R_\omega : R_\omega(s, a_1, a_2) = \phi(s, a_1, a_2)^T, ||\omega||_\infty \leq C_{\text{max}}\}$ where C_{max} is an upper bound on the weight ω, $\phi : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to \mathbb{R}^m$ is a function specifying occurrences of game events and ω is drawn randomly for N agents $\{\omega^{(i)}_{i \in N}\}$ based on a set of predefined values C_j. Essentially, ϕ specifies the game events happening while taking a joint action (a_1, a_2) in state s and ω describes the reward that is given to the agent for triggering the game event. As ω is drawn randomly, it is ensured that different agents receive different rewards for game events, thus biasing them differently. In practice this leads to a set of N hidden reward functions used to train a population of self-play agents in which one of the two agents receives task and hidden rewards. This population of hidden reward agents is then used to train a best response agent. For details see Yu et al. [YGL+23]. In the rest of this work, we will refer to this method as *HSP ToM*.

4.5 Theory of Mind Agent Implementation

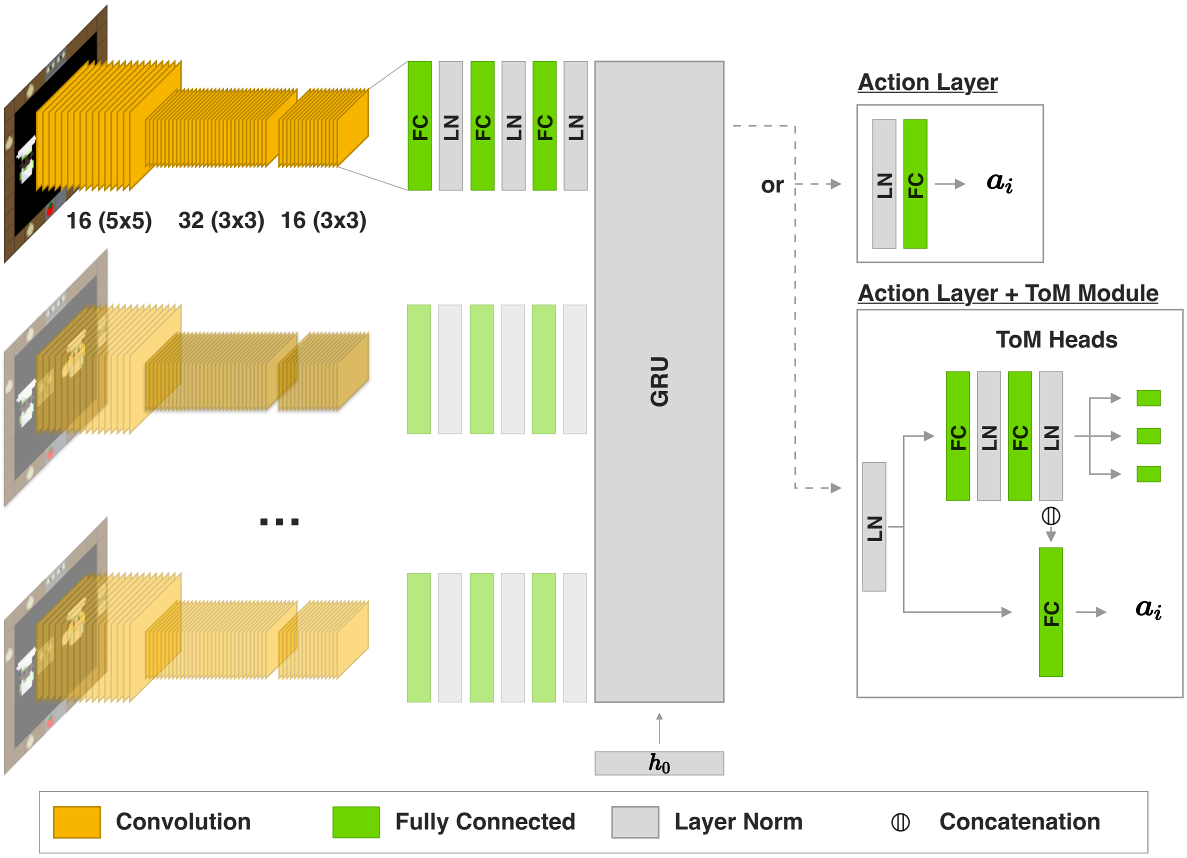

Figure 4.3: Overview over our actor neural network architecture. The actor encodes the state of the environment using a convolutional neural network as a feature encoder. These features are then aggregated across time via a recurrent neural network before being used to choose a next action and possibly perform ToM tasks.

We train our agents using an actor-critic reinforcement learning approach [KT99]. Our base actor consists of an feature extractor that encodes the state via a Convolutional Neural Network (CNN) based feature extractor which then adds temporal information before passing the final hidden state to the heads, see Figure 4.3. More specifically our actor picks an action a_t at time step t given a representation of the current state of the environment s_t by encoding the state into a state embedding x_t^e:

$$x_t^e = \texttt{LN}(\texttt{GRU}(\texttt{FC}_{\texttt{CNN}}(\texttt{CNN}(s)))). \tag{4.15}$$

Here $\texttt{FC}_{\texttt{CNN}}$ is a stack of three blocks of a single Fully Connected (FC) layer followed by Layer Normalisation (LN) [BKH16] each. Temporal information is also added, up to 100 previous steps in the episode via a Gated Recurrent Unit (GRU) [CMBB14]. a_t is then picked from a final head given x_t^e according to

$$P(a_t|s) \propto \exp(\texttt{FC}(x_t^e)). \tag{4.16}$$

This process is depicted in the 'Action Layer' section in Figure 4.3.

To implement the previously described approaches of encoding mental states in Overcooked, we add up to three additional prediction heads for explicitly adding ToM to our model. Moreover, since we want the reasoning about the mental state to affect the choice of action, we also alter the architecture of picking an action. To be exact instead of predicting $P(a_t|s)$, we now estimate $P(a_t|s_t, m_t^e)$ where m_t^e is a mental state embedding. Given x_t^e, our actor computes

$$P(a_t|s_t, m_t^e) \propto \exp(\texttt{FC}(x_t^e||m_t^e)) \tag{4.17}$$

where || represents concatenation and m_t^e is the embedding before the ToM heads

$$m_t^e = \texttt{FC}(\texttt{LN}(\texttt{FC}(x_t^e)) \tag{4.18}$$

The partner's next action $\bar{a}_t$ is computed as

$$P(\bar{a}_t|m_t^e) \propto \exp(\texttt{FC}_{\bar{a}}(m_t^e)). \tag{4.19}$$

Analogously, the agent also estimates the partner's strategic goal $\bar{g}_t$ using $\texttt{FC}_{\bar{g}}$ and its neural state $\bar{n}_t$ as $\texttt{FC}_{\bar{n}}$. A visual representation of this process is also depicted in the 'Action Layer + ToM Module' in Figure 4.3.

4.5.1 Implementation & Training Details

The implementation accompanying this work is an extension of the publicly available implementation of HSP from Yu et al. [YGL+23]. We specifically base our implementation on their work as they hold the current state-of-the-art results on which we built ours.

Our CNN-based feature encoder compromises three convolutional blocks with 16 5×5, 32 3×3, and 16 3×3 filters, respectively. The output of this encoder is flattened and passed to a stack of fully connected layers, finally resulting in an 64 dimensional vector which is then passed into a GRU. We use ReLU [Aga18] as the activation function. The networks are optimised using the Adam [KB14] optimizer using hyper-parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon_{\text{Adam}} = 0.00001$. For balancing the losses we choose $\lambda_{\bar{a}} = \lambda_{\bar{g}} = \lambda_{\bar{n}} = 0.1$. For self-play based training for the populations we run a variation of PPO that is optimised for multi-agent systems, named multi-agent PPO [YSP+22] in which the value function is shared between agents during training. While training our agents retrieve shaped reward for certain game events that linearly decreases as training progresses, exactly as done by Yu et al. [YGL+23]. In terms of PPO hyper-parameters we use $k = 15$ PPO epochs, $\epsilon = 0.2$ and mini-batch size $M = 120000$ for adaptive best response training and $M = 40000$ for self-play ToM. Additionally, we set $\gamma = 0.99$ and $\lambda = 0.95$. For self-play ToM training we use $p_s = 0.2$, i.e. we switch the agent being updated 20 percent of the time. For all details, see the implementation.

5 Experiments

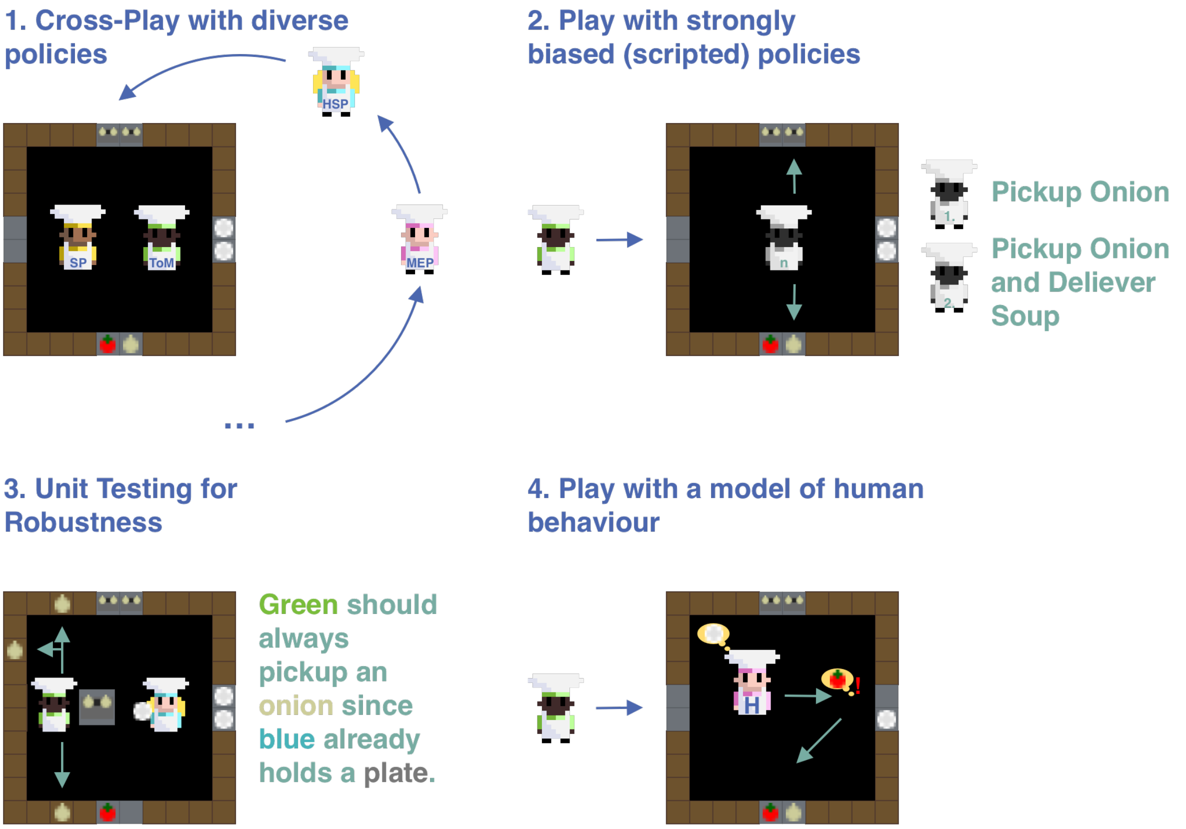
1. Cross-Play with diverse policies
HSP
SP
ToM
MEP
...
2. Play with strongly biased (scripted) policies
n
1.
Pickup Onion
2.
Pickup Onion and Deliever Soup
3. Unit Testing for Robustness
Green should always pickup an onion since blue already holds a plate.
4. Play with a model of human behaviour
H
!

Figure 5.1: Overview over the proposed four step evaluation pipeline. Cross-play [HLPF20; SMB+21], strongly biased policies [YGL+23], unit testing for evaluating robustness [KCD+21], models of human behaviour [CSH+19; KCD+21].

Typically, reinforcement learning agents are evaluated on the environment they have been trained with (i.e. the training distribution) using the average validation reward. This is not a good signal of future performance with other agents as their behaviour often substantially differs from the agents encountered during training. Thus testing and evaluating the performance of an agent that is supposed to collaborate well with novel agents is non-trivial. This is due to the fact that at train time we have no access to the agents we want to cooperate with in the future as they are unknown. In the special case of human-AI cooperation, evaluating with humans is especially expensive and cumbersome. Even if one is willing to evaluate with humans, one typically is still very interested in checking whether their method has a chance of performing well before conducting a user study. As an alternative or to mitigate this issue, we propose to combine evaluation ideas from literature, especially Cross-Play with diverse agents [HLPF20; SMB+21], pairing with biased policies [YGL+23], unit testing for robustness [KCD+21] and evaluating against human models [CSH+19; KCD+21]. We give an overview over our evaluation approach in Figure 5.1.

We pick these because we think that they together form a complete and human-free evaluation suite for robust cooperation. Hu et al. [HLPF20] notes that Cross-Play with diverse agents is a necessary condition for zero-shot cooperation with humans which makes it an obvious first candidate for tracking the performance of our agent. Biased policies moreover test whether an agent can cooperate with extreme forms of human behaviour which as Yu et al. [YGL+23] notes is also biased. Unit testing tests states and agent behaviour that is unlikely to be encountered even when trained with diverse agents and makes sure that any agent can also deal with situations unfamiliar to them. Such states might be encountered when playing with a human that is still figuring out the environment, unsure about the objective or exploring. Lastly, some works have build models of human behavior with [CSH+19] and without human data [KCD+21] which tests our agent against a human proxy.

5.1 Evaluation Layouts

We evaluate our approach on two Overcooked layouts: Asymmetric Advantages and Many Orders, see Figure 5.2. Asymmetric Advantages is a relative simple onion-only layout that has some easily noticeable failure cases that are interesting for analysis. We have presented this layout earlier also. Note that Asymmetric Advantages gives the left agent an advantage to deliver cooked soups as the soup serving location is closer to the pots and the right agent an advantage in putting onions into pots. This both tests whether agents can choose a strategy fitting their strategic advantage as well as if they can divert from it and still play a sub-optimal one given a partner does not play a suitable strategy.

Many Orders on the other hand allows many possible near-optimal strategies in a tight layout with onions and tomatoes and several possible soup recipes with different cooking times. Here, agents need to be careful to be cooking the same recipes and to not block each other during gameplay. It thus tests whether two agents can agree on any particular strategy to achieve the highest reward.

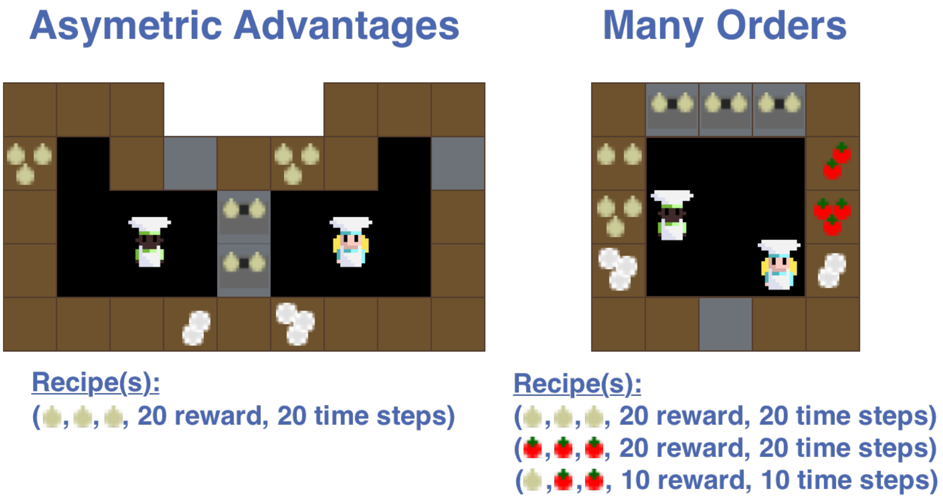
Asymetric Advantages
Many Orders
Recipe(s):
(🧅, 🧅, 🧅, 20 reward, 20 time steps)
Recipe(s):
(🧅, 🧅, 🧅, 20 reward, 20 time steps)
(🍅, 🍅, 🍅, 20 reward, 20 time steps)
(🧅, 🍅, 🍅, 10 reward, 10 time steps)

Figure 5.2: Our approach is evaluated on two Overcooked layouts: Asymmetric Advantages and Many Orders.

5.2 Cross-Play Evaluation

Firstly, an agent that is supposed to cooperate well with others should in general perform well with as many different agents as possible. In many works in the literature thus but especially in [HLPF20; SMB+21], agents are evaluated by pairing them with other agents either from the literature or that have been differently trained. Such agents form a natural evaluation population as they likely behave differently to test agent being tested. The approach is denoted *cross-play*, contrary to self-play. In previous work such as [HLPF20], the best performing cross-play agent also achieved the highest score with humans in a user study. While [HLPF20] only evaluates cross-play between agents trained with the same method but different random seeds we instead evaluate cross-play between different training methods. Since different training methods produce agents far more different from one-another we see this as the more realistic and challenging evaluation. During cross-play we also keep track of the *other-play* score which is simply the average reward being observed when being paired with all other agents, i.e. excluding playing with yourself. A perfectly generalized agent should show similar levels of self-play and other-play performance.

There are several reasons why while this form of testing is informative, it is not *sufficient*. In general the available population is small and the members are often quite capable as it is built from published methods in the literature. Playing well with this population thus gives no insight into how it will perform with less capable or even adversarial agents. Additionally, humans tend to be biased in the way they play [YGL+23], requiring special attention at evaluation time.

5.3 Strongly Biased Policies

Thus, secondly as in the work of [YGL+23] we also evaluate our agent against *strongly biased policies*. In our case these policies are scripted and stubbornly execute the same behavior regardless of their cooperation partner and thus require the partner to adapt to their behavior. The two policies being evaluated on Asymmetric Advantages are: (i) 'Pickup Onion' and (ii) 'Pickup Onion and Deliver Soup'. (i) only takes onions from its dispenser and places them in the soup and (ii) will additionally deliver soup when the cooking has finished. As one of these biased policies does not deliver soups, failures to adopt will result in significantly lower or zero reward, signaling cooperation failure. Still, as these policies omit very predictable behaviour, the agent being evaluated is rarely placed in unexpected scenarios or edge-cases. These situations though might be more common when playing with less capable agents or agents that omit more erratic behaviour. In such cases an agent would need to recover from the situation and continue gameplay.

5.4 Unit Testing for Robustness

Consequently, thirdly we test an agent against a suite of *unit tests for robustness* as argued for by [KCD+21]. The idea of unit tests come from software engineering where parts of a program are tested in isolation for correctness. For creating such unit tests a software engineer is often tasked with coming up with edge cases a program might need to deal with to find flaws in their work. Similarly, we might think of all situations that are challenging for an agent to deal in cooperation. These usually represent cases an agent usually does not encounter during training, thus testing their ability to adopt on the fly. [KCD+21] came up with three testing categories which we give an overview of in Figure 5.3. From left to right they are (a) state robustness, (b) agent robustness and (c) agent and memory robustness tests. To form some intuition around these we describe examples in Figure 5.3. All tests are tested for with a fixed time limit and with several permutation with regards to starting positions and object locations. As [KCD+21] did not create unit tests for the layout Asymmetric Advantages nor Many Orders, we adopted their tests for our layouts. We detail the exact tests performed in the Appendix A.1. In total we perform 15 tests with 54 (Asymmetric Advantages) and 58 (Many Orders) for a total of 112 variants.

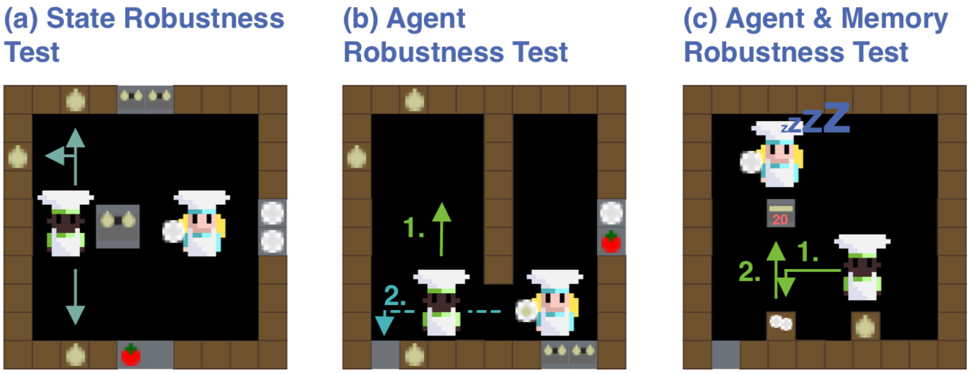
(a) State Robustness Test
(b) Agent Robustness Test
(c) Agent & Memory Robustness Test
1.
2.
zZZ
20
2.
1.

Figure 5.3: Unit testing for robustness. Content adapted from Knott et al. [KCD+21]. In the example for category (a) the green agent is tasked with realizing that their blue partner already holds a plate, thus the only optimal action is to finish cooking the soup. Note that the green agent could expect the blue agent to already have collected an onion given the state of the world and thus fail to adopt to the current state. Other state robustness tests may of instance include objects in unexpected locations or in unusual amounts. In (b) the blue agent carries a finished soup and has its mind set on delivering it to the serving location. Our green agent thus needs to move out of the way, adapting to the policy of the other agent. Note that since blue is stubborn about delivering their soup, our green agent needs to move out of the way regardless of what the optimal move might be in this situation. Lastly in the example for category (c) our green agent needs to realise that the blue agent is sleeping (not acting) even though it is in an optimal position to pickup the soup. This is usually not encountered during training and thus challenging to adapt to fast. Since the sleeping behaviour can only be noticed if the green agent reasons about several time steps this represents an agent and memory robustness test.

5.5 Pairing with Human Models

Models with human behaviour exist in Overcooked. The original work by Carroll et al. [CSH+19] learned a human model via imitation learning [AN04; HE16] - specifically behaviour-cloning (BC) [Pom91] - to obtain a human proxy model for evaluation purposes. This model depends on the availability of human-human game-play data which is an obvious disadvantage. First, game-play data needs to be separately collected for each layout one is interested in, thereby increasing costs with every layout. Second, the already collected data from [CSH+19] is only partially useful as it is only available for layouts they have evaluated in their work. Since current work also focuses on new layouts a human proxy model cannot be obtained through their method. This affects layouts on which current SOTA methods were evaluated and which we would like to evaluate our work on also. Examples for methods that were evaluated on (at least partially) different layouts are HSP and MEP.

Consequently, we have decided against human-data based methods and instead turned to a method that does not require it: the scripted Theory of Mind agent of [KCD+21]. Motivated by the fact that BC only produces human like behaviour on states the BC agent is trained on, they built a ToM inspired planning model capable of operating in the entire state space. This ToM model keeps a list of higher-level tasks to be completed and decides on one at every timestep t which in turn then is the goal for choosing lower level motion actions. The ToM model is parameterized in terms of how many tasks in its look-a-head list, whether to take the other agent into account for planning, how likely the agent takes a stay-action at every step etc. (for the full details see [KCD+21]).

Importantly these parameters model human behaviour and biases. The probability of taking a stay action for instance mimic the stopping and thinking that is sometimes observed in human play. Since this agent is planning based it behaves sensibly on the entire state space and also in layouts not originally designed for. We took the effort to port this agent to the layouts we here consider as they were not originally considered in the work of [KCD+21]. Additionally, we detail the exact parameters used in Appendix A.2.

6 Results

In this chapter, we present results for our new methods and give insight into how additional ToM auxiliary tasks can help agents during cooperation in Overcooked. The agents we evaluate are denoted by their basic training methodology and the ToM modelling employed during training, i.e. MEP$_g$ would denote and MEP based adaptive best response training with strategic goal prediction. Our evaluation focuses on three combinations of ToM auxiliary tasks, (g), (g, a) and (g, a, n). We do so as strategic goal prediction in itself is the core of what we believe most closely resembles the kind of Theory of Mind innate in human cooperation while action prediction and neural state prediction seem to be more pragmatic to us. Additionally, we previously did exploratory analysis on other combinations of auxiliary tasks - especially (a) and (n) - but found these three to be more promising. In total we evaluate SP, MEP and HSP variants based on the evaluation description above. As baselines to compare against, we train agents without auxiliary ToM tasks or use ones provided by the literature. In the following we will evaluate our method in (i) cross- and other-play, (ii) play against strongly scripted policies, (iii) unit testing for robustness and (iv) human models.

6.1 Cross- and Other-Play

	Asymmetric Advantages				Many Orders			
Model	SP	SP$_g$	SP$_{g,a}$	SP$_{g,a,n}$	SP	SP$_g$	SP$_{g,a}$	SP$_{g,a,n}$
SP	187.7	164.6	215.7	165.7	105.5	104.1	13.4	32.6
SP$_g$	159.8	192.9	107.8	280.9	100.0	109.6	14.2	31.1
SP$_{g,a}$	214.8	215.1	224.3	<u>85.2</u>	<u>4.9</u>	115.8	115.8	112.5
SP$_{g,a,n}$	305.9	118.4	107.8	**308.3**	**150.9**	35.8	88.2	129.6

Table 6.1: Exploratory self-play study showing cross-play results. Results on the diagonal are self-play. The metric measured is average validation reward averaged across 100 games. The best results are in **bold**, the worst ones <u>underlined</u> separated by layout.

	Asymmetric Advantages			Many Orders		
Model	OP	SP	OP - SP	OP	SP	OP - SP
SP	204.4	187.7	16.6	86.7	105.5	-18.8
SP_g	174.4	192.9	-18.5	48.3	109.6	-61.3
$\text{SP}_{g,a}$	157.7	224.3	-66.5	58.7	115.8	-57.0
$\text{SP}_{g,a,n}$	177.3	308.3	-130.9	75.2	129.6	-54.4

Table 6.2: Exploratory other-play study across 100 games measured by average validation reward with agents from Table 6.1. The results highlight the need for additional techniques to incorporate play with diverse training partners. Here, OP = other-play and SP = self-play.

Since we assume our self-play variants to be inferior due to them not training against a population with diverse behaviour, we analyze them first also as a proof of concept that auxiliary ToM tasks are effective and a useful signal during training. In Table 6.1 we summarize our results. Note that since in self-play two policies are trained for any run, we average their results in the Table.

Clearly, in both layouts the self-play baseline (SP) is outperformed by its ToM variants when playing with itself (note the diagonal). In fact in both layouts additional ToM auxiliary tasks improve the self-play and results are strictly ordered, where the following relation holds: SP $<$ SP_g $<$ $\text{SP}_{g,a}$ $<$ $\text{SP}_{g,a,n}$. We especially note that modeling all three auxiliary ToM tasks $\text{SP}_{g,a,n}$ yields the highest overall return of 308.3 in Asymmetric Advantages. While the highest value 150.9 in Many Orders comes from the pair ($\text{SP}_{g,a,n}$, SP), the highest reward from the diagonal also comes from $\text{SP}_{g,a,n}$ with a value of 129.6. On first inspection this comes at a cost: $\text{SP}_{g,a,n}$ performs noticeably worse with other agents. For instance together with $\text{SP}_{g,a}$ for a total reward of 85.2. We observe a similar pattern in Many Orders where both $\text{SP}_{g,a}$ and $\text{SP}_{g,a,n}$ produce worse average reward on the non-diagonal.

To analyze the play with others systematically, we calculated the other-play score and present it in Table 6.2. The other-play score for any policy is the mean over all rewards obtained from games that it played one rule in, i.e. all associated table cells excluding the diagonal. In Asymmetric Advantages we can observe that policies that were trained with ToM auxiliary tasks perform noticeably worse. This observation is reinforced by the fact that if you consider the difference between the other-play and self-play scores, one can see that additional ToM modelling increases the gap between self- and other-play and the relation between the performance of the policies *reverses* in this regard, i.e. $\text{SP}_{g,a,n}$ $<$ $\text{SP}_{g,a}$ $<$ SP_g $<$ SP. This suggests that policies with ToM modelling are increasingly competent at the game but at the cost of also being overly specialized

	Asymmetric Advantages							
Model	HSP*	HSP$_g$	HSP$_{g,a}$	HSP$_{g,a,n}$	MEP	MEP$_g$	MEP$_{g,a}$	MEP$_{g,a,n}$
HSP*	377.4							
HSP$_g$	385.3	387.0						
HSP$_{g,a}$	406.1	403.6	442.8					
HSP$_{g,a,n}$	376.5	386.6	407.9	391.4				
MEP	401.9	418.5	441.8	421.8	422.8			
MEP$_g$	400.1	394.5	442.3	418.3	426.6	435.0		
MEP$_{g,a}$	416.0	412.7	440.9	422.5	435.0	**447.8**	**454.8**	
MEP$_{g,a,n}$	415.5	420.1	**448.3**	435.1	425.5	436.1	**453.8**	**448.4**

Table 6.3: Cross-play results in the layout Asymmetric Advantages where results are averaged across positions. The metric measured is average validation reward averaged across 100 games. HSP* refers to the trained model released by Yu et al. [YGL+23]. The five best results are in **bold**, the five worst ones underlined.

towards themselves. Moreover, we observe the same effect in Many Orders where every additional ToM tasks increases the self-play performance but fail to meet the other-play performance of the SP baseline. This confirms our assumption that self-play based theory of mind modelling alone is not capable to increasing performance with other-agents. Given this and the observation of Hu et al. [HLPF20] that good cross-play results are a necessary condition for human-AI cooperation, we reason that these agents likely will not perform well during zero-shot human-AI cooperation and move on to evaluating population-based methods with ToM auxiliary tasks.

The analysis of self-play variants above indicates that auxiliary ToM tasks produce more capable models when it comes to self-play but not capable of zero-shot cooperation. This motivates taking these results and combining them with specific techniques build for collaboration in Overcooked, as introduced in Chapter 4.

We present results for both layouts in two separate tables, i.e. Table 6.3 for Asymmetric Advantages and Table 6.4 for Many Orders. In these tables we average results across positions to help with readability. Full results can be found in Appendix A.3. For Asymmetric Advantages we can again clearly establish that ToM auxiliary tasks improve competency, i.e. that they result in higher self-play performance. The results are less clear for the Many Orders layout, some MEP variants specifically seem to perform surprisingly bad during self-play, while MEP$_{g,a,n}$ at least contributes to two of the five best results and three of the five best results are better than the baseline HSP* in self-play. Overall, the results suggest that the trained models are not capable of cooperation with

	Many Orders							
Model	HSP*	HSP$_{g}$	HSP$_{g,a}$	HSP$_{g,a,n}$	MEP	MEP$_{g}$	MEP$_{g,a}$	MEP$_{g,a,n}$
HSP*	**375.9**							
HSP$_{g}$	**389.0**	199.4						
HSP$_{g,a}$	343.2	202.6	209.0					
HSP$_{g,a,n}$	**376.2**	342.8	352.1	168.9				
MEP	347.9	187.4	202.3	221.0	0.0			
MEP$_{g}$	331.3	199.1	199.3	205.0	0.0	0.0		
MEP$_{g,a}$	343.7	347.3	358.5	173.1	190.0	193.9	187.1	
MEP$_{g,a,n}$	**373.6**	361.4	**362.6**	200.2	238.4	234.4	207.4	227.6

Table 6.4: Cross-play results in the layout Many Orders where results are averaged across positions. The metric measured is average validation reward averaged across 100 games. HSP* refers to the trained model released by Yu et al. [YGL+23]. The five best results are in **bold**, the five worst ones underlined.

themselves. Instead, they seem to be only capable of cooperating with others. We will try to account for this observation later on.

While these results are interesting, overall we care about how well agents can cooperate with others. Therefore, we present other-play statistics in Table 6.5. Let us elaborate on why these other-play tables are interesting to begin with. We believe that there are two important aspects to look for when evaluating other-play: (i) we are interested in finding agents with high other-play score and (ii) we would like to have the difference in other-play and self-play performance be quite close to zero as this indicates good generalization capability. Intuitively one might suspect that a large positive difference indicates better other-play capabilities than self-play, but note that any stationary policy then would clearly win when paired with the most capable policy.

Within Asymmetric Advantages the model with the best other-play score, HSP$_{g,a}$, also has the second lowest difference in other-play and self-play scores. This suggests that the agent is not only most capable but also generalizes nicely. In terms of other-play scores HSP* performs worst by a comfortable margin while standard MEP achieves surprisingly good performance, placing it third-to-last and with the lowest difference between self- and other-play. Generally, all our methods are capable of zero-shot cooperation as the scores are significantly higher than the scores produced by self-play, as shown in Table 6.2. Similarly, all methods perform better than self-play based approaches in Many Orders. Still, the analysis from before is no longer valid here. This is to say, that HSP* - previously the worst method - achieves the highest results in Many Orders and with a significant difference, the second being HSP$_{g,a}$ which previously was the

	Asymmetric Advantages			Many Orders		
Model	OP	SP	OP - SP	OP	SP	OP - SP
HSP*	389.2	377.4	11.8	**356.6**	375.9	−19.0
HSP$_g$	415.7	387.0	28.7	250.3	199.4	50.9
HSP$_{g,a}$	**438.0**	442.8	−4.8	283.3	209.0	74.3
HSP$_{g,a,n}$	415.0	391.4	23.6	260.6	168.9	91.7
MEP	426.4	422.8	3.6	185.0	0.0	185.0
MEP$_g$	415.5	435.0	−19.5	254.2	0.0	254.2
MEP$_{g,a}$	422.1	454.8	−32.7	181.7	187.1	−5.4
MEP$_{g,a,n}$	432.4	448.4	−16.0	271.5	227.6	43.9

Table 6.5: Other-play results across 100 games measured by average validation reward with agents from Table 6.3 and 6.4. As other-play score we again compute the mean over the row and column for any relevant policy, excluding the diagonal. In terms of notation OP is other-play and SP is self-play. The best results are in **bold**, the worst ones underlined.

best one. Recall, that HSP* is provided by Yu et al. [YGL+23] and trained by them with a population they trained. While Yu et al. [YGL+23] provided us with their HSP* agent, they did not provide us their exact training population. This means that the exact population they have used is not available to use and our methods thus were trained with a different population that was obtained the same way. Reinforcement Learning generally is sensitive to randomness and so we wonder whether a difference in training populations might account for the difference observed in performance? We will return to this question at the end of the chapter and discuss implications later.

6.2 Play with Strongly Biased Policies

To evaluate the performance of agents with (extreme) forms of human behaviour we evaluate our agents against the biased policies of [YGL+23]. We show results for both layouts in Table 6.6. We find three results specifically noticeable: (i) Average performance seems to be mostly determined by the training methodology where HSP based methods outperform others, (ii) MEP benefits the most from the additional ToM auxiliary tasks (MEP$_{g,a}$ and MEP$_{g,a,n}$ roughly achieve 50% higher scores compared to the baseline MEP), and (iii) the best and worst performing agents were trained with ToM auxiliary tasks (HSP$_g$ and SP$_{g,a}$). We additionally observe that ToM auxiliary tasks only results in higher or roughly equal performance, except for in self-play.

	Asymmetric Advantages				Many Orders				
Script	O. Plc.		O. Plc. & Dlv.		T. Plc.		T. Plc. & Dlv.		
At Pos.	1	2	1	2	1	2	1	2	Avg
SP	0.6	169.0	258.9	178.5	77.0	78.3	204.2	201.6	146.0
SP$_g$	4.6	173.7	215.9	177.9	36.7	37.8	198.8	197.9	130.4
SP$_{g,a}$	0.0	99.5	231.8	175.6	9.7	10.1	175.2	177.0	109.8
SP$_{g,a,n}$	0.0	327.1	293.5	178.9	13.0	13.7	188.2	183.4	149.7
HSP*	352.5	381.6	357.8	244.6	**327.0**	304.4	254.4	257.3	309.9
HSP$_g$	**372.8**	**419.4**	361.2	234.6	307.7	**307.5**	250.4	238.8	**311.5**
HSP$_{g,a}$	352.8	396.4	357.2	**249.2**	296.1	301.1	259.4	259.9	309.0
HSP$_{g,a,n}$	319.8	414.8	**364.6**	239.4	301.8	304.4	254.4	257.3	307.0
MEP	313.8	362.2	325.6	218.0	31.4	24.8	182.9	179.3	204.7
MEP$_g$	330.2	362.2	325.6	218.0	39.4	24.8	181.7	176.3	207.2
MEP$_{g,a}$	301.0	378.6	355.0	235.6	266.8	264.8	280.7	281.7	295.5
MEP$_{g,a,n}$	330.4	359.2	352.6	244.8	289.6	297.8	**291.0**	**292.4**	307.2

Table 6.6: 'O. Plc.' (i.e. 'Onion Placement') refers a policy that only places onions in pots, 'O. Plc. & Deliv' (i.e. 'Onion Placement & Delivery') also delivers soups if they have finished cooking. The same is true for 'T. Plc.' (i.e. 'Tomato Placement') and 'T. Plc. & Deliv' (i.e. 'Tomato Placement & Delivery') but with tomatoes instead. 'At Pos.' refers to the position the scripted agent takes. Note that some versions of self-play achieve a score of 0.0 (or close to 0.0) with the onion placement policy, essentially failing the game. We investigated the behaviour and discovered the motivating example explained in Figure 2.1, further highlighting the importance of evaluating with strongly biased policies. HSP* refers to the trained model released by Yu et al. [YGL+23]. The best results are in **bold**, the worst ones underlined.

That average performance is mostly dictated by the training methodology is not surprising to us. First note, that HSP is trained to perform well with biased agents. Therefore, we expect a method that sits on top of HSP based training to perform similarly well. Additionally, it is known that self-play overspecialises with itself and therefore performs especially bad with agents that behave very different. MEP performs better than SP since it is trained to perform well with a diverse range of policies but not with any special focus on biased behaviour which places it below HSP based policies. Observation (i) clearly is explained by these differences in training. With regards to (ii), we can only speculate why MEP benefits so much from ToM auxiliary tasks. Clearly, ToM auxiliary tasks biases the trained agent towards the behaviour of its partner. This might also be the reason why ToM increases performance: biasing might be especially beneficial to MEP as it is trained with a diverse population as measured by a population entropy term with no emphasis on creating very biased policies. The benefit MEP receives from ToM modelling is high enough to nearly compensate for the missing training with biased policies, i.e. the difference between HSP* and MEP$_{g,a,n}$ on average only is 2.7 points where the difference between the baselines MEP and HSP* is 105.2. Auxiliary ToM tasks apparently bridge most of this gap. When it comes to the fact that we observed the best and worst performing agent was trained with ToM auxiliary tasks in (iii), we remind the reader that previously we observed that SP based policies are increasingly competent at the game during self-play but at the cost of other-play performance. This is reflected when playing with biased policies as SP performs better or at least nearly as good as its ToM counterparts with biased policies. Excluding the SP policies in determining the worst performer, MEP would achieve the lowest scores in five out of the eight evaluation runs against biased policies, followed by MEP$_g$. Overall, this evidence suggests that self-play is not capable of zero-shot cooperation no matter additional ToM auxiliary tasks for structural reasons. Without self-play these auxiliary tasks improve results in both state-of-the-art methods.

6.3 Unit Testing for Robustness

To establish whether ToM auxiliary tasks improve performance, we evaluate our agents against a suite of different unit tests provided by Knott et al. [KCD+21]. We outline the results in Table 6.7 for Asymmetric Advantages and in Table 6.8 for Many Orders. These tests are challenging as established by the baseline of a random agent which will not get a single test correct for most categories. Moreover, these tests are also challenging for more capable self-play based methods as all best results are achieved by methods who were trained against a diverse population. Between these population based methods though, four of the six best results come from HSP variants, two from HSP* and two from HSP$_g$, while the other two were obtained by MEP and MEP$_{g,a}$. This gives a mixed impression on whether ToM modelling actually improves the unit test scores of agents. It seems much more the case that any additional percentage points obtained might be due to the fact that the agent is more capable at the task. In Knott et al. [KCD+21] work simple tricks and methods - like randomly having agents start from all possible game positions - improved the success rate reliably and we do not observe a similarly strong effect.

In the best cases ToM auxiliary tasks significantly improve results. For instance for HSP-based methods in state robustness tests HSP$_{g,a}$ improves the performance of HSP* from 44.9% by 8.2 points to 53.1% on Asymmetric Advantages, or HSP$_g$ add 6.7 points

	Asymmetric Advantages		
	State	Agent	Agent & Memory
(Random)	0 ± 0.0	0 ± 0.0	0 ± 0.0
SP	26.7 ± 7.9	35.3 ± 8.2	34.0 ± 5.1
SP$_g$	26.4 ± 8.2	28.7 ± 9.4	31.3 ± 7.8
SP$_{g,a}$	29.6 ± 9.7	9.0 ± 3.2	29.7 ± 3.8
SP$_{g,a,n}$	27.4 ± 4.5	22.9 ± 6.9	40.5 ± 6.2
HSP*	44.9 ± 8.4	76.2 ± 11.1	77.1 ± 4.6
HSP$_g$	42.5 ± 13.8	71.2 ± 8.3	**83.8** ± 8.0
HSP$_{g,a}$	53.1 ± 3.0	76.2 ± 5.2	73.6 ± 6.7
HSP$_{g,a,n}$	50.6 ± 12.8	61.2 ± 8.1	71.6 ± 7.1
MEP	52.0 ± 11.9	**81.2** ± 6.2	67.4 ± 6.4
MEP$_g$	54.1 ± 3.7	67.5 ± 9.2	81.9 ± 6.8
MEP$_{g,a}$	**60.8** ± 5.8	65.0 ± 18.0	82.4 ± 7.1
MEP$_{g,a,n}$	48.1 ± 6.1	62.5 ± 9.8	76.9 ± 5.3

Table 6.7: Unit testing results for the Asymmetric Advantages layout. Results are given in percentages out of 100%. Results are averaged across five random seeds. We additionally show the variance. Two out of the three best performing models have received ToM auxiliary task training. The best results are in **bold**, the worst ones underlined.

in agent & memory tests also in Asymmetric Advantages. But these results do not generalize across layouts and both HSP$_g$ and HSP$_{g,a}$ perform worse in Many Orders. Similarly, for SP where the worst performing methods overall are usually some version of SP with ToM auxiliary tasks but some SP ToM versions improve the results of SP in a few categories. Note here, for instance SP versus SP$_{g,a,n}$ on Asymmetric Advantages, as shown in Table 6.7. Again, these results does not generalize well to the other layout, Many Orders. MEP most often benefits from ToM auxiliary tasks. Take MEP$_{g,a,n}$ as an example which usually performs better than MEP on both layouts.

6.4 Performance with a Human Model

We present the results for comparing our model against the Human Theory of Mind model of Knott et al. [KCD+21] in Table 6.9. The human model is planning-based - like the biased policies - and thus also biased in its behaviour. It is thus no surprise that HSP* performs well with this model and in fact outperforms all other models and, in the

	Many Orders		
	State	Agent	Agent & Memory
(Random)	0.0 ± 0.0	53.3 ± 1.2	0.0 ± 0.0
SP	10.8 ± 7.3	57.0 ± 5.1	39.9 ± 6.5
SP$_{g}$	12.2 ± 5.5	51.0 ± 8.5	20.9 ± 22.2
SP$_{g,a}$	16.4 ± 5.3	51.0 ± 5.5	9.5 ± 8.1
SP$_{g,a,n}$	7.4 ± 0.8	44.9 ± 3.7	0.0 ± 0.0
HSP*	**53.2** ± 7.6	76.5 ± 4.8	**73.8** ± 7.2
HSP$_{g}$	37.4 ± 7.4	**84.2** ± 4.3	34.1 ± 8.4
HSP$_{g,a}$	42.4 ± 4.7	75.9 ± 2.0	59.1 ± 6.1
HSP$_{g,a,n}$	29.9 ± 8.1	53.1 ± 24.2	22.4 ± 8.0
MEP	6.2 ± 4.1	53.1 ± 7.2	0.0 ± 0.0
MEP$_{g}$	10.4 ± 5.3	55.6 ± 5.4	2.4 ± 3.7
MEP$_{g,a}$	30.4 ± 5.8	72.9 ± 6.0	53.8 ± 9.0
MEP$_{g,a,n}$	25.3 ± 2.6	61.1 ± 3.2	44.4 ± 2.9

Table 6.8: Unit testing results for the Many Orders layout. Results are given in percentages out of 100%. Results are averaged across five random seeds. We additionally show the variance. One out of the three best performing models has received ToM auxiliary task training. The best results are in **bold**, the worst ones underlined.

case of Many Orders, does so with a huge margin. This also includes ToM variants of HSP. While their performance generally remains competitive on Asymmetric Advantages, it does not do so on Many Orders. In fact, HSP* performs more than twice as good as other models. Still, this does not make ToM modelling useless. MEP for instance readily benefits from ToM modelling and all MEP variants that include ToM auxiliary task training are performing better than the MEP baseline, excluding MEP$_{g}$ on Many Orders which still shows competitive performance. In the case of SP based methods results are more mixed. While SP$_{g,a,n}$ outperforms all other forms of SP on Asymmetric Advantages, ToM versions of SP show large variance in Many Orders when it comes to performance. Specifically, over the five random seeds, the best SP runs showed better performance than any other SP-based runs but the same was true for the worst ones. It seems that ToM versions of SP only performed well in certain cases. This also might be due to the already discussed effect of overspecialisation with oneself which is especially prominent in ToM self-play. Overall, this leaves HSP* as the best model to cooperate with the Human Theory of Mind model.

	Asymmetric Advantages	Many Orders
(Random)	38.2 ± 3.2	3.3 ± 1.7
SP	188.3 ± 5.2	171.8 ± 8.8
SP$_g$	183.3 ± 18.8	87.4 ± 79.8
SP$_{g,a}$	$\underline{143.1} \pm 11.4$	87.0 ± 87.5
SP$_{g,a,n}$	210.2 ± 14.2	76.6 ± 77.9
HSP*	**276.8** ± 2.9	**333.4** ± 3.8
HSP$_g$	270.4 ± 4.1	116.4 ± 12.3
HSP$_{g,a}$	273.6 ± 3.0	120.8 ± 11.2
HSP$_{g,a,n}$	271.8 ± 2.4	185.0 ± 9.5
MEP	257.6 ± 8.3	15.5 ± 2.1
MEP$_g$	270.8 ± 6.3	$\underline{14.8} \pm 1.7$
MEP$_{g,a}$	273.8 ± 2.2	185.4 ± 8.7
MEP$_{g,a,n}$	273.4 ± 4.7	195.8 ± 3.5

Table 6.9: Performance with the Human Theory of Mind model of Knott et al. [KCD+21]. We show the rewards for 2 * 20 = 40 runs, where we alternate the two starting positions for all 20 possible Human Theory of Mind model variants we test. The best results are in **bold**, the worst ones underlined.

6.5 Observations regarding Many Orders and HSP

One might find our results to be quite mixed at first glance. If so, we want to point out the difference in results across layouts. On Asymmetric Advantages, ToM versions have the (i) highest other-play and self-play score (Table 6.3 & 6.5), (ii) are the best with strongly biased policies in all positions with both the ‘Onion Placement’ and the ‘Onion Placement & Delivery’ policies (Table 6.6), (iii) achieve the highest unit test success rate in two out of the three categories (Table 6.7) and (iv) only barely do not manage to get the highest mean reward with human models (Table 6.9). In fact, closely reconsidering the results presented in Table 6.9 regarding point (iv), one notices that several ToM policies are within the range of variation to HSP*, especially MEP$_{g,a,n}$ and MEP$_g$ which improve considerably over MEP, a pattern we could already observe when comparing our methods against strongly biased policies in Table 6.6.

Interestingly, this effect reverses on Many Orders where HSP* (i) has the highest other-play and self-play score (Table 6.4 & 6.5), (ii) wins one out of the four comparisons with strongly biased policies (Table 6.6), (iii) achieves the highest unit test success rate in two out of three cases (Table 6.8) and (iv) wins the evaluation against human models easily (Table 6.9). HSP* thus clearly is the best model on Many Orders. Since we base our implementation of HSP on the one provided by Yu et al. [YGL+23] and use the HSP* model provided by them, this is likely not due to our implementation. Since there is a lot of variance in training reinforcement learning policies, especially if large populations of agents are involved, we wondered whether this might be due to random chance in the training.

Before we have already wondered whether this difference might be due to the training population we have generated. To test our hypothesis, we train our own HSP model from scratch and observe that - while it behaves similarly well on Asymmetric Advantages - it performs considerably worse on Many Orders also. This is in line with all our other training results. Our retrained version of HSP achieves a self-play score of 376.1, a other-play score of 267.1 with a OP - SP score of -100.0 (HSP* achieves scores of 375.9, 356.6 and -19.0 here respectively). This places it below our best performing ToM variant (that was trained with the exact same setup) in terms of other-play score: HSP$_{g,a}$ achieves an other-play score of 283.3, compare Table 6.5. When it comes to playing with biased policies this HSP model actually performs slightly better than the HSP* variant, winning two out of the four categories. In unit testing HSP performs significantly worse, loosing many percentage points in both the state robustness tests (from 53.2 ± 7.6 to 38.3 ± 8.6) and agent & memory robustness tests (from 73.8 ± 7.6 to 63.7 ± 8.7). This would place it second in terms of state robustness tests and only barely above our HSP$_{g,a}$ model in the agent & memory robustness tests. Lastly, in terms of performing with a scripted human model HSP greatly losses performance from 333.4 ± 3.8 to 116.4 ± 5.2 points of reward which would comfortably places it below four of our models HSP$_{g,a}$, HSP$_{g,a,n}$, MEP$_{g,a}$ and MEP$_{g,a,n}$. While we believe this comparison to be more fair as these results were obtained from the same training population and training these populations comes with a lot of randomness, we also believe in comparing ourselves against the best model form the literature which is why we have decided to compare ourselves against HSP* here. Time constraints do not allow us the costly operation of retraining an entire HSP population but this analysis highlights the brittleness of this approach and puts our results into a better perspective.

7 Analysis, Discussion, & Limitations

Given the previously presented results, we now want to go back and try to finalise an answer to the research questions we posed in the Introduction:

H1 *ToM-versions of agents have higher average validation reward when playing with different partners compared to their non-ToM counter parts*

H2 *ToM-versions of agents have higher average validation reward when playing with strongly biased policies compared to their non-ToM counter parts*

H3 *ToM-versions of agents have higher unit test success rate compared to their non-ToM counter parts*

H4 *ToM-versions of agents achieve highest evaluation reward when playing with other ToM agents*

We will address these questions first, before moving on with additional analysis and discussion where we will also point out any limitations.

7.1 ToM agents can have higher average validation reward when playing with different partners

To answer the research questions we need to recall the cross- and other-play results as well as the results obtained when playing with the Human Theory of Mind model. For this first recall our discussion on the SP model results. Here we observed that ToM auxiliary task modelling increases self-play performance (see Table 6.1). Additionally, we also observed in Table 6.2 that our ToM models had a lower other-play score and showed increasingly growing gaps between other-play and self-play scores. We therefore conclude that ToM auxiliary tasks do not *by themselves* help an agent to play well with others. In fact, we reason that the increased competency makes these SP ToM versions more capable of collaborating with *themselves* at the cost of other-play performance (see the discussion in Section 6.1). To us, this does not come as a surprise as higher validation reward during self-play is not correlated with performing better with other partners. In fact quite the contrary is likely true due to overspecialisation with oneself, as we have already hinted at in the introduction. SP, as well as SP with ToM auxiliary tasks, clearly fail the zero-shot cooperation problem, no matter how competent they are at the game in self-play.

As we have assumed so from the start, we also evaluate our method in combination with techniques for best response training against a diverse population, i.e. MEP and HSP. On the one hand, for Asymmetric Advantages the results are clear: ToM auxiliary tasks improve other-play performance over the baselines. Specifically we find that for HSP all versions involving auxiliary tasks are better than the baseline HSP* and we find that in the case of MEP, the version with all three auxiliary tasks performs better than the baseline. Since HSP* to our knowledge is the current SOTA in human-AI cooperation in Overcooked, this makes it a very hard benchmark to beat. On the other hand, results for Many Orders are less easy to interpret. Compared to HSP* our models perform worse in other-play. Above we have argued that this is likely due to differences in the training population which we suspect might be due to randomness in arriving at this population in the first place. If we retrain our own HSP baseline with their implementation and our population from scratch, the tables turn and ToM auxiliary modelling also improves results on Many Orders. This is also the case for playing with a human model where then our models outperform the baseline.

Clearly, ToM auxiliary tasks can improve both MEP and HSP but without either MEP or HSP, ToM auxiliary task training is not enough to guarantee good zero-shot cooperation performance which is what we can observe in the SP analysis. In fact, given this evidence we reason that ToM auxiliary tasks biases the agent towards its training partner(s) which in turn elevates performance on the training distribution. If this training distribution is diverse enough this carries over to the problem of zero-shot cooperation and together produces very capable policies.

We conclude that ToM agents have higher average validation reward when playing with different partners, if they are trained using a method that provides diversity in training partners. We thereby argue that this hypothesis is true *conditionally*.

7.2 ToM-versions of agents have higher average validation reward when playing with strongly biased policies

Our experimental results presented in Table 6.6 showed that ToM auxiliary tasks help with performing well with strongly biased policies. In fact HSP$_g$ has the highest average reward over both layouts, obtaining the new SOTA results compared to the work of Yu et al. [YGL+23] on Asymmetric Advantages (see Table 23 in [YGL+23]). On Many Orders, we can not replicates the results of Yu et al. [YGL+23] even when using the model they provided and consequently do not reach their numbers: not with their provided model and not with ours. Next we can see that MEP especially is greatly improved through ToM auxiliary task training as both MEP$_{g,a}$ and MEP$_{g,a,n}$ improve the baseline by roughly 50%. Only SP agents do not benefit from ToM auxiliary task modelling when it comes to performing with strongly biased policies. We suspect that this is also due to overspecialisation.

In conclusion, our method achieves the SOTA in one of the layouts, achieves the highest average reward and greatly improves another well established method from the literature by up to 50%. We thereby conclude that in fact ToM-versions of agents have higher average validation reward when playing with strongly biased policies.

7.3 ToM-versions of agents can have better unit test success rate compared to their baselines

Looking at the unit testing results we find that a pattern emerges. In two out of the tree categories a ToM model achieves the highest score on Asymmetric Advantages while the opposite is true on Many Orders as long as we compare our models to HSP*. If we train our own HSP agent, this effect vanishes on both layouts. In this case HSP$_{a,g}$ would replace HSP* as the best unit test model on Many Orders.

Currently, HSP$_g$ performs best at unit testing from the family of ToM models. Additionally, if we exclude HSP* and instead compare to HSP, HSP$_g$ performs best *overall*. We find this to be noteworthy as HSP$_g$ also is the best performer on playing with strongly biased policies. It therefore seems that these two problems either are related or HSP$_g$ implements a policy that is useful in both evaluation settings. Still, we do not feel confident in claiming that ToM auxiliary task modelling strictly improves unit test results. Some success rates of ToM variants significantly drop below their baseline and sometimes even under the random baseline, which is especially puzzling. We therefore do not feel confident in asserting that we can positively resolve this research question.

7.4 ToM-versions of agents achieve highest evaluation reward when playing with other ToM agents on Asymmetric Advantages

Table 6.3 clearly shows how ToM variants perform the best when playing with other ToM variants and Table 6.4 shows the exact opposite. As already stated in Section 6.5

the results observed in Table 6.4 are likely due to some random variation in the training population to which reinforcement learning is very sensitive. We can therefore only answer this research question when separating out these cases. ToM-versions of agents achieve highest evaluation reward when playing with other ToM agents on Asymmetric Advantages but also achieve lowest evaluation reward when playing together on Many Orders. Additional research must be carried out to evaluate how this translates to other environments. For now, we resolve hypothesis H4 as being only partly true.

7.5 Do Agents Represent Different Types of Collaborators Differently?

As last part of our analysis, we investigated whether our agents represent different collaborators differently. This is partly inspired form the original work on Machine Theory of Mind in which Rabinowitz et al. [RPS+18] investigated the 2D embedding space of their observer when observing agents with different goals to determine whether their ToMNet was able to distinguish between these in the embedding space. Since our agents do not have different goals but are trained differently we were curious whether these differences are encoded in the embeddings of our agents. However, we expect this task to be noticeably harder than the analysis performed by Rabinowitz et al. [RPS+18] as in our case the goal remains the same and the only difference comes in the observed behaviour. Since differently trained policies might converge on the same optimal collaboration behaviour, we guess that obtaining disentangled representations of different collaboration partners with same behaviour might be impossible.

We collected the embeddings produced by the agents' LSTM at every timestep of the gameplay over $N = 100$ games and calculated the element-wise mean over the vectors of every single game, obtaining 100 total gameplay embeddings for all cooperation partners. We used Principal Component Analysis [FRS01] to calculate a 2D embedding space from these and show them in Figure 7.1. In the figure, columns denote the layout and rows the kind of agent. The colored points represent different training partners.

We would expect that partners that behave similarly are clustered in this embedding space if our agents are able represent their behaviour adequately. This is best highlighted by exploring the 2D embeddings produced when playing with strongly biased policies as these show the most extreme form of behaviour. In the figure, strongly biased policies are represented by different shades of blue. Since one of the biased policies is delivering soups and the other is not, we would expect that the models represent them separately. For most agents this is the case. The 2D embeddings generated from HSP$_{g,a,n}$ or MEP$_{g}$ for instance are especially clearly clustered. Less clear examples include both HSP and

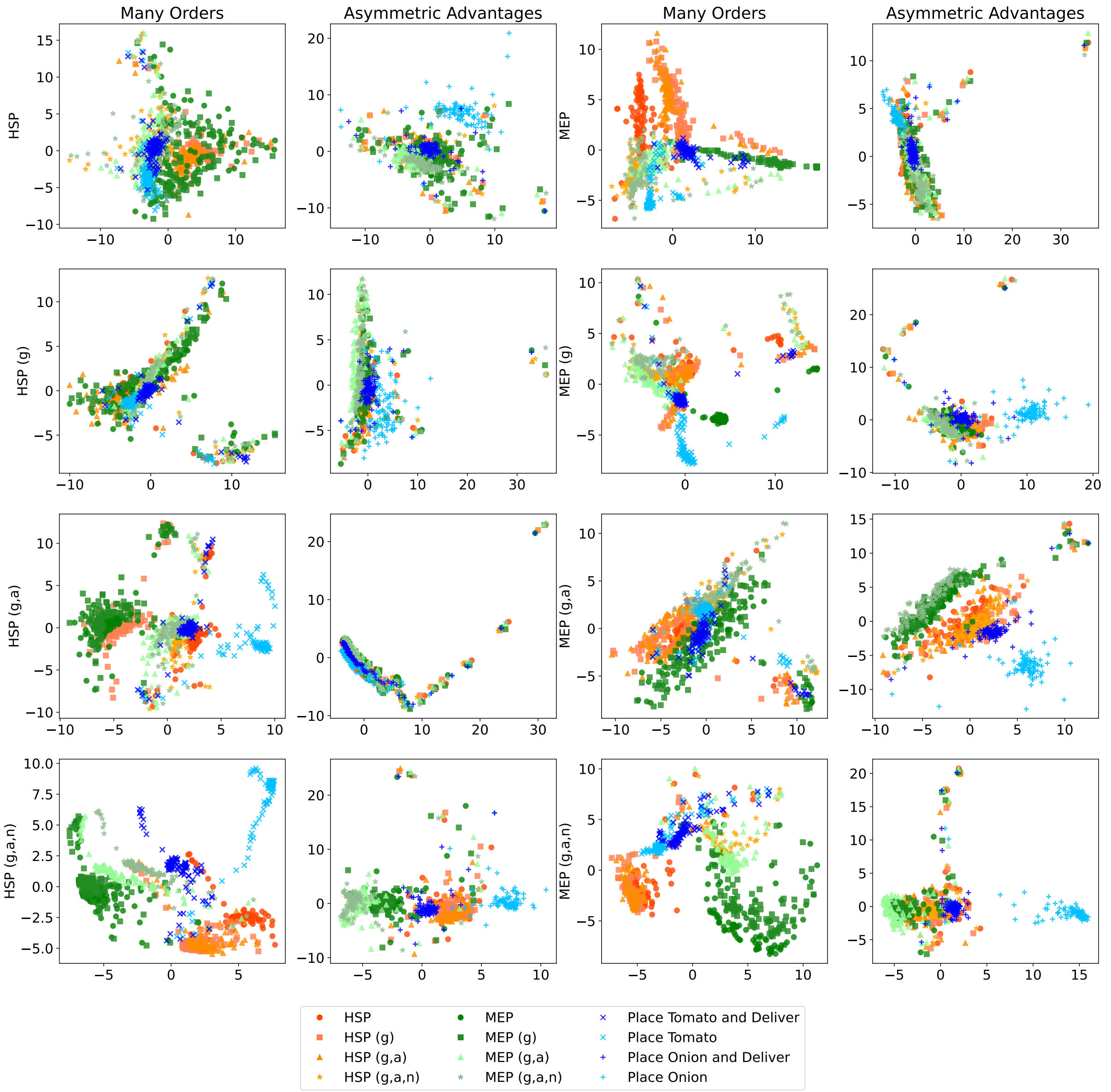

Figure 7.1: Principal Component Analysis with $n = 2$ components computed from the policies network during play with different cooperation partners on both layouts. For this analysis we extracted the embedding produced directly after the LSTM in the actor network as it is common to all architectures.

HSP_g which form no clear clusters on Many Orders. A special case might be $\text{HSP}_{g,a}$ which seemingly forms no clear cluster on Asymmetric Advantages. This might be due to the scale of the chart.

Next, we studied whether these agents are able to distinguish between HSP- and MEP-based cooperation partners. We suspect that, due to the differences in training population, these behave differently. To make a visual examination easier we colored HSP variants in shades of orange and MEP ones in shades of green. Since we suspect this kind of analysis to be prone to confirmation bias, we fit a logistic regression classifier on the data instead of performing visual analysis. We train it on 70% of the 2D embedding data and evaluate whether it is able to correctly identify the other player on the other 30%. Our classifier is trained with default parameters, for 500 iterations. On Many Orders, this classifier reaches the highest classification testing accuracy on HSP$_{g,a,n}$ and the lowest on HSP$_g$ (roughly 70% and 12% respectively). Note that for this task, guessing randomly results in a performance of 10%. HSP$_{g,a,n}$ also has the highest testing classification rate of roughly 50% on Asymmetric Advantages while the classifier achieves the lowest accuracy when working with MEP with roughly 50%.

We repeated this experiment but this time only ask our classifier to predict whether the embedding belongs to a biased, MEP- or HSP-based policy (random baseline performing at 36%). The results stay the same on Many Orders but rise significantly: from 70 to 92% for HSP$_{g,a,n}$ and from 12 to 50% for HSP$_g$. On Asymmetric Advantages the classifier performs the second best with HSP$_{g,a,n}$ resulting in a new score of 85.9% whereas the classifier now works the best on MEP$_{g,a}$ with 88.8%. The classifier still performs the worst on MEP, with 50% testing accuracy. On average, we observed that in the harder task the classifier has a higher testing accuracy on ToM agents with 37% versus 34% when averaged across layouts. The same holds on the task of predicting agent type 67.9 versus 54.4%. We report raw results in the Appendix A.4.

7.6 Summarizing the Analysis

While trying to answer our four research questions, we repeatedly stumble over the surprising difference in performance of our methods between Asymmetric Advantages and Many Orders. With the additional analysis performed in Section 6.5, we established that our methods would also perform great in comparison to an HSP baseline trained by us, from scratch, given the resources available to us and the original implementation. We can therefore not be certain that the blame for failing to achieve SOTA performance on Many Orders is due to the variation in training populations or some other difference in setup. Since reinforcement learning is especially sensitive to randomness, one can never rule out that the differences observed are due to randomness. Throughout the literature we have not observed a method performing great on one and badly on a different layout. This is likely the case as game mechanics do not greatly differ between the layouts. This being said, overall, we believe our results support the hypothesis that ToM auxiliary task modelling is beneficial and useful.

7.7 Limitations

Our method comes with certain limitations one should consider. First, since our approach sits on top of other methods and is dependent on a diverse population of training partners it also inherits many of their limitations. Obtaining a diverse population of agents is computationally expensive and time consuming. Training a population of diverse agents requires running the same training algorithm multiple times. This effect is especially prominent in environments where evaluation is performed on many layouts, like Overcooked, as one final agent needs to be trained for every layout. In this work, we trained 12 MEP agents for a population of 36 total agents and additional HSP agents for both layouts, before training all the best response agents we have presented above.

In terms of our own method we want to highlight two especially prominent limitations: (i) the need for strategic goals and (ii) the fact that auxiliary ToM tasks alone may not be sufficient for state-of-the-art performance. Starting with the latter, as our own analysis shows without diverse populations our method does not improve other-play performance, the contrary is the case. Our approach can thus only improve results given a population. Regarding (i) the limitation is clear: one needs enough domain expertise to be able to find suitable goals for strategic goal predictions. The environment furthermore needs to allow the user of our methods to access gameplay state to determine goals as play occurs for expanding the dataset $\mathcal{D}$. Additionally, these goals have to meaningfully correspond to the goals, desires and preferences our agent might have and thus need to be well designed. While this seems to be quite challenging, recall that access to certain parts of the game events is often the case in reinforcement learning. Specifically, in many settings agent get shaped reward during training that alters the reward structure of the environment to improve learning. These also are often directly based on game events and usually a similar level of access is needed for both. Thereby, as the use of shaped reward is ubiquitous in RL, one could use similar events to design strategic goals. In fact in Overcooked, the same events that make goal prediction possible are also used to perform reward shaping.

Currently, all state-of-the-art approaches share the need for training a population of diverse agents. As this requires a lot of additional compute, future work should focus on methods that are capable of training agents capable of zero-shot coordination without the need of training a big population. Additionally, as currently one agent needs to be trained for each layout separately, we suggest that future work should also tackle zero-shot coordination in unknown layouts. This closely resembles human capabilities which are also not limited to cooking in a single kitchen only.

8 Conclusion

This work started with the intention of evaluating whether Theory of Mind modelling could be used to improve the performance of agents during zero-shot coordination, thereby improving robustness. To do so we have introduced a novel mechanism of belief supervision in Overcooked from game interactions alone and no need for ground truth mental states. We augmented popular training algorithms with this technique and evaluated and characterized their behaviour in a combination of evaluation suits, all designed to evaluate the performance of agents in Overcooked. Our approach proofed powerful in aiding the training of self-play agents and significantly raised their self-play performance. Conversely this increase in self-play performance came with a decrease in other-play performance, showing that self-play alone can not be a good zero-shot cooperation partner regardless of competency. This is in line with previous research. We thus amended current state-of-the-art techniques with our method and showed that this further increased their capabilities also in zero-shot cooperation, especially on the Asymmetric Advantages layout. On the Many Orders layout, we observed that our technique was not competitive to the current stat-of-the-art technique Hidden Utility Self-Play (HSP), as provided by the authors. However, after training our own HSP model, we found it to be competitive again. This suggests that another issue may have been responsible for the poorer results on the Many Orders layout. While our results suggest that ToM auxiliary task training is a powerful mechanism for increasing performance, we do not see the same improvements in unit testing for robustness. This suggests that good performance and good unit testing results might not necessarily be linked. Overall our technique proofs to be a useful way to increase performance from simple dataset augmentations alone.

A Appendix

A.1 Unittesting for Robustness: A List

Table A.1 gives a complete overview over all tests being performed during unittesting for robustness. Note that the number of variations differ based on the layouts as variations also include different starting positions for each test which reasonably differ between layouts. If a test has zero variations it is not applicable to this layout. Test1ai is a special case since in both layouts blocking a dispenser is not feasible either because more are close by (in Many Orders) or because two agents never directly interact (in Asymmetric Advantages).

A.2 Human Theory of Mind Model Parameters

We keep the original parameterization as the original work of [KCD+21]. In total we use two versions of the model, one optimized for collaboration and one that is completely rational and greedy for being the partner in many unittests (i.e. Section 5.4)) and build a population of 20 agents for the validation runs (i.e. Section 5.5). The parameters are consistent between layouts and are presented in Table A.2.

A.3 Cross-Play Matrices

We present full cross-play results in Table A.3 and A.4.

A.4 PCA Logistic Regression Results

We present results on the the predicting the cooperation partner and partner type from 2D PCA embeddings tasks in Table A.5.

			# Variations	
Test	C	Short Test Description	Asym.	Many
Test1ai	a	Pick up a dish from a counter: H blocks dispenser	0	0
Test1aii	a	Pick up an object from a counter: dispenser is available but counter object is much closer than dispenser	8	4
Test1aiii	a	Pick up a soup from a counter: Soup on the counter	8	4
Test1bi	a	R is holding the wrong object, and must drop it. Variants: 1) R has D when O needed (both pots empty) 2) R has O when two Ds needed (both pots cooked)	4	4
Test1bii	a	Drop objects onto counter: R holding the same object as H, but H is closer to using it	8	8
Test2a	b	Getting out the way of H: R in the way of H, where H has the right object. Variants: 1) H has onion, onion needed in pot, 2) H has dish, dish needed for pot	0	12
Test2b	b	Getting out the way of H: H is holding a soup, and R is on the shortest path for H to deliver soup		
Test3ai	c	H is holding nothing or an object that can't currently be used (H is stationary)	6	6
Test3aii	c	H is holding a dish or onion, which can currently be used (H is stationary)	4	4
Test3aiii	c	H is holding a soup (H is stationary)	2	2
Test3bi	c	H is holding nothing (H is random)	2	2
Test3bii	c	H is holding a dish or onion, which can currently be used (H is random)	4	4
Test3biii	c	H is holding a soup (H is random)	2	2
Test4a	a	R has onion, pot needs onion. (H is stationary)	2	2
Test4b	a	R has dish. (H is stationary)	2	2
Test4c	a	R has onion, pot needs onion, and there are onions all over the counters (H is stationary).	2	2
			54	58

Table A.1: A description of the unittests employed for our agent evaluation. R refers to the agent being tested and H to the model that R has to work with (often planning based, can also be stay, random or a neural one). For the exact details please refer to the implementation. Column C with (a), (b) and (c) refer to the taxonomy outlined in section 5.4.

	Unittesting		Play w/ Human Model
Parameter	Teamwork	Greedy	Validation Population (20)
compliance	0.5	0.0	[0.1,0.5,0.9]
retain goals	0.0	0.0	[0.0,0.8]
probability thinking not moving	0.0	0.0	[0.0,0.2,0.4,0.7]
path teamwork	1.0	0.0	[0.1,0.5,0.9]
rationality coefficient	20.0	20.0	[1.0,3.0,5.0,10.0,20.0]
probability greedy	0.2	1.0	[0.0,0.3,0.7,1.0]
probability to observe other	0.0	0.0	[0.0,0.3,0.7,1.0]
look ahead steps	4.0	4.0	4.0
prob pausing	0.0	0.0	[0.3,0.4,0.5,0.6]

Table A.2: Values for Theory of Mind human model parameters by [KCD+21] for both the partners in unittesting and the validation runs against a human model. Note that for the latter we only give the values being picked from and otherwise point the reader to our implementation.

	Asymmetric Advantages							
Model	HSP*	HSP$_g$	HSP$_{g,a}$	HSP$_{g,a,n}$	MEP	MEP$_g$	MEP$_{g,a}$	MEP$_{g,a,n}$
HSP*	377,4	371,6	382,2	366,2	386,8	384,6	419,2	413,8
HSP$_g$	399,0	387,0	405,6	397,6	416,0	414,6	447,6	429,8
HSP$_{g,a}$	430,0	401,6	442,8	423,8	443,0	**455,4**	**459,6**	453,4
HSP$_{g,a,n}$	386,8	375,6	392,0	391,4	421,0	432,4	450,2	447,4
MEP	417,0	421,0	440,6	422,6	422,8	440,8	428,0	414,8
MEP$_g$	415,6	374,4	429,2	404,2	412,4	435,0	443,0	430,2
MEP$_{g,a}$	412,8	377,8	422,2	394,8	442,0	452,6	**454,8**	**452,6**
MEP$_{g,a,n}$	417,2	410,4	443,2	422,8	436,2	442,0	**455,0**	448,4

Table A.3: Cross-Play results in the layout Asymmetric Advantages. The metric measured is average validation reward averaged across 100 games. HSP* refers to the trained model released by [YGL+23]. The five best results are in **bold**, the five worst ones underlined.

	Many Orders							
Model	HSP*	HSP$_g$	HSP$_{g,a}$	HSP$_{g,a,n}$	MEP	MEP$_g$	MEP$_{g,a}$	MEP$_{g,a,n}$
HSP*	375,9	383,9	342,0	372,1	344,2	326,0	346,6	366,8
HSP$_g$	394,2	199,4	202,8	341,2	188,4	199,0	344,7	363,6
HSP$_{g,a}$	344,5	202,4	209,0	351,0	203,2	199,0	357,8	362,6
HSP$_{g,a,n}$	380,3	344,5	353,2	168,9	220,5	204,4	173,6	200,6
MEP	351,6	186,4	201,4	221,6	0,0	0,0	190,0	238,0
MEP$_g$	336,6	199,2	199,6	205,6	0,0	0,0	193,6	234,4
MEP$_{g,a}$	340,8	347,0	359,3	172,6	190,0	194,2	187,1	210,2
MEP$_{g,a,n}$	380,4	359,2	362,6	199,6	238,8	234,4	204,6	227,6

Table A.4: Cross-Play Results in the layout Asymmetric Advantages. The metric measured is average validation reward averaged across 100 games. HSP* refers to the trained model released by [YGL+23]. The five best results are in **bold**, the five worst ones underlined.

	Predicting Cooperation Agent		Predicting Cooperation Agent Type	
Model	Many Orders	Asymm. Adv	Many Orders	Asymm. Adv
HSP*	31,1	30,3	50,3	61,8
HSP$_g$	12,2	25,9	50,0	58,5
HSP$_{g,a}$	45,9	14,0	55,5	50,0
HSP$_{g,a,n}$	70,3	49,6	92,2	85,9
MEP	57,0	20,0	59,2	46,2
MEP$_g$	45,1	31,8	50,7	68,1
MEP$_{g,a}$	29,2	41,1	52,5	88,8
MEP$_{g,a,n}$	47,0	42,9	81,4	81,4

Table A.5: Logistic regression classifier results measured by accuracy in percent on a 30% hold-out-set of the 2D embedding to cooperation partner task. We test two versions of this task in the first the exact agent needs to be predicted, i.e. HSP, HSP$_g$, HSP$_{g,a}$, HSP$_{g,a,n}$, MEP, MEP$_g$, MEP$_{g,a}$, MEP$_{g,a,n}$ and the resptice biased policies for the layout. In the second task only the type of agent needs to be predicted, i.e. HSP, MEP or biased policy.

Bibliography

[Aga18] A. F. Agarap. “Deep learning using rectified linear units (relu).” In: *arXiv preprint arXiv:1803.08375* (2018) (cit. on p. 32).

[AN04] P. Abbeel, A. Y. Ng. “Apprenticeship learning via inverse reinforcement learning.” In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 1 (cit. on p. 38).

[AOS+16] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, D. Mané. “Concrete Problems in AI Safety.” In: *CoRR* abs/1606.06565 (2016). arXiv: 1606.06565. URL: http://arxiv.org/abs/1606.06565 (cit. on p. 21).

[AZI+18] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, O. Khatib. “Progress and Prospects of the Human-Robot Collaboration.” In: *Autonomous Robots* 42 (June 2018). DOI: 10.1007/s10514-017-9677-2 (cit. on p. 12).

[BCC21] C.-P. Bara, S. CH-Wang, J. Chai. “MindCraft: Theory of Mind Modeling for Situated Dialogue in Collaborative Tasks.” In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1112–1125. DOI: 10.18653/v1/2021.emnlp-main.85. URL: https://aclanthology.org/2021.emnlp-main.85 (cit. on p. 20).

[BFC+20] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes, I. Dunning, S. Mourad, H. Larochelle, M. G. Bellemare, M. Bowling. “The Hanabi Challenge: A New Frontier for AI Research.” In: *Artificial Intelligence* 280 (Mar. 2020), p. 103216. ISSN: 0004-3702. DOI: 10.1016/j.artint.2019.103216. (Visited on 08/21/2023) (cit. on p. 16).

[BJST17] C. L. Baker, J. Jara-Ettinger, R. Saxe, J. B. Tenenbaum. “Rational Quantitative Attribution of Beliefs, Desires and Percepts in Human Mentalizing.” In: *Nature Human Behaviour* 1.4 (Mar. 2017), p. 0064. ISSN: 2397-3374. DOI: 10.1038/s41562-017-0064. (Visited on 09/06/2023) (cit. on pp. 12, 19, 20).

[BKH16] J. L. Ba, J. R. Kiros, G. E. Hinton. “Layer normalization.” In: *arXiv preprint arXiv:1607.06450* (2016) (cit. on p. 31).

[BLF85] S. Baron-Cohen, A. M. Leslie, U. Frith. “Does the Autistic Child Have a "Theory of Mind"?” In: *Cognition* 21.1 (Oct. 1985), pp. 37–46. ISSN: 0010-0277. DOI: 10.1016/0010-0277(85)90022-8 (cit. on pp. 12, 19).

[Bos14] N. Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford: Oxford University Press, 2014 (cit. on p. 21).

[BSF+20] A. Bobu, D. R. R. Scobee, J. F. Fisac, S. S. Sastry, A. D. Dragan. “LESS Is More: Rethinking Probabilistic Models of Human Behavior.” In: *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. Cambridge United Kingdom: ACM, Mar. 2020, pp. 429–437. ISBN: 978-1-4503-6746-2. DOI: 10.1145/3319502.3374811. (Visited on 09/27/2023) (cit. on p. 13).

[CMBB14] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio. “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches.” In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. DOI: 10.3115/v1/W14-4012. URL: https://aclanthology.org/W14-4012 (cit. on p. 32).

[CMD20] R. Charakorn, P. Manoonpong, N. Dilokthanakul. “Investigating partner diversification methods in cooperative multi-agent deep reinforcement learning.” In: *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 18–22, 2020, Proceedings, Part V 27*. Springer. 2020, pp. 395–402 (cit. on p. 17).

[CSH+19] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, A. Dragan. “On the Utility of Learning about Humans for Human-AI Coordination.” In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf (cit. on pp. 13, 14, 16, 17, 19, 34, 35, 38).

[CSHD20] R. Choudhury, G. Swamy, D. Hadfield-Menell, A. D. Dragan. “On the Utility of Model Learning in HRI.” In: *Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’19. Daegu, Republic of Korea: IEEE Press, 2020, pp. 317–325. ISBN: 9781538685556 (cit. on p. 12).

[DHB+20] A. Dafoe, E. Hughes, Y. Bachrach, T. Collins, K. R. McKee, J. Z. Leibo, K. Larson, T. Graepel. “Open Problems in Cooperative AI.” In: *CoRR* abs/2012.08630 (2020). arXiv: 2012.08630. URL: https://arxiv.org/abs/2012.08630 (cit. on p. 12).

[DQGY10] P. Doshi, X. Qu, A. Goodie, D. Young. “Modeling recursive reasoning by humans using empirically informed interactive POMDPs.” In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*. 2010, pp. 1223–1230 (cit. on p. 20).

[FGH+20] J. F. Fisac, M. A. Gates, J. B. Hamrick, C. Liu, D. Hadfield-Menell, M. Palaniappan, D. Malik, S. S. Sastry, T. L. Griffiths, A. D. Dragan. “Pragmatic-pedagogic value alignment.” In: *Robotics Research: The 18th International Symposium ISRR*. Springer. 2020, pp. 49–57 (cit. on p. 21).

[FHZ+21] M. C. Fontaine, Y.-C. Hsu, Y. Zhang, B. Tjanaka, S. Nikolaidis. “On the importance of environments in human-robot coordination.” In: *arXiv preprint arXiv:2106.10853* (2021) (cit. on p. 17).

[FRS01] K. P. F.R.S. “LIII. On lines and planes of closest fit to systems of points in space.” In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720 (cit. on p. 54).

[FVHK08] L. Flobbe, R. Verbrugge, P. Hendriks, I. Krämer. “Children’s Application of Theory of Mind in Reasoning and Language.” In: *Journal of Logic Language and Information* 17 (Jan. 2008), pp. 417–442. DOI: 10.1007/s10849-008-9064-7 (cit. on p. 22).

[FWCL21] A. Fuchs, M. Walton, T. Chadwick, D. Lange. “Theory of Mind for Deep Reinforcement Learning in Hanabi.” In: *CoRR* abs/2101.09328 (2021). arXiv: 2101.09328. URL: https://arxiv.org/abs/2101.09328 (cit. on pp. 12, 20).

[HDG+19] D. Hernandez, K. Denamganaï, Y. Gao, P. York, S. Devlin, S. Samothrakis, J. A. Walker. “A Generalized Framework for Self-Play Training.” In: *2019 IEEE Conference on Games (CoG)*. 2019, pp. 1–8. DOI: 10.1109/CIG.2019.8848006 (cit. on p. 13).

[HE16] J. Ho, S. Ermon. “Generative adversarial imitation learning.” In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 38).

[HG18] Y. Han, P. Gmytrasiewicz. “Learning others’ intentional models in multi-agent settings using interactive POMDPs.” In: *Advances in Neural Information Processing Systems* 31 (2018) (cit. on p. 20).

[HLPF20] H. Hu, A. Lerer, A. Peysakhovich, J. Foerster. “"Other-Play" for Zero-Shot Coordination.” In: *Proceedings of the 37th International Conference on Machine Learning*. ICML’20. JMLR.org, 2020 (cit. on pp. 13, 16, 34–36, 42).

[HS44] F. Heider, M. Simmel. “An Experimental Study of Apparent Behavior.” In: *The American Journal of Psychology* 57.2 (Apr. 1944), p. 243. DOI: 10.2307/1416950. URL: https://doi.org/10.2307/1416950 (cit. on p. 19).

[HZ02] T. Hedden, J. Zhang. “What Do You Think I Think You Think?: Strategic Reasoning in Matrix Games.” In: *Cognition* 85.1 (2002), pp. 1–36. ISSN: 1873-7838. DOI: 10.1016/S0010-0277(02)00054-9 (cit. on p. 22).

[HZAL18] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor.” In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy, A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 1861–1870. URL: https://proceedings.mlr.press/v80/haarnoja18b.html (cit. on p. 29).

[JMC+17] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, K. Kavukcuoglu. “Reinforcement Learning with Unsupervised Auxiliary Tasks.” In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=SJ6yPD5xg (cit. on p. 20).

[KB14] D. P. Kingma, J. Ba. “Adam: A method for stochastic optimization.” In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on p. 32).

[KCD+21] P. Knott, M. Carroll, S. Devlin, K. Ciosek, K. Hofmann, A. Dragan, R. Shah. “Evaluating the Robustness of Collaborative Agents.” In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS ’21. Virtual Event, United Kingdom: International Foundation for Autonomous Agents and Multiagent Systems, 2021, pp. 1560–1562. ISBN: 9781450383073 (cit. on pp. 12, 16, 17, 20, 34, 35, 37–39, 46, 47, 49, 59, 61).

[KHA+16] M. Kleiman-Weiner, M. Ho, J. Austerweil, M. Littman, J. Tenenbaum. “Coordinate to Cooperate or Compete: Abstract Goals and Joint Intentions in Social Interaction.” In: Jan. 2016 (cit. on pp. 16, 20).

[KHT19] B. Kartal, P. Hernandez-Leal, M. E. Taylor. “Terminal Prediction as an Auxiliary Task for Deep Reinforcement Learning.” In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 15.1 (Oct. 2019), pp. 38–44. DOI: 10.1609/aiide.v15i1.5222. URL: https://ojs.aaai.org/index.php/AIIDE/article/view/5222 (cit. on p. 20).

[KT99] V. Konda, J. Tsitsiklis. “Actor-Critic Algorithms.” In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, K. Müller. Vol. 12. MIT Press, 1999 (cit. on p. 31).

[KWB+04] G. Klien, D. Woods, J. Bradshaw, R. Hoffman, P. Feltovich. “Ten challenges for making automation a "team player" in joint human-agent activity.” In: *IEEE Intelligent Systems* 19.6 (2004), pp. 91–95. DOI: 10.1109/MIS.2004.74 (cit. on p. 12).

[KZGR23] R. Kirk, A. Zhang, E. Grefenstette, T. Rocktäschel. “A Survey of Zero-shot Generalisation in Deep Reinforcement Learning.” In: *Journal of Artificial Intelligence Research* 76 (Jan. 2023), pp. 201–264. ISSN: 1076-9757. DOI: 10.1613/jair.1.14174. arXiv: 2111.09794 [cs]. (Visited on 09/27/2023) (cit. on pp. 13, 16).

[LC17] G. Lample, D. S. Chaplot. “Playing FPS Games with Deep Reinforcement Learning.” In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI’17. San Francisco, California, USA: AAAI Press, 2017, pp. 2140–2146 (cit. on p. 20).

[LCCS22] C. Langley, B. I. Cirstea, F. Cuzzolin, B. J. Sahakian. “Theory of Mind and Preference Learning at the Interface of Cognitive Science, Neuroscience, and AI: A Review.” In: *Frontiers in Artificial Intelligence* 5 (2022). ISSN: 2624-8212. (Visited on 09/06/2023) (cit. on p. 12).

[LKS+22] L. L. D. Langosco, J. Koch, L. D. Sharkey, J. Pfau, D. Krueger. “Goal Misgeneralization in Deep Reinforcement Learning.” In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 12004–12019. URL: https://proceedings.mlr.press/v162/langosco22a.html (cit. on p. 21).

[LPB17] A. Lazaridou, A. Peysakhovich, M. Baroni. “Multi-Agent Cooperation and the Emergence of (Natural) Language.” In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=Hk8N3Sclg (cit. on p. 16).

[LZL+23] A. Liu, H. Zhu, E. Liu, Y. Bisk, G. Neubig. “Computational Language Acquisition with Theory of Mind.” In: *arXiv preprint arXiv:2303.01502* (2023) (cit. on p. 20).

[MPV+17] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, R. Hadsell. *Learning to Navigate in Complex Environments*. Jan. 2017. DOI: 10.48550/arXiv.1611.03673. arXiv: 1611.03673 [cs]. (Visited on 09/08/2023) (cit. on p. 20).

[NGS+21] P. Nalepka, J. Gregory-Dunsmore, J. Simpson, G. Patil, M. Richardson. “Interaction flexibility in artificial agents teaming with humans.” English. In: *CogSci 2021: program for the 43rd Annual Meeting of the Cognitive Science Society*. Annual Meeting of the Cognitive Science Society (43rd : 2021), CogSci 2021 ; Conference date: 26-07-2021 Through 29-07-2021. Cognitive Science Society, 2021, pp. 112–118 (cit. on p. 17).

[NNL+22] D. Nguyen, P. Nguyen, H. Le, K. Do, S. Venkatesh, T. Tran. “Learning Theory of Mind via Dynamic Traits Attribution.” In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. AAMAS ’22. Virtual Event, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems, 2022, pp. 954–962. ISBN: 9781450392136 (cit. on p. 20).

[NNL+23] D. Nguyen, P. Nguyen, H. Le, K. Do, S. Venkatesh, T. Tran. “Memory-Augmented Theory of Mind Network.” In: *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI’23/IAAI’23/EAAI’23. AAAI Press, 2023. ISBN: 978-1-57735-880-0. DOI: 10.1609/aaai.v37i10.26374. URL: https://doi.org/10.1609/aaai.v37i10.26374 (cit. on p. 20).

[PJB20] S. Padakandla, P. K. J, S. Bhatnagar. “Reinforcement Learning in Non-Stationary Environments.” In: *Applied Intelligence* 50.11 (Nov. 2020), pp. 3590–3606. ISSN: 0924-669X, 1573-7497. DOI: 10.1007/s10489-020-01758-5. arXiv: 1905.03970 [cs, stat]. (Visited on 09/11/2023) (cit. on p. 20).

[Pom91] D. A. Pomerleau. “Efficient Training of Artificial Neural Networks for Autonomous Navigation.” In: *Neural Computation* 3.1 (Mar. 1991), pp. 88–97. ISSN: 0899-7667. DOI: 10.1162/neco.1991.3.1.88 (cit. on p. 38).

[PW78] D. Premack, G. Woodruff. “Does the Chimpanzee Have a Theory of Mind?” In: *Behavioral and Brain Sciences* 1.4 (Dec. 1978), pp. 515–526. ISSN: 1469-1825, 0140-525X. DOI: 10.1017/S0140525X00076512. (Visited on 08/21/2023) (cit. on pp. 12, 19).

[RKCW18] C. Resnick, I. Kulikov, K. Cho, J. Weston. “Vehicle Community Strategies.” In: *CoRR* abs/1804.07178 (2018). arXiv: 1804.07178. URL: http://arxiv.org/abs/1804.07178 (cit. on p. 16).

[RMSM23] J. G. Ribeiro, C. Martinho, A. Sardinha, F. S. Melo. “Making Friends in the Dark: Ad Hoc Teamwork Under Partial Observability.” In: *26th European Conference on Artificial Intelligence ECAI 2023*. 2023. DOI: 10.3233/FAIA230486 (cit. on p. 17).

[RPS+18] N. Rabinowitz, F. Perbet, F. Song, C. Zhang, S. M. A. Eslami, M. Botvinick. “Machine Theory of Mind.” In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy, A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 4218–4227. URL: https://proceedings.mlr.press/v80/rabinowitz18a.html (cit. on pp. 12, 20, 22, 54).

[RZP+22] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-maron, M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, N. de Freitas. “A Generalist Agent.” In: *Transactions on Machine Learning Research* (2022). Featured Certification, Outstanding Certification. ISSN: 2835-8856. URL: https://openreview.net/forum?id=1ikK0kHjvj (cit. on p. 13).

[SC13] R. M. Seyfarth, D. L. Cheney. “Affiliation, Empathy, and the Origins of Theory of Mind.” In: *Proceedings of the National Academy of Sciences* 110 (June 2013), pp. 10349–10356. DOI: 10.1073/pnas.1301223110. (Visited on 09/27/2023) (cit. on p. 12).

[She16] T. B. Sheridan. “Human-Robot Interaction: Status and Challenges.” In: *Human Factors* 58.4 (June 2016), pp. 525–532. ISSN: 1547-8181. DOI: 10.1177/0018720816644364 (cit. on p. 13).

[SHM+16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis. “Mastering the game of Go with deep neural networks and tree search.” In: *Nature* 529 (2016), pp. 484–503. URL: http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html (cit. on p. 13).

[SHS+17] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. P. Lillicrap, K. Simonyan, D. Hassabis. “Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm.” In: *CoRR* abs/1712.01815 (2017). arXiv: 1712.01815. URL: http://arxiv.org/abs/1712.01815 (cit. on p. 13).

[SKKR10] P. Stone, G. A. Kaminka, S. Kraus, J. S. Rosenschein. “Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination.” In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. AAAI’10. Atlanta, Georgia: AAAI Press, 2010, pp. 1504–1509 (cit. on p. 16).

[SMB+21] D. Strouse, K. McKee, M. Botvinick, E. Hughes, R. Everett. “Collaborating with Humans without Human Data.” In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 14502–14515. URL: https://proceedings.neurips.cc/paper/2021/file/797134c3e42371bb4979a462eb2f042a-Paper.pdf (cit. on pp. 13, 17, 18, 34–36).

[SMD+11] R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, D. Precup. “Horde: A Scalable Real-Time Architecture for Learning Knowledge from Unsupervised Sensorimotor Interaction.” In: *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*. AAMAS ’11. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, May 2011, pp. 761–768. ISBN: 978-0-9826571-6-4. (Visited on 09/08/2023) (cit. on pp. 20, 25).

[SNB22] M. Sclar, G. Neubig, Y. Bisk. “Symmetric Machine Theory of Mind.” In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 19450–19466 (cit. on pp. 12, 20, 22, 24).

[SR13] A. Stefano, S. Ramamoorthy. “A game theoretic model and best-response learning method for ad hoc coordination in multi-agente systems.” In: *Proc. of the 12th International Conference on Autonomous Agents and Multiagent Systems*. 2013 (cit. on p. 16).

[SSF16] S. Sukhbaatar, A. Szlam, R. Fergus. “Learning Multiagent Communication with Backpropagation.” In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS’16. Red Hook, NY, USA: Curran Associates Inc., Dec. 2016, pp. 2252–2260. ISBN: 978-1-5108-3881-9. (Visited on 09/11/2023) (cit. on p. 16).

[SSS+17] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, D. Hassabis. “Mastering the game of Go without human knowledge.” In: *Nature* 550.7676 (Oct. 2017), pp. 354–359. DOI: 10.1038/nature24270. URL: https://doi.org/10.1038/nature24270 (cit. on p. 13).

[STSD22] B. Sarkar, A. Talati, A. Shih, S. Dorsa. “PantheonRL: A MARL Library for Dynamic Training Interactions.” In: *Proceedings of the 36th AAAI Conference on Artificial Intelligence (Demo Track)*. 2022 (cit. on p. 17).

[SWD+17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov. “Proximal Policy Optimization Algorithms.” In: *CoRR* abs/1707.06347 (2017). arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347 (cit. on pp. 24, 26).

[Tes94] G. Tesauro. “TD-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play.” In: *Neural Computation* 6.2 (Mar. 1994), pp. 215–219. ISSN: 0899-7667. DOI: 10.1162/neco.1994.6.2.215. (Visited on 09/11/2023) (cit. on pp. 13, 16).

[TK74] A. Tversky, D. Kahneman. “Judgment under Uncertainty: Heuristics and Biases.” In: *Science* 185.4157 (1974), pp. 1124–1131. DOI: 10.1126/science.185.4157.1124. eprint: https://www.science.org/doi/pdf/10.1126/science.185.4157.1124. URL: https://www.science.org/doi/abs/10.1126/science.185.4157.1124 (cit. on pp. 17, 21).

[Var06] H. R. Varian. “Revealed preference.” In: *Samuelsonian economics and the twenty-first century* (2006), pp. 99–115 (cit. on p. 23).

[YDF08] W. Yoshida, R. J. Dolan, K. J. Friston. “Game Theory of Mind.” In: *PLOS Computational Biology* 4.12 (Dec. 2008), e1000254. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1000254. (Visited on 09/27/2023) (cit. on pp. 12, 19).

[YFZ+21] L. Yuan, Z. Fu, L. Zhou, K. Yang, S. Zhu. “Emergence of Theory of Mind Collaboration in Multiagent Systems.” In: *CoRR* abs/2110.00121 (2021). arXiv: 2110.00121. URL: https://arxiv.org/abs/2110.00121 (cit. on pp. 12, 20, 27).

[YGL+23] C. Yu, J. Gao, W. Liu, B. Xu, H. Tang, J. Yang, Y. Wang, Y. Wu. “Learning Zero-Shot Cooperation with Humans, Assuming Humans Are Biased.” In: *The Eleventh International Conference on Learning Representations*. 2023 (cit. on pp. 13, 16, 17, 19, 24, 30, 32–37, 42–45, 50, 52, 61, 62).

[YSP+22] M. Yu, Y. Sang, K. Pu, Z. Wei, H. Wang, J. Li, Y. Yu, J. Zhou. “Few-Shot Character Understanding in Movies as an Assessment to Meta-Learning of Theory-of-Mind.” In: *ArXiv* abs/2211.04684 (2022) (cit. on pp. 20, 32).

[ZSY+23] R. Zhao, J. Song, Y. Yuan, H. Hu, Y. Gao, Y. Wu, Z. Sun, W. Yang. “Maximum Entropy Population-Based Training for Zero-Shot Human-AI Coordination.” In: *Proceedings of the AAAI Conference on Artificial Intelligence* 37.5 (June 2023), pp. 6145–6153. DOI: 10.1609/aaai.v37i5.25758. URL: https://ojs.aaai.org/index.php/AAAI/article/view/25758 (cit. on pp. 17, 18, 29, 30).

[ZZH+23] P. Zhou, A. Zhu, J. Hu, J. Pujara, X. Ren, C. Callison-Burch, Y. Choi, P. Ammanabrolu. "I Cast Detect Thoughts: Learning to Converse and Guide with Intents and Theory-of-Mind in Dungeons and Dragons." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 11136–11155. DOI: 10.18653/v1/2023.acl-long.624. URL: https://aclanthology.org/2023.acl-long.624 (cit. on p. 20).

All links were last followed on October 15, 2023.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature