PROBING FOR THEORY OF MIND IN THE MULTI-AGENT COOPERATION GAME YOKAI

Jonas Allali
B.Sc. Computer Science
st116462@stud.uni-stuttgart.de
University of Stuttgart

ABSTRACT

Training artificial intelligence to master collaborative games presents significant challenges, especially when success relies on understanding and predicting the intentions of other players, a capacity often referred to as Theory of Mind (ToM). One such game is Yokai, a cooperative card game where players must coordinate to achieve a common goal with limited communication. This paper explores the ability of a large language model (LLM) to play Yokai, emphasizing the understanding of the model for Theory of Mind by analyzing the inner state representation through sophisticated evaluation techniques such as probing. Additionally, we compare this approach with Multi-Agent Reinforcement Learning (MARL) where multiple agents learn to cooperate through trial and error and adjusting their strategies based on the rewards received from their interactions, highlighting the strengths and limitations of each method.

1 INTRODUCTION

Creating machine systems that can perform advanced social reasoning in a human-like fashion is a key goal in artificial intelligence. Essential to this effort is providing these systems with a ”Theory of Mind” (ToM) capability. This means they can recognize and attribute mental states—such as beliefs, desires, intentions, and emotions—to themselves and others, while understanding that others may have different mental states Leslie (1987). This core ability is vital not only for effectively navigating human social interactions but also for allowing machines to demonstrate cooperative, adaptive, and empathetic behaviors in a variety of social settings Kleiman-Weiner et al. (2016).

Rabinowitz et al. (2018) expanded this issue into the realm of artificial intelligence, introducing the term Machine Theory of Mind (MToM). In MToM, artificial agents are designed to deduce the mental states of other agents. There are many potential applications for MToM. Learning rich models of others will improve decision-making in complex multi-agent tasks, especially where model-based planning and imagination are required Mobbs et al. (2013). One such field is the realm of games. In recent years, machine learning has made dramatic advances with artificial agents reaching superhuman performance in challenge domains like Go, Atari, and some variants of poker using sophisticated methods of reinforcement learning Silver et al. (2021). However, many of these areas involve the presence of at least one additional agent, increasing the complexity of the model. In particular, Multi-Agent Reinforcement Learning (MARL) deals with the sequential decision-making challenges of multiple autonomous agents. Each agent seeks to optimize its long-term reward by interacting with other agents within a shared environment Busoniu et al. (2008).

In contrast to previous works about games like Overcooked by Carroll et al. (2019) or Hanabi Bard et al. (2020) which mainly focused on reinforcement learning this work focuses on an alternative way to play games, namely through large language models, which got recently a lot of attention because of the widely-spread ChatGPT by OpenAI. The test subject in this work is going to be Yokai, a cooperative game which include spatial and temporal reasoning.

First we build a text-interface for a Yokai environment in Python by fully implementing the game. Then, with the pre-trained LLM *Pythia* we hand-craft suitable inputs to play the game. The main focus of this work is to answer the question whether a LLM is able to understand the Yokai game. Therefore it is crucial for the model to apply theory of mind.

Probing is a new technique introduced by Alain & Bengio (2016) to make visible how neural networks represent their belief of the current state which will be used in this work to fully comprehend the way a LLM would deal with this task in detail. At the end of this work we compare the findings with reinforcements methods to see if there might be a performance difference.

2 Related Work

2.1 (Machine) Theory of Mind

Theory of Mind (ToM) is the concept of understanding and mapping the beliefs and goals of other agents participating in the same environment. This term was first introduced in the cognitive psychology field by Heider & Simmel (1944). Further work by Baron-Cohen et al. (1985) presented the so-called *Sally-Anne test*, which measures a person's social cognitive ability to attribute false beliefs in others. This test is especially helpful in finding early signs of autism in children. During the test a child gets to observe a room with two dolls named Sally and Anne (later they are replaced by humans by Leslie & Frith (1988)), also there is a basket and a box in the room as depicted in figure 2.1. Then Sally places a marble in the basket and leaves the room. Now Anne puts Sally's marble from the basket to the box. To test the child's ability of ToM it now gets asked where Sally will look for her marble when reentering the room.

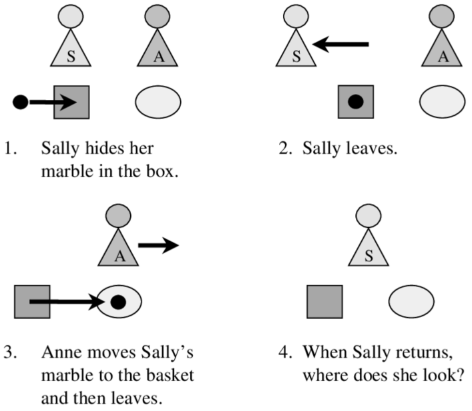

S
A
1. Sally hides her marble in the box.
2. Sally leaves.
3. Anne moves Sally's marble to the basket and then leaves.
4. When Sally returns, where does she look?

Figure 1: The Sally-Anne based on Baron-Cohen et al. (1985)

ToM becomes relevant especially in games with imperfect information like Hanabi and Yokai, as identified by Bard et al. (2020) and Fuchs et al. (2021). In these games, players rely on ToM to interpret others' actions and hints. In Hanabi, intent reasoning (comprehending and anticipating others' actions based on their intentions) and belief reasoning (understanding and predicting others' behaviors based on their beliefs) are dominant factors, whereas in Yokai, additional factors such as temporal reasoning (understanding and predicting others' actions in relation to time) and spatial reasoning (understanding and anticipating how others perceive and interact with their physical space) are essential as stated by Fernandez et al. (2023). Rabinowitz et al. (2018) extended this problem to the field of artificial intelligence, coining the term Machine Theory of Mind (MToM). In MToM artificial agents are used to infer the mental states of other agents, which alos can be used to pass the Sally-Anne test. However, the ability of LMMs to reason about the beliefs of other agents remains limited as shown by Sap et al. (2023), which will be further examined in this work.

2.2 Large Language Models

Recently, LMMs drew a lot of attention because of the release of Chat-GPT in 2023, which had a huge impact on several academic and non-academic sectors. A LLM is known for its ability to process large amounts of data and generate content in the area of Natural Language Processing (NLP), which is acquired through learning statistical relationships from the data in training steps Radford et al. (2019). Chat-GPT is built of Generative-pretrained transformers (GPT) which are a type of LLMs consisting of the so-called transformer architecture Radford et al. (2018). Transformers were introduced with the famous paper "Attention is all you need" by Vaswani et al. Vaswani et al. (2017).

In figure 2.2 a basic transformer architecture is displayed. It consists of an encoder on the left and of a decoder on the right. First, the encoder maps an input sequence $x = (x_1, ..., x_n)$ to a sequence of continuous representations $z = (z_1, ..., z_n)$. Then, the decoder generates an output sequence $y = (y_1, ..., y_n)$. Thereby, the following models are used:

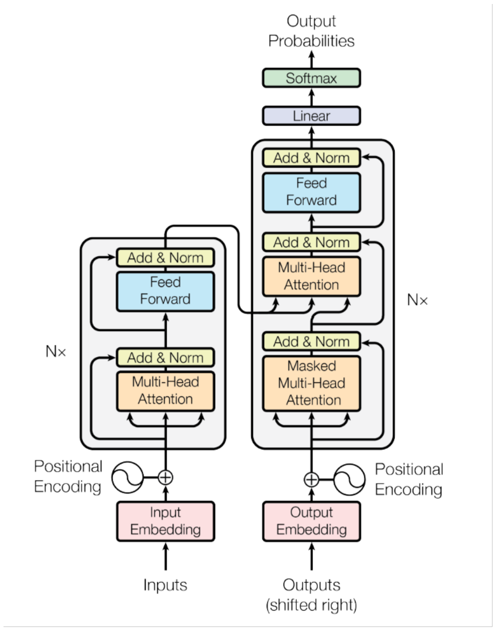


Figure 2: Basic Transformer

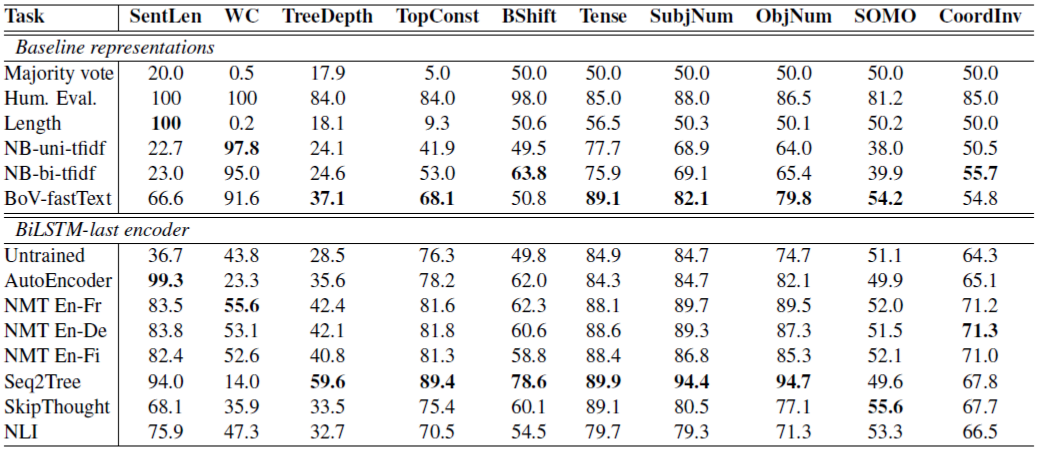
- **Word Embeddings**: Convert input tokens into vectors of dimension d_model through a embedding network
- **Positional Encodings**: Injects information about the relative or absolute position if the tokens in the sequence
- **Self-Attention**: Keeps track of word relationships within the input and output phrases
- **Encoder-Decoder Attention**: makes sure important words in the input are not lost between input and output phrases
- **Residual Connections**: allows each sub-unit, like Self-Attention, to focus on solving just one part of the problem

It is important to note that in 2018 the more advanced BERT architecture, which only uses the encoder without the decoder, was introduced and became ubiquitous Rogers et al. (2021).

Using LLMs as agents as an alternative to methods of reinforcement learning in games is a brand-new field to explore. Ciolino et al. (2020) and Bateni & Whitehead (2024) showed that LLMs can be used in games like Go or ”Slay the Spire” to achieve satisfactory results, however a lot of research still needs to be done in this regard.

2.3 Probing Neural Representations

Probing is a technique in the field of Explainable AI which gives us more insight in how a neural network represents information during the whole process. Probes are supervised models trained to predict properties, like for example sentence length used in machine translation (Hewitt & Liang (2019)). Here a probe would be trained on the sentence embeddings in the neural network by tagging each embedding to be short or long. This way we can understand whether the information of sentence length is lost during the process or is still to be found in the data. Conneau et al. (2018) from Facebook AI Research used ten different such properties to inspect word embeddings. For the probing networks they used a bidirectional LSTM (BiLSTM) and Gated ConvNet which are based on stacked gated temporal convolutions, with different training tasks such as Neural Machine Translation (NMT) or Seq2seq. The goal is to examine whether neural networks encode information of certain properties throughout their processing. In figure 2.3 we find an extract of the probing results which shows that for example the property sentence length can be found in our model whereas word content (WC) is not that easy to extract.



Task	SentLen	WC	TreeDepth	TopConst	BShift	Tense	SubjNum	ObjNum	SOMO	CoordInv
Baseline representations										
Majority vote	20.0	0.5	17.9	5.0	50.0	50.0	50.0	50.0	50.0	50.0
Hum. Eval.	100	100	84.0	84.0	98.0	85.0	88.0	86.5	81.2	85.0
Length	**100**	0.2	18.1	9.3	50.6	56.5	50.3	50.1	50.2	50.0
NB-uni-tfidf	22.7	**97.8**	24.1	41.9	49.5	77.7	68.9	64.0	38.0	50.5
NB-bi-tfidf	23.0	95.0	24.6	53.0	**63.8**	75.9	69.1	65.4	39.9	**55.7**
BoV-fastText	66.6	91.6	**37.1**	**68.1**	50.8	**89.1**	**82.1**	**79.8**	**54.2**	54.8
BiLSTM-last encoder										
Untrained	36.7	43.8	28.5	76.3	49.8	84.9	84.7	74.7	51.1	64.3
AutoEncoder	**99.3**	23.3	35.6	78.2	62.0	84.3	84.7	82.1	49.9	65.1
NMT En-Fr	83.5	**55.6**	42.4	81.6	62.3	88.1	89.7	89.5	52.0	71.2
NMT En-De	83.8	53.1	42.1	81.8	60.6	88.6	89.3	87.3	51.5	**71.3**
NMT En-Fi	82.4	52.6	40.8	81.3	58.8	88.4	86.8	85.3	52.1	71.0
Seq2Tree	94.0	14.0	**59.6**	**89.4**	**78.6**	**89.9**	**94.4**	**94.7**	49.6	67.8
SkipThought	68.1	35.9	33.5	75.4	60.1	89.1	80.5	77.1	**55.6**	67.7
NLI	75.9	47.3	32.7	70.5	54.5	79.7	79.3	71.3	53.3	66.5

Figure 3: Probing Results

In this work the probing technique is used to find representations of Yokai states in language models to comprehend if a LLM is able to fully understand the game.

2.4 Multi-Agent Reinforcement Learning (MARL)

In recent years reinforcement learning was very successful in sequential decision-making problems, e.g in games such as Go and Poker, robotics or autonomous driving Zhang et al. (2021). However, many of these fields involve the participation of at least one additional agent, adding more complexity to the model. Specifically, Multi-Agent Reinforcement Learning (MARL) addresses the sequential decision-making problem of multiple autonomous agents, each of which optimizes its longterm reward by interacting with the other agents in a shared environment Busoniu et al. (2008). A Theory-of-Mind agent can be described as an adaptation of the standard multi-agent reinforcement learning (RL) framework, where agents' policies are influenced by their beliefs about other agents. Formally, a reinforcement learning problem M is defined by a tuple consisting of a state space S, action space A, state transition probability function $T \in S \times A \times S \to [0, 1]$, and reward $R \in S \times A \to \mathbb{R}$, i.e., $M := (S, A, T, R)$. In this framework, an agent learns a policy $\pi : S \to A$, which may be probabilistic, mapping states to actions to maximize its reward (Sclar et al. (2022)).

In a multi-agent RL context, each agent might have distinct state spaces, action spaces, transition probabilities, and reward functions, leading to the definition of an instance $M_i = (S_i, A_i, T_i, R_i)$ for each agent i. For convenience, a joint state space $S = \bigcup_i S_i$ can be defined to represent the entire environment where all agents interact. Crucially, each agent will have its own perspective of the whole world, described by a conditional observation function $\omega : S \to \Omega$ that maps from the state of the entire environment to the information observable by agent i (Sclar et al. (2022)).

MARL can be further categorized in *fully cooperative*, *fully competitive* and *a mix of the two*. In the cooperative category all agents collaborate to optimize a common long-term reward, as it is the case in the Yokai game, whereas in the competitive setting the return of the agents usually sum up to zero.

In this work MARL will be used to compare its performance to the usage of LLMs in this scenario.

2.5 YOKAI AS BENCHMARK

Yokai is a cooperative card game which can be played by two to four players. There are four card types represented by different colors and the goal is to find a configuration of face-down cards, so that all colors are grouped together (for every card there is a connection to every other card with the same color).

2.5.1 MOTIVATION

In this work, we introduce Yokai as a novel and challenging benchmark within the LLM and MARL domain. Much like Hanabi, Yokai is a cooperative game that operates under imperfect information. However, what sets Yokai apart is the need for additional reasoning styles, particularly temporal and spatial reasoning. Theory of Mind (ToM) plays a crucial role in Yokai since direct communication of card information is forbidden. Players must utilize intention reasoning to interpret teammates' actions, belief reasoning to deduce what others know about the game state, and temporal reasoning to remember past observations. Additionally, spatial reasoning is critical for determining current card positions and potential moves based on the layout (Fernandez et al. (2023)). These cognitive processes are essential for strategic coordination and achieving cooperative objectives in Yokai. This increased complexity makes Yokai a valuable benchmark for testing existing and future methods in MARL research.

2.5.2 COMPONENTS

The basic game consists of 16 Yokai cards which are divided into 4 colors (red, yellow, blue and green) and there are 14 hint cards which are used for communication between the players.

2.5.3 SETUP

The Yokai cards are arranged in a 4x4 grid face-down in a random order. Cards always have to be connected by at least one edge.

Furthermore, a deck of hint cards are placed nearby. There are three types of hint cards, which consist of either one, two or three colors. The seize of the deck depends on the number of the players as shown in this Table:

Number of 1-color hint cards	2	2	3
Number of 2-color hint cards	3	4	4
Number of 3-color hint cards	2	3	3

2.5.4 OBJECTIVE

The goal is to find a configuration of face-down Yokai cards, so that all colors are grouped together. A configuration can be seen as a graph where the cards are nodes and neighbouring nodes are connected. Figure 4 shows an illegal move which results in two separate card groups. A winning configuration is present if for every color there exist a path which reaches every card of that color and only traverses nodes of that color. An example winning configuration can be seen in figure 4.

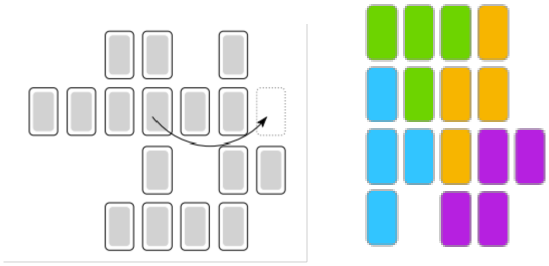


Figure 4: Left: Illegal Move // Right: Winning Configuration

2.5.5 Gameplay

Each player plays the game in a sequential order and on every turn the player has to perform the following actions in this order:

1. Look at two Yokai cards.
2. Choose any Yokai card on the grid and move it to a different location. At the end of this step every card still has to be connected by at least one edge with another card. An example for a wrong move is shown in figure. This move violates the setup condition because it creates two groups of cards which are not connected by at least one edge.
3. Choose between revealing a face-down hint card or placing a face-up hint card on top of a Yokai card. All revealed cards need to be accessible and should not be stacked. A hint placed on a Yokai card should match one of its colors with the Yokai card and locks it down (further viewing or moving is prevented this way).

2.5.6 End Game

The game may end in two ways:

1. A player declares that all colors are grouped together instead of playing their turn
2. The last hint card has been used

Once the game is over, reveal all Yokai cards and check if the hint card colors match with their corresponding Yokai card. If the winning configuration is achieved and all Yokai cards are grouped together the players win the game, otherwise they lose.

2.5.7 Scoring

If a winning configuration is achieved the final score is calculated with this formula: $score = x_1 - x_2 + 2x_3 + 5x_4$ where x_1 is the number of correctly used hint cards, x_2 is the number of wrongly used hint cards, x_3 is the number face-up and unused hint cards and x_4 is the number of face-down hint cards. The final score is int he interval [-7, 35].

3 Key Novelty and Contribution

- **Introduction of Yokai in LLMs and MARL**: We propose Yokai as a new challenge in the fields of LLM and MARL
- **Development of a text-interface for a Yokai Environment**: We create a simple text-interface in Python for simulating Yokai games
- **Development of a Vectorized Environment using Jax**: We extend JaxMARL (Rutherford et al. (2023)), a MARL library written in Jax. It enables significant acceleration and parallelization over existing MARL implementations, being the first open-source library providing JAX-based implementations of a wide range of MARL environments and baselines. In this work, a novel vectorized reinforcement learning environment for Yokai is developed using Jax while extending JaxMARL.
- **Hand design inputs for LLMs and Train RL Policies**: To use a LLM to play Yokai we need to invent information efficient inputs which represent the game state for training and playing. For LLM we use the probing technique to get insight into the neural network how it stores belief systems. We additionally train RL policies in the Jax environment.
- **Performance Benchmarking**: This work conducts performance evaluations of algorithms which are applied to the LLM- and RL-Method.

4 Approach

4.1 Method

4.1.1 LLM as agent

First we build a text-interface for a Yokai environment in Python by fully implementing the Yokai card game. The environment should allow to play the game through textual input. Hence, a LLM should be able to interact with the environment and to be trained on. As for the LLM we use the pre-trained *Pythia*. To build a bridge between the text-interface and the LLM it is further necessary to hand-craft inputs (and outputs) which are processable. The goal is to play the game through the text-interface by redirecting the current state of the game (the output of the environment) to the LLM which processes the next turn to play.

4.1.2 MARL as agent

To compare the performance with state-of-the-art reinforcement learning algorithms we further propose a Yokai environment fit for MARL. Based on Rutherford et al. (2023) we will implement a custom environment using JaxMARL, specifically modeled on the Yokai card game. On it we train different RL policies and compare the findings.

4.2 Evaluation

Drawing inspiration from Bard et al. (2020) and Fernandez et al. (2023), this study's evaluative framework will be organized into two main components: self-play learning and zero-shot coordination Hu et al. (2020). The evaluation will involve agent configurations of 2 to 4 players, utilizing the following metrics:

- **Win rate**: This metric evaluates the proportion of games in which agents win the game
- **Score Curve**: This metric tracks the progression of scores throughout the training steps
- **Score Distribution**: This metric shows a thorough examination of the distribution and variance in normalized scores
- **Statistical Analysis of Scores**: Key statistical indicators, including the mean, median, and standard deviation of the normalized scores, will be computed in this metric.

Additional metrics may be incorporated in the future.

5 Intended Outcomes

The main goal of this work is to explore the ability of LLMs to understand and play the Yokai card game and compare the findings with the performance of RL methods on Yokai. The intended outcome of this master thesis include:

- build text-based and RL-based environments for Yokai
- Analyze and compare the differences between LLMs and RL policies on the Yokai card game
- Evaluate the performance of both heuristics

6 Mandatory and Optional Goals

6.1 Mandatory Goals

The mandatory goals include:

- Built a text-interface for a Yokai Environment
- Explore an LLMs ability to play Yokai

- Hand design inputs for assessing LLM's MTOM capabilities
- Explore the representations via Linear Probing
- Compare results on text benchmarks
- Train RL Policies on the Yokai environment and compare findings

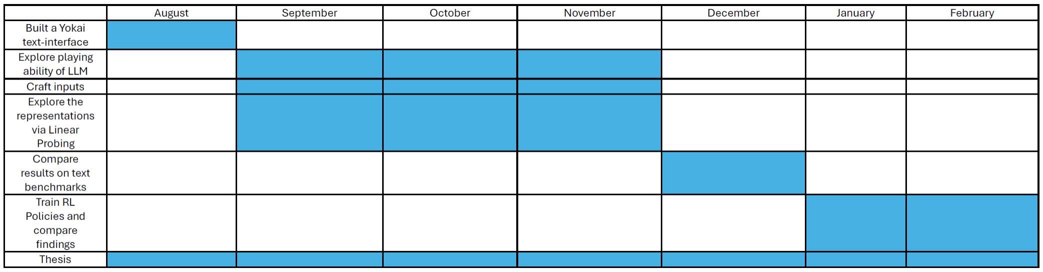
6.2 Optional Goals

Optional goals are worked on after mandatory goals are fulfilled. The optional goals include:

- Collect human-human Yokai gameplay data from a human subject study
- Evaluate trained artificial agents with humans in a zero-shot human-AI cooperative setting
- Compare findings

7 Schedule with Milestones

The thesis schedule is shown in figure 7 ,supposing the master thesis will start from the beginning of August and end in the beginning of February 2025.



	August	September	October	November	December	January	February
Built a Yokai text-interface	🟦						
Explore playing ability of LLM		🟦	🟦	🟦			
Craft inputs		🟦	🟦	🟦			
Explore the representations via Linear Probing		🟦	🟦	🟦			
Compare results on text benchmarks					🟦		
Train RL Policies and compare findings						🟦	🟦
Thesis	🟦	🟦	🟦	🟦	🟦	🟦	🟦

Figure 5: Time Schedule

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Simon Baron-Cohen, Alan M. Leslie, and Uta Frith. Does the autistic child have a "theory of mind" ? *Cognition*, 21(1):37–46, 1985. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(85)90022-8. URL `https://www.sciencedirect.com/science/article/pii/0010027785900228`.

Bahar Bateni and Jim Whitehead. Language-driven play: Large language models as game-playing agents in slay the spire. In *Proceedings of the 19th International Conference on the Foundations of Digital Games*, pp. 1–10, 2024.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019.

Matthew Ciolino, Josh Kalin, and David Noever. The go transformer: natural language modeling for game play. In *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*, pp. 23–26. IEEE, 2020.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*, 2018.

Jorge Fernandez, Dominique Longin, Emiliano Lorini, and Frédéric Maris. A logical modeling of the yōkai board game. *AI Communications*, (Preprint):1–34, 2023.

Andrew Fuchs, Michael Walton, Theresa Chadwick, and Doug Lange. Theory of mind for deep reinforcement learning in hanabi. *arXiv preprint arXiv:2101.09328*, 2021.

Fritz Heider and Marianne Simmel. An experimental study of apparent behavior. *The American Journal of Psychology*, 57(2):243–259, 1944. ISSN 00029556. URL `http://www.jstor.org/stable/1416950`.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*, 2019.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Max Kleiman-Weiner, Mark K Ho, Joseph L Austerweil, Michael L Littman, and Joshua B Tenenbaum. Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. In *CogSci*, 2016.

Alan M Leslie. Pretense and representation: The origins of" theory of mind.". *Psychological review*, 94(4):412, 1987.

Alan M Leslie and Uta Frith. Autistic children's understanding of seeing, knowing and believing. *British Journal of Developmental Psychology*, 6(4):315–324, 1988.

Dean Mobbs, Demis Hassabis, Rongjun Yu, Carlton Chu, Matthew Rushworth, Erie Boorman, and Tim Dalgleish. Foraging under competition: the neural basis of input-matching in humans. *Journal of Neuroscience*, 33(23):9866–9872, 2013.

Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. Machine theory of mind. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4218–4227. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/rabinowitz18a.html`.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2021.

Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Gardar Ingvarsson, Timon Willi, Akbir Khan, Christian Schroeder de Witt, Alexandra Souly, et al. Jaxmarl: Multi-agent rl environments in jax. *arXiv preprint arXiv:2311.10090*, 2023.

Maarten Sap, Ronan LeBras, Daniel Fried, and Yejin Choi. Neural theory-of-mind? on the limits of social intelligence in large lms, 2023.

Melanie Sclar, Graham Neubig, and Yonatan Bisk. Symmetric machine theory of mind. In *International Conference on Machine Learning*, pp. 19450–19466. PMLR, 2022.

David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021.