Representational Geometry of Social Inference and Generalization in a Competitive Game

Mitchell Ostrow
Department of Psychiatry,
Yale University
mitchell.ostrow@yale.edu

Guangyu Robert Yang†
Department of Brain and Cognitive Sciences,
Massachusetts Institute of Technology

Hyojung Seo†
Department of Psychiatry,
Yale University
† co-senior authors

***Abstract*—The use of an internal model to infer and predict others' mental states and actions, broadly referred to as Theory of Mind (ToM), is a fundamental aspect of human social intelligence. Nevertheless, it remains unknown how these models are used during social interactions, and how they help an agent generalize to new contexts. We investigated a putative neural mechanism of ToM in a recurrent circuit through the lens of an artificial neural network trained with reinforcement learning (RL) to play a competitive matching pennies game against many algorithmic opponents. The network showed near-optimal performance against unseen opponents, indicating that it had acquired the capacity to adapt against new strategies online. Analysis of recurrent states during play against out-of-training-distribution (OOD) opponents in relation to those of within-training-distribution (WD) opponents revealed two similarity-based mechanisms by which the network might generalize: mapping to a known strategy (template matching) or known opponent category (interpolation). Even when the network's strategy cannot be explained by template-matching or interpolation, the recurrent activity fell upon the low-dimensional manifold of the WD neural activity, suggesting the contribution of prior experience with WD opponents. Furthermore, these states occupied low-density edges of the WD-manifold, suggesting that the network can extrapolate beyond any learned strategy or category. Our results suggest that a neural implementation for ToM may be a reservoir of learned representations that provide the capacity for generalization via flexible access and reuse of these stored features.**

I. Introduction

Theory of Mind broadly refers to the ability of humans and perhaps other animals to reason about others' mental states [1]. This ability is particularly crucial for intelligent behavior in highly non-stationary, agent-dependent social environments where interactions often require generalization beyond prior experience [2, 3]. In these situations, people might use intuitive theories based on introspection and experience with how people generally react to external events [4]. Alternatively, people might infer optimal strategies by learning others' beliefs implicitly through iterative interactions [5]. Nevertheless, the computational and neural mechanisms of ToM are largely unknown.

In this study, we investigated a potential neural mechanism for ToM that is learned via social interactions, especially examining how this system could be used to help agents adapt to novel contexts. Computational models that interact in a social environment can provide important insights into the neural mechanism of ToM [3, 6]. Interactive models of ToM have been designed in the field of multi-agent RL for both competitive and cooperative environments, but this has traditionally focused on engineering a more successful agent [7–13]. Previous computational studies seeking to understand ToM have not included the capacity to interact [14–16], instead focusing on models that predict agent behavior, one hypothesized function of ToM [17, 18]. Importantly, an interactive agent needs to consider the consequences of its actions. This is taken into account in RL, where the reward function looks forward into future states, for example by utilizing reward-to-go or bootstrapping with a value function [19]. These features should play a significant role in shaping neural representations. Hence, interactive agents may learn more naturalistic representations than predictive models. Moreover, studies on the neural basis of ToM have identified that interactive RL models align well with functional imaging activity during a social simulation task [20], suggesting that predictive ToM can develop by repurposing systems trained for social interactions [21, 22].

To gain insights into how ToM might emerge from the dynamics of neural networks, we analyzed recurrent neural networks (RNNs) trained with meta-RL on iterative Matching Pennies (MP). Iterative MP provides a versatile platform to simulate social interactions based on various types of belief learning [5, 23], and RL provides an interactive learning framework that connects to known computational and neural mechanisms underlying behavior in iterative MP [19, 24–26]. A previous study identified that the hidden layers of a purely feedforward Deep RL agent trained to play a cooperative game learned to represent the other agent's intention, but their agent could not generalize to novel collaborators [27]. We hypothesized that training our recurrent network to play against an array of distinctive algorithmic opponents would improve generalization by encouraging the network to learn a broader representation of opponent behavior [3, 28]. Meta-learning ("learning to learn") further improves generalization ability by teaching the network to self-correct from feedback using only recurrent neural dynamics, a potential mechanism of the prefrontal cortex as suggested by [29].

We asked how the geometry of the RNN's state space supports adaptive behavior in iterative MP. To investigate how the network responds to novel social interactions, we asked

Opponent	Stochastic	Dependent on Self	Dependent on Agent	Adaptive	Optimal P(Reward)
MP 1	x		x	x	0.5
MP 1+2	x	x	x	x	0.5
LC	x		x		>0.5
PB		x			1.0
AB	x				p
SQL	x		x	x	> 0.5
ϵ-QL			x	x	$1-\epsilon$
MC			x		1.0

TABLE I
OPPONENT STRATEGIES AND FEATURES. AN 'X' INDICATES THAT THE OPPONENT'S CHOICE STRATEGY HAS THIS CHARACTERISTIC.

how the geometry of the recurrent state changed when the agent played against novel opponents.

II. METHODS

A. Matching Pennies Task and Opponents

Matching Pennies pits two players against one another in an iterative, binary-choice game. At each step, both players pick one of two 'pennies'. One player wins if they select the same penny, and the other wins if they select different pennies. The winner receives a fixed reward of arbitrary size, and the loser receives zero. Despite this simplistic task structure, MP requires subjects to learn complex, time-varying strategies to outsmart their opponent. We defined multiple classes of opponents with qualitatively different behaviors, and simulated the MP task in NeuroGym (neurogym.github.io). In our simulation, each trial consists of a single step of choice immediately followed by outcome. We briefly summarize key qualities of the eight opponent classes in table I. There are 503 unique strategies total in the eight classes.

Our first two opponents were originally defined to test how macaques learn to compete against an opponent that detects and exploits statistical biases in their recent choice history [26, 30]. We refer to these algorithms as the **Matching Pennies algorithms (MP 1 and MP 1+2)**, which use choice history and choice and outcome history, respectively. Against these opponents, random choices are optimal according to a mixed-strategy Nash equilibrium [26]. The **Linear Combination (LC)** opponents use a stochastic strategy defined by a linear combination of past choices and choice-outcome interaction:

$$\text{logit}(P(c_{t+1}=1)) = b_0 + \mathbf{b}_1^T \mathbf{c}_{t:t-n} + \mathbf{b}_2^T \mathbf{r}_{t:t-n} \odot \mathbf{c}_{t:t-n} \quad (1)$$

Where $b_0 \in \mathbb{R}, \mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^n$ are coefficients with parameters randomly selected from a predefined set ranging from -2 to 2, and the reward and choice history vectors $\mathbf{r}_{t:t-n}, \mathbf{c}_{t:t-n} \in \{-1, 1\}^n$. $\odot$ indicates element-wise product. We defined n to be small, ranging from 2 to 4. The **fourth algorithm (Pattern-Based, PB)** deterministically makes choices according to a binary pattern of length $n \in [1, 6]$. If the agent can identify the pattern, it can play perfectly.

The **fifth algorithm (anti-correlated bandit, AB)** is a classic two-armed bandit task [19], with anti-correlated reward probabilities $\{p, 1-p\}$ that are randomly resampled every 50-75 trials. The next two opponents are tabular Q-Learning RL agents with different exploration processes: the first uses the SoftMax function to determine the policy distribution (**SoftMax Q-Learning, SQL**), whereas the second uses the ϵ-greedy algorithm, selecting the more highly valued choice with probability $1-\epsilon$ (ϵ**-greedy Q-Learning,** ϵ**-QL**). The final opponent class was defined to play the same choice that the agent did on the trial $t-n$, called **n-back Mimicry (MC)**, where n was randomly drawn between 1 and 5.

B. Deep Reinforcement Learning Model and Training scheme

Our agent is an Advantage Actor-Critic [31]. A 4-dimensional input of the agent's binary reward and choice at the previous timestep are fed forward into an LSTM layer of size 128 at each step, allowing the agent to integrate trial information at long timescales [32]. The agent was trained with backpropagation through time [33] using the RMSprop optimizer [34]. We trained the network on the LC, MP 1, and Pattern opponent classes, 310 opponents in total. Importantly, the network never knows the identity of its opponent.

We introduced meta-learning by training our network on multiple tasks in a continual manner as in [35]. To do so, we interleaved opponents during training, randomly swapping them out every 150 steps of the game, which we termed a block. To examine the multi-task agent's generalization capacity, we trained "single-task" agents of the same architecture against individual opponent classes as an empirical baseline.

C. Opponent Representation Space

1) Linear Classifier Probe: We tested the agents against opponents from our three training classes on 50 independent blocks (22,500 total trials) and fit a logistic regression classifier to predict the opponent identity from the recurrent activity in these trials [36]. We then let the agent play for 450 trials against other opponents (the washout period) and tested it again on the three training classes. We trained two types of classifiers: within- and between-block classification, in which the test set was the last 50 trials of the training block and the blocks after the washout period, respectively.

2) Adversarial Perturbation in Classifier Space: We perturbed the recurrent neural activity within the classifier subspace from one class to another, and asked whether the agent's performance became suboptimal as a result. Importantly, all subspace axes were identified to be orthogonal to the policy output axis, so any effect on behavior must be indirect. To do so, we identified the orthonormal class projection matrix with bases $W \in \mathbb{R}^{o \times n}$ of dimensions o opponents by n recurrent neurons, current state $\mathbf{s}_t \in \mathbb{R}^n$ and the desired coordinates in the character subspace $\hat{\mathbf{c}}_t \in \mathbb{R}^o$. Then, we reset activity to the new coordinates within this subspace by

$$\mathbf{s}_{t+1} = \mathbf{s}_t + W^T(\hat{\mathbf{c}}_\mathbf{t} - W\mathbf{s}_\mathbf{t}) \quad (2)$$

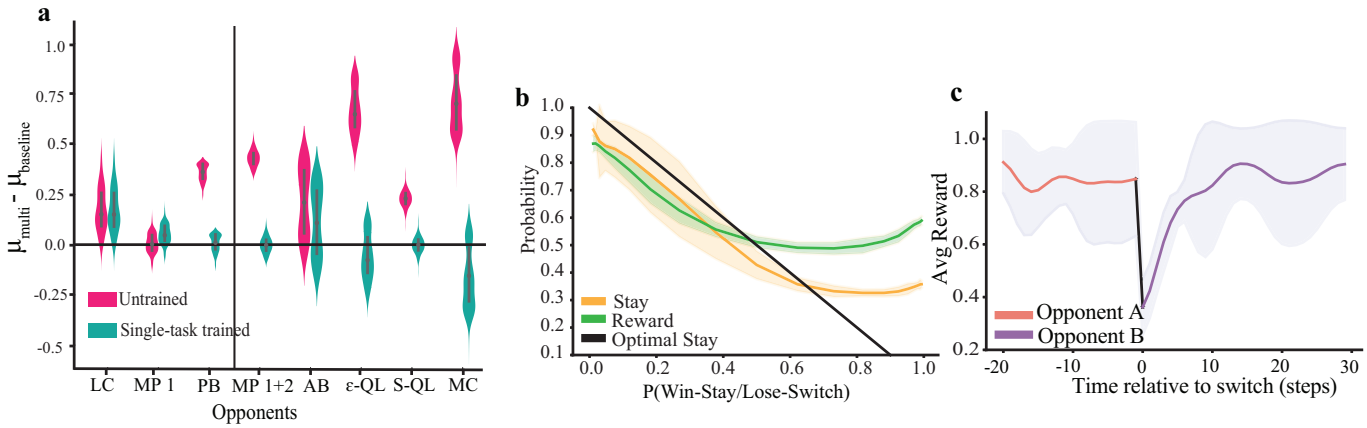

Fig. 1. **a.** Distribution of multi-task agent reward per trial for each opponent compared against two baseline agents. **b.** Choice behavior (Stay vs. Switch) and reward of the multi-task agent against an opponent that plays Win-Stay/Lose Switch with some specified probability (x-axis). Black line indicates the optimal strategy. **c.** Time-varying reward dynamics before and after a switch of the opponent. Shaded regions indicate standard error in all figures.

D. Representation similarity analysis (RSA)

$$\mathbf{R}(o_i) = \frac{1}{64}\Sigma_{j=0}^{64}\mathbb{E}[\mathbf{h}|o_i;\mathbf{s}_j] \tag{3}$$

Random binary input sequences three timesteps long $(\mathbf{s}_j, j = 1, 2, .., 64)$ were generated to probe how the network represents opponents (o_i). Equation 3 eliminates recent reward and action effects by marginalizing over the agent's recent past. First, the network played against a particular opponent o_i for 50 trials, which empirically allowed the representation to asymptote according to the linear classifier (Fig. 2a). We then fed in the random sequence and extracted the recurrent state $\mathbf{h}$ after the final input. By averaging the recurrent state over every sequence, we thereby identified a unique hidden state vector for each opponent, which we termed the recurrent representation or representation center. To perform RSA, we calculated the similarity between these recurrent representations using Pearson's R for all pairs of opponents.

III. Results

A. Agent generalizes, plays close to the theoretical optimal, and learns to quickly adapt to new opponents.

When tested on the agent's training opponents and novel opponents–named within-training-distribution (WD) and out-of-distribution (OOD) opponents respectively–the multi-task agent's performance was not significantly different from the single-task agent, and it exceeded that of the untrained agent for all opponents (Fig. 1a). This indicates that multi-task training provides the ability to generalize on all OOD opponent types we defined, although it remains unclear how the agent can perform against other OOD opponents such as mixture strategies.

Next, we assessed the performance of the multi-task agent against opponents with a known optimal strategy and found that it approaches this optimum. For example (Fig. 1b), we tested the agent against an LC opponent that modulates its probability of playing the Win-Stay/Lose-Switch (WSLS) strategy based on a single parameter in $\mathbf{b}_2$ at position $t-1$. To maximize reward, the agent needs to decrease (increase) its probability of staying on the same choice as the previous trial when the opponent's probability of Win-Stay/Lose-Switch increases (decreases). Our agent does so optimally within the training domain, $P(WSLS) \in [0.3, 0.7]$, and it generalizes well to lower probabilities.

Over a subset of opponents for which the agent's performance can be unequivocally assessed (max reward > 0.5), we found that the multi-task agent adapted its performance to the new opponent within only 20 time steps (Fig. 1c). Thus, after training against multiple classes of opponents, the network learned to quickly infer novel strategies and adapt its behavior accordingly.

B. Distinct within-distribution opponent representations exist in the recurrent space and are necessary for adaptive behavior.

To understand how the multi-task agent plays against novel OOD opponents, we analyzed how these opponents are separated in the recurrent activity of the network. With the linear classifier method, we hypothesized that if the agent had learned a stable representation of opponent character, then the classification accuracy should be maintained across time and be independent of past history (Fig 2a).

We could robustly decode the type of WD opponent with 95% accuracy after only 20 trials into a block, consistent with the time course of behavioral adaptation (Fig. 1c). This accuracy reached 100% after 50 trials and was maintained in the test period, even persisting throughout the washout period (Fig. 2a, Between-block Test). Perturbation of the opponent representation to another region of the classifier space significantly disrupted the agent's performance (Fig. 2b): reward dropped to random and classification accuracy fell to zero. This indicates the causal contribution of the opponent representation to the agent's performance.

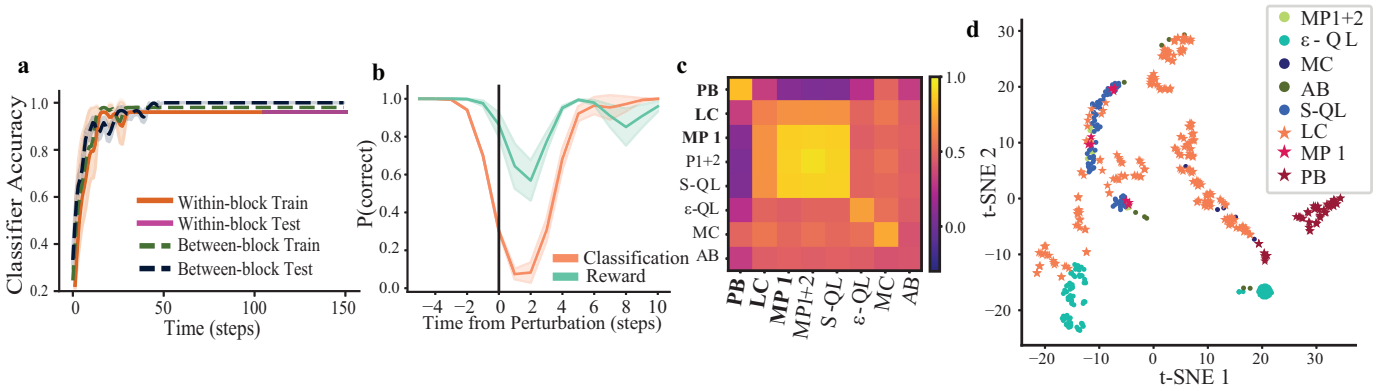

Fig. 2. **a.** Classification accuracy of logistic regression model on neural activity plotted over a single block. Stacked arrangement on RHS is for visualization, all are 100%. **b.** Accuracy of the between-block classifier on neural activity (orange) and probability of reward after the perturbation. The dip before time 0 is due to smoothing. **c.** RSA between recurrent centers, averaged over classes (WD in bold). **d.** Visualization of recurrent representations via t-SNE, colored by training distribution of opponents (WD, yellow/red hues), and the test distribution (OOD, blue/green hues). Each dot represents a single opponent.

C. Recurrent representations separate within- and out-of-distribution opponents and motivate multiple mechanisms of generalization.

Although on paper the out-of-distribution (OOD) classes are distinct from the within-training-distribution (WD), some OOD opponents may play similarly to WD opponents. Here, the agent could use a memorized training strategy. In order to assess how the agent generalizes, we sought to compare network representations of OOD strategies to WD ones. After identifying robust neural representations of WD opponents (Fig. 2a,b), we examined their relationship to the OOD opponent states using Representational Similarity Analysis (RSA, [37]).

Figures 2c and 2d depict the same representations of opponent character in different manners. By averaging the RSA over opponent classes (Fig 2c), we identified that only two OOD classes are very similar to the WD opponents (bold labels in Fig. 2c): MP 1+2 and S-QL. No opponents are very similar to the Pattern-Based opponent (PB). Because this visualization does not display relationships between individual opponent strategies, we reduced the 503 opponent representations with a 2-dimensional t-SNE and plotted the results in figure 2d. t-SNE nonlinearly reduces dimensionality by placing local clusters of data points near each other on the 2-D plane. Fig. 2d clearly shows multiple groups of out-of-distribution opponents that are distinctly separate from the training set (for example, the teal cluster of ϵ-QL opponents in the bottom right). With this in mind, we sought to identify how the neural activity distinguished OOD opponents from the WD distribution.

We hypothesized three possible mechanisms of similarity-based generalization: 1) template-matching in which the network adopts one of the learned strategies, 2) interpolation in which the network maps the novel opponent to one of the learned categories of opponents, and 3) extrapolation in which the network might use its learned representations beyond specific strategies or opponent categories. For example, the agent might template match by using a memorized strategy it had learned against an LC opponent with particular training parameters. It might interpolate by using the abstract strategy it had learned for LC opponents, but identifying parameters not observed in the training set. Finally, the agent might extrapolate by combining strategies, swapping between strategies, or doing something completely different.

We probed the mode of similarity-based generalization for each of the OOD-opponents by comparing them to WD recurrent representations with two measures: The minimum Euclidean distance to a WD recurrent representation, and the Euclidean distance to an LC opponent with parameters that best match the OOD opponent's strategy. We identified these parameters by fitting logistic regression models of the same structure as the LC strategies to the opponent's behavior during play against the agent (LC mapping, Fig. 3a). An OOD center close to a WD center suggests template-matching, whereas a shorter distance to the best-fitting LC strategy suggests interpolation. When both distances were relatively large, we labeled the strategy as extrapolation. We empirically set a distance of 1 to be the maximum value for both template-match and interpolation (boundaries in Fig. 3a, further results do not depend on precise value). We found that the majority of OOD opponents (129 of 193) fell in the template-match region (Fig. 3a, green region, some opponents not pictured). 42 opponents were classified as extrapolation.

D. Extrapolation opponents predominate regions of the manifold rarely occupied by within-distribution opponents.

Next, we investigated how the extrapolation opponents differ from the others. We asked whether the WD dynamics operated on a low-dimensional manifold, and if the OOD dynamics also fell on that manifold. We performed Principal Component Analysis (PCA) on recurrent states from all WD opponents and calculated the explained variance ratio of the OOD states on these PCs. We identified that 95% of the WD variance was explained by the first 10 PCs, with the first two components explaining 60% (Fig. 3b). These same 10 PCs summarize roughly 92% of the variance for all OOD

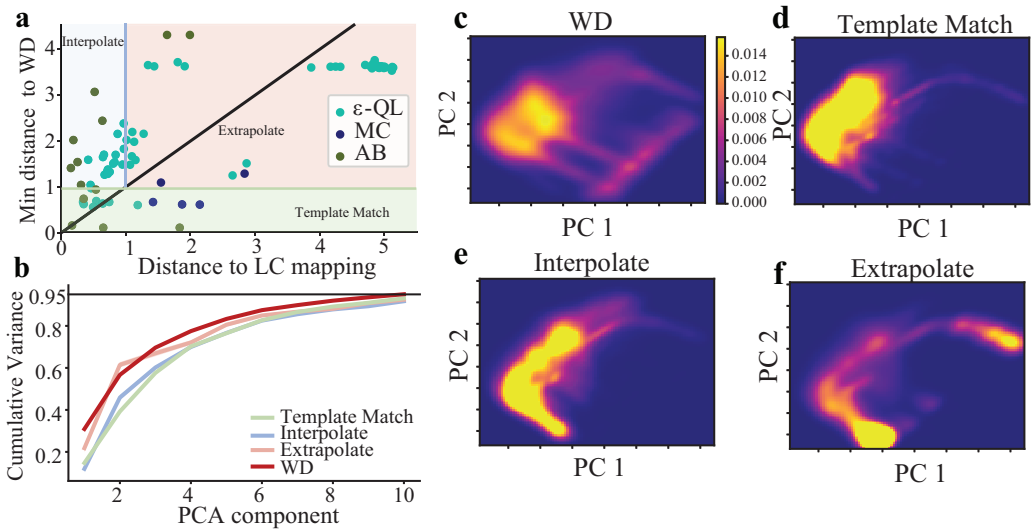

Fig. 3. **a.** Min WD and LC map distances for individual opponents with a sufficiently accurate LC map (classification accuracy $> 80\%$). Not pictured are those without a good LC fit. The green region indicates template matching, the blue indicates interpolation, and the orange indicates extrapolation. **b.** Explained Variance Ratio of the first n PC components of the WD distribution. **c-f.** Density heatmap of the first 2 PCs in **b**. x,y axis and color bar are shared.

classes, indicating that their dynamics also lie on the within-distribution opponent manifold.

Given that the first two PCs explain a majority of the variance and the cumulative variance curves for OOD states differ from the WD curve most drastically in these PCs (Fig. 3b), we asked how the dynamics along these two PCs differed in the OOD states. We extensively tested the agent against each opponent, saving recurrent states at all time points, and estimated the density of recurrent states in the first two PCs using a Gaussian Kernel Density Estimator. We plot the results as a heatmap in figures 3c-f. The majority of extrapolation states are localized in regions where there is low or zero density in the WD case, near the edge of the overall distribution. We quantified these states and found that roughly 50% of the cumulative density for the extrapolation distribution existed in regions of the WD distribution with density less than 0.001. These regions make up less than 5% of the WD distribution. This was not the case with the other generalization types, which followed the WD CDF much more closely and appear more similar to WD in Fig 3d,e. Finally, the distance between the extrapolation representations had an average Euclidean distance of 5.13 from the center of the entire distribution, whereas for template match, interpolate, and WD, this value was significantly lower, at 3.34, 3.78 and 3.6 respectively. Together, these results suggest that the extrapolation states lie on the edge of the learned manifold, in regions rarely occupied by WD states.

IV. Conclusion

By learning to interact in a competitive game with different types of opponents, we showed that our recurrent neural network had developed a dynamical representation of strategy. This representation has a global structure similar to [14]. The recurrent space additionally has a hierarchical structure similar to the network of [38], where broad regions in a low-dimensional manifold define different contexts and local dynamics govern online learning and decision-making.

By studying our agent's neural representations for unseen opponents, we identify possible mechanisms for how neural systems may solve the most important problem of ToM: adapting to novel social interactions. Given our results in Figure 3, we propose that learned opponent representations can be dynamically mixed in a neural network to give rise to new representations that recombine information from multiple training opponents. Two crucial features that provide this capacity are the continuous recurrent space, which allows states to move between representations, and recurrent dynamics that constrain activity to the learned manifold. Based on our final results, we tentatively suggest that on-manifold extrapolation may be one possible mode of generalization, but more work is required to verify its existence in our network, as PCA does not identify the nonlinear manifold itself.

References

[1] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526, 1978. doi: 10.1017/S0140525X00076512.

[2] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition with deep reinforcement learning. *PLOS ONE*, 12(4): 1–15, 04 2017. doi: 10.1371/journal.pone.0172395.

[3] Jaan Aru, Aqeel Labash, Oriol Corcoll, and Raul Vicente. Mind the gap: Challenges of deep learning approaches to Theory of Mind, 2022.

[4] Desmond C. Ong, Jamil Zaki, and Noah D. Goodman. Computational Models of Emotion Inference in Theory of Mind: A Review and Roadmap. *Topics in Cognitive Science*, 11(2):338–357, 2019. doi: https://doi.org/10.1111/tops.12371.

[5] C.F. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. The Roundtable Series in Behavioral Economics. Princeton University Press, 2011. ISBN 9781400840885.

[6] Tessa Rusch, Saurabh Steixner-Kumar, Prashant Doshi, Michael Spezio, and Jan Gläscher. Theory of mind and decision science: Towards a typology of tasks and computational models. *Neuropsychologia*, 146:107488, 2020. ISSN 0028-3932. doi: https://doi.org/10.1016/j.neuropsychologia.2020.107488.

[7] Ismael T. Freire, Xerxes D. Arsiwalla, Jordi-Ysard Puigbò, and Paul Verschure. Modeling Theory of Mind in Multi-Agent Games Using Adaptive Feedback Control, 2019.

[8] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. Agent Modeling as Auxiliary Task for Deep Reinforcement Learning. In *Proceedings of the Fifteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE'19. AAAI Press, 2019. ISBN 978-1-57735-819-0.

[9] Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z Leibo, Karl Tuyls, and Stephen Clark. Emergent Communication through Negotiation, 2018.

[10] Robert Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling Others using Oneself in Multi-Agent Reinforcement Learning. *CoRR*, abs/1802.09640, 2018.

[11] Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[12] Stefano V. Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, may 2018. doi: 10.1016/j.artint.2018.01.002.

[13] 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. doi: 10.1109/TSMCC.2007.913919.

[14] Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR, 2018. ISBN 2640-3498.

[15] Chris L. Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B. Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):0064, 2017. ISSN 2397-3374. doi: 10.1038/s41562-017-0064.

[16] Thuy Ngoc Nguyen and Cleotilde Gonzalez. Theory of Mind From Observation in Cognitive Models and Humans. *Topics in Cognitive Science*, n/a(n/a). doi: https://doi.org/10.1111/tops.12553.

[17] Christian Keysers and Valeria Gazzola. Hebbian learning and predictive mirror neurons for actions, sensations and emotions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1644):20130175, 2014. ISSN 0962-8436.

[18] Jorie Koster-Hale and Rebecca Saxe. Theory of mind: a neural prediction problem. *Neuron*, 79(5):836–848, 2013.

[19] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. ISBN 0262352702.

[20] Shinsuke Suzuki, Norihiro Harasawa, Kenichi Ueno, Justin L. Gardner, Noritaka Ichinohe, Masahiko Haruno, Kang Cheng, and Hiroyuki Nakahara. Learning to Simulate Others' Decisions. *Neuron*, 74(6):1125–1137, 2012. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2012.04.030.

[21] Bryan González and Luke J. Chang. *Computational Models of Mentalizing*, pages 299–315. Springer International Publishing", 2021. ISBN 978-3-030-51890-5. doi: 10.1007/978-3-030-51890-5_15.

[22] Caroline J. Charpentier and John P. O'Doherty. The application of computational models to social neuroscience: promises and pitfalls. *Social Neuroscience*, 13(6): 637–647, 2018. doi: 10.1080/17470919.2018.1518834. PMID: 30173633.

[23] Nick Feltovich. Belief-Based Learning Models. *Encyclopedia of the Sciences of Learning*, page 444–447, 2012. doi: 10.1007/978-1-4419-1428-6_111.

[24] Natalia Vélez and Hyowon Gweon. Learning from other minds: an optimistic critique of reinforcement learning models of social learning. *Current Opinion in Behavioral Sciences*, 38:110–115, 2021. ISSN 2352-1546. doi: https://doi.org/10.1016/j.cobeha.2021.01.006. Computational cognitive neuroscience.

[25] Daeyeol Lee, Hyojung Seo, and Min Whan Jung. Neural Basis of Reinforcement Learning and Decision Making. *Annual Review of Neuroscience*, 35(1):287–308, 2012. doi: 10.1146/annurev-neuro-062111-150512. PMID: 22462543.

[26] Dominic J. Barraclough, Michelle L. Conroy, and Daeyeol Lee. Prefrontal cortex and decision making in a mixed-strategy game. *Nature Neuroscience*, 7(4):404–410, 2004. ISSN 1546-1726. doi: 10.1038/nn1209.

[27] Tambet Matiisen, Aqeel Labash, Daniel Majoral, Jaan Aru, and Raul Vicente. Do deep reinforcement learning agents model intentions?, 2018.

[28] Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-Ended Learning Leads to Generally Capable Agents, 2021.

[29] Jane X. Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Demis

Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21(6):860–868, 2018. ISSN 1546-1726. doi: 10.1038/s41593-018-0147-8.

[30] Hyojung Seo, Xinying Cai, Christopher H. Donahue, and Daeyeol Lee. Neural correlates of strategic reasoning during competitive games. *Science*, 346(6207):340–343, 2014. doi: 10.1126/science.1256254.

[31] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1928–1937, New York, New York, USA, 2016.

[32] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

[33] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990. ISSN 1558-2256. doi: 10.1109/5.58337.

[34] Geoffrey Hinton, Nish Srivastava, and Kevin Swersky. *Neural Networks for Machine Learning Lecture 6a Overview of mini–batch gradient descent*. 2014. URL http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[35] Guangyu Robert Yang, Madhura R. Joglekar, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22 (2):297–306, 2019. ISSN 1546-1726. doi: 10.1038/s41593-018-0310-2.

[36] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2016.

[37] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2, 2008. doi: 10.3389/neuro.06.004.2008.

[38] Rylan Schaeffer, Mikail Khona, Leenoy Meshulam, Brain Laboratory International, and Ila Fiete. Reverse-engineering recurrent neural network solutions to a hierarchical inference task for mice. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4584–4596. Curran Associates, Inc., 2020.