Juiced and Ready to Predict Private Information in Deep Cooperative Reinforcement Learning

Eugene Lim
elimwj@comp.nus.edu.sg
National University of Singapore

Bing Cai Kok
kokbc@comp.nus.edu.sg
National University of Singapore

Songli Wang
e0466664@u.nus.edu
National University of Singapore

Joshua Lee
joshua_lks@u.nus.edu
National University of Singapore

Harold Soh
harold@comp.nus.edu.sg
National University of Singapore

ABSTRACT

In human-robot collaboration settings, each agent often has access to *private information* (PI) that is unavailable to others. Examples include task preferences, objectives, and beliefs. Here, we focus on the human-robot dyadic scenarios where the human has private information, but is unable to directly convey it to the robot. We present Q-Network with Private Information and Cooperation (Q-PICo), a method for training robots that can interactively assist humans with PI. In contrast to existing approaches, we explicitly model PI prediction, leading to a more interpretable network architecture. We also contribute *Juiced*, an environment inspired by the popular video game *Overcooked*, to test Q-PICo and other related methods for human-robot collaboration. Our initial experiments in *Juiced* show that the agents trained with Q-PICo can accurately predict PI and exhibit collaborative behavior.

CCS CONCEPTS

• **Computing methodologies** → **Multi-agent reinforcement learning**; *Cooperation and coordination*; Neural networks.

KEYWORDS

human-robot collaboration; cooperative multi-agent system; deep reinforcement learning

ACM Reference Format:
Eugene Lim, Bing Cai Kok, Songli Wang, Joshua Lee, and Harold Soh. 2020. *Juiced* and Ready to Predict Private Information in Deep Cooperative Reinforcement Learning. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (HRI ’20 Companion), March 23–26, 2020, Cambridge, United Kingdom.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3371382.3378308

1 INTRODUCTION

Many human-robot collaboration scenarios are characterized by the presence of *private information* (PI) that is not accessible by all agents. For example, consider a robot that is attempting to assist a human to serve juice at a juice bar. The human knows the juice preferences of each customer—the private information—but has no direct way of conveying this information to the robot. In this *limited communication* setting, the robot has to infer/predict which juice it should help prepare by observing the actions taken by the human; if the human is walking to a cabinet filled with apples, it can deduce that the customer is likely craving for apple juice.

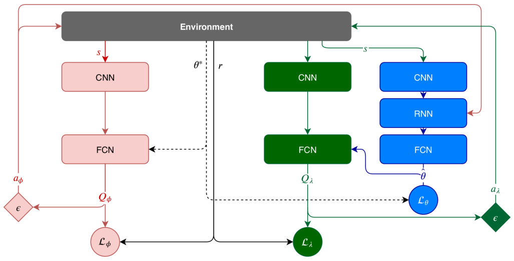


Figure 1: The Q-PICo architecture during training phase (best seen in color). The red Q-network depicts the agent ϕ (human surrogate). The green Q-network represents for robot λ and the blue private information network (PIN) predicts $\theta \in \Theta$ using ϕ's actions. We train the Deep Q-Networks (DQNs) using the losses $\mathcal{L}_\phi$ and $\mathcal{L}_\lambda$ and the PIN via $\mathcal{L}_\theta$.

In this extended abstract, we propose a reinforcement learning (RL) approach for training robots to assist in scenarios with private information and limited communication. Prior work has demonstrated the feasibility of jointly training cooperative agents using a pair of deep recurrent Q-networks [2, 3] (one network acts as a surrogate human). However, training such models can be difficult and the resulting models are often hard-to-interpret black-boxes. We circumvent this issue by separating the prediction of private information from the control module. This separation enables us to examine the robot’s belief of the private information during its interaction with the human, thus allowing for a more interpretable model and facilitating the training process.

We report on preliminary findings using an environment called *Juiced* (Fig. 2). Inspired by the video game *Overcooked*, *Juiced* places agents in a simulated juice bar where they are tasked to serve customers with different orders. Only one of the agents (the human) has access to the orders. We designed *Juiced* to be extensible; it can be configured to simulate scenarios of varying difficulties. Our initial experiments in a simple *Juiced* scenario show that (i) Q-PICo-trained agents demonstrate collaborative behavior, and that (ii) joint performance improves as the PI prediction improves.

HRI ’20 Companion, March 23–26, 2020, Cambridge, United Kingdom

ACM ISBN 978-1-4503-7057-8/20/03.
https://doi.org/10.1145/3371382.3378308

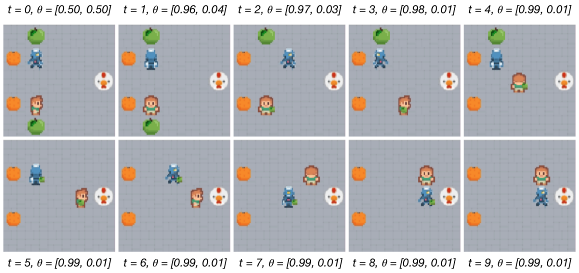


Figure 2: A simple example level in *Juiced* showing behavior of the agents trained using Q-PICo. The customer (chicken, right) wants apples. Time steps t and predicted private information θ ([apple, orange]) are shown for each frame. The robot agent (top, blue) waits for the human agent to pick up the apple *before* it delivers the correct fruit.

2 FORMULATION

We define a cooperative multi-agent Markov decision process (MDP) with private information as a tuple $\langle S, \Lambda, \Theta, \{A_\lambda\}_{\lambda\in\Lambda}, R, T, \gamma\rangle$, where S is a set of observations, Λ is a set of agents, Θ is a set of possible private information, A_λ is a set of actions for $\lambda \in \Lambda$, R is a reward function mapping from $S \times A \times \Theta$ to $\mathbb{R}$, T is a transition function mapping from $S \times A \times \Lambda \times S$ to $[0, 1]$, and $\gamma \in [0, 1]$ is a discount factor. To make our notation brief, we use A to represent $\prod_{\lambda\in\Lambda} A_\lambda$. Suppose that only one agent, $\phi \in \Lambda$, knows the true private information $\theta \in \Theta$. In our context, this agent ϕ is typically the human. The objective of the agents is to find policies $\{\pi_\lambda\}_{\lambda\in\Lambda}$ that maximize the expected sum of discounted reward $G_\pi = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$.

3 METHODOLOGY

Figure 1 illustrates the Q-PICo architecture during training phase. We interleave training of the DQNs and the Private Information Network (PIN), fixing one while we optimize the other for a fixed number of iterations. When training the DQN for agent i, we minimize the loss:

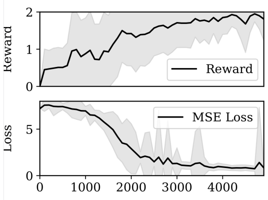
$$\mathcal{L}_i = \left(r^{(t)} + \gamma Q_i\left(s^{(t+1)}, a_i^{(t+1)}, \theta_i^{(t+1)}\right) - Q_i\left(s^{(t)}, a_i^{(t)}, \theta_i^{(t)}\right)\right)^2,$$

where $r^{(t)}$ is the reward obtained at time step t, γ is the discount factor, $\theta_\phi^{(t)} = \theta^*$ for all time steps t, and $\theta_\lambda^{(t)} = \hat{\theta}$ is the predicted θ at time step t. When training the PIN, we regress the predicted $\hat{\theta}$ against the θ^*; we assume that θ^* is available *during the training phase* (the PI is unavailable at test time) and minimize the L2-norm: $\mathcal{L}_\theta = \|\hat{\theta}^{(t)} - \theta^*\|^2$.

4 EVALUATION & DISCUSSION

To evaluate the performance of Q-PICo, we set up a proof-of-concept scenario where the maximum reward is attained only if the robot correctly predicts the private information and acts accordingly. Our experiment used the *Juiced* scenario shown in Fig. 2. At every step, the agents can take one of 6 actions: do nothing, interact with the object, move up, down, left, or right.

Each state is defined by 70 distinct entities (e.g., the agents, fruits, barriers) and their corresponding positions in the 5 x 5 grid. Consider a single cell in the grid; we define a binary vector of length 70 whose j-th element is 1 if the j-th unique entity is in that grid and 0 otherwise. We combine the vectors for all cells into a state tensor s of size $70 \times 5 \times 5$. At the beginning of each episode, we randomly reassign the customer's desired fruit to be either apples ($\theta^* = 0$) or oranges ($\theta^* = 1$). The agents receive a reward of 1 every time they serve the correct item. To prevent any single agent from completing the task by itself, we further restrict the episode to terminate in 10 steps.



	Reward
Q-PICo	1.9500 ± 0.0279
Ablated	1.2900 ± 0.0475

Figure 3: The reward obtained and the loss of the PIN during the PIN training phase; lower PIN loss is correlated with higher reward. The ablated network (without the PIN) obtains lower reward.

Fig. 1 illustrates the overall network architecture used in our experiment. Briefly, for the forward pass of each DQN, the observation tensor s was fed it into a CNN, flattened, and appended with θ. The resultant vector was fed into a fully connected network to obtain a state-action value for each of the 6 actions. The structure for the PIN is similar to the DQN and differs only in the addition a gated recurrent unit (GRU) layer [1]. In total, we trained the DQNs for 250,000 episodes and the PIN for 5,000 episodes using Adam.

We make two key observations from multiple sampled trajectories (Fig. 2 shows a sample trajectory). First, the agents often achieved the maximum possible reward of 2, suggesting that Q-PICo framework is effective at training pairs of agents to exhibit collaborative behavior. Second, agent λ always takes the corresponding fruit only after agent ϕ, i.e., the robot waits for informative actions to adapt its subsequent behavior. We also observed that the performance of the agents improved as the MSE loss incurred by the PIN decreased (Fig. 3). The reward obtained by an ablated model (λ did not explicitly predict PI) is significantly lower, suggesting the importance of the PIN. It may be possible to achieve implicit PI prediction with a more complex network, but the model would be less interpretable.

Moving forward, one limitation of Q-PICo is that the human surrogate model may not be representative of actual human behavior. We are in the process collecting and integrating human data into the training process to address this issue, and plan to investigate more complex scenarios in future work. More broadly, we believe Q-PICo and *Juiced* together represent an important step towards interpretable and reproducible frameworks for training deep policies that assist in realistic scenarios.

ACKNOWLEDGMENTS

This work is supported by the Natural Research Foundation Singapore under Grant No.: AISG-RP-2019-011.

REFERENCES

[1] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179

[2] Matthew Hausknecht and Peter Stone. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. arXiv:cs.LG/1507.06527

[3] Mark Woodward, Chelsea Finn, and Karol Hausman. 2019. Learning to Interactively Learn and Assist. arXiv:cs.AI/1906.10187