Decentralized Likelihood Implicit Quantile Network

Xueguang Lu [1] **Christopher Amato** [1]

Abstract

Recent successes of value-based multi-agent deep reinforcement learning employ optimism by limiting underestimation updates of value function estimator, through carefully controlled learning rate (Omidshafiei et al., 2017) or reduced update probability (Palmer et al., 2018). To achieve full cooperation when learning independently, agent must estimate the state values contingent on having optimal teammates; therefore, value overestimation is frequency injected to counteract negative effects caused by unobservable teammate sub-optimal policies and explorations. Aiming to solve this issue through automatic scheduling, this paper introduces a decentralized quantile estimator, which we found empirically to be more stable, sample efficient and more likely to converge to joint optimal policy.

1. Introduction

Recent development in multi-agent reinforcement learning (MARL) have borrowed increasing amount insights from single-agent RL developments, including deep convolutional networks for solving tasks high dimensional sensory observations (Mnih et al., 2015). In this paper, we bring insight from recent development in Deep Distributional RL to multi-agent settings; aim to be robust to sub-optimal local policies, we utilize estimated distributions to adjust learning rates in a more effective manner. More specifically, we apply Implicit Quantile Network (IQN) (Dabney et al., 2018) to multi-agent settings, along with extensions to estimate update magnitude which is used to control optimism injected, we call the extended architecture Decentralized Responsible IQN.

Our work focuses on fully cooperative Independent Learners which both learn and execute in a distributed manner. This complete decentralization setting poses a harder problem than settings having information, typically actions, shared across agents (Claus & Boutilier, 1998). However, although task dependent, with high probability, the agents will not converge to joint optimal policy but to independent optimal policies under the effect of environment non-stationary caused by other agents' independent optimal policies (Fulda & Ventura, 2007). In other words, as agents are expected to perform cooperative tasks, agent must be robust to often occurring non-effective explorations. Recent attempts on mitigating the issue in high dimensional observation settings have shown success with hysteretic Q-Learning (Matignon et al., 2007) and leniency (Panait et al., 2006). Both approaches limit negative value updates, either proportionally or probabilistically, aiming to partly ignore the effect of non-effective explorations. Empirically, leniency show higher stability compared to hysteretic methods largely due to its adaptive temperature-enabled leniency at different stages of estimation maturity (Palmer et al., 2018).

In this work, we show how deep distributional reinforcement learning method can be extended to multi-agent settings, and how it can be further used to estimate time difference likelihood. Our approach, based on IQN and Hysteretic Q-Learning ideology, yet adapts to different training stage in state-action space, shows more stable convergence towards optimal joint policy without explicit provision of hyper-parameters for task-specific scheduling.

2. Background

2.1. Markov Decision Process and Deep Q-Network

To be relaxed later, we first formulate prefect information sequential decision making problem as a Markov Decision Process $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ a finite action space, $\mathcal{T}(s, a, s')$ the probability of transitioning from state $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ by taking action $a \in \mathcal{A}$, and $\mathcal{R}(s, a, s')$ is the immediate reward for such transition. The Markovian assumption is implied by the transition probability. Under thus formulation, the problem is to find a optimal policy $\pi^\star : \mathcal{S} \to \mathcal{A}$ which maximize sum of rewards.

Q-Learning (Watkins & Dayan, 1992) is a popular model-free method of learning optimal policy, which iterate on a set of Q values or approximators to approach optimal Q values or estimates, where $Q(s, a)$ is the expected maximum

[1]College of Computer and Information Science, Northeastern University, Boston, MA. Correspondence to: Xueguang Lu <lu.xue@husky.neu.edu>, Christopher Amato <c.amato@northeastern.edu>.

Preliminary work. Preprint.

sum of rewards achievable in the future given in state s and taking action a.

$$Q^{\star}(s,a) = \max_{\pi} \mathbb{E}(r_t + \gamma r_{t+1} + ... | s_t = s, a_t = a, \pi) \quad (1)$$

When used with approximation, tabular Q-Learning methods often guarantees to convergence on discrete set of states and actions. The update of tabular approach is shown below.

$$Q(s_t,a_t) \leftarrow (1-\alpha)Q(s_t,a_t) + \alpha(r_t + \gamma \max_a Q(s_{t+1},a)) \quad (2)$$

Where $\alpha \in (0,1]$ is learning rate at which the Q value updates, and $\gamma \in (0,1]$ is the discount factor applied to future Q estimations.

Deep Q-Network (Mnih et al., 2015) develops on a common practice where a function approximator is used for estimating Q values by parameterize Q function $Q^{\theta}(s,a)$ with parameters θ using deep convolutional neural network. DQN uses experience replay (Lin, 1993) where each transition (s_t, a_t, r_t, s_{t+1}) is stored into a large fix-sized experience buffer $D_t = \{(s_1,a_1,r_1,s_2),...,(s_t,a_t,r_t,s_{t+1})\}$ from which all training batches for the network are uniformly sampled to balance network's tendency to bias towards more recent samples. The update of the network follows the following loss function:

$$L_i(\theta_i) = \mathop{\mathbb{E}}_{s,a,r,s' \sim U(D)} [(r + \gamma \max_{a'} Q^{\theta_i^-}(s',a') - Q^{\theta_i}(s,a))^2] \quad (3)$$

where θ_i^- is the parameters for target network at iteration i, a target network is an identical network whose parameters are never directly updated but copied from the main network every C steps in order to maintain value stability.

2.2. Partial Observability

Inspired by real-world tasks, single-agent tasks without access to perfect information is often studied and formalized as Partial Observable MDP (POMDP) problems. POMDP extends MDP by granting agent with observations instead of states: $\langle \mathcal{S}, \mathcal{A}, \mathcal{Z}, \mathcal{T}, \mathcal{O}, \mathcal{R} \rangle$; where $\mathcal{S}$ is still a finite state space, $\mathcal{A}$ is finite action space, $\mathcal{Z}$ is finite set of observations, $\mathcal{T}(s,a,s')$ is the probability of transitioning from state $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ by taking action $a \in \mathcal{A}$, $\mathcal{O}(s,a,z)$ is the probability of observing $z \in \mathcal{Z}$ given state s and action a, $\mathcal{R}(s,a)$ is the immediate reward for such transition.

Extending DQN to accommodate partially observable tasks, (Hausknecht & Stone, 2015) proposed a model called Deep Recurrent Q-Network (DRQN), where a recurrent layer (LSTM) (Hochreiter & Schmidhuber, 1997) was used to replace the first post-convolutional fully-connected layer of DQN. Hausknecht and Stone argues that the recurrent layer is able to integrate an arbitrarily long history which can be used to infer the underlying state. Empirically, DRQN out-performed DQN on partial-observable tasks and is on par with DQN on fully-observable tasks. Formally, the network estimates $Q(o_t, a_t, h_{t-1}|\theta)$ (instead of $Q(s_t, a_t, |\theta)$), where θ is the parameters of the network, and $h_{t-1} = LSTM(h_{t-2}, o_t)$ is the hidden state generated from the LSTM layer from $t-1$ time step, which is zero initialized. Hausknecht and Stone nonetheless points out that DRQN confers no systematic benefit compared to DQN where past k observations is used to approximate s.

2.3. Decentralized POMDP (Dec-POMDP)

Dec-POMDP aims to generalize POMDP to multi-agent settings, as actions and observations becomes joint actions and observations: $\langle \mathcal{I}, \mathcal{S}, \boldsymbol{A}^{\mathcal{I}}, \mathcal{Z}, \mathcal{T}, \boldsymbol{O}^{\mathcal{I}}, \boldsymbol{R}^{\mathcal{I}} \rangle$ where $\mathcal{I}$ is a finite set of agents, $\boldsymbol{A}^i$ is action space for agent $i \in \mathcal{I}$, and $\boldsymbol{O}^i$ is observation space of agent i. At every time step, a joint action $\boldsymbol{a} = \langle a^1, ..., a^{|\mathcal{I}|} \rangle$ is taken, and agents respectively receive immediate rewards $\boldsymbol{R}^i(s, \boldsymbol{a})$.

Collaborative agents receive rewards in partial or fully unified fashion. In particular, fully cooperative agents receive joint rewards where $\boldsymbol{R}^i(x, \boldsymbol{a}) = \boldsymbol{R}^{i'}(x, \boldsymbol{a})$ for all $i, i' \in \mathcal{I}$, $s \in \mathcal{S}$ and $\boldsymbol{a} \in \boldsymbol{A}$. Our work focuses on fully cooperative settings.

Multi-agent Reinforcement Learning (MARL) optimizes expected reward signal received by each agent in the same environment. MARL is usually classified into two classes: Independent Learners (ILs) and Joint Action Learners (JALs) (Claus & Boutilier, 1998). ILs observe only local action a^i for agent i, while JRLs have access to joint action $\boldsymbol{a}$. Our work is in line with solving ILs, which is more difficult, but resemble numerous real-world challenges.

2.4. Challenges of ILs

Even with perfect observability, ILs are non-Markovian due to teammate actions, hence the *environment non-stationary problem* (Bowling & Veloso, 2002). Literature have discussed other prominent problems when trying to apply Markovian methods, such as Q-Learning, to ILs, including *shadowed equilibria* (Fulda & Ventura, 2007), *stochasiticity*, and *alter-exploration* (Matignon et al., 2012).

2.4.1. Shadowed Equilibria

Without communications, independent learners who are maximizing their expected return optimally are known to be susceptible to sub-optimal Nash equilibrium where the sub-optimal joint policy can only be improved by changing all agents' policies simultaneously. Methods developed to battle this issue typically put more focus on high rewarded episodes, with the hope that all agents will be able to pursue maximum reward possible, hence forgoing the objective of maximizing the average return.

2.4.2. STOCHASTICITY

Optimistic methods as mentioned above, although often robust to *shadowed equilibria*, gives up the attempt to precisely estimate transitional stochasticity. Therefore, these methods can mistake a high reward resulted from environment stochasticity as a successful cooperation (Wei & Luke, 2016). In environments where high reward exists at low probability, the agents will then fail at approaching joint optimal policy.

2.4.3. ALTER-EXPLORATION PROBLEM

In order to estimate mean state values under stochasticity, ILs have to consider exploration-exploitation trade-off. For learners with ϵ-greedy exploration strategy, the probability of at lease 1 out of n agent is exploring at arbitrary time step is $1-(1-\epsilon)^n$. The alter-exploration problem amplifies the issue of *shadowed equilibria* (Matignon et al., 2012).

3. Related Work

3.1. Implicit Quantile Network (IQN)

IQN (Dabney et al., 2018) is a single-agent Deep RL method which we will extend to multi-agent settings with partial observability. As a distributional RL method, quantile networks are interested in distribution over returns, denoted Z^π, where $\mathbb{E}(Z^\pi) = Q^\pi$, by estimating the inverse c.d.f. of Z^π, denoted F_π^{-1}. Implicit Quantile Network (IQN) estimates $F_\pi^{-1}(\tau, s, a)$ denoted F $F_{\pi,\tau}^{-1}(x, a)$ from samples drawn from some base distribution ranging from 0 to 1, e.g. $\tau \sim U([0,1])$, where τ represents the quantile value at which the network aim to estimate. The estimated expected return can be obtained by averaging over multiple quantile estimates:

$$Q_\beta(x,a) := \mathop{\mathbb{E}}_{\tau\sim U([0,1])}[F^{-1}_{\pi,\beta(\tau)}(x,a)] \tag{4}$$

where $\beta : [0,1] \to [0,1]$ distorts risk sensitivity, risk neutrality is achieved when $\beta = \mathbb{1}$.

To force interaction between quantile values and observation features extracted by convolutional layers (in our case , LSTM layer), τ is embedded to match the dimension of features $\phi(\tau)$ and the Hadamard (point-wise) product of thus two vectors is used as features for subsequent fully connected layers. The embedding method Dabney et al. proposed is given by:

$$\phi(\tau) = \text{ReLU}(\sum_{i=0}^{n-1} \cos(\pi i \tau) w_i + b) \tag{5}$$

The quantile regression loss (Koenker & Hallock, 2001) for estimating quantile at τ and error δ is defined using Huber loss $\mathcal{H}_\kappa$ with threshold κ

$$\rho_\tau(\delta) = (\tau - \mathbb{I}\{e \le 0\})\frac{\mathcal{H}_\kappa(\delta)}{\kappa} \tag{6}$$

which weighs overestimation by $1-\tau$ and underestimation by τ, $\kappa = 1$ is used for linear loss.

Given two sampled $\tau, \tau' \sim \beta(U([0,1]))$ and policy π_β, the sampled TD error for time step t follows distributional bellmen operator:

$$\delta_t^{\tau,\tau'} = r_t + \gamma F_{\tau'}^{-1}(x_{t+1}, \pi_\beta(x_{t+1})) - F_\tau^{-1}(x_t, a_t) \tag{7}$$

Thus, given $\tau_{1:N}$, $\tau_{1:N'}$, the loss is given by:

$$L = \frac{1}{N'}\sum_{i=1}^{N}\sum_{j=1}^{N'} \rho_{\tau_i}(e^{\tau_i,\tau'_j}) \tag{8}$$

Notice that the loss is a summation of pair-wise quantile losses, and we can sample arbitrary number of τ_i and τ'_j for calculating the loss. Furthermore, Dabney et al. shown that N and N' have no significant effect on long term convergence, and $N = N' = 8$ appears to be sufficient in Atari games.

Distributional RL have shown, in single agent setting, to be robust to hyperparameter variation and to have superior sample complexity and performance (Barth-Maron et al., 2018). IQN, in particular, achieves state-of-the-art performance on Atari games compared to preceding deep distributional RL methods.

3.2. Hysteretic Deep Q-Network (HDQN)

HDQN (Omidshafiei et al., 2017) represents Hysteretic Q-Learning (HQL) (Matignon et al., 2007) in a deep network fashion with added recurrent layer and modified experience buffers. Due to non-stationary nature of multi-agent tasks caused by unobservable teammate policy, which is often sub-optimal due to exploration strategy, vanilla Q-Learning therefore would be forced to learn to estimate the non-stationary transitions. HQL uses two learning rates $0 < \beta < \alpha \le 1$, α is used for overestimation TD updates, and β is used otherwise. Overestimation is defined as $\delta_t > 0$; where TD error $\delta_t = r + \gamma \max_{a'} Q(s_{t+1}, a) - Q(s_t, a_t)$.

3.2.1. CONCURRENT EXPERIENCE REPLAY TRAJECTORIES (CERTS)

CERTs is used to stabilize training for HDQN. Through synchronizing random number generator across distributed experience buffers, CERTs achieves concurrent sampling of experience batches during training. The motivation is to stabilizing coordination despite shadowed equilibria, while earlier attempts disables experience reply due to non-concurrent evolving across agents' policies (Foerster et al., 2016).

CERTs also adds $\eta - 1$ padding transitions (zeros) to start and end of each episodes for training trace length η. The padding approach, although does not contribute to learning, ensures all valid transitions are uniformly sampled; whereas sampling traces from only valid transitions makes near-start and near-end transitions less likely to be sampled.

4. Approach

Learning a distribution over returns provides a richer representation of transitional stochasticity *and* exploratory teammates. Therefore, we extend IQN to multi-agent domains. We also evaluate the model on partial observable settings equipped with a LSTM layer and CERTs.

Additionally, we propose a granular approach for controlling the learning rate in state-specific fashion without an explicit encoder. The measure, which we call Time Difference Likelihood (TDL), measures the overlapping probability mass of two probability density functions supported by quantiles. The motivation is that, for overall similar distribution, even with drastic difference in specific quantile locations, the learning rate remains relatively high to capture local differences. Also, we show from empirical evaluations, TDL act as a state-specific scheduler which cause learning rate to increase over time for states which have received enough training, result in less action shadowing hence converging more robustly towards joint optimal policy.

4.1. Time Difference Likelihood (TDL)

We first introduce a discretized approximation method for estimating the likelihood of samples $F^{-1}_{\tau'_{1:M'}}(x_t, a_t)$, denoted $t_{1:M'}$, given a distribution constituted by set of samples $F^{-1}_{\tau_{1:M}}(x_t, a_t)$, denoted $d_{1:M}$. Let the distribution that $d_{1:M}$ approximates be $\mathcal{F}(X) = F_i(X)$ if $d_i \leq X \leq, d_{i+1}$ otherwise 0, where each F_i is a linearity that fits (τ_i, d_i) and (τ_{i+1}, d_{i+1}). We denote $\mathcal{P}(X) := P(X \,|\, d_{1:M})$ for notation simplicity. We observe that, for arbitrary a and b

$$
\begin{aligned}
&\mathcal{P}(a < X < b) \\
&= \int_a^b Z_{d_{1:M}}(X)dX \\
&= \int_{-\infty}^b Z_{d_{1:M}}(X)dX - \int_{-\infty}^a Z_{d_{1:M}}(X)dX \\
&= \mathcal{F}(b) - \mathcal{F}(a) \\
&= \sum_{i=1}^{M-1} \frac{|(a,b] \cap (d_i, d_{i+1}]|}{d_{i+1} - d_i}(\mathcal{F}(d_{i+1}) - \mathcal{F}(d_i)) \\
&= \sum_{i=1}^{M-1} \frac{|(a,b] \cap (d_i, d_{i+1}]|}{d_{i+1} - d_i}(\tau_{i+1} - \tau_i)
\end{aligned}
$$

Then, we can estimate target set likelihood as:

$$
l_{t_{1:M'}, d_{1:M}} = \mathop{\mathbb{E}}_{j \in 1:M'} \mathcal{P}\Big(\mathbb{E}(t_{j-1}, t_j) \leq t_j \leq \mathbb{E}(t_{j+1}, t_j)\Big) \tag{9}
$$

The likelihood measure differs from KL divergence and wasserstein distance

4.2. TD Likely Update

We introduce Responsible Hysteretic IQN by incorporating sample likelihood discussed above into hysteretic learning rate tuning. While traditional hysteretic uses learning rates $0 < \beta < \alpha \leq 1$, we use $0 < max(\beta, l_{t_{1:M'}, d_{1:M}}) \leq \alpha \leq 1$. More specifically, learning rate μ is given by:

$$
\mu_t = \begin{cases} max(\beta, l_{t_{1:M'}, d_{1:M}}), & \text{if } e_t^{\tau,\tau'} \leq 0 \\ \alpha, & \text{otherwise} \end{cases} \tag{10}
$$

Since $l_{t_{1:M'}, d_{1:M}} \in [0, 1]$, the amount of optimism added by hysteretic updates is reduced; update will be taken in magnitudes proportional to the estimated likelihood.

5. Experiments

5.1. Recurrent architectures evaluation on meeting-in-a-grid

We conduct experiments using HDQN architecture as basis with LSTM layer. The network starts with 2 fully connected layers of 32 and 64 neurons respectively, then a LSTM layer with 64 memory cells, a fully connected layer with 32 neurons which outputs value estimates for each action. We use $\beta = 0.4$, $\gamma = 0.95$ and Adam Optimizer [15] for training. For quantile estimators, we sample 16 τ and τ' to approximate return distribution for both training and acting online, and quantile embeddings were combined with the output of LSTM layer. Training is done in parallel, and results shown are batches of 20 randomly seeded runs. We use recurrence in meeting-in-a-grid domain to keep consistency with previous works [1]. The domain consists of one moving target and two agents in a grid world, agents get reward 1 for simultaneously standing on target location, 0 otherwise. Episode terminates after 40 transitions or upon successful meeting. Observation include probabilistically obscured locations of agent themselves and target, moving actions result in stochastic transitions.

We first evaluate RHIRQN performance against HDRQN and HIRQN. As shown in Figure 1, noticing differences in variance, that using quantile network alone or hysteretic-DQN alone not only fail to convergence to optimal policy on meeting-in-a-grid benchmark, but also susceptible to environment stochasticity, producing lesser effective joint policy over time. [TODO: MAKE FIGURE AND TALK ABOUT LIKELIHOOD TREND DURING TRAINING, SHOW LIKELIHOOD DOMINATES HYSTERETIC OVERTIME AND ROBUST TO DIFFERENT HYSTERETIC VALUES]

5.2. Multi-Agent Object Transportation Problems (CMOTPs)

We evaluate our agents on three variations of CMOTPs (Palmer et al., 2018), consistent with Palmer et al.'s work on LDQN. Also grid-world tasks, CMOTPs requires two agents carrying a box to a desired location where agents get a terminal reward; the box only moves when agents are by its side and moving in the same direction. Different variations include more obstacles in the gird-world and stochastic terminal rewards. CMOTPs have 16 by 16 sensory observations with added noise.

Our network architecture is mostly same as LDQN for comparability: two convolutional layers with 32 and 64 kernels, a fully connected layers with 1024 neurons which combines quantile embedding, followed by another fully connected layer with 1024 neurons which then maps onto value estimates for each action. Noticing from Figure 2 that although both methods converge to joint optimal policy while our method show an improved sample efficiency. The temperature and leniency control parameters were brought from Palmer et al.'s work where they found most suitable for the task; we hypothesize that the temperature is decaying less aggressive than it should be, which is likely due to temperature folding techniques or the hashing space of the autoencoder is larger than needed.

On the other hand, our method which utilize time difference likelihood to guide negative updates, started to learn aggressively early on. Initially we worry that the Q estimates would be not optimistic enough to perform coordinated actions, but the likelihood estimates were able to produce small values in under-explored state-action space while not hesitate to update negatively in explored spaces.

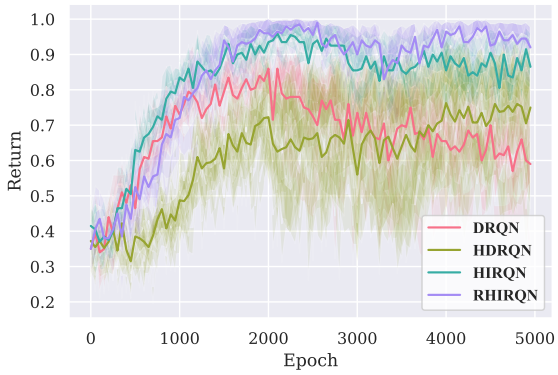

Figure 1. meeting-in-a-grid benchmark

6. Discussion

orthogonal with other improvements such as WDDQN, DUIDQN, state-dependent exploration strategy, etc

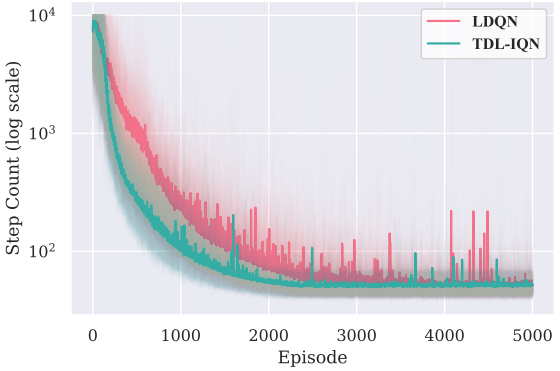

Figure 2. CMOTP benchmark, aggregated three CMOTP variances

References

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., Muldal, A., Heess, N., and Lillicrap, T. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

Bowling, M. and Veloso, M. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2): 215–250, 2002.

Claus, C. and Boutilier, C. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998:746–752, 1998.

Dabney, W., Ostrovski, G., Silver, D., and Munos, R. Im-

plicit quantile networks for distributional reinforcement learning. *arXiv preprint arXiv:1806.06923*, 2018.

Foerster, J. N., Assael, Y. M., de Freitas, N., and Whiteson, S. Learning to communicate to solve riddles with deep distributed recurrent q-networks. *arXiv preprint arXiv:1602.02672*, 2016.

Fulda, N. and Ventura, D. Predicting and preventing coordination problems in cooperative q-learning systems. In *IJCAI*, volume 2007, pp. 780–785, 2007.

Hausknecht, M. and Stone, P. Deep recurrent q-learning for partially observable mdps. *CoRR, abs/1507.06527*, 7(1), 2015.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Koenker, R. and Hallock, K. Quantile regression: An introduction. *Journal of Economic Perspectives*, 15(4):43–56, 2001.

Lin, L.-J. Reinforcement learning for robots using neural networks. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1993.

Matignon, L., Laurent, G., and Le Fort-Piat, N. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'07.*, pp. 64–69, 2007.

Matignon, L., Laurent, G. J., and Le Fort-Piat, N. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1):1–31, 2012.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529, 2015.

Omidshafiei, S., Pazis, J., Amato, C., How, J. P., and Vian, J. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. *arXiv preprint arXiv:1703.06182*, 2017.

Palmer, G., Tuyls, K., Bloembergen, D., and Savani, R. Lenient multi-agent deep reinforcement learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 443–451. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Panait, L., Sullivan, K., and Luke, S. Lenient learners in cooperative multiagent systems. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 801–803. ACM, 2006.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Wei, E. and Luke, S. Lenient learning in independent-learner stochastic cooperative games. *The Journal of Machine Learning Research*, 17(1):2914–2955, 2016.

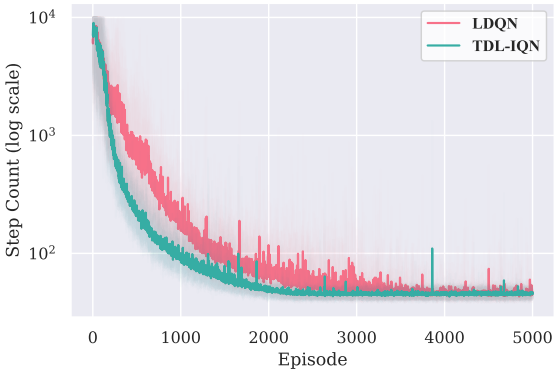

Figure 3. CMOTP Version 1

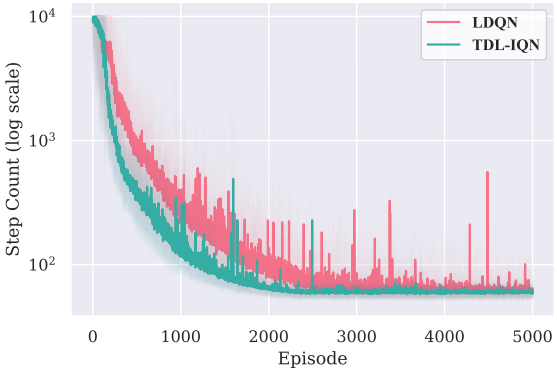

Figure 4. CMOTP Version 2 (Narrow Corredor)

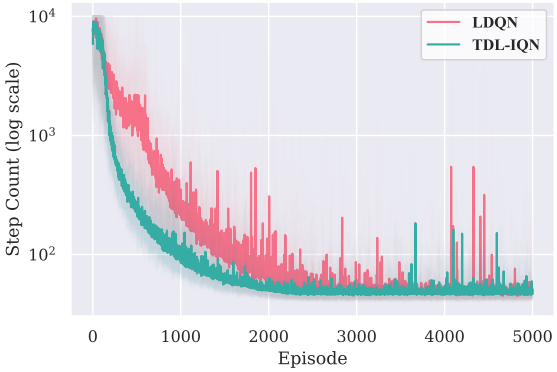

Figure 5. CMOTP Version 3 (Stochastic Reward)

7. Appendix