FedHQL: Federated Heterogeneous Q-Learning

Extended Abstract

Flint Xiaofeng Fan
National University of Singapore
Singapore
fxf@u.nus.edu

Yining Ma
National University of Singapore
Singapore
yiningma@u.nus.edu

Zhongxiang Dai
National University of Singapore
Singapore
daizhongxiang@comp.nus.edu.sg

Cheston Tan
I2R, A*STAR
Singapore
cheston-tan@i2r.a-star.edu.sg

Bryan Kian Hsiang Low
National University of Singapore
Singapore
lowkh@comp.nus.edu.sg

ABSTRACT

This study introduces the problem setting of Federated Reinforcement Learning with Heterogeneous And bLack-box agEnts (FedRL-HALE), in which multiple RL agents with varying policy parameterizations, training configurations, and exploration strategies work together to optimize their policies through the proposed Federated Heterogeneous Q-Learning (FedHQL) algorithm. Empirical results demonstrate the effectiveness of FedHQL in improving system performance and increasing the sample efficiency of individual agents with high confidence.

KEYWORDS

Federated learning; Q-learning; Federated reinforcement learning

ACM Reference Format:
Flint Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Cheston Tan, and Bryan Kian Hsiang Low. 2023. FedHQL: Federated Heterogeneous Q-Learning: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

Leveraging on the growing literature of *federated learning* (FL) [7, 9, 12, etc.], *federated reinforcement learning* (FedRL) [18] has emerged as a promising approach to improve the sample efficiency of RL agents in real-world environments. FedRL achieves collective intelligence [16] from distributed agents without requiring access to the raw trajectories of agent-environment interactions.

Despite their promising theoretical results [2, 3, 6, 8] and practical applications [5, 10, 11, 13, 15, 17, etc.], current FedRL algorithms have a limitation that they presume that all participants are *homogeneous*. This means that all agents must have the same policy parameterization (e.g., the architecture of the policy neural network, including the number of layers, the activation function, etc.) and the same training configurations for the policy (e.g., the learning rate). Such an assumption can be a significant limitation in real-world applications where agents are often *heterogeneous*, due to various disagreements such as computational budgets, assessments of the task's difficulty, etc.

This study introduces the problem of Federated Reinforcement Learning with Heterogeneous And bLack-box agEnts (FedRL-HALE), and proposes the **Fed**erated **H**eterogeneous **Q**-**L**earning (FedHQL) algorithm. FedHQL presents a federated version of Upper Confidence Bound (FedUCB) that addresses the Exploration-Exploitation (E&E) dilemma [1] in the multi-agent case, which we refer to as *Inter*-agents exploration problem. Empirical results demonstrate the effectiveness of FedHQL in improving system performance and increasing the sample efficiency of individual agents with high confidence. For an extended version of this paper, refer to [4].

2 PROBLEM FORMULATION

Consider the task of *federatively* solving a sequential decision-making problem represented by the Markov Decision Process $M \triangleq \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho, T\}$ [14]. Let the set $\mathcal{B} \triangleq \{Q_n(a|\ell_n(s; D_n, \omega_n))\}_{n=1}^N$ denote a group of N distributed heterogeneous and black-box agents. Each agent $\mathcal{B}_n$ independently operates in a separate copy of the underlying MDP M following its policy π_n, and generates its private experience data $D_n \triangleq \{(s, a, s', r)_i\}_{i=1}^{|D_n|}$. Each action valuator $Q_n(a|\ell_n(s; D_n, \omega_n))$, which we will use $Q_n(s, a)$ to denote, consists of a non-linear function $\ell_n(s; D_n, \omega_n)$ which predicts the value of action a given a state s. The non-linear function $\ell_n(s; D_n, \omega_n)$ is parameterized by a neural network with parameters ω_n and learned using the private experience data D_n.

Due to the *heterogeneity* among agents, different agents may choose different neural network architectures and employ different optimization methods to train their networks. To facilitate knowledge aggregation, we let the central server broadcast query state(s)

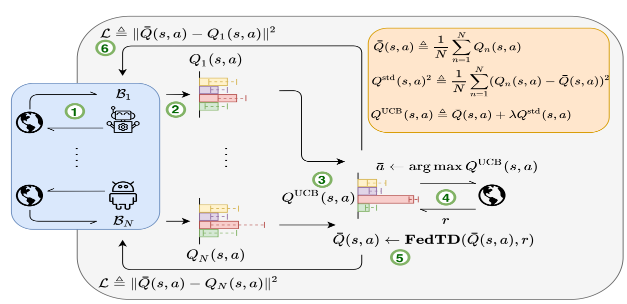

Figure 1: Graphical illustration of FedHQL.

to agents and query their estimations of the values of all actions at these candidate states (i.e., $Q_n(s, a), \forall a$). Then, the server can combine the knowledge of the entire group of agents by aggregating the received action value estimations $Q_n(s, a)$'s. Of note, the information regarding the non-linear function ℓ_n (including its architecture, weights ω_n, training methods, and other training details), as well as the local experience data D_n, is not revealed to any other party, including the central server.

The E&E dilemma [1] requires an agent to balance between exploiting its current knowledge and exploring to acquire new knowledge, which we will refer to as the *intra*-agent exploration problem. Similarly, in the setting of FedRL-HALE when the server selects its action by aggregating information from all agents, a natural trade-off arises: *Should the server select actions by exploiting the current information provided by all agents ? Or should the server select exploratory actions for which the agents have inconsistent value estimations?* This additional exploration-exploitation dilemma similarly highlights the requirement for a principled algorithm to balance the trade-off between exploiting the current knowledge of the entire group of agents and exploring to obtain new knowledge, which we will denote as ***inter*-agent exploration**.

3 FEDHQL

Here we discuss the key components of FedHQL illustrated in Fig. 1.

Federated Q-learning. At the core of FedHQL is the federated version of Q-learning with N heterogeneous and black-box agents. Each agent $\mathcal{B}_n$ *independently* interacts with its own copy of the MDP using its preferred *intra*-agent exploration strategy. Each agent $\mathcal{B}_n$ updates its current estimation of action values $Q_n(s, a)$ through Q-learning: as follows: $Q_n(s_t, a_t) \leftarrow Q_n(s_t, a_t) + \alpha_n[\mathcal{R}(s_{t+1}, a_{t+1}) + \gamma \max_a Q_n(s_{t+1}, a) - Q_n(s_t, a_t)]$.

Federated Upper Confidence Bound (FedUCB). FedUCB begins with the following corollary:

Corollary 3.1 (FedUCB). *Under the same assumptions and notations defined in Theorem 4.2 in [4], for any $c > 0$, with probability at least $1 - 3e^{-c}$, we have*

$$\mu_{s,a} \triangleq Q^*(s, a) \le Q^{UCB}(s, a) \triangleq \bar{Q}(s, a) + \sqrt{\frac{2c\mathbb{V}_{s,a}}{N}} + \frac{3bc}{N}.$$

Corollary 3.1 suggests that the optimal value of action a at state s, $Q^*(s, a)$, is upper-bounded by $Q^{\text{UCB}}(s, a)$ defined above with high confidence. Inspired by Corollary 3.1, we develop our practical FedUCB algorithm for the knowledge aggregation in FedRL-HALE, which firstly calculates (for any s, a):

$$\bar{Q}(s, a) = \frac{1}{N}\sum_{n=1}^{N} Q_n(s, a), \tag{1}$$

$$Q^{\text{std}}(s, a)^2 = \frac{1}{N}\sum_{n=1}^{N}[\bar{Q}(s, a) - Q_n(s, a)]^2, \tag{2}$$

$$Q^{\text{UCB}}(s, a) \simeq \underbrace{\bar{Q}(s, a)}_{\text{exploitation}} + \lambda \underbrace{Q^{\text{std}}(s, a)}_{\text{exploration}}, \tag{3}$$

where the degree of exploration is controlled by the parameter λ, which we will refer to as *inter*-agent exploration coefficient, such that a larger λ encourages the selection of more exploratory actions.

Table 1: Configurations of $N = 5$ agents for FedHQL

Agent No.	*Network*	*Learning rates*	*Intra-exploration coefficient*
1	64x64 (Tanh)	0.005	0.01
2	128x128 (ReLU)	0.01	0.1
3	32x32 (Tanh)	0.01	0.05
4	16x16 (ReLU)	0.02	0.01
5	8x8x8 (ReLU)	0.001	0.01

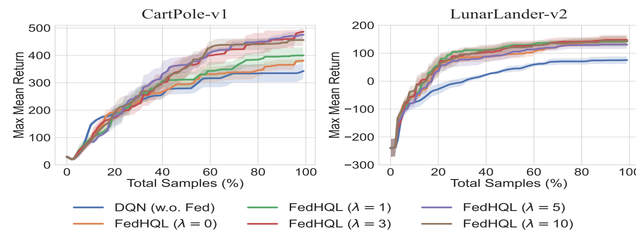

Figure 2: Learning curves of FedHQL against self-learning.

Federated Temporal Difference (FedTD). With the FedUCB derived above , the server is able to optimistically select an action that leads to high returns with high probability. Inspired by Fan et al. [3], we let the server operate in another separate copy of the underlying MDP and execute the selected action $\bar{a}$, hence generating a new sample $(s_t, \bar{a}, s_{t+1}, r_t)$. This new sample will then be used to perform a federated version of Temporal Difference (FedTD) learning: $\bar{Q}(s_t, \bar{a}_t) \leftarrow \bar{Q}(s_t, \bar{a}_t) + \alpha_s(r_t + \gamma \max_b \bar{Q}(s_{t+1}, b) - \bar{Q}(s_t, \bar{a}_t))$ where $\bar{Q}(s_t, a_t)$, the details of which can be found in 1.

Individual Improvement. After the FedTD target $\bar{Q}(s_t, \bar{a}_t)$ is updated, we let the server broadcast the updated $\bar{Q}(s_t, \bar{a}_t)$ back to all agents. An agent $\mathcal{B}_n$ will then update its own action value estimation Q_n using the following regression loss: $\mathcal{L}_n \triangleq \|\bar{Q}(s_t, \bar{a}_t) - Q_n(s_t, \bar{a}_t)\|^2$, which serves as a regularizer that periodically updates agent $\mathcal{B}_n$'s parameter ω_n by $\omega_n \leftarrow \omega_n - \tilde{\alpha}_n \nabla \mathcal{L}_n$ where $\tilde{\alpha}_n$ is a step-size hyper-parameter. This loss essentially helps the agent to improve its knowledge about action $\bar{a}_t$ at state s_t using the knowledge aggregated by FedUCB and updated by FedTD.

4 EMPIRICAL EVALUATION

We investigate the efficacy of FedHQL in improving the overall system performance with 5 heterogeneous agents depicted in Tab. 1. Given the fixed budget of each agent, we examine the average performance of agents versus the average consumption of the budget per agent. The results in both tasks are plotted in Fig. 2. The figures show that FedHQL with different choices of *inter*-agent exploration coefficients, FedHQL ($\lambda = 0, 1, 3, 5, 10$), significantly improves the average performance per agent over that of independent self-learning, DQN (w.o. Fed). For example, in the LunarLander task, an agent is expected to consume at least 40% of its budget (i.e., total 1.6m $= 4 \times 10^6 \times 0.4$ interactions) on average to receive positive returns while an agent in FedHQL ($\lambda = 1$) can achieve a performance close to 100 using only about 20% of its budget (i.e., total 0.8m $= 4 \times 10^6 \times 0.2$ interactions). More experimental results and analysis can be found in [4].

REFERENCES

[1] Laureiro-Martínez D Brusoni S Canessa. [n.d.]. N. Zollo M.(2015). Understanding the exploration–exploitation dilemma: An fMRI study of attention control and decision–making performance. *Strategic Management Journal* 36, 3 ([n. d.]), 319–338.

[2] Zhongxiang Dai, Yao Shu, Arun Verma, Flint Xiaofeng Fan, Bryan Kian Hsiang Low, and Patrick Jaillet. 2023. Federated neural bandits. In *International Conference on Learning Representations*. https://openreview.net/forum?id=38m4h8HcNRL

[3] Flint Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Wei Jing, Cheston Tan, and Bryan Kian Hsiang Low. 2021. Fault-Tolerant Federated Reinforcement Learning with Theoretical Guarantee. In *Advances in Neural Information Processing Systems*.

[4] Flint Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Cheston Tan, Bryan Kian Hsiang Low, and Roger Wattenhofer. 2023. *FedHQL: Federated Heterogeneous Q-Learning*. arXiv:2301.11135.

[5] Koki Fujita, Shugo Fujimura, Yuwei Sun, Hiroshi Esaki, and Hideya Ochiai. 2022. Federated Reinforcement Learning for the Building Facilities. In *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*. 1–6. https://doi.org/10.1109/COINS54846.2022.9854959

[6] Hao Jin, Yang Peng, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2022. Federated Reinforcement Learning with Environment Heterogeneity. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 18–37.

[7] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.

[8] Sajad Khodadadian, Pranay Sharma, Gauri Joshi, and Siva Theja Maguluri. 2022. Federated Reinforcement Learning: Linear Speedup Under Markovian Sampling. In *International Conference on Machine Learning*. PMLR, 10997–11057.

[9] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527* (2016).

[10] Xinle Liang, Yang Liu, Tianjian Chen, Ming Liu, and Qiang Yang. 2023. Federated transfer reinforcement learning for autonomous driving. In *Federated and Transfer Learning*. Springer, 357–371.

[11] Boyi Liu, Lujia Wang, and Ming Liu. 2019. Lifelong Federated Reinforcement Learning: A Learning Architecture for Navigation in Cloud Robotic Systems. *IEEE Robotics and Automation Letters* 4, 4 (2019), 4555–4562. https://doi.org/10.1109/LRA.2019.2931179

[12] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[13] Chetan Nadiger, Anil Kumar, and Sherine Abdelhak. 2019. Federated Reinforcement Learning for Fast Personalization. In *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. 123–127. https://doi.org/10.1109/AIKE.2019.00031

[14] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[15] Xiaofei Wang, Chenyang Wang, Xiuhua Li, Victor CM Leung, and Tarik Taleb. 2020. Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching. *IEEE Internet of Things Journal* 7, 10 (2020), 9441–9455.

[16] Ali Yahya, Adrian Li, Mrinal Kalakrishnan, Yevgen Chebotar, and Sergey Levine. 2017. Collective robot reinforcement learning with distributed asynchronous guided policy search. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 79–86.

[17] Shuai Yu, Xu Chen, Zhi Zhou, Xiaowen Gong, and Di Wu. 2020. When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multitimescale Resource Management for Multiaccess Edge Computing in 5G Ultradense Network. *IEEE Internet of Things Journal* 8, 4 (2020), 2238–2251.

[18] Hankz Hankui Zhuo, Wenfeng Feng, Yufeng Lin, Qian Xu, and Qiang Yang. 2019. *Federated deep reinforcement learning*. arXiv:1901.08277.