http://doi.org/10.35784/iapgos.7545 *received: 07.04.2025 | revised: 18.11.2025 | accepted: 16.12.2025 | available online: 30.03.2026*

KNOWLEDGE SHARING IN INDEPENDENT DEEP Q-NETWORK

Viacheslav Bochok, Nataliia Fedorova
Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine

Abstract. *This paper investigates knowledge sharing mechanisms in weakly coupled multi-agent reinforcement learning systems based on Independent Deep Q-Networks (IDQN). Although parallel agents can accelerate data collection, their learning processes typically remain isolated, resulting in suboptimal use of collective experience. To address this limitation, the study proposes two complementary methods: (1) a teacher-selection mechanism that identifies the most efficient agent based on episodic performance, and (2) a dynamic control mechanism that adjusts the intensity of knowledge transfer according to the performance gap between teacher and student. The experiments were conducted in the OpenAI Gym CartPole-v1 and LunarLander-v3 environments using three independent agents, to validate the effectiveness across tasks with different reward structures, dynamics, and difficulty levels. All agents were trained with Batch TD(0) at the end of each episode, using a replay. Knowledge transfer was implemented through policy distillation on pseudo-labeled transitions sampled from the teacher's experience buffer. The number of distillation epochs was dynamically determined using a nonlinear scaling function bounded by predefined minimum and maximum values. Results demonstrate that the proposed mechanisms consistently accelerate learning and improve stability compared to baseline DQN configurations without knowledge sharing. Systems employing teacher selection outperform random teacher choice and all-to-all sharing. Dynamic intensity adjustment proves more effective than constant-intensity distillation. Normalized AUC analysis further confirms statistically significant improvements in both maximum and average episodic returns, indicating faster convergence of the best agent as well as more uniform progress across all agents. The findings show that knowledge sharing with informed teacher selection and adaptive transfer strength provides a robust and scalable approach for improving the efficiency of independent agents in stationary environments. These mechanisms are compatible with common DQN extensions and can serve as a foundation for future research on adaptive multi-agent knowledge exchange strategies.*

Keywords: Deep Q-Network (DQN), knowledge sharing, multi-agent systems, policy distillation, reinforcement learning

WYMIANA WIEDZY W NIEZALEŻNYCH GŁĘBOKICH SIECIACH Q

Streszczenie. *W niniejszym artykule przeanalizowano mechanizmy dzielenia się wiedzą w słabo sprzężonych wieloagentowych systemach uczenia się przez wzmocnienie opartych na niezależnych głębokich sieciach Q (IDQN). Chociaż równoległe działanie agentów może przyspieszyć gromadzenie danych, ich procesy uczenia się zazwyczaj pozostają odizolowane, co skutkuje nieoptymalnym wykorzystaniem zbiorowego doświadczenia. Aby zaradzić temu ograniczeniu, w badaniu zaproponowano dwie uzupełniające się metody: (1) mechanizm wyboru nauczyciela, który identyfikuje najbardziej wydajnego agenta na podstawie wyników epizodycznych, oraz (2) mechanizm kontroli dynamicznej, który dostosowuje intensywność transferu wiedzy w zależności od różnicy w wynikach między nauczycielem a uczniem. Eksperymenty przeprowadzono w środowiskach OpenAI Gym CartPole-v1 i LunarLander-v3 z wykorzystaniem trzech niezależnych agentów, aby zweryfikować skuteczność w zadaniach o różnych strukturach nagród, dynamice i poziomach trudności. Wszyscy agenci byli szkoleni metodą Batch TD(0) na końcu każdego epizodu, z wykorzystaniem powtórki. Transfer wiedzy został zaimplementowany poprzez destylację polityki na podstawie pseudo-oznaczonych przejść pobranych z bufora doświadczeń nauczyciela. Liczba epok destylacji była ustalana dynamicznie przy użyciu nieliniowej funkcji skalującej, ograniczonej z góry zdefiniowanymi wartościami minimalnymi i maksymalnymi. Wyniki pokazują, że proponowane mechanizmy konsekwentnie przyspieszają uczenie się i poprawiają stabilność w porównaniu z podstawowymi konfiguracjami DQN bez dzielenia się wiedzą. Systemy wykorzystujące selekcję nauczyciela osiągają lepsze wyniki niż losowy wybór nauczyciela i dzielenie się wiedzą typu „wszyscy ze wszystkimi". Dynamiczna regulacja intensywności okazuje się skuteczniejsza niż destylacja o stałej intensywności. Analiza znormalizowanego AUC dodatkowo potwierdza statystycznie istotną poprawę zarówno maksymalnych, jak i średnich zwrotów epizodycznych, wskazując na szybszą konwergencję najlepszego agenta, a także bardziej jednolity postęp wśród wszystkich agentów. Wyniki pokazują, że dzielenie się wiedzą z świadomym wyborem nauczyciela i adaptacyjną siłą transferu zapewnia solidne i skalowalne podejście do poprawy wydajności niezależnych agentów w środowiskach stacjonarnych. Mechanizmy te są kompatybilne z popularnymi rozszerzeniami DQN i mogą służyć jako podstawa dla przyszłych badań nad adaptacyjnymi strategiami wymiany wiedzy między wieloma agentami.*

Słowa kluczowe: głęboka sieć Q (DQN), dzielenie się wiedzą, systemy wieloagentowe, destylacja polityk, uczenie się przez wzmocnienie

Introduction

Reinforcement Learning (RL) is applied in many modern domains, as agents are able to autonomously acquire skills through interaction with the environment and the reception of rewards. Among the most notable real-world achievements of RL are surpassing human-level performance in complex games, particularly due to DeepMind algorithms that achieved super-human results in Go and StarCraft II [10, 15]. RL methods are also effectively used to solve practical tasks such as traffic flow optimization, sensor network control, and robotic swarm coordination [11]. These accomplishments highlight the importance of RL in modern artificial intelligence and motivate further research in this field.

Despite its strengths, agent learning methods often require large amounts of data and training time, and they may suffer from instability [12]. An agent frequently needs to play through hundreds of thousands or even millions of episodes to approach an optimal policy, which becomes a bottleneck for real-world applications [1]. Slow convergence is closely related to inefficient use of collected experience: although each transition may be reused multiple times during learning, a considerable portion of interactions still remains underutilized.

Numerous approaches have been proposed to address these challenges, primarily focusing on advanced algorithms for training on an agent's own experience. Recent studies have shown growing interest in mechanisms for knowledge sharing among agents within a system. The core idea behind such methods is to enable one agent's knowledge to be used – or reused – by others to accelerate or stabilize learning. Several relevant research directions can be observed in the literature.

For example, the authors of the KnowSR framework [4] proposed sharing knowledge in the form of advice among homogeneous agents in multi-agent reinforcement learning as a complementary enhancement to the learning process, improving training speed. By leveraging the concept of knowledge distillation, KnowSR enables agents not only to learn from their own experience but also to integrate the experience of others, reducing learning time and increasing overall effectiveness. In another study [16], researchers proposed using knowledge sharing to accelerate environment exploration by cooperative agents. Their work utilized the MADDPG algorithm, designed for multi-agent systems and based on centralized critics with decentralized actors (centralized training with decentralized execution). These studies confirm the potential of knowledge sharing as a learning enhancement, but they also reveal several underexplored aspects or limitations that restrict broader applicability:

- Lack of mechanisms for assessing the quality of shared knowledge;
- No established method for controlling the intensity of knowledge transfer;

- Focus on specific architectures (Actor–Critic) and cooperative or strongly coupled environments.

For instance, KnowSR suggests an "all-to-all" learning approach, which may risk spreading knowledge from an agent performing worse than others. The intensity of sharing is static and depends only on the exchange frequency set by the designer (e.g., 1 to 9). Other studies focus on experience sharing rather than knowledge transfer [3], or on reusing an already trained agent to teach others [7], or applying it to different tasks [9].

1. Problem statement

The reviewed studies, on the one hand, highlight the need to improve the efficiency of multi-agent systems and confirm the potential of knowledge reuse and knowledge transfer. On the other hand, research on knowledge exchange in weakly coupled decentralized systems remains limited. The risks of negative knowledge transfer, as well as insufficient or excessive transfer strength, are also poorly explored.

Based on the hypothesis that learning from a more efficient agent may yield better results than learning from all agents collectively, this work proposes developing and investigating a teacher-selection method for knowledge transfer. This requires designing a mechanism for evaluating agent performance, as well as a ranking and filtering procedure for selecting suitable teachers.

Similarly, based on the hypothesis that the intensity of knowledge transfer should increase proportionally to the difference in agent performance, the study proposes a method for dynamically controlling knowledge-transfer intensity. This requires both a numerical measure of performance difference and a mechanism that converts this difference into a transfer strength value.

The study must also account for the dynamic nature of the system – that is, the fact that agents update their policies over time. Therefore, performance evaluation refers only to a specific version of an agent at a given moment and must be recomputed after each policy update. Another desirable property is decentralized implementation, without requiring a central entity such as a centralized critic. Furthermore, the study assumes weakly coupled (or stationary) environments, meaning that the actions of one agent do not influence others (or such influence can be neglected). A practical example is trading agents, each with their own budget; in most cases, an individual transaction is too small to significantly affect the global market.

These conditions make the DQN algorithm [6, 14] suitable for the study, as it is relatively easy to implement, does not require substantial computational resources, and operates effectively in weakly coupled environments. Key advantages of DQN include its ability to handle large state spaces and its reuse of past experience via experience replay. Today, DQN and its extensions are applied not only in gaming environments but also in robotics, resource-management systems, and other domains where autonomous agents need to make sequential decisions. DQN can be used in multi-agent systems under the name IDQN (Individual DQN), yet it has not been utilized in most recent knowledge-sharing studies.

Therefore, this work focuses on examining the impact of a knowledge-sharing mechanism in a weakly coupled system of IDQN agents, incorporating teacher-selection and dynamic transfer-intensity control. The proposed methods aim to address the issues of negative, insufficient, or excessive transfer and to improve overall system efficiency.

2. Teacher selection and knowledge-transfer intensity control methods

The essence of the teacher-selection method is that an agent selects another agent (or several agents) from whom it will learn using the pseudo-labeled dataset generated by that teacher. The task can be represented as two sequential operations: ranking agents by their effectiveness and filtering (selecting) only those that perform better, with an additional constraint on how many teachers may be chosen. A decentralized implementation requires that each agent must be able to perform these operations independently (or with the help of a designated agent or an environment-level function). In practice, each agent receives information from all (or most) other agents in the form of: *{performance score, pseudo-labeled dataset}*, after which it selects a teacher and performs learning.

Collected reward over a given period is proposed as the performance measure. Depending on the environment, this may correspond to an episode, a trajectory, or an n-step interval. A key requirement is that the teacher's model must remain unchanged during the evaluation period and until the pseudo-labeled dataset is generated. This constrains the use of online learning, since in many RL environments a single-step reward does not accurately reflect agent effectiveness. Rewards may also be delayed (as in CartPole, where a single step may not reflect performance and the strategy can only be evaluated over a long interval or upon reaching a terminal state). It is also logical that optimal teacher selection requires information from all agents in the system.

It is important to note that the number of selected teachers also matters. If too many are selected and their experience strongly overlaps, this may lead to overfitting on certain transitions, forgetting the student's own experience, or excessive averaging. Training at the end of an episode has both advantages and disadvantages:

- Training occurs less frequently;
- Some environments have extremely long or effectively infinite episodes, requiring explicit termination;
- Intensive training on a large replay buffer becomes possible, improving stability.

The intensity-control mechanism likewise requires performance evaluation based on collected reward. Here, the strength of the transfer is defined as the number of policy-distillation epochs. All calculations can also be performed locally by each agent and benefit from receiving information from all agents in the system. Distillation is performed before training on the agent's own experience (i.e., once per episode or another selected interval). The number of epochs for distillation is determined by the following formula:

$$ep = ep_{min} + ((ep_{max} - ep_{min}) \frac{(R_{teacher} - R_{student})}{(R_{max} - R_{min})})^{expPwr} \quad (1)$$

where ep_{min} and ep_{max} denote the minimum and maximum number of policy-distillation epochs set by the developer; $R_{teacher}$ and $R_{student}$ are the rewards collected by the teacher and student during the evaluation period; R_{min} and R_{max} are the minimum and maximum possible rewards in that period; and $expPwr$ controls the nonlinearity of the dependence between the performance gap and the number of epochs. If this parameter is not equal to 1, it is advisable to additionally clip the resulting epoch value to the interval $[ep_{min}, ep_{max}]$.

It is worth noting that the proposed methods work well together and complement one another. However, they do not account for scenarios where a less efficient agent may possess qualitatively different but still useful experience. Prior work [16] and our own experiments (not included in this article) indicate that filtering agents based on experience diversity can also improve performance, though leveraging such qualitative differences may require specialized environments or coordination mechanisms. Selecting the best-performing agent as the teacher remains the most stable and broadly applicable strategy.

The study also considers random sampling from the replay buffer both for training on the agent's own experience and for constructing pseudo-labeled datasets. The method can also be combined with Prioritized Experience Replay (PER), which theoretically accelerates learning, and it is not restricted from being used with DDQN or other DQN modifications

(in the experiments, only online networks participated in knowledge transfer).

Finally, it should be noted that knowledge transfer is implemented through Policy Distillation. Although the experiments in this article use DQN (where the model outputs Q-values), the literature shows that policy distillation can be applied to value functions $V(s)$ or actor logits. Therefore, the method is not limited to the specific setting described here and can be applied in other scenarios as well.

2.1. Experiments setup

The experiments were conducted in the classical OpenAI Gym CartPole-v1 environment [2]. The task requires keeping a pole balanced in an upright position on a moving cart by applying force to the left or right. The agent received a reward of +1 for every step in which the pole remained within the allowable angular range (±24 degrees) and horizontal displacement (±4.8 units). A step that resulted in episode termination (as defined by the environment) incurred a penalty of −15. Additionally, each episode was capped at a maximum length of 500 steps. A portion of the experiments was also performed in the OpenAI Gym LunarLander-v3 environment [2], where the maximum episode length was limited to 1000 steps. In this environment, the reward structure was entirely defined by the environment and was not modified.

All experiments used a system of three agents that learned sequentially, episode by episode. Each agent interacted with its own independent environment instance (corresponding to the IDQN architecture). For most experiments, the agents' episodes were synchronized—meaning all three agents first completed their episodes before any training or knowledge sharing occurred. Each agent maintained its own replay buffer.

Given restrictions on the use of online TD(0), the experiments employed Batch TD(0), or offline TD(0). Specifically, for simplicity, training was performed at the end of each episode using a randomly sampled batch of transitions from the replay buffer. Pseudo-labeled datasets for knowledge transfer were also generated by sampling random transitions from the teacher's replay buffer in the form *{state, estimated Q}*.

Mean Squared Error (MSE) was used as the loss function for updating the Q-function parameters. The discount factor in the Bellman equation was set to 0.95, enabling the agent to account for future rewards during decision-making [13]. To improve learning stability, a non-prioritized replay buffer was used [8], where transitions were stored in the form {state, action, reward, next state, done}. Each experiment was repeated at least 30 times with different random seeds to avoid initialization bias and ensure the reliability of conclusions [5]. Some parameters differed between CartPole and LunarLander due to the varying complexity of the environments (Table 1).

The plots presented later may show two primary metrics:

- the average of rewards collected by each of the three agents in a given episode,
- the maximum of reward collected by any of the three agents in that episode.

Table 1. CartPoleV1 vs LunarLander-v3 experiment setups

	CartPole	LunarLander
NN hidden layers	RELU(24), RELU(12)	RELU(128), RELU(64), RELU(32)
Self-learning batch size	128	64
Sharing knowledge batch size	128	64
ε-greedy decay (decay per episode)	0.95	0.995
Reply buffer size	10000	15000
R_{min}	0	-300
R_{max}	500	300

Reinforcement learning is inherently unstable, and the performance of individual agents can vary significantly across runs. Therefore, an increase in the average reward may indicate that all agents improved or that weaker agents caught up to the stronger ones. An increase in the maximum reward indicates that the best agent in the system learned faster – which means the proposed knowledge-sharing method not only improves weaker agents but also accelerates the learning of the strongest one. A moving average is applied to the plots to clearly illustrate overall trends.

3. Knowledge sharing results

A series of experiments was conducted in the CartPole environment. Fig. 1 and 2 show the growth of rewards obtained by the three-agent system over 500 consecutive episodes. In Fig. 1, each point represents the average reward collected by each of the three agents in a given episode. In Fig. 2, each point represents the maximum reward collected by any of the three agents in that episode. After each episode, the agents may share knowledge (if enabled) and must perform learning on their own past experience.

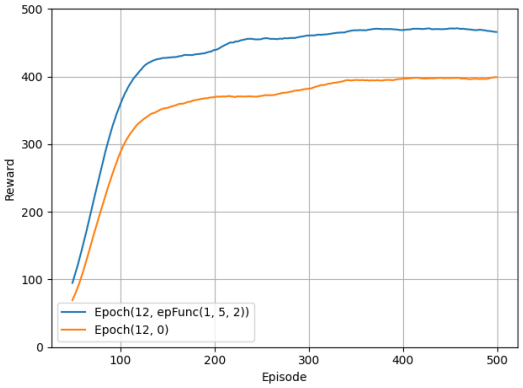

Fig. 1. MA(30) – Average of rewards collected by each of 3 agents in a given episode. CartPole-v1

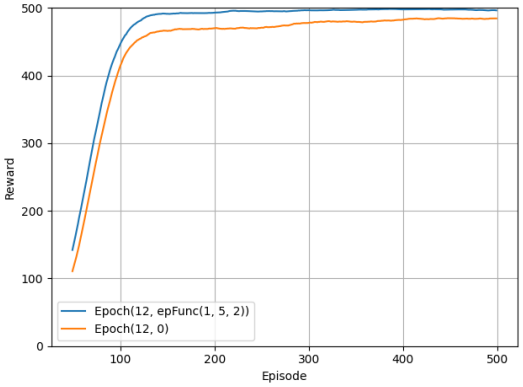

Fig. 2. MA(30) – Max of rewards collected by each of 3 agents in a given episode. CartPole-v1

The line labeled *Epoch(12, 0)* represents a system in which agents trained only on their own experience (12 epochs per episode) without any knowledge sharing. This curve shows the best performance achievable without sharing, obtained by tuning only the number of learning epochs (all other parameters were kept identical across experiments and significantly influenced the outcome).

The line labeled *Epoch(12, epFunc(1, 5, 2))* represents a system in which agents trained on their own experience for 12 epochs and additionally shared knowledge using teacher selection and dynamic intensity control. This curve reflects the best performance achieved with knowledge sharing, taking into account random teacher selection, a constant number of distillation epochs, and other variations tested.

The parameters of the function determining the number of distillation epochs were set as: $ep_{min} = 1$, $ep_{max} = 5$, $expPwr = 2$.

During the experiments, several additional observations were made:

- When agents receive insufficient training on their own experience, the improvement in both average and maximum episodic rewards grows more noticeably from episode to episode;
- Random teacher selection (with a constant number of distillation epochs = 1) improves the system's average performance, but it is inferior to selecting the most effective agent as the teacher (Fig. 3);
- A constant number of distillation epochs performs worse than dynamically determining the number of epochs based on performance differences (Fig. 4);
- All-to-all learning (with constant 1 epoch for policy distillation) doesn't show stable positive effect on learning efficiency in current setup for CartPole experiments (Fig. 5);
- The best performance is achieved when the number of self-learning epochs is optimally selected and the proposed knowledge-sharing methods are used simultaneously;
- Copying weights of the most efficient agent makes learning efficiency event worse;
- DDQN in this configuration does not provide advantages over the baseline DQN. However, when knowledge sharing is applied to the online network, it yields a 15% increase in average reward compared to DDQN without sharing.

Similar experiments were conducted in the LunarLander environment (Fig. 6 and Fig. 7). In this case, the baseline number of self-learning epochs was 18, and $ep_{max} = 15$. Across all experiments, the nonlinear relationship between the performance gap and the number of distillation epochs consistently produced the best results.

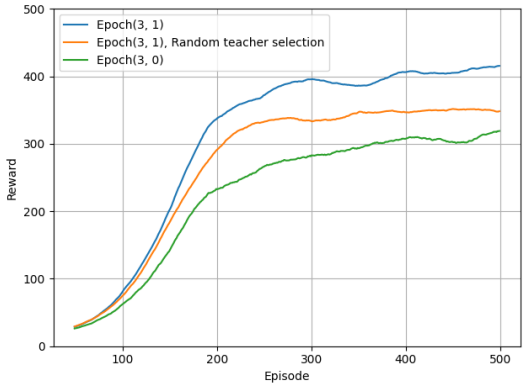

Fig. 3. MA(30) – Average of rewards collected by each of 3 agents in a given episode. CartPole-v1

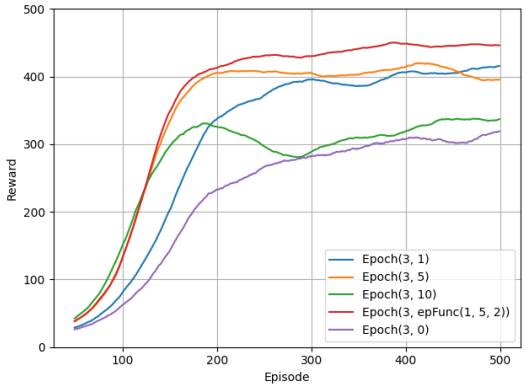

Fig. 4. MA(30) – Average of rewards collected by each of 3 agents in a given episode. CartPole-v1

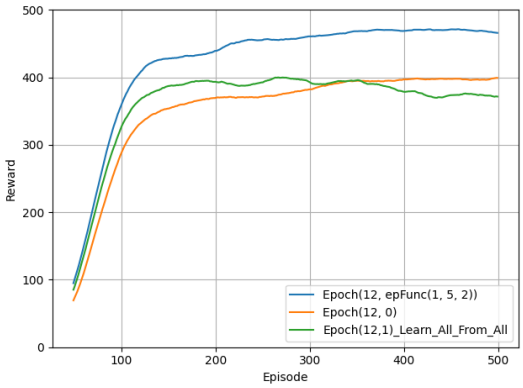

Fig. 5. MA(30) – Average of rewards collected by each of 3 agents in a given episode. CartPole-v1

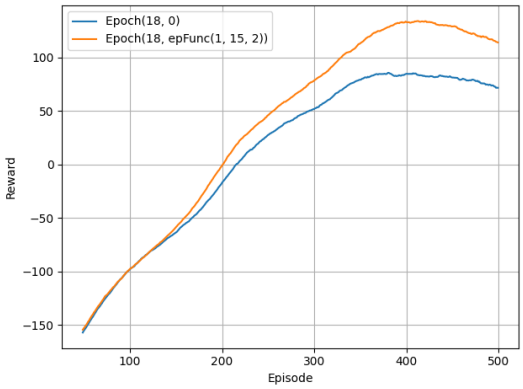

Fig. 6. MA(50) – Average of rewards collected by each of 3 agents in a given episode. LunarLander-v3

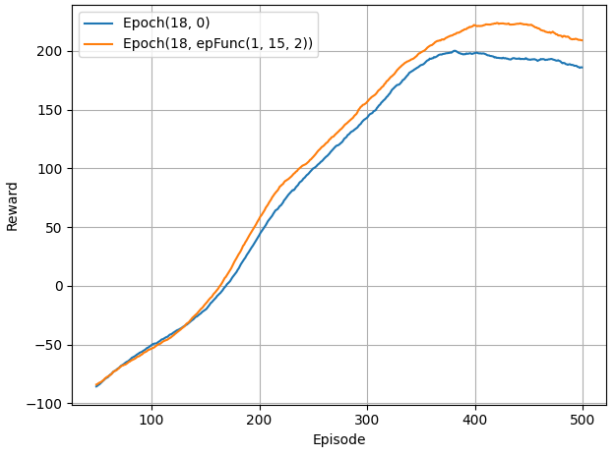

Fig. 7. MA(50) – Max of rewards collected by each of 3 agents in a given episode. LunarLander-v3

As a rough measure for comparing the impact of the methods, the normalized area under the curve (AUC) can be used (where AUC = 1 indicates that the agent achieved the maximum possible reward in every episode, and AUC = 0 indicates the minimum). Considering the computed relative error for the 95% confidence interval, the following results were obtained (Table 2). The improvement of normalized AUC compared to the experiment without knowledge sharing (just 3 independent agents) in same env listed in Table 3. It is important to note that this metric should be interpreted only as an approximate indicator, as it tends to underestimate gained enhancement. If an agent improves over time, its maximum achievable performance naturally increases, yet the normalized AUC assumes that the maximum reward is attainable from the very first episode. Additionally, ε-greedy exploration introduces extra noise at the beginning of training. So, the relatively small difference in AUC might not display the noticeable difference in charts.

Moreover, comparisons between different environments are not entirely fair because their parameters differ. Nevertheless, both the plots and the Normalized AUC values clearly indicate that knowledge sharing – using the proposed teacher-selection and intensity-control methods – positively affects the learning speed of the system, improving both the best-performing agent and the average performance across agents.

Table 2. Normalized AUC with 95% CI

Env	Agg Type	Line	Normalized AUC
LunarLander	Max	Epoch(18, epFunc(1, 15, 2))	0.670 ± 0.007
LunarLander	Max	Epoch(18, 0)	0.648 ± 0.006
LunarLander	Avg	Epoch(18, epFunc(1, 15, 2))	0.550 ± 0.010
LunarLander	Avg	Epoch(18, 0)	0.510 ± 0.007
CartPole	Max	Epoch(12, epFunc(1, 5, 2))	0.911 ± 0.007
CartPole	Max	Epoch(12, 0)	0.868 ± 0.026
CartPole	Avg	Epoch(12, epFunc(1, 5, 2))	0.822 ± 0.016
CartPole	Avg	Epoch(12, 0)	0.684 ± 0.0343

Table 3. Normalized AUC improvement relative to the Best non-sharing baseline (with CI considered)

Env	Agg Type	Line	Normalized AUC
LunarLander	Max	Epoch(18, epFunc(1, 15, 2))	[1.38, 5.13]%
LunarLander	Avg	Epoch(18, epFunc(1, 15, 2))	[4.45, 11.35]%
CartPole	Max	Epoch(12, epFunc(1, 5, 2))	[1.01, 9.03]%
CartPole	Avg	Epoch(12, epFunc(1, 5, 2))	[12.26, 28.92]%

4. Summary

The results of the study confirm the effectiveness of knowledge sharing among independent agents operating under the DQN algorithm in a stationary environment. The proposed teacher-selection method and the dynamic control of knowledge-transfer intensity demonstrated the ability to accelerate agent learning and improve overall performance compared to classical DQN without sharing, random teacher selection, all-to-all learning, or constant-intensity transfer.

These methods enable the system to leverage the resources of parallel learning across multiple agents to form a single more efficient agent or a more capable multi-agent system. Importantly, the reliability of the findings is supported by repeated experimental runs with different random seeds, ensuring the robustness of the conclusions.

The proposed knowledge-sharing mechanisms can also be applied to other DQN variants, such as Dueling DQN [17], Prioritized Experience Replay, and their combinations. This opens opportunities for further experiments involving different architectures, knowledge-exchange frequencies, and degrees of external experience influence. Future research may also explore adaptive knowledge-sharing strategies that account for qualitative differences in agent experience.

References

[1] Bellemare, M. G., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., & Munos, R. (2016). *Unifying Count-Based Exploration and Intrinsic Motivation* (arXiv:1606.01868). arXiv. https://doi.org/10.48550/arXiv.1606.01868

[2] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). *OpenAI Gym* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1606.01540

[3] Dahal, M., & Vaezi, M. (2025). *Selective Experience Sharing in Reinforcement Learning Enhances Interference Management* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2501.15735

[4] Gao, Z., Xu, K., Ding, B., Wang, H., Li, Y., & Jia, H. (2021). *KnowSR: Knowledge Sharing among Homogeneous Agents in Multi-agent Reinforcement Learning* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2105.11611

[5] Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2017). *Deep Reinforcement Learning that Matters* (Version 3). arXiv. https://doi.org/10.48550/ARXIV.1709.06560

[6] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

[7] Qiao, D., Li, W., Yang, S., Zha, H., & Wang, B. (2025). *Offline Multi-agent Reinforcement Learning via Score Decomposition* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2505.05968

[8] Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). *Prioritized Experience Replay* (Version 4). arXiv. https://doi.org/10.48550/ARXIV.1511.05952

[9] Shi, D., Tong, J., Liu, Y., & Fan, W. (2022). Knowledge Reuse of Multi-Agent Reinforcement Learning in Cooperative Tasks. *Entropy*, *24*(4), 470. https://doi.org/10.3390/e24040470

[10] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489. https://doi.org/10.1038/nature16961

[11] Singh, A., et al. (2020). Reinforcement learning for autonomous traffic signal control. Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS).

[12] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (Second edition). The MIT Press.

[13] Tokic, M. (2010). Adaptive ε-Greedy Exploration in Reinforcement Learning Based on Value Differences. In R. Dillmann, J. Beyerer, U. D. Hanebeck, & T. Schultz (Eds), *KI 2010: Advances in Artificial Intelligence* (Vol. 6359, pp. 203–210). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-16111-7_23

[14] Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep Reinforcement Learning with Double Q-Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1). https://doi.org/10.1609/aaai.v30i1.10295

[15] Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., … Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, *575*(7782), 350–354. https://doi.org/10.1038/s41586-019-1724-z

[16] Wadhwania, S., Kim, D.-K., Omidshafiei, S., & How, J. P. (2019). *Policy Distillation and Value Matching in Multiagent Reinforcement Learning* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1903.06592

[17] Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M. & Freitas, N.. (2016). Dueling Network Architectures for Deep Reinforcement Learning. *Proceedings of The 33rd International Conference on Machine Learning*, 48. https://proceedings.mlr.press/v48/wangf16.html

M.Sc. Viacheslav Bochok
e-mail: vybochok@gmail.com

Ph.D. student of Software Engineering of Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine. The research is focused on the field of AI.

https://orcid.org/0009-0000-3929-2758

Prof. Nataliia V. Fedorova
e-mail: fedorova_natalia@lll.kpi.ua

Doctor of Technical Sciences, professor in the Department of Software Engineering for Power Industry at the National Technical University of Ukraine. Official member of the Ukrainian Academy of Higher Education Sciences and member of the IEEE Computer Society.
Research field: intelligent systems, sensor networks, software engineering, internet of things, network programming.

https://orcid.org/0000-0002-4548-4198