DOI 10.37882/2223-2966.2025.04.06

РОБОТИЗИРОВАННОЕ ОБУЧЕНИЕ НА ОСНОВЕ Q-LEARNING И ANN С ИСПОЛЬЗОВАНИЕМ ДЕМОНСТРАЦИЙ (ADQN)

ROBOTIC LEARNING BASED ON Q-LEARNING AND ANN USING DEMONSTRATIONS (ADQN)

Gao Tianci
Bo Yang
Shengren Rao

Summary. In the field of robotic systems training, the classical approach based on pure Q-Learning requires a large volume of trial-and-error interactions. This not only reduces efficiency but can also be unsafe when working with real robots. In this paper, we propose an ADQN (Augmented Deep Q-Network) method that combines Q-Learning, an artificial neural network (ANN), and demonstration data. In the first phase, the Q-network is trained offline on expert trajectories. Then, during the online phase, TD updates and Margin-based supervision on demonstration actions are used simultaneously. This approach accelerates the convergence of the algorithm and increases overall success rates. We compare ADQN with two baseline methods: (1) pure DQN (no demonstrations) and (2) pure imitation (ANN). Experiments in the MATLAB/Simulink environment and on a real Kinova Gen3 robot show that ADQN achieves higher performance and reaches target results faster. We also analyze the impact of prioritized replay and various modules of the algorithm. The results confirm that the proposed approach effectively combines the advantages of reinforcement learning and demonstration-based training.

Keywords: reinforcement learning, Q-Learning, artificial neural networks, learning from demonstrations, robotic manipulator.

Гао Тяньцы
аспирант, Московский государственный технический университет имени Н.Э. Баумана
Gaotianci0088@gmail.com
Бо Ян
аспирант, Московский государственный технический университет имени Н.Э. Баумана
yangbo.123@hotmail.com
Шэнжэнь Жао
аспирант, Московский государственный технический университет имени Н.Э. Баумана
raoshengren@gmail.com

Аннотация. В области обучения робототехнических систем классический подход на основе чистого Q-обучения (Q-Learning) требует больших объёмов проб и ошибок, что снижает эффективность и может быть небезопасно при работе с реальными роботами. В данной работе предлагается метод ADQN (Augmented Deep Q-Network), совмещающий Q-Learning, искусственную нейронную сеть (ANN) и данные демонстраций. На первом этапе Q-сеть обучается офлайн на экспертных траекториях, затем в ходе онлайн-фазы одновременно используются TD-обновления и Margin-супервизия по демонстрационным действиям. Такая схема ускоряет сходимость алгоритма и повышает итоговую успешность. Мы сравниваем ADQN с двумя базовыми методами: (1) чистым DQN (без демонстраций) и (2) чистой имитацией (ANN). Эксперименты в среде MATLAB/Simulink и на реальном роботе Kinova Gen3 показывают, что ADQN достигает более высоких показателей и быстрее выходит на целевые результаты. Дополнительно проанализировано влияние приоритетного реплея и различных модулей алгоритма. Результаты подтверждают, что рассматриваемый подход эффективно совмещает преимущества обучения с подкреплением и демонстраций.

Ключевые слова: обучение с подкреплением, Q-Learning, искусственные нейронные сети, обучение по демонстрациям, робот-манипулятор.

Введение

Роботизированные системы становятся всё более автономными благодаря методам обучения с подкреплением (Reinforcement Learning, RL) [1]. Однако в классических методах, например Deep Q-Network (DQN) [2], требуется большое число шагов случайного исследования, что зачастую неприемлемо при работе с реальными роботами (риск повреждений, большая продолжительность экспериментов и т. д.).

Одно из возможных решений — обучение по демонстрациям (Learning from Demonstration, LfD) [3, 4]. Суть метода состоит в том, чтобы показать агенту предпочитаемые экспертом действия в ряде типичных сценариев. Тем не менее, простая имитация (без учёта вознаграждения) не даёт улучшения сверх действий эксперта [5]. Поэтому целесообразно совмещать имитацию и RL-обучение.

В данной работе мы предлагаем метод ADQN (Augmented Deep Q-Network), в котором:

1. На первом этапе выполняется офлайн-обучение (предобучение) Q-сети на ограниченном наборе экспертных переходов.
2. На втором этапе (онлайн) одновременно применяются TD-обновления (Q-Learning) и Margin-супервизия для демонстрационных данных.
3. Используется приоритетный реплей, в котором примерам из демонстраций назначается повы-

шенный приоритет. Это ускоряет начальную фазу обучения и предотвращает «забывание» экспертных действий.

Мы исследуем эффективность ADQN на задачах позиционирования конечного звена робот-манипулятора (6 DoF), реализованных в MATLAB/Simulink, а также проводим контрольные эксперименты на реальном роботе Kinova Gen3. Сравнение с DQN (без демонстраций) и чистой имитацией (ANN) показывает, что ADQN достигает более высоких результатов и быстрее сходится.

Теоретические основы

1. *Q-Learning и DQN*

Пусть задача задана Марковским процессом принятия решений с состояниями s, действиями a, функцией вознаграждения $R(s,a)$ и фактором дисконтирования $\gamma \in [0,1)$. Оптимальная Q-функция $Q^*(s,a)$ удовлетворяет уравнению Беллмана:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s} P(s'|s,a) \max_{a} Q^*(s',a') \quad (1)$$

Классический алгоритм Q-Learning итеративно обновляет оценку Q-функции по формуле:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a} Q(s',a') - Q(s,a) \right] \quad (2)$$

где α — скорость обучения, r — мгновенная награда, s — следующее состояние. Для высокоразмерных задач используют нейронную аппроксимацию $Q(s,a;\theta)$ (DQN), где минимизируется TD-потеря:

$$L_{\text{DQN}}(\theta) = \mathrm{E}_{(s,a,r,s')} \left[\left(y_{\text{target}} - Q(s,a;\theta) \right)^2 \right] \quad (3)$$

$$y_{\text{target}} = r + \gamma \max_{a} Q(s',a';\theta^-) \quad (4)$$

Параметры θ^- – (целевой сети) обновляются реже для стабилизации обучения. Мини-батчи для обучения выбираются из реплей-буфера.

На рис. 1 представлена обобщённая блок-схема алгоритмов DQN и ADQN. Пока что в разделе 2.1 мы описываем классические аспекты DQN. В разделе 2.3 будет показано, каким образом ADQN расширяет DQN за счёт использования данных демонстраций.

2. *Супервизия эксперта*

Если имеются демонстрации $\{(s,a_E) \in D_{\text{demo}}\}$, их можно использовать для обучения супервизора по методу BC (Behavioral Cloning):

$$L_{\text{BC}}(\phi) = \frac{1}{N} \sum_{(s,a_E) \in D_{\text{demo}}} \ell\left(\pi(s;\phi), a_E\right) \quad (5)$$

где ℓ — функция потерь, например кросс-энтропия или MSE. Однако чистое BC не учитывает вознаграждение среды. Более продвинутая Margin-супервизия [6] для Q-сети требует, чтобы $Q(s,a_E)$ превосходило $Q(s,a)$ на некоторую константу $\Delta > 0$:

$$L_{\text{sup}}(\theta) = \mathrm{E}_{(s,a_E) \in D_{\text{demo}}} \left[\sum_{a \neq a_E} \max\left\{0, Q(s,a;\theta) + \Delta - Q(s,a_E;\theta)\right\} \right], \quad (6)$$

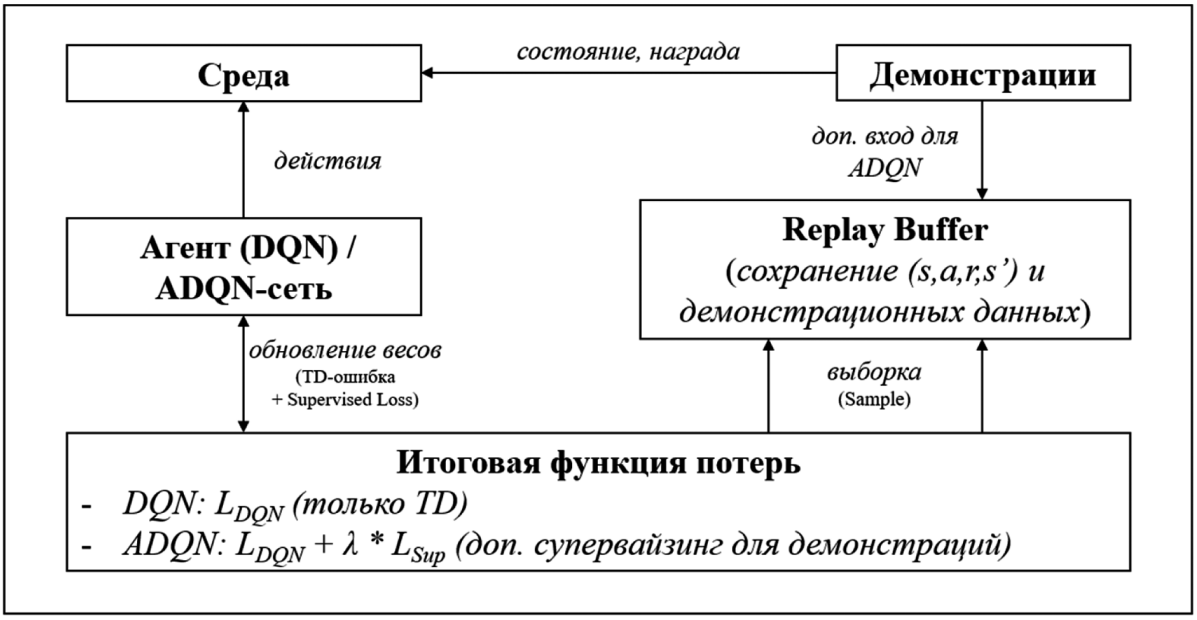

Рис. 1. Обобщённая блок-схема. Разница DQN (только TD-ошибка) и ADQN (TD + супервизия по действиям эксперта)

3. *Предложенный метод ADQN*

Предлагаемый алгоритм ADQN добавляет термин супервизии в функцию потерь DQN. Итоговая формула:

$$L_{\text{ADQN}}(\theta) = L_{\text{DQN}}(\theta) + \lambda L_{\text{sup-demo}}(\theta), \quad (7)$$

где λ — коэффициент, регулирующий вклад демонстраций. При онлайн-обучении для выборки (s_i, a_i, r_i, s_i') вычисляется TD-ошибка:

$$\delta_i = y_i - Q(s_i, a_i; \theta), y_i = r_i + \gamma max Q(s_i', a; \theta^-), \quad (8)$$

Тогда суммарная функция потерь сочетает δ_i^2 и Margin-слагаемое (только для демо-примеров):

$$L_{\text{ADQN}}(\theta) = \\ = \frac{1}{B}\sum_{i=1}^{B}(\delta_i^2) + \lambda \sum_{i:(s_i,a_i)\in \mathcal{D}_{\text{demo}}} \text{MarginLoss}(Q, s_i, a_E). \quad (9)$$

На рис. 2 показана двухэтапная схема обучения:

1. **Офлайн-предобучение** на демонстрационных переходах.
2. **Онлайн-обучение** с объединением новых данных и демонстраций в общем буфере, где приоритетный реплей увеличивает частоту выбора демо-примеров.

Основные этапы:

1. **Сбор демонстраций:** оператор (или другая готовая стратегия) генерирует набор успешных переходов (s, a_E).
2. **Офлайн-предобучение:** сеть обучается на этих демо-переходах, используя L_{DQN} и L_{sup}.
3. **Онлайн-фаза:** агент действует в среде, записывая новые переходы в реплей-буфер вместе с демонстрационными. Для некоторых переходов (демо) рассчитывается Margin-потеря, для всех — TD-потеря.
4. **Приоритетный реплей:** демо-примеры получают повышенный приоритет, чтобы не «затеряться» среди новых выборок.

Таким образом, метод ADQN совмещает достоинства обучения с подкреплением (поиск действий, потенциально более выгодных, чем у эксперта) и имитации (быстрый старт и снижение количества случайных ошибок).

Эксперименты

1. *Архитектура Q-сети*

В ADQN используется двухслойная полносвязная нейронная сеть (MLP) с 128 нейронами (ReLU) в каждом скрытом слое, см. рис. 3. На вход подаются признаки состояния (например, 10 скалярных признаков), на выходе — *Q*-значения для 4 дискретных действий (в иллюстративном примере). Для демонстрационных примеров добавляется Margin Loss (см. формулу (6)).

2. *3D-визуализация манипулятора*

Для проверки работоспособности метода применялась среда MATLAB/Simulink с моделью шестистепенно-

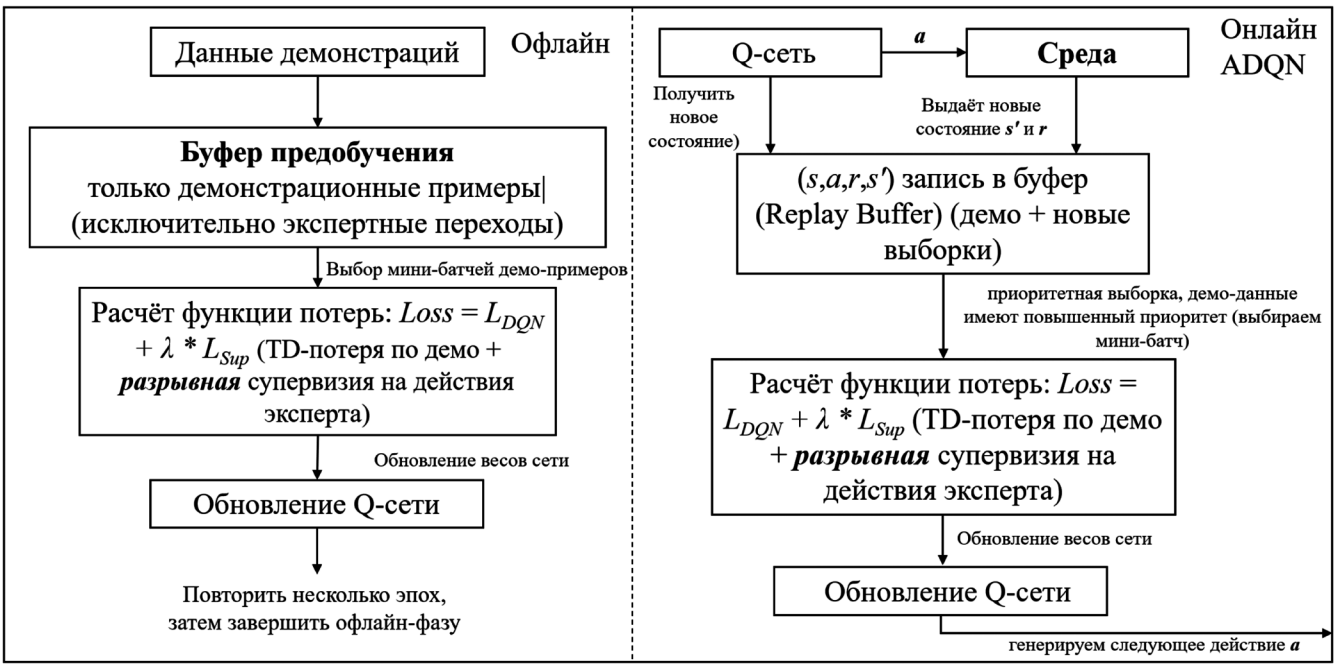

Рис. 2. Общая схема обучения ADQN

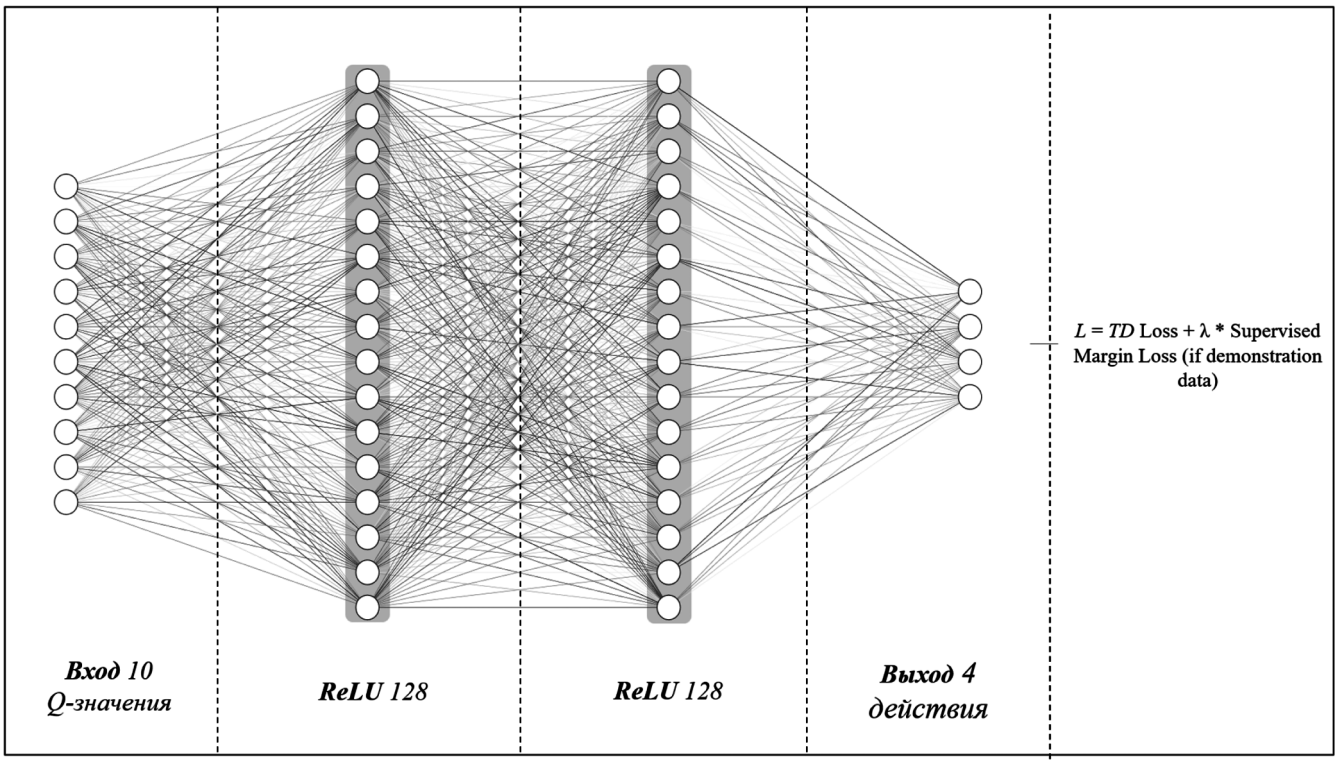

Рис. 3. Архитектура сети (двухслойная MLP с 128 нейронами в каждом скрытом слое)

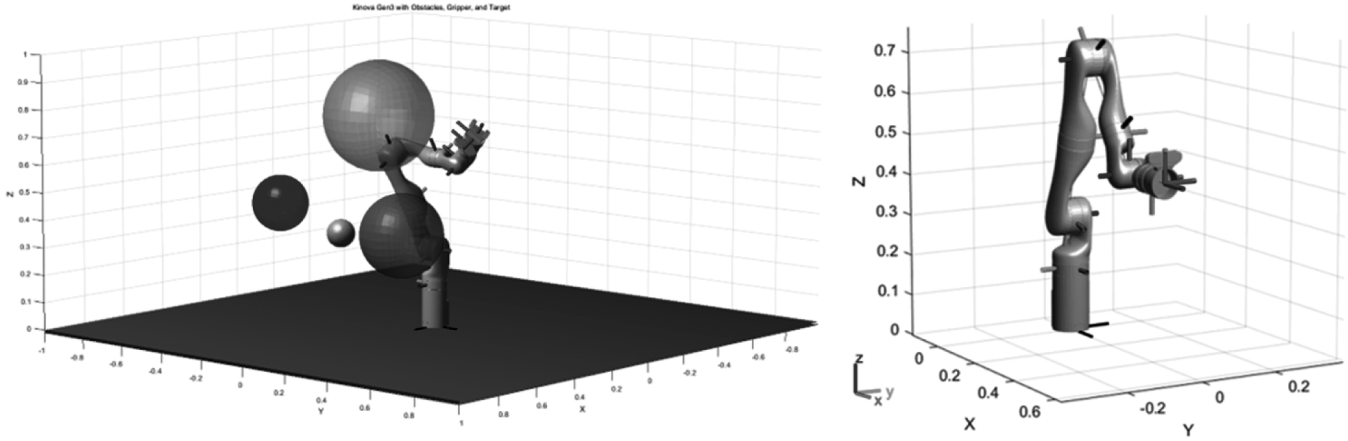

Рис. 4. 3D-визуализация среды с роботом (Kinova Gen3) и препятствиями (слева) и упрощённая модель (справа)

го робота-манипулятора. На рис. 4 показан пример сцены с несколькими сферическими препятствиями (слева) и упрощённая визуализация (справа). Цель состоит в том, чтобы переместить инструмент из начального положения в целевую точку и получить положительную награду +100.

Результаты экспериментов

1. *Сравнение ADQN, ANN и DQN*

На рис. 5 представлена кривая обучения (доля успешных эпизодов, усреднённая и сглаженная по нескольким испытаниям) в зависимости от числа шагов. Видно, что чистый DQN (базовый вариант) начинается с ~0 % и лишь после 6–7 $\times 10^5$ шагов достигает ~60–70 %. Алгоритм ANN (чистая имитация) стартует с ~50–60 %, но почти не улучшает результат. Предложенный метод ADQN уже на начальных этапах достигает 60 %+ и далее выходит на 80–90 %, превосходя оба базовых способа.

2. *Исследование отдельных модулей (абляция)*

На рис. 6 и 7 представлены результаты при отключении отдельных элементов ADQN.

На рис. 6 видно: если использовать демонстрации только в офлайн-режиме («Только предобучение»), конечный результат почти совпадает с обычным DQN. Лишь «Полный ADQN» (где демо сохраняются и в онлайн-фазе) даёт рост до ~90–95 %.

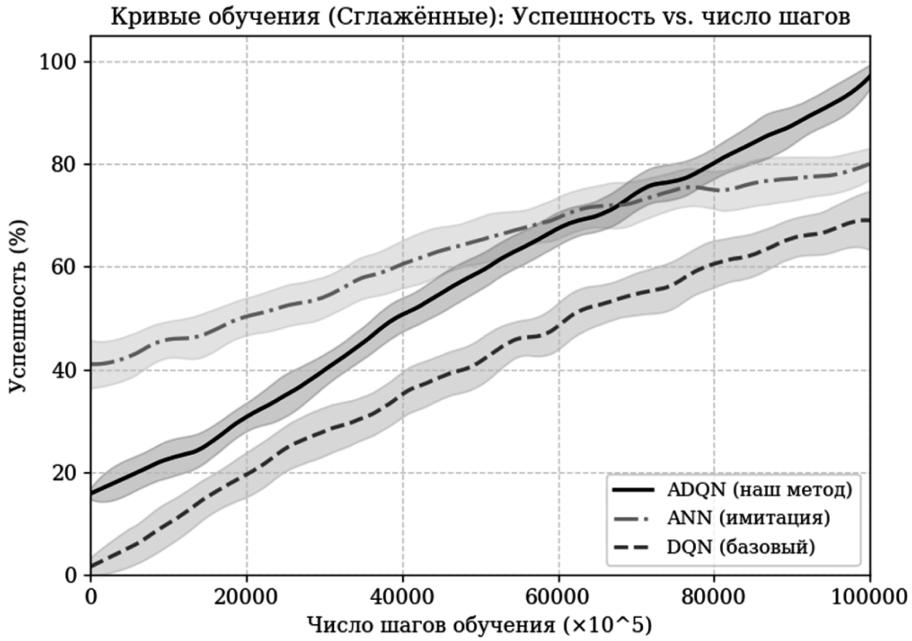

Рис. 5. Кривые обучения (ADQN, ANN, DQN): преимущество ADQN как на ранних этапах, так и по конечной успешности

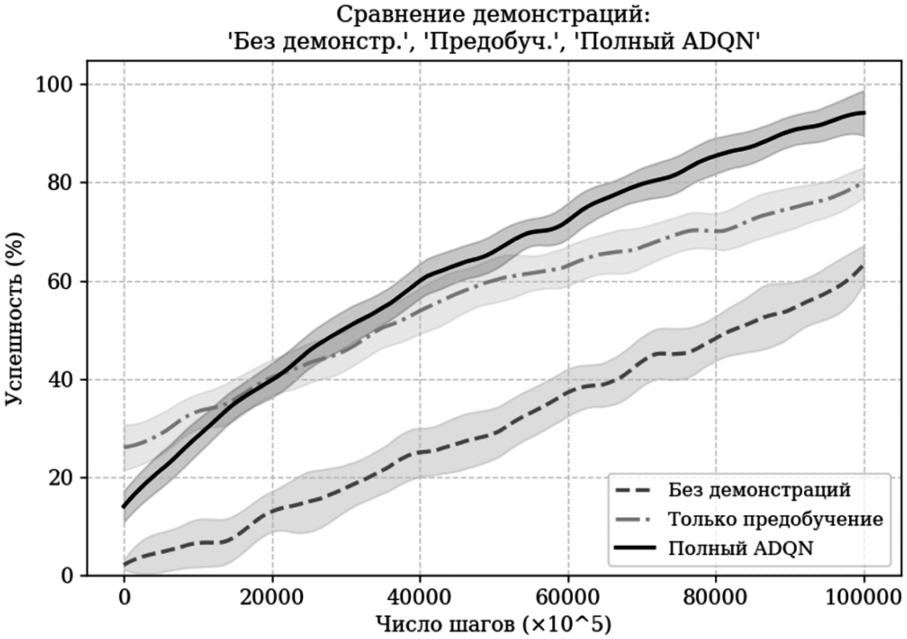

Рис. 6. Сравнение демонстраций: «Без демонстраций», «Только предобучение», «Полный ADQN»

На рис. 7 показано, что приоритетная выборка демо-переходов ускоряет рост успешности и обеспечивает более высокий конечный результат.

3. *Сравнение с другими алгоритмами RL*

Для полноты экспериментов ADQN был также сопоставлен с PPO, A2C и SAC. Как показано на рис. 8, ADQN даёт более высокую итоговую успешность, большую суммарную награду и при этом требует меньше шагов для сходимости.

Выводы

В работе предложен метод ADQN, совмещающий:

1. **Q-обучение** с TD-ошибкой,
2. **Margin-супервизию** действий эксперта,
3. **Приоритетный реплей** для более частого использования демо-переходов.

Эксперименты по задаче позиционирования манипулятора и контрольные тесты на реальном роботе Kinova Gen3 показывают, что ADQN даёт значимый выигрыш по сравнению с чистым DQN и чистой имитацией (ANN).

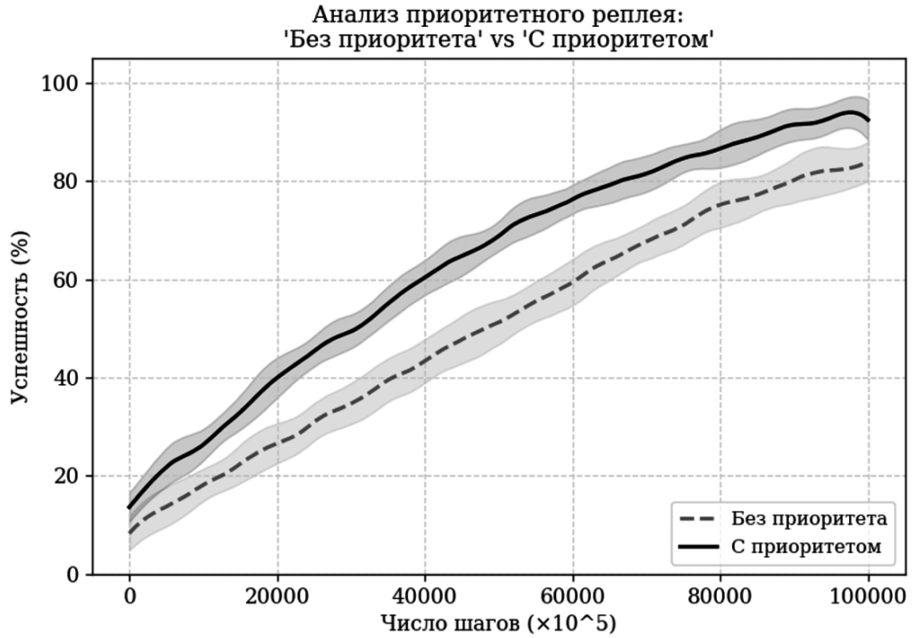

Рис. 7. Анализ приоритетного реплея: «Без приоритета» vs «С приоритетом»

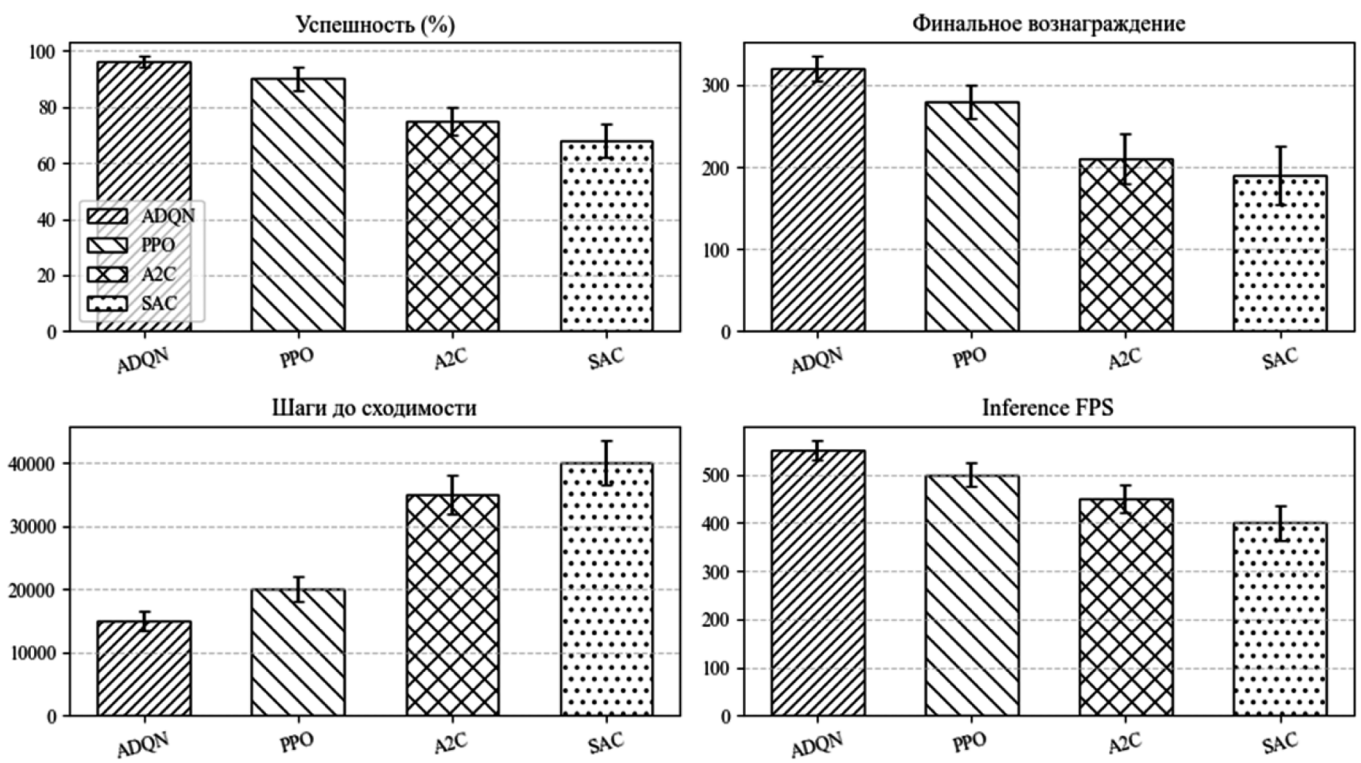

Рис. 8. Сравнение нескольких алгоритмов (PPO, A2C, SAC) по четырём метрикам. ADQN показывает преимущество как по успеху и вознаграждению, так и по скорости сходимости.

Метод обеспечивает высокий стартовый уровень (благодаря демонстрациям) и в то же время способен существенно превзойти эксперта за счёт дальнейшего обучения с подкреплением.

Основные направления будущих исследований:

1. Расширение на непрерывные пространства действий (напр. DDPG или SAC + демо).
2. Учёт ошибок и неточностей в демонстрациях, включая автоматическую фильтрацию «плохих» примеров.
3. Онлайн-демонстрации: расширение алгоритма для интерактивной работы с «учителем».
4. Разработка безопасных стратегий обучения на физическом роботе, минимизирующих рискованную динамику в ранних фазах экспериментов.

ЛИТЕРАТУРА

1. Манакитса Н., Мараслидис Г.С., Мойсис Л. и др. (2024). Обзор методов машинного обучения и глубокого обучения для обнаружения объектов, семантической сегментации и распознавания действий человека в задачах машинного и робототехнического зрения. Technologies, 12(2): 15.
2. Осбанд И., Бланделл К., Притцель А. и др. (2016). Глубокая исследовательская стратегия посредством Bootstrapped DQN. Advances in Neural Information Processing Systems, 29.
3. Мандлекар А., Сюй Д., Вонг Дж. и др. (2021). Что важно в обучении офлайн на человеческих демонстрациях для манипуляции роботами. arXiv preprint, arXiv:2108.03298.
4. Ду У., Дин С. (2021). Обзор многоагентного глубокого обучения с подкреплением: проблемы и приложения. Artificial Intelligence Review, 54(5): 3215–3238.
5. Чжэн Б., Верма С., Чжоу Дж. и др. (2022). Имитационное обучение: прогресс, таксономии и проблемы. IEEE Transactions on Neural Networks and Learning Systems, 35(5): 6322–6337.
6. Чэн Ю., Лам Ч. Т., Пау Г. и др. (2025). Из виртуального в реальность: решение на основе глубокого обучения с подкреплением для реализации автономного вождения с использованием 3D-LiDAR. Applied Sciences, 15(3): 1423.

© Гао Тяньцы (Gaotianci0088@gmail.com); Бо Ян (yangbo.123@hotmail.com); Шэнжэнь Жао (raoshengren@gmail.com)

Журнал «Современная наука: актуальные проблемы теории и практики»