ВЕСТНИК ВОЕННОГО ИННОВАЦИОННОГО ТЕХНОПОЛИСА «ЭРА», 2022, том 3, № 3, с. 282–288

ЭЛЕКТРОННАЯ КОМПОНЕНТНАЯ БАЗА И ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

УДК 004.85

МЕХАНИЗМ ЛЮБОПЫТСТВА НА ОСНОВЕ ФУНКЦИИ ЦЕННОСТИ В ОНЛАЙН ОБУЧЕНИИ С ПОДКРЕПЛЕНИЕМ

© 2022 г. А. В. Андроненко[1,*], М. С. Авшалумов[1], В. А. Дёмин[1]

[1] *Национальный исследовательский центр "Курчатовский институт", Москва, Россия*

*****E-mail: andronenko.andrey@bk.ru*

Поступила в редакцию 15.03.2022 г.
После доработки 20.03.2022 г.
Принята к публикации 20.03.2022 г.

Использование механизмов внутренней мотивации в обучении с подкреплением способствует более эффективному исследованию сред агентами, а также помогает лучше действовать в средах с разреженными наградами. Представлен новый вариант внутренней мотивации в виде дополнительного вознаграждения, предоставляемого за исследование состояний, ценность которых трудно предсказать с помощью упрощенного *critic*-модуля. Этот бонус суммируется с внешним вознаграждением, получаемым от окружающей среды, и комбинированное вознаграждение используется для непосредственного обучения агента. Предложенный механизм внутренней мотивации рассмотрен на примере его внедрения в алгоритм Asynchronous Advantage Actor Critic (A3C). При сравнении с классическим A3C-алгоритмом, модифицированный показывает большую скорость обучения для задачи Atari Pong, имея схожую архитектуру и сопоставимое число свободных параметров модели.

DOI: 10.56304/S2782375X22030032

ВВЕДЕНИЕ

Слабыми местами многих методов обучения с подкреплением являются эффективность использования данных для настройки весов алгоритма, работа с редкой наградой и баланс между эксплуатацией обученного алгоритма и разведкой новых данных, необходимых для дальнейшего обучения. Одним из используемых подходов, помогающим справиться в той или иной мере с каждой из обозначенных сложностей, является обучение с применением механизмов внутреннего любопытства, т.е. с положительным самоподкреплением состояний. Расчет дополнительной награды за любознательность может базироваться на статистике частоты посещений [1], трудности достижения состояний [2], различных формах предсказуемости [3, 4], оценке вклада состояния в изменение мировоззрения агента ("*informational gain*") [5]. Упомянутые работы показывают состоятельность и актуальность использования механизмов внутреннего подкрепления для ускорения обучения и решения проблемы "*exploration vs exploitation*" и поощряют исследование различных подходов такого рода.

Некоторые среды, в которых может применяться обучение с подкреплением (**RL**), изобилуют незначимыми состояниями [6, 7]. В игре *Pong* мяч летит достаточно медленно, отчего становится возможным отбить его, реагируя "в последний момент". Как следствие, в промежуточных состояниях принятые агентом решения не влияют на получение награды, что порождает проблемы с верной оценкой действий и состояний [8]. В то же время некоторые состояния, напротив, являются критически важными. В работе рассмотрен новый механизм любопытства, увеличивающий награду для важных состояний.

МАТЕРИАЛЫ И МЕТОДЫ

В качестве стартовой тестовой среды был выбран Pong-deterministic-v4 из пакета Atari gym [9] (**Pong**). На каждом шаге от агента ожидается выбор одного из шести действий: "NOOP", "FIRE", "RIGHT", "LEFT", "RIGHTFIRE", "LEFTFIRE". Цель игры – забить 21 мяч в "ворота" оппонента до того, как тот забьет 21 мяч в ворота агента. При этом лучшим считается агент, забивший 21 мяч, не пропустив при этом ни одного. Агент получает положительное подкрепление, равное 1, за забитые мячи и отрицательное, равное –1, за пропущенные. Потенциальная длительность игры может быть бесконечной, однако для экспериментов ставилось ограничение в 10000 кадров. Также для упрощения обучения проводилась предобработка входных изображений для всех используемых алгоритмов, включая обрезание не значащей части кадра (счет игры),

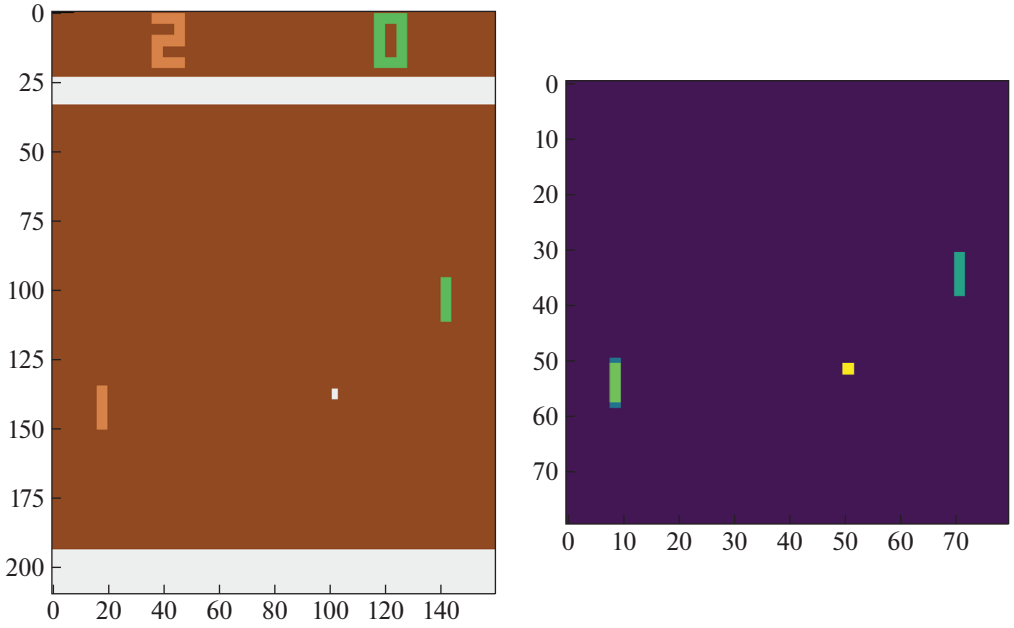

Рис. 1. Оригинальное и предобработанное входное изображение.

после чего изображения преобразовывались в один канал к размеру 80 на 80 пикселей (рис. 1).

В качестве оппонента для реализуемого агента выступает примитивный алгоритм игровой среды, стратегия которого заключается в постоянной минимизации расстояния между платформой и мячом по вертикальной оси, т.е. платформа непрерывно движется вниз или вверх вместе с движением мяча.

Предлагаемый метод базируется на алгоритме обучения с подкреплением Asynchronous Advantage Actor Critic (**A3C**) [10], однако реализуемая модель расширяется дополнительным блоком *R*, отвечающим за генерацию внутренней награды. Общая конфигурация сети содержит блок *S* (*encoder*) – сверточную сеть, отвечающую за кодирование состояния, и блоки *V* (*value*) *A* (*actor*) и *R* (*reward*) – рекуррентные сети, отвечающие за оценку функции *V*, предсказание вероятностей действий, и оценку функции *Q* соответственно (рис. 2).

Блоки *A* и *V* получают полную информацию о состоянии окружающей среды, в то время как блок *R* является "слепым" и получает только информацию о значениях, предсказанных блоками *V* и *A*.

В архитектуре A3C при обучении используются несколько асинхронно функционирующих агентов, которые передают свой опыт в "общую" модель после каждого обработанного пакета данных (*batch*). Здесь и далее размер этого пакета (*batch size*) обозначается как T.

Обозначим внутреннее вознаграждение, полученное агентом на шаге t, как r_t^i, а внешнее вознаграждение как r_t^e. Введенный блок *R* предсказывает значения функции *Q* для входного состояния s_t и минимизирует ошибку:

$$L_R = \sum_{n=1}^{T} L_n^R$$

$$L_n^R = 0.5 \times \left(\gamma^n q_{T+1} - q_{T-(n-1)} + \sum_{k=1}^{n} \gamma^{n-k} * r_{T-(k-1)}^e \right)^2,$$

где $q_{T-(n-1)}$ и q_{T+1} – значения функции *Q*, предсказанные блоком *R* для состояний $s_{T-(n-1)}$ и s_{T+1} соответственно, γ – коэффициент дисконтирования. Для вычисления внутренней награды r_t^i используются одношаговые ошибки δ предсказаний блока *R* с коэффициентом ε:

$$\delta = \gamma^* q_{t+1} - q_t + r_t^e.$$

$$r_t^i = \varepsilon^* \delta.$$

Блок критика *V* тренируется предсказывать ценность состояния с учетом внутренней награды $r_t = r_t^e + r_t^i$, оптимизируя функцию потерь:

$$L_V = \sum_{n=1}^{T} L_n^V$$

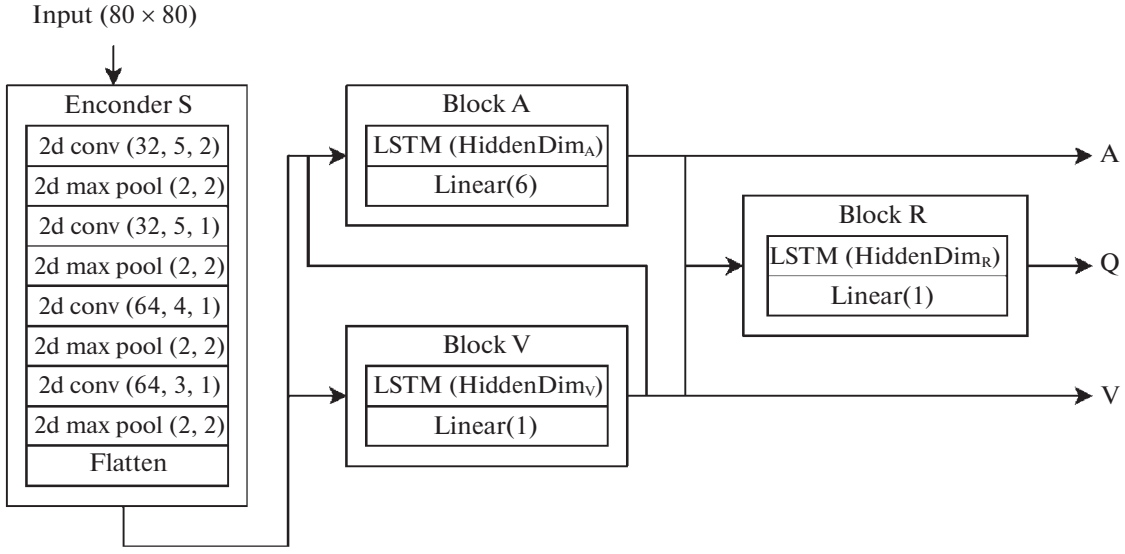

Рис. 2. Архитектура нейросети для предлагаемого метода. Числа в скобках для 2*d conv* означают гиперпараметры сверхточных слоев (число выходных каналов, размерность ядра, отступ, добавляемый по краям), $HiddenDim_V$, $HiddenDim_A$ и $HiddenDim_R$ – размерности LSTM слоев, *Linear* – линейный полносвязный слой, *2d max pool* – слой максимального пулинга с параметрами (размер окна, величина шага), *Flatten* – перевод двумерной матрицы в вектор.

$$L_n^V = \left(\gamma^n * v_{T+1} - v_{T-(n-1)} + \sum_{k=1}^{n} \gamma^{n-k} * r_{T-(k-1)} \right),$$

где $v_{T-(n-1)}$ и v_{T+1} – значения, предсказанные блоком V для состояний $s_{T-(n-1)}$ и s_{T+1} соответственно.

При этом важно, что r_i может быть и отрицательным. Таким образом, состояния, полученные благодаря последовательностям действий, при которых с агентом не происходило событий, сильно влияющих на предсказываемые значения функции V, будут менее интересны агенту. И наоборот, состояния, в которых привычная последовательность действий привела к непредсказуемым результатам, будут иметь большие внутренние награды.

Число параметров дополнительного блока R оказывается достаточно мало, так как набор получаемых им на вход признаков существенно меньше, чем полное признаковое описание состояния, что позволяет аппроксимировать блок R сравнительно небольшой нейросетью.

ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

Тестирование и получение экспериментальных значений качества работы алгоритма происходило путем запуска на решаемой задаче “общей” модели, объединяющей текущий наработанный опыт всех агентов, на разных этапах обучения (при разном числе кадров, просмотренных суммарно каждым из тренировочных агентов). В проводимых экспериментах для запусков использовалось 16 агентов, работающих по схеме A3C. Для получения более точных результатов значения для построения графика усреднялись за несколько повторений эксперимента с теми же настройками.

В качестве базовой модели для сравнения был взят алгоритм A3C без дополнительного блока.

Для A3C выходные размерности LSTM слоев $HiddenDim_A$ и $HiddenDim_V$ равны 849. Для предложенного алгоритма $HiddenDim_A$ и $HiddenDim_V$ также равны 849, а $HiddenDim_R$ равно 128. Число обучаемых параметров для базовой и предложенной модели составило 6.636 и 6.705 M соответственно. Таким образом, используемые модели и архитектуры сетей сопоставимы по сложности. Для настройки использовались следующие гиперпараметры: $Batch\_size = 20$ кадров, $\gamma = 0.99$, алгоритм оптимизации весов Adam со стартовым $Learning\_rate = 0.001$.

Основным параметром, регулирующим величину r_i, является ε. На рис. 3 приведены кривые зависимости количества очков, набираемого тестовым агентом на разных этапах обучения для разных ε при $\gamma = 0.99$ для предлагаемого алгоритма.

Для снижения влияния случайных величин и уточнения полученных кривых для значений ε, показавших наилучшие результаты, было проведено усреднение по 30 экспериментам с одинаковыми настройками (рис. 4).

На рис. 5 приведены кривые для количества очков, набираемых тестовым агентом на разных

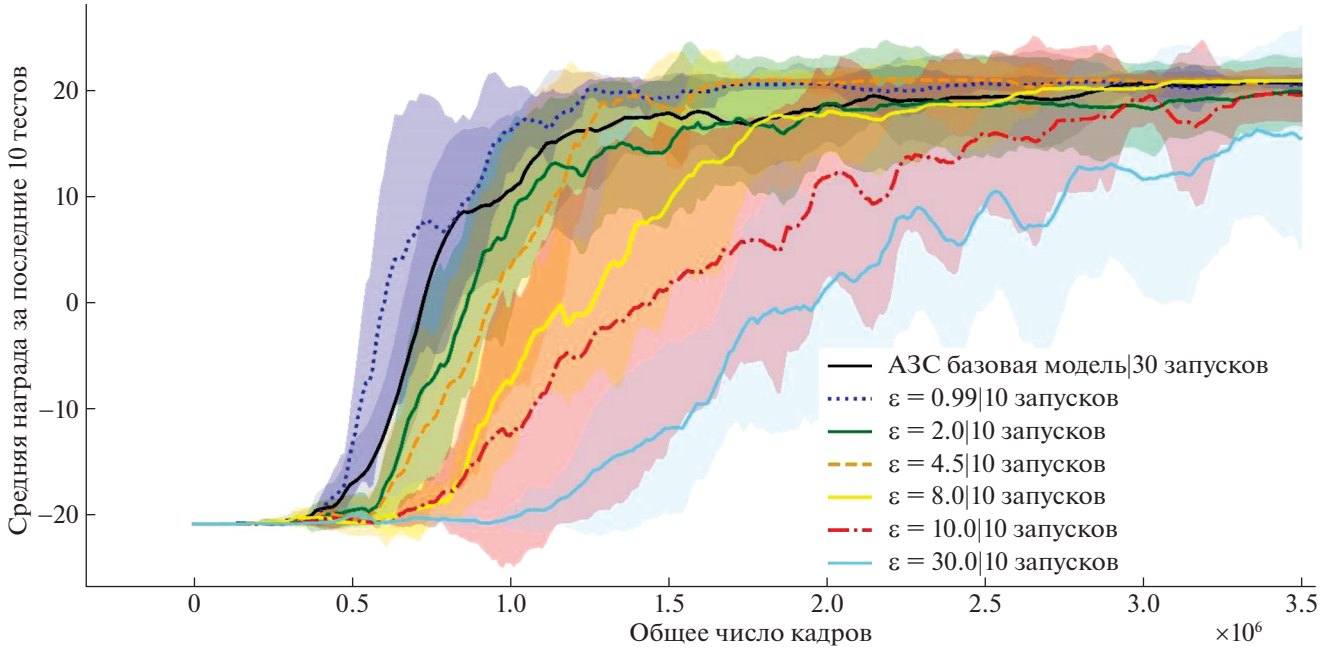

Рис. 3. Зависимость получаемой награды от длительности обучения (количество кадров) с усреднением по 10 запускам с 16 обучаемыми агентами для разных ε.

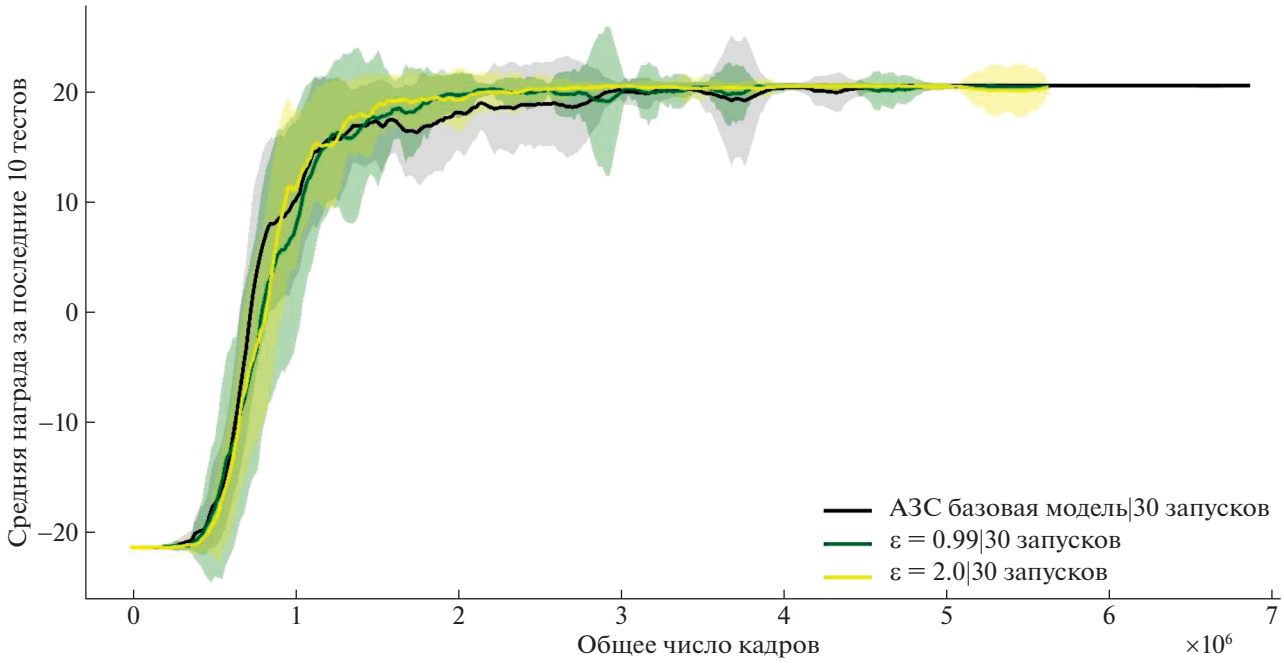

Рис. 4. Зависимость получаемой награды от длительности обучения (количество кадров) с усреднением по 30 запускам с 16 обучаемыми агентами для разных ε.

этапах обучения, представляющие каждый эксперимент отдельно. Можно видеть, что отдельные агенты, обучающиеся при помощи предложенного алгоритма, достигают максимальной награды быстрее каждого из агентов, обучающихся по базовому алгоритму, однако присутствуют агенты, обучающиеся медленнее, что может говорить о снижении стабильности алгоритма с добавлением предлагаемого механизма внутреннего любопытства.

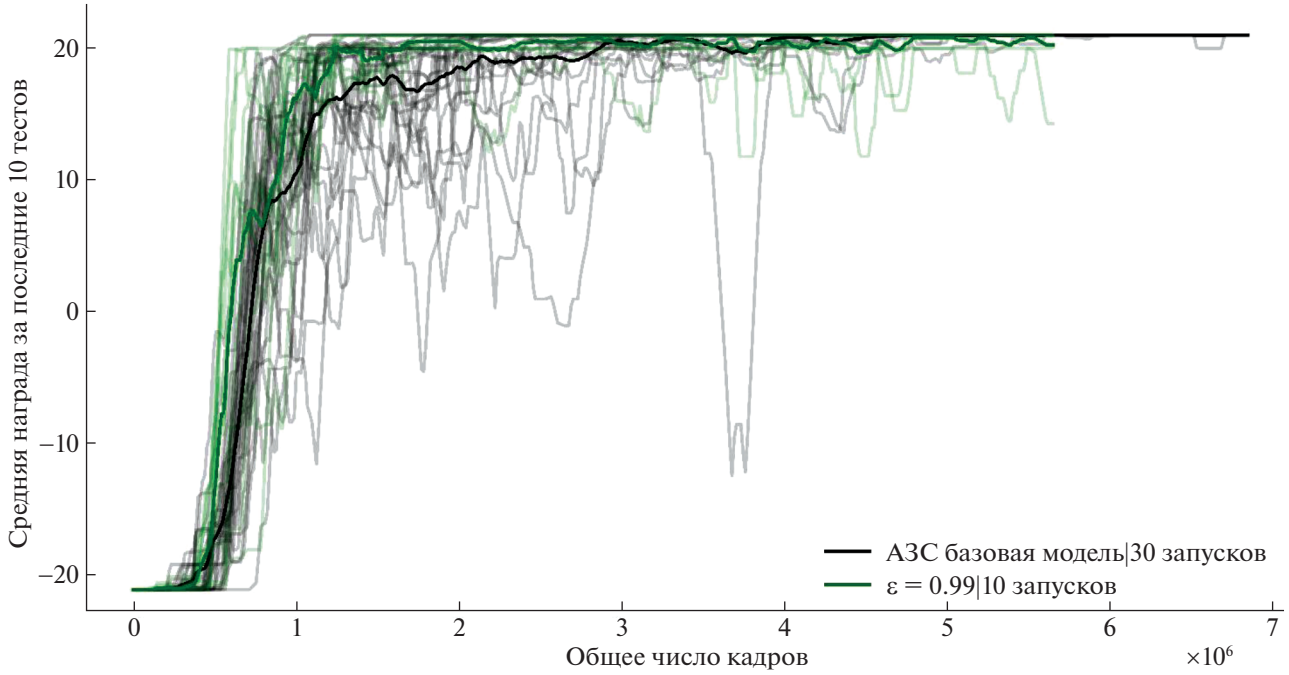

Рис. 5. Зависимость получаемой награды от длительности обучения (количество кадров) с визуализацией отдельных запусков (полупрозрачные линии) и усредненного результата.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Для оценки значимости результатов был использован статистический критерий Манна–Уитни. Для расчета критерия использовались значения количества очков, полученных тестовой моделью в 30 запусках тестового алгоритма и 30 запусках базового алгоритма при одних и тех же количествах кадров, использованных для обучения. Точки для оценки критерия распределялись равномерно от нуля до общего количества кадров, использованных для обучения. Полученные результаты для запусков с ε, равным 0.99, и ε, равным 2, приведены в табл. 1, 2 соответственно.

На рис. 6, 7 приведены графики с подсветкой зон, имеющих подавляющее большинство значимых точек для запусков с ε, равным 0.99, и ε, равным 2.

ЗАКЛЮЧЕНИЕ

Проведенное исследование позволяет заключить, что предложенный механизм внутренней мотивации может ускорять обучение агента – в среднем в каждом из запусков аугментированному блоком R агенту требовалось меньше кадров для достижения стабильно победного результата, и отдельные запуски для агентов, использующих

Таблица 1. Результаты запусков для ε = 0.99

Число кадров	(Общее) от 1871 до 5624427	От 1871 до 1128635	От 1128635 до 2255400	От 2255400 до 3382165	От 3382165 до 4508929	От 4508929 до 5624427
Количество значимых/незначимых точек	92/408	25/75	63/37	4/96	0/100	0/100

Таблица 2. Результаты запусков для ε = 2.0

Число кадров	(Общее) от 1871 до 5624427	От 1871 до 1128635	От 1128635 до 2255400	От 2255400 до 3382165	От 3382165 до 4508929	От 4508929 до 5624427
Количество значимых/незначимых точек	170/330	32/68	92/8	46/54	0/100	0/100

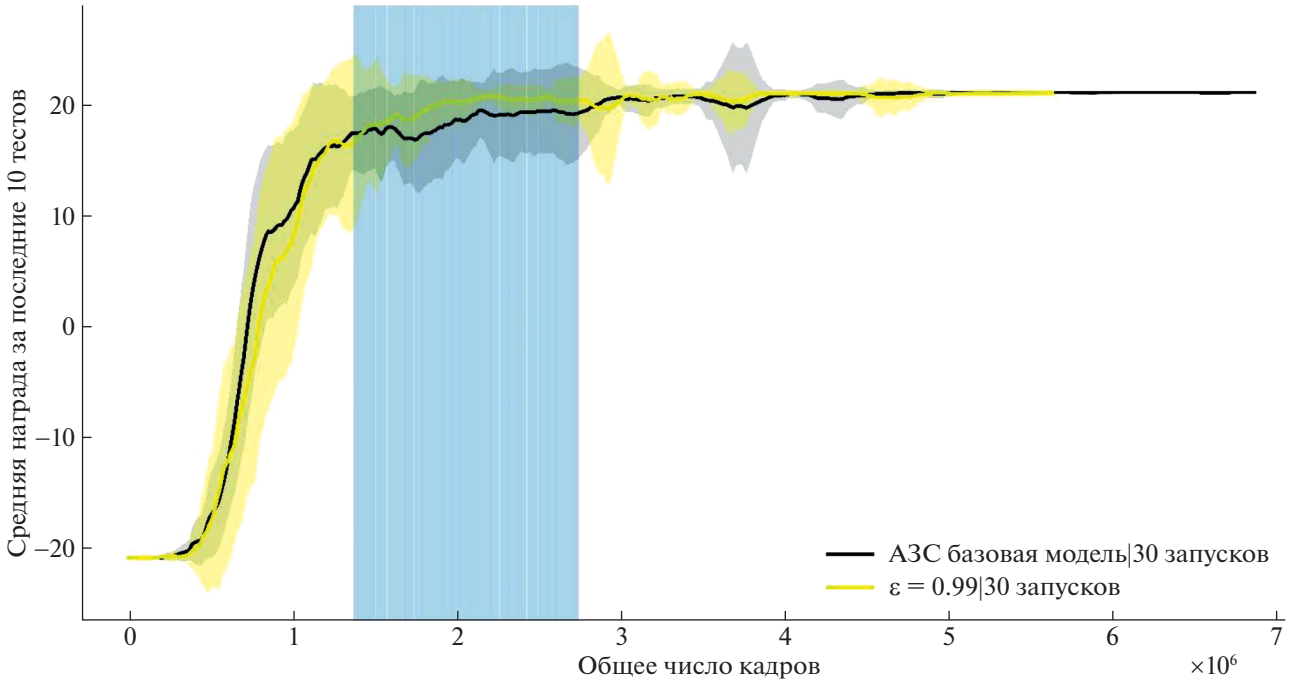

Рис. 6. Зависимость получаемой награды от длительности обучения (количество кадров) с выделенной зоной статистически значимых расхождений для $\varepsilon = 0.99$.

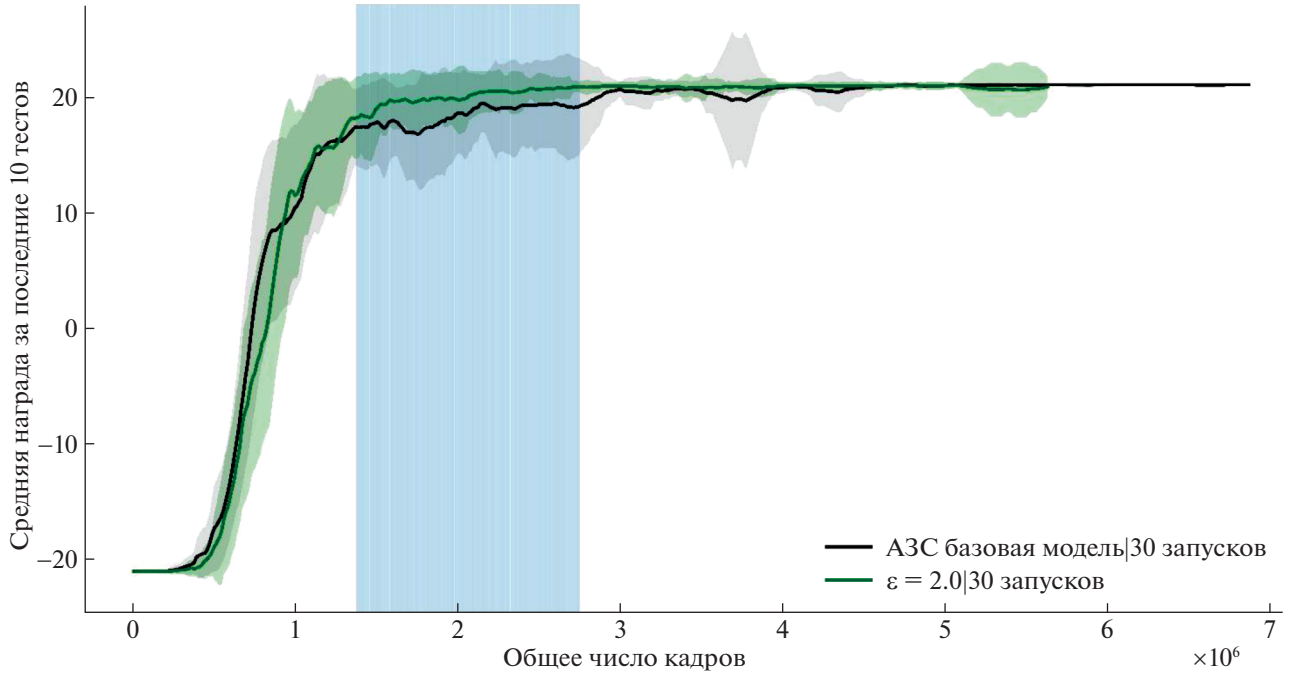

Рис. 7. Зависимость получаемой награды от длительности обучения (количество кадров) с выделенной зоной статистически значимых расхождений для $\varepsilon = 2$.

предлагаемый механизм, показывают наилучшие результаты в сравнении с любыми агентами базового алгоритма. Дальнейшие исследования будут направлены на поиск возможностей для стабилизации предлагаемого метода и его применения в иных средах.

Работа была выполнена с использованием оборудования центра коллективного пользования “Комплекс моделирования и обработки данных исследовательских установок мега-класса” НИЦ “Курчатовский институт”, http://ckp.nrcki.ru/.

СПИСОК ЛИТЕРАТУРЫ

1. *Bellemare M., Srinivasan S., Ostrovski G. et al.* // Adv. Neural Inf. Process. Syst. 2016. V. 29. P. 1471.
2. *Savinov N., Raichuk A., Marinier R. et al.* // Seventh International Conference on Learning Representations. 2019. P. 731.
3. *Burda Y., Edwards H., Storkey A., Klimov O.* // Seventh International Conference on Learning Representations. 2019. P. 950.
4. *Pathak D., Agrawal P., Efros A.A., Darrel T.* // 34th International Conference on Machine Learning, ICML. 2017. V. 70. P. 2778.
5. *Houthooft R., Chen X., Duan Y. et al.* // Adv. Neural Inf. Process. Syst. 2016. V. 29. P. 1109.
6. *Wang Z., Schaul T., Hessel M. et al.* // Proceedings of The 33rd International Conference on Machine Learning. 2016. V. 48. P. 1995.
7. *Schaul T., Quan J., Antonoglou I., Silver D.* // arXiv: 1511.05952. 2015. https://arxiv.org/abs/1511.05952. P 1.
8. *Harutyunyan A., Dabney W., Mesnard T. et al.* // Adv. Neural Inf. Process. Syst. 2019. V. 32. P. 12488.
9. *Bellemare M.G., Naddaf Y., Veness J., Bowling M.* // J. Artif. Intell. Res. 2013. V. 47. P. 253.
10. *Mnih V., Badia A.P., Mirza M. et al.* // Int. Conf. Mach. Learn. 2013. V. 48. P. 1928.