

МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

УДК 004.89

ПРИМЕНЕНИЕ ТРАНСФОРМЕРА ДЛЯ КОДИРОВАНИЯ СОСТОЯНИЙ В ОБУЧЕНИИ С ПОДКРЕПЛЕНИЕМ

© Д. А. Козлов

*Самарский национальный исследовательский университет им. академика С. П. Королёва,
443086, г. Самара, Московское шоссе, 34
E-mail: djoade100@gmail.com*

Исследуется применение архитектуры трансформера для кодирования состояний в алгоритмах обучения с подкреплением. Представлен новый подход, который интегрирует трансформеры с существующими методами, такими как SAC (Soft Actor-Critic), для улучшения их производительности и обобщающей способности. Результаты экспериментальных исследований показывают, что предложенный подход может улучшить обучение в сложных задачах приобретения навыков передвижения в трёхмерном пространстве.

Ключевые слова: обучение с подкреплением, трансформер, SAC.

DOI: 10.15372/AUT20240507

EDN: HZVAJP

Введение. В последние годы глубокое обучение с подкреплением (Deep Reinforcement Learning, DRL) значительно продвинулось, демонстрируя впечатляющие результаты в различных областях: от управления дорожными сигналами [1] до робототехники. Особенно заметным является алгоритм Soft Actor-Critic (SAC), который сочетает в себе гибкость и эффективность в обучении сложных поведенческих стратегий. Алгоритм SAC, первоначально представленный в [2], и последующие разработки [3] показали, что максимизация энтропии может значительно улучшить производительность и устойчивость алгоритмов DRL.

Однако, несмотря на успехи SAC, существует потенциал для дальнейшего улучшения, особенно в контексте обработки последовательностей и пространственных данных. В этом контексте архитектура трансформера, представленная в [4], предлагает значительные перспективы. Трансформеры, изначально разработанные для задач обработки естественного языка, демонстрируют выдающуюся способность к обработке последовательностей и контекстуальному пониманию. Их применение в глубоком обучении с подкреплением, как показано в [5–7], открывает новые возможности для улучшения политик обучения.

В частности, исследования, такие как [8], подчёркивают потенциал трансформеров в улучшении подходов глубокого обучения с подкреплением в сложных визуальных и пространственных задачах. Это подтверждается также исследованиями [9], которые демонстрируют эффективность трансформерных кодировщиков в качестве политик в переменных средах действий. Кроме того, в [10] предложен новый взгляд на трансформеры как на инструменты мета-обучения в контексте глубокого обучения с подкреплением, подчёркивая их универсальность и адаптивность.

В предлагаемой статье архитектура кодировщика трансформера используется для кодирования состояний и действий перед передачей их в алгоритм SAC. Предполагается, что такой подход может не только улучшить производительность SAC в сложных задачах, но и обеспечить более глубокое понимание динамики среды и поведения агента.

Таким образом, целью работы является повышение эффективности методов обучения с подкреплением в задачах приобретения навыков передвижения в трёхмерных пространствах путём интеграции в них кодировщика трансформера для лучшей обработки последовательности состояний.

Основы архитектуры трансформера. Трансформер, представленный в работе [4], революционизировал подходы к обработке естественного языка. Основная особенность трансформера — это механизм внимания, который позволяет модели фокусироваться на различных частях входных данных с учётом контекста. Это преимущество делает трансформер подходящим для кодирования последовательностей, где важно учитывать динамические связи в данных. Основные компоненты трансформера включают в себя блоки внимания, слои нормализации, полносвязные слои и, также важно отметить, отсутствие рекуррентности или свёрток, что, в свою очередь, позволяет эффективно параллельно обрабатывать данные.

Алгоритм SAC в контексте обучения с подкреплением. Soft Actor-Critic [2] является современным алгоритмом обучения с подкреплением, который оптимизирует стохастическую политику в офлайн-режиме. Алгоритм SAC обладает преимуществами в виде устойчивости к выбору гиперпараметров и эффективности обучения. Это достигается за счёт максимизации баланса между ожидаемым вознаграждением и энтропией политики, что приводит к более исследовательскому поведению агента.

Марковские процессы. Марковский процесс — это математическая модель, используемая для описания случайных процессов, которые могут изменять своё состояние в дискретные моменты времени, исходя из определённых вероятностных правил. Основная идея марковских процессов состоит в том, что будущее состояние зависит только от текущего состояния и не зависит от предыдущих состояний или последовательности событий. Это свойство называется свойством Маркова.

Процесс считается марковским, если выполняются два ключевых условия:

1. Свойство Маркова. Будущее состояние зависит только от текущего состояния и не зависит от истории состояний. Математически это можно записать как $P(X_{n+1} | X_1, X_2, \dots, X_n) = P(X_{n+1} | X_n)$, где X_n — текущее состояние;
2. Дискретность времени. Состояния процесса изменяются в дискретные моменты времени, например, на каждом шаге времени n .

Важно отметить, что несмотря на то, что марковские процессы представляют удобную математическую абстракцию, многие реальные задачи требуют учёта более сложных зависимостей и истории, что делает их более сложными для моделирования с использованием классических марковских процессов. Марковский процесс применяется также в постановке задачи обучения с подкреплением. В реальном мире часто встречаются процессы, которые не являются марковскими. Это связано с тем, что в реальности события и состояния могут зависеть не только от текущего состояния, но и от предыдущих состояний или последовательности событий.

Использование трансформеров в задачах обучения с подкреплением представляет собой мощный подход, особенно в контексте реальных и сложных сред, которые не подчиняются строгому марковскому свойству. Трансформеры, изначально разработанные для обработки последовательностей данных, таких как текст и изображения, позволяют моделировать сложные зависимости и учитывать контекст на разных временных и пространственных шкалах.

Одним из ключевых преимуществ трансформеров является их способность учитывать долгосрочные зависимости и историю в данных, что делает их более пригодными для реальных ситуаций, где текущее состояние может зависеть от множества предшествующих событий. Это особенно важно в таких задачах, как автономное вождение, финансовое

моделирование и управление роботами, где решения принимаются на основе сложных и многоуровневых взаимодействий с окружающей средой.

Такие факторы делают трансформеры незаменимым инструментом для решения задач обучения с подкреплением в реальных мирах, где марковское свойство может быть нарушено.

Предложенный подход. В данной работе предлагается подход к интеграции трансформера с алгоритмами SAC. Идея заключается в использовании трансформера для кодировщика последовательностей состояний и действий перед их подачей в сети актора и критика в рамках SAC. Это позволяет не только улучшить представление состояний и действий, но и учитывать контекстуальную информацию, что особенно важно в сложных и динамически меняющихся средах.

Описание архитектуры модели и процесса обучения. Разработанный подход включает в себя кодировщик трансформера, который преобразует входные последовательности состояний и действий в латентное пространство. Далее эти латентные представления используются в сетях актора и критика в качестве входа, как это происходит в оригинальном SAC.

Таким образом, основными компонентами разработанного подхода являются следующие элементы:

1. Кодировщик трансформера:
 - вход: последовательности состояний и действий,
 - выход: латентное представление этих последовательностей.
2. Сеть актора:
 - вход: латентные представления от трансформера,
 - выход: предполагаемое действие в данном состоянии.
3. Сеть критика:
 - вход: латентные представления и действия от сети актора,
 - выход: Q-значение, соответствующее паре состояние–действие.

В ходе исследования были проведены обширные эксперименты в симуляционных средах с целью оценки эффективности предложенного метода. Основное внимание уделялось средам, требующим от агентов сложных навыков передвижения в трёхмерном пространстве.

Модификация буфера воспроизведения для последовательностного сэмплирования. Буфер воспроизведения является ключевым компонентом в алгоритмах обучения с подкреплением, особенно в методах, основанных на временных различиях, таких как SAC. Этот буфер, включающий в себя состояния, действия, вознаграждения, следующие состояния и информацию о завершении эпизода, хранит предыдущие переходы. Основная цель буфера воспроизведения заключается в повторном использовании предыдущего опыта для эффективного обучения, что, в свою очередь, обеспечивает большую стабильность и отсутствие проблем, связанных с коррелированными данными.

В контексте интеграции трансформера с SAC возникла необходимость адаптации буфера воспроизведения для поддержки сэмплирования последовательностей, а не отдельных единиц опыта, так как трансформеры требуют последовательностей данных для адекватного контекстуального анализа и обработки временных зависимостей в данных. Следовательно, было важно модифицировать буфер воспроизведения таким образом, чтобы он мог эффективно хранить и предоставлять последовательности переходов.

Изменена структура хранения данных в буфере воспроизведения посредством организации их в форме последовательностей. Каждая последовательность включает в себя серию переходов, которые следуют друг за другом во времени. Это позволяет модели трансформера использовать контекстуальную информацию из последовательности переходов.

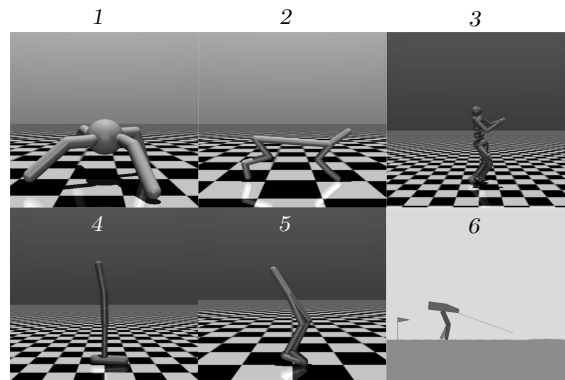


Рис. 1. Среды из пакета Gymnasium: Ant-v4 (1), Half-Cheetah-v4 (2), Humanoid-v4 (3), Hopper-v4 (4), Walker2d-v4 (5). Среда из пакета Box2D BipedalWalker-v3 (6)

Также разработан механизм сэмплирования, который извлекает последовательности данных из буфера для обучения. Данный механизм гарантирует, что сэмплированные последовательности являются непрерывными и сохраняют временную целостность переходов. Для границ эпизодов была введена специальная обработка, чтобы избежать смешения данных из различных эпизодов. Это обеспечивает корректность контекстуальной информации, используемой трансформером.

Данные изменения в буфере воспроизведения являются ключевыми для реализации подхода, интегрирующего архитектуру трансформера с SAC, и обеспечивают необходимую поддержку для обработки последовательностей данных, что является критически важным для достижения улучшенной производительности и обобщающей способности в сложных задачах обучения с подкреплением.

Экспериментальное исследование. Для экспериментов выбраны следующие среды симуляции из пакета Gymnasium [11]: Ant-v4, Humanoid-v4, Half-Cheetah-v4, Hopper-v4 и Walker2d-v4, а также среда из пакета Box2D BipedalWalker-v3. Среды показаны на рис. 1. Эти среды были выбраны из-за их разнообразия и способности имитировать реальные физические взаимодействия и движения. Производительность агентов оценивалась по скорости обучения и эффективности действий. Таким образом, основной метрикой было выбрано значение средней награды в эпизоде для некоторого шага обучения.

Среда Ant-v4 («Муравей»), реализованная в пакете Gymnasium, основана на среде, впервые описанной в [12], и представляет собой трёхмерного робота, состоящего из одного торса (свободного вращательного тела) с четырьмя прикреплёнными к нему ногами, каждая из которых включает в себя две части. Задачей является согласование движения четырёх ног для перемещения вперёд, применяя моменты к восьми шарнирам, соединяющим две части тела каждой ноги и торс (девять частей тела и восемь шарниров).

Среда Humanoid-v4 реализована на основе среды, описанной авторами работы [13]. Трёхмерный бипедальный робот разработан для имитации человека. У него есть торс (брюшная часть) с парой ног и рук. Каждая нога состоит из трёх частей тела, а руки — из двух (представляющих колени и локти соответственно). Цель среды заключается в том, чтобы двигаться вперёд как можно быстрее, не падая.

Среда Half-Cheetah-v4 («Гепард») разработана на основе предыдущих исследований в данной области [14] и представляет собой двухмерного робота, состоящего из девяти частей тела и восьми соединяющих их шарниров (включая две лапы). Задачей является приложение момента к шарнирам, чтобы заставить «гепарда» двигаться вперёд как можно быстрее с положительным вознаграждением, начисляемым на основе пройденного расстоя-

яния вперёд, и отрицательным вознаграждением за движение назад. Туловище и голова «гепарда» неподвижны, и момент можно прикладывать только к шести другим шарнирам на передних и задних бёдрах (соединяющихся с туловищем), голенищах (с бёдрами) и лапах (с голенищами).

Среда Norreg-v4 («Хоппер») была адаптирована из исследований, приведённых в [15]. Основной целью среды является увеличение количества независимых переменных состояния и управления по сравнению с классическими средами управления. «Хоппер» представляет собой двухмерное одноногое существо, состоящее из четырёх основных частей тела: торса в верхней части, бедра в середине, голенища в нижней части и одной ноги, на которой всё тело покоится. Задача состоит в выполнении прыжков для движения вперёд путём применения моментов к трём шарнирам, соединяющим четыре части тела.

Среда Walker2d-v4 основана на среде Norreg-v4 с добавлением ещё одного набора ног, что позволяет роботу двигаться вперёд, а не прыгать. Как и в других средах MuJoCo, целью этой среды является увеличение количества независимых переменных состояния и управления по сравнению с классическими средами управления. Walker2d-v4 — это двухмерная двуногая фигура, состоящая из семи основных частей тела: одного торса в верхней части (с двумя ногами, разделяющимися после торса), двух бёдер в середине под торсом, двух ног в нижней части под бёдрами и двух ног, прикреплённых к ногам, на которых покоится весь корпус. Задачей является движение вперёд посредством применения моментов к шести шарнирам, соединяющим семь частей тела.

Среда BipedalWalker-v3 основана на физическом управлении, использующем физику на базе Box2D и визуализацию на основе PyGame. BipedalWalker-v3 — это одна из таких сред, в которой моделируется двуногий робот. Робот состоит из различных частей тела, и целью среды является обучение робота ходить, управляя моментами и балансируя на двух ногах. Весь процесс основан на физических законах и требует от агента умения управлять движением робота, чтобы достичь заданной цели.

В данной работе используется фреймворк TorchRL [16], который предоставляет мощные инструменты и абстракции для реализации, обучения и оценки алгоритмов обучения с подкреплением на платформе PyTorch [17]. Этот фреймворк демонстрирует высокую гибкость и расширяемость, позволяя эффективно разрабатывать и настраивать различные алгоритмы обучения с подкреплением.

Результаты экспериментального исследования. В рамках экспериментального исследования обнаружено, что внедрение архитектуры трансформера в традиционные методы обучения с подкреплением, в частности в метод SAC, ведёт к улучшению производительности агентов в задачах, связанных со сложными трёхмерными средами. Это улучшение проявлялось в повышенной эффективности решения задач, что отражалось в более высоком уровне наград, получаемых агентами. Конкретно было замечено, что уровень награды был примерно на 5–10 % выше по сравнению с традиционным SAC при аналогичном количестве шагов обучения. Такое повышение производительности свидетельствует о большей выборочной эффективности предложенного метода.

Однако в ходе экспериментов также было выявлено, что не во всех средах наблюдался прирост производительности. В частности, в среде Half-Cheetah-v4 были зафиксированы случаи, когда интеграция трансформеров не приводила к улучшению результатов, а иногда даже отмечалось их ухудшение. Предполагается, что это может быть связано со спецификой настройки алгоритмов обучения с подкреплением для данной среды. Возможно, что сложность и динамика среды Half-Cheetah-v4 требуют более тонкой настройки параметров алгоритма, чтобы полностью раскрыть потенциал подхода. Это предположение подтверждает необходимость дальнейшего изучения и оптимизации параметров алгоритма для разнообразных сред и задач обучения с подкреплением.

Выборочная эффективность является ключевым показателем в обучении с подкреплением, отражающим, насколько эффективно алгоритм использует собранные данные для обучения. Этот термин описывает способность алгоритма достигать высокого уровня производительности, используя ограниченное количество обучающих примеров, или выборок. Важность выборочной эффективности особенно высока в сценариях, где сбор данных является затратным или временно ограниченным процессом. В контексте обучения с подкреплением, где агенты должны исследовать среду и собирать данные через взаимодействие, более высокая выборочная эффективность означает, что агенты могут быстрее обучаться и адаптироваться к среде, минимизируя количество необходимых итераций обучения. Это делает алгоритмы с высокой выборочной эффективностью особенно ценными для практических приложений, где время и ресурсы для обучения ограничены.

В нашем исследовании, результаты которого представлены на рис. 2–4, видно, как SAC, дополненный трансформером в качестве кодировщика, показывает себя в решении различных задач. Замечено, что в большинстве случаев такой подход демонстрирует лучшую или, по крайней мере, сопоставимую производительность по сравнению с базовым алгоритмом SAC. Однако стоит отметить исключения для задач, связанных с симуляциями Half-Cheetah-v4 и Humanoid-v4, где результаты были менее убедительны.

В рамках экспериментов были испытаны три различных конфигурации кодировщика трансформера. Первая конфигурация включала один слой трансформера, вторая — два слоя, а третья использовала три слоя трансформера с двумя головами в механизме многоголового внимания. Эти различные конфигурации позволили изучить, как количество слоёв и голов влияет на способность кодировщика адекватно обрабатывать и интерпретировать входные данные для обучения с подкреплением. Результаты показали, что изменения в архитектуре кодировщика трансформера оказывают заметное влияние на эффективность решения задач в различных симулированных средах.

Разработанный подход к интеграции архитектуры трансформера в методы обучения с подкреплением, такие как SAC, демонстрирует ряд преимуществ, основным из которых является улучшенная способность извлекать и использовать информацию из состояний. Это становится возможным благодаря мощным возможностям трансформеров по обработке и анализу последовательностей данных. Агенты, обученные с применением этого подхода, способны более эффективно реагировать на изменения в среде, так как они лучше понимают контекст и динамику ситуаций. Это особенно важно в сложных или быстро меняющихся сценариях, где способность быстро адаптироваться к новым условиям и принимать информированные решения является ключевым фактором успешности.

Однако вместе с преимуществами подход имеет определённые ограничения. Одним из них является увеличение вычислительной сложности, связанное с применением трансформеров. Это означает, что для реализации и обучения таких моделей требуется больше вычислительных ресурсов по сравнению с традиционными методами. Также критически важной оказывается задача тщательной настройки параметров модели трансформера для достижения оптимальной производительности. Неправильно настроенные параметры могут привести к ухудшению результатов обучения или к ненужному усложнению модели без значительного повышения эффективности.

Применение интегрированной архитектуры трансформера и SAC в робототехнике обещает значительные улучшения в производительности роботов в сложных средах. Эта технология может быть особенно эффективной для задач по навигации, манипуляции объектами и адаптации к непредсказуемым изменениям среды. Благодаря улучшенному кодированию состояний и прогнозированию действий, роботы смогут более эффективно адаптироваться к новым задачам и условиям, что является ключевым аспектом в развитии автономных систем.

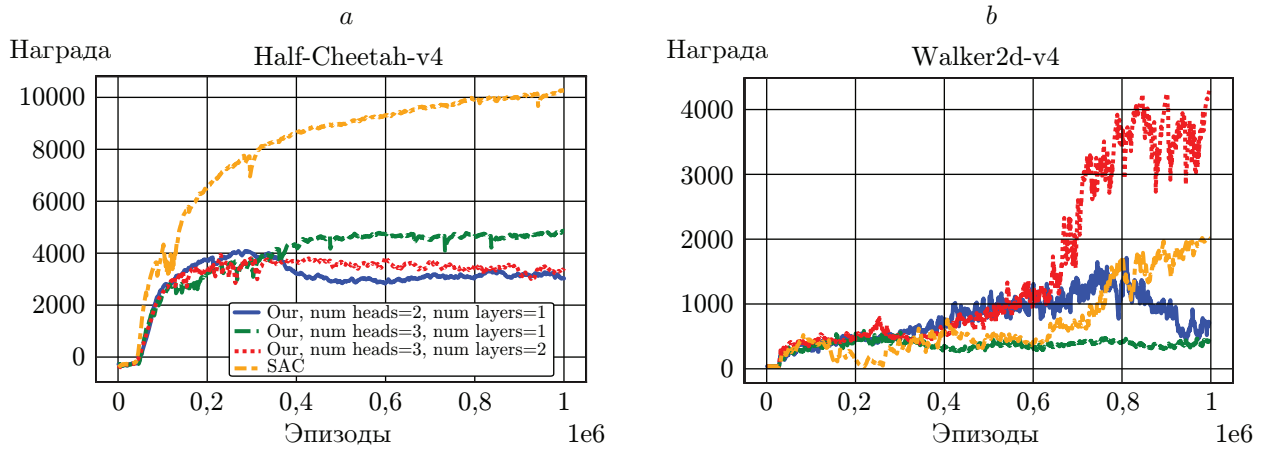


Рис. 2. График зависимости значения средней награды от шага обучения:
 а — среда Half-Cheetah-v4; б — Walker2d-v4

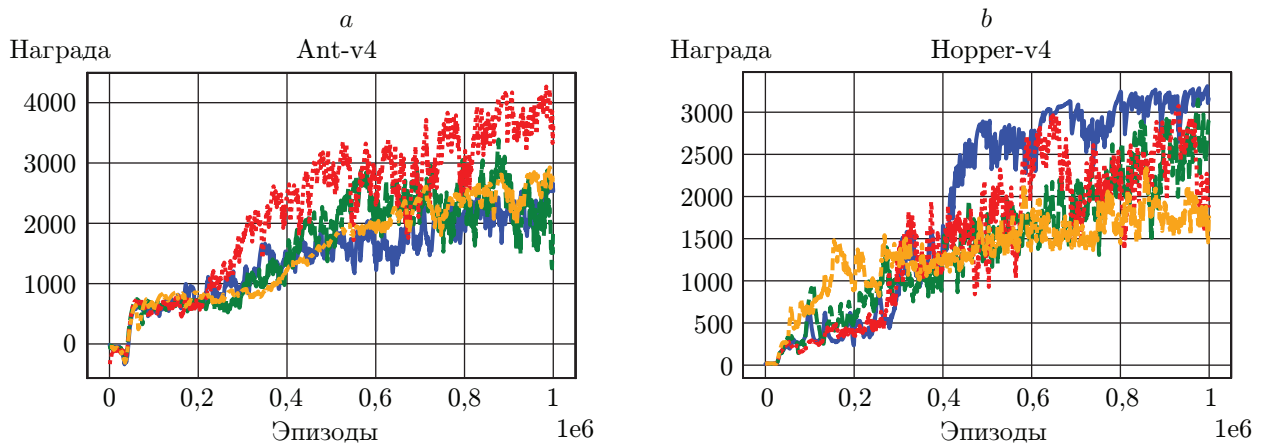


Рис. 3. График зависимости значения средней награды от шага обучения:
 а — среда Ant-v4; б — Hopper-v4

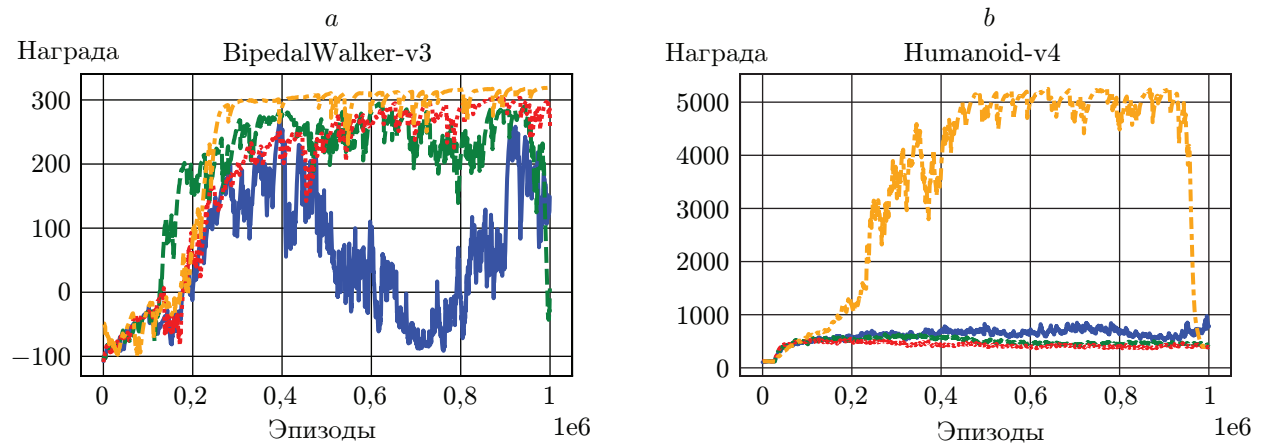


Рис. 4. График зависимости значения средней награды от шага обучения:
 а — среда BipedalWalker-v3; б — Humanoid-v4

В области компьютерных игр и симуляций подход может значительно улучшить поведение искусственного интеллекта, делая его более адаптивным и реалистичным. Использование трансформеров в сочетании с SAC может способствовать разработке сложных и динамичных стратегий ИИ, адаптирующихся к действиям игрока, что повышает уровень погружения и интерактивности в играх.

В сфере автоматизированного управления, например в автономном вождении, метод может улучшить процесс принятия решений в многоуровневых сценариях. Это особенно важно для повышения безопасности и эффективности автономных транспортных средств, где требуются точное восприятие окружающей среды и прогнозирование действий других участников движения.

Для дальнейшего понимания эффективности подхода предполагается проведение экспериментов в разнообразных средах. Это включает не только симулированные условия, но и реальные сценарии, что позволит оценить универсальность и адаптивность метода. Планируется исследовать способы улучшения и оптимизации алгоритма для повышения его производительности и сокращения времени обучения. Значимым аспектом является достижение баланса между сложностью модели и её эффективностью, особенно в условиях реального времени. Особое внимание будет уделено исследованию способности модели адаптироваться к изменениям в среде и обучаться новым задачам. Важным является анализ реакции модели на изменения в окружающей среде и скорости её обучения новым навыкам.

Заключение. В представленной работе исследована возможность интеграции архитектуры трансформера с алгоритмами обучения с подкреплением, а именно с методом Soft Actor-Critic. Продемонстрировано, как использование трансформеров может улучшить обработку последовательностей состояний и действий, что, в свою очередь, приводит к повышению эффективности обучения и обобщающей способности моделей в обучении с подкреплением. Экспериментальные результаты показали, что такой подход способствует улучшению производительности в различных сложных задачах, особенно в таких областях, как робототехника, игровые симуляции и автоматизированное управление.

Показано, что разработанный подход позволяет достигнуть аналогичной или лучшей выборочной эффективности в большинстве исследуемых сценариев. Однако в некоторых случаях наблюдается ухудшение. Таким образом, особенно интересны для разработанного метода являются реальные среды с реальными робототехническими системами, поскольку именно в таких сценариях обнаруживается нарушение марковского свойства среды, что является целевой причиной введения архитектуры кодировщика трансформера в качестве кодировщика состояний и действий для алгоритмов обучения с подкреплением.

Интеграция трансформеров с SAC открывает новые возможности для создания более мощных и адаптивных систем обучения, которые могут эффективно функционировать в динамически изменяющихся средах, а также средах с отложенным вознаграждением.

СПИСОК ЛИТЕРАТУРЫ

1. Агафонов А. А., Юмаганов А. С., Мясников В. В. Адаптивное управление дорожными сигналами на основе нейросетевого прогноза максимального взвешенного потока // Автометрия. 2022. 58, № 5. С. 85–97. DOI: 10.15372/AUT20220510.
2. Haarnoja T., Zhou A., Abbeel P., Levine S. Region growing and region merging image segmentation // Proc. of the 35th Int. Conf. on Machine Learning. Stockholm, Sweden, 10–15 July, 2018. Vol. 80. P. 1861–1870.
3. Haarnoja T., Zhou A., Hartikainen K. et al. Soft Actor-Critic Algorithms and Applications // ArXiv preprint. 2018. DOI: 10.48550/arXiv.1812.05905.

4. **Vaswani A., Shazeer N., Parmar N. et al.** Attention Is All You Need // ArXiv preprint. 2017. DOI: 10.48550/arXiv.1706.03762.
5. **Pamplona J., Madrigal C., Herrera-Ramirez J.** Transformer point net: Cost-efficient classification of on-road objects captured by light ranging sensors on low-resolution conditions // Journ. Comput. Opt. 2022. **46**, N 2. P. 326–334. DOI: 10.18287/2412-6179-CO-1001.
6. **Chen L., Lu K., Rajeswaran A. et al.** Decision Transformer: Reinforcement Learning via Sequence Modeling // ArXiv preprint. 2021. DOI: 10.48550/arXiv.2106.01345.
7. **Upadhyay U., Shah N., Ravikanti S., Medhe M.** Transformer based reinforcement learning for games // ArXiv preprint. 2019. DOI: 10.48550/arXiv.1912.03918.
8. **Meng L., Goodwin M., Yazidi A., Engelstad P.** Deep Reinforcement Learning with Swin Transformers // ArXiv preprint. 2023. DOI: 10.48550/arXiv.2206.15269.
9. **Zwigenberger N.** Transformers as Policies for Variable Action Environments // ArXiv preprint. 2023. DOI: 10.48550/arXiv.2301.03679.
10. **Melo L. C.** Transformers are Meta-Reinforcement Learners // ArXiv preprint. 2022. DOI: 10.48550/arXiv.2206.06614.
11. **Towers M., Kwiatkowski A., Terry J. et al.** Gymnasium: A Standard Interface for Reinforcement Learning Environments // ArXiv preprint. 2024. DOI: 10.48550/arXiv.2407.17032.
12. **Schulman J., Moritz P., Levine S. et al.** High-Dimensional Continuous Control Using Generalized Advantage Estimation // ArXiv preprint. 2015. DOI: 10.48550/arXiv.1506.02438.
13. **Tassa Y., Erez T., Todorov E.** Synthesis and Stabilization of Complex Behaviors through Online Trajectory Optimization // Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS 2012). Vilamoura, Portugal, 7–12 Oct., 2012. P. 4906–4913.
14. **Wawrzyński P.** A Cat-Like Robot Real-Time Learning to Run // Lecture Notes in Computer Science / Ed. by G. Goos // Adaptive and Natural Computing Algorithms (ICANNGA 2009). **5495**. P. 380–390. DOI: 10.1007/978-3-642-04921-7_39.
15. **Durrant-Whyte H., Roy N., Abbeel P.** Robotics: Science and Systems VII. Ch. 10.: Infinite-Horizon Model Predictive Control for Periodic Tasks with Contacts. California: MIT Press, 2012. P. 73–80.
16. **Bou A., Bettini M., Dittert S. et al.** TorchRL: A data-driven decision-making library for PyTorch // ArXiv preprint. 2023. DOI: 10.48550/arXiv.2306.00577.
17. **Ansel J., Yang E., He H. et al.** PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation // Proc. of the 29th ACM Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2024). San Diego, USA, 27 April – 1 May, 2024. Vol. 2. P. 929–947. DOI: 10.1145/3620665.3640366.

Поступила в редакцию 05.07.2024

После доработки 17.07.2024

Принята к публикации 24.07.2024