ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет Санкт-Петербургская школа физико-математических и компьютерных наук

Никулин Александр Павлович

ОЦЕНКА НЕОПРЕДЕЛЕННОСТИ В ОБУЧЕНИИ С ПОДКРЕПЛЕНИЕМ С ПОМОЩЬЮ ДИСТИЛЛЯЦИИ СЛУЧАЙНЫХ СЕТЕЙ

Выпускная квалификационная работа
по направлению подготовки 01.04.02 Прикладная математика и информатика
образовательная программа «Машинное обучение и анализ данных»

Рецензент
главный аналитик данных,
Ассоциация “Искусственный интеллект в промышленности”
_____
О. А. Свидченко

Научный руководитель
д.ф.-м.н., профессор,
Департамент информатики
Санкт-Петербургская школа физико-математических и компьютерных наук НИУ ВШЭ
_____
А. В. Омельченко

Консультант
старший исследователь,
Центр технологий искусственного интеллекта “Тинькофф”
_____
В. В. Куренков

Санкт-Петербург 2023

The Government of the Russian Federation

Federal State Autonomous Institution for Higher Education
National Research University Higher School of Economics

St. Petersburg Branch

St. Petersburg School of Physics, Mathematics and Computer Science

Alexander Nikulin

UNCERTAINTY-BASED OFFLINE REINFORCEMENT LEARNING WITH RANDOM NETWORK DISTILLATION

Master dissertation

Area of studies *01.04.02 «Applied Mathematics and Informatics»*

Master Program “Machine Learning and Data Analysis”

Reviewer
M. Sc., chief data analyst,
Artificial Intelligence in Industry Association
_____
Oleg Svidchenko

Research Supervisor
D.Sc., professor,
Department of Informatics
St. Petersburg School of Physics, Mathematics, and Computer Science
HSE University
_____
Alexander Omelchenko

Saint Petersburg – 2023

Оглавление

1. Введение

1.1. Актуальность работы

В последние годы был достигнут значительный прогресс в применении обучения с подкреплением (Reinforcement Learning) для решения таких сложных и объемных задач, как Atari [1], Go [2], Dota 2 [3], и даже Minecraft [4]. Многие из этих проблем еще совсем недавно считались нерешаемыми в перспективе даже далекого будущего. И тем не менее, благодаря активному развитию обучения с подкреплением как области и интеграции достижений глубокого обучения, появляется все больше алгоритмов способных обучаться решать сложные проблемы с нуля - без апроирного знания о проблеме.

Несмотря на перечисленные успехи, в отличие от других областей машинного обучения, таких как обучение с учителем и без учителя, применимость обучения с подкреплением для решения проблем вне контролируемых симуляций и игр остается сильно ограниченной - по ряду причин. Во-первых, даже лучшие алгоритмы требуют огромного количества данных для обучения, а соответственно и взаимодействий со средой. В отличие от симулятора, скорость генерации данных в реальном времени, например для робота или машины-автопилота, физически ограниченна и плохо масштабируется. Во-вторых, мы не можем заранее предсказать паттерны поведения агента, которые будут проявлятся в течении обучения, а значит не можем гарантировать безопасность - для пользователей, оборудования, окружающей среды. Есть и другие причины, однако их перечисление выходит за рамки данной работы.

Подобный контраст - быстро растущая популярность области без сопоставимого прогресса в применимости к реальным задачам - не остался без внимания исследователей. Поэтому, начиная с 2018 года начала набирать популярность подобласть обучения с подкреплением, где предлагалось полностью ограничить взаимодействие агента со средой, тем самым дав больше контроля исследователям и инженерам над процессом сбора данных, их качеством и безопасностью процесса [5]. Об-

ласть называется *офлайн-обучение с подкреплением*, отражая тем самым факт того, что агент обучается в изоляции на фиксированном и заранее собранном датасете с размеченной наградой, то есть - офлайн.

Несмотря на первичный успех в практическом применении и развитии офлайн-обучения с подкреплением [6,7,8,9], ограничения во взаимодействиях со средой приводят к новым фундаментальным проблемам. Одна из таких проблем - невозможность оценить будущую суммарную награду (также return) для действий, которые отсутствуют в датасете. Поскольку для обучения используются нейронные сети, которые по определению способны выдавать *какое-то* предсказание на любой вход, для действий вне датасета предсказания будет совершенно случайными, что ломает большинство традиционных алгоритмов [5]. Поэтому алгоритмы для офлайн-обучения с подкреплением так или иначе пытаются быть консервативными - то есть избегать действий вне датасета. Например, оставаясь близко к поведенческой политике, которой были собраны данные [10, 11, 12, 13], вводя штраф занижающий награду [14], либо полностью избегая оценки действий вне датасета [15].

Методы основанные на оценке неопределенности с помощью ансамблей моделей показали себя как наиболее успешный подход для офлайн-обучения с подкреплением. Алгоритмы основанные на ансамблях, такие как SAC-N, EDAC [16], и MSG [17] на данный момент показывают state-of-the-art результаты, превосходя остальные подходы с большим отрывом. К сожалению, для хороших результатов требуются ансамбли больших размеров, что ведет к значительным затратам на обучение по памяти и вычислительным ресурсам [18]. В недавней работе исследователи смогли уменьшить размер ансамбля до десятков [19]. Однако, с учетом тенденций к увеличению размера моделей в обучении с подкреплением и в глубоком обучении в целом, обучение даже десяти моделей размером в 80 миллионов параметров каждая остается тяжелой задачей. Более того, в [17] авторы показали, что методы для эффективного обучения ансамблей в глубоком обучении не переносятся в обучение с подкреплением и могут даже ухудшать итоговый результат.

Таким образом, необходимы новые исследования по эффективной

оценке неопределенности в офлайн-обучении с подкреплением, с целью уменьшить размер ансамбля как можно сильнее, либо найти альтернативные подходы.

1.2. Основные результаты

В данной работе исследуется альтернативный подход для оценки неопределенности в офлайн-обучении с подкреплением с помощью дистилляции случайных сетей [20] (далее по тексту как RND).

RND является привлекательной и легковесной альтернативой для оцени неопределенности (epistemic uncertainty). Однако предыдущие попытки применения RND в данной области оказались неудачными [21], поэтому значительная часть данной работы посвящена анализу и воспроизведению предыдущих результатов (Глава 4). Во время анализа были выявлены основные причины неудовлетворительных результатов, а также возможные решения, которые были далее проверены.

На основе анализа, а также предложенных решений, в Главе 5 выводится новый эффективный алгоритм SAC-RND, позволяющий избежать использования ансамблей без потери в качестве. Предложенный алгоритм был проверен на стандартном для офлайн-обучения с подкреплением наборе данных D4RL [22] и показал результаты сравнимые с ансамблевыми методами, при этом на порядок превосходя результаты методов без ансамблей. В заключение, были проведены дополнительные эксперименты, более подробно обьясняющие успех предложенного алгоритма (Глава 6). Научная публикация[1] написанная по итогам данного диплома была принята на конференцию ICML[2] 2023.

[1]Публикация доступна в открытом доступе на портале arXiv по ссылке `https://arxiv.org/abs/2301.13616`. Исходный код доступен на Github по ссылке `https://github.com/tinkoff-ai/sac-rnd`.

[2]Одна из лучших конференций в области машинного обучения. Процент принятых работ составляет 25%. Импакт-фактор ~ 30.

2. Теоретические основы

В данной главе будут разобраны теоретические основы, необходимые для понимания предметной области, а также используемых методов. В параграфе 2.1 дано краткое введение в область обучения с подкреплением. В параграфе 2.2 будет введена подобласть офлайн-обучения с подкреплением. В ней обсуждается основная мотивация подобной постановки задачи обучения, а также возникающие проблемы и последствия, уникальные для данной подобласти. Далее, в параграфе 2.3 разбирается альтернативный взгляд, обьединяющий обе области через оценку неопределенности и ее использование для исследования среды. В оставшихся параграфах вводятся технические термины и подходы, на которые опирается предложенный в данной работе метод.

2.1. Введение в обучение с подкреплением

Обучение с подкреплением является одним из трех основных направлений машинного обучения помимо обучения с учителем и обучения без учителя. В отличие от обучения с учителем, где решается задача предсказания заранее размеченных меток разного вида, и обучения без учителя, где основной задачей является поиск скрытой структуры в большом массиве неразмеченных данных, обучение с подкреплением имеет дело с *задачами последовательного принятия решений*.

В своей основе обучение с подкреплением предполагает взаимодействие между агентом и окружающей средой (см. рис. 1), в ходе которого агент предпринимает какие-то действия и получает обратную связь в виде вознаграждения (или наказания). Со временем агент должен обучиться используя обратную связь выбирать действия, которые ведут к максимизации суммарной награды. В качестве примеры среды можно привести любую игру, такую как Go, шахматы или Minecraft. Одним из ключевых преимуществ обучения с подкреплением является универсальность: одни и те же фундаментальные принципы могут быть применены к очень широкому спектру практических применений, начиная от простых игр, таких как крестики-нолики или шахматы, до

сложных задач, таких как управление беспилотными автомобилями и роботами, изобретение новых алгоритмов, даже удержание плазмы в токамаке. Несмотря на все достижения, в области обучения с подкреплением существует множество интересных научных вопросов, еще нерешенных задач. Например, как оптимально соблюсти баланс между исследованием нового и эксплуатацией уже известного, как работать с разреженной (очень редкой) наградой, как повышать эффективность обучения относительно количества шагов в среде, как переносить опыт в одной среды на другую и так далее. Самым главным вызовом остается применимость в реальных задачах т.к. сейчас применение обучения с подкреплением в основном ограничено симуляциями и играми.

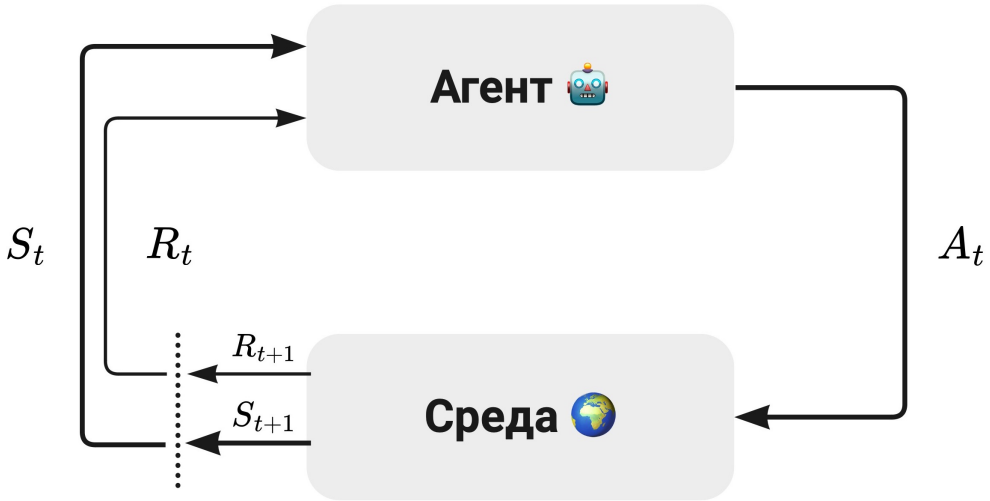


Рис. 1: Схематическое изображение цикла взаимодействия агента и среды в онлайн-обучении с подкреплением. Агент совершает конкретное действие A_t из пространства всех возможных, после чего среда переходит из состояния S_t в S_{t+1} в соответствии с динамикой $P(S_{t+1}|A_t, S_t)$. Помимо следующего состояния, среда возвращает также награду R_t, в качестве обратной связи. Основная цель агента - максимизировать суммарную награду.

Теперь опишем проблему обучения с подкреплением более формально. Начнем с того, что обучение с подкреплением опирается на **гипотезу вознаграждения** [23], а именно, что любая сколь угодно сложная цель может быть формализована как результат максимизации суммарной награды, ведущей к ее выполнению. Несмотря на то, что к данной гипотезе существуют контрпримеры и теоретические возражения [24],

алгоритмы и теория обучения с подкреплением предполагают, что гипотеза верна. Непосредственно теория обучения с подкреплением строится на моделировании проблемы с помощью **марковского процесса принятия решений** (MDP). Математически подобный процесс определяется как кортеж $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma\}$, где:

$\mathcal{S}$ — конечное множество всех возможных состояний в среде. Например, если состояние среды описывает вещественным вектором из 3 составляющих, то пространством будет $\mathcal{S} \subset \mathbb{R}^3$.

$\mathcal{A}$ — конечное множество всех доступных действий в среде. Аналогично состояниям, может быть либо вещественным вектором, либо дискретным набором, либо чем-то более сложным и составным.

$\mathcal{P}$ — функция перехода или динамика $\mathcal{P}(S_{t+1}|S_t, A_t)$, определяющая то, как среда переходит из одного состояния в другое при условии совершенного действия. Может быть стохастической или детерменированной.

$\mathcal{R}$ — функция вознаграждения $\mathcal{R}(S_t, A_t)$, получаемого при совершении действия A_t в состоянии S_t. Фактически определяет цель, которую необходимо достичь.

γ — коэффициент дисконтирования < 1.0. Неформально определяет представление о том, что награда в отдаленном будущем должна иметь меньшую важность, чем награда в ближайшем будущем. Имеет также и теоретическое значение, т.к. при бесконечном горизонте планирования сумма ряда дисконтированных наград сходится, а не растет бесконечно.

Целью обучения с подкреплением в случае бесконечного горизонта является обучение политики (а также агента или стратегии) $\pi(s, a)$ максимизирующей суммарную дисконтированную награду $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$. Полезно также определить функции полезности. Пусть

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

Тогда, функция полезности состояния s при условии, что дальше будет действовать политика π определяется как

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t|s_t = s]$$

Аналогично можно определить функцию полезности состояния при условии действия (которое не обязательно выбирала текущая политика)

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[G_t|s_t = s, a_t = a]$$

Как можно заметить, эти функции связаны простым свойством:

$$V^{\pi}(s) = \mathbb{E}_{a,\pi}[Q^{\pi}(s, a)]$$

В основе всех современных алгоритмов лежат уравнения Беллмана, которые позволяют выразить функции полезности рекуррентно:

$$V^{\pi}(s_t) = \mathbb{E}_{a_t,s_{t+1}}[r(s_t, a_t) + \gamma V^{\pi}(s_{t+1})]$$

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{s_{t+1}}[r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1}} Q^{\pi}(s_{t+1}, a_{t+1})]$$

Благодаря этому становится возможно применять алгоритмы динамического программирования для решения проблем обучения с подкреплением. Можно определить также функции полезности для оптимальной политики и соответствующие им уравнения оптимальности Беллмана. Более подробно теоретические основы описаны в [25].

Стоит также обратить внимание на свойство, которое дало название процессу принятия решений, а именно марковское свойство, т.к. оно накладывает фундаментальные ограничения на то, как именно структурирован процесс. Неформально, марковское свойство соблюдает тогда, когда наше знание о будущем зависит только от текущего состояния, а не от всех предыдущих. Более формально, $P(R_{t+1}, S_{t+1}|S_t, A_t) = P(S_{t+1}|S_t, S_{<t}, A_t, A_{<t})$. В таком случае можно показать, что зная только текущее состояние и действие возможно предсказать все будущие состояния, а также ожидаемую награду [25].

2.2. Офлайн-обучение с подкреплением

Офлайн-обучение с подкреплением, как уже было кратко описано в параграфе 1.2, отличается от онлайна тем, что в нем ограничивается взаимодействие агента со средой, а все обучение происходит на фиксированном и заранее собранном датасете с размеченной наградой. Обычно не делается никаких предположений о том, как был собран датасет, будь то одной политикой или смесью нескольких, насколько хорошо датасет покрывает пространство состояний и т.д., хотя это и может сильно влиять на итоговый результат [26]. Основное отличие от обучения с учителем в том, что наивное обучение предсказанию действий как в датасете не ведет к поведеную максимизирующему награду, в то время как агенты обученные с помощью офлайн-обучения с подкреплением способны на порядок превзойти лучшие результаты в датасете.

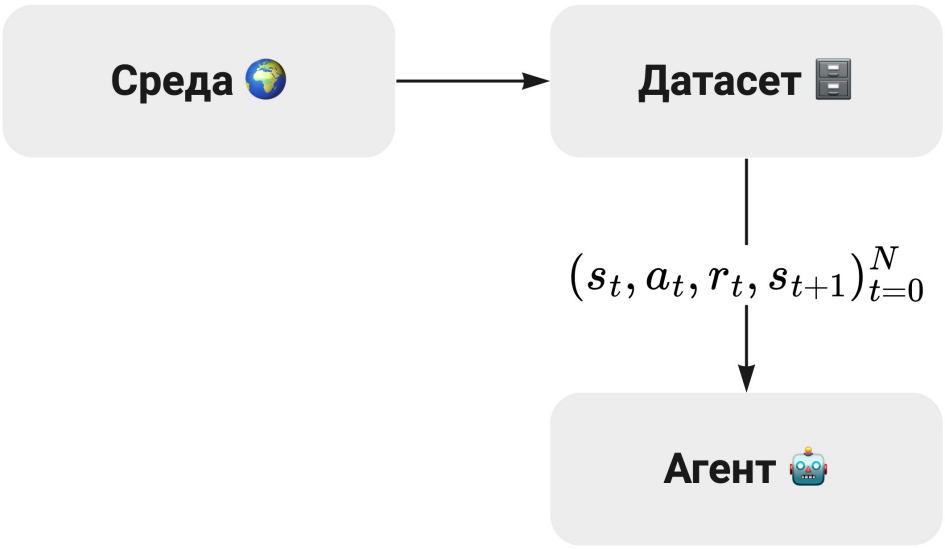


Рис. 2: Схематическое изображение цикла офлайн-обучения с подкреплением. В отличие от Рис. 1 агент не взаимодействует со средой напрямую, а только с заранее собранными данными. Данные по структуре в точности копируют взаимодействие агента со средой, а также размечены наградой.

Офлайн-обучение с подкреплением часто используется в случаях, когда сбор данных в онлайне непрактичен или запредельно дорог, например, при использовании роботов в опасных средах, при моделировании редких событий и т.д. (больше примеров можно найти в парагра-

фе 1.2). Такой подход также открывает путь к использованию больших данных, доступных для многих задач, т.к. онлайн методы не умеют использовать их эффективно. Таким образом, офлайн-обучение с подкреплением также можно рассматривать как способ предобучения и добавления априорных знаний, которые можно использовать для решения новых задач более эффективно и за меньшее количество онлайн взаимодействий.

Для данной работы также важно обсудить некоторые проблемы, возникающие при переходе к офлайн-обучению. Одной из наиболее важных считается невозможность оценить ценность действия произвольного действия. Чтобы показать это более наглядно, стоит вспомнить как оценивается уравнение Беллмана для Q-функции во время обучения:

$$[Q(s,a) - (r + \gamma Q(s',a'))]^2$$

где $a' \sim \pi(\cdot|s')$ сэмплируется из обучаемой политики, а s, a, r, s' сэмплируются из датасета. В течении обучения, благодаря такой функции потерь, наш критик старается все больше соответствовать уравнению Беллмана для произвольных s, a. К сожалению, в офлайн-обучении даже сколь угодно большой датасет все равно покроет лишь малую часть возможных сочетаний состояний и действий. Поэтому во время обучения критик будет удовлетворять уравнению Беллмана только для существующих в датасете пар, в то время как для сочетаний вне датасета, например для известных состояний, но новых действий, предсказания будут неверными. Они могут быть как завышенными, так и заниженными. Поскольку задача актора выбирать действия $max_{a \sim \pi \cdot|s)} Q(s,a)$, то он легко может найти невалидное действие для некоторого состояния s, предсказание награды для которого будет существенно выше, чем для действий из датасета (т.к. критик это нейросеть, способная выдать некоторое предсказание для любого входа), тем самым *"взломав"* критика, после чего метод разойдется. Именно поэтому большинство методов офлайн-обучения с подкреплениям [14, 16, 17, 21] стремятся к консерватизму, а именно к обучению критика таким образом, чтобы предсказания суммарной награды для действий вне датасета было по

определению ниже, чем для действий внутри датасета. Далее будет разобран способ как можно добиться подобного консерватизма через интерпретацию онлайн-обучения как исследования, а офлайн-обучения как анти-исследования в рамках одного алгоритма.

2.3. Офлайн-обучение с подкреплением как анти-исследование

Исследование среды крайне важно так как помогает агенту лучше понимать задачу и быстрее находить решение. Например в задачах где награда дается крайне редко, вероятность того, что необученный агент случайным образом наткнется на нее крайне мала. Поэтому важно давать агенту дополнительную награду за систематеческое исследование среды, что повышает его шансы получить основную награду в процессе исследования. Такие награды называются бонусами за исследование среды (exploration bonus).

Часто бонусы за исследование связаны со степенью предсказанной неопределенности, т.к. неопределенность позволяет определить степень новизны текущего региона пространства для агента. В регионах где агент бывал часто неопределенность будет маленькой, в то время как в новых состояниях неопределенность будет высокой, что поощряет агента исследовать данную область в надежде, что он может получить более высокую основную награду. Как можно заметить, основное свойство бонусов за исследование в том, что со временем они стремятся к нулю, тем самым асимптотически не меняя конечную цель заданную основной наградой.

Поскольку в офлайн-обучении нам наоборот необходимо поощрять консервативность, то есть штрафовать агента за выбор действий ведущих к большей неопределенности, мы можем переиспользовать бонус за исследование, однако вместо добавления к основной награде, он будет вычитаться, тем самым занижая основную награду в новых регионах пространства. Такие бонусы называются *бонусами за анти-исследование*. Подобное использование бонусов фактически **стирает**

алгоритмическую разницу между офлайн и онлайн обучением, позволяя использовать алгоритмы из онлайн обучения в офлайне. Однако, в отличие от онлайн обучения, вычитание бонуса из награды субоптимально, т.к. бонус по определению близок к нулю для данных в датасете (их неопределенность мала). Поэтому, более эффективным является использование бонуса там где могут появляться действия из вне, а именно в правой части уравнения Беллмана для Q-функции:

$$r + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s')}[Q(s', a') - b(s', a')] \tag{1}$$

Можно показать, что теоретически это эквивалентно вычитанию из бонуса из награды [21], однако, как было обьяснено выше, данная формулировка больше подходит для офлайн-обучения.

Примером алгоритмов использующих анти-исследование могут быть SAC & EDAC [16] и MSG [17] основанные на больших ансамблия критиков. В качестве бонуса они вычисляют дисперсию предсказаний ансамбля как меру неопределенности. Перечисленные методы достигают результатов существенно лучших, чем их аналоги без использования ансамблей, однако обучение ансамблей требует больших вычислительных затрат. Основной целью данной работы является применение дистилляции случайных сетей в качестве метода для оценки неопределенности в офлайн-обучении без использования ансамблей.

2.4. Дистилляция случайных сетей

Метод дистилляции случайных сетей (RND) был впервые предложен именно в онлайн-обучении с подкреплением [20] как простой и эффективный способ улучшить или ускорить процесс исследования среды. Основной целью метода является оценка новизны некоторого состояния, чтобы определить степень знакомства агента с данным регионом пространства состояний. Если новизна высокая, то вероятно стоит более тщательно изучить окрестности, чтобы понять ведет ли такой путь к более высокой основной награде.

RND это крайне простой и вычислительно легкий метод, тем не ме-

нее, успешно преодолевший проблемы предыдущих подходов (например проблему шумного телевизора [20]), а также показавший state-of-the-art результаты на играх с крайне разреженной наградой, таких как Montezuma's Revenge от Atari. Даже сейчас, спустя пять лет с момента публикации, RND является сильным бейзлайном для исследования среды, который не требует долгой настройки и может работать даже в стохастических средах, в отличие от некоторых более новых методов [27].

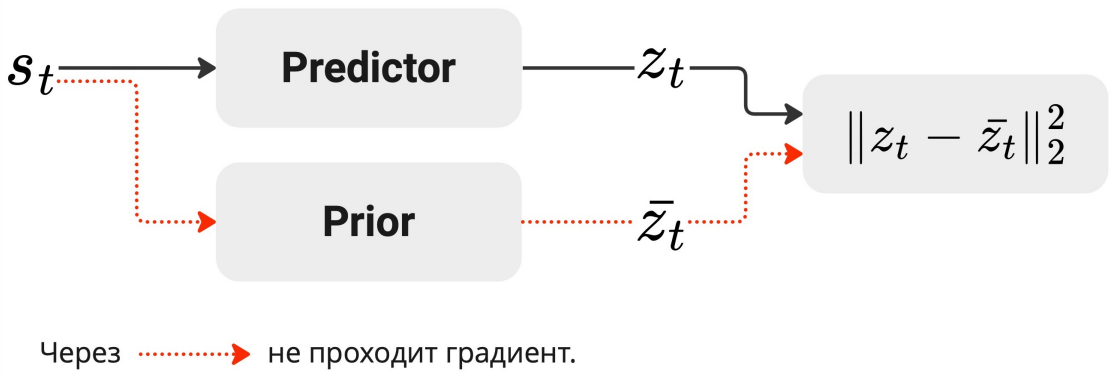


Рис. 3: Схематическое изображение архитектуры RND, состоящей из двух сетей *predictor* и *prior*. Обе сети являются отображением состояний s_t в многомерные эмбеддинги z_t. Функцией потерь является MSE между эмбеддингами, причем градиент через *prior* не пропускается.

Рассмотрим теперь устройство RND. Архитектура метода состоит из двух нейронных сетей (см. Рис. 3):

Prior $f_{\bar{\psi}}(s)$**.** Случайно инициализированная, но при этом зафиксированная, то есть не меняющаяся во время обучения. В некотором смысле задает случайное априорное распределение.

Predictor $f_{\psi}(s)$**.** Также случайно инициализированная, но при этом обучаемая сеть. Основной задачей данной сети является предсказание выхода $f_{\bar{\psi}}(s)$ на известных данных, то есть минимизация данной MSE функции потерь: $\|f_{\psi}(s) - f_{\bar{\psi}}(s)\|_2^2$. Стоит заметить, что ошибка на неизвестных данных (тех, что не было в обучаемой выборке) в таком случае не минизируется т.к. *predictor* не может полностью дистиллировать в себя *prior*. Это важно в дальнейшем.

Обе сети задают отображение из пространства состояний среды в пространство $\mathbb{R}^K$ многомерных эмбеддингов, а градиент через *prior* сеть не пропускается (т.к. она не должна изменяться, а также чтобы не давать дополнительной информации об априорном распределении эмбеддингов). Обычно обе сети имеют одинаковую архитектуру, но как мы покажем далее в данной работе, иногда выгодно, чтобы они были разными.

На основе этой архитектуры, интерпретация новизны становится тривиальной. При достаточно разнообразном *prior*, *predictor* учится предсказывать эмбеддинги *prior* на данных похожих на обучающую выборку (in distribution, ID), в то время как для данных вне датасета (out of distribution, OOD) *predictor* будет сильно ошибаться т.к. ему неизвестны *prior* эмбеддинги на них. Оценкой новизны или бонусом для исследования в таком случае може быть ошибка предсказания, то есть $\|f_\psi(s) - f_{\bar{\psi}}(s)\|_2^2$. В последующих работах [28], было показано, что дистилляция случайных сетей может быть конкурентоспособной альтернативой ансамблям для оценки эпистемологический неопределенности, то есть неопределенности, которая уменьшается с увеличением данных и идет от неуверенности метода в предсказании, а не от шума в данных (здесь можно заметить аналогию с оценкой новизны).

Заметим, что на практике выбор одинаковой архитектуры для обеих сетей, а также оценка новизны только на основе состояний - наиболее распространненый, однако произвольный. Более того, в офлайн-обучении с подкреплением нам интересна оценка новизны *действия при условии текущего состояния*, поэтому в данной работе RND зависит от состояния и действия, а именно $f_\psi(s, a)$.

2.5. Мультипликативные взаимодействия

Часто в глубоком обучении бывает необходимо агрегировать информацию с нескольких разных модальностей, например как в RND, где на вход подаются состояния s и действия a. Наиболее частым способом агрегации является простая конкатенация признаков всех модальностей.

Несмотря на простоту, данный способ может оказаться субоптимальным [29]. Бывают и другие способы, так в [30] показали, что на практике мультипликативные взаимодействия предоставляют более подходящие апроирные механизмы (inductive biases) для агрегации разных модальностей или обусловливания на новые модальности.

Далее мы более подробно опишем способы агрегации (помимо конкатенации), которые используются в данной работе: *gating* [31], *bilinear* [30], *feature-wise linear modulation* (FiLM) [32].

Gating. Простое обусловливание с помощью двух полносвязных слоев и поэлементным умножением полученных признаков с некоторой нелинейностью. Легко интерпретируется из-за сигмоиды, т.к. после домножения признаков a на признаки s, мы адаптивно выбираем какую часть информации в a надо занулить, а какую наоборот оставить.

$$g(a, s) = tanh(W_1 a + b_1) \odot \sigma(W_2 s + b_2)$$

Bilinear. Билинейный слой в наиболее общей форме был предложен в [30]. Задает билинейное отображение.

$$g(a, s) = s^T \mathbb{W} a + s^T \mathbb{U} + \mathbb{V} a + b$$

где $\mathbb{W}$ это трехмерная матрица, $\mathbb{U}$, $\mathbb{V}$ двумерные, а b это вектор. В данной работе также используется упрощенная имплементация подобного слоя из библиотеки PyTorch, которая не обучает $\mathbb{U}$, $\mathbb{V}$.

Feature-wise Linear Modulation (FiLM). Частный случай билинейного отображения с низкоранговыми матрицами весов [32].

$$g(h, s) = \gamma(s) \odot h + \beta(s)$$

Обычно FiLM применятся в внутренним репрезентациям h между слоями, которые зависят от a, в то время как обусловливание происходит с помощью s.

3. Обзор предметной области

В данной главе производится обзор предметной области с целью более подробно изложить уже существующие подходы, релеватные данной работе, а также их недостатки. В параграфе 3.1 разбираются подходы к оценке неопределенности, а также мотивируется их полезность. Проводится контраст между существующими методами и подходом предолженным в данной работе. В параграфе 3.2 подробно обозреваются подходы к эффективному обучению ансамблей и их ограничения при применениях в офлайн-обучении с подкреплением.

3.1. Оценка неопределенности в офлайн-обучении с подкреплением

Оценка неопределенности это крайне популярная техника в обучении с подкреплением и используется для большого многообразия применений, таких как исследование среды, планирование, робастность, безопасность и многих других. В офлайн-обучении с подкреплением оценка неопределенности пригодится для моделирования *эпистемологичекой неопределенности* [33]. В отличии от содержащегося в данных шума и происткающей из него неопределенности, эпистемологическая неопределенность говорит об *уверености модели в своем предсказании.* При бесконечных данных, данная неопределенность будет стремится к нулю, в то время как неопределенность из-за шума - нет. Поскольку в офлайн-обучении с подкреплением датасет ограничен, агенту крайне важно понимать пределы, за которыми предсказанные ценности действий уже не вызывают доверия. Так, оценка неопределенности используется для выражения уверенности модели динамики $P(s_{t+1}|s_t, a_t)$ в своих предсказаниях [34, 35], а также для предсказаний критиков $Q(s, a)$ [16, 21]. Как было описано в параграфе 2.3 ранее, такой подход может быть использован для поощрения консерватизма во время обучения агента. Помимо этого, оценка неопределенности может помогать с субоптимальным консерватизмом, приводя к более универсальным под-

ходам. Например, во время обучения можно обусловливаться на разный уровень неопределенности, чтобы динамически менять требуемый уровень консерватизма после обучения (обычно он зафиксирован) [36], или использовать Байесовский вывод, чтобы получать оптимально адаптивные алгоритмы для офлайн-обучения с подкреплением [37]. В данной работы мы используем RND оценку эпистемологический неопределенности как бонус за анти-исследование чтобы поощрять консерватизм. В отличие от других подходов, мы не используем ансамбли.

3.2. Эффективное обучение ансамблей и их недостатки

Ансамбли это мощный и при этом очень простой не Байесовский подход для оценки неопределенности, который на практике почти всегда превосходит даже Байесовские нейронные сети [38]. Однако обучение ансамблей как правило затратно по памяти и вычислительным ресурсам, так как обучать приходится N полных копий модели. Данное ограничение делает привлекательным исследование более эффективных способов обучения ансамблей без потери в разнообразии признаков (что является основным плюсом ансамблей). Так, вместо обучения N копий, можно использовать метод исключения (dropout) чтобы аппроксимировать Байесовский вывод в глубоких гаусовсских процессах [39]. За счет того, что исключение стохастически зануляет некоторые внутренние выходы или веса модели, во время обучения неявно получается ансамбль многих моделей. В более новой работе, показали, что можно вывести метод, который позволить интерполироваться между полным ансамблем и исключением (dropout) с помощью зафиксированных зануляющих масок, а также контролируемой степенью перекрытия между ними. Обучение ансамблей также можно сделать гораздо более эффективным как-то уменьшив количество обучаемых параметров в каждой модели. Например, в [40] существенно уменьшили стоимость обучения представив каждую матрицу весов как произведение Адамара между одной общей матрицей весов, а также некоторым набором уникальных

для каждого члена ансамбля векторов.

Несмотря на кажущееся разнообразие методов для более эффективного обучения ансамблей, как показали в [17], ни один из перечисленных методов при переносе в офлайн-обучение с подкреплением не получает результатов схожих с наивным обучением ансамблей. Более того, во многих случаях даже ухудшает результаты так, что использование ансамбля становится невыгодным. Авторы замечают, что необходимо дальнейшие исследования по обучению ансамблей в офлайн-обучении с подкреплением. Текущая работа по применению RND является логическим продолжением в этом направлении, т.к. RND позволяет добиваться результатов схожих с ансамблями, но гораздо дешевле.

4. Предварительные исследования

Несмотря на то, что с момента публикации метода RND прошло немало времени, а потенциал подобного метода для оценки неопределенности был замечен давно, попытки применить его в офлайн-обучении с подкреплением оказались редки, что вероятно связано с исследованием [21]. В нем авторы пришли к выводу, что RND не способен в достаточной мере различать действия из датасета и из вне (а соответственно не может служить и бонусом за анти-исследование) и отказались от применения RND в пользу вариационных автоэнкодеров (VAE). Основной целью данной главы является воспроизведение результатов предыдущего исследования, а также анализ возможных причин неудовлетворительных результатов.

4.1. Воспроизведение результатов предыдущего исследования

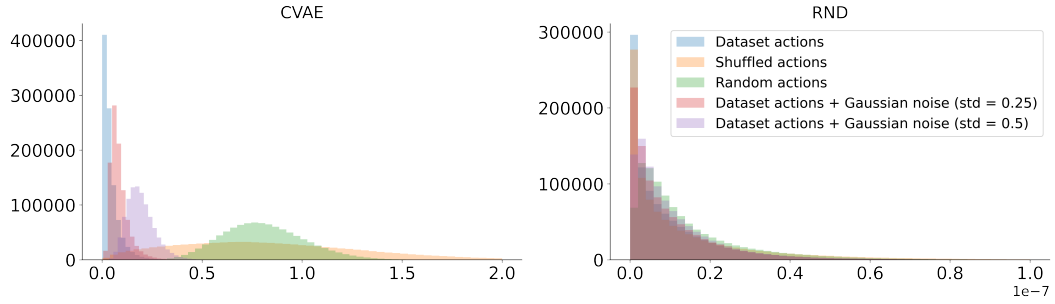


Рис. 4: Результат предыдущего исследования [21]. Основная цель эксперимента - показать распределение бонуса за анти-исследование для действий из обучающей выборки, а также для различных смещений. Можно заметить, что RND практически не реагирует на действия из вне обучающей выборки. Как показано далее, данный результат ошибочен.

Чтобы лучше понимать возможные трудности применения RND в офлайн-обучении с подкреплением, мы начнем с воспроизведения главного эксперимента из работы [21] (см. Рис. 4 для оригинальных результатов). Целью эксперимента является визуальное отображение бонуса за

анти-исследование на основе обученного RND (о том как он обучается более подробно было рассказано в параграфе 2.4) для состояний и действия из датасета, а также для разных пертурбации действий, чтобы смоделировать данные из вне распределения датасета. К действиям из датасетя применяется случайный гауссовский шум с разной величиной стандартного отклонения, а также берутся полностью случайные действия из стандартного равномерного распределения. Дополнительно, мы также визуализируем бонус за анти-исследование на основе обученного ансамбля из критиков как в SAC-N ($N = 25$) алгоритме [16], где в качестве бонуса берется стандартное отклонение предсказаний ансамбля. Состояния и действия подаются в нейронные сети с помощью простой конкатенации. Результаты эксперимента отображены на Рис. 5. В качестве датасета был использован *walker2d-medium* из набора данных D4RL [22] (более подробно набор описан далее в параграфе 6.1).

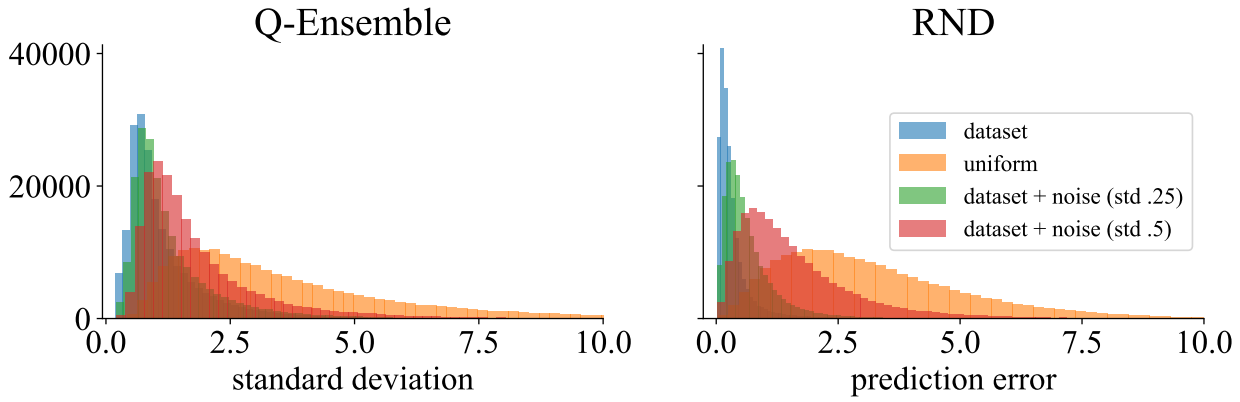


Рис. 5: Визуализация бонуса за анти-исследования на основе предобученного ансамбля и RND для состояний и действия из датасета walker2d-medium. Также рассмотрены действия из вне обучающей выборки, а именно случайные действия и действия с добавлением шума. Как можно заметить, RND сравним с ансамблем в детектировании новых данных, т.к. бонус за анти-исследование максимальный для случайных действий и минимальный для действий из датасета.

Как можно увидеть на Рис. 5 результаты эксперимента значительно отличаются от результатов в работе [21] (см. Рис. 4), несмотря на воспроизведение близкое к условиям оригинальной работы. Результаты показывают, что обученный RND модуль обладает высокой чувсти-

тельностью к действиям из датасета и из вне распределения датасета с разной степенью смещения, а значит можешь различать их, что и требуется от эффективного босуна за анти-исследование. В предыдущем исследовании авторы пришли к обратному выводу, что RND может работать только для дискретныйх пространств действий и визуальных пространств состояний, заключая, что применение RND для более широкого спектра сред нетривиально.

В попытках разобраться в причине такого противоречия в полученных результатах, был проведен подробный анализ выложенного в открытый доступ исходного кода[3] оригинальной публикации в поисках различий в имплементации алгоритмов. Главным отличием оказалась разница в размерах сетей *predictor* и *prior*. Вопреки советам из [28], в оригинальной публикации *predictor* меньше, чем *prior* на два полносвязных слоя. Важно, чтобы *predictor* обладал большей или сопоставимой выразительной силой, т.к. в противном случае он не сможет эффективно минимизировать функцию потерь RND на тренировочной выборке, а значит и не сможет точно реагировать на данные из вне распределения.

4.2. Анализ результатов воспроизведения

Хорошо работающий бонус за анти-исследование для непрерывных пространств действий, будь то RND, ансамбль критиков или любой другой, должен удовлетворять по крайне мере двум критериям. Во-первых, он должен быть достаточно чувствительным, чтобы смочь обнаружить действия из вне тренировочного распределения и занизить предсказанные значения суммарной награды (см. выражение 1). В идеале, бонус также должен быть как можно ближе к нулю для данных из тренировочного распределения, чтобы не смещать Q-функцию, т.к. сильное смещение может привести к замедлению сходимости или расхождению алгоритма. Во-вторых, бонус должен позволять агенту легко минизировать его с помощью градиентного спуска по время обучения.

В предыдущем параграфе было показано, что RND достатчочно

[3] https://github.com/shidilrzf/Anti-exploration-RL

чувствителен к действиям из вне обучающего распределения. Тем не менее наивное применение RND в качестве бонуса за анти-исследование на основе Soft Actor Critic (SAC) [41] приводит к плохим результатам (см. Рис. 9), а также нестабильному обучению, что сходится с итоговыми выводами в исследовании [21]. Подобный результат может говорить о том, что проблема кроется в чем-то не связанном со способностью RND оценивать новизну, а в том, что агент не может эффективно минимизировать бонус за анти-исследование во время обучения.

(a) (b) (c)

Рис. 6: Визуализация эффекта от разного обусловливания в *prior* сети RND на обучение агента. Для визуализации были использованы датасеты halfcheetah-medium, walker2d-medium и hopper-medium. **(a)** Бонус за анти-исследование для действий из датасета во время предобучения RND. Бонус дополнительно делится на бегущее среднее квадратичное отклонение, чтобы увеличить масштаб. **(b)** Бонус за анти-исследование для действий агента во время его обучения. В идеале бонус должен сойтись к финальным значениям в пункте (a). **(c)** Расстояние между действиями актора и действиями в датасете для одинаковых состояний. В идеале расстояние должно уменьшаться, т.к. действия из датасета имеют наименьший бонус по определению.

Чтобы проверить гипотезу о том, что агент не может эффективно минимизировать бонус за анти-исследование, мы упростили проблему, убрав из алгоритма SAC критика, так чтобы агент находился в тех же условях, но оптимизировал только бонус за анти-исследование. Так можно изолированно исследовать способность агента оптимизировать бонус. Ожидается, что в таких условиях, агент сможет успешно минимизировать бонус до теоретического минимума, то есть до значения функции потерь RND полученного во время предобучения (т.к. дальше RND зафиксирована и не обучается). Дополнительно, т.к. действия из

датасета по определению приводят к наболее низкому бонусу, ожидается, что расстояние от действий агента, до действий в датасете для тех же состояний будет снижаться во время обучения.

В данном эксперименте *predictor* использует простую конкатенацию состояний и действий на входе. Дополнительно, мы также исследуем более сложные методы обусловливания (более подробно были описани в параграфе 2.5) для *prior* сети. Обучение происходит на датасетах *halfcheetah-medium*, *walker2d-medium* и *hopper-medoum* из D4RL [22], а результаты усредняются по трем независимым запускам. Результаты эксперимента отображены на Рис. 6.

Результаты эксперимента подтверждают гипотезу о то, что актор не может эффективно минимизировать бонус за анти-исследование. Как можно увидеть, почти для всех способов обусловливания в *prior* сети бонус во время обучения агента гораздо выше (см. Рис. 6b), чем он должен быть в сравнении с бонусом для действий из датасета во время предобучения RND (см. Рис. 6a). Более того, Рис. 6c более наглядно показывает, что агент не может приблизиться к политике собиравшей датасет, т.к. в течении обучения расстояние между действиями агента и действиями датасета растет, а не падает. Однако, есть и исключение в виде обусловливания основанного на FiLM, т.к. с ним актор успешно справляется с минимизацией бонуса за анти-исследование, а расстояние до действий датасета быстро падает до минимума. Данный результат закладывает основу для нашего метода (который будет описан далее), а также показывает, что правильный механизм обусловливания в *prior* может существенно упростить оптимизацию бонуса за анти-исследование.

5. Метод

Таблица 1: Сравнение разных механизмов обусловливания для prior сети в RND.

Датасет	Concat	Gating	Bilinear	FiLM
hopper-medium-v2	94.8	39.7	98.4	86.3
hopper-medium-expert-v2	71.5	59.3	110.3	102.7
hopper-medium-replay-v2	100.3	51.3	100.8	100.3
walker2d-medium-v2	94.8	82.3	92.8	95.1
walker2d-medium-expert-v2	86.1	84.2	108.9	110.0
walker2d-medium-replay-v2	90.3	87.5	88.3	75.7
Среднее	89.6	67.3	**99.9**	95.0

Основываясь на результатах экспериментов в предыдущей главе, а также с учетом совершенных в предыдущих работах ошибок, в данной работе был разработан алгоритм **SAC-RND**. В качестве основы метода был выбрал алгоритм Soft Actor-Critic [41], так как он считается одним из лучших базовых алгоритмов в онлайн-обучении с подкреплением. Дополнительной причиной послужило и то, что на нем основаны уже упомянутые ранее SAC-N и EDAC [16], что значительно упрощает сравнение эффективности RND и ансамблямей, поскольку все остальные детали зафиксированы. Основными приемуществами SAC-RND являются простота имплементации, отсутствие ансабмля и как следствие скорость обучения. Как будет показано в следующей главе, SAC-RND показывает результаты сравнимые с SAC-N, сильно обходя аналоги без ансамблей. Далее более подробно опишем каждый компонент метода, а также детали имплементации.

Дистилляция случайных сетей. RND предобучается с помощью среднеквадратичной функции потерь (MSE) между (s, a) эмбеддингами *prior* и *predictor* сетей. Градиент считается только по параметрам *predictor* сети, а после предобучения (то есть во время обучения SAC) обе сети остаются фиксированными, хоть и пропускают градиент по

действиям для обучения агента. Обе сети RND имеют одинакой размер в 4 полносвязных слоя. В отличие от работ [20, 28], мы не добавляем дополнительные слои к *predictor* сети, чтобы предотвратить нежелательные результаты. В противном случае может получиться так, что слишком выразительный *predictor* может обобщиться на весь *prior*, тем самым научившись совершенно точно предсказывать эмбеддинги даже на данных вне тренировочного распределения, тем самым перестав быть мерой неопределенности.

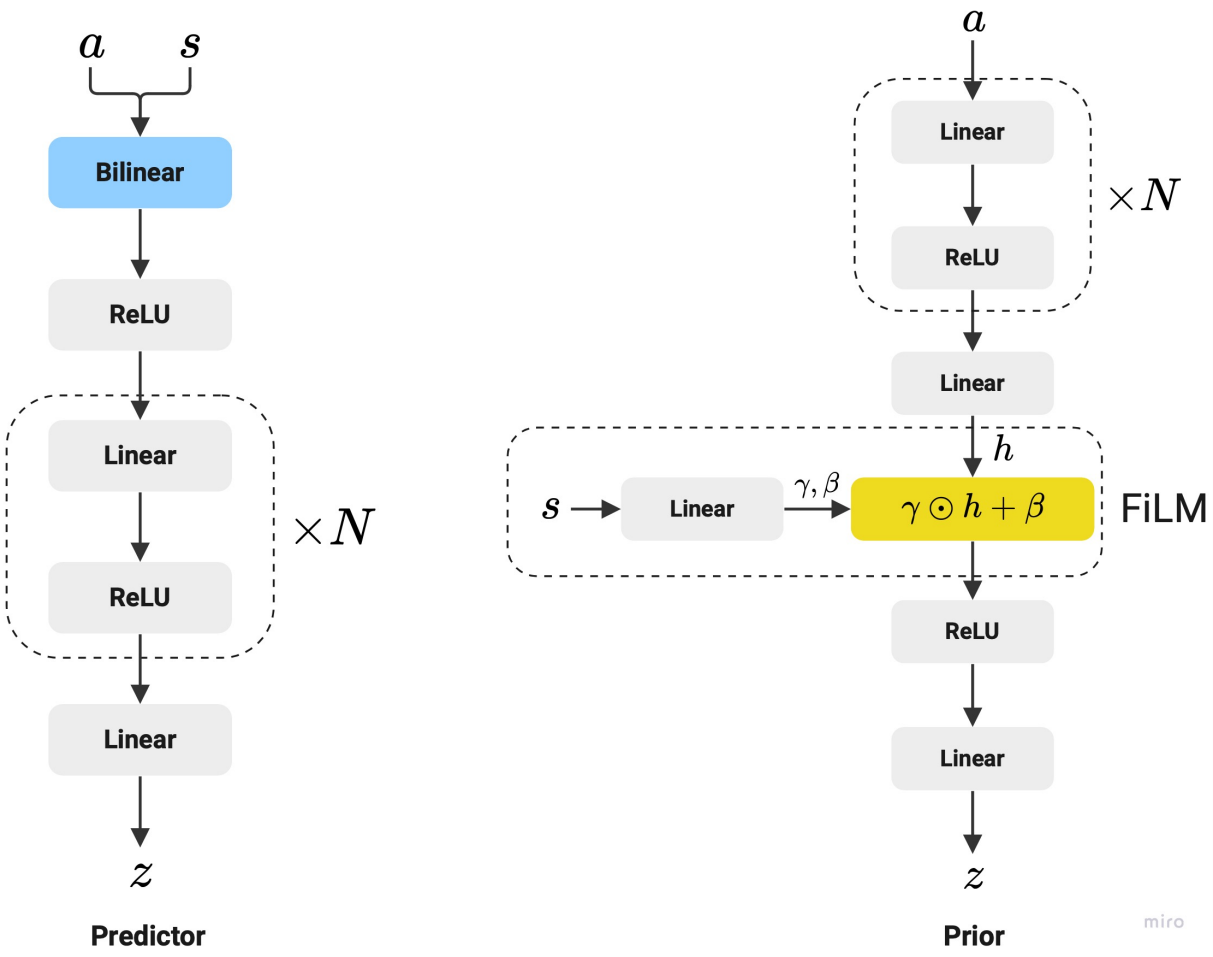


Рис. 7: Схематическое представление архитектуры RND.

В главе 4 было показано, что для оптимальной работы необходимо использовать обусловливание основанное на FiLM для *prior* на предпоследнем слое сети и до функции активации. В теории, обусловливание для *predictor* сети может быть произвольным [28], однако на практике выбор обусловливания также может дополнительно влиять на итоговые результаты. Чтобы выбрать лучший вариант был проведен эксперимент

на маленьком подмножестве датасетов D4RL, где был зафиксирован FiLM и варьировалось только обусловливание для *predictor* (см. Таблица 1). По результатам эксперимента был выбрал метод обусловливания с помощью билинейного преобразования на первом слое, т.к. он показал наилучшие результаты. Финальная архитетура RND представлена на Рис. 7.

Бонус за анти-исследование. Бонус определяется аналогично функции потерь при обучении RND как:

$$b(s,a) = \|f_\psi(s,a) - f_{\bar{\psi}}(s,a)\|_2^2$$

Он также дополнительно делится на скользящее стандартное отклонение, которое записывается во время фазы предобучения, чтобы дополнительно увеличить масштаб бонуса равномерно для всех сред. Такое масштабирование упрощает подбор гиперпараметров, уменьшая возможный диапазон полезных α коэффициентов, которые контролируют уровень консерватизм в течении обучения.

Детали имплементации. Метод реализован на языке программирования Python с помощью фреймворка для обучения нейронных сетей Jax [42]. Аналогично работам [9, 18, 43] при обучении критика дополнительно использовалась нормализация с помощью LayerNorm перед каждым полносвязным слоем, т.к. это существенно ускоряет сходимость и стабильность метода. Более подробно описание алгоритма в виде псевдокода, а также полное описание гиперпараметров представлено в публикации, которая выложена в открытый доступ на портале arXiv: `https://arxiv.org/abs/2301.13616`.

6. Эксперименты

В данной главе представлены эмпирические результаты оценки предложенного метода SAC-RND, а также сравнение с уже существующими методами основанными на ансамблях и без. Оценка проводилась на нескольких разделах из набора данных D4RL [22], а именно раздел Gym, который состоит сред HalfCheetah, Hopper и Walker2d, а также на более сложном AntMaze (см Рис. 8). Далее, в параграфе 6.2, проведен дополнительный визуальный анализ поясняющий, почему применение FiLM обусловливания в *prior* сети так сильно помогает агенту в оптимизации бонуса за анти-исследование. В заключение, в парграфе 6.3 дополнительно исследуются сочетания новых вариаций механизов обусловливания.

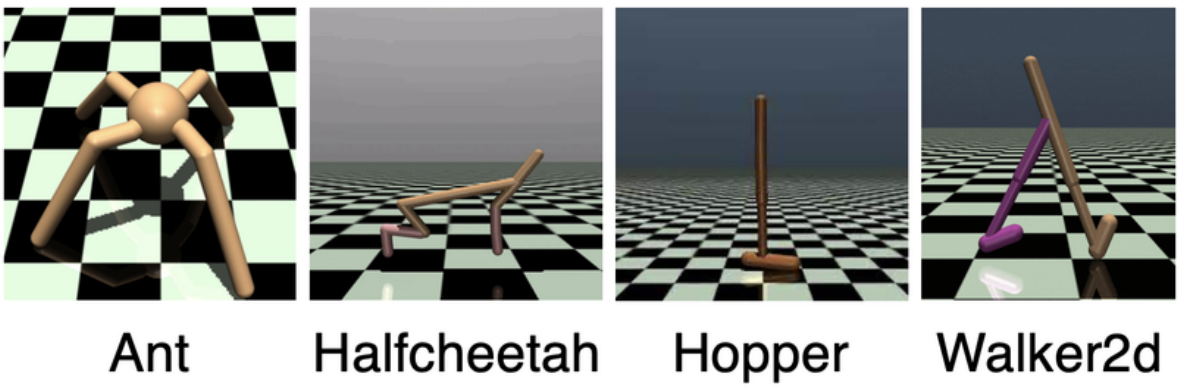


Рис. 8: Визуализация сред для собранных в D4RL датасетов.

Все эксперименты были проведены с помощью V100 и A100 GPU. Обучение на 3 миллиона градиентных шагов занимает $\sim$ 40 минут, в то время как на более популярные 1 миллон всего $\sim$ 15. Более подробно сравнения скорости с предыдущими методами приводится в Таблице 3. Во всех экспериментах, где это возможно, используются гиперпараметры аналогичные предыдущим работам, в то время как новые (такие как α) перебираются, чтобы выбрать лучший для каждого датасета. Основной метрикой является награда, нормализованная на среднюю награду датасета, где значение выше 100 означает, что метод превзошел агента, который собирал датасет.

6.1. Результаты на D4RL

Раздел Gym. Оценка SAC-RND проводится на всех доступных датасетах для сред HalfCheetah, Walker2d и Hopper в Gym разделе набора датасетов D4RL (см. Рис. 8). Для сравнения были взяты алгоритмы, использующие и не использующие ансамбли. В качестве неиспользующих ансамбли были выбраны CQL [14], IQL [15], TD3 + BC [10], т.к. они показывают хорошие результаты, а также широко используются на практике. В качестве использующих ансамбли были выбраны уже упомянутые SAC-N & EDAC [16], а также более новый RORL [19], на данный момент достигающий наиболее высоких метрик на выбранных датасетах. В соответствии с подходом [16] обучение длится 3 миллиона градиентных шагов, а оценка в среде проводится на 10 эпизодах. Обучение повторяется 3 раза с разными инициализаторами генератора случайных чисел, а результаты усредняются.

Таблица 2: Сравнение результатов SAC-RND на всех датасетах D4RL из раздела Gym с предыдущими методами.

	Не используют ансамбли			Используют ансамбли			
Датасет	**TD3+BC**	**IQL**	**CQL**	**SAC-N**	**EDAC**	**RORL**	**SAC-RND**
halfcheetah-random	11.0 ± 1.1	13.1 ± 1.3	31.1 ± 3.5	28.0 ± 0.9	28.4 ± 1.0	28.5 ± 0.8	29.0 ± 1.5
halfcheetah-medium	48.3 ± 0.3	47.4 ± 0.2	46.9 ± 0.4	67.5 ± 1.2	65.9 ± 0.6	66.8 ± 0.7	66.6 ± 1.6
halfcheetah-expert	96.7 ± 1.1	95.0 ± 0.5	97.3 ± 1.1	105.2 ± 2.6	106.8 ± 3.4	105.2 ± 0.7	105.8 ± 1.9
halfcheetah-medium-expert	90.7 ± 4.3	86.7 ± 5.3	95.0 ± 1.4	107.1 ± 2.0	106.3 ± 1.9	107.8 ± 1.1	107.6 ± 2.8
halfcheetah-medium-replay	44.6 ± 0.5	44.2 ± 1.2	45.3 ± 0.3	63.9 ± 0.8	61.3 ± 1.9	61.9 ± 1.5	54.9 ± 0.6
halfcheetah-full-replay	-	-	76.9 ± 0.9	84.5 ± 1.2	84.6 ± 0.9	-	82.7 ± 0.9
hopper-random	8.5 ± 0.6	7.9 ± 0.2	5.3 ± 0.6	31.3 ± 0.0	25.3 ± 10.4	31.4 ± 0.1	31.3 ± 0.1
hopper-medium	59.3 ± 4.2	66.2 ± 5.7	61.9 ± 6.4	100.3 ± 0.3	101.6 ± 0.6	104.8 ± 0.1	97.8 ± 2.3
hopper-expert	107.8 ± 7.0	109.4 ± 0.5	106.5 ± 9.1	110.3 ± 0.3	110.1 ± 0.1	112.8 ± 0.2	109.7 ± 0.3
hopper-medium-expert	98.0 ± 9.4	91.5 ± 14.3	96.9 ± 15.1	110.1 ± 0.3	110.7 ± 0.1	112.7 ± 0.2	109.8 ± 0.6
hopper-medium-replay	60.9 ± 18.8	94.7 ± 8.6	86.3 ± 7.3	101.8 ± 0.5	101.0 ± 0.5	102.8 ± 0.5	100.5 ± 1.0
hopper-full-replay	-	-	101.9 ± 0.6	102.9 ± 0.3	105.4 ± 0.7	-	107.3 ± 0.1
walker2d-random	1.6 ± 1.7	5.4 ± 1.2	5.1 ± 1.7	21.7 ± 0.0	16.6 ± 7.0	21.4 ± 0.2	21.5 ± 0.1
walker2d-medium	83.7 ± 2.1	78.3 ± 8.7	79.5 ± 3.2	87.9 ± 0.2	92.5 ± 0.8	102.4 ± 1.4	91.6 ± 2.8
walker2d-expert	110.2 ± 0.3	109.9 ± 1.2	109.3 ± 0.1	107.4 ± 2.4	115.1 ± 1.9	115.4 ± 0.5	114.3 ± 0.6
walker2d-medium-expert	110.1 ± 0.5	109.6 ± 1.0	109.1 ± 0.2	116.7 ± 0.4	114.7 ± 0.9	121.2 ± 1.5	105.0 ± 7.9
walker2d-medium-replay	81.8 ± 5.5	73.8 ± 7.1	76.8 ± 10.0	78.7 ± 0.7	87.1 ± 2.4	90.4 ± 0.5	88.7 ± 7.7
walker2d-full-replay	-	-	94.2 ± 1.9	94.6 ± 0.5	99.8 ± 0.7	-	109.2 ± 1.8
Среднее	67.5	68.9	73.6	84.4	**85.2**	**85.7**	**85.2**

Итоговые результаты представлены в Таблице 2. Как можно увидеть, SAC-RND существенно выделяется на фоне алгоритом без ансамблей, превосходя их по средней нормализованной награды с большим отрывом. Более того, SAC-RND результаты сравнимы с алгоритмами

использующие ансамбли, которым для достижения хороших результатов может потребоваться до 500 членов в ансамбле для SAC-N & EDAC и 20 членов для RORL. Чтобы лучше показать важность предложенных в данной работе изменений, на Рис. 9 дополнительно визуализирован прирост в нормализованной награде при изменениях описанных в Главе 5 по сравнению с наивным применением RND.

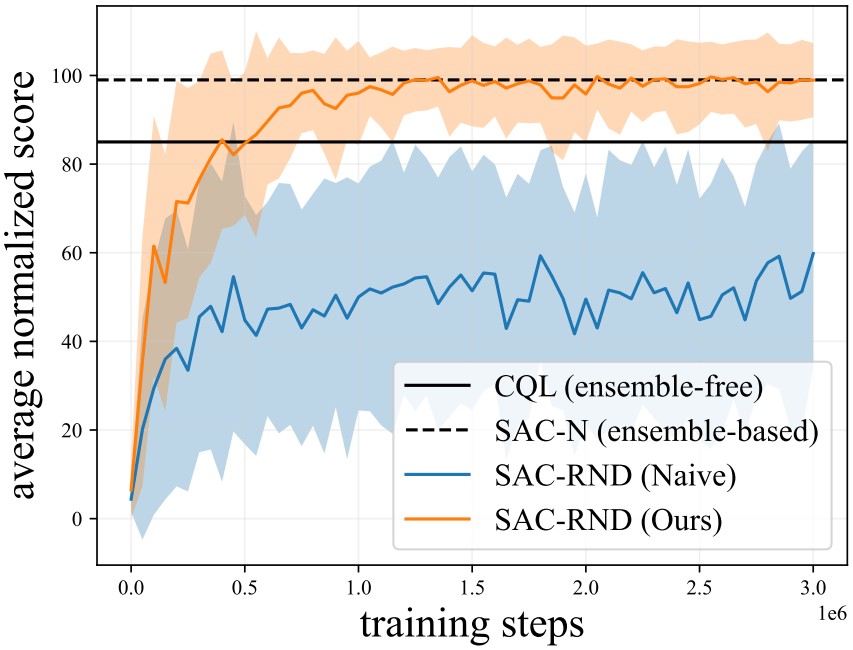


Рис. 9: Визуализация нормализованной награды усредненной по всем датасетам Gym. Показывает, что предложенные в данной работе изменения являются основополагающими для достижения хорошего результата, т.к. наивная версия SAC-RND сходится к очень низкой нормализованной награде и обладает плохой стабильностью.

Раздел AntMaze. Оценка SAC-RND проводится на всех доступных датасетах в AntMaze разделе набора датасетов D4RL (см. Рис. 8). Как и для раздела **Gym** для сравнения были взяты алгоритмы с ансамблями и без. В качестве неиспользующих ансамбли были взяты те же алгоритмы, что и для раздела **Gym**. В качестве использующих ансамбли были выбраны RORL [19] и MSG [17], последний из которых, насколько нам известно, в настоящее время имеет лучшую среднюю нормализованную награду в разделе **AntMaze**. SAC-N и EDAC не были включены, т.к. для них нет публичных результатов в этом разделе, а также при воспроизведении они получают нулевые награды. В со-

Таблица 3: Сравнение скорости обучения.

Алгоритм	**Шагов / сек ↑**	**Суммарное время (мин) ↓**
TD3+BC	1485	11.2
SAC-2	1285	12.9
SAC-RND	850	19.6
SAC-10	809	20.5
SAC-20	559	29.7
SAC-100	171	97.3
SAC-200	93	178.3
SAC-500	39	424.0

ответствии с подходом [16] обучение длится 3 миллиона градиентных шагов, а оценка в среде проводится на 100 эпизодах. Обучение повторяется 3 раза с разными инициализаторами генератора случайных чисел, а результаты усредняются.

Таблица 4: Сравнение результатов SAC-RND на всех датасетах D4RL из раздела AntMaze с предыдущими методами.

	Не используют ансамбли			Используют ансамбли		
Датасет	**TD3+BC**	**IQL**	**CQL**	**RORL**	**MSG**	**SAC-RND**
antmaze-umaze	78.6	87.5	74.0	97.7 ± 1.9	97.8 ± 1.2	97.2 ± 1.2
antmaze-umaze-diverse	71.4	62.2	84.0	90.7 ± 2.9	81.8 ± 3.0	83.5 ± 7.7
antmaze-medium-play	10.6	71.2	61.2	76.3 ± 2.5	89.6 ± 2.2	65.5 ± 35.7
antmaze-medium-diverse	3.0	70.0	53.7	69.3 ± 3.3	88.6 ± 2.6	88.5 ± 9.2
antmaze-large-play	0.2	39.6	15.8	16.3 ± 11.1	72.6 ± 7.0	67.2 ± 6.1
antmaze-large-diverse	0.0	47.5	14.9	41.0 ± 10.7	71.4 ± 12.2	57.6 ± 22.7
Среднее	27.3	63.0	50.6	65.2	**83.6**	**76.6**

Итоговые результаты представлены в Таблице 4. В работе [15] было показано, что многие алгоритмы, которые показывают хорошие результаты в разделе **Gym**, терпят неудачу при переходе в раздел **AntMaze**. Как можно увидеть, SAC-RND и тут показывает результаты сравнимые с методами использующими ансамбли. Результаты также подтверждают, что выбор механизмов обусловливания сделанный в Главе 5 хорошо обобщается на новые среды, на которых они не были протестированы ранее. Стоит заметить, что в дополнение к ансамблям MSG и RORL оба нуждаются в предобучении с помощью behavioural cloning, чтобы

достигнуть приведенных результатов, в то время как SAC-RND не требует никаких дополнительных модификаций, чтобы работать в данном разделе.

6.2. Почему обусловливание с помощью FiLM работает?

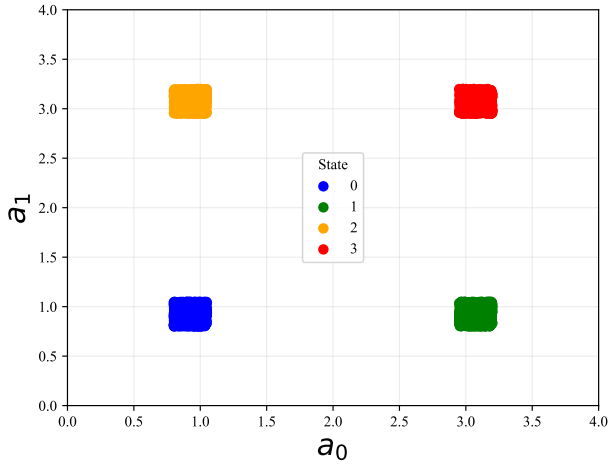
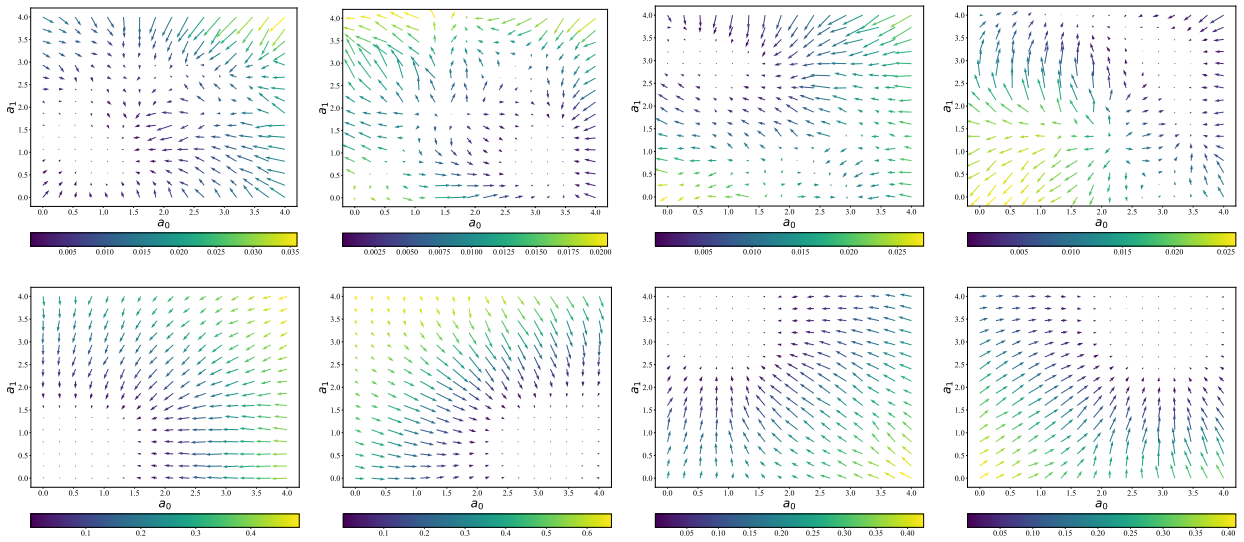


Рис. 10: Визуализация игрушченого датасета для эксперимента в параграфе 6.2.

В Главе 4 было показано, что обусловливание для *prior* сети в RND значительно улучшило способность актора минимизировать бонус за анти-исследование. Посколькоу проблема возникла во время обучения агента, может возникнуть гипотеза, что это как-то связано с ландшафтом функции бонуса за анти-исследование. В данном паргарфе проверяется данная гипотеза путем визуализации антиградиентов функции бонуса относительно действий для обусловливания с помощью простой конкатенации и с помощью FiLM. Для возможности визуализации в двумерном пространстве был разработан игрушечный датасет с четырьмя дискретными состояниями в каждом углу ограниченного пространства, где для каждого состояния были засэмплированы двухмерные действия из равномерного распределения (см Рис. 10). Все гиперпараметры были зафиксированы, после чего предобучены две RND отличающиеся только механизмом обусловливания в *prior* сети, в то время

как *predictor* использует простую конкатенацию. Далее на Рис. 11 визуализировано пространство антиградиента бонуса за анти-исследование относительно действий, обусловленного на каждое состояние по отдельности.



(a) Состояние 0 (b) Состояние 1 (c) Состояние 2 (d) Состояние 3

Рис. 11: Визуализация антиградиента для бонуса за анти-исследование на основе предобученного RND с обусловливанием с помощью простой конкатенации и с помощью FiLM. Для предобучения использовался игрушечный датасет (см Рис. 10).

Эффект обусловливания с помощью FiLM становится более очевидным при взгляде на результаты визуализации (см. Рис. 11). Для конкатенации получившиеся антиградиенты содержат много шума и локальных минимумов, указывая в направлении глобального минимума только в небольшой окрестности вокруг него, в то время как для FiLM антиградиенты точно указывают на глобальный минимум из любой точки пространства действий и каждого состояния. Хотя из подобного эксперимента нельзя совершенно точно заключить, что так происходит всегда, все же можно предположить что именно данный феномен проявляет себя и в пространствах более высокой размерности, как в датасетах D4RL, что было подтверждено экспериментально ранее.

6.3. Дополнительные сравнения механизмов обусловливания

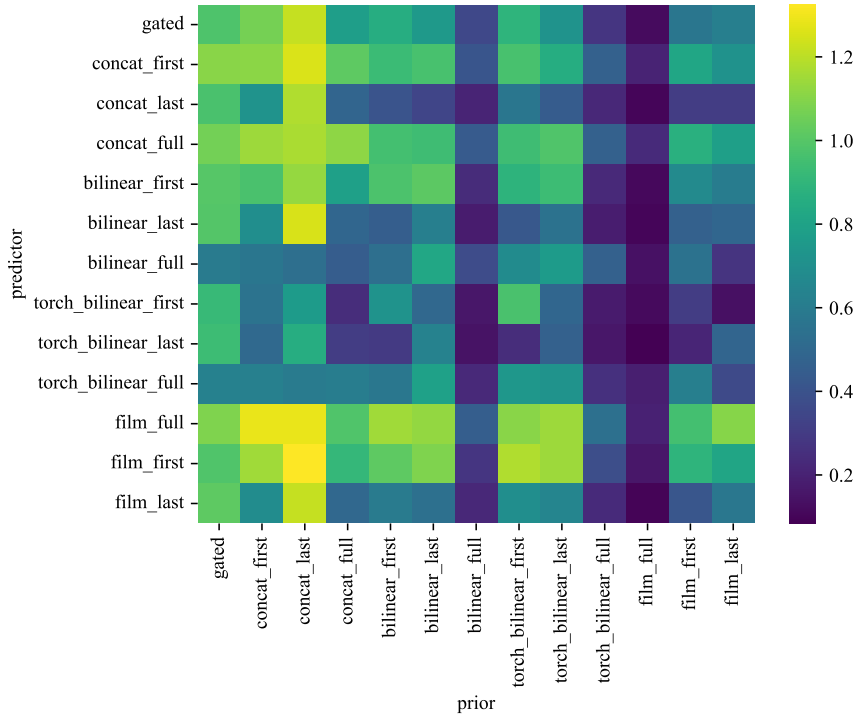


Рис. 12: Финальное значение MSE функции потерь между действиями агента обученного с помощью RND (как в Главе 4) и реальными действиями из датасета на одних и тех же состояниях. Для предобучения рассматривались все возможные сочетания механизмов обусловливания, предложенных в параграфе 6.3.

В свете полученных результатом можно задаться несколькими вопросами: как разные механзимы обусловливания для *predictor* и *prior* взаимодействуют между собой? на какой глубине по слоям выгоднее всего использовать обусловливание? Чтобы ответить на эти вопросы, были проведены дополнительные эксперименты с новыми вариациями механизом обусловливания: обсулавливание на первом слое, на последнем слое и на каждом слое. Также было взято две вариации билинейного преобразования: полный, как описано в [30], а также упрощенный, который по умолчанию используется в библиотеке PyTorch. На Рис. 12 визуализировано финальное значение MSE функции потерь между действиями агента обученного с помощью RND (как в Главе 4) и реальными действиями из датасета на одних и тех же состояниях. На осно-

ве результатов можно сделать два интересных замечания. Во-первых, обусловливание с помощью FiLM возможно является не единственным способом достичь хорошего результата, т.к. обусловливание на каждом слое с помощью обоих вариантов билинейного преобразования позволяет достигать похожих на FiLM результатов. Однако, по сравнению с FiLM, билинейные преобразования на внутренних слоях гораздо дороже вычислительно, т.к. включают в себя как минимум одну трехмерную и две двухмерных матрицы весов, а внутренние размерности как правило сильно выше входных. Во-вторых, оказывается, что обусловливание наиболее выгодно на последнем слое для *predictor* и на всех слоях для *prior*. Тем не менее, несмотря на достаточно большое и разнообразное количество датасетов в D4RL, на новых средах и датасетах зависимости могут измениться.

7. Заключение

В данной работе были пересмотрены результаты предыдущего исследования [21], показав, что RND может быть использован как эффективный метод для оценки неопределенности в офлайн-обучении с подкреплением. Тем не менее, наивное использование RND может привести в неудовлетворительным результатам. Был проведен подробный анализ возможных причин и найдена проблема, заключающаяся в том, что при наивном обусловливании на состояния актор не может эффективно минимизировать бонус за анти-исследование. В качестве решения были исследованы различные способы обусловливания, среди которых лучше всего себя показад FiLM. Именно он лег в основу нового эффективного алгоритма SAC-RND. Алгоритм SAC-RND был тщательно проверен на множестве датасетов из набора данных D4RL, где показал результаты сравнимые с алгоритмами использующими ансамбли и на порядок превзошел алгоритмы не использующие ансамбли. Дополнительно было показаны возможные причины подобного прироста в результатах при использовании FiLM, а также исследованы новые вариации обусловливания. Таким образом данная работа вносит существенный вклад в развитие быстрых и эффективных методов офлайн-обучения с подкреплением на основе оценки неопределенности, а научная публикация написанная по итогам данной работы была принята на конференцию ICML 2023.

Список литературы

[1] Agent57: Outperforming the atari human benchmark / Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski et al. // International Conference on Machine Learning / PMLR. — 2020. — P. 507–517.

[2] Mastering atari, go, chess and shogi by planning with a learned model / Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert et al. // Nature. — 2020. — Vol. 588, no. 7839. — P. 604–609.

[3] Dota 2 with large scale deep reinforcement learning / Christopher Berner, Greg Brockman, Brooke Chan et al. // arXiv preprint arXiv:1912.06680. — 2019.

[4] Video pretraining (vpt): Learning to act by watching unlabeled online videos / Bowen Baker, Ilge Akkaya, Peter Zhokhov et al. // arXiv preprint arXiv:2206.11795. — 2022.

[5] Offline reinforcement learning: Tutorial, review, and perspectives on open problems / Sergey Levine, Aviral Kumar, George Tucker, Justin Fu // arXiv preprint arXiv:2005.01643. — 2020.

[6] Personalization for Web-based Services using Offline Reinforcement Learning / Pavlos Athanasios Apostolopoulos, Zehui Wang, Hanson Wang et al. // arXiv preprint arXiv:2102.05612. — 2021.

[7] Pulserl: Enabling offline reinforcement learning for digital marketing systems via conservative q-learning / Douglas W Soares, Acordo Certo, Telma Lima, Deep Learning Brazil // Advances in Neural Information Processing Systems, 2nd Workshop on Offline Reinforcement Learning. — 2021.

[8] Deepthermal: Combustion optimization for thermal power generating units using offline reinforcement learning / Xianyuan Zhan, Haoran Xu, Yue Zhang et al. // Proceedings of the AAAI Conference on Artificial Intelligence. — Vol. 36. — 2022. — P. 4680–4688.

[9] Offline Q-Learning on Diverse Multi-Task Data Both Scales And Generalizes / Aviral Kumar, Rishabh Agarwal, Xinyang Geng et al. // arXiv preprint arXiv:2211.15144. — 2022.

[10] Fujimoto Scott, Gu Shixiang Shane. A minimalist approach to offline reinforcement learning // Advances in neural information processing systems. — 2021. — Vol. 34. — P. 20132–20145.

[11] Awac: Accelerating online reinforcement learning with offline datasets / Ashvin Nair, Abhishek Gupta, Murtaza Dalal, Sergey Levine // arXiv preprint arXiv:2006.09359. — 2020.

[12] Stabilizing off-policy q-learning via bootstrapping error reduction / Aviral Kumar, Justin Fu, Matthew Soh et al. // Advances in Neural Information Processing Systems. — 2019. — Vol. 32.

[13] Wu Yifan, Tucker George, Nachum Ofir. Behavior regularized offline reinforcement learning // arXiv preprint arXiv:1911.11361. — 2019.

[14] Conservative q-learning for offline reinforcement learning / Aviral Kumar, Aurick Zhou, George Tucker, Sergey Levine // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 1179–1191.

[15] Kostrikov Ilya, Nair Ashvin, Levine Sergey. Offline reinforcement learning with implicit q-learning // arXiv preprint arXiv:2110.06169. — 2021.

[16] Uncertainty-based offline reinforcement learning with diversified q-ensemble / Gaon An, Seungyong Moon, Jang-Hyun Kim, Hyun Oh Song // Advances in neural information processing systems. — 2021. — Vol. 34. — P. 7436–7447.

[17] Ghasemipour Seyed Kamyar Seyed, Gu Shixiang Shane, Nachum Ofir. Why So Pessimistic? Estimating Uncertainties for Offline RL through Ensembles, and Why Their Independence Matters // arXiv preprint arXiv:2205.13703. — 2022.

[18] Q-Ensemble for Offline RL: Don't Scale the Ensemble, Scale the Batch Size / Alexander Nikulin, Vladislav Kurenkov, Denis Tarasov et al. // arXiv preprint arXiv:2211.11092. — 2022.

[19] RORL: Robust Offline Reinforcement Learning via Conservative Smoothing / Rui Yang, Chenjia Bai, Xiaoteng Ma et al. // arXiv preprint arXiv:2206.02829. — 2022.

[20] Exploration by random network distillation / Yuri Burda, Harrison Edwards, Amos Storkey, Oleg Klimov // arXiv preprint arXiv:1810.12894. — 2018.

[21] Offline reinforcement learning as anti-exploration / Shideh Rezaeifar, Robert Dadashi, Nino Vieillard et al. // Proceedings of the AAAI Conference on Artificial Intelligence. — Vol. 36. — 2022. — P. 8106–8114.

[22] D4rl: Datasets for deep data-driven reinforcement learning / Justin Fu, Aviral Kumar, Ofir Nachum et al. // arXiv preprint arXiv:2004.07219. — 2020.

[23] Reward is enough / David Silver, Satinder Singh, Doina Precup, Richard S. Sutton // Artificial Intelligence. — 2021. — Vol. 299. — P. 103535. — URL: https://www.sciencedirect.com/science/article/pii/S0004370221000862.

[24] Scalar reward is not enough: A response to Silver, Singh, Precup and Sutton (2021) / Peter Vamplew, Benjamin J Smith, Johan Källström et al. // Autonomous Agents and Multi-Agent Systems. — 2022. — Vol. 36, no. 2. — P. 41.

[25] Sutton Richard S, Barto Andrew G. Reinforcement learning: An introduction. — MIT press, 2018.

[26] Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning / Denis Yarats, David Brandfonbrener, Hao Liu et al. // arXiv preprint arXiv:2201.13425. — 2022.

[27] Curiosity in hindsight / Daniel Jarrett, Corentin Tallec, Florent Altché et al. // arXiv preprint arXiv:2211.10515. — 2022.

[28] Conservative uncertainty estimation by fitting prior networks / Kamil Ciosek, Vincent Fortuin, Ryota Tomioka et al. // International Conference on Learning Representations. — 2019.

[29] Feature-wise transformations / Vincent Dumoulin, Ethan Perez, Nathan Schucher et al. // Distill. — 2018. — https://distill.pub/2018/feature-wise-transformations.

[30] Multiplicative interactions and where to find them / Siddhant M Jayakumar, Wojciech M Czarnecki, Jacob Menick et al. — 2020.

[31] Training agents using upside-down reinforcement learning / Rupesh Kumar Srivastava, Pranav Shyam, Filipe Mutz et al. // arXiv preprint arXiv:1912.02877. — 2019.

[32] Film: Visual reasoning with a general conditioning layer / Ethan Perez, Florian Strub, Harm De Vries et al. // Proceedings of the AAAI Conference on Artificial Intelligence. — Vol. 32. — 2018.

[33] Estimating risk and uncertainty in deep reinforcement learning / William R Clements, Bastien Van Delft, Benoît-Marie Robaglia et al. // arXiv preprint arXiv:1905.09638. — 2019.

[34] Mopo: Model-based offline policy optimization / Tianhe Yu, Garrett Thomas, Lantao Yu et al. // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 14129–14142.

[35] Morel: Model-based offline reinforcement learning / Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, Thorsten Joachims // Advances in neural information processing systems. — 2020. — Vol. 33. — P. 21810–21823.

[36] Hong Joey, Kumar Aviral, Levine Sergey. Confidence-Conditioned Value Functions for Offline Reinforcement Learning // arXiv preprint arXiv:2212.04607. — 2022.

[37] Offline rl policies should be trained to be adaptive / Dibya Ghosh, Anurag Ajay, Pulkit Agrawal, Sergey Levine // International Conference on Machine Learning / PMLR. — 2022. — P. 7513–7530.

[38] Lakshminarayanan Balaji, Pritzel Alexander, Blundell Charles. Simple and scalable predictive uncertainty estimation using deep ensembles // Advances in neural information processing systems. — 2017. — Vol. 30.

[39] Gal Yarin, Ghahramani Zoubin. Dropout as a bayesian approximation: Representing model uncertainty in deep learning // international conference on machine learning / PMLR. — 2016. — P. 1050–1059.

[40] Wen Yeming, Tran Dustin, Ba Jimmy. Batchensemble: an alternative approach to efficient ensemble and lifelong learning // arXiv preprint arXiv:2002.06715. — 2020.

[41] Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor / Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, Sergey Levine // International conference on machine learning / PMLR. — 2018. — P. 1861–1870.

[42] Bradbury James, Frostig Roy, Hawkins Peter et al. JAX: composable transformations of Python+NumPy programs. — 2018. — URL: `http://github.com/google/jax`.

[43] Smith Laura, Kostrikov Ilya, Levine Sergey. A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning // arXiv preprint arXiv:2208.07860. — 2022.