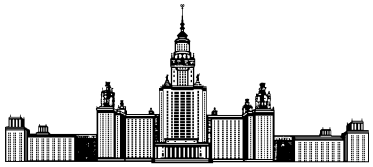
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

КУРСОВОЙ ПРОЕКТ

«Сравнение методов введения внутренней мотивации в обучении с подкреплением»

Выполнили:

Цыпин Артем Андреевич

Шамшиев Мамат Мамбетович

Филимонов Владислав Аскольдович

Москва, 2020

Содержание

1 Введение

В настоящее время обучение с подкреплением (RL) является одной из наиболее быстрорастущих областей в машинном обучении. С помощью обучения с подкреплением удалось решить ряд задач, традиционно считавшихся крайне сложными для компьютеров. Так, например, OpenAI и DeepMind добились успехов в таких играх, как StarCraft II и Dota2 [1], [2].

В обучении с подкреплением агент обучается максимизировать полученную от среды награду. Такая награда является для агента внешней. Во многих задачах, чтобы добиться желаемого поведения от агента, функцию награды приходится аккуратно настраивать. Такой подход плохо масштабируется и является непростой инженерной задачей.

Несмотря на то, что обучение с подкреплением помогло превзойти человека во многих дисциплинах, в большинстве решенных задач функция награды является «плотной». Такая функция награды непрерывно даёт агенту информацию о корректности его действий. Примером такой функции награды является бегущий счет в видеоиграх.

Однако во многих задачах в реальной жизни функция награды является крайне разреженной, в связи с чем агент может не получить никакого положительного отклика от среды на протяжении всего игрового эпизода. В таких задачах возникает необходимость во внутренней функции награды, мотивирующей агента к исследованию среды и поиску «редкой» внешней награды. Такая внутренняя награда также называется внутренней мотивацией.

В [3], [4] в качестве внутренней награды предлагается использовать ошибку предсказания некоторой модели. Подобные методы имеют большое количество гиперпараметров, значения которых сильно влияют на обучение агента, а также итоговое качество. К таким параметрам относится наличие нормализации награды и наблюдений, масштаб внутренней награды, архитектура модели для предсказания, наличие «прогрева» и др.

В данной работе будет проведено сравнение методов введения внутренней мотивации, а также изучено влияние различных гиперпараметров на процесс обучения.

Будут проведены эксперименты в среде MountainCar-v0 [5] и сделаны выводы относительно области применимости каждого из рассмотренных методов.

1.1 Определения и обозначения

Неформально взаимодействие системы, принимающей решения, с окружающим миром можно описать следующим образом:

- Систему, принимающую решения, будем называть агентом.
- Агент взаимодействует со средой.
- Среда в каждый момент времени задается текущим состоянием.
- Агент задаёт процедуру выбора действия, называемую политикой.
- Взаимодействие агента со средой задаётся динамикой среды, которая также называется функцией переходов.

Формально процесс взаимодействия агента со средой задается с помощью Марковского процесса принятия решений.

Определение 1.1. Марковским процессом принятия решений (MDP) называется пятерка $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r, s_0)$, где

- $\mathcal{S}$ – множество допустимых состояний.
- $\mathcal{A}$ – множество допустимых действий.
- $\mathcal{T}$ – функция переходов, т.е. $p(s'|s, a)$.
- $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ – функция награды.
- s_0 – начальное состояние.

Заметим, что процесс зависит лишь от текущего состояния и не зависит от предыдущей истории: $p(s_{t+1}|s_t, s_{t-1}, \ldots, s_0) = p(s_{t+1}|s_t)$. Также отметим, что в будущем будут рассматриваться только стационарные марковские процессы, т.е. такие что: $\forall t\ p(s_{t+1}|s_t, a) = p(s_1|s_0, a)$.

Определение 1.2. Набор $\mathcal{T} = \{s_0, a_0, r_0, s_1, a_1, r_1, \dots\}$ называется траекторией.

Определение 1.3. Для заданного MDP и политики π определено распределение на траекториях:

$$p(\mathcal{T}|\pi) = p(s_0, a_0, r_0, s_1, a_1, r_1, \dots|\pi) = \prod_{t=0} p(s_{t+1}|s_t, a_t)\pi(a_t|s_t.) \tag{1}$$

Определение 1.4. Кумулятивной наградой называется следующий функционал:

$$\mathbb{E}_{\mathcal{T}\sim\pi}\sum_{t=0}\gamma^t r_t, \tag{2}$$

где $\gamma \in [0,\, 1]$ – коэффициент дисконтирования.

Соответственно, задача агента – максимизировать кумулятивную награду:

$$J(\pi) = \mathbb{E}_{\mathcal{T}\sim\pi}\sum_{t=0}\gamma^t r_t \to \max_{\pi}. \tag{3}$$

Определение 1.5. Для заданного MDP оценочной функцией состояния или V-функцией для политики π называется следующая величина:

$$V^{\pi}(s) = \mathbb{E}_{\mathcal{T}\sim\pi|s_0=s}\sum_{t=0}\gamma^t r_t. \tag{4}$$

Определение 1.6. Приближение функции переходов из Опр. 1.1 с обучаемыми параметрами называется моделью мира [6].

Функция оценивает среднюю кумулятивную награду, которую агент может получить, находясь в состоянии s и действуя согласно политике π.

Как видно из выражения (3), агент обучается максимизировать среднюю кумулятивную награду. При это на каждом шаге агент получает награду r_t от среды. В случае редкой внешней награды предлагается добавить к ней слагаемое, отвечающее «плотной» внутренней награде:

$$r_t = r_t^{extr} + r_t^{intr}. \tag{5}$$

2 Методы введения внутренней мотивации

Чтобы стимулировать агента к исследованию среды с редкой внешней наградой, в [3], [4], [7] предлагается использовать меру новизны некоторого состояния в качестве внутренней мотивации. Одним из подходов к определению новизны состояния

является обучение некоторой модели внутренней мотивации в self-supervised режиме. Ошибки данной модели могу трактоваться как мера новизны.

Причинами ошибок предсказания модели помимо новизны состояний
могут быть [4]:

- Стохастичность откликов для модели внутренней мотивации. Если в среде присутствует некоторая случайность, например, бросок монетки или иной источник энтропии в среде, отклик для модели среды будет стохастической функцией от факторов.
- Недостаточная выразительность модели. Если модель недостаточно выразительна или же в факторах недостает важной информации, ошибка предсказания будет ненулевой.
- Нестационарность откликов. Если отклики меняются со временем, ошибка предсказания будет меняться пропорционально силе изменений.
- Модель плохо обучена.

Перечисленные источники ошибки предсказания являются нежелательными, и модели внутренней мотивации должны учитывать их наличие.

2.1 Определение новизны через обучение модели мира

Модель прямой динамики. Рассмотрим функцию переходов $f(\hat{s}_{t+1}|s_t, a_t, \theta)$. Будем приближать это распределение с помощью δ - функции $\hat{f}(s_t, a_t)$. Такую функцию можно обучать предсказывать следующее состояние, решая следующую оптимизационную задачу:

$$\|\hat{f}(s_t, a_t|\theta) - s_{t+1}\|_2^2 \to \min_{\theta}, \tag{6}$$

где $(s_t, a_t, s_{t+1}) \sim p(\mathcal{T}|\pi)$.

На протяжении всего взаимодействия агента со средой в буффер складываются переходы переходы (s_t, a_t, s_{t+1}), на которых $\hat{f}$ в последствие обучается. Обучение модели мира можно встроить в любой алгоритм в обучении с подкреплением.

В некоторых задачах в обучении с подкреплением состояния $s \in \mathcal{S}$ являются очень сложными. К таким задачам относятся многие видеоигры, в которых состояния являются изображениями большого размера, содержащими много ненужной для агента информации. Поэтому предлагается использовать энкодер $\phi(s|\omega)$, переводящий состояния в некоторое внутреннее представление. Тогда оптимизационная задача выглядит следующим образом:

$$\|\hat{f}(\phi(s_t|\omega), a_t|\theta) - \phi(s_{t+1}|\omega)\|_2^2 \to \min_{\theta,\omega}. \tag{7}$$

Таким образом, функция $\hat{f}$ обучается предсказывать выход энкодера. Имея обученную функцию $\hat{f}$, можно определить внутреннюю награду в момент времени t как $r_t^{intr} = \frac{1}{2}\|\hat{f}(\phi(s_t), a_t) - \phi(s_{t+1})\|_2^2$.

Модель прямой динамики подвержена проблеме стохастичности откликов для модели внутренней мотивации, которую также называют проблемой «Шумного Телевизора» [4]. Приведем два простых примера.

Чтобы удостовериться, что агент не «заучивает» правильную последовательность действий, во многих задачах в обучении с подкреплением используют т.н. «липкие» действия (sticky actions). В этом случае на каждом шаге среда независимо от текущего состояния и действия, выбранное агентом, с некоторой вероятностью копирует предыдущее действие. В таком случае функция переходов в среде задается смесью распределений. Для модели прямой динамики это означает, что при одинаковых входных наборах (s_t, a_t) отклик s_{t+1} будет случайным.

Вторым примером может служить случайность, заданная непосредственно средой, например, экран, показывающий агенту белый шум. В таком случае любые действия агента будут приводить к случайному результату.

Модель обратной динамики. Рассмотрим энкодер $\phi(s|w)$, используемый для переведения состояний во внутреннее представление. Для решения описанной проблемы со стохастическими откликами энкодеру необходимо «отфильтровывать» из внутренних представлений потенциально бесполезную для агента информацию. Для этого введём функцию $g(s_t, s_{t+1}|\psi)$, предсказывающую по двум последовательным состояниям действие, которое было совершено между ними.

Для обучения g предлагается решить следующую оптимизационную задачу [3]:

$$-\sum_{a\in A}[a == a_t]\log(g_a(\phi(s_t|\omega),\phi(s_{t+1}|\omega)|\psi)) \to \min_{\omega,\psi}. \tag{8}$$

Как и для функции $\hat{f}$, приближающей функцию переходов, переходы (s_t, a_t, s_{t+1}) сэмплируются из $p(\mathcal{T}|\pi)$. Энкодер обучается совместно с моделью g end-to-end. Интуиция заключается в том, что во внутренних представлениях, полученных с помощью модели обратной динамики, будет отсутствовать информация об аспектах среды, на которые агент не может повлиять. Такие внутренние представления могут помочь уменьшить влияние эффекта «Шумного Телевизора» на агента.

Однако ошибка предсказания модели обратной динамики не является хорошей внутренней наградой. Примером может служить ситуация во многих видеоиграх, в которых состояния среды не зависят от некоторых действий агента. Например, ситуация, в которой агент находится в тупике, и лишь одно из действий может привести агента в новое состояния. Более того, модель обратной динамики также уязвима к «липким» действиям, так как отклик для неё становится стохастическим.

Intrinsic Curiosity Module (ICM). Возникает идея использовать преимущества обеих описанных моделей. В [3] было предложено использовать в качестве внутренней награды ошибку предсказания модели прямой динамики, которая получает на вход внутреннее представление, полученное энкодером, обученным с помощью модели обратной динами (Рис. 1).

Такие эмбеддинги помогают внутренней награде быть инвариантной к аспектам среды, неподконтрольным агенту, которые не влияют на него. Таким образом решается проблема со случайностью, заданной средой. Однако для ICM липкие действия по-прежнему являются источником стохастичности в откликах. Отметим также, что в модели прямой динамики, а также в ICM внутренние представления обучаются на протяжение всего взаимодействия агента со средой, что приводит к нестационарности откликов и, соответственно, нежелательной ошибке предсказания.

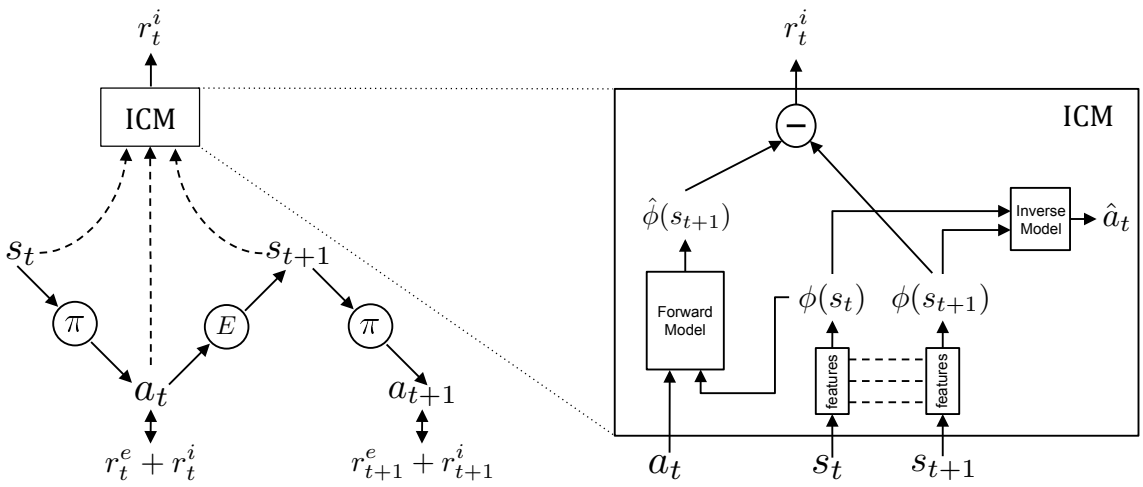


Рис. 1: Архитектура модели ICM. Модель обратной динамики служит своеобразным регуляризатором для эмбеддингов, которые в последствие используются в модели прямой динамики. При этом в качестве внутренней награды используется лишь ошибка предсказания модели прямой динамики.

2.2 Определение новизны через дистялляцию случайной нейросети

Альтернативный подход для определения состояний был предложен в [4]. Авторы заметили, что использование представлений, полученных фиксированным после случайной инициализации энкодером, в модели прямой динамики даёт хорошие результаты в ряде сложных задач. Одной из причин является то, что внутренние представления не изменяются в процессе взаимодействия агента со средой, что исключает фактор нестационарности откликов.

Исходя из этого наблюдения, было предложено заменить обучение модели мира на дистилляцию случайно инициализированной нейросети. Перед началом обучения случайно инициализируются две нейронных сети одинаковой архитектуры: $h : \mathcal{S} \to \mathbb{R}^n, \hat{h} : \mathcal{S} \to \mathbb{R}^n$. Одна из них фиксируется. Далее $\hat{h}$ обучается предсказывать выход фиксированной сети h:

$$\|\hat{h}(s_t|\theta) - h(s_t)\|_2^2 \to \min_\theta . \tag{9}$$

Интуиция заключается в том, что нейросеть $\hat{h}$ будет меньше ошибаться в состояниях, в которых агент был много раз. В то же время ошибка нейросети будет велика в новых состояниях из-за отсутствия подобных состояний в обучении.

Такой подход к определению новизны состояний позволяет избежать ряда проблем, связанных с обучением модели мира. Использование одинаковых нейросетевых архитектур позволяет исключить фактор недостаточной выразительности модели внутренней мотивации. Использование выхода фиксированной нейросети в качестве отклика для обучаемой модели внутренней мотивации позволяет исключить фактор нестационарности откликов модели. В силу того, что данная модель внутренней мотивации ни в какой момент времени не рассматривает действия, совершенные агентом, она не подвержена проблеме с «липкими» действиями. Однако, данный подход, в отличие от ICM, подвержен влиянию источников стохастичности в самой среде.

2.3 Влияние гиперпараметров методов на обучение агента

Описанные выше модели для определения новизны состояний на практике зависят от большого количества гиперпараметров. Иногда изменение одного из них приводит к тому, что агент не может выучить, какие переходы приводят его в новые состояния, и не получает внешней награды на протяжение всего взаимодействия со средой.

В данной работе будет рассмотрено влияния гиперпараметров методов на процесс обучения, а также найдены лучшие конфигурации для каждого из методов. Эксперименты будут проведены в среде MountainCar-v0. Несмотря на то, что пространство состояний и действий в этой задаче крайне простые, она не решается методами, не стимулирующими агента к исследованию среды.

Будут рассмотрены следующие гиперпараметры:

- Наличие прогрева для модели внутренней мотивации. Прогрев заключается в запуске случайного агента и обучении модели внутренний мотивации на траекториях, собранных данным агентом. Интуиция заключается в том, что в самом начале взаимодействия со средой модель внутренней мотивации обучена плохо и ошибка предсказания велика для всех состояний.

- Раздельные V-функции для внешней и внутренней награды. В случае, когда награда состоит из двух слагаемых $r_t = r_t^{extr} + r_t^{intr}$, V-функцию можно также представить в виде суммы: $V^\pi(s) = V_{exrt}^\pi(s) + V_{intr}^\pi(s)$. В случае общей V-функции масштаб внутренней награды может существеннее влиять на обучение агента.

- Нормализация награды. Для модели RND авторы отметили, что наличие нормализации награды необходимо для стабильного обучения, однако авторы ICM не обсуждают влияние нормализации на процесс обучения.

- Выразительность модели внутренней мотивации. В то время, как устройство метода RND позволяет утверждать, что выразительности модели внутренней мотивации всегда будет достаточно, методы, обучающие модель мира, требуют подбора архитектуры.

- Отсутствие внешней мотивации. Есть предположение [8] что во многих средах агент, не имеющий внешней награды, способен действовать «разумно». В аркадных играх его может мотивировать наличие новых объектов, которые появляются в процессе взаимодействия со средой.

3 Эксперименты

Все эксперименты будут проведены с методом PPO [9]. Для экспериментов параметры метода PPO были выбраны по умолчанию [10]. Модели внутренней мотивации будут оцениваться с помощью графиков обучения агента. По оси x будет отложено количество взаимодействий со средой, по оси y – средняя награда за эпизод. Все эксперименты будут проведены с пятью различными random seed'ами, на графиках будет изображено среднее значение и стандартное отклонение по запускам. Во всех экспериментах с раздельными V-функциями внутренняя награда будет рассматриваться как неэпизодическая [4].

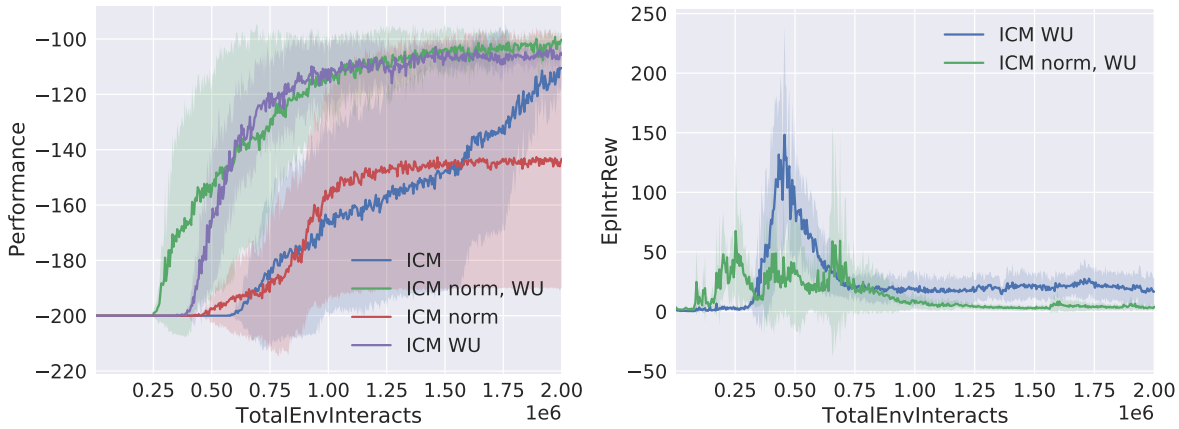
Как таковой внешней награды за достижение цели в среде MountainCar-v0 нет. Агент получает награду -1 за каждое совершенное действие. Такое устройство функции внешней награды стимулирует агента закончить эпизод как можно быстрее.

Команды для воспроизвдения экспериментов приведены в аппендиксе A.

3.1 Влияние прогрева и нормализации на процесс обучения

Целью данного эксперимента является анализ влияния прогрева, а также нормализации внутренней награды на обучения агента. Эксперимент будет проведен следующим образом. Будут рассмотрены модели прямой и обратной динамики, ICM, а также RND. Для моделей прямой и обратной динамики и ICM зафиксируем одинаковую архитектуру для энкодеров. Все модели внутренней мотивации будем рассматривать с раздельными V-функциями, без «липких» действий. Нормализовать награду будем с помощью стандартного отклонения. Длину прогрева примем равной одной эпохе.

Авторы метода RND в своей работе утверждают, что без прогрева и нормализации метод работает плохо. При этом прогрев и нормализация не упоминается в статьях, где обучается модель мира. Рассмотрим влияние прогрева и наличия нормализации на процесс обучения агентов, внутренняя мотивация для которых определяется ошибкой модели мира.



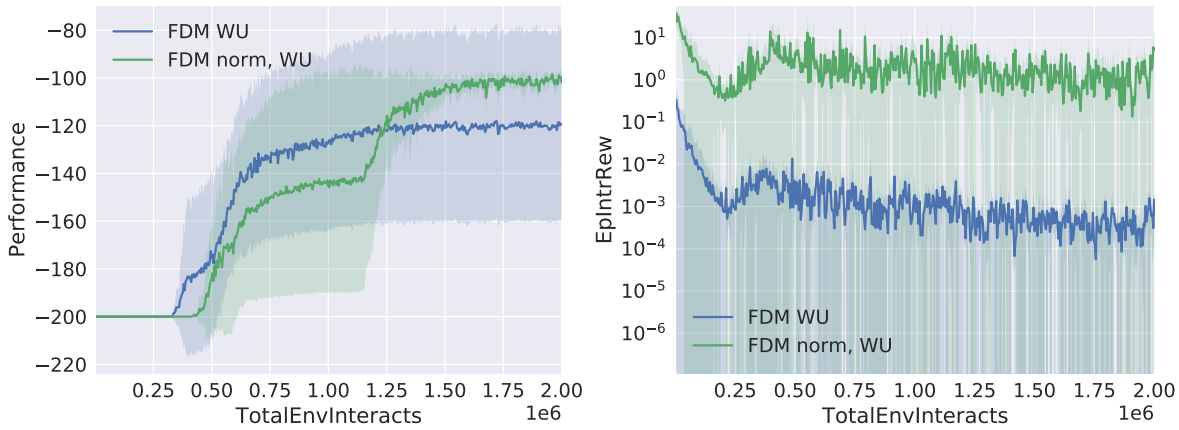
(a) Зависимость внешней награды от числа взаимодействий со средой.

(b) Зависимость внутренней награды от числа взаимодействий со средой.

Рис. 2: Обучение агента с ICM. «Norm» означает наличие нормализации, «WU» – наличие прогрева.

На Рис. 2a видно, что отсутствие прогрева сильно ухудшает процесс обучения. Большая дисперсия у «ICM norm» означает, что в некоторых запусках агент так и

не смог получить от среды положительной внешней награды. Таким образом, можно утверждать, что прогрев является ключевым для стабильного обучения агента фактором. В связи с этим, все последующие эксперименты будут проведены с прогревом.



(a) Зависимость внешней награды от числа взаимодействий со средой.

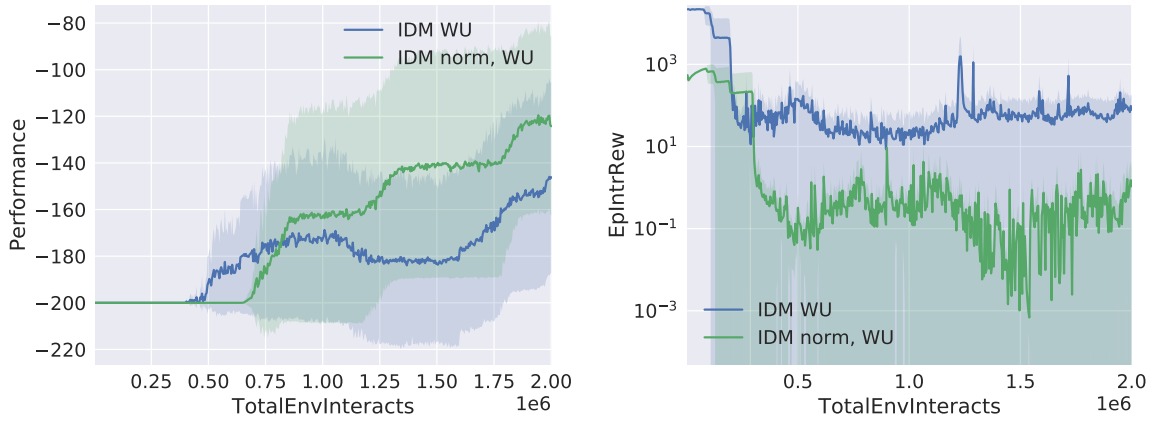
(b) Зависимость внутренней награды от числа взаимодействий со средой.

Рис. 3: Обучение агента с моделью прямой динамики.

Рассмотрим теперь, как наличие нормализации влияет на процесс обучения. Как видно из Рис. 2a, наличие нормализации помогает ускорить процесс обучения, а также позволяет достичь лучшего результата. Такой эффект возникает из-за того, что масштаб внутренней награды (Рис. 2b) при отсутствие нормализации влияет на политику агента, который пытается максимизировать сумму внутренней и внешней награды. Это наблюдение подтверждает предположение о том, что хорошая внутренняя награда должна убывать к нулю в процессе обучения.

Как видно на Рис. 3a, нормализация помогает более стабильно обучать агента с моделью прямой динамики. Из Рис. 3b видно, что в отличие от внутренней награды в ICM, внутренняя награды для модели прямой динамики без нормализации слишком быстро уходит в ноль, что в некоторых случаях приводит к тому, что агенту не удается получить положительную внешнюю награду во время взаимодействия.

Из Рис. 4 следует, что агент с моделью обратной динамики обучается хуже. Из Рис. 4a видно, что во многих запусках агент не получает положительной внешней награды на протяжение всего взаимодействия со средой. Как видно на Рис. 4b, внутренняя награда без нормализации по масштабу сопоставима с внешней, что приводит



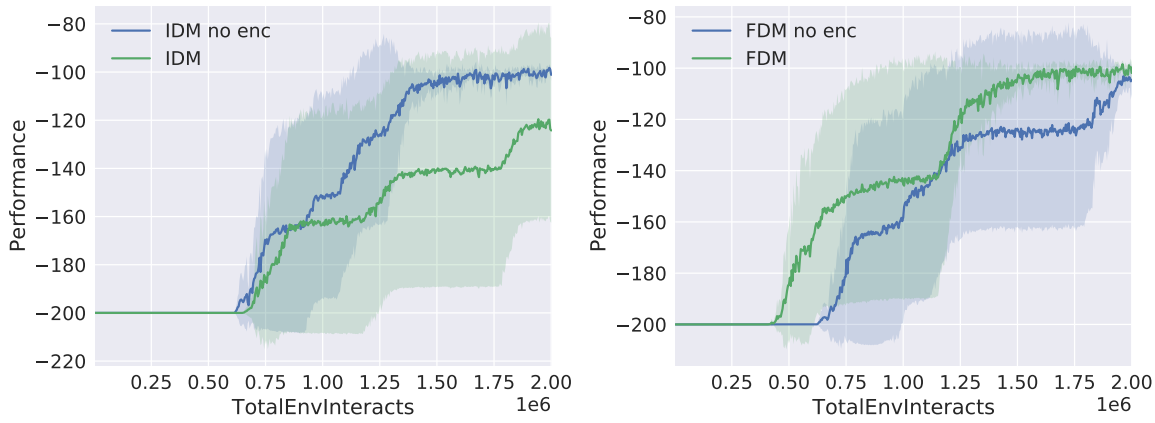
(a) Зависимость внешней награды от числа взаимодействий со средой.

(b) Зависимость внутренней награды от числа взаимодействий со средой.

Рис. 4: Обучение агента с моделью обратной динамики.

к тому, что агент предпочитает посещение как можно большего количества состояний быстрому завершению эпизода. Также на Рис. 2b видно, что внутренняя награда не убывает в процессе обучения модели обратной динамики, что подтверждает предположение, что ошибка предсказания модели обратной динамики плохо подходит для внутренней награды.

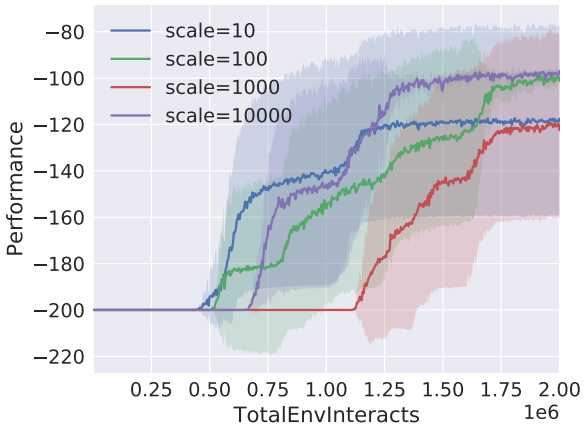
Итак, дальнейшие эксперименты будут проведены с прогревом, а также нормализацией.



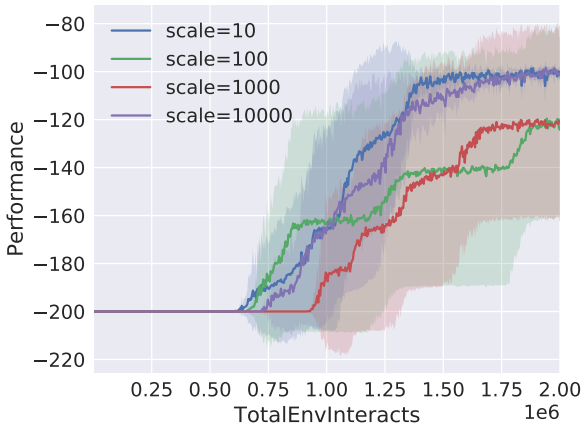
(a) Модель прямой динамики с энкодером и без.

(b) Модель обратной динамики с энкодером и без.

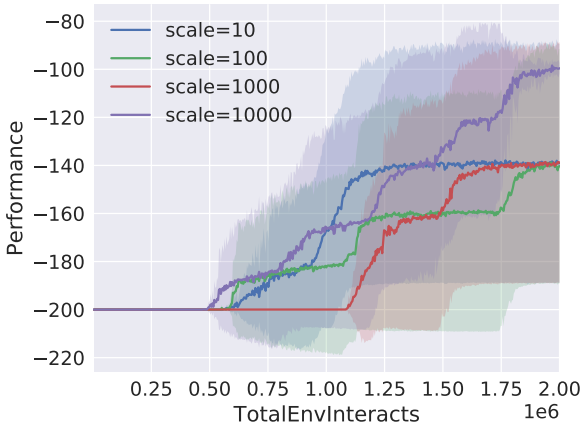
Рис. 5: Сравнение моделей прямой и обратной динамики.



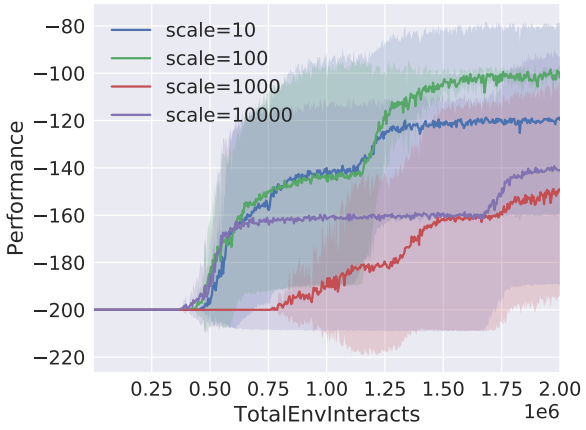
(a) Модель обратной динамики, одна V-функция.



(b) Модель обратной динамики, две V-функции.



(c) Модель прямой динамики, одна V-функция.



(d) Модель прямой динамики, две V-функции.

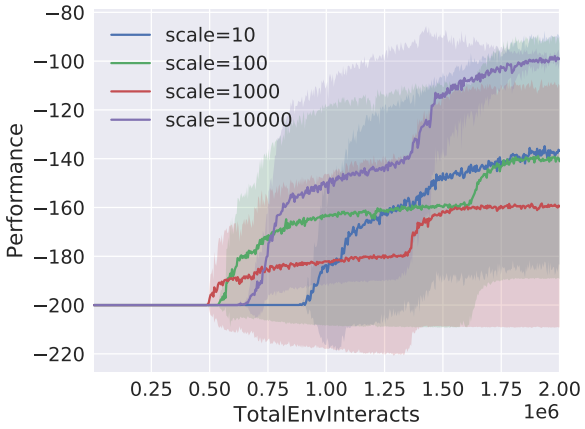
Рис. 6: Влияние раздельных V-функций на модели прямой и обратной динамики.

3.2 Выразительность модели

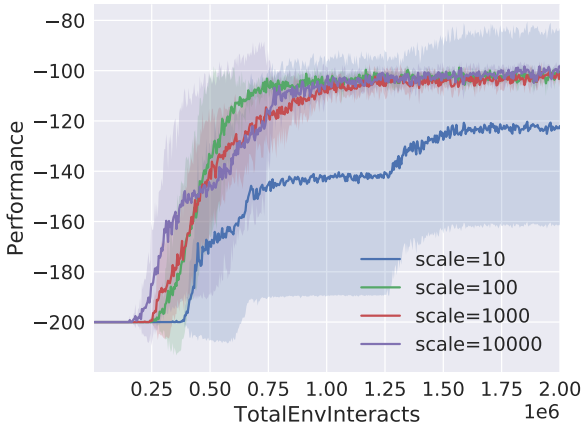
В связи с тем, что пространство состояний в среде очень простое, энкодер для моделей прямой и обратной динамики может рассматриваться как средство увеличения выразительности модели внутренней мотивации. Целью эксперимента является исследование влияния выразительности модели на обучение агента. Во всех методах в данном эксперименте используются раздельные V-функции.

Как видно из Рис. 5, наличие энкодера ускоряет обучение агента с моделью прямой динамики, однако ухудшает процесс обучения агента с моделью обратной ди-

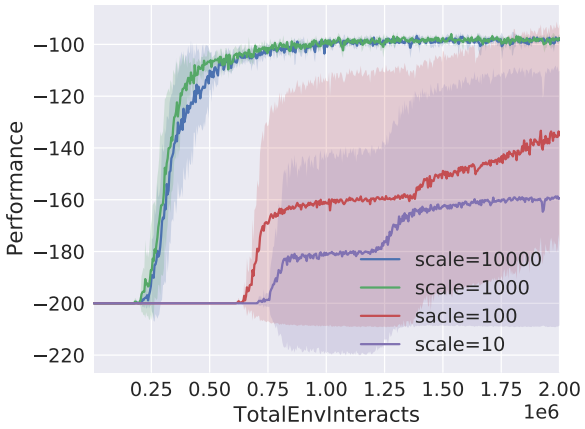
намики, что может быть вызвано нестационарностью эмбеддингов. Однако, в силу устройства метода ICM, наличие энкодера, обучаемого с помощью функции потерь обратной модели динамики, – единственное отличие от модели прямой динамики. Поэтому все дальнейшие эксперименты будут проведены с энкодером.



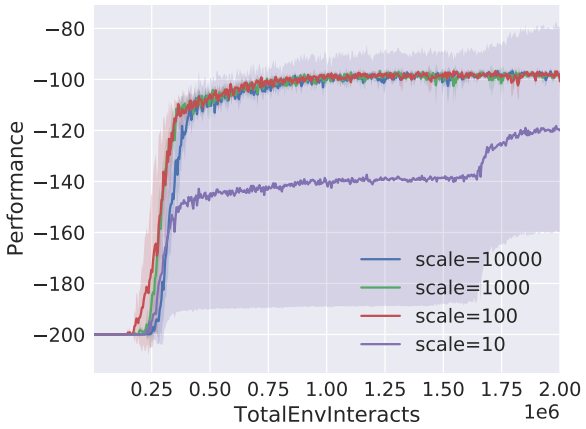
(a) ICM, одна V-функция.



(b) ICM, две V-функции.



(c) RND, одна V-функция.



(d) RND, две V-функции.

Рис. 7: Влияние раздельных V-функций на ICM и RND.

3.3 Раздельные V-функции

Эксперимент, описанный в разделе 3.1, позволяет утверждать, что масштаб внутренней награды существенно влияет на процесс обучения. Масштаб также влияет на значения оценочной функции V. В силу того, что внутренняя и внешняя награда принципиально различаются, возникает идея разделить также и V-функции. Такое разделение может помочь сделать метод инвариантным к масштабу внешней награ-

ды, а также сделать обучение более стабильным. Более того, отдельная V-функция для внешней награде может помочь агенту сфокусироваться на оптимизации внешней награды, когда внутренняя начнет затухать. В данном эксперименте будет рассмотрено влияние разделения V-функций на процесс обучения агента с различными методами внутренней мотивации.

Как видно из Рис. 6, введение раздельных V-функций ускоряет обучение агента с моделью прямой динамики. При этом для модели обратной динамики видимого преимущества не наблюдается.

Из Рис. 7 можно сделать вывод, что раздельные V-функции помогают существенно улучшить стабильность обучения агента с ICM и RND. Все последующие эксперименты будут проведены с раздельными V-функциями.

3.4 Сравнение лучших конфигураций

В данном эксперименте будет проведено сравнение лучших из найденных конфигурация для моделей прямой и обратной динамики, ICM и RND.

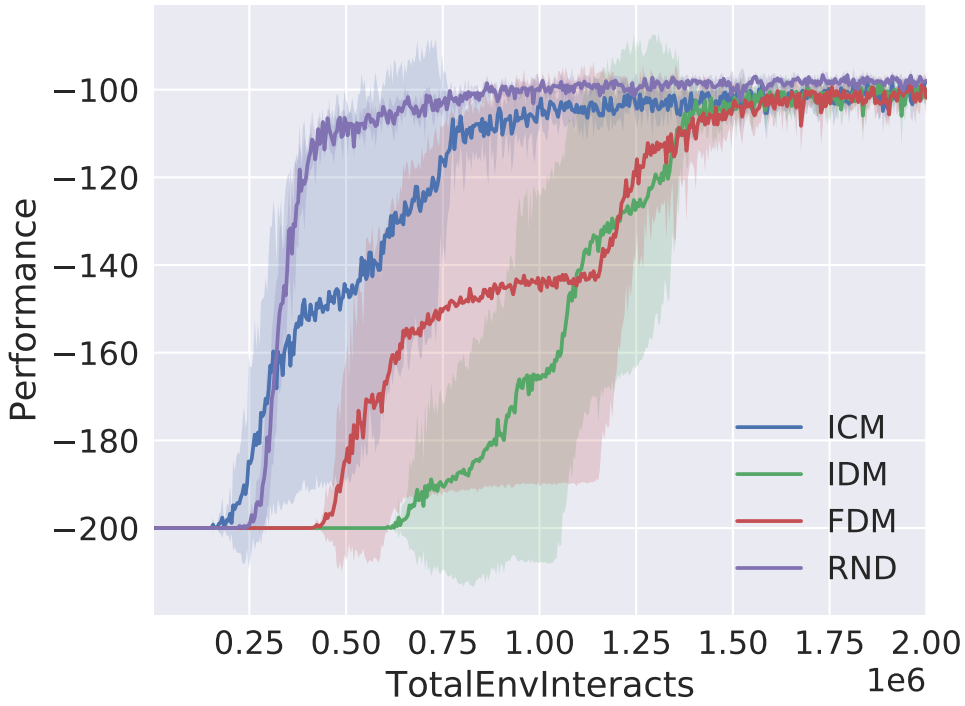
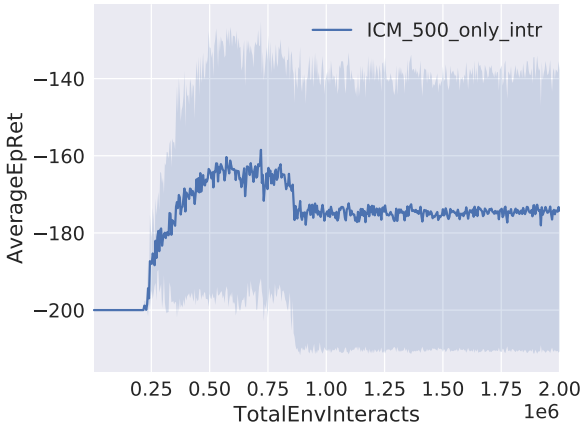


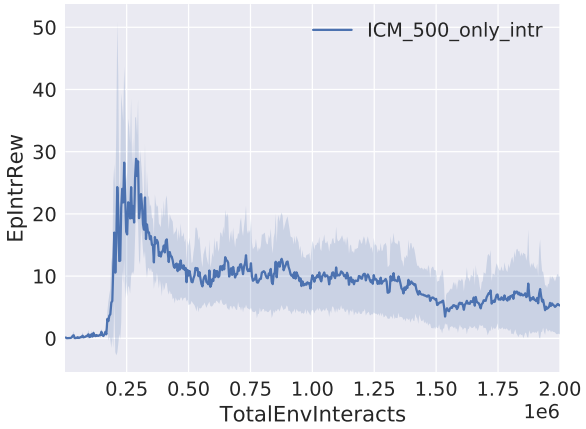
Рис. 8: Сравнение лучших конфигураций рассмотренных методов.

Как следует из Рис. 8, агент с RND обучается стабильнее всего и достигает наилучшего результата. В силу отсутствия источников стохастичности в среде, ключевым фактором для метода внутренней мотивации становится стационарность откликов, присущая методу RND. Более того, RND прост в реализации, а также обладает меньшим числом обучаемых параметров, что делает его предпочтительным методом введения внутренней мотивации в данной среде.

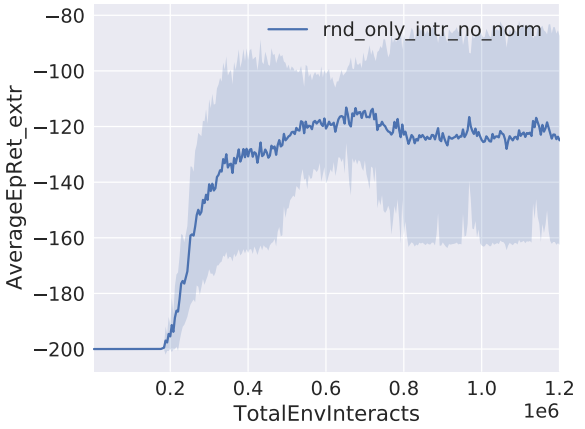
3.5 Отсутствие внешней награды



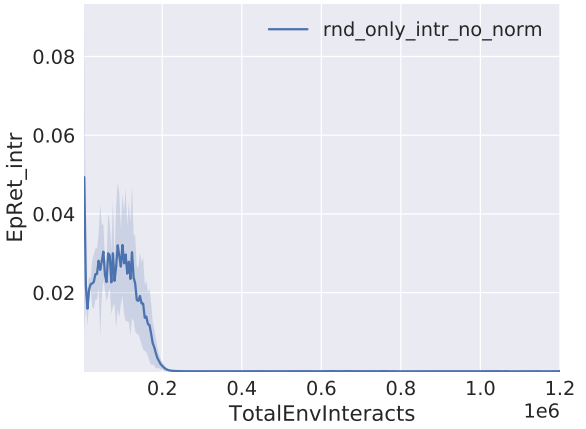
(a) ICM, зависимость внешней награды от числа взаимодействий.



(b) ICM, зависимость внутренней награды от числа взаимодействий.



(c) RND, зависимость внешней награды от числа взаимодействий.



(d) RND, зависимость внутренней награды от числа взаимодействий.

Рис. 9: Обучение ICM и RND без внешней мотивации.

Эксперименты, в которых агент не получает никакой награды непосредственно от среды проводились в [8]. Как уже отмечалось ранее, хорошая внутренняя награда должна убывать по мере того, как агент исследует среду. Целью эксперимента является исследование поведения внутренней награды по мере взаимодействия агента со средой. Рассмотрим обучение агентов с ICM и RND без внешней награды.

Как видно из Рис. 9b, Рис. 9d, внутренняя награда в RND и ICM затухает со временем. Однако, из Рис. 9d следует, что внутренняя награда в методе RND затухает гораздо быстрее, что приводит к лучшему результату для агента. Это наблюдение подтверждает предположение о том, что стационарность откликов играет важную роль для метода введения внутренней награды.

4 Заключение

В данной работе были рассмотрены различные методы введения внутренней мотивации для стимуляции агента к исследованию среды в обучении с подкреплением. Было приведено краткое описание методов, описаны их достоинства и недостатки. Были выделены ключевые гиперпараметры, влияющие на работу методов. Были проведены эксперименты в среде MountainCar-v0, показывающие важность правильного выбора гиперпараметров. В том числе было показано:

- Важность наличия прогрева.
- Важность раздельных V-функций для внутренней и внешней награды.
- Важность нормализации внутренней награды.
- Важность стационарности откликов для метода введения внутренней мотивации.

Также был сделан вывод относительно лучшего метода для введения внутренней мотивации в среде MountainCar-v0.

Список литературы

[1] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[2] Kai Arulkumaran, Antoine Cully, and Julian Togelius. Alphastar: An evolutionary computation perspective. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 314–315, 2019.

[3] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[4] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[6] Jürgen Schmidhuber. Curious model-building control systems. In *Proc. international joint conference on neural networks*, pages 1458–1463, 1991.

[7] Nick Haber, Damian Mrowca, Stephanie Wang, Li F Fei-Fei, and Daniel L Yamins. Learning to play with intrinsically-motivated, self-aware agents. In *Advances in Neural Information Processing Systems*, pages 8388–8399, 2018.

[8] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[10] Joshua Achiam. Openai spinning up. *GitHub, GitHub repository*, 2018.

A Команды для повторения экспериментов

A.1 Влияние прогрева и нормализации на процесс обучения

Листинг 1: Обучение агента с ICM, нормализацией и прогревом (Рис. 2).

```
python -m spinup.run ppo_icm --env MountainCar-v0 \
--exp_name ExpName --intr_rew_model ICM --epochs 500 \
--normalize_rewards True False --epochs_warmup 0 1 \
--two_v_heads True --scaling_factor 100 --seed 0 10 20 30 40 \
```

Листинг 2: Обучение агента с моделью прямой динамики и нормализацией (Рис. 3).

```
python -m spinup.run ppo_icm --env MountainCar-v0 \
--exp_name ExpName --intr_rew_model FDM --epochs 500 \
--normalize_rewards False True --epochs_warmup 1 \
--two_v_heads True --scaling_factor 100  --seed 0 10 20 30 40
```

Листинг 3: Обучение агента с моделью обратной динамики и нормализацией (Рис. 4).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model IDM --epochs 500
--normalize_rewards False True --epochs_warmup 1
--two_v_heads True --scaling_factor 100 --seed 0 10 20 30 40
```

A.2 Выразительность модели

Листинг 4: Обучение агента с моделью прямой динамики без энкодера (Рис. 5b).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model FDM_no_enc --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads True
--scaling_factor 100 --seed 0 10 20 30 40
```

Листинг 5: Обучение агента с моделью обратной динамики без энкодера (Рис. 5a).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model IDM_no_enc --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads True
--scaling_factor 100 --seed 0 10 20 30 40
```

A.3 Раздельные V-функции

Листинг 6: Обучение агента с моделью обратной динамики с одной V-функцией и разными scaling факторами. (Рис. 6a).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model IDM --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads False
--scaling_factor 10 100 1000 10000 --seed 0 10 20 30 40
```

Листинг 7: Обучение агента с моделью обратной динамики с двумя V-функциями и разными scaling факторами. (Рис. 6b).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model IDM --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads True
--scaling_factor 10 100 1000 10000 --seed 0 10 20 30 40
```

Листинг 8: Обучение агента с моделью прямой динамики с одной V-функцией и разными scaling факторами. (Рис. 6c).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model FDM --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads False
--scaling_factor 10 100 1000 10000 --seed 0 10 20 30 40
```

Листинг 9: Обучение агента с моделью прямой динамики с двумя V-функциями и разными scaling факторами. (Рис. 6d).

```
python -m spinup.run ppo_icm --env MountainCar-v0
```

```
--exp_name ExpName --intr_rew_model FDM --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads True
--scaling_factor 10 100 1000 10000 --seed 0 10 20 30 40
```

Листинг 10: Обучение агента с ICM с одной V-функцией и разными scaling факторами. (Рис. 7a).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model ICM --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads False
--scaling_factor 10 100 1000 10000 --seed 0 10 20 30 40
```

Листинг 11: Обучение агента с ICM с двумя V-функциями и разными scaling факторами. (Рис. 7b).

```
python -m spinup.run ppo_icm --env MountainCar-v0 \
--exp_name ExpName --intr_rew_model ICM --epochs 500 \
--normalize_rewards  True --epochs_warmup 1 --two_v_heads True \
--scaling_factor 10 100 1000 10000 --seed 0 10 20 30 40
```

Листинг 12: Обучение агента с RND с одной V-функцией и разными scaling факторами. (Рис. 7c).

```
python -m spinup.run ppo_rnd --env MountainCar-v0  \
--exp_name ExpName --epochs 500 --single_head True \
--scale_reward 10 100 1000 10000 --seed 0 10 20 30 40
```

Листинг 13: Обучение агента с RND с двумя V-функциями и разными scaling факторами. (Рис. 7d).

```
python -m spinup.run ppo_rnd --env MountainCar-v0 \
--exp_name ExpName --epochs 500 \
--scale_reward 10 100 1000 10000  --seed 0 10 20 30 40
```

A.4 Сравнение лучших конфигураций

Листинг 14: Обучение агента с ICM в лучшей конфигурации. (Рис. 8).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model ICM --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads True
--scaling_factor 10000 --seed 0 10 20 30 40
```

Листинг 15: Обучение агента с моделью прямой динамики в лучшей конфигурации. (Рис. 8).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model FDM --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads True
--scaling_factor 100 --seed 0 10 20 30 40
```

Листинг 16: Обучение агента с моделью обратной динамики в лучшей конфигурации. (Рис. 8).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model IDM --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads True
--scaling_factor 10 --seed 0 10 20 30 40
```

Листинг 17: Обучение агента с RND в лучшей конфигурации. (Рис. 8).

```
python -m spinup.run ppo_rnd --env MountainCar-v0 --epochs 500
--exp_name ExpName --scale_reward 1000 --seed 0 10 20 30 40
```

A.5 Отсутствие внешней награды

Листинг 18: Обучение агента с ICM без внешней награды. (Рис. 9a).

```
python -m spinup.run ppo_icm --env MountainCar-v0
--exp_name ExpName --intr_rew_model ICM --epochs 500
--normalize_rewards  True --epochs_warmup 1 --two_v_heads True
--scaling_factor 10 --seed 0 10 20 30 40 --only_intrinsic True
```

Листинг 19: Обучение агента с RND без внешней награды. (Рис. 9c).

```
python -m spinup.run ppo_rnd --env MountainCar-v0 --epochs 500
```

```
--exp_name ExpName --only_intr True --seed 0 10 20 30 40
```