Министерство науки и высшего образования Российской Федерации
Московский физико-технический институт
(национальный исследовательский университет)
Физтех-школа прикладной математики и информатики
Магистерская программа
«Методы и технологии искусственного интеллекта»

Пшеницын Артем Сергеевич

Мультиагентное обучение с подкреплением с непрерывными действиями

Обзор исследований

Научный руководитель:
доцент, д-р физ.-мат. наук
Панов Александр Игоревич

Москва, 2024

Содержание

1 Введение **3**

2 Постановка задачи **4**

2.1 Типы сценариев в мультиагентной постановке 4

3 Подходы к моделированию мультиагентных систем **5**

3.1 Централизованные и децентрализованные подходы 5

3.2 Parameter Sharing между агентами 6

3.3 Централизованное обучение и децентрализованное исполнение (ЦОДИ) 7

4 Методы ЦОДИ, основанные на функции ценности **7**

5 Методы ЦОДИ, основанные на стратегии **9**

6 Мультиагентные модификации PPO **11**

7 Мультиагентное планирование пути **13**

7.1 PRIMAL и PRIMAL_2 14

7.2 Follower 17

7.3 MAPF-GPT 18

7.4 Графовый трансформер как часть поиска 20

8 Контроль разнообразия поведения в мультиагентном обучении с подкреплением **21**

9 Выводы **23**

1 Введение

Мультиагентные системы представляют собой совокупность автономных агентов, каждый из которых способен принимать собственные решения и взаимодействовать в общей среде для достижения поставленных целей. Такие системы находят применение в различных областях, включая управление мобильными роботами на складах, координацию дронов для мониторинга объектов и управление торговыми процессами на виртуальных рынках.

Мультиагентное обучение с подкреплением направлено на разработку алгоритмов, позволяющих агентам обучаться взаимодействию в сложных средах. Отличительной особенностью является то, что в процессе обучения агенты также исследуют поведение других агентов. Со временем они осваивают навыки, необходимые для выполнения поставленных задач, и разрабатывают механизмы координации.

Обучение с подкреплением в мультиагентных системах основывается на принципах, при которых агенты изучают оптимальные стратегии на основе получаемых вознаграждений за свои действия, подобно тому, как это происходит в одноагентном обучении с подкреплением. Но в отличие от последних, мультиагентные подходы сталкиваются с рядом уникальных вызовов, включая изменчивость окружающей среды из-за действий других агентов, а также вычислительную сложность при масштабировании количества агентов.

Особое внимание в данной области уделяется применению методов обучения с подкреплением для задач мультиагентного планирования пути. В таких сценариях, как управление автономными транспортными средствами или координация дронов, необходимо находить оптимальные маршруты в сложных динамических средах с большим количеством агентов. Решение этих задач требует использования алгоритмов, способных справляться с высокой размерностью, динамичностью и частичной обозреваемостью среды.

Цель данного обзора - проанализировать современные подходы к мультиагентному обучению с подкреплением, включая методы координации и планирования пути. Рассмотреть как теоретические аспекты, так и практические применения.

2 Постановка задачи

Мультиагентное обучение с подкреплением можно рассматривать как обобщение классической задачи обучения с подкреплением, которое моделируется с использованием марковского процесса принятия решений в полной или частично обозреваемой форме.

Более строго, мультиагентный процесс принятия решений (полностью обозреваемый) задается как $M = \langle S, A, I, P, \vec{r}, \gamma \rangle$, где:

- S - пространство состояний,
- A - пространство действий для каждого агента,
- $I = \{1, \ldots, n\}$ - агенты,

  ! В каждый момент времени агент $i \in I$ совершает действие $a^i \in A$. Совокупность действий всех агентов образует объединенное действие $\vec{a} \in A^n$
- $P = P(s'|s, \vec{a})$ - динамика среды, функция вероятности переходов между состояниями,
- $\vec{r} = \vec{r}(s, \vec{a})$ - функция награды для каждого агента, возможно, из одинаковых элементов
- γ - фактор дисконтирования для дисконтированной кумулятивной награды $\vec{R} = \sum\limits_{t \geqslant 0} \gamma^t \vec{r}(s_t, \vec{a_t})$

Цель каждого агента - определить стратегию $\pi^i(a^i|s)$, которая максимизирует его ожидаемую дисконтированную кумулятивную награду с учетом действий других агентов.

Помимо описанной формализации, существуют и альтернативные постановки задачи мультиагентного обучения с подкреплением, такие как поиск пути в графе состояний [24] или марковские игры [15].

2.1 Типы сценариев в мультиагентной постановке

В связи с тем, что несколько агентов взаимодействуют в общей среде, можно выделить различные типы сценариев и/или сред, определяющих функцию награды для агентов:

- **Соревновательные среды**: награды агентов противоположны, и они действуют в условиях конкуренции.
- **Кооперативные среды**: агенты получают общую награду, что предполагает

совместные действия для достижения общей цели.

- **Смешанные кооперативно-соревновательные среды**: награда включает как и кооперативные, так и соревновательные компоненты, требуя от агентов сложных стратегий

3 Подходы к моделированию мультиагентных систем

Существуют различные подходы к моделированию мультиагентных систем, среди них можно выделить следующие:

- **Подходы на основе правил**

  Эти методы опираются на заранее определенные правила поведения агентов. Они часто используют эвристики или фиксированные сценарии, обеспечивая быструю работу и интерпретируемость, но ограничивая гибкость и адаптивность [5].

- **Графовые подходы**

  Задачи моделируются в виде графовых структур, где узлы представляют состояния, а рёбра — возможные переходы. Алгоритмы, такие как A^* и его модификации, позволяют находить оптимальные пути, но имеют ограничения в динамичных и частично наблюдаемых средах.

- **Обучаемые подходы**

  Включают использование методов машинного обучения, в частности, обучения с подкреплением. Эти подходы позволяют агентам адаптироваться к сложным и изменяющимся средам.

Ввиду того, что настоящая работа посвящена обучению с подкреплением, в последующих разделах внимание будет сосредоточено на этом подходе.

3.1 Централизованные и децентрализованные подходы

Одной из ключевых классификаций в мультиагентных системах является разделение на централизованные и децентрализованные методы:

- **Централизованные подходы**

  В централизованных системах решения принимаются единой управляющей сущностью, обладающей полной информацией о состоянии всех агентов.

Это позволяет находить глобально оптимальные стратегии, но требует значительных вычислительных ресурсов и плохо масштабируется при увеличении числа агентов.

- **Децентрализованные подходы**

  В децентрализованных системах каждый агент действует на основе локальной информации и собственной стратегии. Такие системы более гибкие, лучше масштабируются и сохраняют работоспособность в условиях частичного наблюдения.

3.2 Parameter Sharing между агентами

Другая классификация подходов в мультиагентных системах — использование общего набора параметров (parameter sharing). В этом подходе агенты используют общую модель (например, нейронную сеть) для определения своих действий, что имеет следующие особенности:

- **Ускорение обучения** благодаря уменьшению количества обучаемых параметров, что ведёт к снижению вычислительных затрат на обучение, что особенно важно при **масштабировании** по количеству агентов.
- **Проблема симметрии**: одинаковая стратегия у всех агентов может привести к неразрешимости системы в симметричных ситуациях, например, когда одному агенту нужно уступить путь другому, но они находятся в идентичных состояниях.

Для алгоритмов семейства Actor-Critic parameter sharing может быть применён как к части actor, так и к части critic, делая функцию ценности общей для агентов или раздельной.

Parameter sharing часто используется в сочетании с децентрализованными подходами, позволяя агентам действовать индивидуально, рассматривая других агентов как часть среды. Это приводит к тому, что среда становится динамической и стохастической, что объясняет успешность современных методов обучения с подкреплением на основе моделей среды для мультиагентных систем [25, 15].

3.3 Централизованное обучение и децентрализованное исполнение (ЦОДИ)

Основная идея этого подхода, представленного в 2016 году [1], заключается в том, что агенты получают доступ к дополнительной информации во время обучения, такой как наблюдения других агентов, вознаграждения, градиенты и параметры модели. В процессе исполнения агенты применяют свою стратегию, используя только локальные наблюдения. Централизованное обучение и децентрализованное исполнение (ЦОДИ) позволяет снизить нестационарность и частичную наблюдаемость, поскольку доступ к дополнительной информации во время обучения стабилизирует процесс даже при изменении стратегий других агентов. Методы ЦОДИ делятся на подходы, основанные на функции ценности, и подходы, основанные на стратегии.

4 Методы ЦОДИ, основанные на функции ценности

Методы, основанные на функции ценности, направлены на разложение централизованно изучаемой функции ценности на функции ценности отдельных агентов, которые используются для децентрализованного исполнения. Среди них наиболее известны сети декомпозиции функции ценности, такие как VDN, QMIX и их модификации [10, 4, 7, 13].

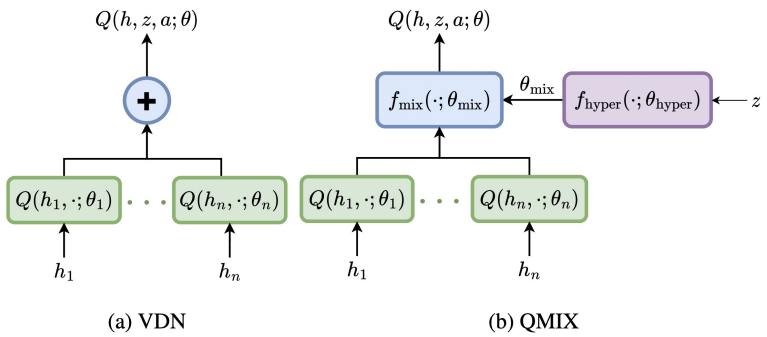

Рис. 1: Схема методов, основанных на функции ценности. Адаптировано из [21].

В методе VDN [4] предполагается, что централизованная функция ценности ко-

манды является суммой функций ценности действий отдельных агентов:

$$Q(\vec{s}, \vec{a}) = \sum_i Q_i(s_i, a_i)$$

Функция ценности команды линеаризуется в сумму индивидуальных функций ценности агентов (Рис.1 (a)), что обеспечивает хорошую сходимость в кооперативных сценариях и масштабируемость при тестировании в децентрализованном режиме.

QMIX [10] улучшает VDN, рассматривая функцию ценности всей команды как нелинейную комбинацию отдельных функций с монотонным ограничением (Рис.1 (b)). Однако такое ограничение может снижать производительность агентов в условиях сложной координации.

Эксперименты показывают, что даже в простых матричных играх VDN не может сходиться к оптимальным стратегиям, в отличие от QMIX (Рис.2).

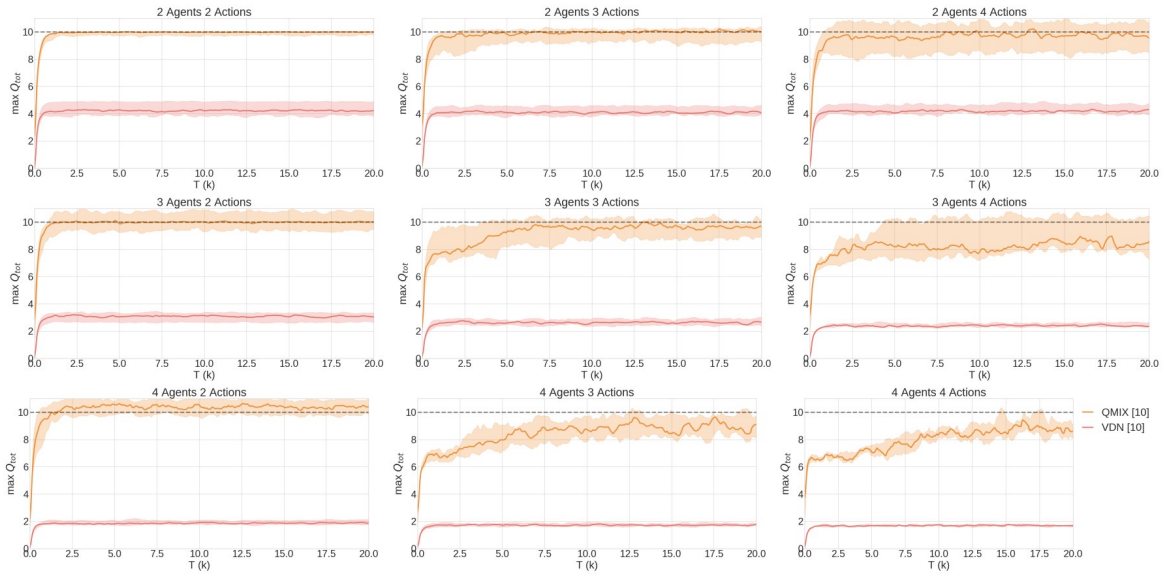

Рис. 2: Сравнение QMIX и VDN на 2 стадийных матричных играх. Адаптировано из [10].

Авторы QMIX также демонстрируют превосходство над VDN в сложных средах, таких как игра StarCraft (Рис.3).

Для дальнейшего улучшения факторизации функции ценности была предложена модификация - QTRAN [7], которая устраняет ограничения монотонности и аддитивности. Однако необходимость регуляризации увеличивает вычислительные затраты и может снижать производительность в сложных многоагентных средах.

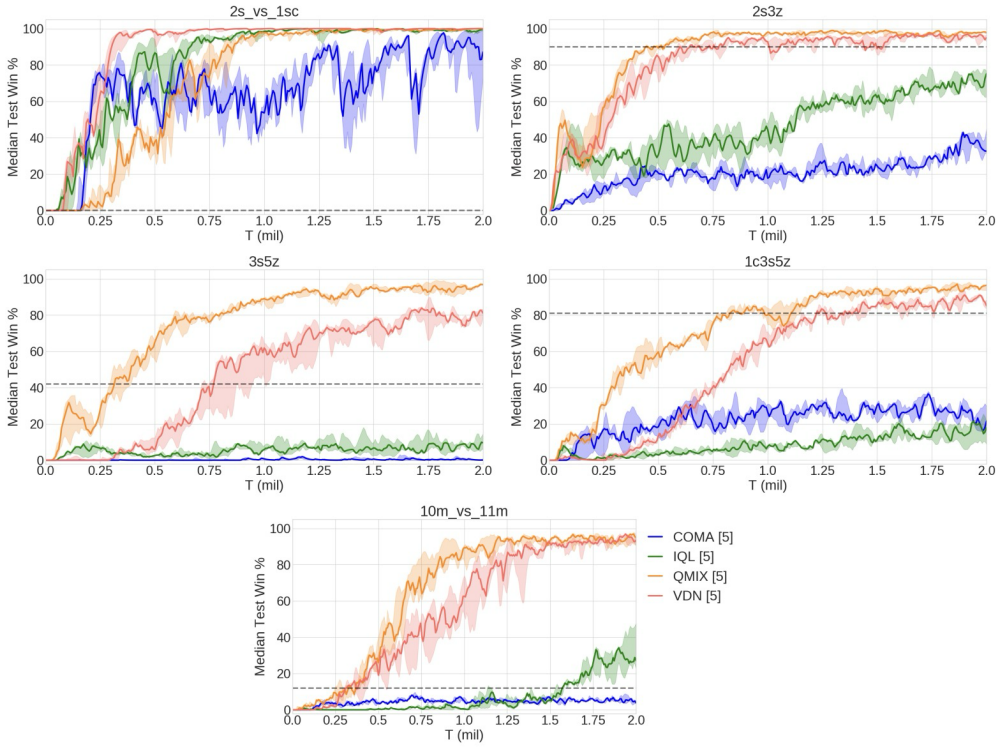

Рис. 3: Сравнение QMIX с аналогами в среде StarCraft. Адаптировано из [10].

5 Методы ЦОДИ, основанные на стратегии

Методы, основанные на стратегии, опираются на архитектуры актор-критик, где централизованный критик используется для обучения децентрализованных агентов. Примером такого подхода является COMA [2], который применяет централизованного критика для аппроксимации Q-функции и децентрализованных агентов для оптимизации их стратегий. Централизованный критик получает доступ к совместным действиям и полной информации о состоянии, тогда как стратегия каждого агента зависит только от его локальных наблюдений и исторических действий (Рис. 4).

Мультиагентная версия алгоритма DDPG — MADDPG [11] — расширяет архитектуру актор-критик, предоставляя критику доступ к дополнительной информации во время обучения, тогда как актор оперирует только локальными наблюдениями. В отличие от COMA, где используется единый централизованный критик для всех агентов, в MADDPG каждый агент имеет своего централизованного критика, что позволяет моделировать сложные конкурентные среды с различными функциями вознаграждения.

Основное различие между COMA и MADDPG заключается в использовании

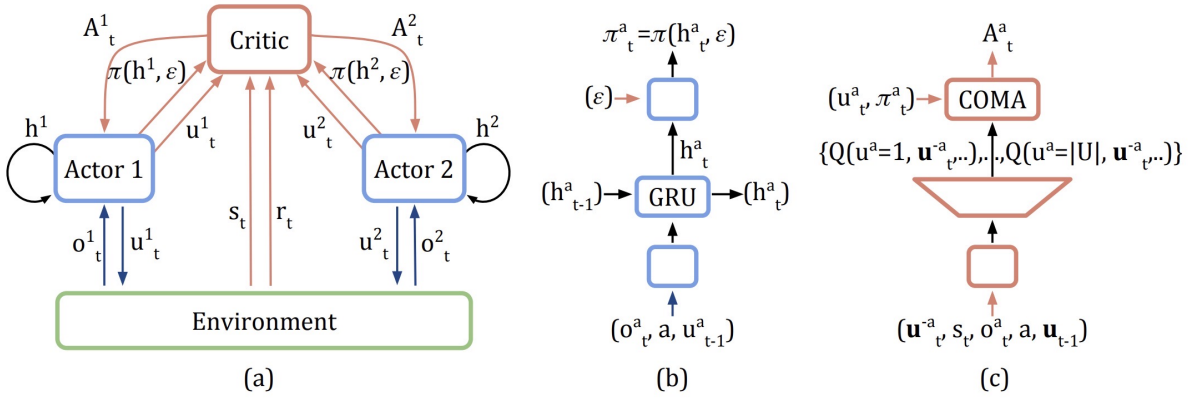

Рис. 4: (a) Схема модели COMA. (b) и (c) Архитектуры децентрализованных акторов и централизованного критика, соответсвенно. Красные стрелки и компоненты используются только во время обучения. Адаптировано из [2].

parameter sharing для критика в COMA, что снижает вычислительные затраты, но ограничивает применение в конкурентных сценариях (Рис. 5).

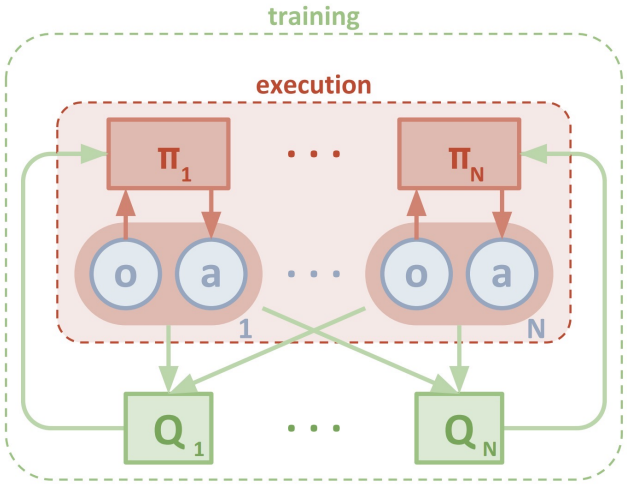

Рис. 5: Схема модели MADDPG. В отличие от COMA для каждого агента используется собственный централизованный критик. Адаптировано из [11].

На основе MADDPG было предложено несколько модификаций, включая R-MADDPG [12] и M3DDPG [8], которые улучшают устойчивость обучения и справляются с более сложными многоагентными сценариями.

6 Мультиагентные модификации PPO

Proximal Policy Optimization (PPO), представленный в 2017 году [3], стал основой для множества модификаций, включая те, которые ориентированы на мультиагентные системы:

- **Centralized PPO** [18]

  Полностью централизованная версия PPO, рассматривающая мультиагентную задачу как задачу с одним супер-агентом.

- **MAPPO** [17]

  Использует децентрализованного актора с централизованным критиком, при этом parameter sharing применяется к обоим.

- **IPPO** [9]

  Полностью децентрализованная версия PPO с использованием parameter sharing.

- **HetIPPO** [18]

  Вариант IPPO без parameter sharing, где каждый агент представлен отдельной нейронной сетью.

- **GPPO** [20]

  Включает графовые нейронные модули, которые обрабатывают закодированную информацию, а parameter sharing применяется ко всем модулям.

- **HetGPPO** [20]

  Гетерогенная версия GPPO, где для каждого агента используется собственная обучаемая модель, без parameter sharing.

Результаты IPPO показывают, что PPO в мультиагентных задачах превосходит QMIX, обученный в парадигме DRQN. Несмотря на то, что DRQN способен учитывать временные зависимости благодаря рекуррентным слоям, PPO, не имеющий рекуррентности, демонстрирует более высокие результаты (Рис.6).

Эксперименты MAPPO, представленные в [17], демонстрируют сопоставимые с IPPO результаты в различных средах, включая StarCraft (Рис.7).

Ранее работы в основном тестировались на StarCraft и простых сценариях MPE (мультиагентные среды с кооперативными и соревновательными задачами, представленными вместе с MADDPG [11]). В работе [18] представлен обширный набор мультиагентных сред VMAS, где сравниваются CPPO, IPPO, MAPPO и HetIPPO (Рис.8).

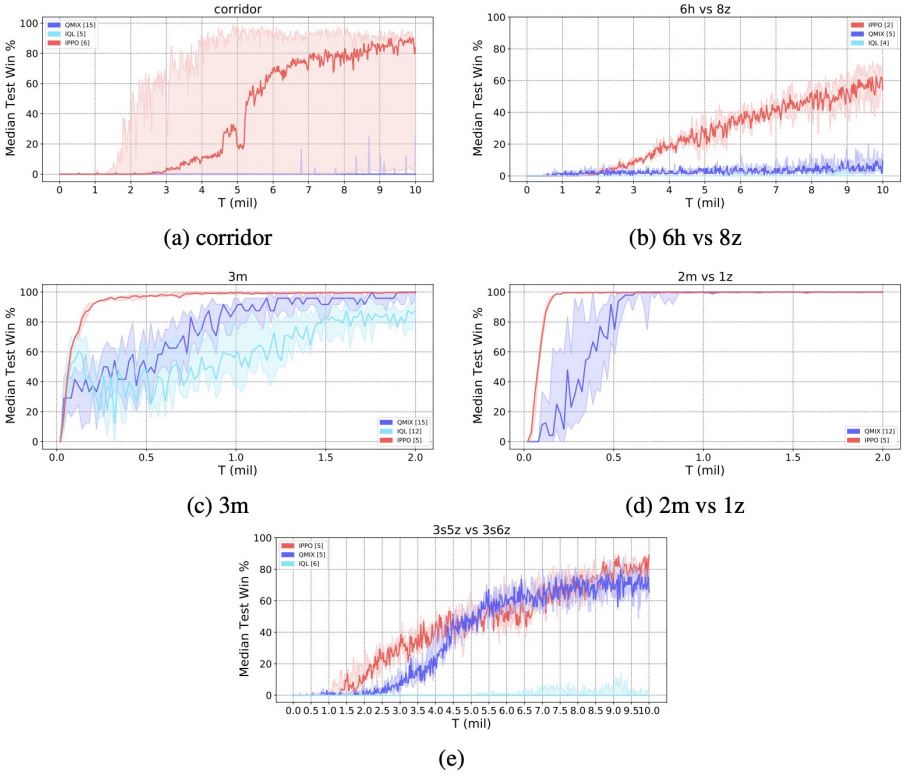

Рис. 6: Результаты экспериментов IPPO, сравнение с QMIX в среде StarCraft. Адаптировано из [9].

Map	MAPPO(FP)	MAPPO(AS)	IPPO	QMix
2m vs_1z	**100.0**(0.0)	**100.0(0.0)**	**100.0**(0.0)	**95.3**(5.2)
3m	**100.0**(0.0)	**100.0**(1.5)	**100.0**(0.0)	96.9(1.3)
2svs1sc	**100.0**(0.0)	**100.0**(0.0)	**100.0**(1.5)	96.9(2.9)
2s3z	**100.0**(0.7)	**100.0**(1.5)	**100.0**(0.0)	95.3(2.5)
3svs3z	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)	**96.9**(12.5)
3svs4z	**100.0**(1.3)	**98.4**(1.6)	**99.2**(1.5)	**97.7**(1.7)
so many baneling	**100.0**(0.0)	**100.0**(0.7)	**100.0**(1.5)	96.9(2.3)
8m	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.7)	97.7(1.9)
MMM	**96.9**(0.6)	93.8(1.5)	**96.9**(0.0)	**95.3**(2.5)
1c3s5z	**100.0**(0.0)	96.9(2.6)	**100.0**(0.0)	96.1(1.7)
bane vs bane	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)
3svs5z	**100.0**(0.6)	**99.2**(1.4)	**100.0**(0.0)	**98.4**(2.4)
2cvs64zg	**100.0**(0.0)	**100.0**(0.0)	98.4(1.3)	92.2(4.0)
8mvs9m	**96.9**(0.6)	**96.9**(0.6)	**96.9**(0.7)	92.2(2.0)
25m	**100.0**(1.5)	**100.0**(4.0)	**100.0**(0.0)	85.9(7.1)
5mvs6m	**89.1**(2.5)	**88.3**(1.2)	**87.5**(2.3)	75.8(3.7)
3s5z	**96.9**(0.7)	**96.9**(1.9)	**96.9**(1.5)	88.3(2.9)
10mvs11m	**96.9**(4.8)	**96.9**(1.2)	**93.0**(7.4)	**95.3**(1.0)
MMM2	**90.6**(2.8)	**87.5**(5.1)	**86.7**(7.3)	**87.5**(2.6)
3s5zvs3s6z	**84.4**(34.0)	63.3(19.2)	**82.8**(19.1)	**82.8**(5.3)
27mvs30m	**93.8**(2.4)	85.9(3.8)	69.5(11.8)	39.1(9.8)
6hvs8z	**88.3**(3.7)	**85.9**(30.9)	**84.4**(33.3)	9.4(2.0)
corridor	**100.0**(1.2)	**98.4**(0.8)	**98.4**(3.1)	84.4(2.5)

Рис. 7: Результаты экспериментов MAPPO, сравнение с IPPO и QMIX в среде StarCraft. Адаптировано из [17].

Результаты показывают, что IPPO превосходит MAPPO в некоторых кооперативных сценариях (Рис.8 (a)), но оба алгоритма не достигают оптимальных стратегий

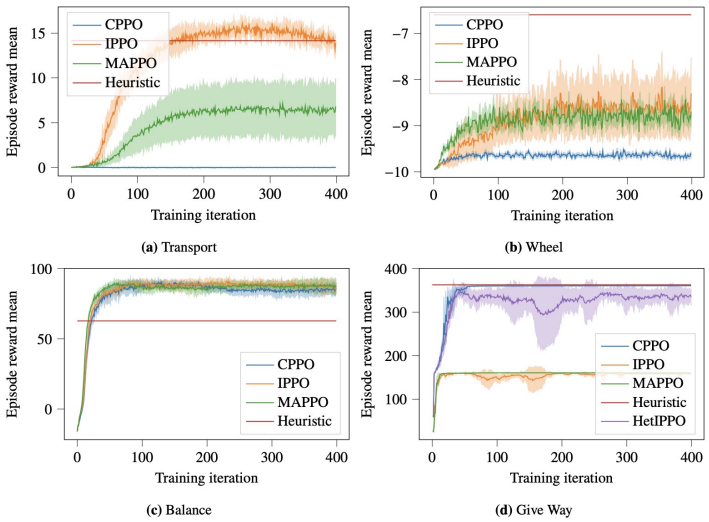

Рис. 8: Сравнение CPPO, IPPO, MAPPO и HetIPPO в средах VMAS. (a) Transport: агенты кооперируются для перемещения груза. (b) Wheel: агенты вращают стержень с заданной скоростью. (c) Balance: груз поднимается против гравитации. (d) Give Way: один агент уступает путь другому в туннеле. Адаптировано из [18].

в сложных задачах (Рис.8 (d)), в отличие от CPPO и HetIPPO.

7 Мультиагентное планирование пути

Другой ряд задач, решаемых с помощью мультиагентного обучения с подкреплением, — мультиагентное планирование пути. Эта область имеет прямое практическое приложение, например, в координации дронов, автономных транспортных средств или роботов на складе и производстве.

Специфика таких задач заключается в необходимости согласованного движения агентов в ограниченном пространстве при соблюдении определённых правил, таких как избегание столкновений, минимизация времени выполнения задач и оптимизация маршрутов. Решение этих задач требует учёта взаимодействий между агентами и их влияния на динамику среды.

В ряде работ исследуются подходы, объединяющие обучение с подкреплением с различными стратегиями координации агентов, например [6, 14, 23]

Для обучения и тестирования применяются разнообразные симулированные среды, включая лабиринты, склады и производственные линии. Обычно выделяют три основных типа задач:

- **Одноразовый поиск пути:**

получение оптимальных маршрутов для агентов к заданным целям, после чего выполнение задачи завершается.

- **Непрерывный поиск пути:**

  после достижения текущей цели агенту немедленно назначается новая цель, что требует постоянного онлайн-обновления маршрута.

- **Задачи доставки:**

  агенты должны сначала достичь одной точки для загрузки груза, а затем доставить его в конечную точку назначения.

Таким образом, мультиагентное планирование пути остаётся активно развивающейся областью, объединяющей подходы из теории графов, классических методов поиска и современных методов обучения с подкреплением для решения сложных координационных задач.

7.1 PRIMAL и PRIMAL$_2$

Работы [6, 14] представили первый успешный подход к адаптации методов обучения с подкреплением для задач мультиагентного планирования пути. Ключевая особенность предложенного метода заключается в интеграции классического планирования с обучением с подкреплением и имитационным обучением.

Обе модели являются полностью децентрализованными с использованием parameter sharing. Основное различие между методами состоит в том, что PRIMAL был разработан для одноразового поиска пути, тогда как PRIMAL$_2$ предназначен для непрерывного планирования, что определяет различия в наблюдениях агентов в обоих случаях.

В PRIMAL агенты получают наблюдения, включающие информацию о расположении препятствий, положении других агентов в поле зрения и целях как собственных, так и соседних агентов (Рис.9).

В PRIMAL$_2$ к этим наблюдениям добавляются пути, рассчитанные с использованием классического планировщика A^*, аналогичные пути для соседних агентов, а также расстояния между началом и концом узких коридоров на картах с ограниченными проходами (Рис.10).

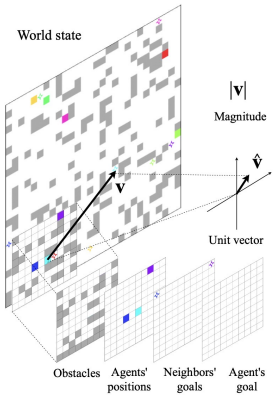

Рис. 9: Наблюдение каждого агента в модели PRIMAL. Адаптировано из [6].

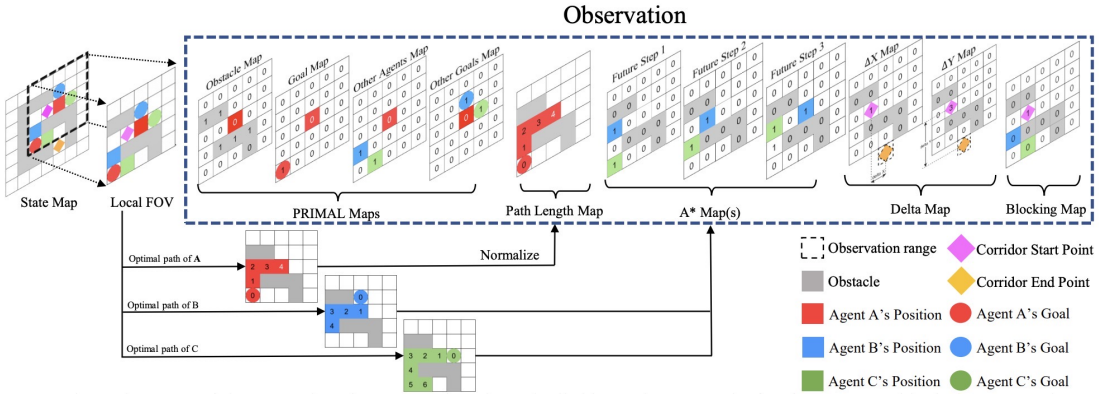

Рис. 10: Наблюдение каждого агента в модели PRIMAL$_2$. Адаптировано из [14].

Архитектура модели включает VGG-энкодер для кодирования состояний агентов, отдельный энкодер для координат цели, а также LSTM-блок, отвечающий за учёт истории действий агентов. Модель содержит три выходные головы: актор, критик и блокировщик действий (Рис.11).

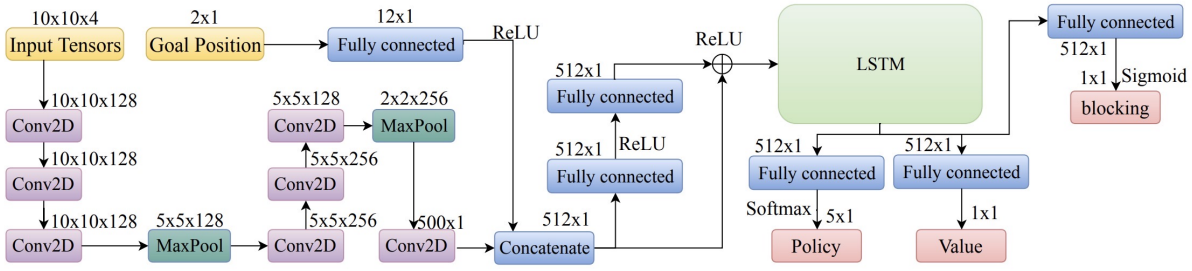

Рис. 11: Архитектура PRIMAL и PRIMAL$_2$. Адаптировано из [6].

Блокировщик предназначен для классификации допустимых действий, что ускоряет сходимость модели. В качестве алгоритма обучения использовался A3C в сочетании с имитационным обучением на путях, сгенерированных классическим плани-

ровщиком. Несмотря на то, что авторы пришли к выводу о незначительном влиянии имитационного обучения на сходимость, во второй версии метода оно было сохранено.

Авторы протестировали метод на масштабируемость при большом количестве агентов (до тысячи) и сравнили его с классическими методами (Рис.12, 13).

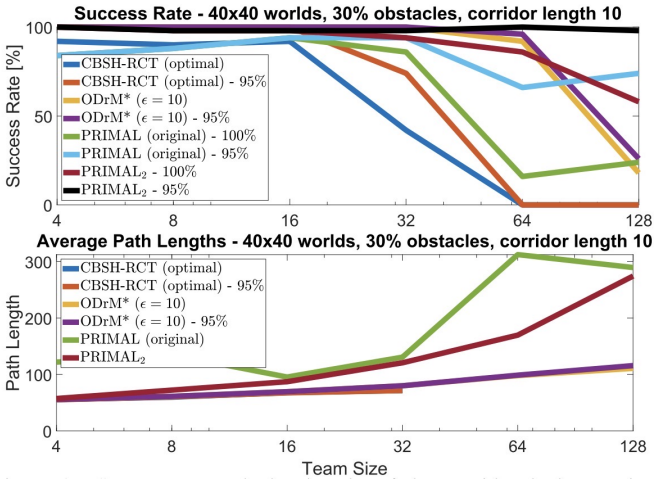

Рис. 12: Сравнение PRIMAL и PRIMAL_2 в задаче одноразового поиска пути. Адаптировано из [14].

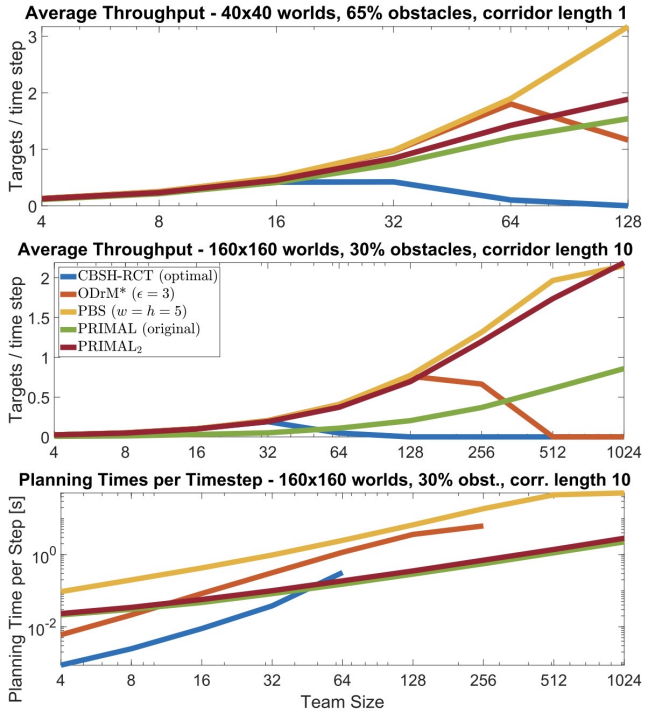

Рис. 13: Сравнение PRIMAL и PRIMAL_2 в задаче непрерывного поиска пути. Адаптировано из [14].

7.2 Follower

Работа [23] также представляет гибридный метод, объединяющий эвристическое планирование и обучение с подкреплением. Основная идея заключается в использовании планировщика для генерации начального пути и применении обучаемой модели для корректировки маршрута с учетом динамических препятствий и других агентов.

В данном подходе цели агентов не являются частью наблюдения и не подаются на вход нейронной сети. Вместо этого они используются только на этапе эвристического планирования маршрута. Для построения пути применяется алгоритм A^* с эвристически заданной функцией стоимости переходов, которая включает статическую и динамическую компоненты (Рис.14).

На следующем этапе построенные пути, препятствия и местоположение других агентов используются в качестве входных данных для нейронной сети. На основе этого наблюдения сеть формирует стратегию управления движением агента. В качестве алгоритма обучения применяется PPO, реализованный в полностью децентрализованной архитектуре с parameter sharing (Рис.14).

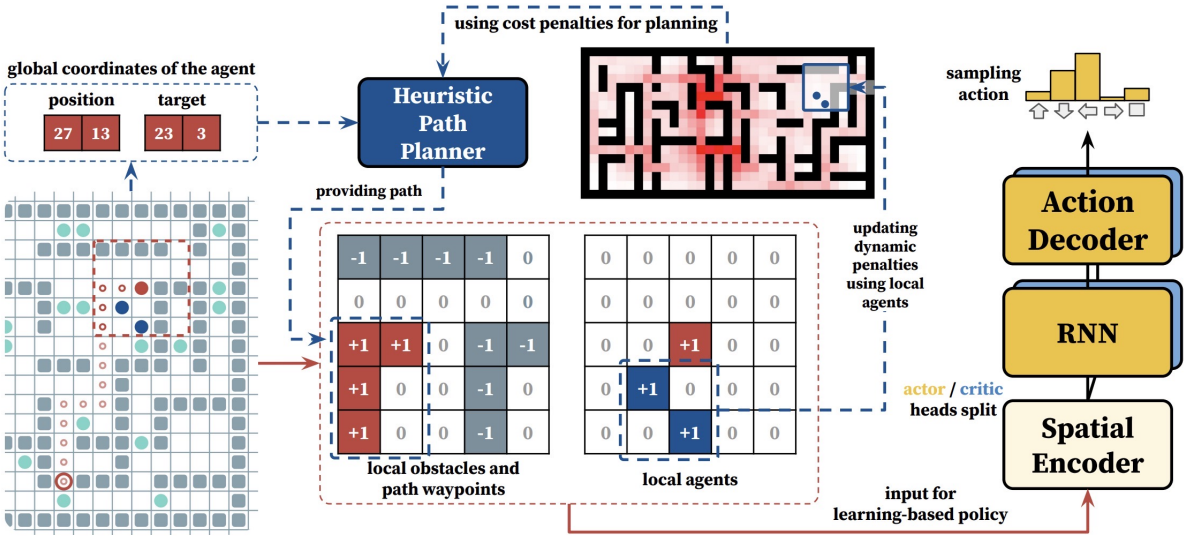

Рис. 14: Схема метода Follower. Адаптировано из [23].

Отличительной особенностью данного подхода является отсутствие этапа имитационного обучения. Несмотря на относительную простоту, метод демонстрирует результаты, превосходящие PRIMAL_2 (Рис.15).

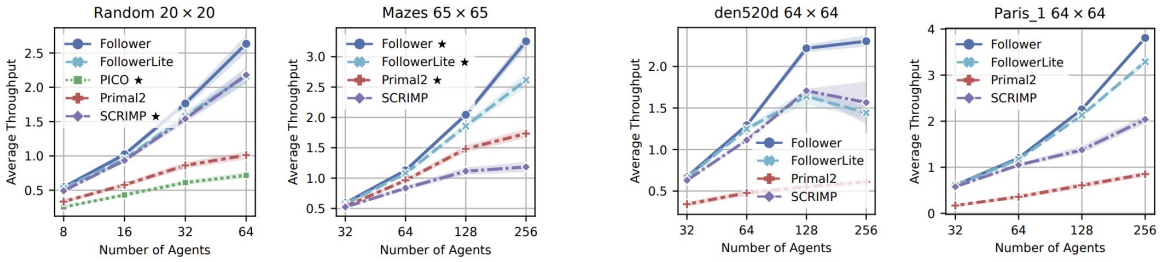

Рис. 15: Результаты экспериментов Follower, сравнение с PRIMAL$_2$. Адаптировано из [23].

7.3 MAPF-GPT

В работе [24] представлен принципиально новый подход к мультиагентному планированию пути, основанный на архитектуре трансформеров. Учитывая способность трансформеров эффективно масштабироваться и быстро сходиться, авторы предложили обучить фундаментальные модели для задач мультиагентного планирования пути с использованием исключительно трансформерной архитектуры и имитационного обучения.

В качестве архитектуры используется декодерный трансформер, обучаемый без каузального маскирования с применением кросс-энтропийной функции ошибки.

Ключевой особенностью метода является способ токенизации состояния агента. В качестве токенов в трансформер подается следующая информация:

- положение препятствий,
- эгоцентричная кумулятивная стоимость пути до каждой клетки в поле зрения,
- положение до 13 соседних агентов,
- история действий этих агентов.

Данный подход позволяет модели обобщать сложные взаимодействия между агентами и учитывать пространственные зависимости в среде (Рис.16).

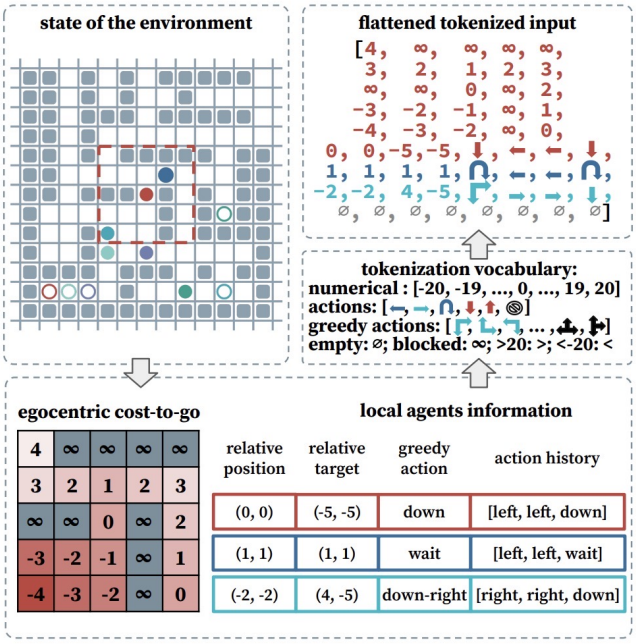

Рис. 16: Процесс токенизации MAPF-GPT. Адаптировано из [24].

Для обучения модели был создан датасет объёмом 1 миллиард наблюдений, собранных в симулированной среде POGEMA [16]. Экспертные траектории генерировались с использованием одного из современных методов — LaCAM, обеспечивающего высокое качество данных (Рис.17).

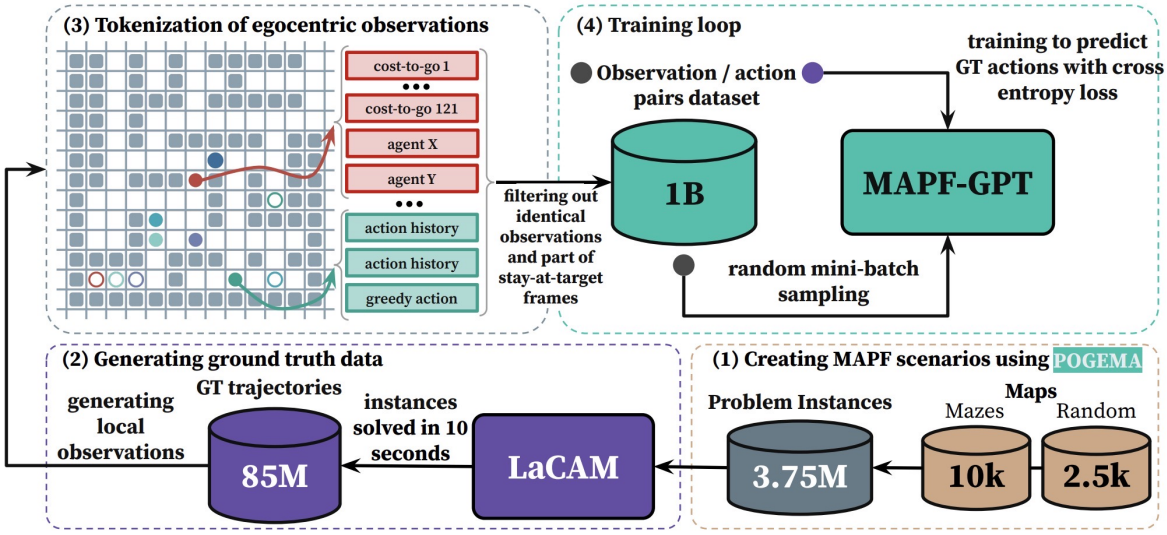

Рис. 17: Схема обучения MAPF-GPT. Адаптировано из [24].

В результате были обучены три модели различного размера с числом параметров 2, 6 и 85 миллионов соответственно. Все модели показали конкурентоспособные

результаты в задачах мультиагентного планирования пути (Рис.18).

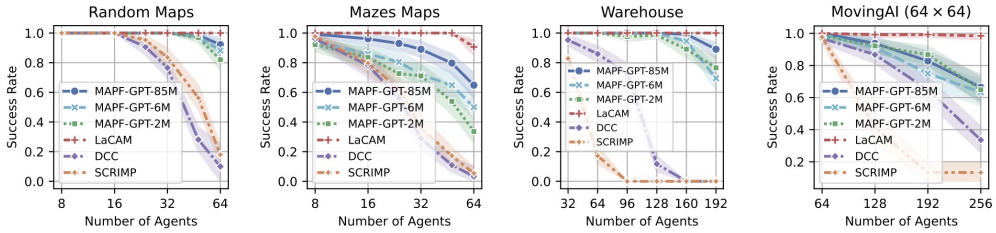

Рис. 18: Результаты MAPF-GPT. Адаптировано из [24].

7.4 Графовый трансформер как часть поиска

В предыдущих разделах были рассмотрены методы мультиагентного поиска пути в сеточных средах. В работе [19] предложен один из первых обучаемых методов для несеточных сред. В отличие от PRIMAL и MAPF-GPT, которые используют имитационное обучение на стратегиях, сгенерированных классическим планировщиком, или от Follower, применяющего эвристический планировщик для генерации начальных путей, в данном подходе модель-графовый трансформер используется в качестве нелинейной эвристики для классического метода.

Основой метода является один из наиболее популярных алгоритмов мультиагентного планирования пути — поиск на основе конфликтов (CBS), двухуровневый планировщик. Верхний уровень представляет собой дерево для разрешения конфликтов между агентами, а нижний генерирует локальные субоптимальные пути для агентов.

Предложенный метод состоит из нескольких этапов и способен предлагать пути для агентов в непрерывных средах, представляя собой один из интересных вариантов решения мультиагентных задач в непрерывных пространствах состояний и действий:

- Генерация случайных узлов графа в среде (Рис.19, слева),
- Построение связей в графе путём соединения k ближайших соседей с учётом отсутствия возможных столкновений,
- Запуск эвристического планировщика на графе с эвристикой, формируемой графовым трансформером.

Особенностью данного подхода также является способ тренировки трансформера для генерации эвристики. Обучение осуществляется с применением контрастивного обучения, где негативные примеры генерируются классическим поиском на основе

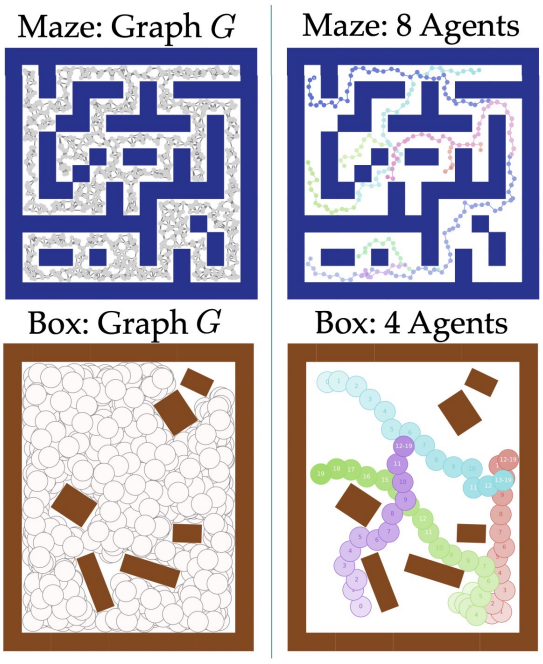

Рис. 19: Слева - сгенерированные узлы графа, справа - полученные решения. Адаптировано из [19].

конфликта (Рис.20).

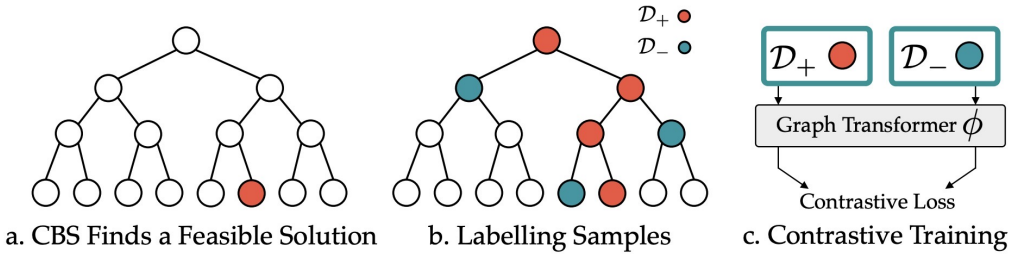

Рис. 20: Схема обучения трансформера для генерации эвристики. Адаптировано из [19].

8 Контроль разнообразия поведения в мультиагентном обучении с подкреплением

В работе [22] представлен метод управления разнообразием стратегий в мультиагентных системах с непрерывными действиями. В кооперативных сценариях часто требуется выполнение редких событий, что требует эффективного исследования среды и разнообразных действий агентов.

Обычно исследование среды стимулируется с помощью подходов, связанных с дизайном функции награды или ошибкой обучения модели. Однако авторы пред-

ложили новый подход, ориентированный специально на мультиагентные системы с непрерывными пространствами действий. Метод направлен на поддержку разнообразия стратегий, что повышает вероятность успешного кооперативного поведения агентов.

Для измерения разнообразия используется усреднённая метрика Вассерштейна, применённая к выходам акторов. Задача состоит в поддержании этой метрики на заданном уровне в процессе обучения. Архитектура акторов каждого агента включает две компоненты:

- **Совместная компонента (shared):** общая для всех агентов, что обеспечивает хорошую сходимость алгоритма.
- **Гетерогенная компонента (non-shared):** индивидуальная для каждого агента, что способствует появлению сложного кооперативного поведения.

Итоговый выход акторов определяется как сумма выходов этих двух компонентов, задающих параметры гауссовского распределения, из которого генерируются непрерывные действия:

$$\pi^i(o) = \pi_{shared}(o) + \pi^i_{non-shared}(o)$$

Для достижения целевого уровня разнообразия гетерогенная компонента масштабируется в зависимости от текущего значения метрики Вассерштейна. Авторы провели эксперименты в среде VMAS на различных сценариях с разными уровнями целевого разнообразия и алгоритмами, такими как MADDPG (Рис.21) и IPPO (Рис.22).

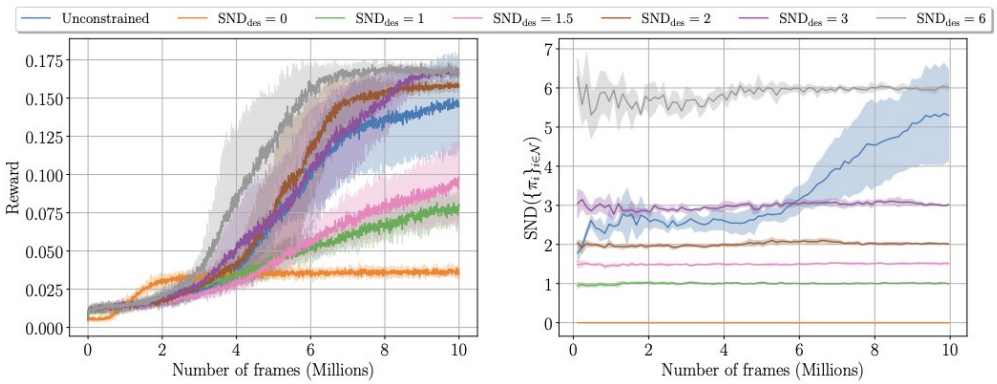

Рис. 21: Кривые обучения MADDPG с разными значениями уровня разнообразия поведений. Адаптировано из [22].

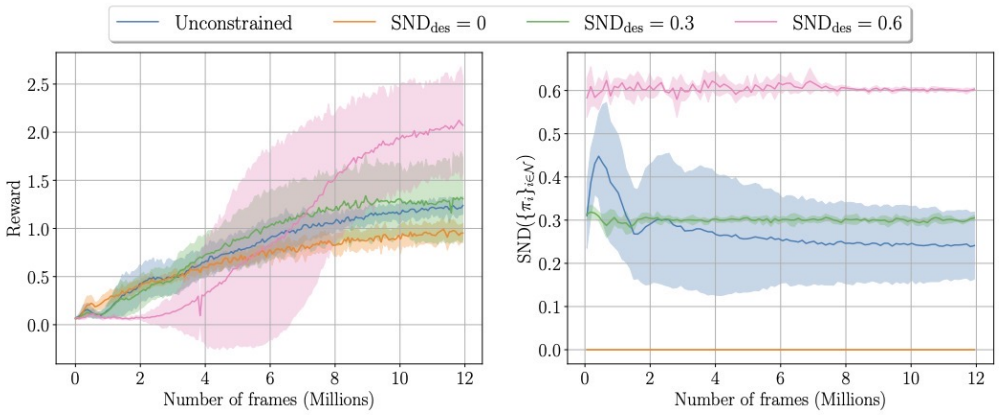

Рис. 22: Кривые обучения IPPO с разными значениями уровня разнообразия поведений. Адаптировано из [22].

9 Выводы

В ходе обзора были проанализированы различные подходы к решению задач мультиагентного обучения с подкреплением. На основе проведённого анализа выделены ключевые факторы, влияющие на успех обучения и производительность моделей:

- **Ограничения архитектуры:**

  Модели могут не сходиться к оптимальной стратегии при использовании централизованного актор-критика с parameter sharing, особенно в условиях сложных взаимодействий между агентами.

- **Проблемы сходимости:**

  Децентрализованные архитектуры с большим числом параметров также могут испытывать трудности со сходимостью.

- **Гибридные подходы:**

  Интеграция классических методов планирования с методами обучения с подкреплением, а также продуманный дизайн функции награды значительно улучшают сходимость и эффективность обучения в многокомпонентных системах.

- **Непрерывные действия:**

  Для построения эффективных и масштабируемых алгоритмов для мультиагентного планирования пути с непрерывными действиями нужно использовать дополнительных техники, направленные на исследование среды агентами, для обучения сложным кооперативным сценариям.

Список литературы

1. *Kraemer L.*, *Banerjee B.* Multi-agent reinforcement learning as a rehearsal for decentralized planning // Neurocomputing. — 2016. — Т. 190. — С. 82—94. — ISSN 0925-2312. — DOI: https://doi.org/10.1016/j.neucom.2016.01.031. — URL: https://www.sciencedirect.com/science/article/pii/S0925231216000783.
2. Counterfactual Multi-Agent Policy Gradients / J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, S. Whiteson // Proceedings of the AAAI Conference on Artificial Intelligence. — 2017. — Май. — Т. 32. — DOI: 10.1609/aaai.v32i1.11794.
3. Proximal Policy Optimization Algorithms / J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov. — 2017. — arXiv: 1707.06347 [cs.LG]. — URL: https://arxiv.org/abs/1707.06347.
4. Value-Decomposition Networks For Cooperative Multi-Agent Learning / P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, T. Graepel. — 2017. — arXiv: 1706.05296 [cs.AI]. — URL: https://arxiv.org/abs/1706.05296.
5. *Han S. D.*, *Yu J.* DDM: Fast Near-Optimal Multi-Robot Path Planning using Diversified-Path and Optimal Sub-Problem Solution Database Heuristics. — 2019. — arXiv: 1904.02598 [cs.RO]. — URL: https://arxiv.org/abs/1904.02598.
6. PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning / G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. K. S. Kumar, S. Koenig, H. Choset // IEEE Robotics and Automation Letters. — 2019. — Июль. — Т. 4, № 3. — С. 2378—2385. — ISSN 2377-3774. — DOI: 10.1109/lra.2019.2903261. — URL: http://dx.doi.org/10.1109/LRA.2019.2903261.
7. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning / K. Son, D. Kim, W. J. Kang, D. E. Hostallero, Y. Yi. — 2019. — arXiv: 1905.05408 [cs.LG]. — URL: https://arxiv.org/abs/1905.05408.
8. Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient / S. Li, Y. Wu, X. Cui, H. Dong, F. Fang, S. Russell // Proceedings of the AAAI Conference on Artificial Intelligence. — 2019. — Июль. — Т. 33, № 01. — С. 4213—4220. — DOI: 10.1609/aaai.v33i01.33014213. — URL: https://ojs.aaai.org/index.php/AAAI/article/view/4327.

9. Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge? / C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, S. Whiteson. — 2020. — arXiv: 2011.09533 [cs.AI]. — URL: https://arxiv.org/abs/2011.09533.

10. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning / T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, S. Whiteson. — 2020. — arXiv: 2003.08839 [cs.LG]. — URL: https://arxiv.org/abs/2003.08839.

11. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments / R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, I. Mordatch. — 2020. — arXiv: 1706.02275 [cs.LG]. — URL: https://arxiv.org/abs/1706.02275.

12. *Wang R. E.*, *Everett M.*, *How J. P.* R-MADDPG for Partially Observable Environments and Limited Communication. — 2020. — arXiv: 2002.06684 [cs.MA]. — URL: https://arxiv.org/abs/2002.06684.

13. Weighted QMIX: Expanding Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning / T. Rashid, G. Farquhar, B. Peng, S. Whiteson. — 2020. — arXiv: 2006.10800 [cs.LG]. — URL: https://arxiv.org/abs/2006.10800.

14. PRIMAL$_2$: Pathfinding Via Reinforcement and Imitation Multi-Agent Learning - Lifelong / M. Damani, Z. Luo, E. Wenzel, G. Sartoretti // IEEE Robotics and Automation Letters. — 2021. — Апр. — Т. 6, № 2. — С. 2666—2673. — ISSN 2377-3774. — DOI: 10.1109/lra.2021.3062803. — URL: http://dx.doi.org/10.1109/LRA.2021.3062803.

15. *Egorov V.*, *Shpilman A.* Scalable Multi-Agent Model-Based Reinforcement Learning. — 2022. — arXiv: 2205.15023 [cs.LG]. — URL: https://arxiv.org/abs/2205.15023.

16. POGEMA: Partially Observable Grid Environment for Multiple Agents / A. Skrynnik, A. Andreychuk, K. Yakovlev, A. I. Panov. — 2022. — arXiv: 2206.10944 [cs.LG]. — URL: https://arxiv.org/abs/2206.10944.

17. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games / C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, Y. Wu. — 2022. — arXiv: 2103.01955 [cs.LG]. — URL: https://arxiv.org/abs/2103.01955.

18. VMAS: A Vectorized Multi-Agent Simulator for Collective Robot Learning / M. Bettini, R. Kortvelesy, J. Blumenkamp, A. Prorok. — 2022. — arXiv: 2207.03530 [cs.RO]. — URL: https://arxiv.org/abs/2207.03530.

19. Accelerating Multi-Agent Planning Using Graph Transformers with Bounded Suboptimality / C. Yu, Q. Li, S. Gao, A. Prorok. — 2023. — arXiv: 2301.08451 [cs.AI]. — URL: https://arxiv.org/abs/2301.08451.

20. *Bettini M.*, *Shankar A.*, *Prorok A.* Heterogeneous Multi-Robot Reinforcement Learning. — 2023. — arXiv: 2301.07137 [cs.RO]. — URL: https://arxiv.org/abs/2301.07137.

21. *Albrecht S. V.*, *Christianos F.*, *Schäfer L.* Multi-Agent Reinforcement Learning: Foundations and Modern Approaches. — MIT Press, 2024. — URL: https://www.marl-book.com.

22. *Bettini M.*, *Kortvelesy R.*, *Prorok A.* Controlling Behavioral Diversity in Multi-Agent Reinforcement Learning. — 2024. — arXiv: 2405.15054 [cs.MA]. — URL: https://arxiv.org/abs/2405.15054.

23. Learn to Follow: Decentralized Lifelong Multi-Agent Pathfinding via Planning and Learning / A. Skrynnik, A. Andreychuk, M. Nesterova, K. Yakovlev, A. Panov // Proceedings of the AAAI Conference on Artificial Intelligence. — 2024. — Март. — Т. 38, № 16. — С. 17541—17549. — DOI: 10.1609/aaai.v38i16.29704. — URL: https://ojs.aaai.org/index.php/AAAI/article/view/29704.

24. MAPF-GPT: Imitation Learning for Multi-Agent Pathfinding at Scale / A. Andreychuk, K. Yakovlev, A. Panov, A. Skrynnik. — 2024. — arXiv: 2409.00134 [cs.MA]. — URL: https://arxiv.org/abs/2409.00134.

25. *Chernyavskiy A.*, *Skrynnik A.*, *Panov A.* Applying opponent and environment modelling in decentralised multi-agent reinforcement learning // Cognitive Systems Research. — 2025. — Т. 89. — С. 101306. — ISSN 1389-0417. — DOI: https://doi.org/10.1016/j.cogsys.2024.101306. — URL: https://www.sciencedirect.com/science/article/pii/S1389041724001001.