На правах рукописи

Панов Александр Игоревич

Методы и алгоритмы нейросимвольного обучения и планирования поведения когнитивных агентов

Специальность: 1.2.1. Искусственный интеллект и машинное обучение

АВТОРЕФЕРАТ

диссертации на соискание учёной степени доктора физико-математических наук

Москва — 2024

Работа выполнена в федеральном государственном учреждении «Федеральный исследовательский центр «Информатика и Управление» Российской академии наук».

Ведущая организация: Федеральное государственное бюджетное учреждение науки Институт проблем управления им. В.А. Трапезникова Российской академии наук

Защита состоится **14 ноября 2024 г. в 14 часов 00 минут** на заседании диссертационного совета **ФПМИ.1.2.1.003**, созданного на базе федерального государственного автономного образовательного учреждения высшего образования «Московский физико-технический институт (национальный исследовательский университет)» (МФТИ, Физтех)

по адресу: 141701, Московская область, г. Долгопрудный, Институтский переулок, д. 9.

С диссертацией можно ознакомиться в библиотеке МФТИ, Физтех и на сайте организации https://mipt.ru.

Автореферат разослан «___» _____ 2024 г.

Ученый секретарь
диссертационного совета **Войтиков Константин Юрьевич**

Общая характеристика работы

Актуальность темы.

В промышленности, в сфере логистики и в системах автоматизации в последние годы наблюдается курс на повсеместную роботизацию. На сборочных конвейерах, на складах и в логистических центрах все большее применение находят робототехнические платформы, как стационарные, манипуляционные, так и мобильные платформы. Однако, класс решаемых с помощью них задач существенно ограничен заранее определенными и смоделированными сценариями с минимальной степенью неопределенности в среде и с максимальным уровнем детерминированности всех выполняемых действий. Данная ситуация во многом связна с низкой степенью автономности робототехнических платформ (агентов) и с низкой адаптивностью систем управления, реализующих синтез поведения (последовательности реакций или действий). Разработка методов и алгоритмов синтеза целенаправленного поведения агентов, взаимодействующих со средой, является одной из центральных тем в исследованиях по искусственному интеллекту, что неоднократно подчеркивалось в работах Д.А. Поспелова, Г.С.Осипова. Существенный прогресс, достигнутый в направлении автоматического планирования и отмеченный в монографии М. Галлаба и П. Траверсо, позволил создать ряд эффективных символьных планировщиков, которые могут быть адаптированы для многоагентной постановки задачи и для планирования на основе прецедентов. В большинстве случаев, в том числе и в современных планировщиках, используются символьные способы представления знаний на основе нотации логики предикатов.

Достигнутые результаты в области планирования, однако, оказались плохо применимы для робототехнических интеллектуальных агентов. Во многом это связано с тем, что в области автоматического планирования понятие внешней среды используется очень ограниченно, только при рассмотрении вопросов выполнения построенного плана и принятия решений о модификации плана или полном перепланировании. В действительности в робототехнике более существенной и важной является решение задачи привязки символов (symbol grounding problem), сформулированной еще С. Харнадом, для соотнесения используемых в классическом планировщике символов, представляющих знания агента об объектах внешней среды, и данных, поступающих с сенсоров и датчиков. Оказалось, что прямолинейная модификация используемых при планировании символов в направлении учета неопределенностей, возникающих при интерпретации сенсорных данных и подаче управляющих сигналов, не позволяет существенно улучшить эффективность классических планировщиков при работе в модельной или реальной среде.

Теория интеллектуальных агентов нашла свое применение как напрямую в системах управления беспилотными транспортными средствами, так и косвенно через реализацию систем поддержки принятия решений оператора объекта управления. В данных прикладных системах реализованы некоторые важные архитектурные аспекты принятия решения и синтеза программы поведения агента: иерархичность в обработке сигналов и выработке управляющих воздействий, выбор рациональной тактики достижения цели, оперативное целеполагание.

Современные системы управления беспилотным транспортом и мобильными робототехническими платформами реализуют модульный подход к генерации автономного поведения. Различные подсистемы отвечают за выполнение определенного рода подзадач: генерация траектории движения, реализация предложенной траектории с учетом динамики объекта управления, детекция и сегментирование объектов во внешней среде, планирование действий по манипуляции объектами, обучение модели взаимодействия со средой. При усложнении задач, которые ставятся перед объектом управления, увеличивается количество необходимых подсистем и усложняется их внутренняя организация, межмодульное взаимодействие.

При этом в последнее время в области разработки общих систем искусственного интеллекта наметилась обратная тенденция по объединению функциональности различных модулей в связи с тем, что для повышения эффективности и адаптивности решения перечисленных выше подзадач, требуется комплексирование результатов или, во многих случаях, одновременная взаимосвязанная работа разных подсистем. Примерами подобных ситуаций могут служить варианты интеграции подсистем компьютерного зрения в задаче управления беспилотным автомобилем, двигающимся в среде с большим количеством других автомобилей и пешеходов, когда для повышения эффективности предсказания траекторий других участников движения необходимо интегрировать в этот модуль работу подсистем сегментации и трекинга объектов.

В настоящей работе предлагается новый подход к интеграции подсистем планирования поведения (т.е. действий как по перемещению, так и, например, по манипуляции предметами внешней среды) и обучения поведению, в котором формируется адаптивная стратегия по достижению поставленной перед агентом цели. Такая интеграция является естественной, так как обе подсистемы представляют собой различную реализацию модуля последовательного принятия решений. Однако при планировании необходима модель функционирования внешней среды, а модуль обучения может автоматически формировать такую модель в явном или неявном виде. Для эффективного учета возможностей обеих подсистем предлагается использовать иерархическую организацию как для всей системы управления, так и для разделения высокоуровневого планировщика, для которого уже не требуется

полной и точной модели, и низкоуровневой стратегии, которая обучается на основе оригинального метода обучения с подкреплением.

В настоящее время обучение с подкреплением без использования модели среды показывает впечатляющие результаты для многих задач управления поведением когнитивных агентов и их групп. Впрочем, большинство из используемых в этой области методов требуют чрезвычайно большого количества эпизодов взаимодействия агента со средой. Эта так называемая проблема эффективности выборок является одним из ключевых препятствий на пути развития новых методов и их внедрения в практику. Среди подходов по решению этой проблемы необходимо отметить имитационное обучение (с использованием заранее подготовленных демонстраций) и обучение на основе модели. Последний перспективен также по той причине, что допускает возможность использования эффективных методов планирования и поиска по графу марковского процесса принятия решений.

Построение модели с использованием аппроксиматоров является вычислительно затратной процедурой, и эффективные методы ее решения, не использующие заранее подготовленные и набранные агентом данные, только разрабатываются. Здесь перспективным видится подход по использованию объектного представления состояний среды, который позволяет декомпозировать модель среды на частные объектные модели. Актуальной является разработка именно объектной формулировки задачи обучения с подкреплением на основе модели. Для этого резонно использовать недавние результаты по свойствам эквивалентности моделей по функции полезности, чтобы уточнить постановку оптимизационной задачи обучения.

Таким образом, **объектом исследований** диссертационной работы являются методы и модели генерации поведения когнитивных агентов в недетерминированных, частично-наблюдаемых, динамических средах с возможность задания цели поведения на естественном языке. **Предметом исследования** является разработка новых концептуальных и математических моделей, методов и численных алгоритмов генерации поведения когнитивных агентов в недетерминированной, частично-наблюдаемой, динамической среде с одновременным обучением и планированием, использующих концепцию нейросимвольной интеграции, объектно-центричного представления и формирования модели среды. **Фундаментальной научной проблемой**, на решение которой направлено диссертационное исследование, является теоретическое обобщение и развитие методов, моделей и технологий генерации поведения когнитивных агентов при работе в сложных динамических средах и в условиях задания целей поведения на естественном языке.

Целью диссертационной работы – повышение эффективности, адаптивности и степени автономности систем управления мобильными робототехническими системами общего назначения за счет создания, внедрения и использования математического обеспечения для генерации поведения

когнитивных агентов в недетерминированных, частично-наблюдаемых, динамических средах.

В соответствии с поставленной целью были сформулированы следующие **задачи**:

1. Развить теорию генерации поведения когнитивного агента: уточнить терминологический аппарат, выявить имеющиеся закономерности, принципы и правила организации исследуемой предметной области, систематизировать имеющиеся и разрабатываемые методы и модели.
2. Разработать комплекс математических моделей, методы и алгоритмы генерации поведения когнитивного агента в недетерминированной динамической среде.
3. Выработать общую методику построения архитектуры когнитивного агента, способного обучаться стратегии поведения и планировать поведение на основе модели среды.
4. Спроектировать новую нейросимвольную архитектуру одновременного обучения и планирования когнитивного агента с использованием обновляемой модели мира и языковых моделей для генерации концептуального объектно-центричного плана.
5. Разработать новые методы нейросимвольного анализа сцен, в том числе использующие распутанные, факторизованные и объектно-центричные представления.
6. Создать новые методы обучения с подкреплением на основе модели среды, в том числе с использованием объектно-центричного представления.
7. Разработать новые алгоритмы интеграции планирования и обучения с подкреплением, применимые для многоагентной постановки задач в динамических средах.
8. Разработать алгоритмический инструментарий на основе предложенного комплекса математических моделей и экспериментальные программные реализации основных алгоритмов.
9. Применить предложенную общую методику по построению архитектуры когнитивного агента в ряде практических задач в области многоагентного планирования пути, навигации мобильных роботов, одновременного планирования задач и перемещений робота с манипулятором и адаптивного планирования маневров беспилотного транспортного средства.

Методология и методы исследования. Методология исследования базируется на комплексном использовании и развитии следующих методов и научно-методического обеспечения:

– методы обучения с подкреплением на основе функции полезности и градиента стратегии, в том числе методы оптимизации второго порядка и итеративные алгоритмы динамического программирования;

– методы эвристического планирования и поиска с функцией полезности по дереву состояний-действий;
– методы контрастивного, самоконтролируемого и слабоконтролируемого машинного обучения с использованием нелинейных аппроксиматоров (нейронных сетей, в том числе глубоких);
– методы построения специфической и неспецифической обратной связи к мультимодальной модели динамической среды.

Научная новизна: в диссертации получены следующие новые научные результаты:

– предложена новая универсальная архитектура когнитивного агента, использующая принципы иерархичности, нейросимвольной обработки информации и генерации поведения, обучения с подкреплением и адаптации языковых моделей для задачи генерации верхнеуровневого плана, расширяющая класс задач, в которых возможно эффективное обучения стратегии агента и построение оптимального плана действий;
– разработаны новые математические модели, методы и алгоритмы одновременного планирования и обучения когнитивных агентов, действующих в сложной динамической среде, в том числе с участием других агентов, повышающие общую эффективность и обобщающую способность генерируемой агентом стратегии и уменьшающие время обучения (адаптации) данной стратегии;
– предложен новый подход к обучению объектно-центричных представлений визуальных сцен, основанный на модели смеси гауссовых распределений и реализующий нейросимвольный уровень в предлагаемой универсальной архитектуре когнитивного агента;
– создан новый класс методов обучения с подкреплением на основе объектно-центричной модели мира, превосходящий современные аналоги по метрикам качества обучения стратегии агента в специализированных объектно-центричных средах;
– усовершенствован ряд существующих моделей и методов обучения с подкреплением на основе модели мира, с моделями внутренней мотивации и с использованием эвристических планировщиков;
– разработан новый подход к интеграции методов планирования действий агента в среде с помощью больших языковых моделей и методов обучения с подкреплением для адаптации получаемого плана к конкретным текущим условиям среды.

Помимо перечисленных результатов в диссертации разработана программная реализация системы управления робототехническими платформами с использованием языковых моделей для подзадачи планирования, развернутая на различных платформах и впервые позволившая решать задачи генерации последовательности действий по языковой инструкции.

Теоретическая значимость диссертационной работы заключается в развитии теории генерации поведения когнитивного агента в сложных динамических средах. В диссертационном исследовании предложена новая постановка задачи и предложены новые подходы к построению систем управления воплощенными когнитивными агентами. В работе предложены возможности развития методов, исследуемых в данной работе, как в рамках рассматриваемой постановки задачи, так и в рамках других классов задач обучения с подкреплением на основе модели среды. В диссертационном исследовании получены теоретические результаты и научно-обоснованные решения, которые вносят значительный вклад в развитие методов и подходов к управлению когнитивными агентами в динамических средах.

Практическая значимость диссертационной работы определяется тем, что полученные теоретические результаты в области построения архитектур управления когнитивными агентами были реализованы в виде модульного программного комплекса на базе операционной системы ROS, полностью или помодульно использованы в ряде прикладных проектов и могут в дальнейшем применяться для целого ряда задач в области робототехники, беспилотного транспорта, автоматизации любых процессов, связанных с последовательным принятием решений с обратной связью от внешних условий. Ключевой практической задачей, на решение которой направлены полученные в диссертационной работе результаты, является повышение степени автономности любых автономных воплощенных систем, действующих в сложной динамической среде.

Основные положения, выносимые на защиту:

1. Предложена нейросимвольная архитектура управления поведением когнитивного агента, включающая в себя компоненты одновременного планирования и обучения, а также компонент концептуального планирования с использованием языковых моделей.
2. Созданы модели и методы интеграции планирования и обучения с подкреплением, в том числе с использованием модели среды, для решения сложных визуальных и векторных задач управления поведением когнитивным агентом, включая задачи в многоагентной постановке.
3. Разработаны модели и методы объектно-центричного подхода к представлению сенсорной информации о статических сценах для использования в нейросимвольной архитектуре управления поведением когнитивного агента.
4. Созданы модели и методы объектно-центричного обучения с подкреплением с использованием динамической модели среды для интеграции планирования и обучения в нейросимвольной архитектуре управления поведением когнитивного агента.

5. Разработан программно-алгоритмический инструментарий, основанный, в том числе, на полученных теоретических результатах, для решения задачи генерации действий робототехнической платформой в сложной динамической среде, позволяющий использовать как обучаемые компоненты, так и классические планировочные. Создана экспериментальная программная реализация элементов данного инструментария, использующаяся для решения практических задач управления поведением.
6. Предложено использование разработанных моделей и методов одновременного планирования и обучения в ряде практически важных робототехнических задачах: навигация мобильной платформы внутри помещений, адаптивное планирование маневров беспилотным транспортным средством, перемещение и манипуляция объектами мобильной платформы по языковым инструкциям.

Достоверность полученных результатов обеспечивается общей методикой проведения математического анализа разработанных моделей и методов, а также методикой численного эксперимента для конкретных элементов программно-алгоритмического инструментария. Обоснованность научных результатов и выводов, представленных в работе, определяется корректным применением апробированных нейросетевых методов, методов планирования поведения и обучения с подкреплением. Результаты находятся в соответствии с результатами, полученными другими авторами. Для каждого из элементов программно-алгоритмического инструментария предлагается его детальное описание, а также полный список гиперпараметров, используемых при обучении. Основные результаты представлены в публикациях с высоким уровнем цитируемости, также на ведущих конференциях по тематике диссертации. Программные комплексы, созданные на основе результатов, полученных в диссертационном исследовании, успешно внедрены в целом ряде организаций.

Основные результаты получены автором в рамках научной деятельности и научных проектов, поддержанных грантами Российского научного фонда (№18-71-00143 «Иерархическое обучение с подкреплением в задаче приобретения концептуальных процедурных знаний когнитивными агентами», №20-71-10116 «Обучение с подкреплением с использованием сетевых векторно-символьных представлений в задаче интеллектуальной навигации когнитивных агентов»), Российского фонда фундаментальных исследований (№18-29-22027 «Персональные когнитивные ассистенты, сопровождающие деятельность человека в информационном пространстве», №17-29-07051 «Сетевая модель знаковой картины мира и реализация в ней когнитивных функций»), Министерства науки и высшего образования Российской Федерации (проект №075-15-2024-544 «Математические модели и численные методы как основа для разработки робототехнических комплексов, новых материалов и

интеллектуальных технологий конструирования»). Результаты **реализованы и внедрены** в ряде индустриальных компаний, в таких как НПК БИС, ООО «ИнтеграНТ», ПАО «Сбербанк».

Апробация работы. Основные результаты диссертации докладывались и обсуждались на ряде международных и российских конференциях. Часть результатов, вошедших в диссертационное исследование, были отмечены медалью Российской академии наук для молодых ученых за 2017 год, призами за лучшее решение задачи на соревнованиях NeurIPS MineRL 2019 (команда CDS) и CVPR Habitat 2023 (команда SkillFusion).

Публикации. Материалы диссертации опубликованы в **98** рецензируемых печатных работах, относящихся к следующим категориям:

- **54** статьи в изданиях из собственного перечня журналов МФТИ категории К1: [1; 3—7; 9; 12; 13; 15; 18; 20; 22; 24; 28; 30—33; 35—37; 39; 42—46; 49; 51; 52; 54; 57; 60—62; 66—77; 82; 84; 87; 89; 95; 98];
- **8** статей в изданиях, приравненных к журналам перечня ВАК категории К1: [21; 23; 26; 34; 38; 47; 48; 59];
- **7** статей в изданиях, приравненных к журналам перечня ВАК категории К3: [11; 25; 27; 29; 40; 50; 56];
- **4** статьи в трудах конференций уровня A*, индексируемых в Web of Science и/или Scopus: [2; 10; 14; 16];
- **7** статей в остальных изданиях, индексируемых в Web of Science и/или Scopus: [8; 19; 41; 53; 63—65];
- **1** монография: [78];
- **17** статей в остальных рецензируемых изданиях: [17; 55; 58; 79—81; 83; 85; 86; 88; 90—94; 96; 97].

Личный вклад. Содержание диссертации и основные положения, выносимые на защиту, отражают персональный вклад автора в опубликованные работы. Все представленные в диссертации результаты получены лично автором.

Объем и структура работы. Диссертация состоит из введения, шести глав, заключения и приложения. Полный объем диссертации **404** страницы текста с **104** рисунками и **23** таблицами. Список литературы содержит **595** наименования.

Содержание работы

Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, приводится обзор научной литературы по изучаемой проблеме, формулируется цель, ставятся задачи работы, излагается научная новизна и практическая значимость представляемой работы.

В **первой главе** дается краткое изложение основ обучения с подкреплением как без использования модели среды, так и на основе обновляемой модели, а также методов планирования по известной модели. **Раздел 1.1** посвящен безмодельному обучению с подкреплением и его месте в классе методов, относящихся к категории подходов к принятию решений в условиях неопределенности. Рассматривается классический марковский процесс принятия решений, описывается обучение на основе функции полезности и параметризация стратегии. Даются основные теоретические основы работы архитектуры агента семейства «актор-критик», описываются современные методы оптимизации градиента стратегии. **Раздел 1.2** посвящен планированию поведения по известной модели. Даются постановки задачи и основные базовые решения для случая классических допущений и в условиях неопределенности. Отдельно обсуждается задача представления состояний в задаче планирования поведения. **Раздел 1.3** посвящен методам и подходам к интеграции планирования и обучения в единой архитектуре агента. Обсуждаются методы обучения модели мира и отдельно задача обучения эффективных представлений для моделирования среды. Описываются особенности планирования по обучаемой модели. Дана классификация подходов к интеграции планирования и обучения в полном цикле.

Во **второй главе** предлагается концептуальное описание разработанной автором нейросимвольной архитектуры для планирования и обучения NSLP. Уточняется постановка задачи, для которой используется данная архитектура, проводится теоретический анализ особенностей ее работы в процессе приобретения знаний и их использования при планировании. Основные результаты главы опубликованы в работах [13; 61; 62; 64; 66; 88; 89; 91; 95; 96; 98].

На рисунке 1 представлена взаимосвязь описываемых в главе 2 архитектур STRL и NSLP с представленным далее в данном диссертационном исследовании программно-алгоритмическим инструментарием. Один из вариантов полной реализации с помощью программного обеспечения ROS представлен в разделе 5.1 (STRL-Robotics). Реализация нейросимвольной компоненты архитектуры NSLP подробно представлены методами VQ-SA, SMM и SBWM. Реализация концепции одновременного обучения и планирования в архитектуре NSLP изложена с помощью методов MB-AC, MB-NODE, MA-MCTS и MATS-LP (глава 3). Использование объектного принципа нейросимвольной интеграции в обучениия и планировании иллюстрируется методами ROCA (раздел 4.3) и L2S (раздел 4.4). Прикладные и когнитивные аспекты освещаются в главах 5 и 6.

Раздел 2.1 посвящен краткому обзору когнитивных архитектур, особенностей их построения и ключевых недостатков, которые привели к пересмотру подходов по построению архитектур управления поведением когнитивных агентов и необходимости развития новых нейросимвольных подходов.

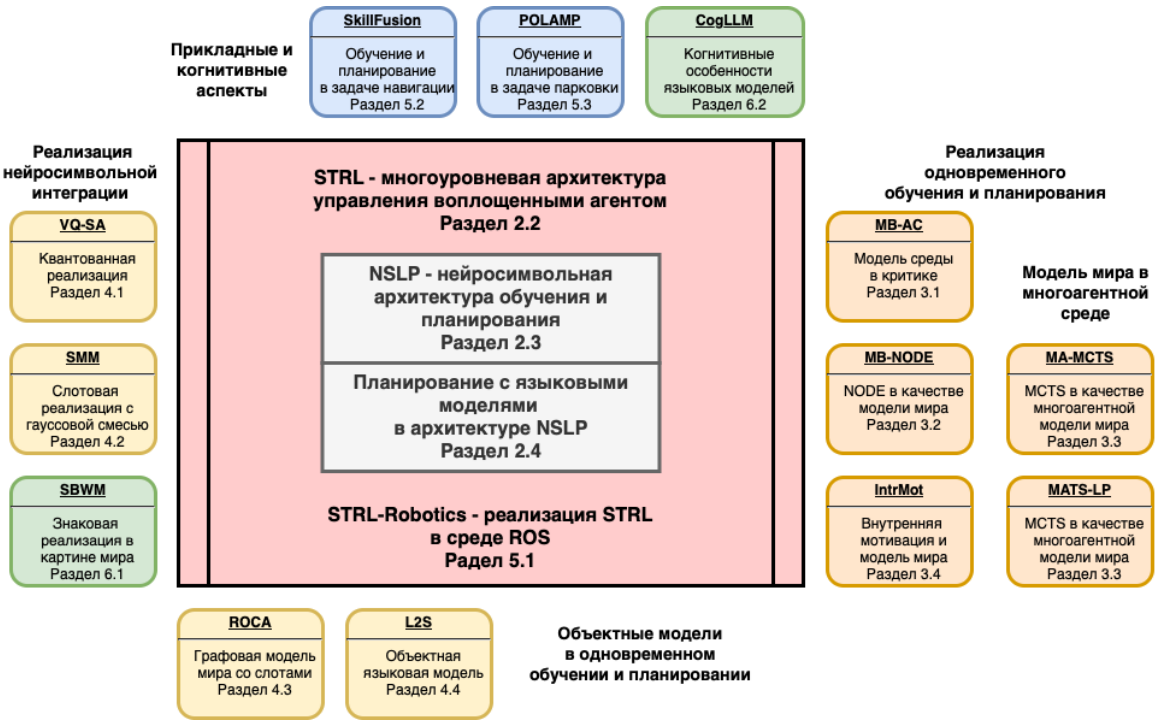

Рисунок 1 — Взаимосвязь архитектур STRL и NSLP с программно-алгоритмическим инструментарием, представленным в дальнейших главах диссертационного исследования.

Раздел 2.2 посвящен описанию предложенной автором базовой иерархической архитектуры STRL, изначально предназначенной для управления группой сложных технических объектов. В условиях динамической среды, свойства и поведение которой заранее не известны когнитивному агенту, в качестве которого будет пониматься мобильная робототехническая платформа, в диссертационном исследовании предлагается использовать обновленную версию архитектуры STRL, в которой сделан акцент на возможность обучения как в процессе выполнения действий в среде, так и на предобучение на заранее собранных наборах данных.

Раздел 2.3 содержит основную информацию и описание NSLP архитектуры, являющейся одним из основных результатов диссертационного исследования. Опишем наиболее общий класс задач, для решения которого предназначена описываемая в данном разделе диссертационного исследования архитектура NSLP (neuro-symbolic learning and planning, нейросимвольное обучение и планирование). В диссертационной работе рассматривается ситуация последовательного принятия решений когнитивным воплощенным агентом, действующим в неизвестной внешней среде, в условиях неопределенности. Предполагается, что агенту известно только доступное множество действий A и пространство наблюдений O. Ни функция переходов T, ни множество информационных состояний S МППР для агенты не доступны. Поведение, характеризуемое последовательностью решений агента, является

целенаправленным, то есть агенту предоставляется информация о множестве целевых состояний $S_g \subset S$ в следующем виде:

– через функцию вознаграждения $R = r(s_t,a_{t+1},s_{t+1})$, сигнализирующую о которая в простейшем виде записывается как

$$r(s_t,a_{t+1},s_{t+1}) = \begin{cases} 1, \text{если } s_{t+1} \in S_g, \\ 0, \text{иначе;} \end{cases} \tag{1}$$

– через языковую инструкцию (множество токенов) $I = \{q_1, \ldots, q_n\}$, описывающую критерии достижения цели или выполнения задания.

Необходимо отметить, что задание цели не подразумевает фиксацию условий в среде, в которых будет происходить процесс принятия решений. То есть, если задано множество экземпляров среды $E = \{E_1, \ldots, E_m\}$, где E_i – конкретный экземпляр отличающийся от других функцией переходов T_i, соответствующего этому экземпляру МППР, то одна и та же задача $task = \langle R, I \rangle$ может быть корректно поставлена в некотором подмножестве данных сред. Целью агента $goal_i$ будет называться задача в конкретных условиях среды E_i, то есть пара $goal_i = \langle task, E_i \rangle$.

Пусть в процессе принятия решений (взаимодействия) агентом в среде E_i был получен эпизод (траектория) $\tau \sim E_i$ вида $\tau = (s_0,a_0,r_0, \ldots, s_{n-1}, a_{n-1}, r_{n-1}, s_n)$. Метриками качества принятых агентом решений являются две величины:

– критерий достижения цели (success rate) SR:

$$SR(\tau, task) = \begin{cases} 1, \text{если } s_n \in S_g(task), \\ 0, \text{иначе;} \end{cases} \tag{2}$$

– мягкий критерий достижения цели (soft success rate) $SSR(\tau, task) = SR(\tau,task) \sum_{t=0}^{n-1} \gamma^t r_t$.

Формулируемый класс задач будет подразумевать работу агенту в двух режимах. *Режим обучения:* пусть сформулирована задача $task \langle R, I \rangle$ и имеется набор сред E, тогда задачей одновременного планирования и обучения называется задача поиска такой функции $NSLP(\theta_{enc}, \theta_{wm}, \theta_{pol}) : O \to A$ с параметрами кодировщика наблюдений θ_{enc}, модели мира θ_{wm} и стратегии θ_{pol}, что метрика SSR максимизируется на наборе сред E:

$$\max_{\theta_{enc},\theta_{wm},\theta_{pol}} \mathbb{E}_{\substack{NSLP(\theta_{enc},\theta_{wm},\theta_{pol}) \\ E_i \in E, \tau \sim E_i}} SSR(\tau,task). \tag{3}$$

Режим вывода (оценки): подразумевает фиксацию параметров $\theta_{enc}, \theta_{wm}, \theta_{pol}$ и запуск функции $NSLP(o_t)$ в режиме генерации действий a_t и подсчет метрики SSR по получающимся сгенерированным траекториям τ в новом наборе сред, отличающемся от используемых в режиме обучения.

Общая схема архитектуры NSLP представлена на рисунке 2. Она разделена на три уровня со своим набором внешних параметров и необходимых данных для предобучения используемых компонент. *Концептуальный уровень* предназначен для формирования концептуального плана с набором подцелей для достижения цели, то есть выполнения задачи $task = \langle R, I \rangle$ с функцией вознаграждения R и/или языковым описанием I в условиях конкретной среды E. Для этого используется графовое представление сцены $G_t = \langle En, Rel \rangle$, поступающее с нейросимвольного уровня, а на выходе генерируется символьный план действий $planner(\theta_{plan}) \rightarrow P$. Здесь в графе сцены $En = \{e_1, \ldots, e_n\}$ – множество выделенных в среде E объектов, в том числе и другие участники деятельности (другие агенты), а Rel – множество отношений (бинарных), устанавливаемых на множестве объектов En. Внешними параметрами на этом уровне являются описание задачи $task$ и информация о других частниках совместной деятельности (например, координаты других агентов). Предобученным компонентом на данном уровне является языковая модель $L(\theta_{llm})$, которая обычно обучается в самоконтролируемом режиме на множестве текстов и инструкций из специализированных наборов данных. Эвристические параметры θ_{plan} планировщика $planner$ также считаются заданными.

Сенсорный уровень собственно реализует одновременное обучения и планирование с использованием некоторого сенсорного представления информационного состояния s_t (в смысле марковского процесса принятия решений, формализующего взаимодействие агента и среды E), получаемого на основе сенсорной модели кодировщика $enc(o_t) \rightarrow s_t$, в свою очередь, обрабатывающей наблюдение o_t с тактического уровня STRL. Представление сцены s_t передается на нейросимвольный уровень. На выходе это уровня при выполнении стратегии $\pi(s_t, \beta | \theta_{hl})$ и/или навыков κ_j генерируется низкоуровневое действие a_t, реализуемое конкретной операцией на тактическом или реактивном уровне STRL. β в стратегии π – это набор подцелей с нейросимвольного уровня архитектуры NSLP. Здесь в качестве внешних параметров выступает конкретная модель сенсорного ситуации enc, то есть модель кодировщика наблюдения агента o_t, сгенерированного на тактическом уровне STRL (например, в качестве o_t могут выступать представления мультимодальных или сегментационных карт). Для предобучения библиотеки навыков $\{\kappa_1, \ldots, \kappa_m\}$, используемых в верхнеуровневой стратегии π, применяются симуляторы внешней среды E с заданными сценариями для предобучения конкретных навыков.

Нейросимвольный уровень собственно связывает концептуальный и сенсорный уровни за счет двух ключевых моделей: нейросимвольной объектной модели $NS(s_t | \theta_{enc})$ и мультимодальной модели привязки $MM(s_t)$, сопоставляющих названия (имена) объектов и действий с их внутренними представлениями сенсорного уровня. Результатом работы модели NS является набор представлений объектов $\{e_1, \ldots, e_n\}$, которые на концептуальном

уровне уже могут быть проинтерпретированы в виде графа сущностей G_t (например, с помощью графовой нейронной сети). На нейросимвольном уровне происходит разложение верхнеуровневого плана P на множество подцелей $\beta = \{\beta_1, \ldots, \beta_k\}$ с помощью генератора подцелей $GP(P, s_t) \to \beta$. Предполагается, что текстовое представление этапов плана с помощью мультимодальной модели привязки MM сопоставляется с внутренними представлениями подцелей β_i. В простейшем случает такое сопоставление может реализовываться заранее заданными отображением. Внешними параметрами служат конкретная реализация объектной модели (знаковый или слотовый подходы к представлению объектов) и мультимодальные наборы данных (обычно пары текст-изображение) для предобучения как данной объектной модели NS, так и мультимодальной модели привязки MM.

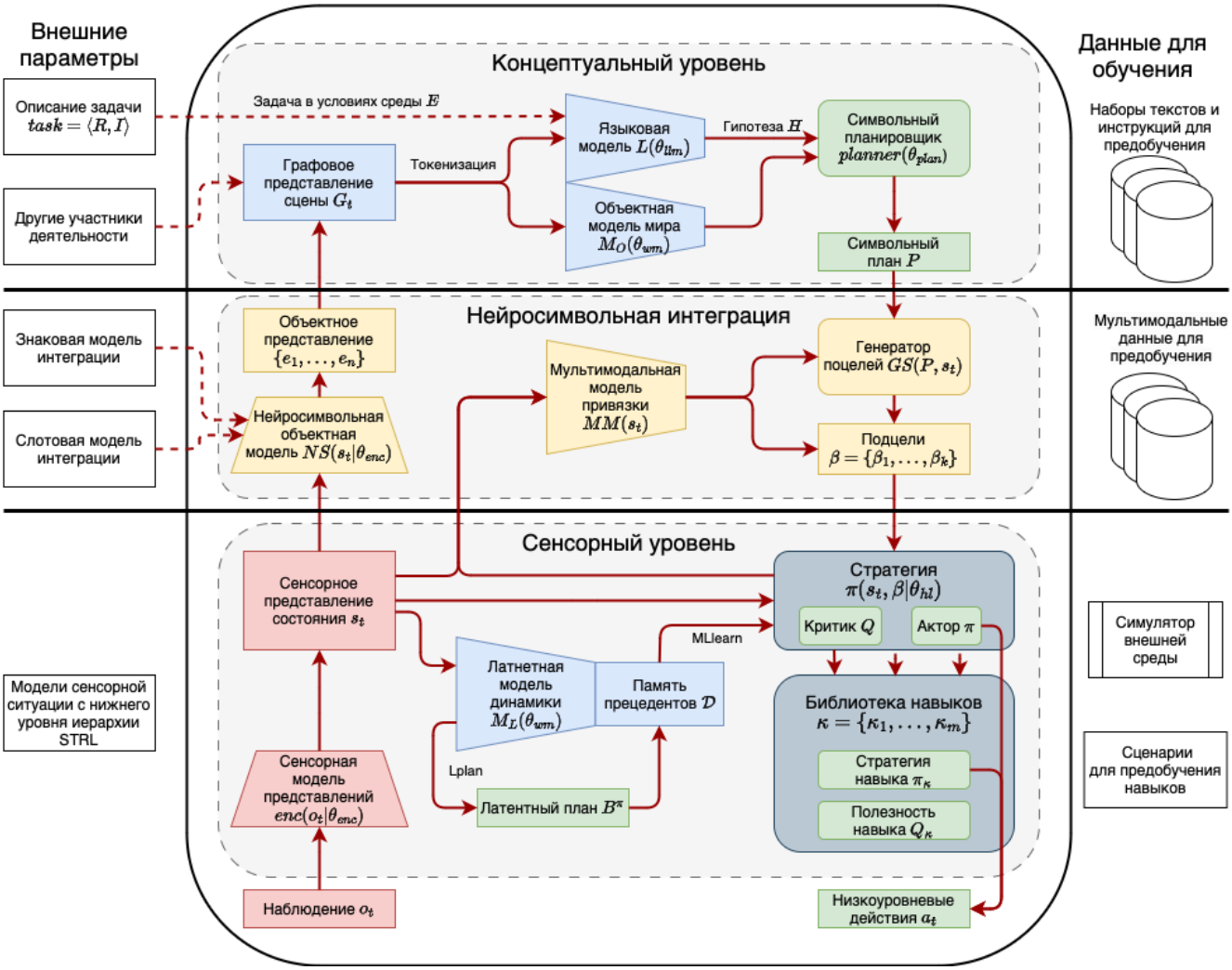

Рисунок 2 — Архитектура NSLP агента с подсистемами обучения и планирования поведения.

Необходимо отметить, что предполагается, что NSLP встроена в более общую STRL архитектуру и наблюдения $o_t \in O$ в нее поступают с тактического уровня STRL, а действия $a_t \in A$, по сути, являются, в свою очередь, операциями, реализуемыми на тактическом и реактивном уровне

за счет тактического планирования и реактивного следования конкретной траектории в обобщенном конфигурационном пространстве (например, для мобильной робототехнической платформы и манипулятора). Далее опишем функционирование архитектуры NSLP в двух режимах: вывода (inference), когда используется текущая архитектура и идет накопление опыта (памяти прецедентов $\mathcal{D}$, данных для обучения моделей NS, enc, M_O, M_L), и обучения (learning) стратегии π и моделей с использованием накопленного опыта.

В режиме вывода когнитивный агент на основе архитектуры NSLP фиксирует все параметры обучаемых компонент архитектуры: языковой модели $L(\theta_{llm})$, объектной модели мира $M_O(\theta_{wm})$, нейросимвольной объектной модели $NS(\theta_{enc})$, мультимодальной модели привязки MM, сенсорной модели представлений $enc(\theta_{enc})$, латентной модели динамики $M_L(\theta_{wm})$, стратегии агента $\pi(\theta_{hl})$ и библиотеки навыков κ – и использует их в генеративном режиме, получая на входе наблюдение o_t и выдавая низкоуровневые действия a_t. По сути агент реализует многопараметрическую многокомпонентную функцию

$$NSLP(o_t) = \pi \circ GS \circ planner \circ (L,M_O) \circ NS \circ enc \to a_t. \tag{4}$$

В режиме обучения используется накопленная память прецедентов $\mathcal{D}$ для обновления верхнеуровневой стратегии агента по достижения конкретных подцелей β_i, полученных из символьного плана P с помощью генератора подцелей GS. В режиме «актор-критик» обновляется также функция полезности Q, которая, в соответствии с теоремой о градиенте стратегии, используется для коррекции градиентного шага, вычисляемого по функции полезности стратегии π. Также возможно дообучение языковой модели L на сформированных признаках успешности выполнения сгенерированного плана на основе вознаграждений из памяти прецедентов (в режиме RLHF или DPO).

В задаче одновременного обучения и планирования агент автоматически строит обновляемую латентную модель среды $M_L =< \hat{T},\hat{R},\hat{\Omega} >$. Рассмотрим случай, когда модели M_L и M_O совпадают, то есть модель M_L будет иметь доступ не к состоянию s_t, а к уже декомпозированному представлению после модели NS. В этом случае латентный план B^π будет совпадать с объектным планом P, а стратегия π будет реализовываться планировщиком $planner$. Таким образом, модель M_L позволяет находить план B^π достижения цели, задаваемой в данной среде с помощью формулировки задачи $task = \langle R,I \rangle$(см. предыдущий параграф), за счет моделирования переходов при некоторой модельной стратегии π:

$$B^\pi =< o_0,r_0,a_0,o_1,r_1,a_1,\ldots,a_{l-1},o_l >, \tag{5}$$

где $s_i = \hat{T}(s_{i-1},a_{i-1})$, $r_i = \hat{R}(s_i,a_i)$, $a_i \sim \pi(a|o_i)$, а заключительное наблюдение o_l соответствует целевому состоянию s_l согласно функции $\hat{\Omega}$. Здесь под $\hat{T},\hat{\Omega}$ и $\hat{R}$ подразумеваются приближенные (аппроксимируемые) значения функций

переходов, наблюдений и вознаграждений, соответственно. План поведения агента, таким образом, составляется «жадным» алгоритмом в точности до небольшой поправки, отвечающей за исследование среды, в предположении корректности текущего приближения модели $M_L =< \hat{T},\hat{R},\hat{\Omega} >$. Далее будет предполагаться, что $\hat{\Omega} = enc$ и модель будет состоять только из двух первых компонент кортежа.

Под процессом обучения агента будет пониматься итерационное обновление модели $M_L =< \hat{T},\hat{R} >$, функций полезности Q, стратегии π и, соответственно, плана поведения B^π. Возможны четыре основных варианта составления общей схемы обучения агента с использованием фазы планирования по модели. Ниже представлены краткие алгоритмические схемы для этих вариантов. Как и в предыдущем параграфе весь собранный агентом опыт будет представлен в виде множества прецедентов $\mathcal{D} = \{(o_t,r_t,a_t)_{t=1}^T\}$. Траекторией называется некоторая последовательность таких прецедентов в порядке их формирования при взаимодействии со средой.

Принимая во внимание все приведенные уточнения реализации одновременного обучения и планирования, получается иерархическая постановка обучения с подкреплением на основе модели, в которой агенту необходимо максимизировать получаемую в рамках эпизода отдачу с возможностью декомпозиции наблюдения по предметному принципу, автоматическому построению стратегий умений и одновременному автоматическому формированию модели среды, используемой для точного планирования на множестве умений.

Подсистема обучения агента делится на две составляющие, как это принято в теории обучения с подкреплением. Критик обновляет функцию полезности действия Q_a, а актор формирует стратегию π_κ в рамках текущего умения. Цикл взаимодействия NSLP агента со средой будет выглядеть следующем образом:

1. Агент использует текущую модель M_L для того, чтобы сформировать план $B^\pi =< o_0,r_0,a_0,o_1,r_1,a_1,\ldots,a_{l-1},o_l >$ на множестве умений с помощью процедуры MLplan (реализуемой планировщиком *planner*). В общем случае предполагается, что уровней иерархии действий может быть несколько (вложенные умения), и может быть сформировано несколько вложенных планов: от высокоуровневого до низкоуровневого. Далее предполагается, что план B^π является низкоуровневым.
2. В соответствии с планом B^π агент выбирает текущее умение κ_t.
3. Агент получает из среды текущее наблюдение, которое с помощью функции NS переводится в набор сущностей $o_t \to \{e_1,\ldots,e_n\}$.
4. В соответствии со стратегией π_κ для умения κ_t агент выбирает текущее действие a_t.

5. Агент выполняет действие a_t в среде и получает новые наблюдения и вознаграждение.
6. В режиме обучения агент выполняет оценку выполненного действия и самого умения с помощью критика, а затем обновляет параметры аппроксиматоров критика и актора при помощи процедуры MLlearn.
7. Если текущее умение завершилось, выполняется перепланирование, и затем происходит переход к шагу 3.

При планировании агент использует модель $M_L =< \hat{T},\hat{R} >$ для построения плана своего поведения. В качестве примера в данном параграфе предлагается использовать реализацию модели в виде расширенного дерева поиска Монте-Карло, где каждый узел дерева отвечает за конкретный объект, выделяемый из наблюдения. Ребро дерева соответствует выбору некоторого объекта из наблюдения, выполнению некоторого действия (умения) и переходу к наблюдению, где выделяется следующий объект. В том случае, когда для выполнения выбирается действие (умение), для которого в модели не известно следующее наблюдение, образуются новые узлы с соответствующими объектами, выделенными из наблюдения, полученного из среды.

Планирование $MLplan(M,e)$ в данном дереве $M_L =< \hat{T},\hat{R} >$ происходит за счет поиска кратчайшего пути с учетом дополнительного веса для действий, направленных на исследование среды:

1. Выбирается планируемое умение $\kappa \leftarrow \arg\max_{\kappa} Q_\kappa(e) + \eta\sqrt{\frac{\log N(e)}{N(e,\kappa)}}$. Здесь второе слагаемое отвечает за верхнюю доверительную границу (UTC) эффективной стратегии исследования среды, $\eta \in \mathbb{R}$ — константа, N — счетчики.
2. Производится переход по дереву $e' \leftarrow \hat{T}_c(\kappa)$, $r \leftarrow \hat{R}_c(\kappa)$, где c — класс объекта e.
3. Производится планирование для следующего объекта с подсчетом получаемого вознаграждения $\tilde{R} \leftarrow r + \gamma MLplan(M,e')$.
4. Обновляются счетчики $N(e,\kappa) \leftarrow N(e,\kappa) + 1$, $N(e) \leftarrow N(e) + 1$.
5. Настраивается модель — обновляется полезность умения $Q_\kappa(b,\kappa) \leftarrow Q_\kappa(b,\kappa) + \frac{\tilde{R} - Q_\kappa(b,\kappa)}{N(e,\kappa)}$, где b — предполагаемое состояние, для которого выделяется объект e.

В процедуре обучения критика и актора MLlearn производится обновление как критерия достижения подцели β_κ, так и стратегии π_κ конкретного выбранного на верхнем уровне иерархии умения. Здесь используется параметризация с помощью набора параметров ϑ для условия завершения и набора θ — для стратегии.

1. Агент выбирает действие в соответствии с текущей стратегией умения $a \sim \pi_{\kappa,\theta}(a|o)$.
2. Агент выполняет действие a, наблюдает o' и r.

3. Критик обновляет оценку полезности действия в рамках текущего умения:

$$Q_a(b,\kappa,a) \leftarrow Q_a(b,\kappa,a) + \\ +\alpha\left(r+\gamma\left((1-\beta_{\kappa,\vartheta}(o'))Q_\kappa(b',\kappa)+\beta_{\kappa,\vartheta}(o')\max_{\kappa'}Q_\kappa(b',\kappa')\right)-Q_a(b,\kappa,a)\right)$$

где α — шаг обучения критика, g — обновляемое целевое значение для критика.

4. Актор обновляет параметры для стратегии и для подцели:

$$\theta \leftarrow \theta + \alpha_\theta \nabla_\theta \log \pi_{\kappa,\theta}(a|o) Q_a(b,\kappa,a), \tag{6}$$

$$\vartheta \leftarrow \vartheta + \alpha_\vartheta \nabla_\vartheta \tilde{Q}_\kappa(b',\kappa), \tag{7}$$

где α_θ и α_ϑ — шаги обучения.

5. Обновляется значение градиента полезности умения $\nabla_\theta Q_\kappa$, который может быть использован на верхнем уровне иерархии.
6. Если в соответствии с $\beta_{\kappa,\vartheta}$ подцель достигнута, то в соответствии с высокоуровневым планом выбирается новое умение κ.

Здесь предполагается, что полезность текущего умения передается из подсистемы планирования, где их обновление происходит во время обновления самой модели. Далее будет дан вывод выражений для градиента $\nabla_\theta Q_\kappa$ (теорема о градиенте критика) и для градиента $\nabla_\vartheta \tilde{Q}_\kappa$ генератора подцелей (теорема о градиенте актора).

Выше была представлена общая схема взаимодействия агента со средой в режиме одновременного обучения и планирования, где для обучения агента необходимо знание целевого значения критика g и градиента функции полезности стратегии ∇J. В диссертации доказаны следующие теоремы о вычислении этих значений.

Теорема 1 (о градиенте критика умений)**.** *Для фиксированного множества марковских умений со стохастической реализующей стратегией, дифференцируемой по параметрам θ, градиент ожидаемой дисконтированной отдачи по параметрам θ с начальными условиями (b_0,κ_0) равен*

$$\nabla_\theta Q_\kappa(b,\kappa) = \sum_{b,\kappa} \mu_\kappa(b,\kappa|b_0,\kappa_0) \sum_a \left(\nabla_\theta \pi_{\kappa,\theta}(a|o)\right) Q_a(b,\kappa,a),$$

где $\mu_\kappa(b,\kappa|b_0,\kappa_0)$ — дисконтированные частоты появления предполагаемого состояния и умения по траекториям, начинающимся с начальных условий (b_0,κ_0).

Теорема 2 (о градиенте генератора подцелей)**.** *Для фиксированного множества марковских умений со стохастической реализующей стратегией,*

дифференцируемой по параметрам ϑ, градиент ожидаемой дисконтированной отдачи по параметрам ϑ с начальными условиями (b_1,κ_0) равен

$$\nabla_\vartheta \tilde{Q}_\kappa(b,\kappa) = \sum_{b',\kappa} \mu_\kappa(b',\kappa|b_1,\kappa_0)\nabla_\vartheta \beta_{\kappa,\vartheta}(o')(V_\kappa(b') - Q_\kappa(b',\kappa)),$$

где $\mu_\kappa(b',\kappa|b_1,\kappa_0)$ — дисконтированные частоты появления предполагаемого состояния и умения по траекториям, начинающимся с начальных условий (b_1,κ_0).

Когда всю информацию об объектах удается закодировать в признаках, в том числе информацию об их взаимодействии с другими объектами, модель среды может быть декомпозирована пообъектно: $\tilde{T} = \{\tilde{T}_{c_i}|c_i \in C\}$, $\tilde{R} = \{\tilde{R}_{c_i}|c_i \in C\}$. Такая декомпозиция естественным образом продолжается на множество действий и функцию полезности, т. е. в том случае, когда выполняется условие пообъектного разделения условий и эффектов каждого действия, множество действий разбивается на подмножества для каждого класса объектов: $A = \{A_{c_i}|c_i \in C\}$. Классовое действие $a_{c_i} \in A_{c_i}$ меняет свойства только одного объекта $e \in c_i$, $e \in s$. Аналогично можно проследить вознаграждения, получаемые для каждого класса объектов отдельно, и, таким образом, ввести объектную параметризацию функции полезности: $V^\pi(e)$, $e \in s$.

Для формулировки полезных свойств объектного представления модели в диссертации используется принцип эквивалентности модели по полезности. Любая модель $M_O =< \hat{T},\hat{R} >$ (далее просто M) индуцирует оператор Беллмана , где функциями переходов и вознаграждения служат элементы модели M.

Определение 1 (Эквивалентность моделей)**.** *Пусть $\mathcal{P}$ — множество стратегий, а $\mathcal{V}$ — множество некоторых функций (полезности). Две модели $M =< T,R >$ и $\tilde{M} =< \tilde{T},\tilde{R} >$ называются эквивалентными по полезности относительно$\mathcal{P}$ и $\mathcal{V}$ тогда и только тогда, когда*

$$\mathcal{B}_\pi[V] = \tilde{\mathcal{B}}_\pi[V], \forall\pi\in\mathcal{P}, V \in \mathcal{V},$$

где $\mathcal{B}_\pi$ и $\tilde{\mathcal{B}}_\pi$ — операторы Беллмана, индуцированные моделью M и $\tilde{M}$, соответственно.

Данное определение приводит к заданию функциональных классов моделей, в которых элементы являются неразличимыми с точки зрения структуры функции полезности, задаваемой уравнением Беллмана.

Определение 2 (Пространство эквивалентности)**.** *Пусть $\mathcal{P}$ — множество стратегий, $\mathcal{V}$ — множество некоторых функций (полезности), а $\mathcal{M}$ — пространство моделей. При заданной некоторой модели m $\mathcal{M}_m(\mathcal{P},\mathcal{V})$ называется пространством эквивалентных по полезности моделей, каждая из которых эквивалентна по полезности модели m относительно $\mathcal{P}$и $\mathcal{V}$.*

В обучении с подкреплением на основе модели целью агента является в том числе и поиск оптимальной модели $m^* =< T,R >$ с истинными функциями переходов и вознаграждения. Если предполагается использование модели именно для более быстрой оценки стратегии, то агенту достаточно найти любую модель, которая эквивалентна по полезности оптимальной модели m^*, т.е. найти $m \in \mathcal{M}_{m^*}(\mathcal{P},\mathcal{V}) \equiv \mathcal{M}(\mathcal{P},\mathcal{V})$. Для этого оптимизационную задачу, которая ставится изначально как обычное обучение с учителем

$$\begin{cases} \arg\min_{\hat{R}} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[(R(s,a) - \hat{R}(s,a))^2 \right] \\ \arg\min_{\hat{T}} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[D_{KL}(T(\cdot|s,a)||\hat{T}(\cdot|s,a)) \right], \end{cases} \tag{8}$$

можно переписать, объединив минимизирующие функционалы по функциям вознаграждения и переходов в один функционал для уравнения Беллмана:

$$\arg\min_{\tilde{m}} \sum_{\pi\in\mathcal{P}} \sum_{V\in\mathcal{V}} \left\| \mathcal{B}_\pi [V] - \tilde{\mathcal{B}}_\pi [V] \right\|. \tag{9}$$

Соответствующая функция потерь уже явно зависит от выбора стратегии и соответствующей функции полезности, что позволяет, во-первых, сократить пространство поиска, а во-вторых, использовать свойство функциональной эквивалентности по полезности моделей. Основной трудностью в данном случае является определение пространств $\mathcal{P}$ и $\mathcal{V}$, по которым проходит сравнение двух операторов Беллмана.

В диссертации рассмотрена объектная декомпозиция модели, чтобы переформулировать оптимизационную задачу и сравнить решения, получаемые при использовании объектного представления и без него. Так как ранее была введена объектная параметризацию всех функций, участвующих в определении оператора Беллмана, то его объектная версия будет выглядеть следующим образом:

$$\mathcal{B}_\pi [V](e) = E_{a(e)\sim\pi, s'\sim T} \left[R(e,a) + \gamma V\left(e'\right) \middle| e' \in s' \right]. \tag{10}$$

Особенностью данного оператора является то, что выполнение действия для агента становится двухэтапным: вначале агент должен определить объект $e \in s$, с которым он будет взаимодействовать, а затем уже выбрать действие из множества доступных действий A_c, определенных для класса данного объекта $c(e)$. Таким образом, получается иерархический вариант обучения с подкреплением, для которого обычно используется формулировка абстрактных действий в виде умений.

В соответствии с определением умения пусть $\kappa(c) =< I_c,\ \pi_c,\ \beta_c >$ — длящееся во времени действие, или умение, для которого $I_c \subseteq S$ — инициирующее множество состояний, такое, что $\forall s \in S, e \in s, c(e) = c$, π_c — стратегия, реализующая данное умение, такая, что $\forall a \sim \pi_c, a \in A_c$, $c(e) = c$, $\beta_c : S \to \{0,1\}$ — это терминальная функция, завершающая данное умение. Каждое

умение $\kappa(c)$ задает типичную стратегию действия агента с любым объектом данного класса c. Умение может быть и одношаговым, когда $\beta_c\left(s_{t+1}\right)=1, \forall s_t \in I_c$.

Иерархическая формулировка объектного обучения с подкреплением с использованием умений приводит к следующей реализации двухуровневой стратегии. На верхнем уровне агент на каждом шаге определяет объект определенного класса, с которым он будет взаимодействовать, — это стратегия на умениях $\pi_\kappa : S \to C$. Второй уровень представляет собой стратегию π_c, которая задает последовательность действий с объектом класса c, реализуемую умением $\kappa(c)$. В простейшем случае можно считать, что для каждого класса объектов существует только одно умение.

Для иерархического случая уравнение Беллмана для стратегии на множестве умений практически не изменится за исключением учета длительности самого умения, равной τ:

$$\mathcal{B}_{\pi_\kappa}[V](s)=E_{\kappa(c)\sim\pi_\kappa, s'\sim T}\left[R\left(s,\kappa(c)\right)+\gamma^\tau V\left(s'\right)\middle| s\in I_c(\kappa)\right]. \tag{11}$$

Определение 3 (Частичная модель)**.** *Частной моделью называется модель* $m(c) =< T_c,R_c >$, *которая характеризует часть марковского процесса принятия решений задающего функции переходов и вознаграждений для конкретного класса объектов* c.

Оптимизационная задача для объектно-центричного обучения с подкреплением на основе модели будет включать в себя, помимо стандартной максимизации ожидаемой отдачи, $k+1$ подзадачи минимизации, отвечающие за поиск оператора Беллмана в рамках эквивалентных по полезности k частичных моделей и одной общей (для умений):

$$\begin{cases} \operatorname{argmax}_{\pi_\kappa} E\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,\kappa_t)\right], \\ \operatorname{argmin}_{\tilde{m}} \sum_{\pi_\kappa\in\mathcal{P}_\kappa}\sum_{V\in\mathcal{V}_\kappa}\left\|\mathcal{B}_{\pi_\kappa}[V]-\tilde{\mathcal{B}}_{\pi_\kappa}[V]\right\|, \\ \operatorname{argmin}_{\tilde{m}_{c_1}} \sum_{\pi_{c_1}\in\mathcal{P}_{c_1}}\sum_{V\in\mathcal{V}_{c_1}}\left\|\mathcal{B}_{\pi_{c_1}}[V]-\tilde{\mathcal{B}}_{\pi_{c_1}}[V]\right\|, \\ \dots \\ \operatorname{argmin}_{\tilde{m}_{c_k}} \sum_{\pi\in\mathcal{P}_{c_k}}\sum_{V\in\mathcal{V}_{c_k}}\left\|\mathcal{B}_{\pi_{c_k}}[V]-\tilde{\mathcal{B}}_{\pi_{c_k}}[V]\right\|. \end{cases} \tag{12}$$

Утверждение 1. $k+1$ *подзадачи минимизации ошибки в определении оператора Беллмана могут решаться параллельно с использованием одной и той же памяти прецедентов или траекторий* $\mathcal{D}$, *для каждого состояний которой применена функция выделения объектов* Φ.

Замечание 1 (о локальной организации)**.** *Полученный результат справедлив только в том случае, когда существует локальность эффектов каждого действия. Это означает, что изменение свойств одних объектов не*

сказывается на изменении свойств других. Это достаточно частный, но важный случай сред, пример которых будет рассмотрен ниже.

В качестве практической реализации подсчета близости модели к оптимальной эффективно использовать эмпирическую версию расстояния между предсказаниями полезности для всех состояний, которые встречаются для некоторой сгенерированной выборки:

$$\mathcal{L}\left(m^*,\tilde{m}\right) = \sum_{\pi\in\mathcal{P}}\sum_{V\in\mathcal{V}}\sum_{s\sim\mathcal{D}}\left[\frac{\sum_{\mathcal{D},s_i=s} R(s_i) + \gamma V(s_i^{'})}{N_{\mathcal{D}}(s_i)} - \tilde{\mathcal{B}}_\pi\left[V\right](s)\right], \qquad (13)$$

где $N_{\mathcal{D}}(s_i)$ — число вхождений состояния s_i в выборку $\mathcal{D}$.

Итоговый алгоритм обучения когнитивного агента с использованием объектного представления в архитектуре NSLP будет выглядеть следующим образом:

1. Инициализация верхнеуровневой стратегии агента π, объектных умений π_{c_i}, общей модели $\tilde{m}$ и частных моделей $\tilde{m}_{c_i}$.
2. Генерация дополнительного опыта агента $\mathcal{D}$, полученного в среде с использованием стратегий π и π_{c_i}.
3. Обновление общей и частных моделей с использованием уточненной оптимизационной задачи (12).
4. Решение задачи планирования (определения субоптимальных функций полезности для умений V^i и общей стратегии V^κ) по общей и частным моделям.
5. Использование «жадных» по полезностям V^i и V^κ стратегий в качестве новой высокоуровневой стратегии π и объектных умений π_{c_i}. Переход к шагу 2.

Раздел 2.4 посвящен вопросу использования языковых моделей для верхнеуровневого планирования в NSLP. Обсуждается применение предобученных больших языковых моделей, их возможностей при генерации плана поведения и необходимости организации обратной связи.

В **третьей главе** рассматриваются особенности реализации процессов обучения и использования модели в архитектуре NSLP. Основные результаты главы опубликованы в работах [2; 7; 10; 12; 18; 22; 28; 57; 67; 70; 77; 87; 94; 97]. **Раздел 3.1** посвящен особенностям интеграции модели среды в архитектуру семейства «актор-критик» при реализации процесса обучения архитектуры NSLP. Представлены экспериментальные результаты различных вариантов интеграции на модельных средах с обучением по наблюдениям, представленными изображениями. **Раздел 3.2** посвящен особенностям реализации архитектуры NSLP в задачах робототехнического управления. Рассматривается возможность генерации аналитического выражения, описывающего динамику собственно робота и использования ее в качестве модели мира в стандартном цикле «обучение-планирование».

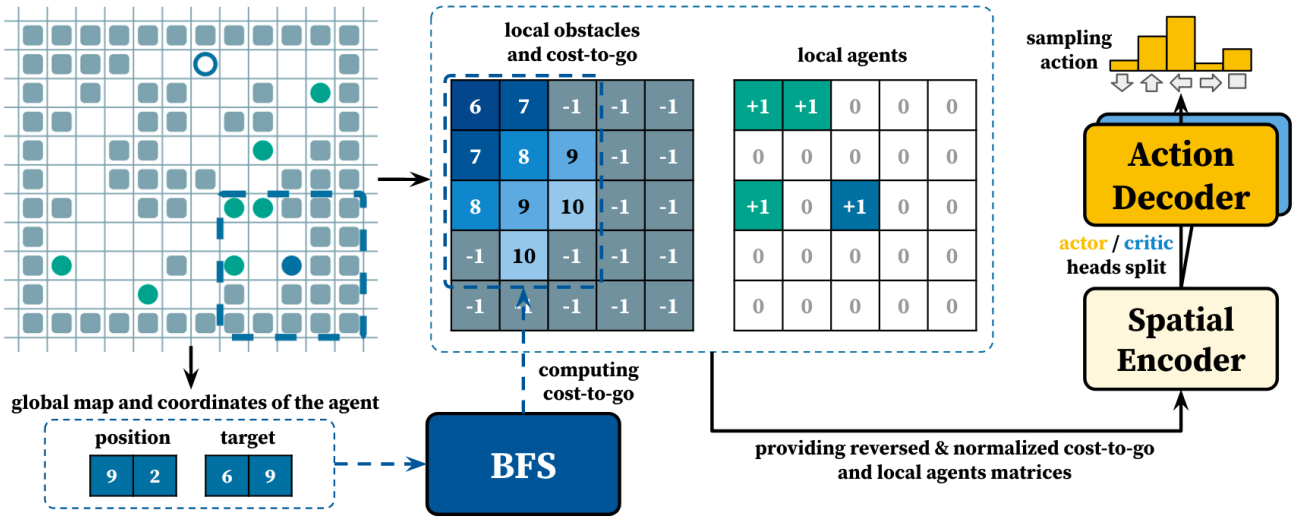

Рисунок 3 — На рисунке изображена схема алгоритма CostTracer.

Раздел 3.3 посвящен реализации для обучения и планирования в многоагентных средах. Рассматривается подход поиска по дереву с эвристическими функциями для сокращения перебора и метод поиска по дереву Монте-Карло с маскированием действий. Предлагаемый подход к безмодельному обучению называется CostTracer, что подчеркивает дизайн функции вознаграждения и входных данных для нейросетевого аппроксиматора. Он использует только две входные матрицы и простую функцию вознаграждения. Схематическое представление CostTracer показано на рисунке 3.

Схема многоагентного нейросетевого MCTS представлена на рисунке 4. Из-за того, что каждому агенту в среде доступна только частичная информация, использование централизованного планировщика невозможно. Чтобы иметь возможность планировать в таких ситуациях, предлагается использовать так называемый *внутренний МППР* (IMDP). Для этого создается внутренняя среда, основанная на эгоцентричном наблюдении агента (препятствия, другие агенты и их текущие цели). В эту среду включаются только те агенты, за которыми наблюдает текущий агент на текущем шаге. Все остальные клетки, которые не являются препятствиями, считаются пустыми.

Пусть множество удаленных агентов обозначается как $\mathbf{D}$, а пространство действий этих агентов ограничивается одним действием с наибольшей вероятностью, обозначаемым как $A^u_{\mathbf{D}} = \arg\max_{a_u \in A} \pi(o_u, a_u)$ (предсказывается с помощью CostTracer). Конечное количество переходов определяется путем перемножения немаскированных действий для каждого ближайшего агента.

Во время прямого поиска в таком МППР максимизируется совместное вознаграждение в размере r для всех агентов. Функция вознаграждения IMDP идентична функции вознаграждения для CostTracer. Каждый узел в дереве поиска соответствует внутреннему состоянию s IMDP. Для каждого совместного действия $\mathbf{j}$ из состояния s устанавливается ребро $(s, \mathbf{j})$ для хранения

набора статистических данных $\{N(s,\mathbf{j}), Q, r, \pi_{\mathbf{j}}\}$. Здесь N представляет количество посещений узла, Q — среднее совместная полезность действия, r — совместное вознаграждение, полученное от IMDP при выполнении действия $\mathbf{j}$, а π_j обозначает вероятность совместного действия $\mathbf{j}$. Примечательно, что здесь используется обозначение s для состояния IMDP.

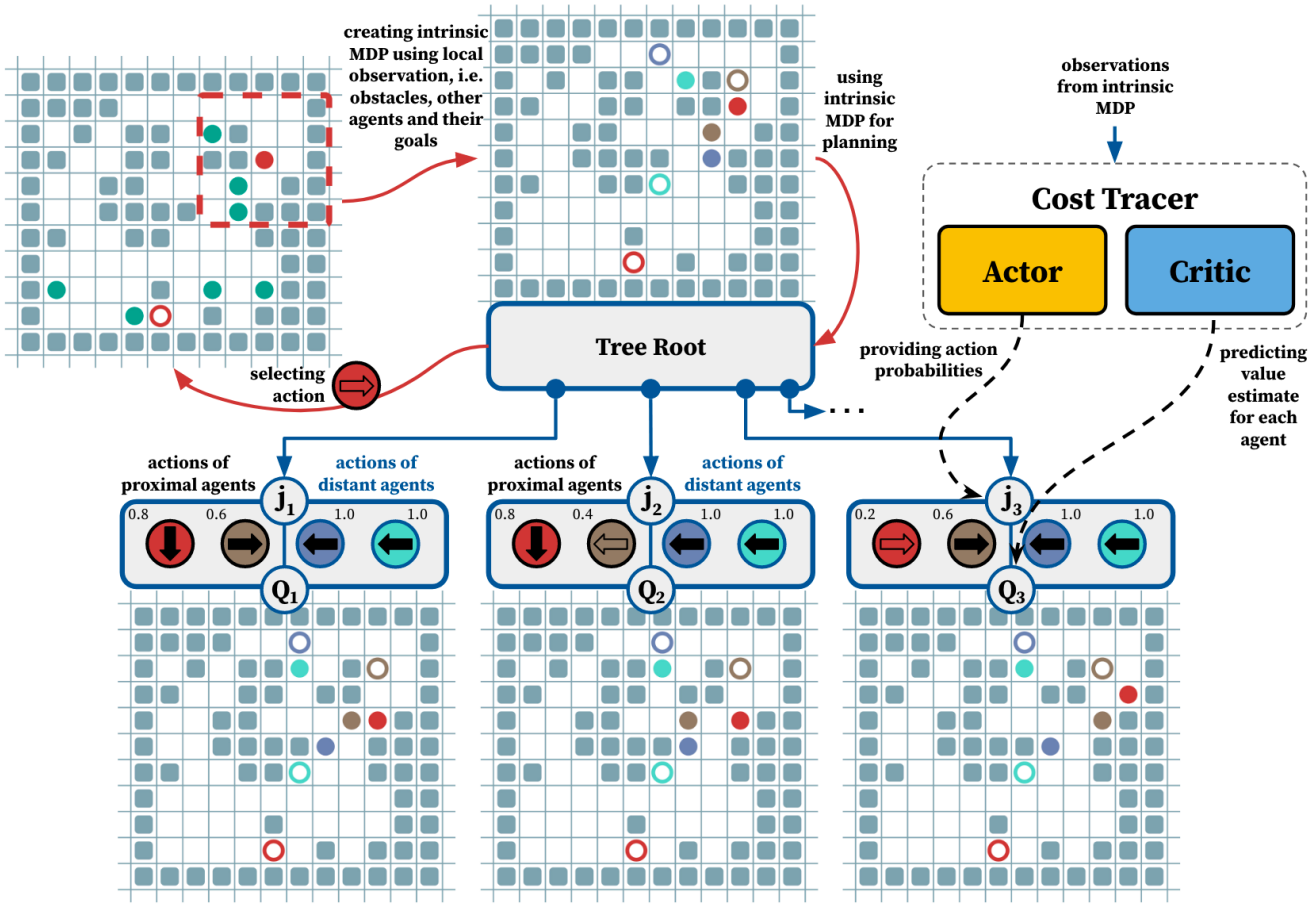

Рисунок 4 — Схема подхода MATS-LP.

Итоговое действие, выполняемое агентом в среде, определяется как действие, относящееся к наиболее исследованному ребру из корня дерева, определяемое в свою очередь количеством посещений $N(s,\mathbf{j})$. Действие a_u агента, для которого был создан IMDP, берется из $\mathbf{j}$. Окончательное совместное действие в глобальной среде формируется как действия всех эгоцентричных агентов, запланированные с помощью MCTS в их IMDP. После выполнения этого действия в среде каждый агент получает свои локальные наблюдения, заново воссоздает свой IMDP, и процесс повторяется.

Представлены экспериментальные результаты на классической задаче многоагентного поиска пути в среде POGEMA.

Раздел 3.4 посвящен вопросу реализации внутренней мотивации в архитектуре NSLP при исследовании среды и эффективном обучении в ней. Предложена реализация конкретных компонентов генерации внутреннего

вознаграждения и его использования при обновлении модели среды и стратегии агента.

В **четвертой главе** обсуждается реализация нейросимвольного механизма в архитектуре NSLP с помощью объектно-центричных представлений как в слотовом, так и в факторизованном вариантах. Основные результаты главы опубликованы в работах [8; 16; 17; 55; 58; 72]. В **разделе 4.1** посвящен описанию метода конструирования распутанных представлений на основе поддержания факторизации латентного пространства нейросетевого кодировщика. Описывается модель с использованием гиперразмерных векторов обладающих свойствами регуляризации представлений.

Раздел 4.2 освещается другой подход по формирования объектно-центричных представлений – метод обучения слотовых нейросетевых представлений объектных статических сцен. Модуль слотового внимания (SA) реализует итеративную процедуру с использованием внимания, предназначенную для сопоставления распределенной карты признаков $\mathbf{x} \in \mathbb{R}^{N \times D}$ с набором K слотов $\mathbf{s} \in \mathbb{R}^{K \times D}$. Слоты инициализируются случайным образом, а обучаемая матрица $q \in \mathbb{R}^{D \times D}$ используется для получения проекций запросов к слотам, в то время как матрицы $k, v \in \mathbb{R}^{D \times D}$ используются для получения векторов ключей и значений по карте признаков $\mathbf{x}$. Внимание, вычисляемое по скалярному произведению проекций q и k с помощью функции мягкого максимума (SoftMax) по q измерениям, подразумевает моделирование конкуренции между слотами за наибольший вклад в объяснение частей входных данных. Коэффициенты внимания $A \in \mathbb{R}^{N \times K}$ используются для присвоения v проекций слотам с помощью средневзвешенного значения:

$$M = \frac{1}{\sqrt{D}} k(\mathbf{x}) q(\mathbf{s})^T \in \mathbb{R}^{N \times K}, \quad A_{i,j} = \frac{e^{M_{i,j}}}{\sum_{j=1}^{K} e^{M_{i,j}}}, \tag{14}$$

$$W_{i,j} = \frac{\mathrm{A}_{i,j}}{\sum_{i=1}^{N} \mathrm{A}_{i,j}}, \quad \mathbf{s}^* = W^T v(\mathbf{x}) \ \in \mathbb{R}^{K \times D}. \tag{15}$$

Для уточнения значения слотов используется управляемый рекуррентный блок (gated recurrent unit, GRU), принимающий на входе предварительно обновленные представления слотов $\mathbf{s}$ в качестве скрытых состояний и обновленные слоты $\mathbf{s}^*$. Ключевой характеристикой слотового внимания является его инвариантность к перестановкам входных векторов и эквивариантность перестановок слотов. Это делает слотовое внимание подходящим методом для обработки множества свойств и построения объектно-центричных представлений.

Технически слотовое внимание является обучаемым аналогом алгоритма кластеризации k-средних с дополнительным обучаемым шагом обновления GRU и скалярным произведением (с обучаемыми проекциями q, k, v) вместо евклидова расстояния в качестве меры сходства между входными векторами

и центроидами кластера. В свою очередь, кластеризацию k-средних можно рассматривать как частный случай модели гауссовой смеси.

Модели смесей (ММ) — это класс параметрических вероятностных моделей, в которых предполагается, что каждый $\boldsymbol{x}_i$ из множества некоторых наблюдений $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \in \mathbb{R}^{N \times D}$ выбирается по распределению смеси с K компонентами смеси и априорными весами смеси $\boldsymbol{\pi} \in \mathbb{R}^K$:

$$\boldsymbol{x}_i \sim p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}_i|\boldsymbol{\theta}_k), \quad P(\boldsymbol{X}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta}), \quad \sum_k \pi_k = 1. \tag{16}$$

Такие модели можно рассматривать как модели со скрытыми переменными $z_{i,k} \in \{z_1, \ldots, z_K\}$, которые указывают, из какого компонента был получен $\boldsymbol{x}_i$. Основная задача состоит в том, чтобы найти такие K групп параметров компонентов $\boldsymbol{\theta}_k$ и отнесение компонентов к каждой выборке $\boldsymbol{x}_i$, которые максимизируют правдоподобие модели $P(\boldsymbol{X}|\boldsymbol{\theta})$. Для решения этой задачи обычно используется алгоритм максимизации математического ожидания (EM-алгоритм) — итеративный подход, включающий в себя два общих шага. Шаг **вычисления ожидания (E)** реализует подсчет математического ожидания значения полной вероятности $P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}^*)$ с учетом условного распределения $P(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta})$:

$$Q(\boldsymbol{\theta}^*, \boldsymbol{\pi}^*|\boldsymbol{\theta}, \boldsymbol{\pi}) = \mathbb{E}_{P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}[\log P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}^*)], \tag{17}$$

$$P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}^*) = \prod_{i=1}^{N} \prod_{k=1}^{K} [\pi_k p(\boldsymbol{x}_i|\boldsymbol{\theta}_k^*)]^{I(z_i = z_k)}, \tag{18}$$

где $I(*)$ — индикаторная функция.

Шаг **Максимизации (M)** реализует процедуру поиска значений $\boldsymbol{\theta}^*, \boldsymbol{\pi}^*$, которые максимизируют значение $Q(\boldsymbol{\theta}^*, \boldsymbol{\pi}^*|\boldsymbol{\theta}, \boldsymbol{\pi})$:

$$(\boldsymbol{\theta}, \boldsymbol{\pi}) = \text{argmax}_{(\boldsymbol{\theta}^*, \boldsymbol{\pi}^*)} Q(\boldsymbol{\theta}^*, \boldsymbol{\pi}^*|\boldsymbol{\theta}, \boldsymbol{\pi}). \tag{19}$$

Одной из наиболее широко используемых моделей такого рода является модель гауссовой смеси (GMM), где каждый компонент смеси моделируется как гауссово распределение, параметризованное его средними значениями и ковариационной матрицей, которая в простейшем случае является диагональной: $P(\boldsymbol{x}_i|\boldsymbol{\theta}_k) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, $\boldsymbol{\Sigma}_k = \text{diag}(\boldsymbol{\sigma}_k^2)$. В этом случае EM-алгоритм сводится к следующим вычислениям:

E шаг :

$$p(z_k|\boldsymbol{x}_i) = \frac{p(z_k)p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)}{\sum_{k=1}^{K} p(z_k)p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} = \gamma_{k,i}. \tag{20}$$

M шаг :

$$\pi_k^* = \frac{\sum_{i=1}^{N} \gamma_{k,i}}{N}, \quad \boldsymbol{\mu}_k^* = \frac{\sum_{i=1}^{N} \gamma_{k,i} \boldsymbol{x}_i}{\sum_{i=1}^{N} \gamma_{k,i}}, \quad \boldsymbol{\Sigma}_k^* = \frac{\sum_{i=1}^{N} \gamma_{k,i} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^*)(\boldsymbol{x}_i - \boldsymbol{\mu}_k^*)^T}{\sum_{i=1}^{N} \gamma_{k,i}}. \tag{21}$$

Ключевое различие между моделью гауссовой смеси и кластеризацией k-средних заключается в том, что GMM учитывает не только центры кластеров, но и расстояние между кластерами и принадлежащими им векторами с априорными вероятностями для каждого кластера.

Для объектно-центричного обучения предлагается модифицированный подход с использованием модели гауссовой смеси, называемый модулем смешивания слотов (SMM). Этот модуль использует из GMM шаги **E** и **M** для сопоставления карт признаков, получаемых из сверточного нейросетевого кодировщика (CNN), с набором векторных представлений слотов, где слоты представляют собой объединение средних значений распределений и диагонали ковариационной матрицы.

Как и слотовое внимание, SMM задействует GRU, который использует текущие и предыдущие средние значения в качестве входных данных и скрытых состояний. В слотовом внимании представления слотов являются центрами кластеров, поэтому этот подход ограничен информацией, содержащейся и представленной в этих центрах кластеров. Рассмотрение не только средневзвешенного значения группы векторов в качестве ее представления дает преимущество в различении схожих, но все-таки разных групп. Например, два набора векторов, отобранных из разных распределений, но с одинаковым математическим ожиданием, были бы неразличимы, если бы их среднее значение было единственным представлением этих наборов.

Этот набор слотов в дальнейшем используется для решения конкретной (downstream) задачи. Также применяется тот же дополнительный шаг обновления нейронной сети для средних значений перед обновлением значений матрицы ковариаций:

$$\boldsymbol{\mu}_k = \mathrm{RNN}(\mathrm{input}{=}\boldsymbol{\mu}_k^*,\ \mathrm{hidden}{=}\boldsymbol{\mu}_k), \quad \boldsymbol{\mu}_k = \mathrm{MLP}(\mathrm{LayerNorm}(\boldsymbol{\mu}_k)) + \boldsymbol{\mu}_k. \quad (22)$$

Эти два шага необходимы для решения последующей задачи, связывая внешнюю и внутреннюю модели. Внутренняя модель (шаги **E** и **M** в SMM) пытается обновить свои параметры $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ таким образом, чтобы входные векторы x были назначены слотам с максимальным правдоподобием. Напротив, внешняя модель принимает эти параметры в качестве входных данных.

Далее в данном разделе представлено экспериментальное исследование на таких наборах данных как CLEVER, Emoji.

Раздел 4.3 посвящен использованию факторизованных моделей мира для задачи обучения с подкреплением в контексте архитектуры NSLP. На основе базового алгоритма обучения «актор-критик» предлагается использовать графовую факторизацию модели мира для разделения причинно-следственных связей, влияющих на стратегию агента. В данном разделе в качестве базового алгоритма обучения с подкреплением используется так называемый мягкий актор-критик (SAC), современный алгоритм обучения

подкреплением на основе отложенного опыта для формирования стратегии с использованием непрерывных действий.

Целевая функцию при обучении этого алгоритма направлена на поиск стратегии, которая максимизирует энтропийный целевой функционал:

$$\pi^* = \arg\max_{\pi} \sum_{i=0}^{\tau} \mathbb{E}_{(s_t,a_t)\sim d_\pi}\left[\gamma^t (R(s_t,a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t))\right], \tag{23}$$

где α — параметр температуры, $\mathcal{H}(\pi(\cdot|s_t)) = -\log \pi(\cdot|s_t)$ — энтропия стратегии π в состоянии s_t, d_π — распределение траекторий, индуцированное стратегией π.

Функция мягкой полезности действий $Q_\theta(s_t, a_t)$, параметризованная с использованием нейронной сети с параметрами θ, обучается путем минимизации сглаженного оператора Беллмана:

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim D}\left[\left(Q_\theta(s_t,a_t) - R(s_t,a_t) - \gamma \mathbb{E}_{s_{t+1}\sim T(s_t,a_t)} V_{\bar{\theta}}(s_{t+1})\right)^2\right], \tag{24}$$

где D — память прецедентов взаимодействия агента со средой (прошлый опыт агента), а $V_{\bar{\theta}}(s_{t+1})$ оценивается с использованием целевой сети для функции Q и Монте-Карло оценки сглаженной функции полезности состояния после выборки прецедентов из памяти D.

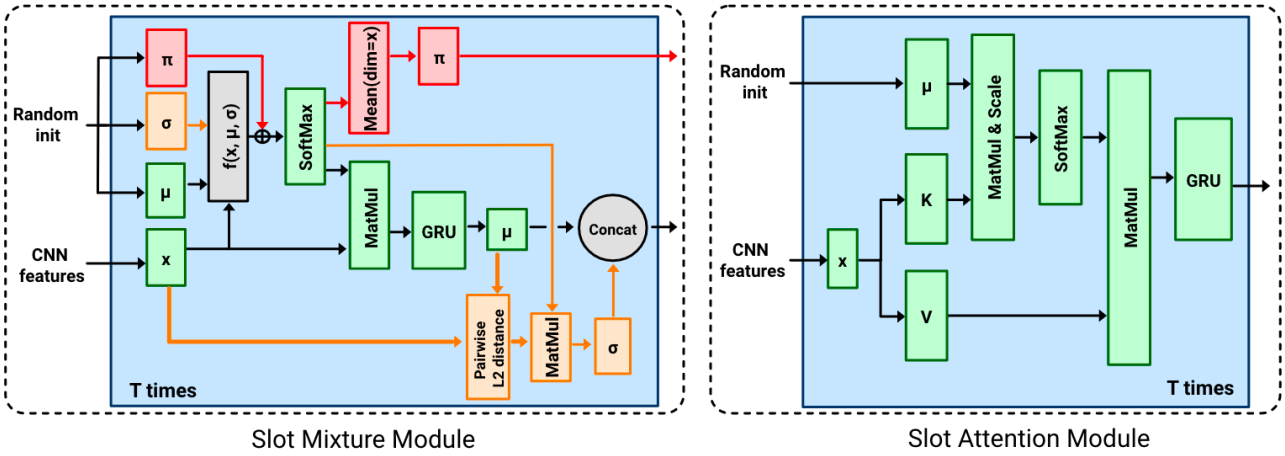

Рисунок 5 — Сравнение архитектур предлагаемого модуля смеси слотов (SMM) и модуля слотового внимания.

Стратегия π параметризуется с использованием нейронной сети с параметрами ϕ. Параметры обучаются путем минимизации ожидаемого расстояния Кульбака-Лейблера (KL-дивергенции) между стратегией и экспонентой от Q-функции:

$$J_\pi(\phi) = \mathbb{E}_{s_t\sim D}\left[\mathbb{E}_{a_t\sim\pi_\phi(\cdot|s_t)}\left[\alpha \log(\pi_\phi(a_t|s_t)) - Q_\theta(s_t,a_t)\right]\right]. \tag{25}$$

Целевая функция для параметра температуры задается с помощью следующего выражения:

$$J(\alpha) = \mathbb{E}_{a_t\sim\pi(\cdot|s_t)}\left[-\alpha(\log \pi(a_t|s_t) + \bar{H})\right], \tag{26}$$

где $\bar{H}$ — гиперпараметр, интерпретируемый как целевая энтропия. На практике поддерживаются две отдельных обучаемых сглаженных Q-сети, а затем минимальное значение, полученное от этих двух аппроксиматоров, используется в качестве выходного результата подсчета сглаженной Q-сети.

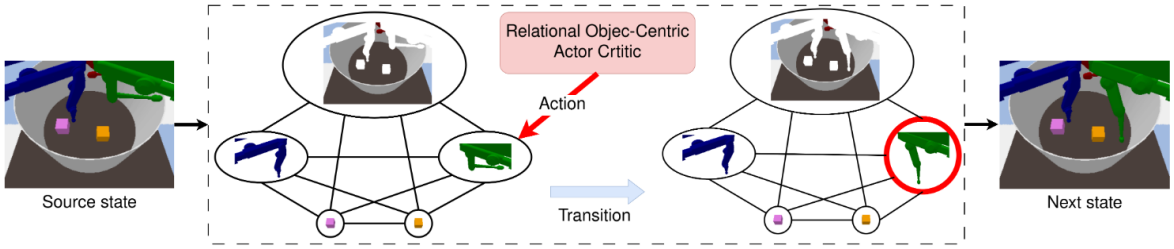

Рисунок 6 — Схема работы предлагаемого подхода ROCA, в котором модель формирует стратегию, извлекая объектно-центричные представления из исходного изображения и обрабатывая их в виде полного графа.

В то время как оригинальная версия SAC решает задачи с непрерывным пространством действий, предлагается использовать версию для дискретных пространств действий. В случае дискретного пространства действий стратегия $\pi_\phi(a_t|s_t)$ выдает вероятность для действий вместо плотности вероятности. Такая параметризация стратегии незначительно изменяет целевые функционалы (24), (25) и (26).

На рисунке 6 представлена верхнеуровневая схема работы предлагаемой архитектуры актора-критика (ROCA). В качестве кодировщика предлагается использовать SLATE, недавно представленную объектно-центричную модель. SLATE включает в себя дискретный вариационный кодировщик для извлечения представлений объектов, GPT-подобную трансформерную модель для декодирования и модуль слотового внимания для группировки признаков, связанных с одним и тем же объектом. В ROCA предварительно обученная модель SLATE с фиксированными параметрами принимает в качестве входных данных наблюдение в виде изображения s_t и создает набор объектных векторов, называемых слотами, $\boldsymbol{z}_t = (z_t^1, \ldots, z_t^K)$ (K — максимальное количество извлекаемых объектов).

Эффективность работы предложенного алгоритма ROCA была продемонстрирована в трехмерной робототехнической симуляционной среде CausalWorld для задачи достижения объекта и в синтетической двумерной среде Shapes2D для задач навигации и вытеснения без агента.

В **разделе 4.4** рассматривается использование языковых моделей в качестве планировщиков при объектно-центричной постановке задачи в таких средах как IGLU и Crafter. Обсуждаются особенности их интеграции в архитектуру NSLP с необходимой декомпозицией задачи.

В **пятой главе** описывается нескольких прикладных задач, для решения которых был использован алгоритмический инструментарий, созданный при

реализации моделей и методов, включенных в архитектуру NSLP. Основные результаты главы опубликованы в работах [3; 9; 27; 29; 30; 49; 73; 75; 76].

Раздел 5.1 фокусируется на реализации ряда компонентов архитектуры NSLP в программной библиотеке STRL-Robotics, основанной на операционной системе ROS. Архитектура STRL была изначально предложена для координации действий группы интеллектуальных агентов, например, беспилотных летательных аппаратов (БПЛА) [13]. Специфика мобильных манипуляторов по сравнению с БПЛА состоит в том, что они представляют собой композицию двух механических систем (платформы и манипулятора), которые могут управляться независимо. В диссертации предполагается, что мобильная платформа применяется для перемещения робототехнической системы в человеко-ориентированной среде, а манипулятор — для взаимодействия с объектами человеко-ориентированной среды. Впоследствии такая схема может быть расширена на задачи, в которых при взаимодействии с объектами среды используется не только манипулятор, но и мобильная платформа.

Общая схема взаимодействия компонентов системы управления при обеспечении мобильности робота в человеко-ориентированной среде приведена на рисунке 7. Управление движением как платформы, так и манипулятора должно учитывать данные от сенсоров системы, которые преобразуются алгоритмами компьютерного зрения в представление большего уровня абстракции на тактическом уровне модели STRL-Robotics (построение сенсорной ситуации). Если в [88] это преобразование использовалось только для задач навигации, то настоящей работе добавляется также задача распознавания и позиционирования объектов среды, с которыми взаимодействует коллаборативная робототехническая система. Определение действий, выполняемых робототехнической системой, осуществляется отдельно на тактическом (определение геометрического пути, который должна пройти система для достижения поставленной цели) и на реактивном (определение управляющих команд, обеспечивающих движение системы по геометрическому пути) уровнях модели.

В диссертации подробно рассмотрен частный пример такого взаимодействия: использование лифта. Распознаваемыми объектами в этом случае будут кнопки лифта, а движения манипулятора планируются таким образом, чтобы нажимать нужные кнопки.

Раздел 5.2 посвящен реализации архитектуры NSLP в задаче визуальной навигации робототехнической платформы внутри помещений. Сначала ставится собственно задача генерации маневров мобильного робота для достижения конкретных объектов, описываются существующие подходы и рассматривается реализация верхнеуровневой стратегии NSLP на базе метода интеграции умений, сочетающего классические подходы по планированию перемещения и обучаемые стратегии. **Раздел 5.3** посвящен реализации

элементов архитектуры NSLP в задаче генерации маневров беспилотного автомобиля на основе программной платформы Apollo. Представлен пример реализации подхода в задаче генерации маневра парковки.

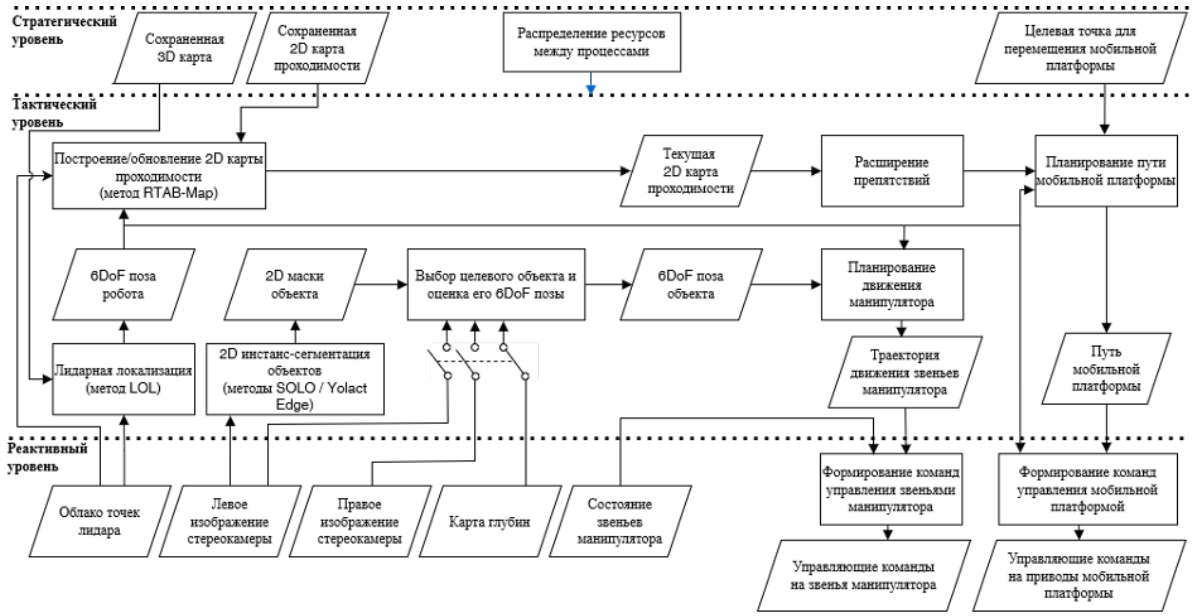

Рисунок 7 — Взаимодействие программных модулей системы в рамках предлагаемой архитектуры STRL-Robotics.

В **шестой главе** описываются когнитивные аспекты реализации архитектуры NSLP и, в частности, семиотического механизма реализации нейросимвольной интеграции. Основные результаты главы опубликованы в работах [24; 45; 69; 78]. **Раздел 6.1** посвящен теории знаковой картины мира, реализующей семиотический механизм нейросимвольной интеграции. Рассмотрена архитектура семиотического агента как варианта нейросимвольной архитектуры NSLP, проанализирована проблема привязки символов к сенсомоторным данным, представлены сценарии поведения в картине мира и дан модельный пример работы семиотического агента. Наконец, в **разделе 6.2** исследуются когнитивные особенности реализации алгоритма планирования с использованием языковых моделей. В начале раздела дано краткое описание психологических подходов к моделированию мышления и речи, а затем описывается психологический эксперимент, на основе которого возможно проанализировать особенности процесса генерации плана языковой моделью.

Заключение

В диссертации предложены и теоретически обоснованы новые математические модели, методы и алгоритмы генерации поведения когнитивного агента в динамической среде. В данной работе впервые предложен единый подход к проблеме привязки символов за счет разработки методов нейросимвольной интеграции в обучении и планировании.

Основные результаты работы заключаются в следующем:

1. Была разработана нейросимвольная архитектура управления поведением когнитивного агента, включающая в себя компоненты одновременного планирования и обучения, а также компонент концептуального планирования с использованием языковых моделей.
2. Были разработаны модели и методы интеграции планирования и обучения с подкреплением, в том числе с использованием модели среды, для решения сложных визуальных и векторных задач управления поведением когнитивным агентом, в том числе в многоагентной постановке.
3. Были созданы модели и методы объектно-центричного подхода к представлению сенсорной информации о статических сценах для использования в нейросимвольной архитектуре управления поведением когнитивного агента.
4. Были разработаны модели и методы объектно-центричного обучения с подкреплением с использованием динамической модели среды для интеграции планирования и обучения в нейросимвольной архитектуре управления поведением когнитивного агента.
5. Был усовершенствован ряд существующих моделей и методов обучения с подкреплением на основе модели мира, с моделями внутренней мотивации и с использованием эвристических планировщиков.
6. Была разработана программная реализация системы управления робототехническими платформами с использованием языковых моделей для подзадачи планирования.

Разработан программно-алгоритмический инструментарий, основанный, в том числе на полученных теоретических результатах, для решения задачи генерации действий робототехнической платформой в сложной динамической среде, позволяющий использовать как обучаемые компоненты, так и классические планировочные. Создана экспериментальная программная реализация элементов данного инструментария, использующаяся для решения практических задач управления поведением. Продемонстрировано использование разработанных моделей и методов одновременного планирования и обучения в ряде практически важных робототехнических задачах: навигация мобильной платформы внутри помещений, адаптивное планирование маневров беспилотным транспортным средством, перемещение и манипуляция объектами мобильной платформы по языковым инструкциям.

Разработанные в рамках диссертации методы и алгоритмы использовались для решения следующих прикладных задач:

– автоматическое планирование маршрута и траектории движения функционирующего в составе комплекта аппаратуры управления транспортного средства;

– одновременные локализация и картирование местности для мобильных роботов, функционирующих в разделяемой с людьми среде;
– построение динамической карты проходимости и планирование на ее основе движения мобильных наземных роботов;
– автоматическое управление движением автомобильного транспортного средства в условиях дорог общего пользования;
– управление роботом с модулем планирования по языковым инструкциям для сортировки объектов в помещении с использованием мобильного робота и манипулятора.

Перспективы дальнейшего развития текущих исследований. В качестве одного из основных направлений необходимо указать разработку более эффективных методов объектно-центричного представления информации и объектно-центричных методов обучения с подкреплением на основе модели среды, функционирующие в более сложных и реалистичных средах. Также требуется усовершенствование представленного программно-аппаратного комплекса для более глубокой интеграции с описанными в диссертации гибридными методами обучения с подкреплением и планирования.

В целях дальнейшей проработки нейросимвольного уровня архитектуры NSLP необходимо создавать новые мультимодальные модели привязки действий к подцелям на основе имеющихся моделей привязки текстовых описаний к изображениям (CLIP, R3M).Следуя полученным в данном диссертационном исследовании результатам, такие модели также должны быть объектно-центричными.

Наконец, для улучшения работы концептуального уровня архитектуры NSLP необходимо дообучать большие языковые модели используемые для генерации гипотез, подаваемых планировщику. Известные работы в области дообучения на заранее собранных данных с интерпретируемой как вознаграждение разметкой качества (RLHF, DPO) генерируемых ответов модели должны быть расширены на режим дообучения в интерактивной среде. Многообещающее направление работы задают исследования в области обучения мультимодальных моделей мира (Dynalang).

Публикации автора по теме диссертации

1. Applying Vector Symbolic Architecture and Semiotic Approach to Visual Dialog / A. Panov [et al.] // Hybrid Artificial Intelligent Systems. HAIS 2021. Lecture Notes in Computer Science. Vol. 12886 / ed. by H. S. González [et al.]. — 2021. — P. 243—255.
2. Decentralized Monte Carlo Tree Search for Partially Observable Multi-agent Pathfinding / A. Panov [et al.] // Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 38 (AAAI). — 2024. — P. 17531—17540.
3. Fine-tuning Multimodal Transformer Models for Generating Actions in Virtual and Real Environments / A. Panov [et al.] // IEEE Access. — 2023. — Vol. 11. — P. 130548—130559.

4. Forgetful experience replay in hierarchical reinforcement learning from expert demonstrations / A. Panov [et al.] // Knowledge-Based Systems. — 2021. — Vol. 218. — P. 106844. — Publisher: Elsevier B.V.

5. Hierarchical Deep Q-Network from imperfect demonstrations in Minecraft / A. Panov [et al.] // Cognitive Systems Research. — 2021. — Vol. 65. — P. 74—78.

6. Hierarchical intrinsically motivated agent planning behavior with dreaming in grid environments / A. Panov [et al.] // Brain Informatics. — 2022. — Vol. 9, no. 1. — P. 8.

7. Hybrid Policy Learning for Multi-Agent Pathfinding / A. I. Panov [et al.] // IEEE Access. — 2021. — Vol. 9. — P. 126034—126047.

8. Interactive Grounded Language Understanding in a Collaborative Environment: Retrospective on IGLU 2022 Competition / A. Panov [et al.] // Proceedings of the NeurIPS 2022 Competitions Track, PMLR. — 2023. — Vol. 220. — P. 204—216.

9. Interactive Semantic Map Representation for Skill-Based Visual Object Navigation / A. Panov [et al.] // IEEE Access. — 2024. — Vol. 12. — P. 44628—44639.

10. Learn to Follow: Decentralized Lifelong Multi-Agent Pathfinding via Planning and Learning / A. Panov [et al.] // Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). — 2024.

11. Learning embodied agents with policy gradients to navigate in realistic environments / A. Panov [et al.] // Advances in Neural Computation, Machine Learning, and Cognitive Research IV. NEUROINFORMATICS 2020. Studies in Computational Intelligence. Vol. 925 / ed. by B. Kryzhanovsky [et al.]. — Springer International Publishing, 2021. — P. 212—221.

12. Monte-Carlo Tree Search for Multi-Agent Pathfinding: Preliminary Results / A. Panov [et al.] // Hybrid Artificial Intelligent Systems. HAIS 2023. Lecture Notes in Computer Science. Vol. 14001 (HAIS 2023). — Springer Cham, 2023. — P. 649—660. — (Lecture Notes in Computer Science).

13. Multilayer cognitive architecture for UAV control / A. I. Panov [et al.] // Cognitive Systems Research. — 2016. — Vol. 39. — P. 58—72.

14. Neural Potential Field for Obstacle-Aware Local Motion Planning / A. Panov [et al.] // 2024 IEEE International Conference on Robotics and Automation (ICRA) (ICRA 2024). — 2024. — P. 9313—9320.

15. Object Detection with Deep Neural Networks for Reinforcement Learning in the Task of Autonomous Vehicles Path Planning at the Intersection / A. Panov [et al.] // Optical Memory and Neural Networks. — 2019. — Vol. 28, no. 4. — P. 283—295.

16. Object-Centric Learning with Slot Mixture Module / A. Panov [et al.] // The Twelfth International Conference on Learning Representations (ICLR 2024). — 2024.

17. *Panov, A. I.* Symbolic Disentangled Representations in Hyperdimensional Latent Space / A. I. Panov, A. Korchemniy, A. Kovalev // ICLR NeSy-GeMs Workshop (ICLR 2023). — 2023.

18. *Panov, A. I.* A World Model for Actor—Critic in Reinforcement Learning / A. I. Panov, L. Ugadiarov // Pattern Recognition and Image Analysis. — 2023. — Vol. 33, no. 3. — P. 467—477.

19. *Panov, A.* Transfer Learning with Demonstration Forgetting for Robotic Manipulator / A. Panov, E. Aitygulov // Procedia Computer Science. — 2021. — Vol. 186. — P. 374—380.

20. *Panov, A.* Task and Spatial Planning by the Cognitive Agent with Human-like Knowledge Representation / A. Panov, E. Aitygulov, G. Kiselev // Interactive Collaborative Robotics. ICR 2018. Lecture Notes in Computer Science. Vol. 11097 / ed. by A. Ronzhin, G. Rigoll, R. Meshcheryakov. — Springer, 2018. — P. 1—12.

21. *Panov, A.* Approximation Methods for Monte Carlo Tree Search / A. Panov, K. Aksenov // Proceedings of the Fourth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'19). IITI'19 2019. Advances in Intelligent Systems and Computing. Vol. 1156 / ed. by S. Kovalev [et al.]. — Springer International Publishing, 2020. — P. 68—74.

22. *Panov, A.* Policy Optimization to Learn Adaptive Motion Primitives in Path Planning With Dynamic Obstacles / A. Panov, B. Angulo, K. Yakovlev // IEEE Robotics and Automation Letters. — 2023. — Vol. 8, no. 2. — P. 824—831.

23. *Panov, A.* Task Planning in "Block World" with Deep Reinforcement Learning / A. Panov, E. Ayunts // Biologically Inspired Cognitive Architectures (BICA) for Young Scientists. BICA 2017. Advances in Intelligent Systems and Computing. Vol. 636 / ed. by A. V. Samsonovich, V. V. Klimov. — Springer, 2018. — P. 3—9.

24. *Panov, A.* The Problem of Concept Learning and Goals of Reasoning in Large Language Models / A. Panov, A. A. Chuganskaya, A. K. Kovalev // Hybrid Artificial Intelligent Systems. HAIS 2023. Lecture Notes in Computer Science. Vol. 14001 (HAIS 2023). — Springer, Cham, 2023. — P. 661—672. — (Lecture Notes in Computer Science).

25. *Panov, A.* Self and Other Modelling in Cooperative Resource Gathering with Multi-Agent Reinforcement Learning / A. Panov, V. Davydov, T. Liusko // Brain-Inspired Cognitive Architectures for Artificial Intelligence: BICA*AI 2020. Advances in Intelligent Systems and Computing. Vol. 1310 / ed. by A. V. Samsonovich, R. R. Gudwin, A. d. S. Simões. — Springer International Publishing, 2021. — P. 69—77.

26. *Panov, A.* Applying a Neural Network Architecture with Spatio-Temporal Connections to the Maze Exploration / A. Panov, D. Filin // Biologically Inspired Cognitive Architectures (BICA) for Young Scientists. BICA 2017. Advances in Intelligent Systems and Computing. Vol. 636 / ed. by A. V. Samsonovich, V. V. Klimov. — Springer, 2018. — P. 57—64.

27. *Panov, A.* Learning Adaptive Parking Maneuvers for Self-driving Cars / A. Panov, G. Gorbov, M. Jamal // Proceedings of the Sixth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'22). IITI 2022. Lecture Notes in Networks and Systems. Vol. 566 (IITI 2022) / ed. by S. Kovalev [et al.]. — 2023. — P. 283—292.

28. *Panov, A.* Model-based Policy Optimization with Neural Differential Equations for Robotic Arm Control / A. Panov, A. Gorodetskiy, K. Mironov // Interactive Collaborative Robotics. ICR 2023. Lecture Notes in Computer Science. Vol. 14214 (ICR 2023). — 2023. — P. 258—266.

29. *Panov, A.* Application of Reinforcement Learning in Open Space Planner for Apollo Auto / A. Panov, D. Ivanov // Proceedings of the Fifth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'21). IITI 2021. Lecture Notes in Networks and Systems. Vol. 330 / ed. by S. Kovalev [et al.]. — Springer, 2022. — P. 35—43.

30. *Panov, A.* Adaptive Maneuver Planning for Autonomous Vehicles Using Behavior Tree on Apollo Platform / A. Panov, M. Jamal // Artificial Intelligence XXXVIII. SGAI 2021. Lecture Notes in Computer Science. Vol. 13101 / ed. by M. Bramer, R. Ellis. — 2021. — P. 327—340.

31. *Panov, A.* Sign-based Approach to the Task of Role Distribution in the Coalition of Cognitive Agents / A. Panov, G. A. Kiselev // SPIIRAS Proceedings. — 2018. — No. 57. — P. 161—187.

32. *Panov, A.* Hierarchical Psychologically Inspired Planning for Human-Robot Interaction Tasks / A. Panov, G. Kiselev // Interactive Collaborative Robotics. ICR 2019. Lecture Notes in Computer Science. Vol. 11659 / ed. by A. Ronzhin, G. Rigoll, R. Meshcheryakov. — Springer, 2019. — P. 150—160.

33. *Panov, A.* Q-learning of Spatial Actions for Hierarchical Planner of Cognitive Agents / A. Panov, G. Kiselev // Interactive Collaborative Robotics. ICR 2020. Lecture Notes in Computer Science. Vol. 12336 / ed. by A. Ronzhin, G. Rigoll, R. Meshcheryakov. — Springer International Publishing, 2020. — P. 160—169.

34. *Panov, A.* Spatial reasoning and planning in sign-based world model / A. Panov, G. Kiselev, A. Kovalev // Artificial Intelligence. RCAI 2018. Communications in Computer and Information Science. Vol. 934 / ed. by S. Kuznetsov, G. S. Osipov, V. Stefanuk. — Springer, 2018. — P. 1—10.

35. *Panov, A.* Synthesis of the Behavior Plan for Group of Robots with Sign Based World Model / A. Panov, G. A. Kiselev // Interactive Collaborative Robotics. ICR 2017. Lecture Notes in Computer Science. Vol. 10459 / ed. by A. Ronzhin, G. Rigoll, R. Meshcheryakov. — Springer, 2017. — P. 83—94.

36. *Panov, A.* Mental Actions and Modelling of Reasoning in Semiotic Approach to AGI / A. Panov, A. K. Kovalev // Artificial General Intelligence. AGI 2019. Lecture Notes in Computer Science. Vol. 11654 / ed. by P. Hammer [et al.]. — Springer, 2019. — P. 121—131.

37. *Panov, A.* Hyperdimensional Representations in Semiotic Approach to AGI / A. Panov, A. K. Kovalev, E. Osipov // Artificial General Intelligence. AGI 2020. Lecture Notes in Computer Science. Vol. 12177. — Springer, 2020. — P. 231—241.

38. *Panov, A.* Hierarchical Reinforcement Learning with Options and United Neural Network Approximation / A. Panov, V. Kuzmin // Proceedings of the Third International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'18). IITI'18 2018. Advances in Intelligent Systems and Computing. Vol. 874 / ed. by A. Abraham [et al.]. — Springer, 2019. — P. 453—462.

39. *Panov, A.* Navigating Autonomous Vehicle at the Road Intersection Simulator with Reinforcement Learning / A. Panov, M. Martinson, A. Skrynnik // Artificial Intelligence. RCAI 2020. Lecture Notes in Computer Science. Vol. 12412 / ed. by S. O. Kuznetsov, A. I. Panov, K. S. Yakovlev. — Springer International Publishing, 2020. — P. 71—84.

40. *Panov, A.* Planning Maneuvers for Autonomous Driving Based on Offline Reinforcement Learning: Comparative Study / A. Panov, M. Melkumov // Proceedings of the Seventh International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'23). IITI 2023. Lecture Notes in Networks and Systems. Vol. 776 (IITI 2023). — Springer, Cham, 2023. — P. 65—74.

41. *Panov, A.* Hierarchical Temporal Memory with Reinforcement Learning / A. Panov, E. Nugamanov // Procedia Computer Science. — 2020. — Vol. 169. — P. 123—131.

42. *Panov, A.* Behavior control as a function of consciousness. I. World model and goal setting / A. Panov, G. S. Osipov, N. V. Chudova // Journal of Computer and Systems Sciences International. — 2014. — Vol. 53, no. 4. — P. 517—529.

43. *Panov, A.* Behavior Control as a Function of Consciousness. II. Synthesis of a Behavior Plan / A. Panov, G. S. Osipov, N. V. Chudova // Journal of Computer and Systems Sciences International. — 2015. — Vol. 54, no. 6. — P. 882—896.

44. *Panov, A.* Relationships and Operations in a Sign-Based World Model of the Actor / A. Panov, G. S. Osipov // Scientific and Technical Information Processing. — 2018. — Vol. 45, no. 5. — P. 317—330.

45. *Panov, A.* Planning Rational Behavior of Cognitive Semiotic Agents in a Dynamic Environment / A. Panov, G. S. Osipov // Scientific and Technical Information Processing. — 2021. — Vol. 48, no. 6. — P. 502—516.

46. *Panov, A.* Flexible Data Augmentation in Off-Policy Reinforcement Learning / A. Panov, A. Rak, A. Skrynnik // Artificial Intelligence and Soft Computing. ICAISC 2021. Lecture Notes in Computer Science. Vol. 12854 / ed. by L. Rutkowski. — Springer, Cham, 2021. — P. 224—235.

47. *Panov, A.* Hierarchical Reinforcement Learning Approach for the Road Intersection Task / A. Panov, M. Shikunov // Biologically Inspired Cognitive Architectures 2019. BICA 2019. Advances in Intelligent Systems and Computing. Vol. 948 / ed. by A. V. Samsonovich. — Springer, 2020. — P. 495—506.

48. *Panov, A.* Hierarchical Reinforcement Learning with Clustering Abstract Machines / A. Panov, A. Skrynnik // Artificial Intelligence. RCAI 2019. Communications in Computer and Information Science. Vol. 1093 / ed. by S. O. Kuznetsov, A. I. Panov. — Springer, 2019. — P. 30—43.

49. *Panov, A.* Hierarchical Landmark Policy Optimization for Visual Indoor Navigation / A. Panov, A. Staroverov // IEEE Access. — 2022. — Vol. 10. — P. 70447—70455.

50. *Panov, A.* Hierarchical Actor-Critic with Hindsight for Mobile Robot with Continuous State Space / A. Panov, A. Staroverov // Advances in Neural Computation, Machine Learning, and Cognitive Research III. Studies in Computational Intelligence. Vol. 856 / ed. by B. Kryzhanovsky [et al.]. — Springer, 2020. — P. 62—70.

51. *Panov, A.* Long-Term Exploration in Persistent MDPs / A. Panov, L. Ugadiarov, A. Skrynnik // Advances in Soft Computing. MICAI 2021. Part I. Lecture Notes in Computer Science. Vol. 13067 / ed. by I. Batyrshin, A. Gelbukh, G. Sidorov. — Springer, 2021. — P. 108—120.

52. *Panov, A.* Toward Faster Reinforcement Learning for Robotics : Using Gaussian Processes / A. Panov, A. Younes // RAAI Summer School 2019. Lecture Notes in Computer Science. Vol. 11866 / ed. by G. S. Osipov, A. I. Panov, K. S. Yakovlev. — Springer, 2019. — P. 160—174.

53. *Panov, A.* Sequential Contrastive Learning to Master Effective Representations For Reinforcement Learning and Control / A. Panov, A. Younes // Russian Advances in Artificial Intelligence 2020. RAAI 2020. CEUR Workshop Proceedings. Vol. 2648 / ed. by O. P. Kuznetsov [et al.]. — 2020. — P. 111—121.

54. *Panov, A.* Case-based Task Generalization in Model-based Reinforcement Learning / A. Panov, A. Zholus // Artificial General Intelligence. AGI 2021. Lecture Notes in Computer Science. Vol. 13154 / ed. by B. Goertzel, M. Iklé, A. Potapov. — Springer International Publishing, 2022. — P. 344—354.

55. *Panov, A.* Factorized World Models for Learning Causal Relationships / A. Panov, A. Zholus, Y. Ivchenkov // ICLR Workshop on the Elements of Reasoning: Objects, Structure and Causality. — 2022. — URL: https://openreview.net/forum?id=BCGfDBOIcec (visited on May 15, 2024).

56. *Panov, A.* Addressing Task Prioritization in Model-based Reinforcement Learning / A. Panov, A. Zholus, Y. Ivchenkov // Advances in Neural Computation, Machine Learning, and Cognitive Research VI. NEUROINFORMATICS 2022. Vol. 1064 (NeuroInfo 2022) / ed. by B. Kryzhanovsky [et al.]. — Springer, Cham, 2023. — P. 19—30.

57. *Panov, A. I.* Delta Schema Network in Model-based Reinforcement Learning / A. I. Panov, A. Gorodetskiy, A. Shlychkova // Artificial General Intelligence. AGI 2020. Lecture Notes in Computer Science. Vol. 12177 / ed. by B. Goertzel [et al.]. — Springer, 2020. — P. 172—182.

58. *Panov, A. I.* Object-Oriented Decomposition of World Model in Reinforcement Learning / A. I. Panov, L. Ugadiarov // IJCAI Neuro-Symbolic Agents Workshop. — 2023. — URL: https://nsa-wksp.github.io/assets/papers/Object-Oriented%20Decomposition%20of%20World%20Model%20in%20Reinforcement%20Learning.pdf (visited on May 15, 2024).

59. *Panov, A. I.* Behavior Planning of Intelligent Agent with Sign World Model / A. I. Panov // Biologically Inspired Cognitive Architectures. — 2017. — Vol. 19. — P. 21—31.

60. *Panov, A. I.* Goal Setting and Behavior Planning for Cognitive Agents / A. I. Panov // Scientific and Technical Information Processing. — 2019. — Vol. 46, no. 6. — P. 404—415.

61. *Panov, A. I.* Simultaneous Learning and Planning in a Hierarchical Control System for a Cognitive Agent / A. I. Panov // Automation and Remote Control. — 2022. — Vol. 83, no. 6. — P. 869—883.

62. *Panov, A. I.* Application of Pretrained Large Language Models in Embodied Artificial Intelligence / A. I. Panov, A. K. Kovalev // Doklady Mathematics. — 2022. — Vol. 106, S1. — S85—S90.

63. *Panov, A. I.* Behavior and Path Planning for the Coalition of Cognitive Robots in Smart Relocation Tasks / A. I. Panov, K. Yakovlev // Robot Intelligence Technology and Applications 4. Advances in Intelligent Systems and Computing. Vol. 447 / ed. by J.-H. Kim [et al.]. — Springer, 2017. — P. 3—20.

64. *Panov, A. I.* Psychologically Inspired Planning Method for Smart Relocation Task / A. I. Panov, K. S. Yakovlev // Procedia Computer Science. Vol. 88. — Elsevier, 2016. — P. 115—124.

65. *Panov, A. I.* Grid Path Planning with Deep Reinforcement Learning: Preliminary Results / A. I. Panov, K. S. Yakovlev, R. Suvorov // Procedia Computer Science. Vol. 123 / ed. by V. Klimov, A. Samsonovich. — Elsevier, 2018. — P. 347—353.

66. *Panov Aleksandr I nad Sarkisyan, C.* Evaluation of Pretrained Large Language Models in Embodied Planning Tasks / C. Panov Aleksandr I nad Sarkisyan, A. K. Kovalev // Artificial General Intelligence. AGI 2023. Lecture Notes in Computer Science. Vol. 13921 (AGI 2023). — Springer Cham, 2023. — P. 222—232.

67. Pathfinding in stochastic environments: learning vs planning / A. Panov [et al.] // PeerJ Computer Science. — 2022. — Vol. 8. — e1056. — URL: https://peerj.com/articles/cs-1056 (visited on May 15, 2024).

68. Personal Cognitive Assistant: Concept and Key Principals / A. Panov [et al.] // Informatika i ee Primeneniya. — 2019. — Vol. 13, no. 3. — P. 105—113.

69. Personal Cognitive Assistant: Planning Activity with Scripts / A. Panov [et al.] // Informatics and Applications. — 2022. — Vol. 16, no. 1. — P. 46—53.

70. Planning and Learning in Multi-Agent Path Finding / A. Panov [et al.] // Doklady Mathematics. — 2022. — Vol. 106, S1. — S79—S84.

71. Q-Mixing Network for Multi-agent Pathfinding in Partially Observable Grid Environments / A. Panov [et al.] // Artificial Intelligence. RCAI 2021. Lecture Notes in Computer Science. Vol. 12948 / ed. by S. M. Kovalev, S. O. Kuznetsov, A. I. Panov. — Springer, 2021. — P. 169—179. — arXiv: 2108.06148.

72. Quantized Disentangled Representations for Object-Centric Visual Tasks / A. I. Panov [et al.] // Pattern Recognition and Machine Intelligence. PReMI 2023. Lecture Notes in Computer Science. Vol. 14301 (PReMI 2023). — Springer Cham, 2023. — P. 514—522.

73. Question Answering for Visual Navigation in Human-Centered Environments / A. I. Panov [et al.] // Advances in Soft Computing. MICAI 2021. Part II. Lecture Notes in Computer Science. Vol. 13068 / ed. by I. Batyrshin, A. Gelbukh, G. Sidorov. — Springer, 2021. — P. 31—45.

74. Real-Time Object Navigation with Deep Neural Networks and Hierarchical Reinforcement Learning / A. Panov [et al.] // IEEE Access. — 2020. — Vol. 8. — P. 195608—195621.

75. Skill Fusion in Hybrid Robotic Framework for Visual Object Goal Navigation / A. Panov [et al.] // Robotics. — 2023. — Vol. 12. — URL: https://www.mdpi.com/2218-6581/12/4/104 (visited on May 15, 2024).

76. STRL-Robotics: интеллектуальное управление поведением робототехнической платформы в человеко-ориентированной среде / А. Панов [и др.] // Искусственный интеллект и принятие решений. — 2023. — № 2. — С. 45—63.

77. When to Switch: Planning and Learning For Partially Observable Multi-Agent Pathfinding / A. Panov [et al.] // IEEE Transactions on Neural Networks and Learning Systems. — 2023. — URL: https://ieeexplore.ieee.org/document/10236574 (visited on May 15, 2024).

78. Знаковая картина мира субъекта поведения / А. И. Панов [и др.]. — М. : Физматлит, 2018. — 264 с.

79. *Панов, А. И.* Моделирование процесса принятия решения агентом со знаковой картиной мира / А. И. Панов // Теория и практика системного анализа: Труды II Всероссийской научной конференции молодых учёных с международным участием. Т. I. — Рыбинск : РГАТУ имени П.А. Соловьева, 2012. — С. 126—137.

80. *Панов, А. И.* Семейства отношений в знаковой картине мира / А. И. Панов // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ–2012 (16–20 октября 2012г., г. Белгород, Россия): Труды конференции. Т. 1. — Белгород : Издательство БГТУ, 2012. — С. 301—309.

81. *Панов, А. И.* Алгебраические свойства операторов распознавания в моделях зрительного восприятия / А. И. Панов // Машинное обучение и анализ данных. — 2014. — Т. 1, № 7. — С. 863—874.

82. *Панов, А. И.* Представление знаний автономных агентов, планирующих согласованные перемещения / А. И. Панов // Робототехника и техническая кибернетика. — 2015. — № 4. — С. 34—40.

83. *Панов, А. И.* Представление знаний в задачах согласованного перемещения группы БПЛА / А. И. Панов // Второй Всероссийский научно-практический семинар “Беспилотные транспортные средства с элементами искусственного интеллекта (БТС-ИИ-2015)”, (9 октября 2015г., г. Санкт-Петербург, Россия): Труды семинара. — Санкт-Петербург : Изд-во “Политехника-сервис”, 2015. — С. 74—82.

84. *Панов, А. И.* Формирование образной компоненты знаний когнитивного агента со знаковой картиной мира / А. И. Панов // Информационные технологии и вычислительные системы. — 2018. — № 4. — С. 84—96.

85. *Панов, А. И.* STRIPS постановка задачи планирования поведения в знаковой картине мира / А. И. Панов, Г. А. Киселев // Информатика, управление и системный анализ: Труды IV Всероссийской научной конференции молодых учёных с международным участием. Т. I. — Тверь : Тверской государственный технический университет, 2016. — С. 131—138.

86. *Панов, А. И.* Большие языковые модели как аппроксиматоры значения в знаковой картине мира / А. И. Панов, А. К. Ковалев, А. А. Чуганская // Всероссийская конференция "Поспеловские чтения: искусственный интеллект - проблемы и перспективы Поспеловские чтения-2022 (Москва, 19-20 декабря 2022 г.). Труды конференции. — Издательство ФИЦ ИУ РАН, 2022. — С. 53—70.

87. *Панов, А. И.* Методы внутренней мотивации в задачах обучения с подкреплением на основе модели / А. И. Панов, А. Латышев // Искусственный интеллект и принятие решений. — 2023. — № 3. — С. 84—97.

88. *Панов, А. И.* STRL: многоуровневая система управления интеллектуальными агентами / А. И. Панов, Д. А. Макаров, К. С. Яковлев // Пятнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2016 (3-7 октября 2016г., г.Смоленск, Россия): Труды конференции. Т. 1. — Смоленск : Универсум, 2016. — С. 179—188.

89. *Панов, А. И.* Архитектура многоуровневой интеллектуальной системы управления беспилотными летательными аппаратами / А. И. Панов, К. С. Макаров Д. А. andЯковлев // Искусственный интеллект и принятие решений. — 2015. — № 3. — С. 18—33.

90. *Панов, А. И.* Моделирование поведения автономного мобильного робота / А. И. Панов, А. В. Петров // Вестник РГАТУ имени П.А. Соловьева. — 2012. — № 2. — С. 179—185.

91. *Панов, А. И.* Когнитивные архитектуры и проекты систем управления автономных мобильных роботов / А. И. Панов, А. В. Петров, Р. Г. Березовский // Вестник РГАТУ имени П.А. Соловьева. — 2013. — № 1. — С. 111—113.

92. *Панов, А. И.* Автоматическое построение иерархии абстрактных автоматов для задачи обучения с подкреплением / А. И. Панов, А. А. Скрынник // Информатика, управление и системный анализ: Труды V Всероссийской научной конференции молодых учёных с международным участием. — Ростов-на-Дону : Мини-Тайп, 2018. — С. 7—16.

93. *Панов, А. И.* Автоматическое формирование правил перемещения с использованием обучения с подкреплением / А. И. Панов, Р. Е. Суворов // Седьмая Международная конференция "Системный анализ и информационные технологии"САИТ-2017 (13-18 июня 2017 г., г. Светлогорск, Россия): Труды конференции. — М. : ФИЦ ИУ РАН, 2017. — С. 303—310.

94. *Панов, А. И.* Модель среды для актора и критика в обучении с подкреплением / А. И. Панов, Л. Угадяров // Двадцатая Национальная конференция по искусственному интеллекту с международным участием, КИИ-2022 (Москва, 21-23 декабря 2022 г.). Труды конференции. В 2 т. Т. 2. — М. : Издательство МЭИ, 2022. — С. 39—54.

95. *Панов, А. И.* Взаимодействие стратегического и тактического планирования поведения коалиций агентов в динамической среде / А. И. Панов, К. С. Яковлев // Искусственный интеллект и принятие решений. — 2016. — № 4. — С. 68—78.

96. *Панов, А.* Иерархическая постановка задачи объектно-центричного обучения с подкреплением / А. Панов // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов XI Международной научно-практической конференции (ИММВ-2022, Коломна, 16-19 мая 2022 г.). В 2-х томах. Т. 2. — 2022. — С. 248—256.

97. *Панов, А.* Формирование умений агента по принципу достижимости в обучении с подкреплением / А. Панов, А. Латышев // Двадцать первая Национальная конференция по искусственному интеллекту с международным участием, КИИ-2023 (Смоленск, 16-20 октября 2023 г.). Труды конференции. В 2 томах. Т. 1 (КИИ-2023). — 2023. — С. 264—274.

98. Принципы построения многоуровневых архитектур систем управления беспилотными летательными аппаратами / А. И. Панов [и др.] // Авиакосмическое приборостроение. — 2013. — № 4. — С. 10—28.

Панов Александр Игоревич

Методы и алгоритмы нейросимвольного обучения и планирования поведения когнитивных агентов

Автореф. дис. на соискание ученой степени докт. физ.-мат. наук

Подписано в печать ___.___._____. Заказ № _____
Формат _____. Усл. печ. л. 2,5. Тираж ___ экз.
Типография _____