Совместное обучение высокоуровневого планировщика и низкоуровневых стратегий

A Preprint

Григорий Бухтуев
Центр когнетивного моделирования
Московский Физико–Технический Институт
Долгопрудный
bukhtuev.ga@phystech.edu

December 16, 2024

Abstract

Данная работа является обзором методов совместного обучения высокоуровневого планировщика и низкоуровневых стратегий для решения задач навигации и манипуляции с объектами. Рассматриваются работы, в которых используются языковые модели для генерации планов и стратегий, а также подходы к обучению с подкреплением, направленные на улучшение взаимодействия между высокоуровневыми планировщиками и низкоуровневыми действиями. Обсуждаются примеры различных виртуальных сред и бенчмарков для оценивания качества генерации планов и их выполнения агентом.

Keywords Совместное обучение, Высокоуровневый планировщик, Низкоуровневые стратегии, Обучение с подкреплением, Языковые модели

1 Введение

Современные достижения в области робототехники и искусственного интеллекта создают новые возможности для решения сложных задач навигации и манипуляции. Большие языковые модели (LLM) продемонстрировали свою эффективность в построении высокоуровневых планов, которые могут быть использованы роботами с длительным горизонтом планирования. Эти модели способны интегрировать знания, полученные из глобальных источников информации, для декомпозиции долгосрочных задач на более мелкие, достижимые подзадачи. Однако, следует отметить, что LLM не располагают механизмами для разработки низкоуровневых стратегий управления роботами.

С другой стороны, модели компьютерного зрения зарекомендовали себя в задачах детекции целевых объектов и точного определения позы агента-манипулятора, что создает удобные условия для решения низкоуровневых задач взаимодействия с конкретными объектами. В то же время методы обучения с подкреплением (RL) значительно развились и теперь способны реализовывать низкоуровневые стратегии взаимодействия. Тем не менее, RL сталкивается с трудностями в осуществлении долгосрочного планирования.

В идеале агент RL должен иметь возможность следовать указаниям LLM, что позволит ему эффективно обучаться решению каждой предсказанной подзадачи в режиме онлайн. Объединение этих подходов предоставляет возможность интеграции сильных сторон различных моделей и разработки эффективных стратегий поведения роботов, которые требуют синергии высокоуровневых планов и низкоуровневых действий. В данной статье рассматривается концепция совместного обучения высокоуровневого планировщика и низкоуровневых стратегий, что способствует повышению эффективности выполнения задач.

2 Постановка задачи

В последние годы были проведены исследования, позволяющие эффективно решать задачи робототехники с длительным горизонтом планирования. Применяются методы создания высокоуровневых планов с помощью языковых моделей. Далее применяется планирование движения и обучение низкоуровневым стратегиям управления.

2.1 Высокоуровневый планировщик

В качестве высокоуровнего планировщика могут выступать:

- Большая языковая модель
- Методы обучения с подкреплением
- Визуальный модуль(CV), который отвечает за планирование движения и определение позы робота для выполнения каждой задачи

2.2 Низкоуровневая стратегия

В качестве низкоуровневой стратегии могут использоваться:

- Заранее определенная библиотека навыков
- Методы обучения с подкреплением

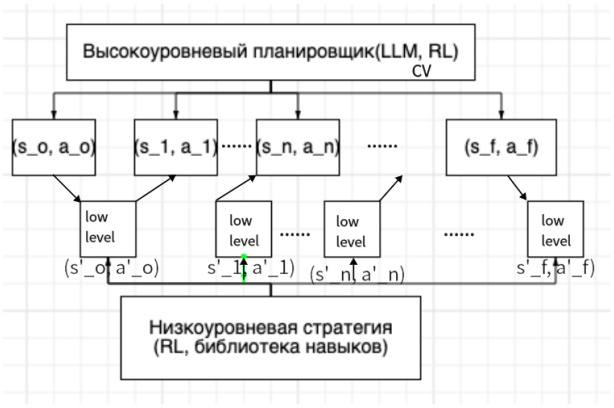


Figure 1: Схема связи высокоуровневого планировщика и низкоуровневых стратегий

2.3 Методы обучения с подкреплением

Говоря об алгоритмах RL, мы рассматриваем частично наблюдаемые марковские процессы принятия решений (POMDP) в форме $(S, A, T, R, \pi_0, O, \pi_O, \gamma)$.

Здесь S - это набор состояний среды, A - это набор действий, $T(s'|s, a)$ — распределение вероятностей перехода, $R(s, a, s')$ — это функция вознаграждения, π_0 — распределение по начальному состоянию $s_0 \sim \pi_0$, O — это набор наблюдений, π_O — распределение по наблюдениям, обусловленное состоянием $O \sim \pi_O(O|s)$, и γ — это дисконтный фактор. В нашем случае пространство наблюдения — это набор всех изображений RGB-D (RGB и глубина). Функция вознаграждения определяется средой. Цель агента — максимизировать ожидаемую сумму вознаграждений по траектории, $\mathbb{E}\left[\sum_t \gamma^t \mathcal{R}(s_t, a_t, s_{t+1})\right]$. В нашей работе мы рассматриваем POMDP, которые описывают воплощенного робота-агента, взаимодействующего со сценой. Мы предполагаем, что текстовое описание задачи, g, предоставляется агенту на естественном языке.

2.4 Генерация синтетических задач

Параллельно с этим была разработана методика, позволяющая в реальном времени генерировать новые задачи на основе существующих задач для воплощённых роботов с помощью тех же LLM. Эта методика

демонстрирует значительный прирост качества выполнения задач с коротким горизонтом планирования и хорошую обобщаемость на задачи, состоящие из нескольких последовательных действий.

2.5 Цель работы

Целью данной работы является объединение двух подходов: Совместное обучение высокоуровневого планировщика и низкоуровневых стратегий и обучение низкоуровневых стратегий, не только на заранее определенных задачах из заданной среды, но и на синтетических задачах, которые будут генерироваться в процессе обучения с помощью LLM.

3 Обзор актуальных исследований

3.1 Plan-Seq-Learn [1]

Статья посвящена проблеме решения долгосрочных задач в робототехнике с использованием языковых моделей (LLM) и методов обучения с подкреплением (RL). Авторы представляют новый подход под названием Plan-Seq-Learn (PSL), который объединяет высокоуровневое планирование с LLM и низкоуровневое управление с помощью RL. PSL позволяет роботам адаптироваться к сложным задачам, не полагаясь на заранее определенные навыки, что является значительным шагом вперед в области робототехники.

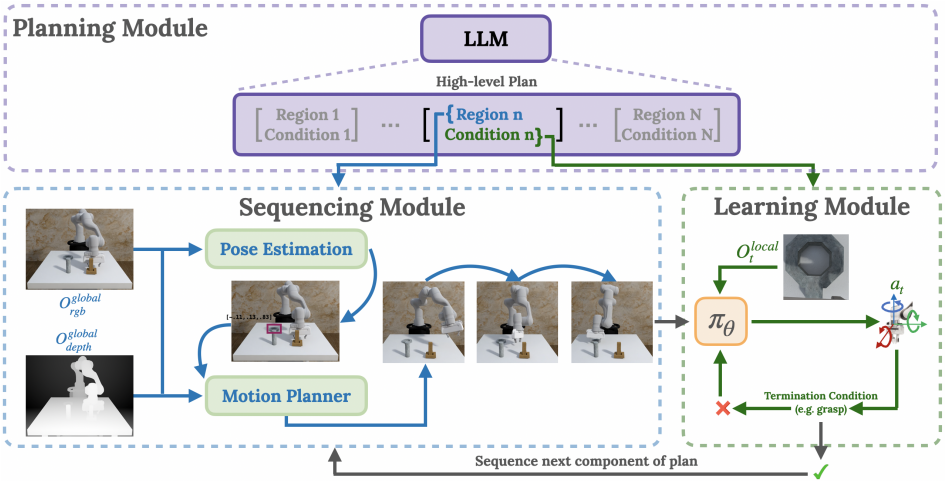


Figure 2: Обзор метода. PSL декомпозирует задачи на список регионов и условия завершения этапов, используя LLM (сверху), последовательно планирует действия с помощью планирования движений (слева) и обучает управляющие политики с использованием RL (справа).

3.1.1 Основные идеи

- Проблема: Существующие методы планирования, основанные на LLM, требуют наличия заранее определенной библиотеки навыков. Это ограничивает их применение, особенно в задачах, требующих тонкой настройки поведения робота.
- Решение: PSL использует LLM для создания высокоуровневых планов, которые затем преобразуются в низкоуровневые действия с помощью RL и планирования движений. Этот подход позволяет эффективно решать долгосрочные задачи, используя визуальные данные в качестве входных данных.

3.1.2 Структура PSL

Метод состоит из трех основных компонентов:

- Планирование (Plan): Использует языковую модель для генерации высокоуровневого плана на основе текстового описания задачи. Языковая модель разбивает задачу на подзадачи и формирует последовательность действий, которые необходимо выполнить.

- Навигация (Seq): Переводит высокоуровневый план в целевые состояния для робота, используя данные от визуальных сенсоров. Этот модуль отвечает за планирование движения и определение позы робота для выполнения каждой подзадачи.
- Обучает низкоуровневые стратегии с использованием методов обучения с подкреплением (RL), адаптируя поведение робота на основе достигнутых результатов и полученной обратной связи от окружающей среды.

3.1.3 Результаты

- PSL продемонстрировал превосходные результаты на более чем 25 сложных задачах с успехом более 85% в четырех различных бенчмарках.
- В частности, PSL достиг 96% успеха в задаче NutAssembly, что подчеркивает его эффективность в решении контактно-ориентированных задач.
- Метод показал высокую устойчивость к ошибкам в оценке поз, что делает его подходящим для применения в реальных условиях.

3.1.4 Сравнение с другими методами

- PSL превосходит существующие методы, такие как RAPS, TAMP [2] и SayCan, которые зависят от заранее определенных навыков или имеют ограничения в плане адаптации к новым задачам.
- PSL эффективно справляется с задачами, требующими взаимодействия с объектами в условиях контакта, что является сложной задачей для традиционных методов.

Этот подход позволяет более эффективно решать задачи с длинными горизонтами, где требуется последовательное выполнение множества действий.

3.2 MoPA-RL [3]

Статья представляет новый подход, называемый Motion Planner Augmented Reinforcement Learning, который объединяет методы глубокого обучения с подкреплением и планирования движений для решения задач манипуляции в сложных, загроможденных средах. Авторы подчеркивают, что традиционные RL-агенты требуют большого объема опыта для обучения, особенно в условиях с множеством препятствий, что затрудняет исследование. С другой стороны, методы планирования движений используют явные модели среды для нахождения безопасных путей, но сталкиваются с трудностями при взаимодействии с объектами.

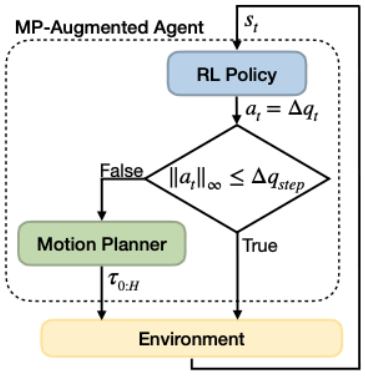


Figure 3: Обзор метода. Структура расширяет политику RL с помощью планировщика движений. Если предсказанное действие политикой превышает порог qstep, вызывается планировщик движений; в противном случае действие выполняется напрямую.

3.2.1 Основные идеи

- Проблема: RL-агенты могут обучаться манипуляционным задачам, но требуют много опыта, особенно в загроможденных средах.

- Решение: MoPA-RL комбинирует возможности RL и планирования движений, расширяя пространство действий RL-агента, что позволяет эффективно использовать планирование движений для достижения целей.

3.2.2 Структура MoPA-RL

- Планирование: Агент использует планирование движений для нахождения безопасных путей к целям.
- Обучение: RL-агент обучается выполнять действия, используя как MP, так и прямое выполнение действий в зависимости от величины действия.

3.2.3 Результаты

- MoPA-RL продемонстрировал улучшение эффективности обучения по сравнению с традиционными методами, такими как SAC, RAPS и другими, особенно в задачах, требующих манипуляции в сложных условиях.
- Метод также показал способность к более быстрой и безопасной исследовательской деятельности в загроможденных средах.

Статья подчеркивает важность интеграции методов планирования и обучения с подкреплением для решения сложных задач манипуляции в загроможденных средах. MoPA-RL представляет собой эффективный подход, который значительно улучшает обучаемость роботов в реальных условиях. По сравнению с PSL, MoPA-RL требует, чтобы агент сам определял, когда и где использовать планировщик движений, а PSL более эффективно управляет этим процессом, обеспечивая четкое разделение задач.

3.3 RAPS [4]

Статья представляет новый подход, называемый Robot Action Primitives for Reinforcement Learning, который объединяет методы обучения с подкреплением и параметризированные действия для решения задач манипуляции роботами. Авторы отмечают, что традиционные RL-агенты сталкиваются с трудностями в исследовании непрерывных пространств действий, что замедляет процесс обучения. В отличие от этого, RAPS предлагает заранее определенные примитивы действий, которые могут быть параметризованы, что позволяет агентам более эффективно исследовать и обучаться.

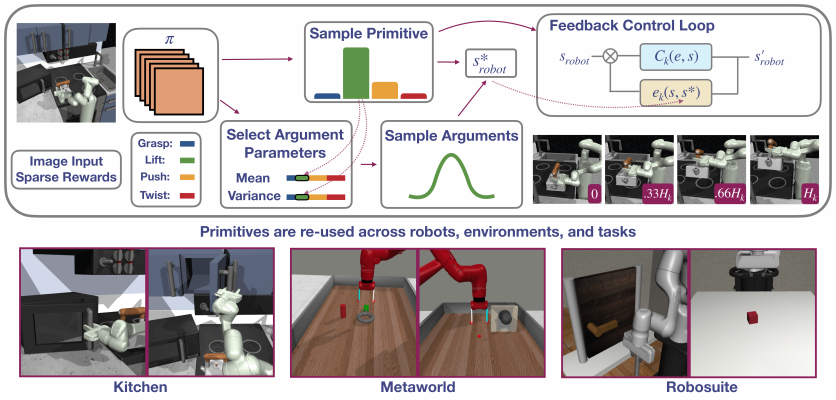


Figure 4: Обзор метода. Структура расширяет политику RL с помощью планировщика движений. Если предсказанное действие политикой превышает порог qstep, вызывается планировщик движений; в противном случае действие выполняется напрямую.

3.3.1 Основные идеи

- Проблема: RL-агенты требуют большого объема опыта для обучения, особенно в сложных манипуляционных задачах, где исследование может быть затруднено. Традиционные методы параметризации действий часто зависят от конкретных задач и не могут быть легко адаптированы к новым условиям.

- Решение: RAPS комбинирует заранее определенные примитивы действий с параметризацией, позволяя RL-агенту выбирать и настраивать действия в зависимости от текущих условий среды. Это улучшает исследовательскую способность и повышает эффективность обучения.

3.3.2 Структура RAPS

- Примитивы действий: Каждый примитив представляет собой заранее определенное действие (например: подъем, толкание), параметризованное аргументами, которые агент обучается выбирать.
- Обучение: RL-агент использует выбранные примитивы для выполнения действий, что позволяет ему сосредоточиться на том, что нужно сделать, а не на том, как это выполнить.

3.3.3 Результаты

- RAPS продемонстрировал значительное улучшение эффективности обучения по сравнению с традиционными методами, требующих манипуляции в сложных условиях.
- Эксперименты показывают, что RAPS успешно решает задачи, такие как "Kitchen Kettle" и "Metaworld Drawer Close", с высокой эффективностью, обеспечивая надежное выполнение манипуляций.

Статья подчеркивает важность интеграции параметризированных действий в обучение с подкреплением для решения сложных задач манипуляции роботами.

3.4 Game On [5]

Статья представляет новую архитектуру агента, которая автоматизирует части рабочего процесса экспериментов в области обучения с подкреплением (RL), используя большие языковые модели (VLM). Авторы подчеркивают, что традиционные методы требуют значительного участия человека в процессе экспериментов, особенно в задачах, связанных с контролем и манипуляцией. Основная цель работы — использование VLM для автоматизации предложений задач, анализа прогресса и построения учебных планов.

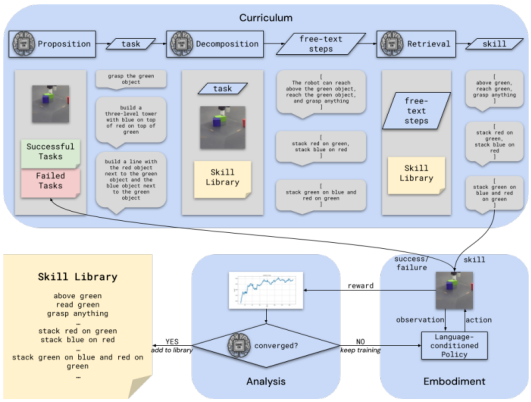


Figure 5: Иллюстрация архитектуры системы и взаимодействия ее компонентов. Модуль учебной программы генерирует предложения свободного текста, разлагает их на шаги свободного текста и пытается сопоставить их с навыками фиксированного текста из библиотеки. Генерированная последовательность навыков выполняется политикой, обусловленной текстом, модуля воплощения и разворачивается в эпизод, который используется для улучшения политики.

3.4.1 Основные идеи

- Проблема: Традиционные RL-агенты требуют большого объема доменно-специфических знаний и человеческого вмешательства на различных этапах эксперимента, таких как предложение новых задач и оценка прогресса.

- Решение: вторы предлагают архитектуру, которая включает в себя три основных модуля: модуль учебного плана (для генерации и декомпозиции задач), модуль воплощения (для выполнения задач) и аналитический модуль (для оценки прогресса обучения).

3.4.2 Структура системы

- Модуль учебного плана: Генерирует высокоуровневые задачи и разбивает их на подзадачи, используя VLM для анализа предыдущих успехов и неудач.
- Модуль воплощения: Выполняет предложенные подзадачи, используя библиотеку навыков и собирая данные о выполнении.
- Аналитический модуль: Оценивает, когда обучение задачи считается завершенным, и обновляет библиотеку навыков.

3.4.3 Результаты

- Прототип системы продемонстрировал способность автоматически генерировать разнообразные данные, которые затем использовались для самоулучшения контроля политики агента.
- Эксперименты в симуляторе MuJoCo показали, что VLM может эффективно направлять агента в сложных задачах манипуляции, улучшая производительность и снижая количество столкновений.
- Система показала способность адаптироваться и осваивать новые задачи, что свидетельствует о ее потенциальной полезности в реальных приложениях.

Статья подчеркивает важность интеграции языковых моделей в процессы обучения с подкреплением для автоматизации экспериментов. Предложенная архитектура представляет собой значительный шаг вперед в области автоматизации RL-экспериментов и может значительно повысить эффективность и безопасность роботов в сложных задачах манипуляции.

3.5 IGOR [6]

Статья посвящена разработке подхода, который позволяет искусственному интеллекту выполнять сложные языковые инструкции в виртуальных средах. Авторы предлагают иерархическую структуру, которая объединяет глубокое понимание языка больших языковых моделей (LLM) с адаптивными возможностями выполнения действий агентов, обученных с помощью методов обучения с подкреплением (RL). Этот подход позволяет агентам эффективно управлять сложными задачами, состоящими из нескольких взаимозависимых действий.

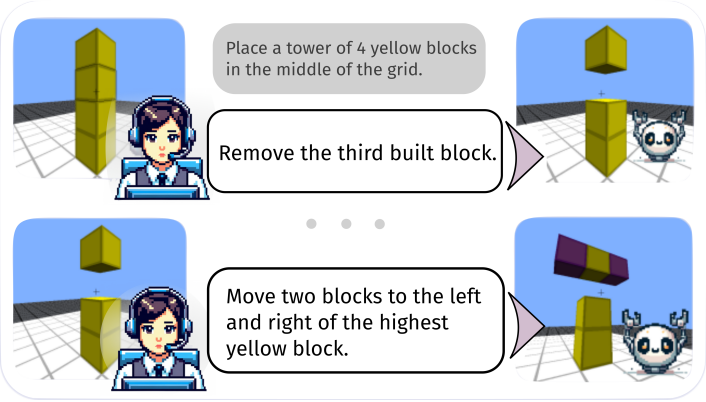


Figure 6: Задача совместного взаимодействия между агентом, средой и пользователем включает в себя следующее: пользователь предоставляет инструкции агенту, а агент выполняет действия в среде для достижения задачи на основе этих инструкций.

3.5.1 Основные идеи

- Проблема: Несмотря на успехи LLM в выполнении многошаговых задач, их использование в автономных агентах все еще ограничено. Традиционные RL-агенты, хотя и эффективны в

обучении последовательностям действий, часто сталкиваются с трудностями при выполнении сложных языковых инструкций.

- Авторы представляют иерархическую структуру IGOR (Instruction Following with Goal-Conditioned RL), которая включает два основных компонента:
  - Языковой модуль: преобразует естественные языковые инструкции в высокоуровневый план действий с подцелями.
  - Модуль стратегии: отвечает за выполнение этого плана и достижение подцелей.
  - Обучение: Каждый модуль обучается независимо, что позволяет применять эффективные стратегии обучения для LLM и RL. Для LLM используются методы увеличения данных, а для RL — целевое обучение на основе целей.

3.5.2 Результаты

- Метод IGOR был протестирован в двух виртуальных средах: IGLU, где агенты строят конструкции по языковым инструкциям, и Crafter, где агенты выполняют задачи в динамически изменяющейся среде. Результаты показывают, что IGOR превосходит существующие алгоритмы, достигнув более высокой эффективности выполнения задач.
- В отличие от традиционных методов, IGOR эффективно справляется с многозадачностью и сложными языковыми структурами, что делает его более подходящим для применения в реальных сценариях.

Статья подчеркивает важность интеграции языковых моделей и методов обучения с подкреплением для решения задач, связанных с выполнением естественноязыковых инструкций в виртуальных средах.

3.6 SayCan [7]

Статья представляет метод SayCan, который объединяет возможности больших языковых моделей (LLM) и предобученных навыков роботов для выполнения сложных задач на основе естественного языка. Авторы подчеркивают, что языковые модели обладают богатым семантическим знанием, но не имеют реального опыта взаимодействия с физическим миром, что может привести к неэффективным или даже опасным рекомендациям для роботов.



Figure 7: LLM не взаимодействовали со своей средой и не наблюдали за результатами своих ответов, и, следовательно, не имеют привязки к реальному миру. SayCan связывает LLM с помощью функций ценности предобученных навыков, что позволяет им выполнять реальные, абстрактные команды на длительный срок на роботах.

3.6.1 Основные идеи

- Проблема: Языковые модели, хотя и могут генерировать разумные ответы на команды, часто не способны адекватно интерпретировать инструкции, учитывая физические ограничения и возможности робота.

- Решение: Метод SayCan сочетает в себе высокоуровневые знания от языковых моделей (Say) и функции ценности, которые определяют, какие действия возможны в текущем состоянии мира (Can).

3.6.2 Структура SayCan

- Языковая модель предоставляет высокоуровневые инструкции и оценивает, какие навыки могут быть полезны для выполнения задачи.
- Функции ценности, обученные с помощью методов обучения с подкреплением, определяют, насколько каждый из навыков может быть успешно выполнен в текущей среде.

3.6.3 Результаты

- SayCan продемонстрировал высокие уровни успешности в планировании и выполнении 101 задачи в реальной кухне, достигая 84% успешности в планировании и 74% в выполнении.
- Метод также обеспечивает объяснимость процесса принятия решений, позволяя пользователям понять, какие действия рассматриваются и почему они выбираются.

Статья подчеркивает важность интеграции языковых моделей и робототехники для выполнения сложных задач, демонстрируя, как предварительно обученные навыки могут улучшить понимание и выполнение команд на естественном языке. PSL превосходит этот подход, поскольку не требует заранее определенной библиотеки навыков.

3.7 GenSim [8]

Статья представляет новый подход, который использует большие языковые модели для автоматической генерации задач в области робототехники. Основная проблема, которую решает GenSim, заключается в том, что сбор больших объемов данных взаимодействия в реальном мире для обучения универсальных робототехнических политик является дорогостоящим и трудоемким процессом. Существующие методы генерации данных сосредоточены на разнообразии на уровне сцен, но не на уровне задач, что затрудняет обобщение политик, обученных на симуляционных данных.

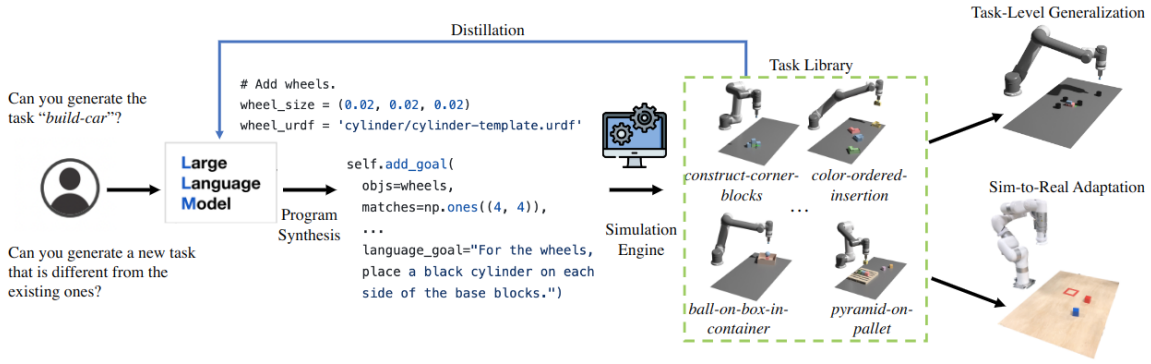


Figure 8: GenSim — это фреймворк LLM для расширения разнообразия симуляционных задач для обучения политик роботов. Мы исследуем режим, ориентированный на достижение целей (верхний запрос), и исследовательский режим (нижний запрос), которые генерируют коды симуляционных задач для роботов. Сгенерированные коды задач кэшируются в библиотеке задач, которая может быть использована для обучения политик с целью достижения лучшей обобщаемости на уровне задач и адаптации от симуляции к реальности.

GenSim предлагает два режима генерации:

- Целевая генерация — когда LLM получает конкретную задачу и предлагает учебный план для ее решения.
- Исследовательская генерация — когда LLM использует предыдущие задачи для итеративного создания новых задач.

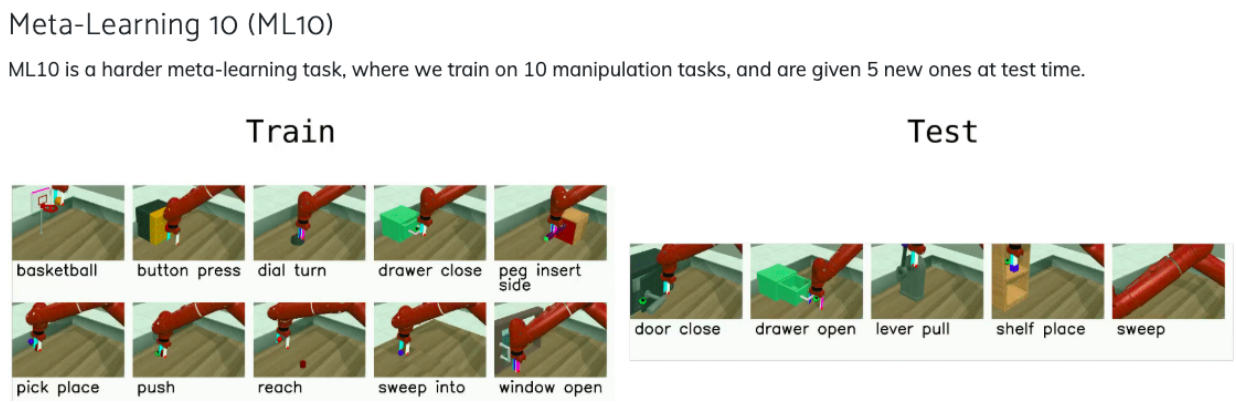
Используя GPT-4, авторы расширили существующий набор задач в десять раз, создав более 100 новых задач для обучения и тестирования.

GenSim продемонстрировал, что задачи, сгенерированные LLM, значительно улучшают обобщение на уровне задач при обучении многозадачным политикам.

Политики, предварительно обученные на задачах, сгенерированных GPT-4, показали улучшение на 50% в обобщении по сравнению с моделями, обученными только на задачах, созданных человеком.

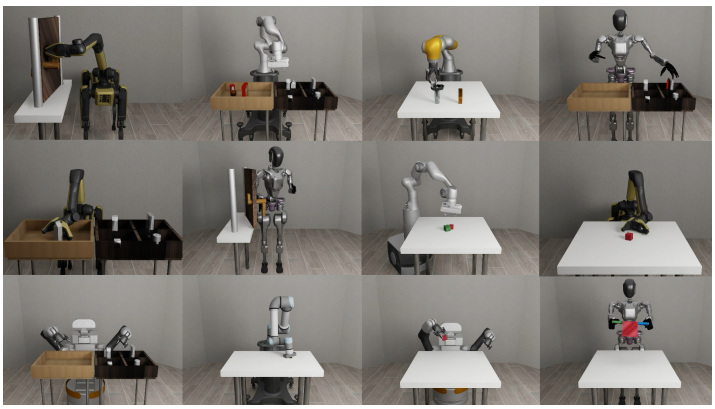
4 Бенчмарки и экспериментальные результаты

4.1 Meta-World [9]



Meta-World представляет собой набор задач для обучения и оценки алгоритмов в области робототехники и обучения с подкреплением. Тест состоит из 50 отдельных задач манипуляции с роботом. так же есть хорошо сформированные функции вознаграждения. Тесты включают манипулирование одним объектом в случайном положении цели или несколькими объектами в детерминированном положении цели. Он включает в себя разнообразные задачи, такие как MW-Disassemble (разборка), MW-BinPick (подбор и размещение объекта) и MW-Assembly (сборка). Каждая из этих задач требует от агента выполнения сложных манипуляций, что делает их подходящими для тестирования методов, направленных на улучшение навыков планирования и управления. PSL показал высокую эффективность в решении этих задач, достигая успеха более 90%.

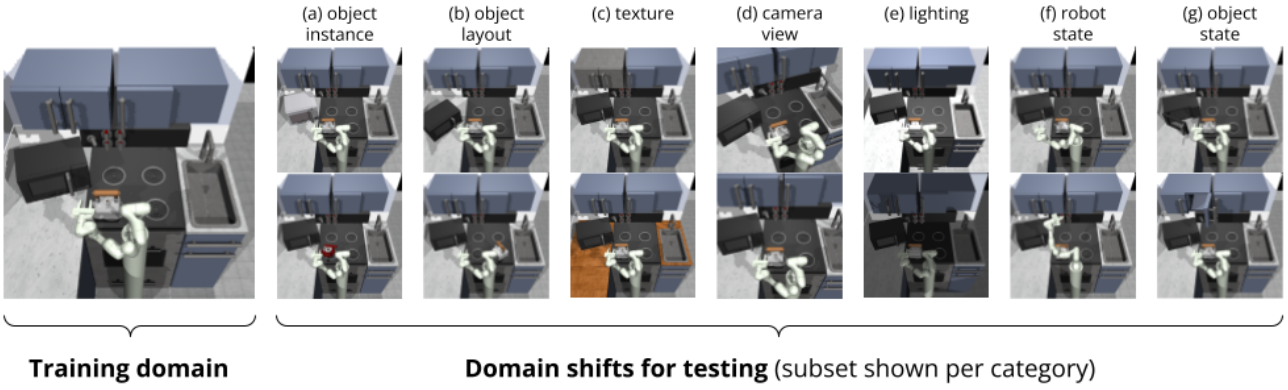
4.2 Robosuite [10]



Тест Robosuite содержит сложные, долгосрочные задачи манипуляции, включающие выбор места и сборку гаек, а также более простые задачи, включающие подъем куба и открытие двери. Награды грубо определены с точки зрения расстояний до целей, а также условий захвата/размещения, которые, по сути, легко реализовать в реальном мире, а также с помощью оценки позы. Это контрастирует с Meta-World, который тратит значительные инженерные усилия на определение хорошо сформированных плотных наград, часто используя преимущества геометрии объекта.

В рамках задач Robosuite, архитектура Plan-Seq-Learn продемонстрировала выдающиеся результаты, достигая успешности в 96% при выполнении задач с длительным горизонтом. Это подчеркивает превосходство данного подхода по сравнению с другими методами, так как PSL эффективно использует высокоуровневое планирование, основанное на языковых моделях, в сочетании с планированием движения и обучением низкоуровневым стратегиям управления.

4.3 Kitchen [11]



Тест Kitchen представляет собой набор сложных, разреженных вознаграждений, контролируемых манипуляционных задач на одной кухне. Задания требуют способности эффективно исследовать, а также способности связывать навыки в длинных временных горизонтах, чтобы достичь поведения, такого как открытие микроволновки, перемещение чайника, щелчок выключателя, поворот верхней левой горелки и, наконец, сдвиг дверцы шкафа. В рамках задач Kitchen архитектура Plan-Seq-Learn продемонстрировала улучшенные результаты по сравнению с традиционными методами.

5 Заключение

Совместное обучение высокоуровневого планировщика и низкоуровневых стратегий является перспективным направлением в области робототехники. Интеграция языковых моделей и обучение с подкреплением позволяют создавать более эффективные системы, способные решать сложные задачи навигации и манипуляции. В дальнейшем необходимо продолжать исследовать возможности улучшения взаимодействия между различными уровнями управления и оптимизации процессов обучения.

References

[1] Murtaza Dalal, Tarun Chiruvolu, Devendra Chaplot, and Ruslan Salakhutdinov. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. 2024.

[2] Rachel Holladay Caelan Reed Garrett, Rohan Chitnis. Integrated task and motion planning. 2020.

[3] Jun Yamada, Youngwoon Lee, Gautam Salhotra, Karl Pertsch, Max Pflueger, Gaurav S. Sukhatme, Joseph J. Lim, and Peter Englert. Motion planner augmented reinforcement learning for robot manipulation in obstructed environments. In Conference on Robot Learning, 2020.

[4] Murtaza Dalal, Deepak Pathak, and Ruslan Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. In NeurIPS, 2021.

[5] Abbas Abdolmaleki Jost Tobias Springenberg Martin Riedmiller Jingwei Zhang, Thomas Lampe. Game on: Towards language models as rl experimenters. arXiv:2409.03402, 2024.

[6] Petr Kuderov Aleksandr I. Panov Zoya Volovikova, Alexey Skrynnik. Instruction following with goal-conditioned reinforcement learning in virtual environments. arXiv:2409.03402, 2024.

[7] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse,

Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can and not as i say: Grounding language in robotic affordances. In arXiv preprint arXiv:2204.01691, 2022.

[8] Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. Gensim: Generating robotic simulation tasks via large language models. In Arxiv, 2023.

[9] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In Conference on Robot Learning (CoRL), 2019.

[10] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, Yifeng Zhu, and Kevin Lin. robosuite: A modular simulation framework and benchmark for robot learning. In arXiv preprint arXiv:2009.12293, 2020.

[11] Corey Lynch Sergey Levine Abhishek Gupta, Vikash Kumar and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. arXiv preprint arXiv:1910.11956, 2019.