Планирование поведения с помощью большой языковой модели с визуальной обратной связью от среды для задачи навигации и перемещения объектов

Патрацкий М.А.
patratskii.ma@phystech.edu

Центр когнитивного моделирования МФТИ

2024 – Научно-исследовательский семинар

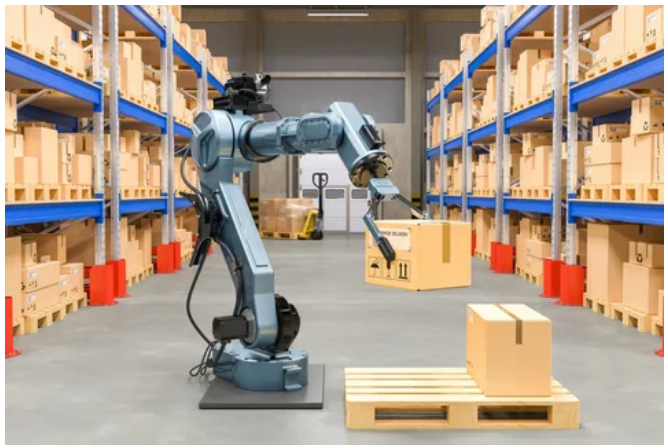


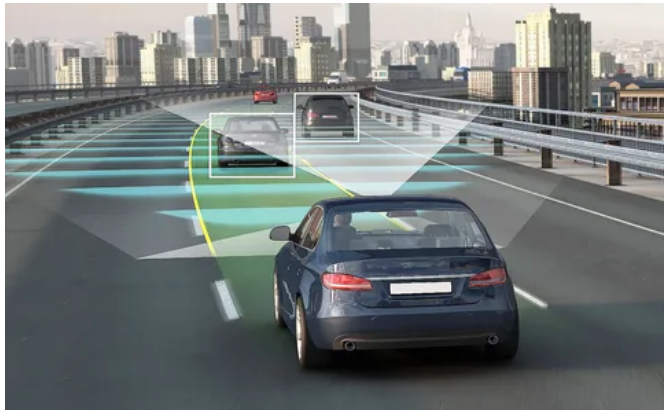


Мотивация

Задача планирования поведения возникает в:

- Роботизированных складских системам
- Автономном вождении
- Планировании поведения антропоморфных роботов





Мотивация

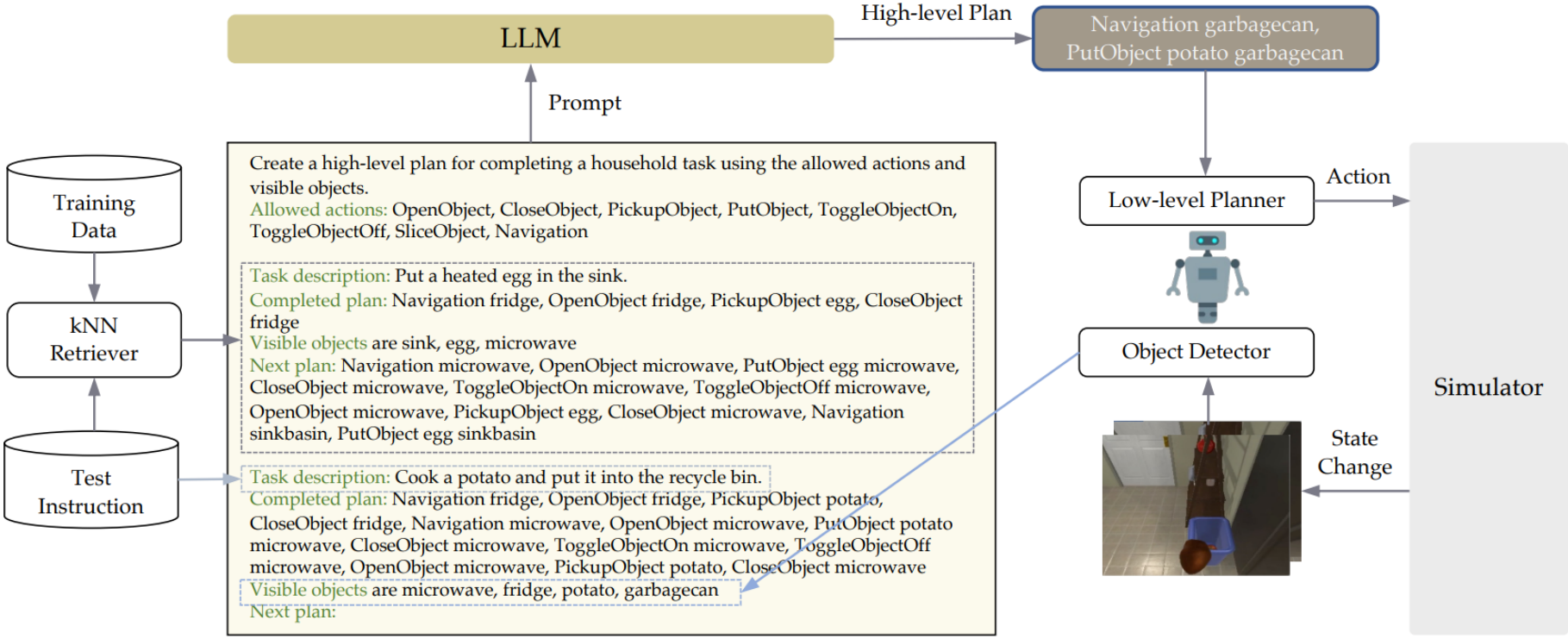
Существует множество методов, решающих задачу планирования с помощью больших языковых моделей:

- LLM Planer [1]
- SayCan [2]
- Inner Monologue [3]

Проблема данных подходов заключается в отсутствии передачи визуальной информации о среде в языковую модель. Многие работы используют список обнаруженных объектов на сегментации, однако языковая модель не может получить информацию о чем-либо другом.

Мотивация

Языковая модель имеет доступ к списку обнаруженных объектов, но не ко всей наблюдаемой информации о среде.



Постановка задачи

Задача планирования с помощью языковых моделей с использованием обратной связи:

- Агент в частично наблюдаемой среде
- Использование изображений внутри языковой модели
- Цель планирования – генерация плана действий по текстовой инструкции

Цель работы: создание алгоритма планирования поведения агента с помощью большой языковой модели, учитывающей специфику среды на основе виузальной обратной связи, для задачи манипулирования и навигации, максимизирующий выбранные метрики.

Постановка задачи

Задачей планирования является набор $(S, O, A, T, I, \mathfrak{O}, R)$:

- S – множество состояний среды
- O – множество наблюдений среды
- A – множество возможных действий в среде
- $T : S \times A \to S$ – матрица перехода
- $I \in S$ – начальное состояния среды
- $\mathfrak{O} : S \to O$ – функция наблюдения
- R – инструкция на естественном языке

Целью планирования является генерация последовательности действий $\{a_i\}_{i=1}^{n}$, которая будет выполнять инструкцию R в среде по текущему наблюдаемому состоянию среды $\mathfrak{O}(I) = o$.

Метрики качества

Метрики качества решения задачи:

- **Success Rate** (SR) – это отношение числа успешно завершенных задач к числу всех задач
- **Goal-Condition Success** (GCS) – это число успешно завершенных подзадач к числу всех подзадач на всех задача
- **Success Rate of Replanning** (SRep) – это число предсказанных планов, оцененых человеком как успешные, к числу всех предсказанных планов

Описание датасета

Задача решалась на датасете ALFRED-ChaOS

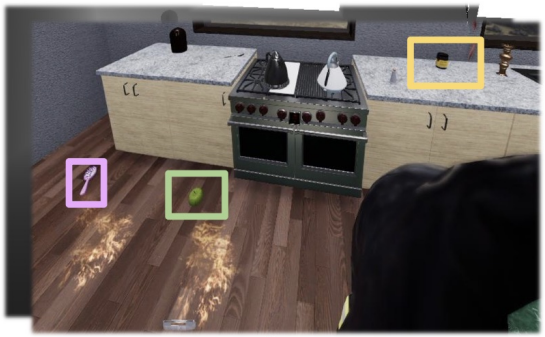
- Датасет основан на ALFRED [4]
- Имеет два сплита
- Первый состоит из 112 уникальных задач из части val seen ALFRED
- Второй из 149 из части val unseen ALFRED
- Каждая задача рассматривается в двух вариациях: с измененными состояниями объектов и без изменений
- Список изменяющихся объектов: Холодильник, Микроволновка, Настольная и Напольная лампа





HAZARD

Observations



RGB-D observation



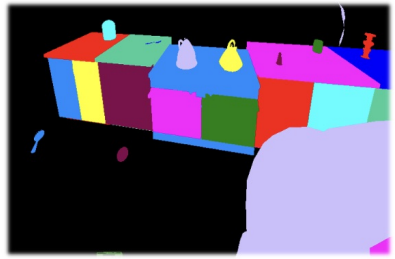
Thermal observation

Toothbrush, value 1

Apple, value 1

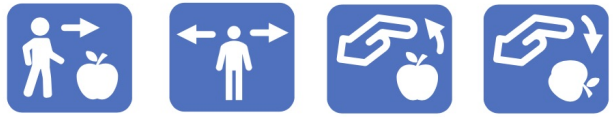
Bottle, value 5

Target object information



Segmentation mask (optional)

Actions



walk to explore pick up drop

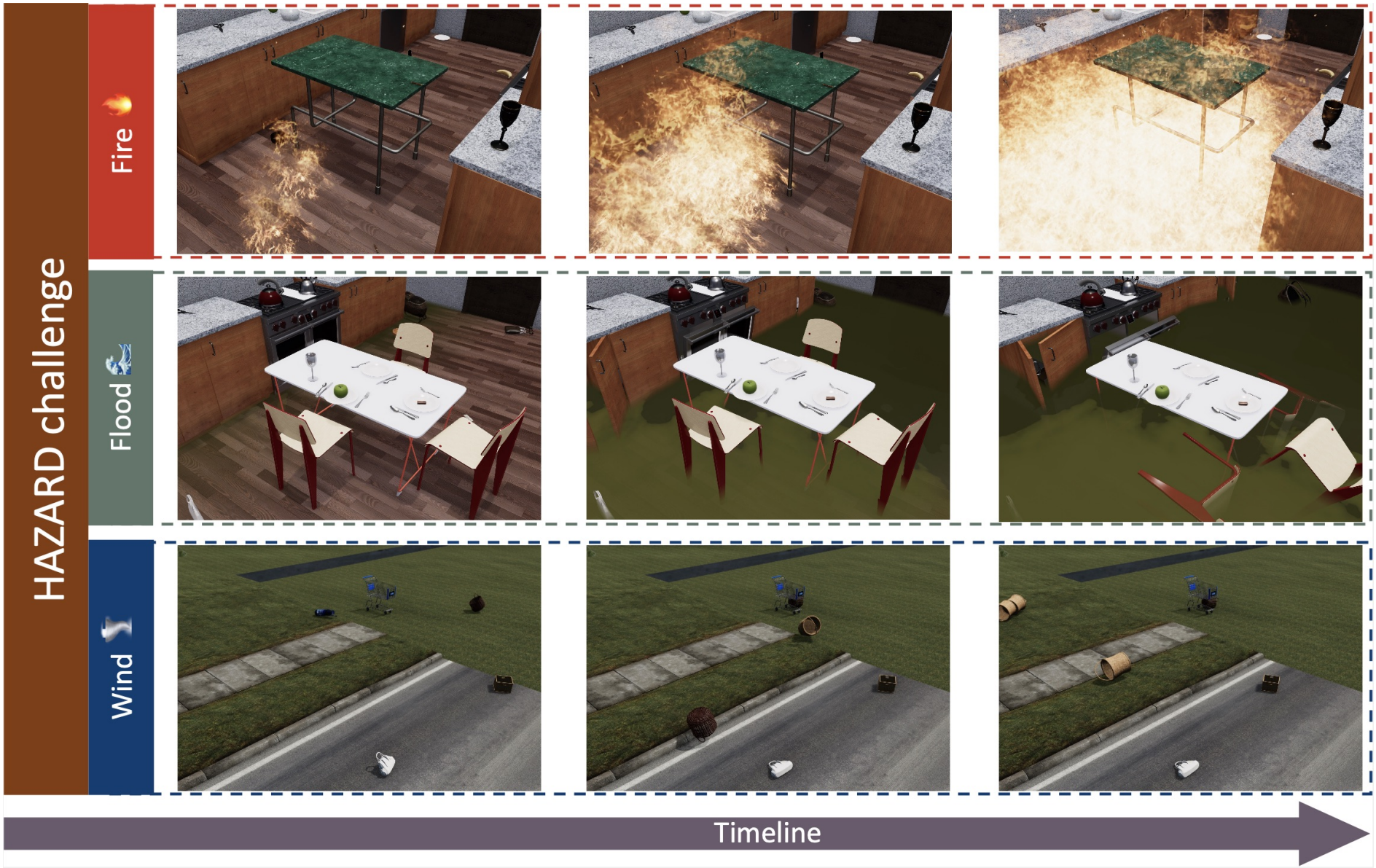
Evaluation metrics

$$\textbf{Value} = \frac{\text{Value of rescued objects}}{\text{Value of all objects}}$$

$$\textbf{Step} = \frac{\text{Total steps taken}}{\text{Number of rescued objects}}$$

$$\textbf{Damage} = \frac{\text{Damaged objects amount}}{\text{Objects amount}}$$

HAZARD



Постановка решенной в работе задачи

Задача ставится следующим образом:

- S – различные состояния и положения объектов, положение агента в среде
- A – move, slice, pick up, put, open, close, turn on, turn off
- O – изображение с камеры от первого лица агента
- I – текстовая инструкция
- $P = [a_0, ..., a_n]$ – текущий план действий
- Ch – индикатор проверяющий успешность исполнения каждого действия плана

Цель – исправление текущего плана действий при возникновении ошибки с учетом визуальной информации от среды.

Описание метода LERa

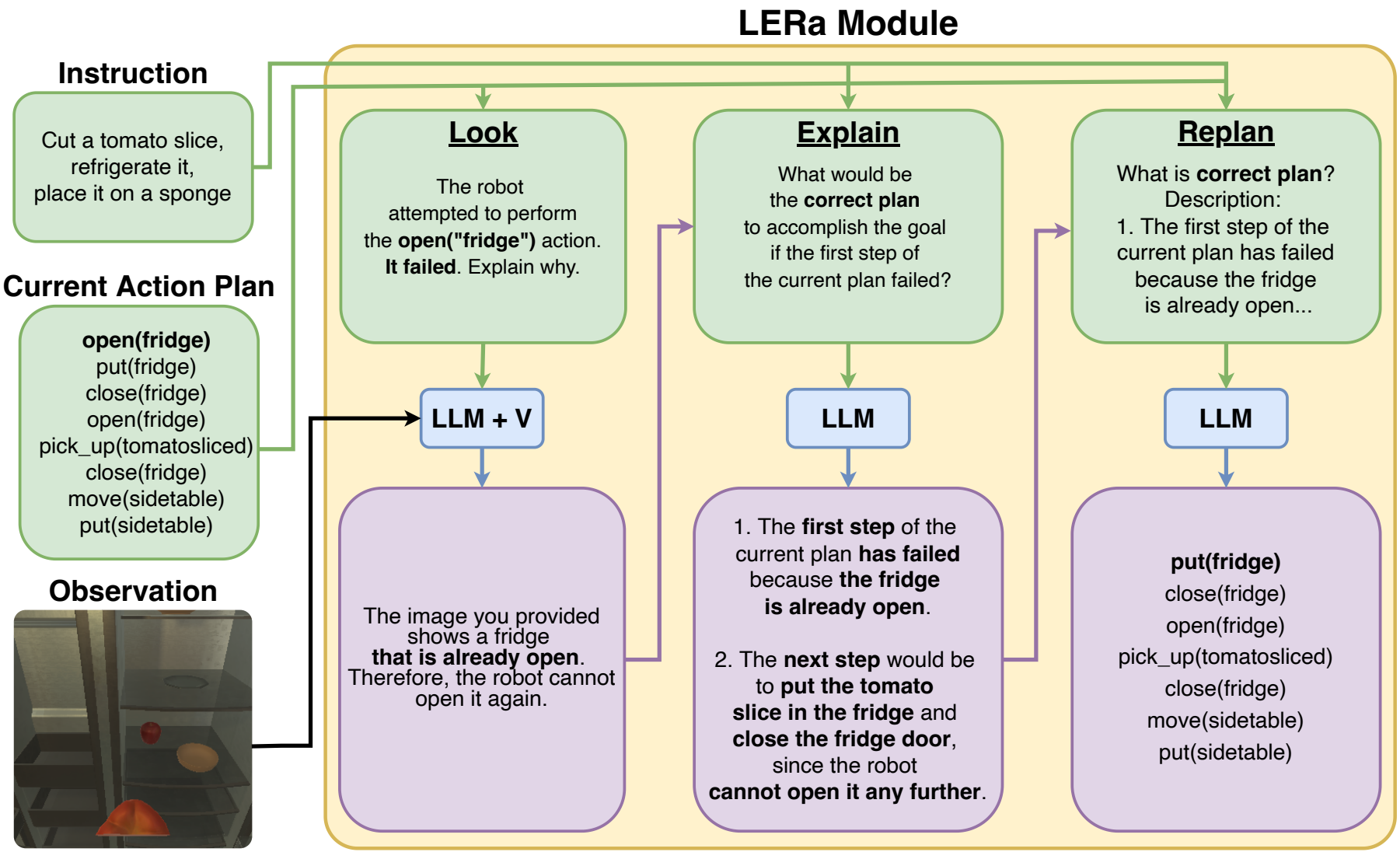
Предложенный метод состоит из трёх этапов:

- L – Look, анализ среды по визуальному наблюдению
- E – Explain, объяснение причины возникновения ошибки
- R – Replan, построение нового плана, учитывающего причины возникшей ошибки

Формальное описание метода

- Пусть есть план $P = [a_0, ..., a_n]$ и на i-ом шаге индекатор Ch сообщил о наличие ошибки.
- Описание сцены $D = L(O_i, a_i)$, где a_i – невыполненное действие и O_i – текущее наблюдение среды
- Описание нового плана действий $Ex = E(D, P, I)$
- Новый план $P' = R(Ex, P, I)$

Архитектура модуля LERa



Описание метода LERa

Шаблоны запросов на модель для каждого шага модуля:

Look request

<Img><ImageHere></Img> The robot attempted to perform the '{}' action. It failed. Based on the given image, explain why.

Explain request

What would be the correct plan to accomplish the goal if the first step of the current plan failed?
Current goal:
{}
Current action plan:
{}
Additional information after checking:
{}

Replan request

What is correct plan?
Current goal:
{}
Current plan:
{}
Description:
{}
Available actions:
[move, slice, open, close, put, pick_up, turn_on, turn_off, done()]

Примеры генерации

Human: The robot attempted to perform the '**open("fridge")**' action. It failed. Explain why.
Assistant: **The refrigerator is already be open.**

Human: **What would be the correct plan to accomplish the goal if the first step of the current plan** failed?
Current goal:
place a chilled bowl in a cabinet
Current action plan:
open("fridge"), put("fridge"), close("fridge"), open("fridge"), pick_up("bowl"), close("fridge"), move("cabinet"), put("cabinet")
Additional information after checking:
The robot failed to open the refrigerator because it is already open. The image shows that the refrigerator door is open, and there is no need for the robot to open it again.
Assistant:
1. The first step of the current plan has failed because the fridge is already open.
2. The next step would be to put bowl in the fridge and close the fridge door, since the robot cannot open it any further.
3. The updated plan would be: put("fridge"), close("fridge"), open("fridge"), pick_up("bowl"), close("fridge"), move("cabinet"), put("cabinet")
4. This updated plan will still accomplish the goal of placing a chilled bowl in a cabinet, but without trying to open the fridge door again, which is already open.

Human: **What is correct plan?**
Current goal:
place a chilled bowl in a cabinet
Current plan:
open("fridge"), put("fridge"), close("fridge"), open("fridge"), pick_up("bowl"), close("fridge"), move("cabinet"), put("cabinet")
Description:
1. The first step of the current plan has failed because the fridge is already open.
2. The next step would be to put bowl in the fridge and close the fridge door, since the robot cannot open it any further.
3. The updated plan would be: put("fridge"), close("fridge"), open("fridge"), pick_up("bowl"), close("fridge"), move("cabinet"), put("cabinet")
4. This updated plan will still accomplish the goal of placing a chilled bowl in a cabinet, but without trying to open the fridge door again, which is already open.
Available actions:
[move, slice, open, close, put, pick_up, turn_on, turn_off, done()]
Assistant: **1. put("fridge"), 2. close("fridge"), 3. open("fridge"), 4. pick_up("bowl"), 5. close("fridge"), 6. move("cabinet"), 7. put("cabinet"), 8. done().**

Описание метода LERa

В работе используется мультимодальная модель MiniGPT-v2 [5]. Для оценки влияния модели на качество использовалась ещё одна модель MiniGPT-4 [6].

- LERa – модуль, использующий MiniGPT-v2
- LERa* – модуль, использующий MiniGPT-4

Бэйзлайны

- **Oracle** – агент, имеющий доступ ко всей информации в среде. Не способен к изменению плана действий
- **OracleN** – Oracle агент имеющий неидеальный индикатор возникновения ошибки
- **Oracle + PC** – агент способный перемещаться в небольшой области (корректировать позу) при неудаче исполнения действия
- **Oracle + LERa** – Oracle агент с модулем перепланирования LERa
- **Oracle + ERa** – Oracle агент с модулем перепланирования LERa, однако не использующим только текстовую обратную связь от среды
- **FILM** – агент основанный на одноименной работе FILM [7]
- **FILM + LERa** – FILM с модулем перепланирования
- **Base** – Базовый агент для среды **HAZARD**

Основные результаты в среде ALFRED

Результаты выполнения заданий из датасета ALFRED-ChaOS.
Сравнение на каждом из разбиений **seen** и **unseen**.

	Seen split			**Unseen split**		
Agent	SR	GCS	SRep	SR	GCS	SRep
Oracle	33.04	50.04	-	31.65	51.71	-
Oracle + PC	36.61	53.24	-	35.25	55.31	-
Oracle + LERa	**46.88**	**60.34**	**46.61**	**46.04**	**62.71**	**47.80**
FILM	16.52	34.04	-	**16.91**	36.66	-
FILM + LERa	**16.96**	**34.30**	**70.58**	16.55	**36.78**	**18.18**

Основные результаты в среде HAZARD

Результаты прохождения задач в среде HAZARD. Представлены задачи **Fire** и **Flood**.

	Fire			**Flood**		
Agent	Value↑	Damage↓	Step↓	Value↑	Damage↓	Step↓
Base	0.16	0.27	**392.76**	0.14	**0.65**	382.33
Base+ERa	0.17	**0.23**	428.98	**0.12**	0.72	388.80
Base+LERa	**0.15**	0.24	416.73	0.14	0.69	**353.27**

Оценка вклада визульной информации в качество планирования

Сравнение выполнения задачи Oracle агента, **различных модификаций**.

	Seen split			Unseen split		
Agent	SR	GCS	SRep	SR	GCS	SRep
Oracle	33.04	50.04	-	31.65	51.71	-
Oracle + ERa	34.82	50.93	6.89	42.81	61.42	23.15
Oracle + LERa	**46.88**	**60.34**	**46.61**	**46.04**	**62.71**	**47.80**

Сравнение качества предсказаний каждого отдельного шага модуля LERa **при отсутствии и наличии визуальной связи**.

	Seen split			Unseen split		
Agent	LS	ES	RS	LS	ES	RS
Oracle + ERa	-	6.89	6.89	-	63.54	23.15
Oracle + LERa	**66.10**	**44.91**	**46.61**	**74.84**	**47.80**	**47.80**

Основные результаты в среде HAZARD

Эксперименты с **неидеальными проверяющими** выполнения задания.

	Seen split			Unseen split		
Agent	SR	GSR	SRep	SR	GSR	SRep
Oracle0.15	12.50	35.90	-	12.95	39.75	-
Oracle0.10	17.41	40.89	-	17.63	43.38	-
Oracle0.05	24.55	45.01	-	24.82	47.54	-
OracleFC	33.04	50.04	-	31.65	51.71	-
Oracle0.15 + LERa	22.32	44.42	35.48	20.50	45.29	25.00
Oracle0.10 + LERa	25.45	46.91	47.32	25.18	48.95	28.70
Oracle0.05 + LERa	34.38	52.57	**54.16**	35.97	55.76	35.43
OracleFC + LERa	**44.64**	**59.00**	52.83	**46.76**	**63.46**	**49.34**

Оценка влияния языковой модели

Сравнение выполнения эпизодов из датасета **при различных визуально языковых моделях MiniGPT-4** и **MiniGPT-4-v2**.

Agent	**SR%**	**GCS%**	**SRep%**
Oracle + LERa*	34.38	50.93	6.74
Oracle + LERa	**46.88**	**60.34**	**46.61**

Анализ результатов

Использование модуля LERa в работе агента значительно повышает качество работы:

- Модуль позволяет лучше решать задачи планирования, даже на примерах, не использованных в few-shot примерах
- Использование визуальной обратной связи значительно увеличивает число правильно решенных задач
- Визуальная информация позволяет модулю лучше предсказывать объяснение и новый план
- Модуль LERa показал, что способен исправлять ошибки исполнения, даже если ошибки не было
- Использование более совершенных моделей качественно улучшает работу всего модуля

Вклад в работу

Данная работа выполнялась группой исследователей. Моими задачами в рамках командной работы было:

- Формирование архитектуры LERa модуля
- Проведение эксперименты с выбором шаблонов запросов для модели на каждом из шагов модуля
- Проведение экспериментов с LERa, LERa* и ERa модулями

Текущие задачи

Текущие задачи и первые результаты:

- Разрабатывается метод пошагового предсказания низкоуровневых действий и обучения стратегии решения высокоуровневых действий
- Подготовлены текстовые инструкции и высокоуровневые планы
- Реализована инфраструктура для взаимодействия с различными vlm и виртуальными средами

Новые виртуальные среды

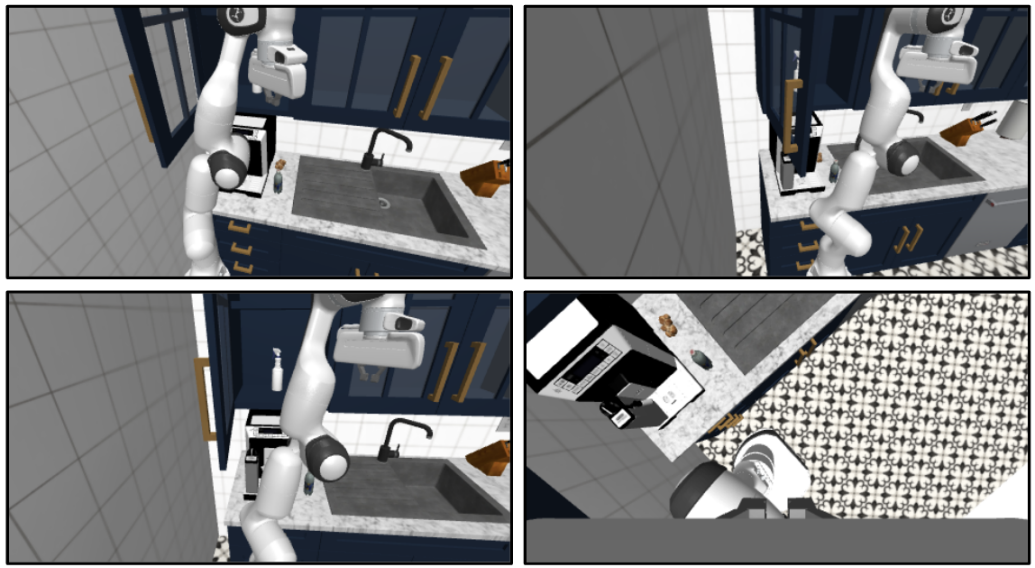
Используются среды для решения задачи манипулировния с роботизированной рукой:

- **Robosuite**[8]
- **Robocasa**[9]

Robosuite



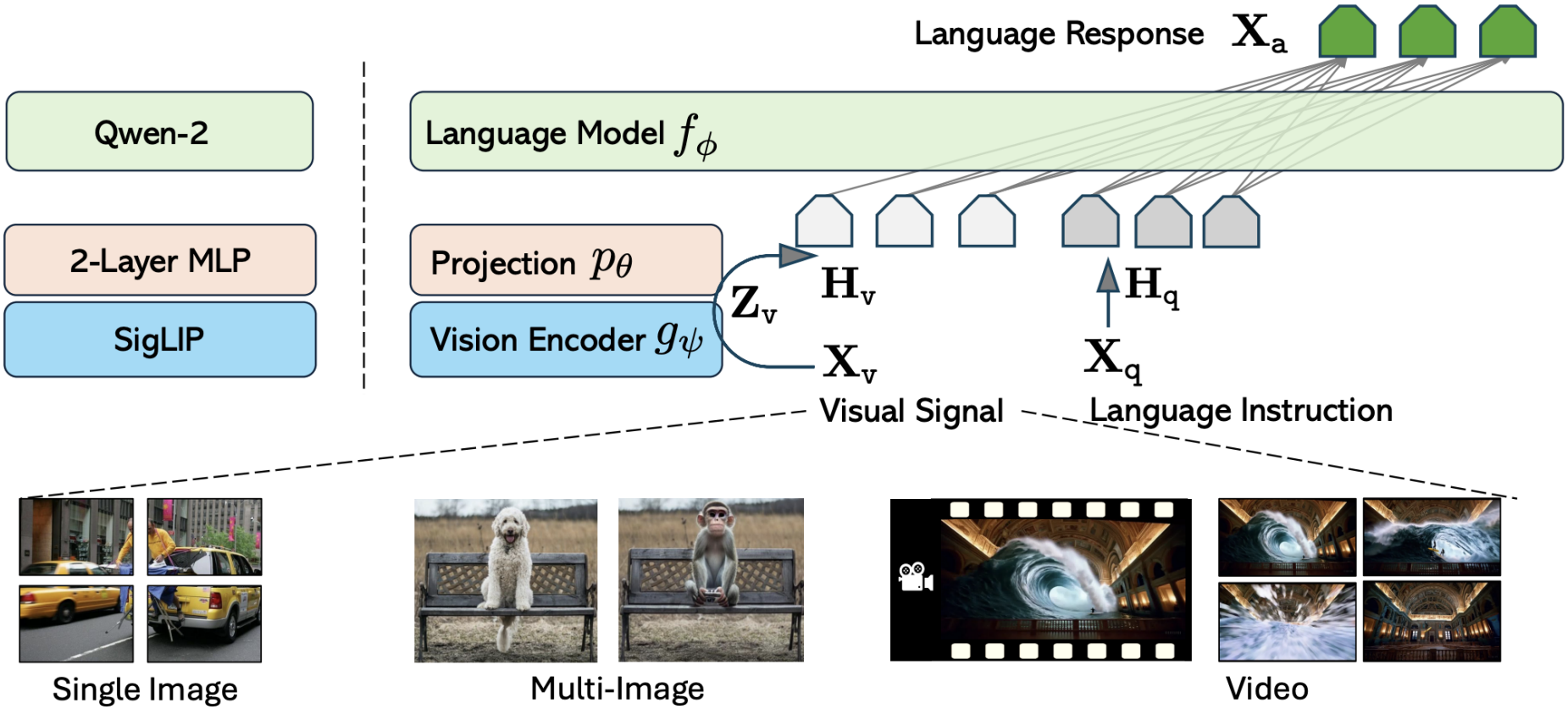
RoboCasa



Новые визуально языковые модели

В текущих экспериментах используются модели из семейства LLaVA для работы с несколькими изображениями:

- **LLaVA-NeXT-Interleave**[10]
- **LLaVA-OneVision**[11]



Новые текстовые инструкции

Сгенерированы 900 текстовых инструкций и соответствующие к ним планы действий. Планы различаются целевыми объектами и их свойствами.

- **Task:** *put square green plate on square yellow teapot*
  GT plan: *[(move to, square green plate), (pick up, square green plate), (move to, square yellow teapot), (put down, square yellow teapot), (done,)]*
- **Task:** *put square white apple to green spoon*
  GT plan: *[(move to, square white apple), (pick up, square white apple), (move to, green spoon), (put down, green spoon), (done,)]*

Качество предсказания плана

На наборе инструкций оценены способности выбранных моделей предсказывать полный план действий. Метрики **CER** – отношение совпавших символов к числу всех, **WER** – отношение совпавших слов к числу всех слов, **Eq** – число полностью правльно предсказанных планов ко всем планам.

Model	**CER**	**WER**	**Eq**
LLaVA-NeXT-Interleave	0.4	0.4	0.0
LLaVA-OneVision	0.4	0.42	0.0

Источники I



Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su.
Llm-planner: Few-shot grounded planning for embodied agents with large language models.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009, 2023.



Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al.
Do as i can, not as i say: Grounding language in robotic affordances.
arXiv preprint arXiv:2204.01691, 2022.

Источники II



Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al.
Inner monologue: Embodied reasoning through planning with language models.
In *Conference on Robot Learning*, pages 1769–1782. PMLR, 2023.



Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox.
ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks.
In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Источники III



Jun Chen.
Minigpt-v2: Large language model as a unified interface for vision-language multi-task learning.
arXiv:2310.09478, 2023.



Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny.
Minigpt-4: Enhancing vision-language understanding with advanced large language models.
2023.



So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov.
Film: Following instructions in language with modular methods.
arXiv preprint arXiv:2110.07342, 2021.

Источники IV



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.



Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu.
Robocasa: Large-scale simulation of everyday tasks for generalist robots.
In *Robotics: Science and Systems*, 2024.

Источники V



Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li.
Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models.
arXiv preprint arXiv:2407.07895, 2024.



Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li.
Llava-onevision: Easy visual task transfer.
arXiv preprint arXiv:2408.03326, 2024.