

Путь к LLM-Агентам



Ваня Ершов

Старший разработчик
службы ML разработки Алисы

Кто я?



Ваня Ершов

Старший разработчик
службы ML разработки Алисы

- Занимаюсь LLM-Агентов и внедряю агентов в Алису
- В прошлом: ASR для Алисы
- В прошлом: выпускник ФКН ПМИ 2024

Про что поговорим

01 Какую задачу вообще решаем

02 Chain-of-Thought

03 LLM как мозг или что такое func-calling

04 Что такое LLM-Agent

05 Как обучать

01

Проблематика

Какую задачу мы вообще решаем
и в чем проблемы прошлых подходов

Задача

Задача

🎯 Задание дня! Желаю успехов в выполнении! 📝 Задание 1/6. Шифр Цезаря. Описание: Шифр Цезаря - это простой метод шифрования, при котором каждая буква текста заменяется буквой, находящейся на фиксированное количество позиций ниже в алфавите. Например, при сдвиге на 3 буквы "Г", "Д", "Е" становятся "А", "Б", "В" соответственно. 🔑 Задание: Расшифруйте сообщение, если известно, что оно на русском языке и было зашифровано с использованием определенных параметров русского и английского алфавитов. Шифрованное сообщение: пфыхчужэпхффжё злохцжшфхшщг сдвиг на 7.

Задача

🎯 Задание дня! Желаю успехов в выполнении! 📝 Задание 1/6. Шифр Цезаря. Описание: Шифр Цезаря - это простой метод шифрования, при котором каждая буква текста заменяется буквой, находящейся на фиксированное количество позиций ниже в алфавите. Например, при сдвиге на 3 буквы "Г", "Д", "Е" становятся "А", "Б", "В" соответственно. 🔑 Задание: Расшифруйте сообщение, если известно, что оно на русском языке и было зашифровано с использованием определенных параметров русского и английского алфавитов. Шифрованное сообщение: пфыхчужэпхффжё злохцжшфхшшг сдвиг на 7.

Найти частное решение дифференциального уравнения, соответствующее заданному начальному условию $y(0) = -1$.

$$\sqrt{1-x^2} \cdot y' + y = \arcsin(x) \cdot y^2$$

Задача

🎯 Задание дня! Желаю успехов в выполнении! 📝 Задание 1/6. Шифр Цезаря. Описание: Шифр Цезаря - это простой метод шифрования, при котором каждая буква текста заменяется буквой, находящейся на фиксированное количество позиций ниже в алфавите. Например, при сдвиге на 3 буквы "Г", "Д", "Е" становятся "А", "Б", "В" соответственно. 🔑 Задание: Расшифруйте сообщение, если известно, что оно на русском языке и было зашифровано с использованием определенных параметров русского и английского алфавитов. Шифрованное сообщение: пфыхчужэпхффжё злохцжшфхшшг сдвиг на 7.

Найти частное решение дифференциального уравнения, соответствующее заданному начальному условию $y(0) = -1$.

$$\sqrt{1-x^2} \cdot y' + y = \arcsin(x) \cdot y^2$$

Напиши на python программу которая на вход получает два аргумента: имя файла и строку.
Задача программы вывести все строки файла и их номера содержащие заданную строку.
Каждое совпадение должно быть подсвеченено цветом радуги по очереди начиная с красного.
Используй 256-цветную палитру для терминала.
Программа должна использовать только встроенные библиотеки языка без дополнительных внешних модулей.

Лакмусовый grep

Напиши на python программу которая на вход получает два аргумента: имя файла и строку. Задача программы вывести все строки файла и их номера содержащие заданную строку. Каждое совпадение должно быть подсвеченено цветом радуги по очереди начиная с красного. Используй 256-цветную палитру для терминала. Программа должна использовать только встроенные библиотеки языка без дополнительных внешних модулей.

Лакмусовый grep

```
import sys
import os

# Функция для получения цвета из 256-цветовой палитры
def get_rainbow_color(index):
    # Определяем базовые цвета радуги в 256-цветной палитре
    rainbow_colors = [
        196, # Красный
        208, # Оранжевый
        222, # Жёлтый
        46, # Зелёный
        34, # Голубой
        61, # Синий
        123 # Фиолетовый
    ]
    return rainbow_colors[index % len(rainbow_colors)]

# Функция для вывода строки с подсвечиванием заданной подстроки
def print_colored_line(line, search_string, color_index):
    color = get_rainbow_color(color_index)
    # Формируем управляющие последовательности для установки и сброса цвета
    set_color = f'\033[38;5;{color}m'
    reset_color = '\033[0m'
    # Разделяем строку на части до и после искомой подстроки
    parts = line.split(search_string)
    # Выводим строку с подсветкой
    for i, part in enumerate(parts):
        print(part, end=' ')
        if i < len(parts) - 1:
            print(f'{set_color}{search_string}{reset_color}', end=' ')
    print()
```

Напиши на python программу которая на вход получает два аргумента: имя файла и строку.
Задача программы вывести все строки файла и их номера содержащие заданную строку.
Каждое совпадение должно быть подсвеченено цветом радуги по очереди начиная с красного.
Используй 256-цветную палитру для терминала.
Программа должна использовать только встроенные библиотеки языка без дополнительных внешних модулей.

Лакмусовый grep

```
import sys
import os

# Функция для получения цвета из 256-цветовой палитры
def get_rainbow_color(index):
    # Определяем базовые цвета радуги в 256-цветной палитре
    rainbow_colors = [
        196, # Красный
        208, # Оранжевый
        222, # Жёлтый
        46, # Зелёный
        34, # Голубой
        61, # Синий
        123 # Фиолетовый
    ]
    return rainbow_colors[index % len(rainbow_colors)]

# Функция для вывода строки с подсвечиванием заданной подстроки
def print_colored_line(line, search_string, color_index):
    color = get_rainbow_color(color_index)
    # Формируем управляющие последовательности для установки и сброса цвета
    set_color = f'\u033[38;5;{color}m'
    reset_color = '\u033[0m'
    # Разделяем строку на части до и после искомой подстроки
    parts = line.split(search_string)
    # Выводим строку с подсветкой
    for i, part in enumerate(parts):
        print(part, end='')
        if i < len(parts) - 1:
            print(f'{set_color}{search_string}{reset_color}', end=' ')
    print()
```

Напиши на python программу которая на вход получает два аргумента: имя файла и строку.
Задача программы вывести все строки файла и их номера содержащие заданную строку.
Каждое совпадение должно быть подсвеченено цветом радуги поочереди начиная с красного.
Используй 256-цветную палитру для терминала.
Программа должна использовать только встроенные библиотеки языка без дополнительных внешних модулей.

Проверка:

```
klyukinds-osx:h_event klyukinds$ python3 pro_highlight.py test_example.txt search
3: def highlight_matches(filename, search_string):
19:             if search_string in line: # search
23:                 start = line.find(search_string, start)
31:                 f'\u033[38;5;{color_code}m{search_string}\u033[0m' +
32:                     line[start + len(search_string):]
36:                     start += len(search_string)
48:     search_string = sys.argv[2]
49:     highlight_matches(filename, search_string)
```

Проблемы

Проблемы

01

Отсутствие самопроверки

Хочется, чтобы модель
саморефлексировала над тем,
что она сделала.

Хочется, чтобы был у нее “aha-moment”, когда понимает,
что что-то идет не так

Проблемы

01

Отсутствие самопроверки

Хочется, чтобы модель
саморефлексировала над тем,
что она сделала.

Хочется, чтобы был у нее “aha-moment”, когда понимает,
что что-то идет не так

02

Галлюцинации

Модели обладают большим
количество знаний,
но как и люди (даже хуже людей)
путают знания.

Нельзя забывать, что модель —
это все еще модель
предсказывающая
распределение токенов

Проблемы

01

Отсутствие самопроверки

Хочется, чтобы модель саморефлексировала над тем, что она сделала.

Хочется, чтобы был у нее “aha-moment”, когда понимает, что что-то идет не так

02

Галлюцинации

Модели обладают большим количеством знаний, но как и люди (даже хуже людей) путают знания.

Нельзя забывать, что модель — это все еще модель предсказывающая распределение токенов

03

Нет взаимодействия с окружающей средой

Модель решает прикладные задачи, а не просто в вакууме

Хочется, чтобы модель знала о среде и использовала в свою пользу

02

Chain-of-thought

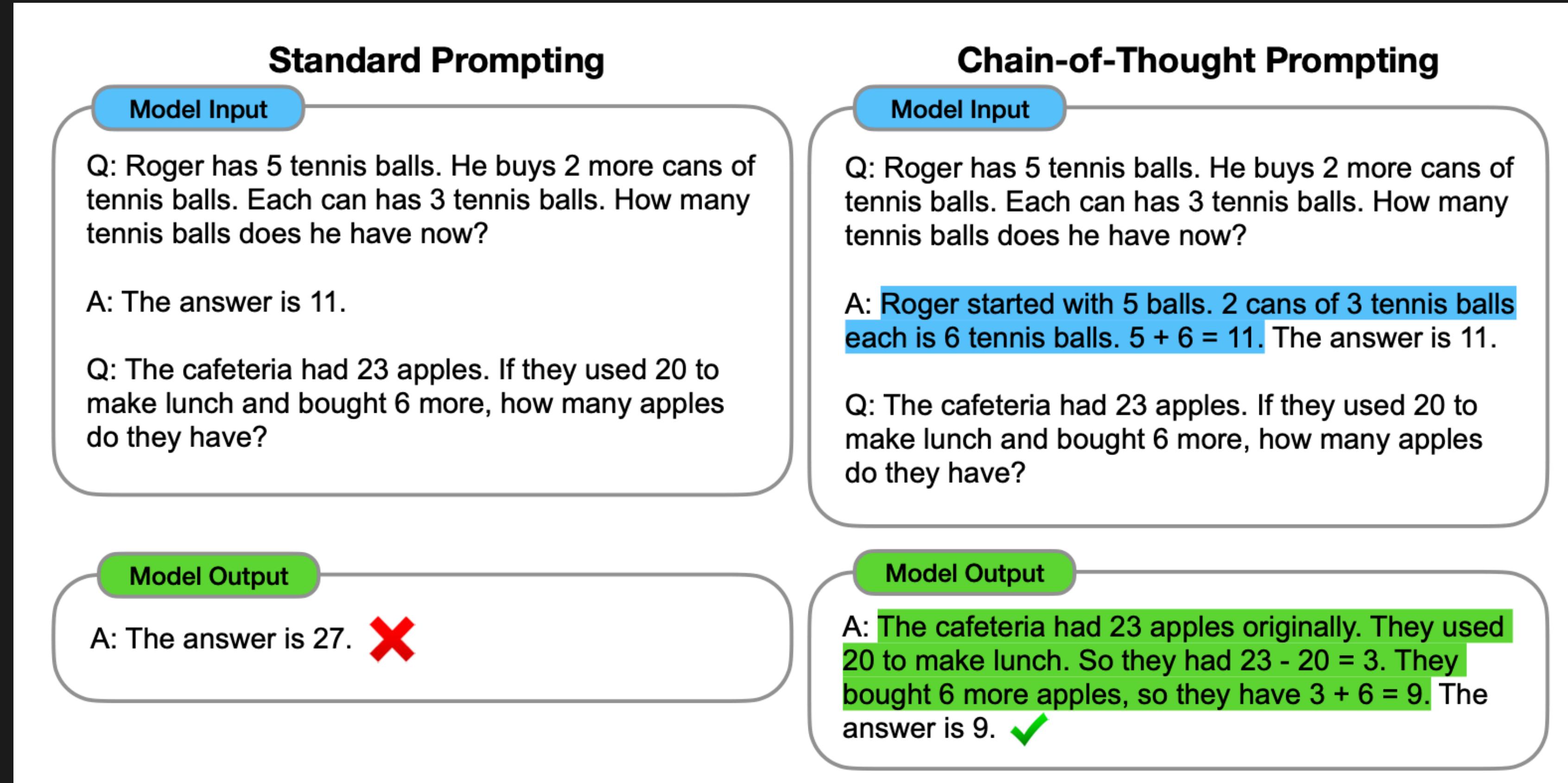
Рассуждения / Ризонинг / Саморефлексия

Зачем нужен СоT

- 01 Debug нерабочего код
- 02 Комплексное объяснение
- 03 Генерация структурированного контента
- 04 Решение математики

Chain-of-Thought prompting

В самом начале СоТ делался через промпт



Chain-of-Thought prompting

В самом начале СоТ делался через промпт

PROMPT FOR MATH WORD PROBLEMS

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?

A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been $21 - 15 = 6$. The answer is 6.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?

A: There are originally 3 cars. 2 more cars arrive. $3 + 2 = 5$. The answer is 5.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?

A: Originally, Leah had 32 chocolates. Her sister had 42. So in total they had $32 + 42 = 74$. After eating 35, they had $74 - 35 = 39$. The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?

A: Jason started with 20 lollipops. Then he had 12 after giving some to Denny. So he gave Denny $20 - 12 = 8$. The answer is 8.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?

A: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 4 more toys. $5 + 4 = 9$. The answer is 9.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

A: There were originally 9 computers. For each of 4 days, 5 more computers were added. So $5 * 4 = 20$ computers were added. $9 + 20$ is 29. The answer is 29.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?

A: Michael started with 58 golf balls. After losing 23 on tuesday, he had $58 - 23 = 35$. After losing 2 more, he had $35 - 2 = 33$ golf balls. The answer is 33.

Q: Olivia has \$23. She bought five bagels for \$3 each. How much money does she have left?

A: Olivia had 23 dollars. 5 bagels for 3 dollars each will be $5 \times 3 = 15$ dollars. So she has $23 - 15$ dollars left. $23 - 15$ is 8. The answer is 8.

Chain-of-Thought prompting

В самом начале CoT делался через промпт

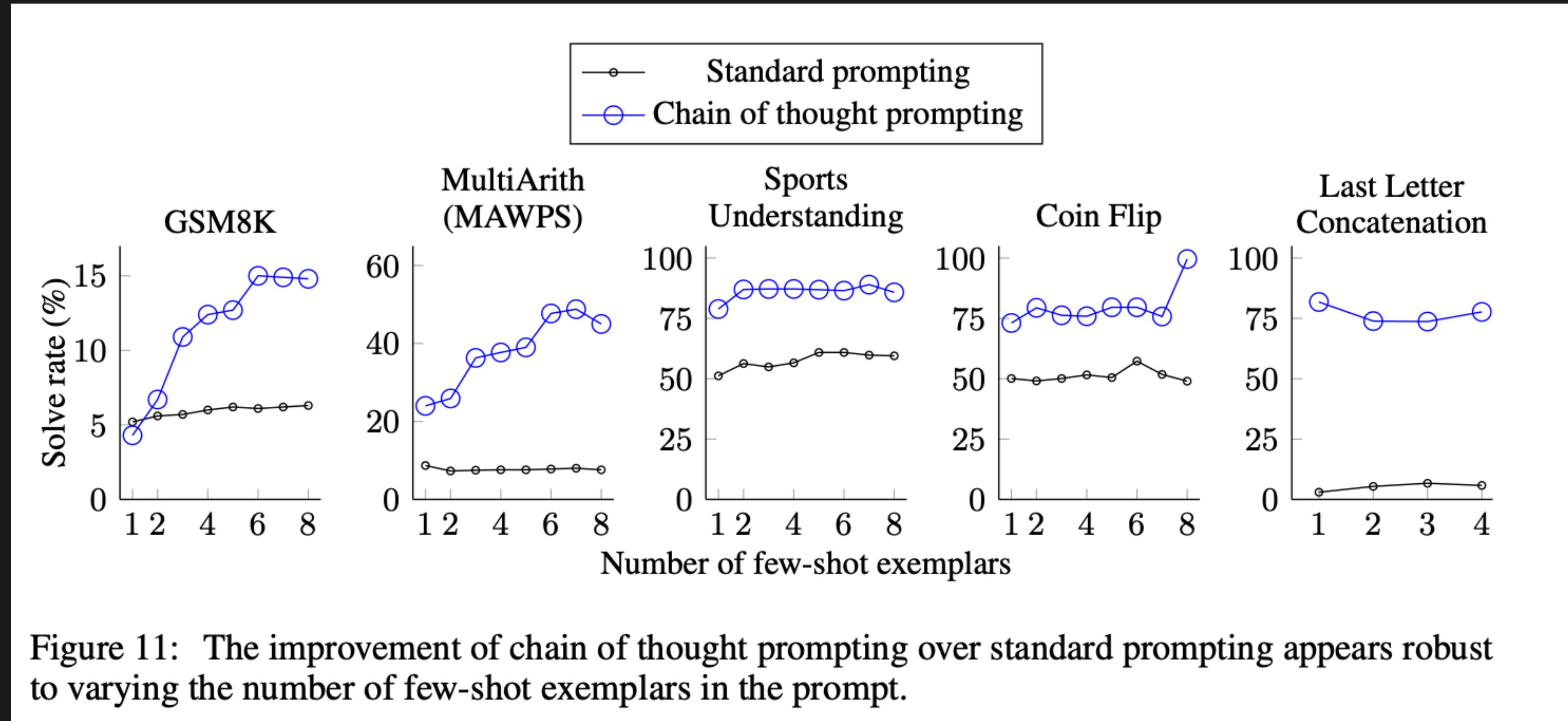


Figure 11: The improvement of chain of thought prompting over standard prompting appears robust to varying the number of few-shot exemplars in the prompt.

Chain-of-Thought prompting

В самом начала СоТ делался через промпт

- Промпт это хорошо, но...

Chain-of-Thought prompting

В самом начала СоТ делался через промпт

- Промпт это хорошо, но...
- Интересно, что LLM “из коробки” умеет думать

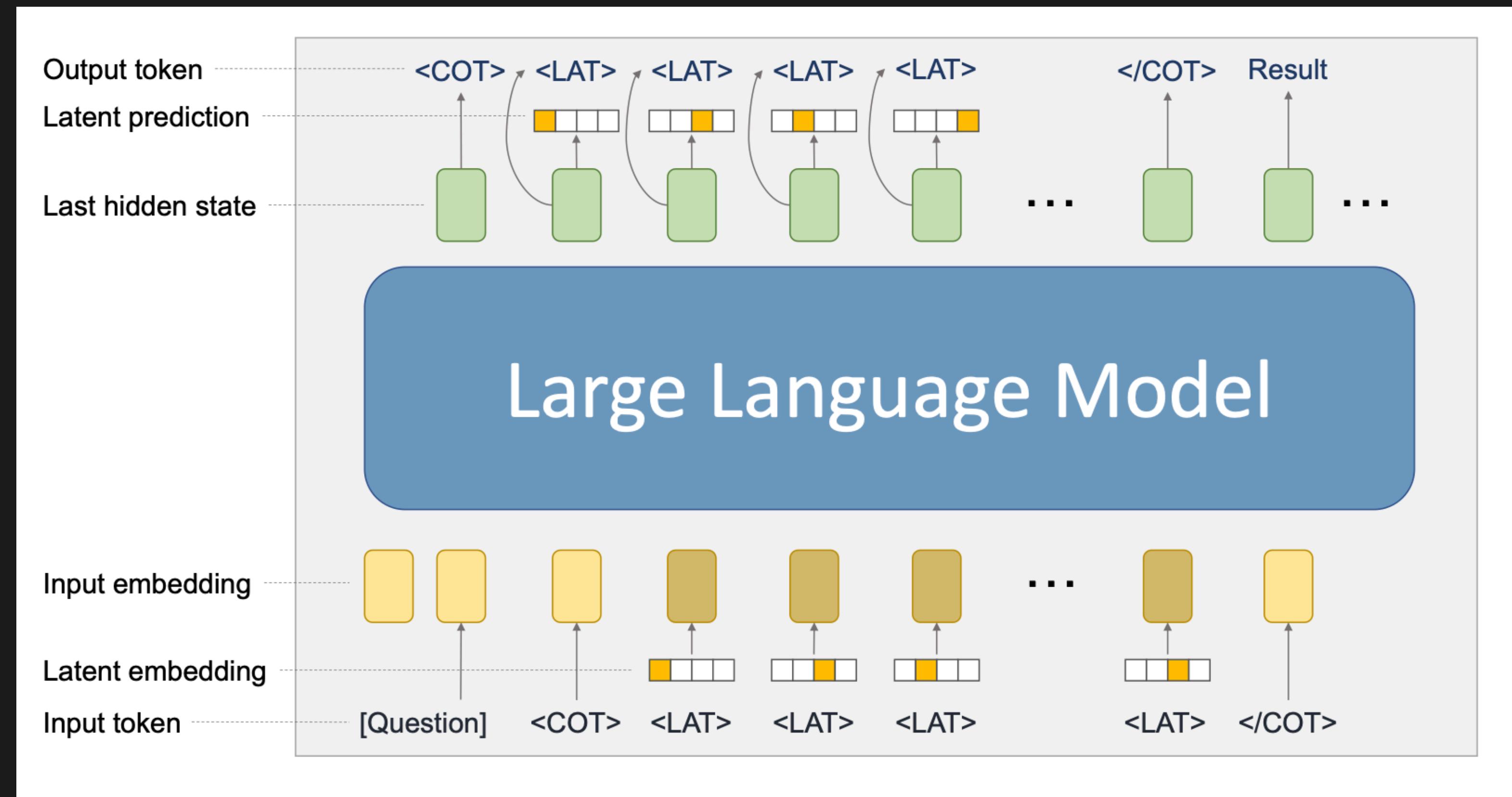
Chain-of-Thought prompting

В самом начале СоТ делался через промпт

- Промпт это хорошо, но...
- Интересно, что LLM “из коробки” умеет думать
- Хочется:
 1. Иметь больше контроля над этим процессом
 2. Улучшить его
 3. Отказаться от промптирования

Chain-of-Thought tokens

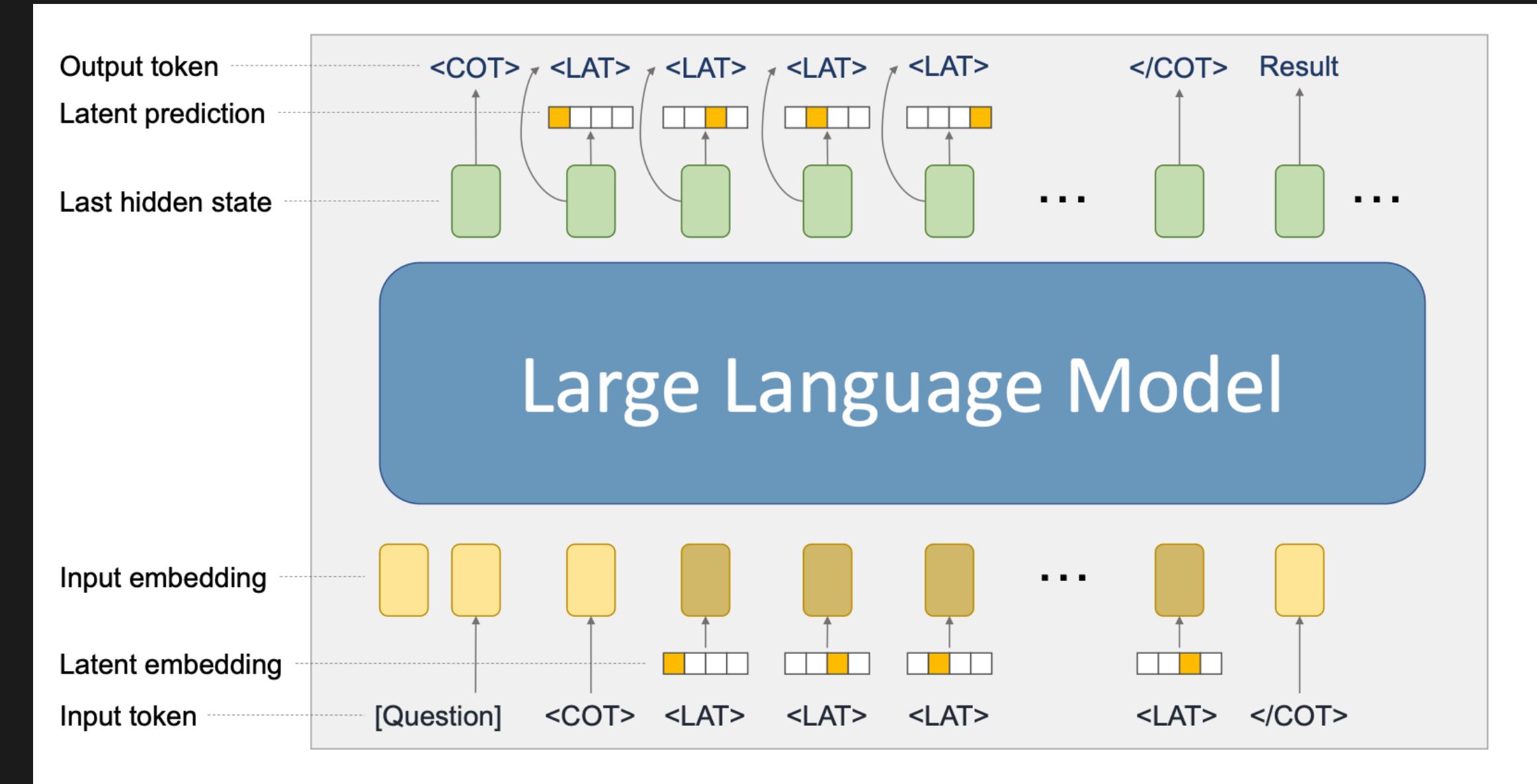
Размышления помещаем в специальное место



Chain-of-Thought tokens

01

Визуально красиво 🎨



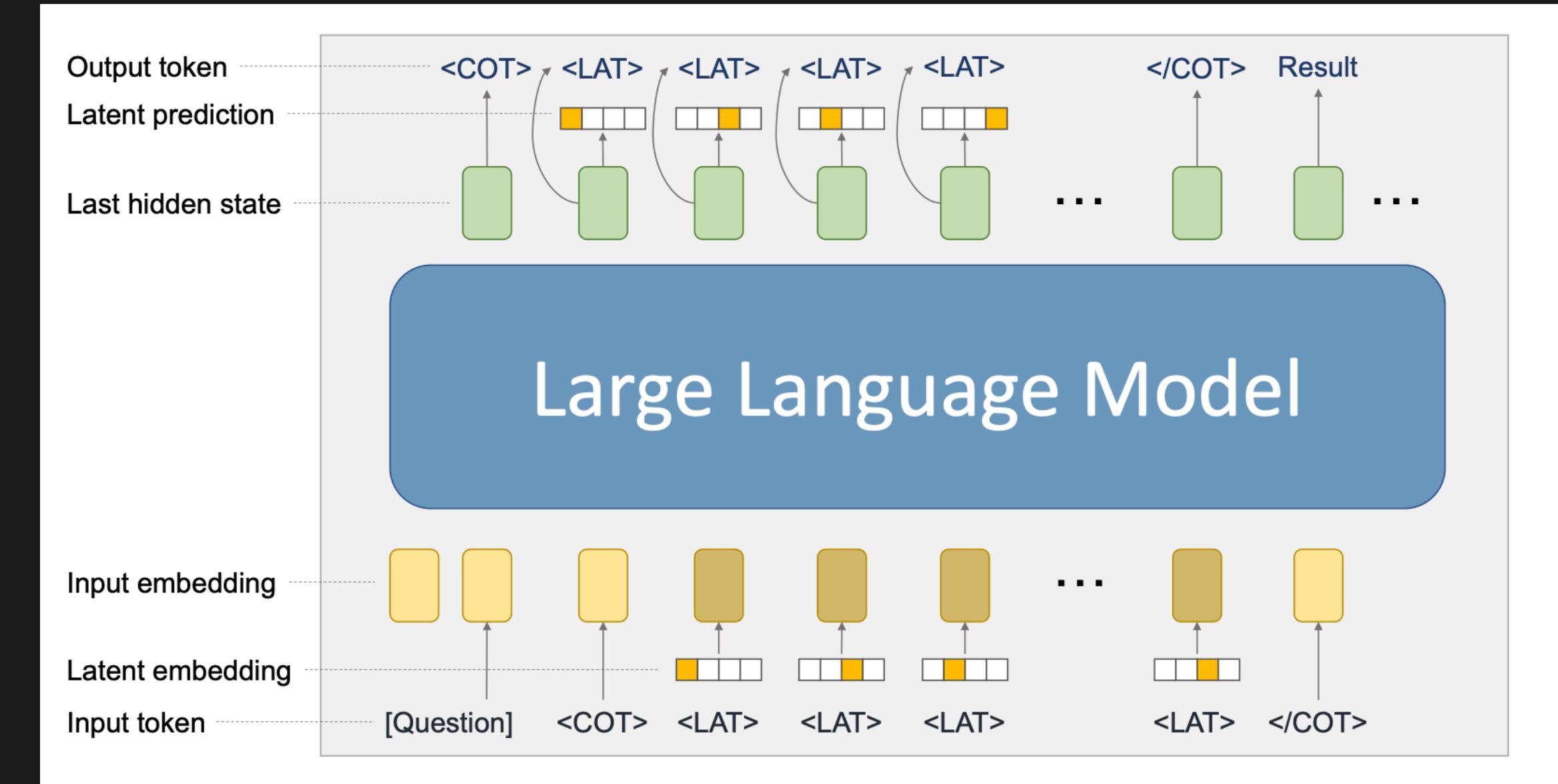
Chain-of-Thought tokens

01

Визуально красиво 🎨

02

Вкл/Выкл через токен COT_ENABLE



Chain-of-Thought tokens

01

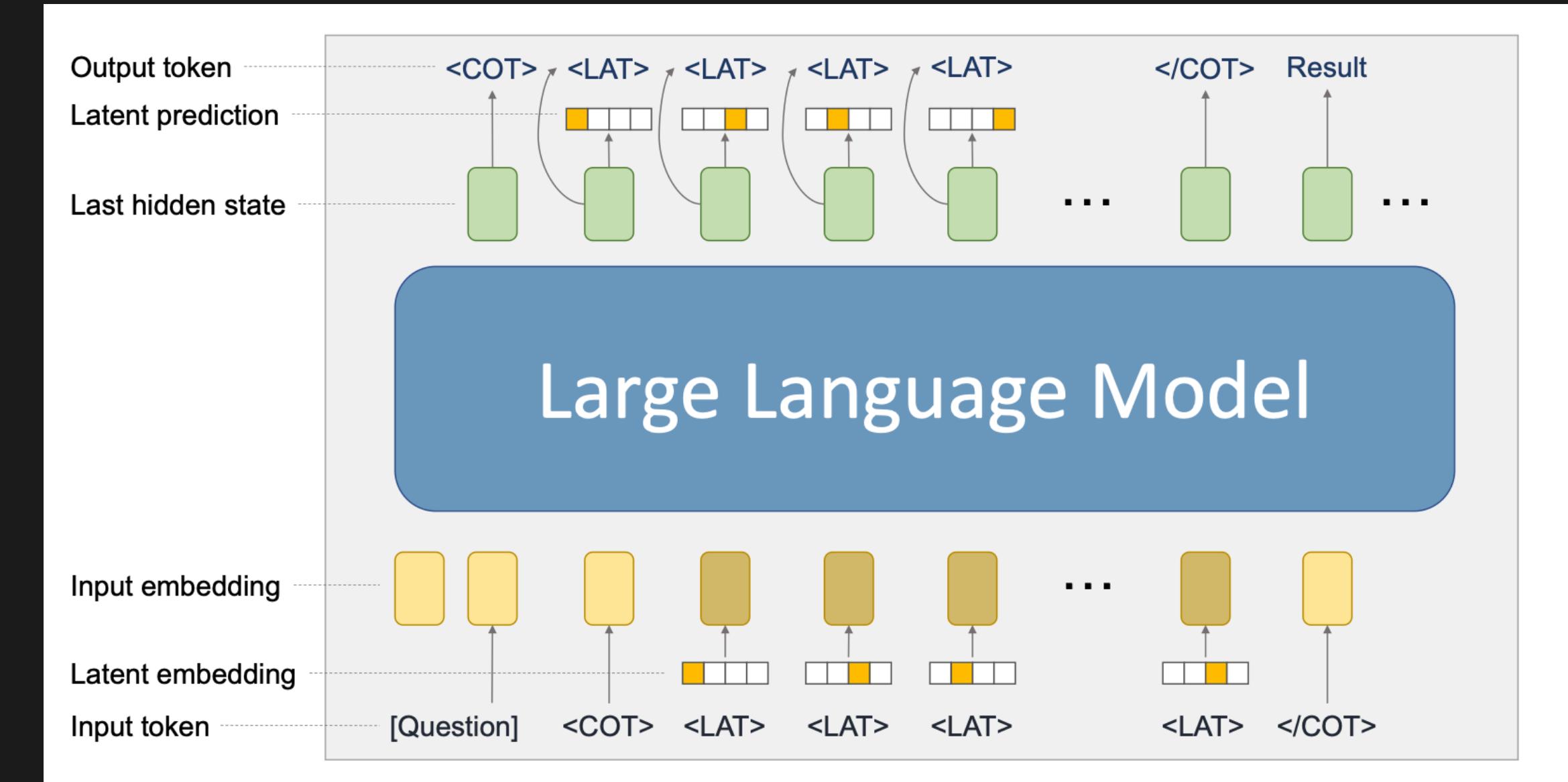
Визуально красиво 🎨

02

Вкл/Выкл через токен SOT_ENABLE

03

Можем просить модель думать дольше



Chain-of-Thought tokens

01

Визуально красиво 🎨

02

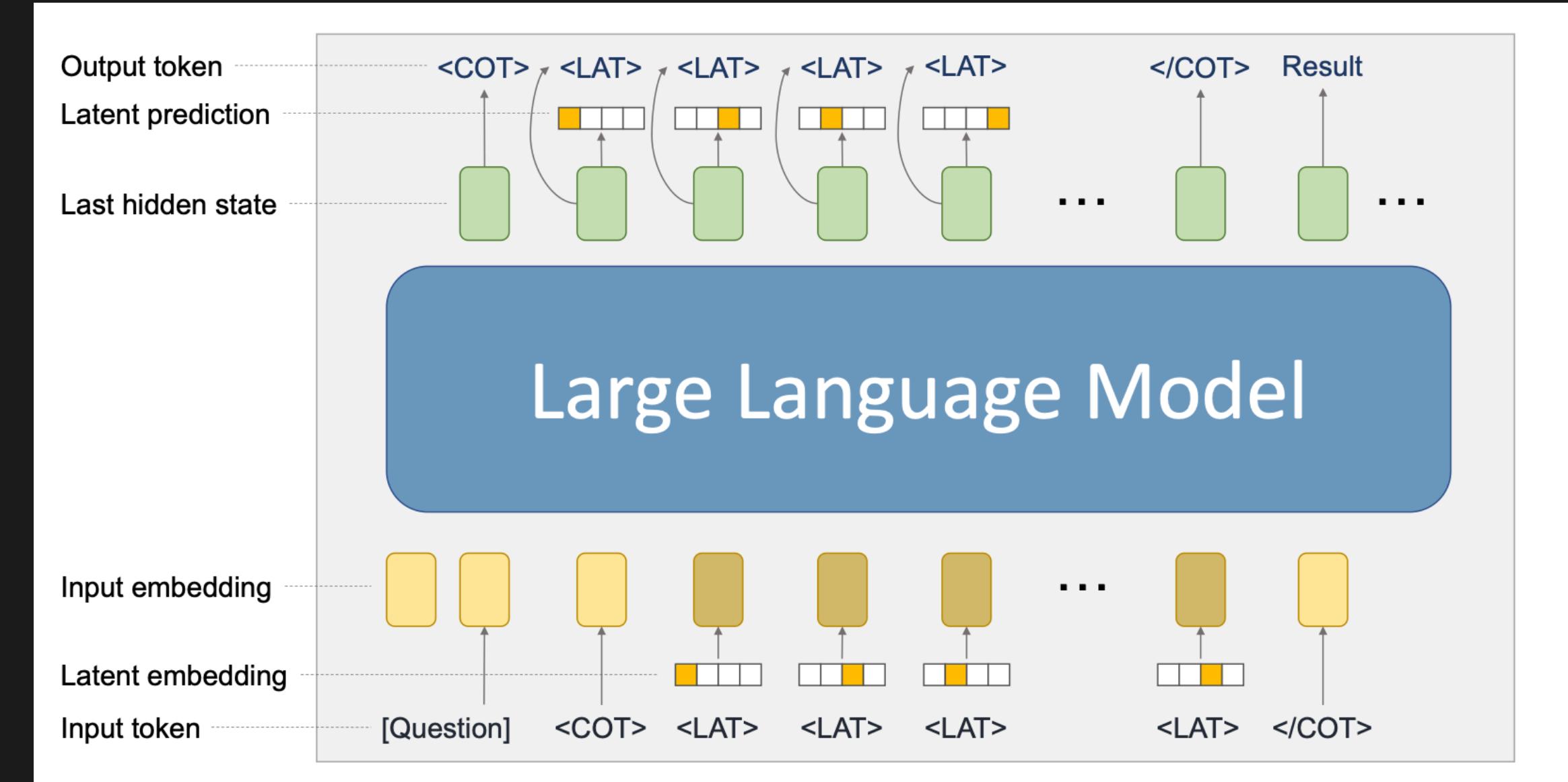
Вкл/Выкл через токен SOT_ENABLE

03

Можем просить модель думать дольше

04

Можем вырезать СоT из контекста, чтобы не захламлять контекст
(частично решает проблему работы с длинным контекстом)



03

LLM function calling

Как модель может выйти за рамки болталки

Пойду издалека: зачем это нужно

Почему бы не написать классификатор
ответа LLM, чтобы он вызывал функции?

Пойду издалека: зачем это нужно

Почему бы не написать классификатор
ответа LLM, чтобы он вызывал функции?

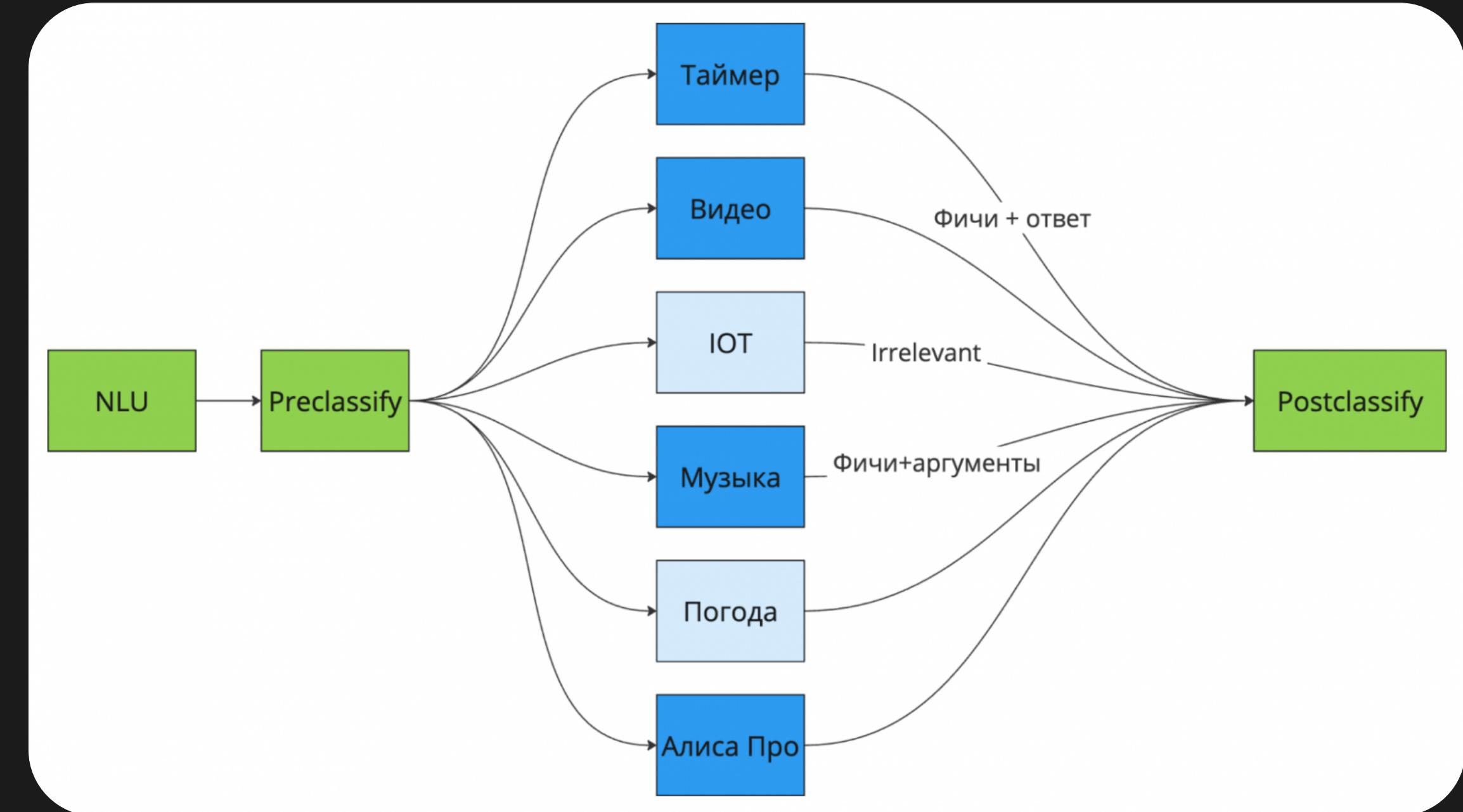
Разберем на примере Алисы!



Пойду издалека: зачем это нужно

Почему бы не написать классификатор ответа LLM, чтобы он вызывал функции?

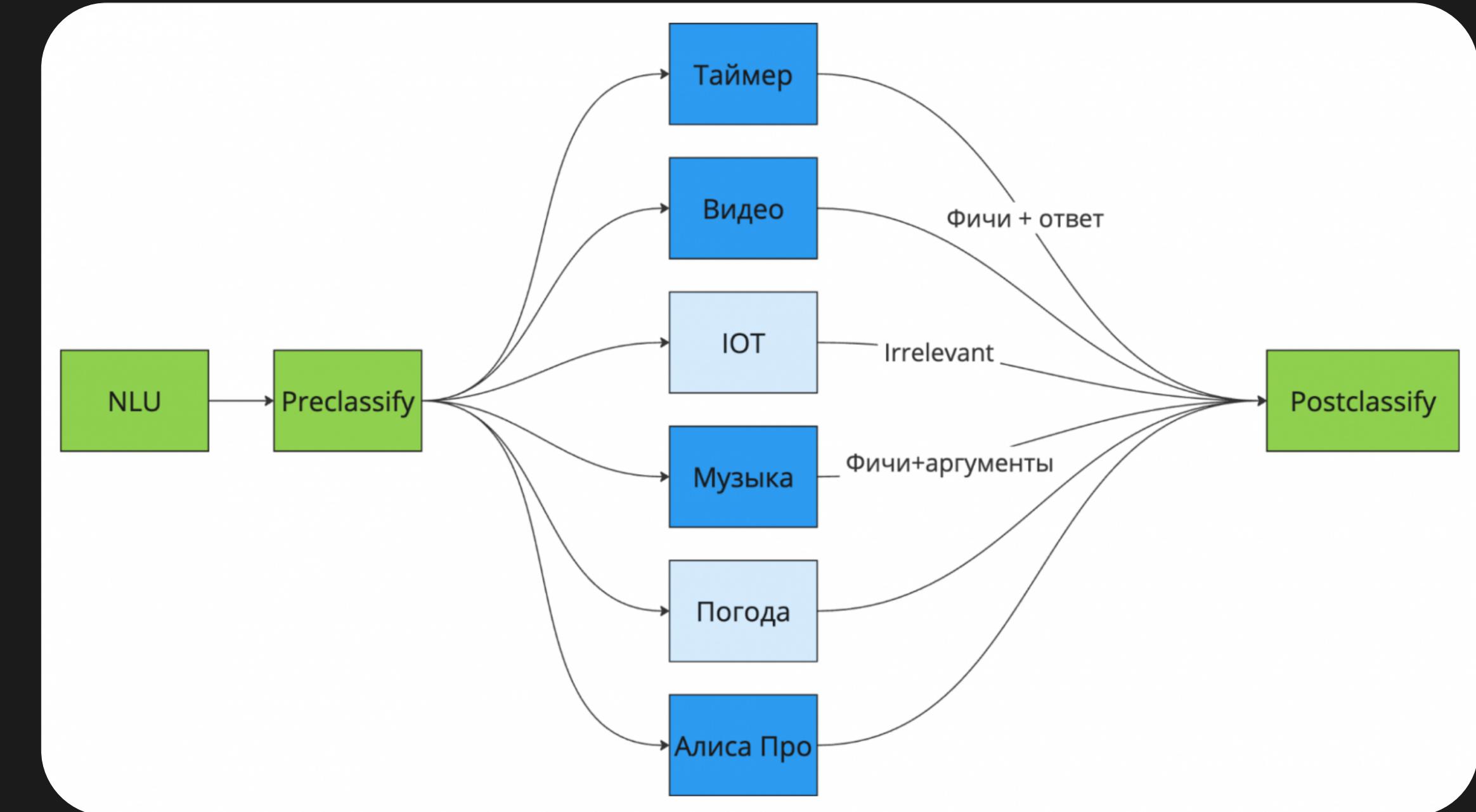
Разберем на примере Алисы!



Пойду издалека: зачем это нужно

Почему бы не написать классификатор ответа LLM, чтобы он вызывал функции?

Разберем на примере Алисы!



✖ Проблема

Ограниченно разнообразие функциональности:

«Алиса, выключи телевизор, когда закончится этот матч»

Как сделать Алису умнее?

Смарт-функции:
как LLM помогает понять,
чего хочет пользователь

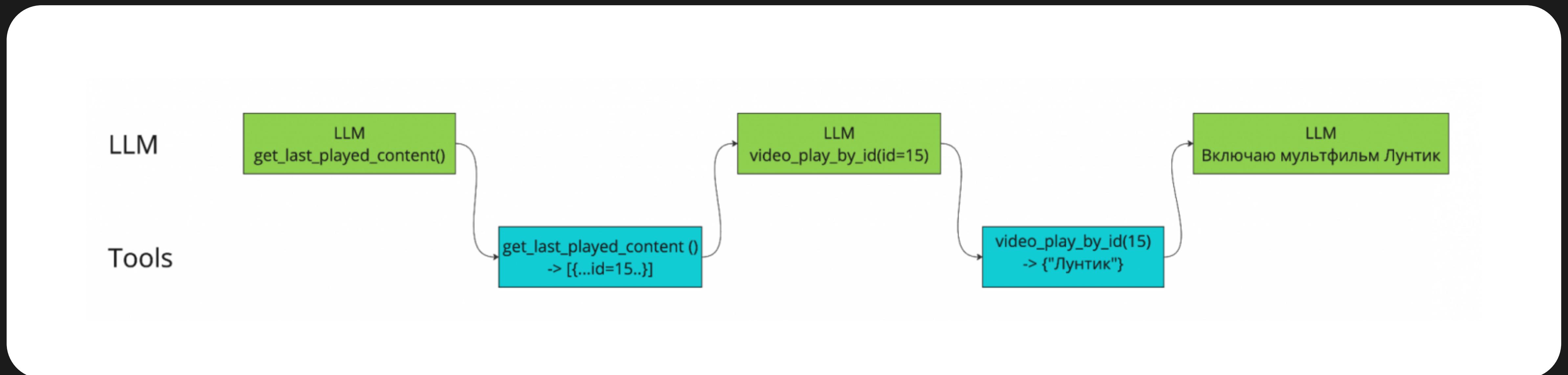
Смарт-функции

- LLM решает, какие ей вызывать инструменты
- Инструменты – (как пример) функциональности умного дома

Смарт-функции

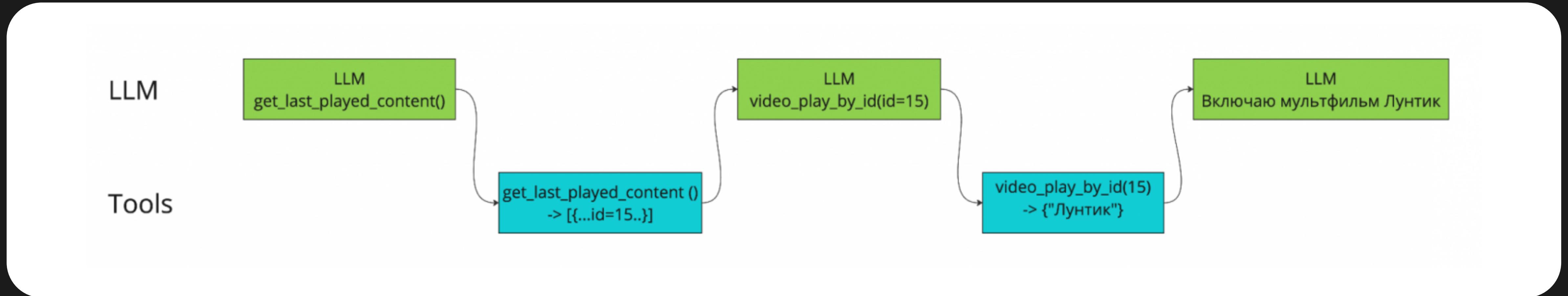
- LLM решает, какие ей вызывать инструменты
- Инструменты — (как пример) функциональности умного дома

«Включи мультфильм, который мы вчера смотрели»



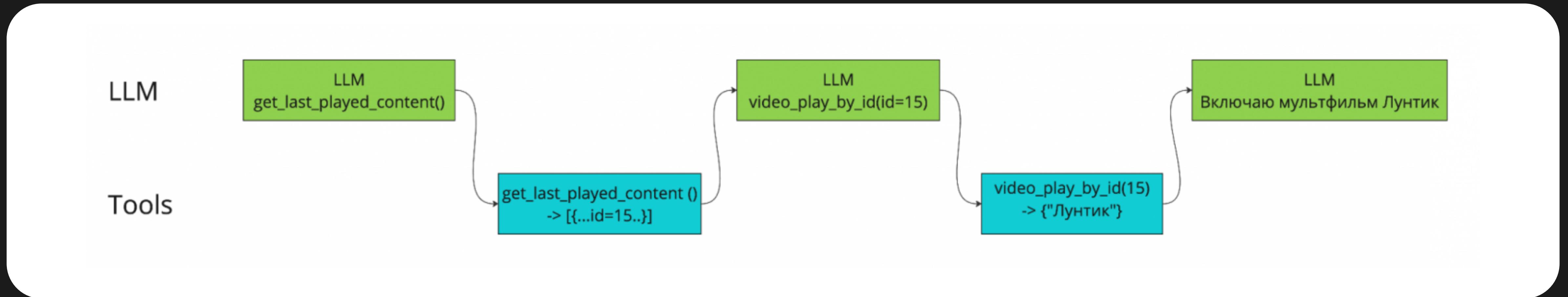
Смарт-функции

«Включи мультфильм, который мы вчера смотрели»



Смарт-функции

«Включи мультфильм, который мы вчера смотрели»



[AVAILABLE_TOOL]

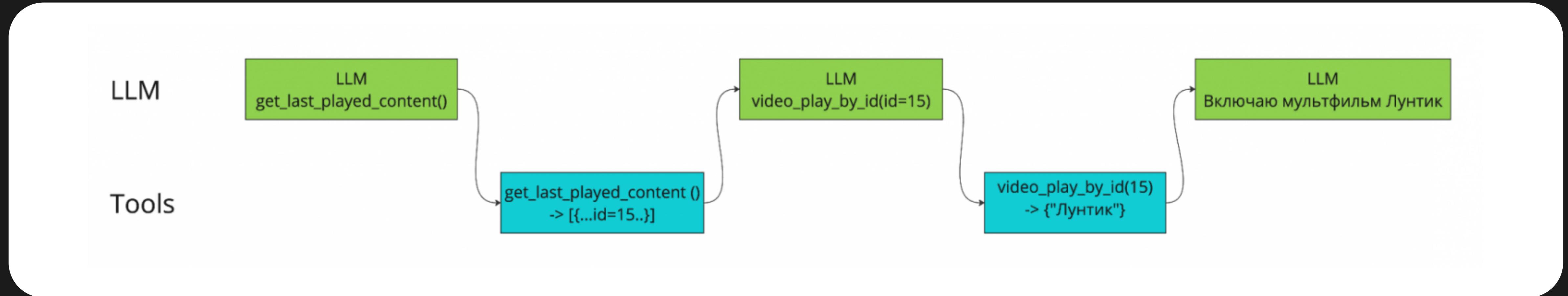
name: get_last_played_content
description: Получение информации о последних прослушанных аудиотреках, проигранных видео и телеканалах
parameters: {"properties": {"content_type": {"description": "Тип контента, последнее воспроизведение которого хочется получить..."}}}
return_parameters: {"properties": {"last_played_items": {"description": "Воспроизводимый ранее контент..."}}}
[/AVAILABLE_TOOL]

[AVAILABLE_TOOL]

name: video_play_by_id
description: Включение видео контента по Id
parameters: {"properties": {"id": {"description": "Id видео контента", "type": "string"}}}
return_parameters: {"properties": {"episode": {"description": "Серия которую включили..."}}}
[/AVAILABLE_TOOL]

Смарт-функции

«Включи мультфильм, который мы вчера смотрели»



...

[DEVICE]

Текущая дата - 2024-12-09. Текущее время - 12:45:00. День недели -
Понедельник

Устройство — Яндекс ТВ Станция с Алисой

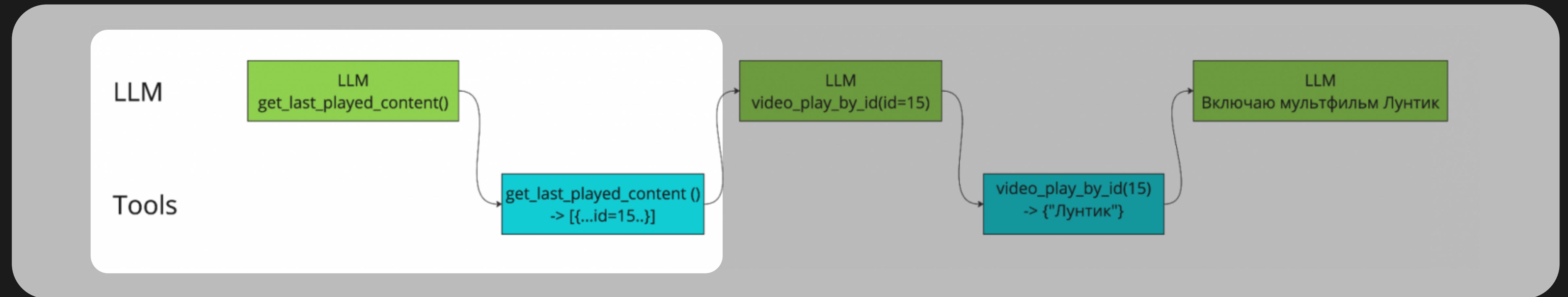
На экране отображается: Главный экран

[/DEVICE]

[INST]включи мультфильм который мы вчера смотрели[/INST]

Смарт-функции

«Включи мультфильм, который мы вчера смотрели»



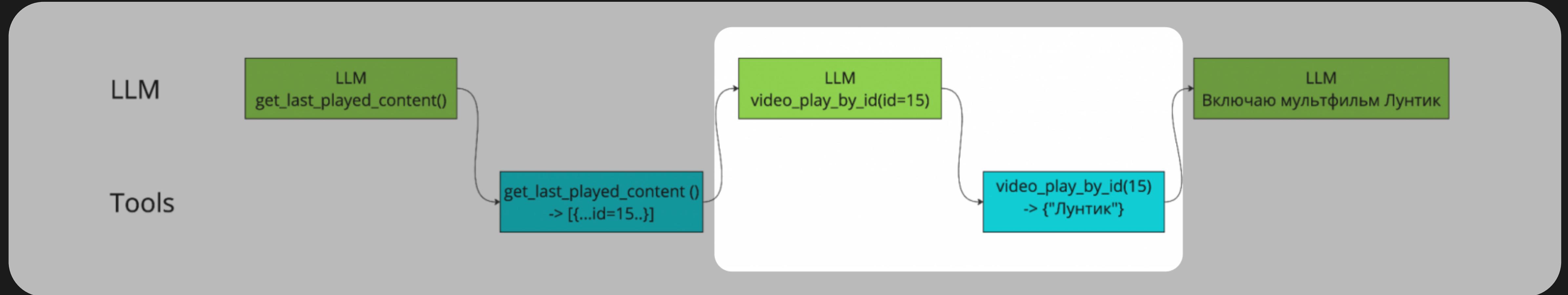
...

[REPLY][TOOL_CALL]get_last_played_content
{"content_type":"video"}
[/REPLY]

[TOOL_RESULT]get_last_played_content
{"last_played_items": [
 {"audio_content_id":"12"},
 {"video_content_id":"15"}
]}
[/TOOL_RESULT]

Смарт-функции

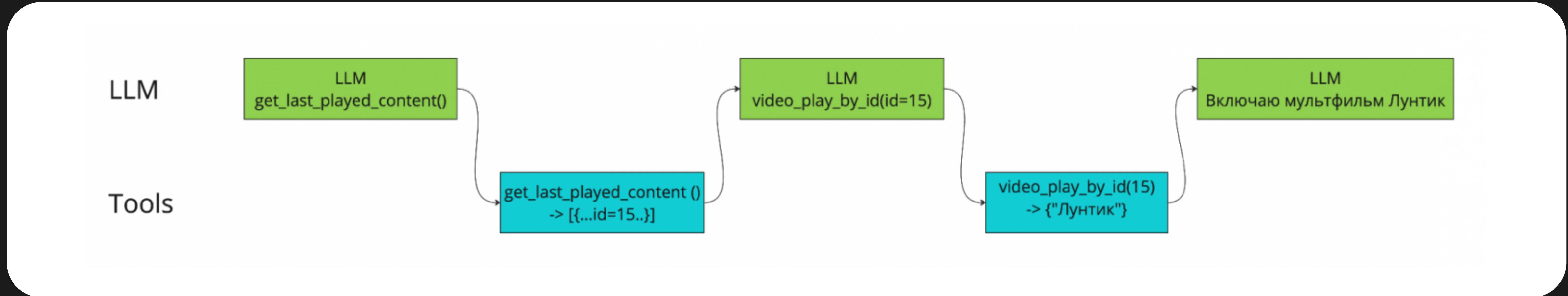
«Включи мультфильм, который мы вчера смотрели»



- ✓ LLM решает задачу итеративно, постепенно увеличивая свой промт

Смарт-функции

«Включи мультфильм, который мы вчера смотрели»



ⓧ Трудности

- Выбор релевантных инструментов – RAG (выбор топ-N релевантный)
- Правильный вызов инструментов – борьба с галлюцинациями
- Добавление новых инструментов

04

Кто такие LLM-Agents

Alfred The Agent



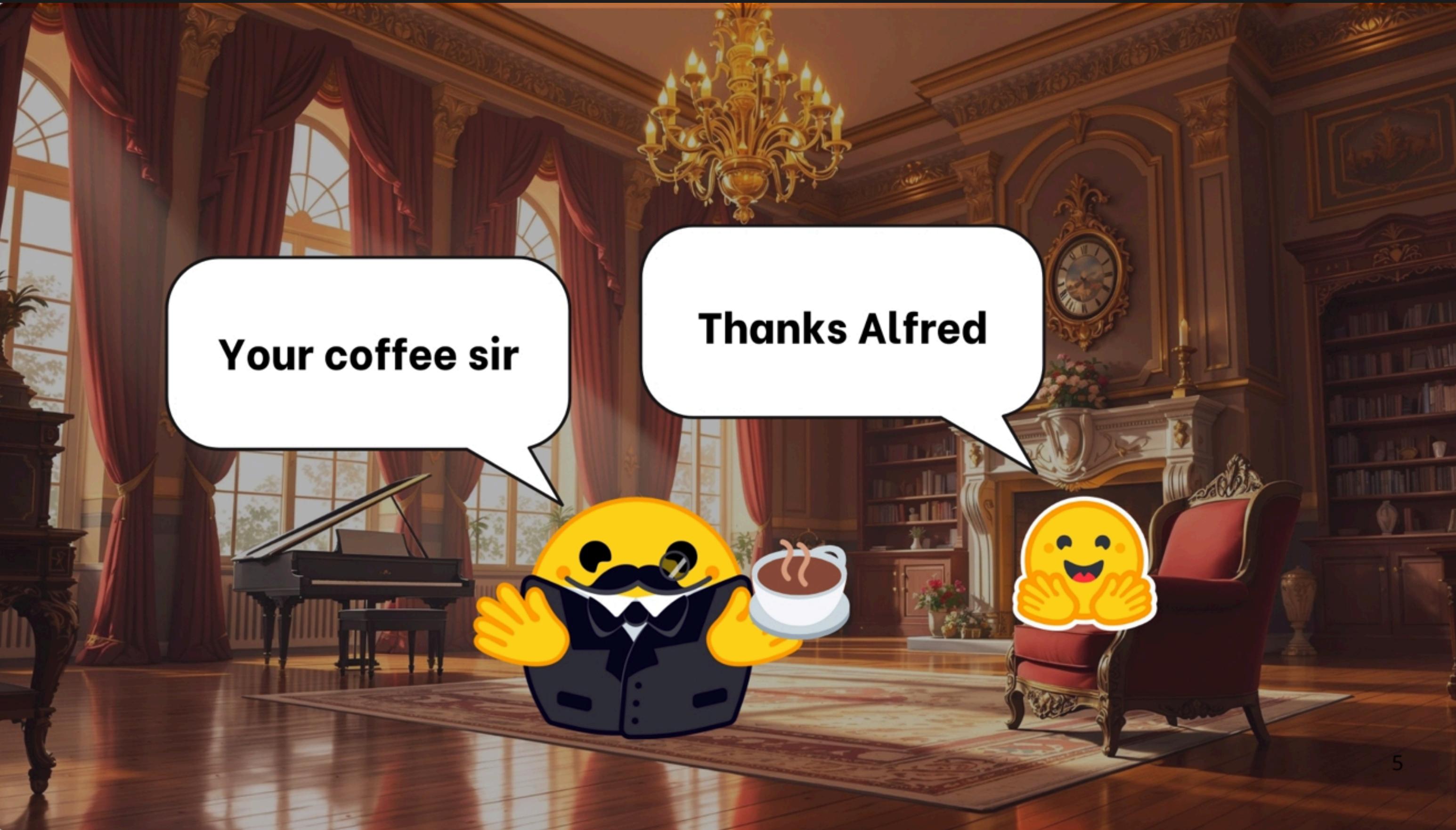
Alfred The Agent



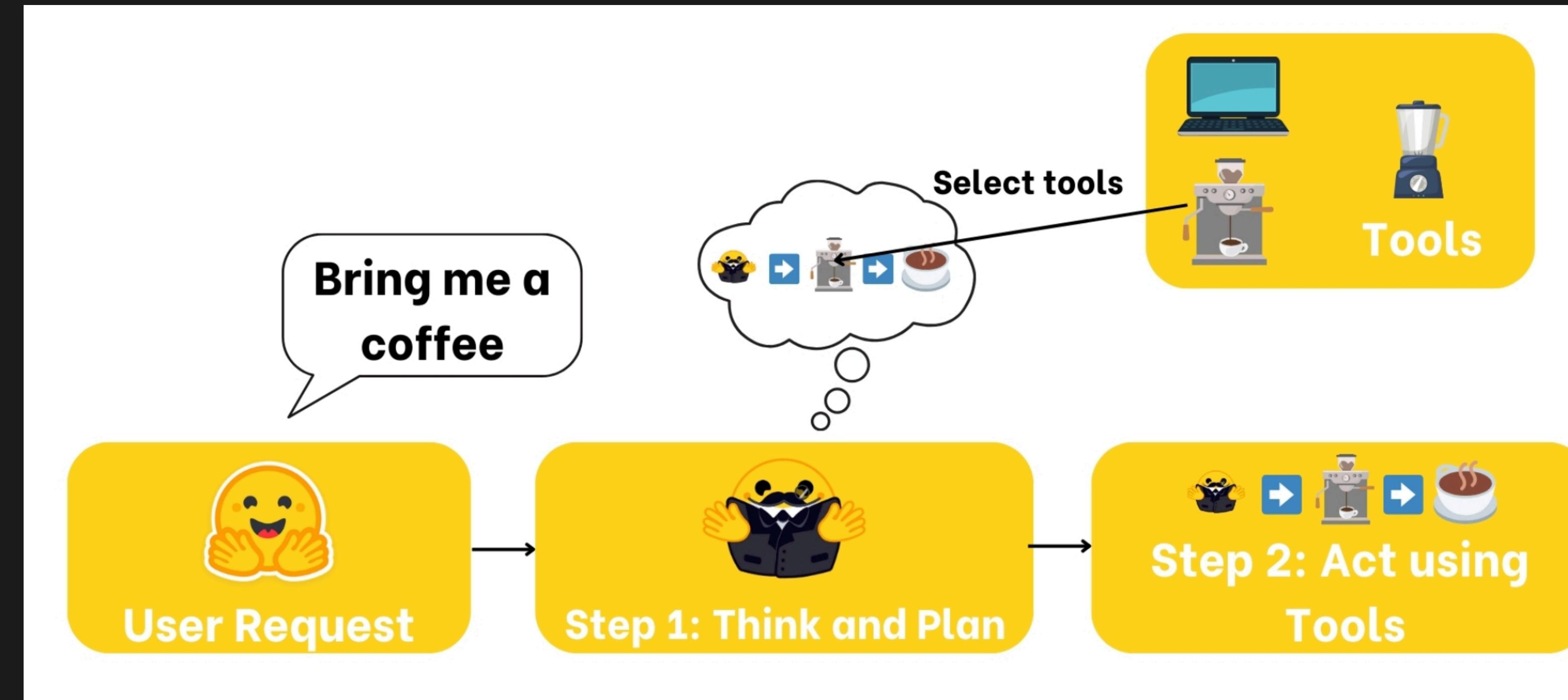
Alfred The Agent



Alfred The Agent



Alfred The Agent



Кто такие LLM-Агенты

Кто такие LLM-Агенты

01

Общение с пользователем — discovery

Кто такие LLM-Агенты

01

Общение с пользователем — discovery

02

Исследует, саморефлексирует — reasoning

Кто такие LLM-Агенты

- 01 Общение с пользователем — discovery
- 02 Исследует, саморефлексирует — reasoning
- 03 Совершает действия — инструменты

Виды Агентов

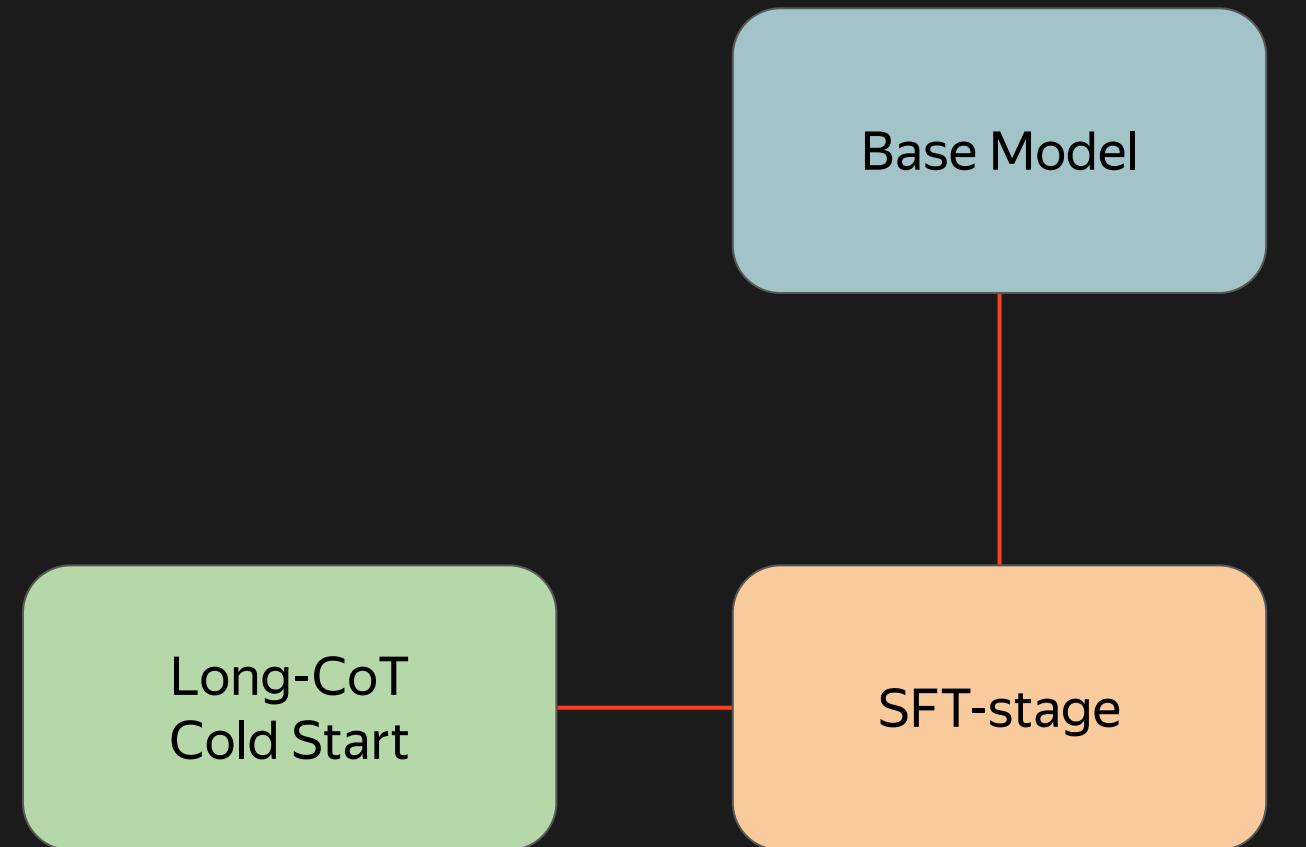
Agency Level	Description	What that's called	Example pattern
★★★	Agent output has no impact on program flow	Simple processor	<code>process_llm_output(llm_response)</code>
★★★	Agent output determines basic control flow	Router	<code>if llm_decision(): path_a() else: path_b()</code>
★★★	Agent output determines function execution	Tool caller	<code>run_function(llm_chosen_tool, llm_chosen_args)</code>
★★★	Agent output controls iteration and program continuation	Multi-step Agent	<code>while llm_should_continue(): execute_next_step()</code>
★★★	One agentic workflow can start another agentic workflow	Multi-Agent	<code>if llm_trigger(): execute_agent()</code>

05

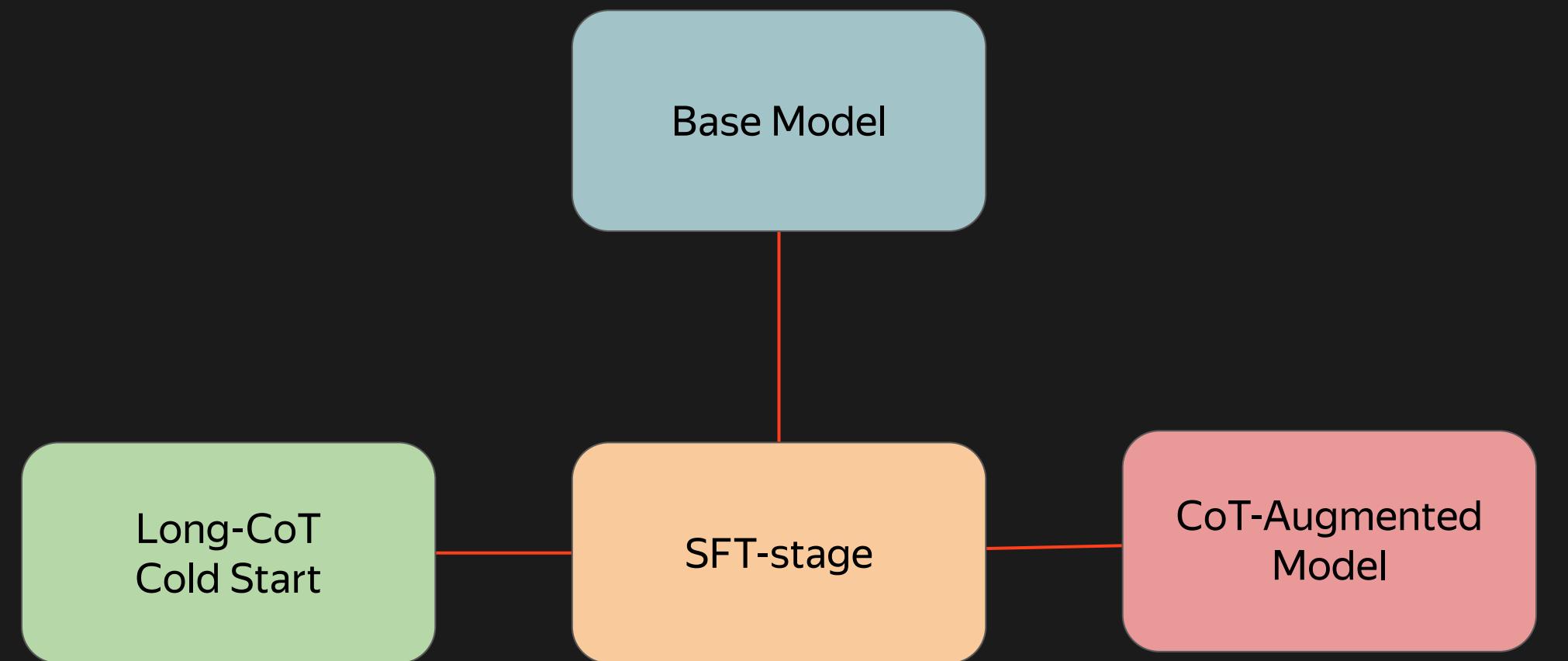
Как обучать?

Современные подходы в RL

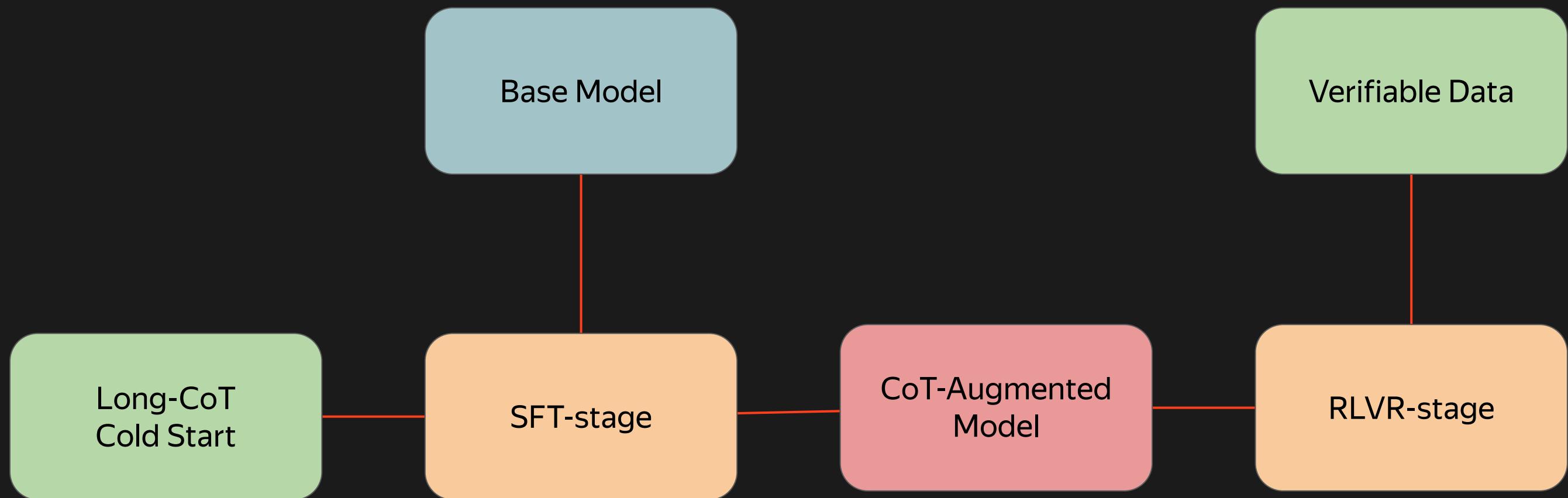
Как приготовить ризонера



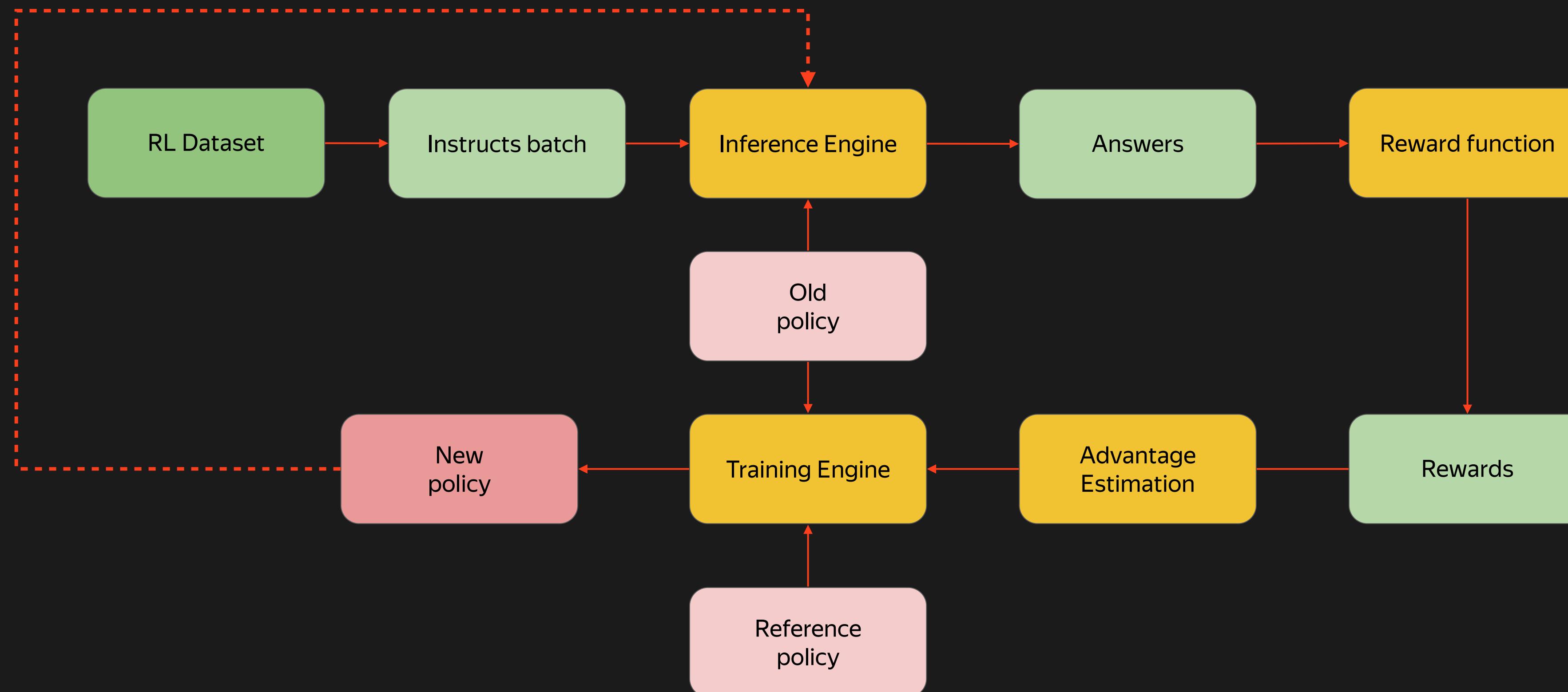
Как приготовить ризонера



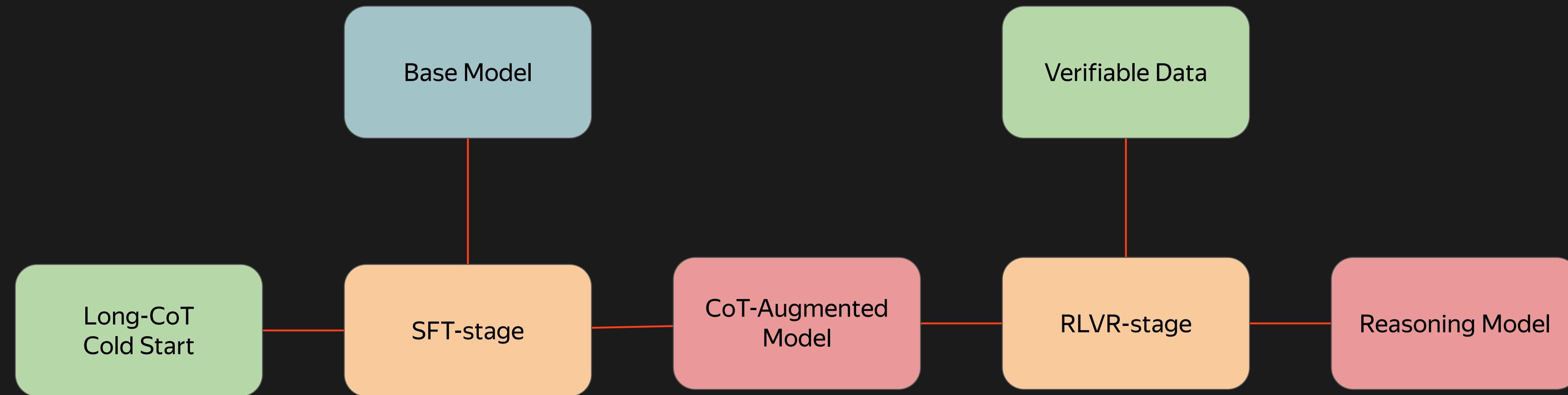
Как приготовить ризонера



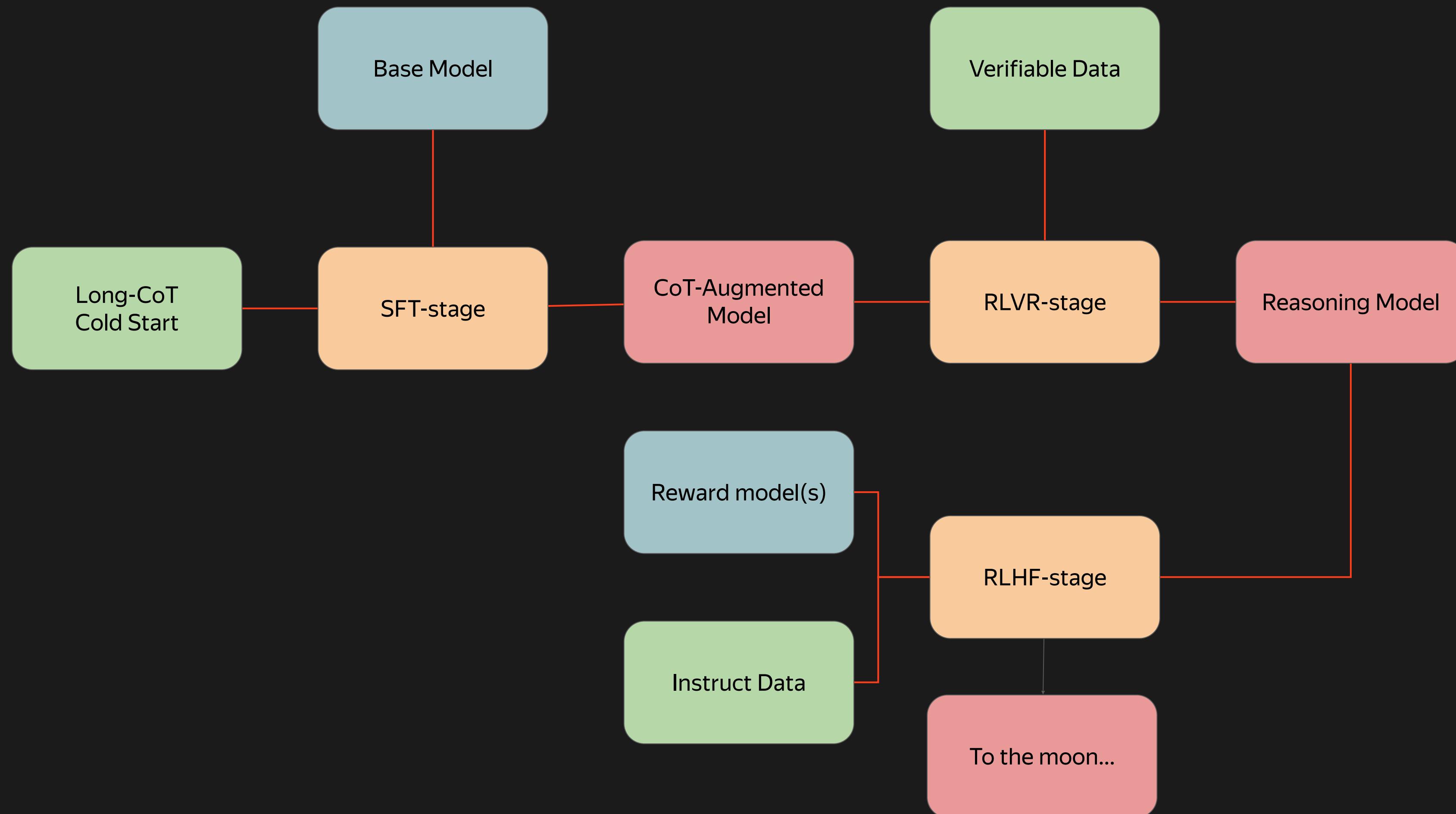
RLVR loop



Как приготовить ризонера



Как подготовить ризонера



Cold-start данные

Тематики задач:

- Математика
- Код
- Логика
- STEM: Science, Technology, Engineering, Mathematics
- Любые hard-to-answer & easy-to-verify

Фильтрация задач:

- Простые задачи
- Неверифицируемые задачи
- Задачи с потенциалом хака реворда

Источники CoT:

- CoT prompting
- Human-in-the-loop написания данных через цепочку рассуждений
- Синтетические данные от сильных моделей

Фильтрация ответов:

- За N генераций не пришли к правильному ответу
- Рассуждения повторяют ответ
- Рассуждения расходятся с ответом

RL данные

Тематики задач:

- Аналогично Cold-start

Фильтрация задач:

- Простые задачи
- Сложные задачи
- Те, что были на Cold-start

Функции наград:

- Rule-based
(проверка конечного состояния среды)

- LLM-as-judge
(для задач с однозначным/ограниченным числом ответов)
- Rubric-based
(clarity, correctness, safety)

RL Подходы

RL Подходы

Offline policy

- “Статичный” датасет
- Без доступа к окружению
- Обучение — оптимизация датасета
- Плюсы: простота и экономия
- Минусы: ограниченность датасетом

RL Подходы

Offline policy

- “Статичный” датасет
- Без доступа к окружению
- Обучение — оптимизация датасета
- Плюсы: простота и экономия
- Минусы: ограниченность датасетом

Online policy

- Сбор данных через взаимодействие с окружением
- Данные — (состояние среды, действие, награда, следующая среда)
- Плюсы: exploration
- Минусы: дорого, возможны галлюцинации среды

RL Подходы

Виды online данных

Single-turn

- 1 задача
- 1 СоТ
- Награда в конце СоТ

Multi-turn

- 1 задача
- Наличие Iron User — внешняя LLM, с которой общается обучаемый агент, узнает новую информацию
- Длинная траектория из общений
- Множество СоТ
- Награда на всю траекторию

RL подходы

01 DPO

02 PPO

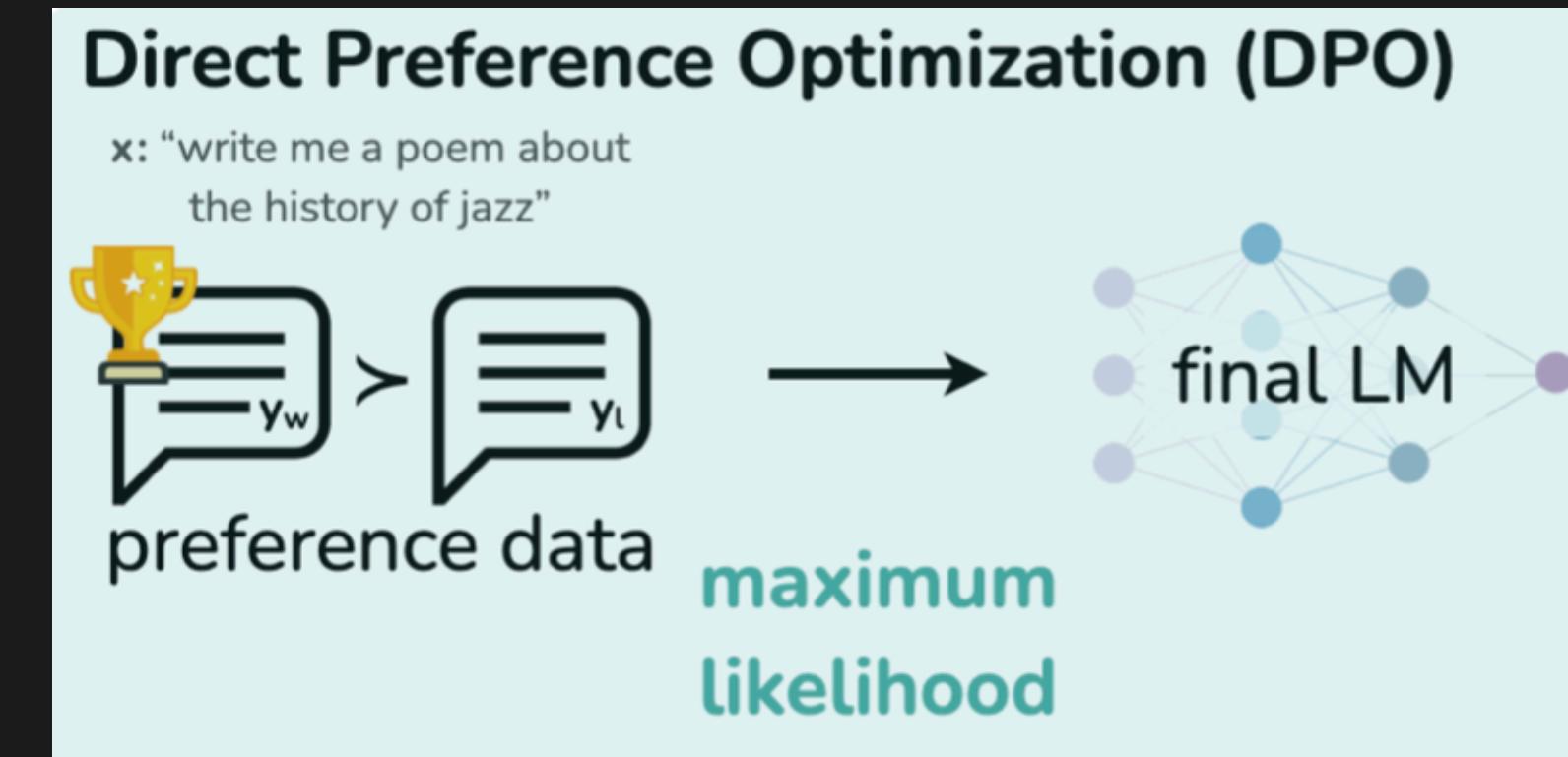
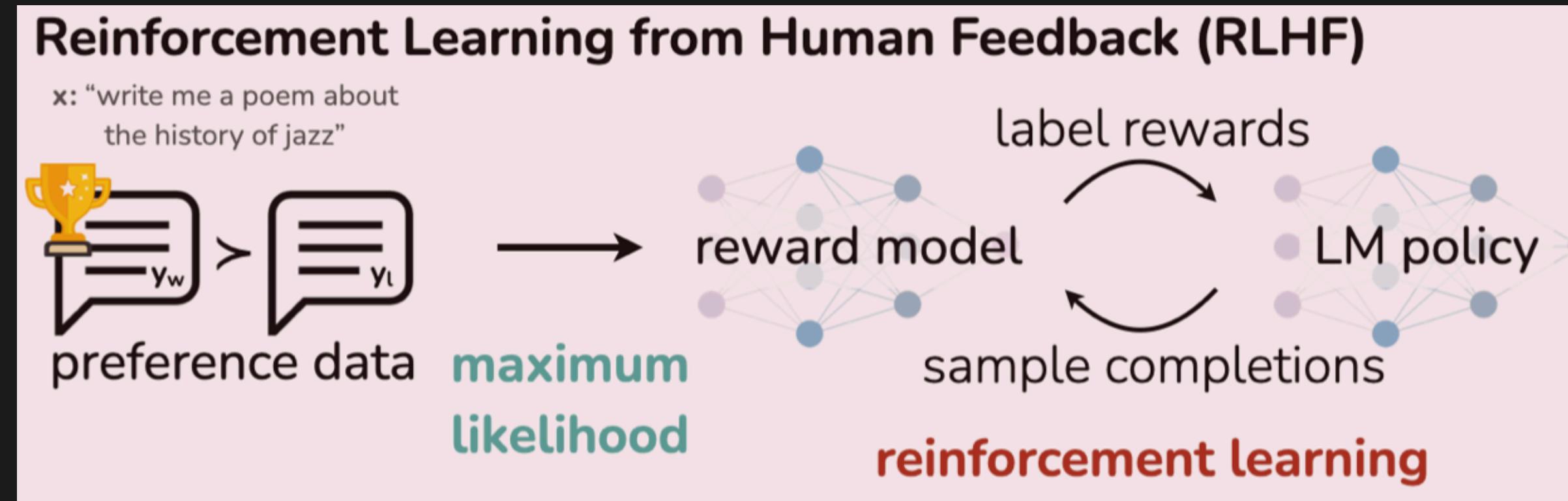
03 GRPO

04 DAPO

05 GRESO

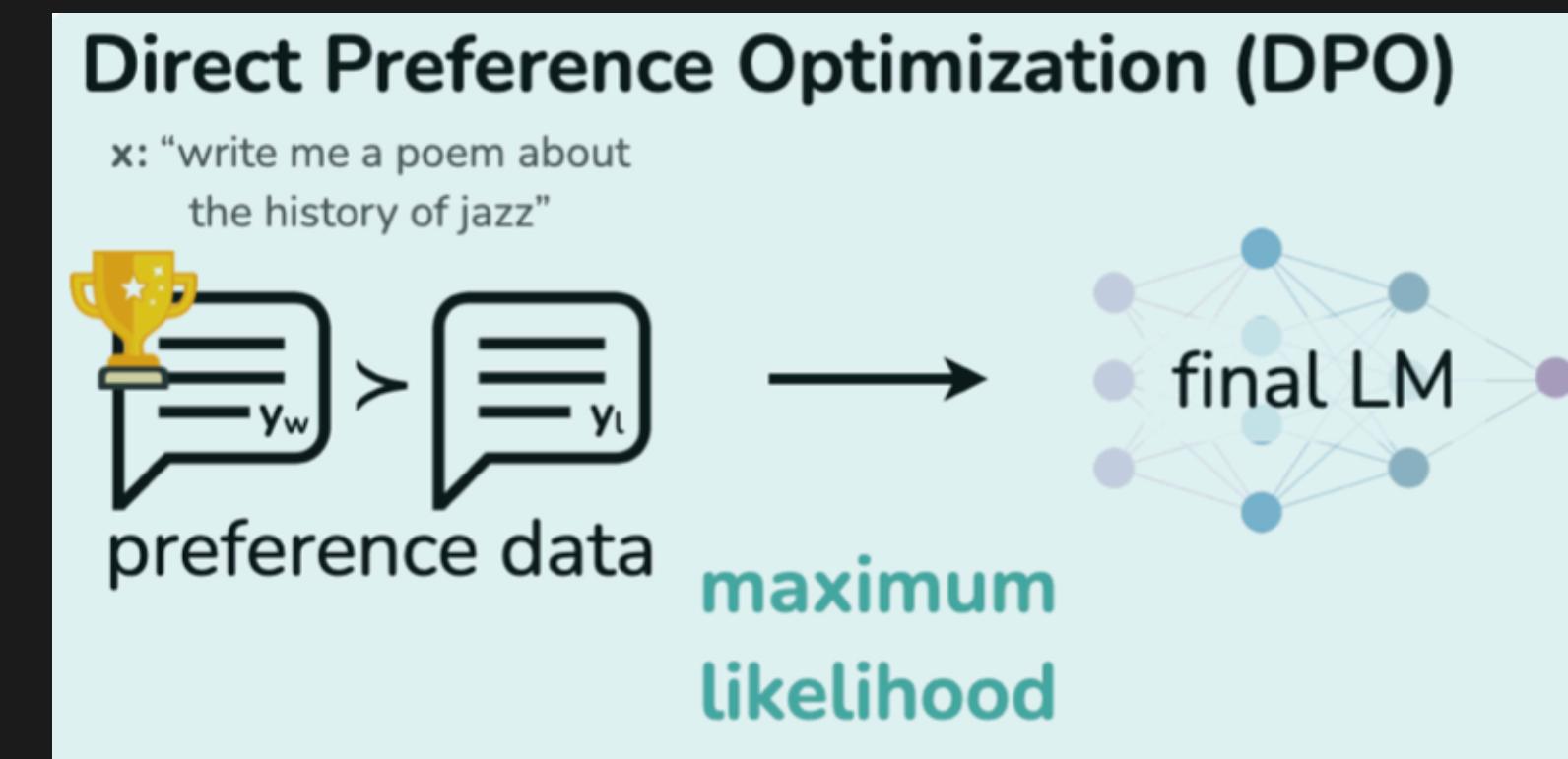
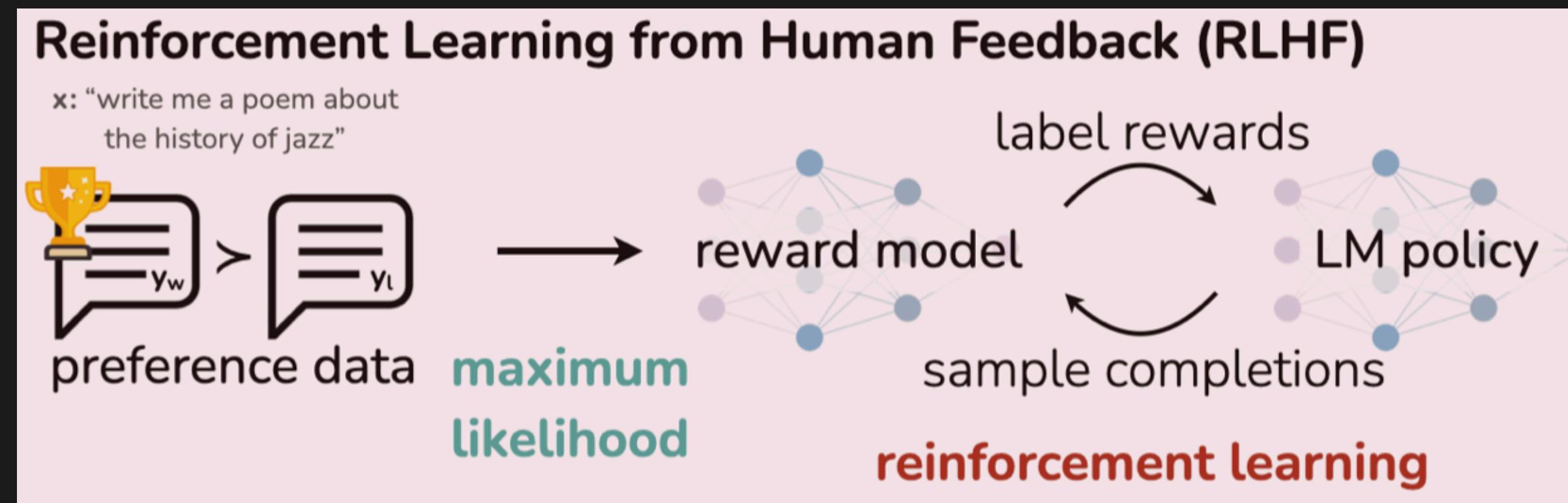
DPO

Direct Preference Optimization: RL для бедных



DPO

Direct Preference Optimization: RL для бедных

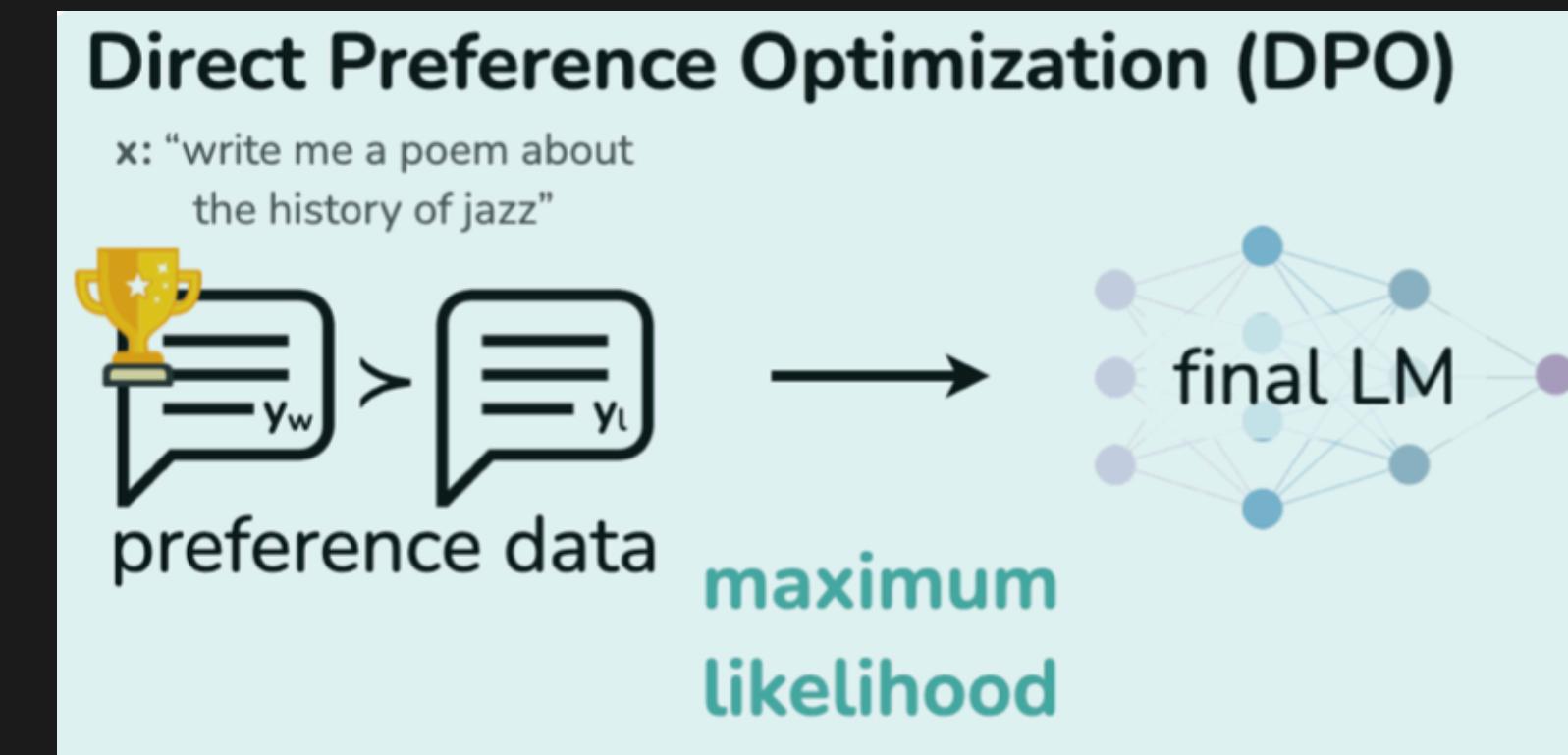
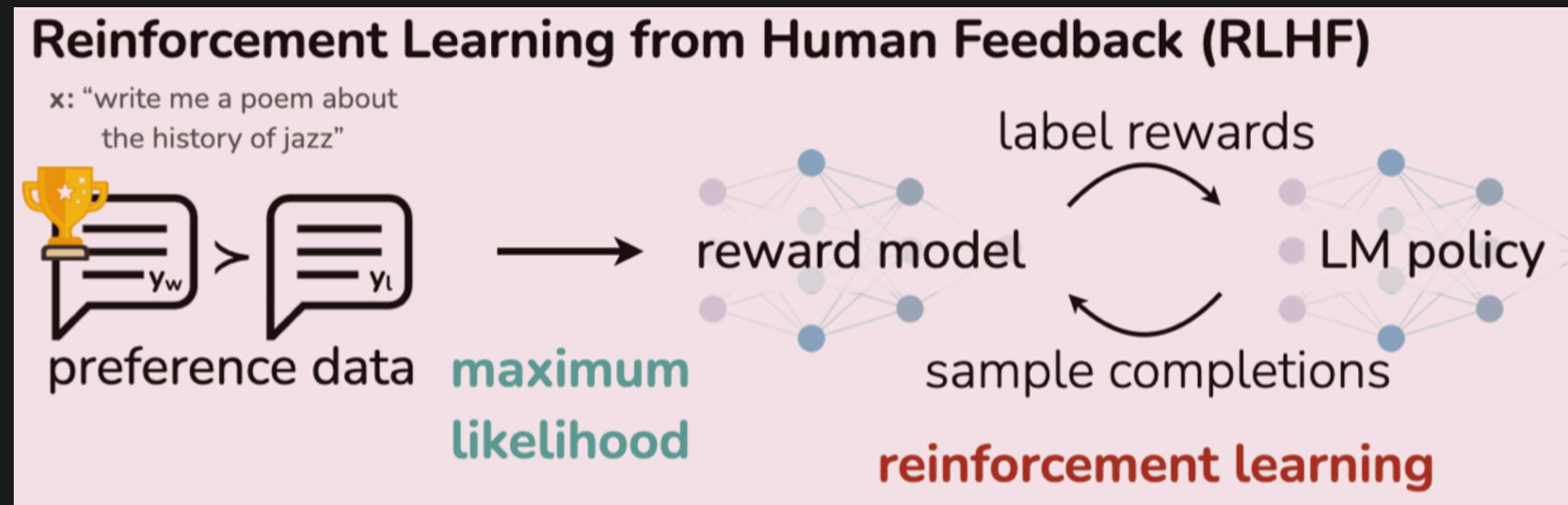


Плюсы:

- RL для бедных
- Требует только один собранный датасет
- Быстро обучается

DPO

Direct Preference Optimization: RL для бедных



Плюсы:

- RL для бедных
- Требует только один собранный датасет
- Быстро обучается

Минусы:

- Ограниченностъ данных
- Нет "exploration"
- Может улучшать Out-Of-Domain

RL подходы

01 DPO – Direct Preference Optimization

02 PPO

03 GRPO

04 DAPO

05 GRESO

PPO

Proximal Preference Optimization: самый простой online-RL

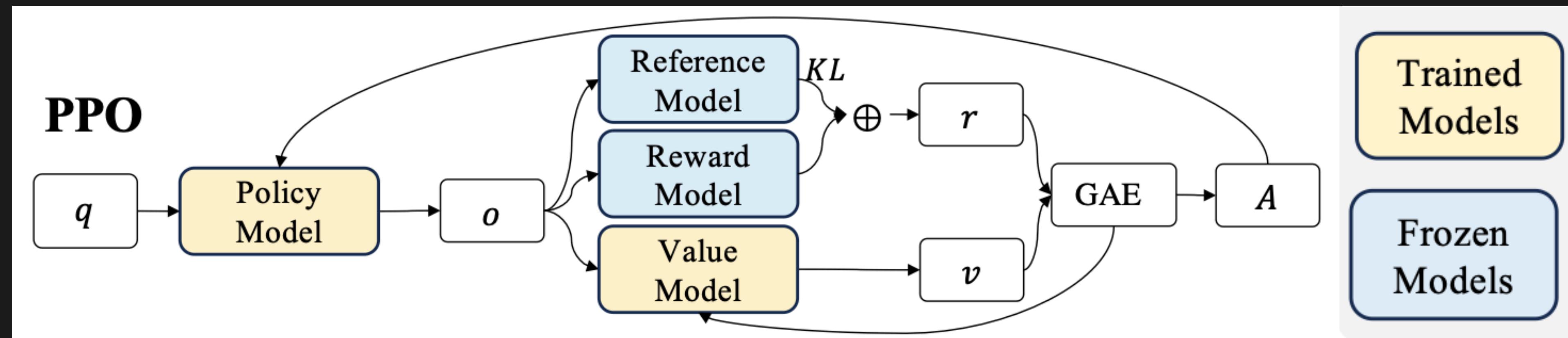
Proximal Policy Optimization (PPO) (Schulman et al., 2017) is an actor-critic RL algorithm that is widely used in the RL fine-tuning stage of LLMs (Ouyang et al., 2022). In particular, it optimizes LLMs by maximizing the following surrogate objective:

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right], \quad (1)$$

where π_θ and $\pi_{\theta_{old}}$ are the current and old policy models, and q, o are questions and outputs sampled from the question dataset and the old policy $\pi_{\theta_{old}}$, respectively. ε is a clipping-related hyper-parameter introduced in PPO for stabilizing training. A_t is the advantage, which is computed by applying Generalized Advantage Estimation (GAE) (Schulman et al., 2015), based

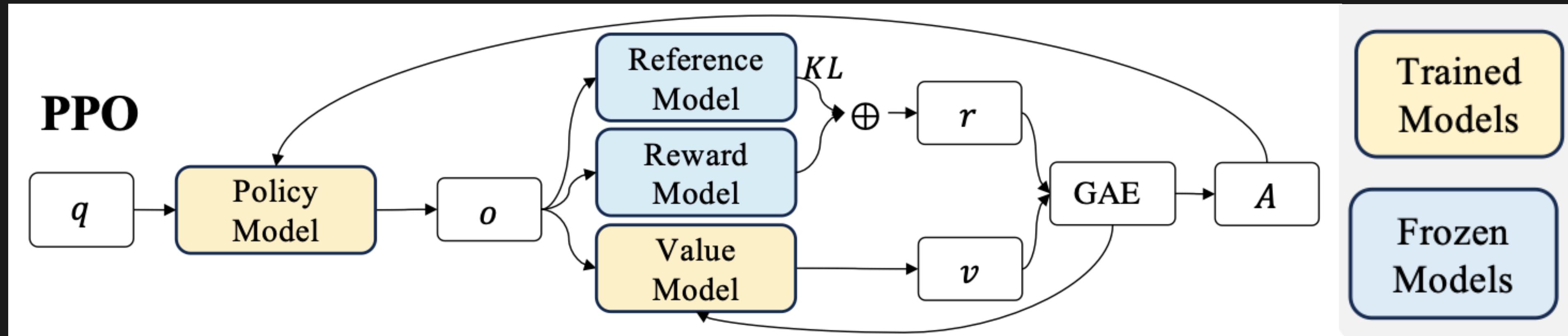
PPO

Proximal Preference Optimization: самый простой online-RL



PPO

Proximal Preference Optimization: самый простой online-RL



Минусы:

- Value Model, замедляющая обучение
- Абсолютные значения reward — могут быть нестабильные обучения
- Оценивает только один output, не знает про вариативность

RL подходы

01 DPO – Direct Preference Optimization

02 PPO – Proximal Preference Optimization

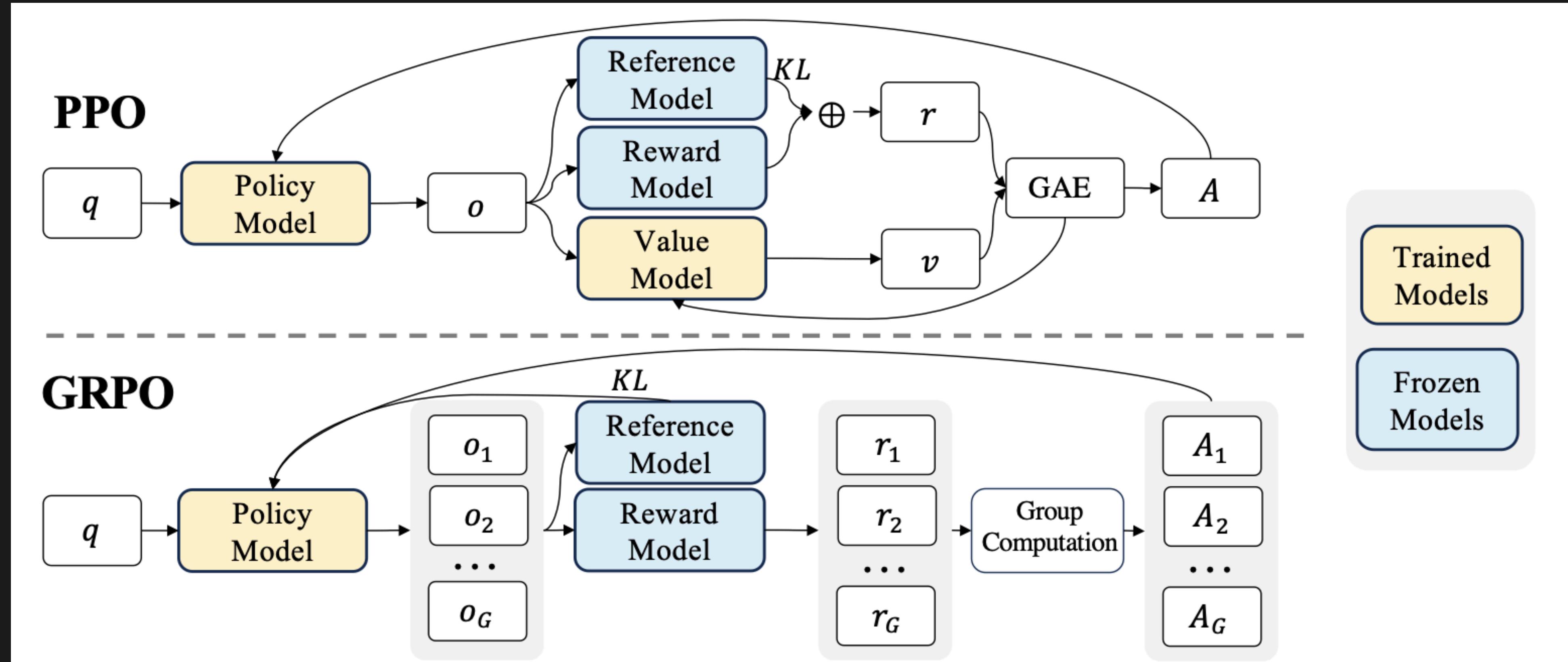
03 GRPO

04 DAPO

05 GRESO

GRPO

Group Relative Preference Optimization (DeepSeek — Apr 2024)



GRPO

Group Relative Preference Optimization (DeepSeek — Apr 2024)

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right]$$

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)]$$
$$\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL} [\pi_\theta || \pi_{ref}] \right\}$$

GRPO

Group Relative Preference Optimization (DeepSeek — Apr 2024)

$$\begin{aligned}\mathcal{J}_{GRPO}(\theta) = & \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)] \\ & \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL} [\pi_\theta || \pi_{ref}] \right\}\end{aligned}$$

Уже хорошо

GRPO

Group Relative Preference Optimization (DeepSeek — Apr 2024)

$$\begin{aligned}\mathcal{J}_{GRPO}(\theta) = & \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)] \\ & \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL} [\pi_\theta || \pi_{ref}] \right\}\end{aligned}$$

Уже хорошо, но можно лучше

GRPO

Group Relative Preference Optimization (DeepSeek — Apr 2024)

$$\begin{aligned}\mathcal{J}_{GRPO}(\theta) = & \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)] \\ & \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL} [\pi_\theta || \pi_{ref}] \right\}\end{aligned}$$

Уже хорошо, но можно лучше. Проблемы:

1. Коллапс энтропии модели из-за клиппинга exploration-токенов
2. Неэффективная утилизация батча при обучении ($\text{Adv.} = 0$)
3. Зависимость вклада токена в общий градиент от длины ответа
4. Что делать с парами промпт+ризонинг > макс длина

RL подходы

01 DPO – Direct Preference Optimization

02 PPO – Proximal Preference Optimization

03 GRPO – Group Relative Preference Optimization

04 DAPO

05 GRESO

DAPO

Dynamic Sampling Policy Optimization (ByteDance — May 2025)

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) = & \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \\ & \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right) - \beta D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}}) \right) \right]\end{aligned}$$

Сейчас начнет свистеть фляга...

$$\begin{aligned}\mathcal{J}_{\text{DAPO}}(\theta) = & \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \\ & \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}} \right) \hat{A}_{i,t} \right) \right] \\ \text{s.t. } & 0 < \left| \{o_i \mid \text{is_equivalent}(a, o_i)\} \right| < G,\end{aligned}$$

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)}.$$

DAPO

Dynamic Sampling Policy Optimization (ByteDance — May 2025)

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q, a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | q)} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right) - \boxed{\beta D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}})} \right) \right]$$

Во время обучения длинного СоТ
распределение токенов сильно
меняется и это нормально

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}_{(q, a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | q)} \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}} \right) \hat{A}_{i,t} \right) \right]$$

s.t. $0 < \left| \{o_i \mid \text{is_equivalent}(a, o_i)\} \right| < G,$

Поэтому можем избавиться от KL

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)}.$$

DAPO

Dynamic Sampling Policy Optimization (ByteDance — May 2025)

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon) \hat{A}_{i,t}) - \beta D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}}) \right) \right]$$

$$\begin{aligned} \mathcal{J}_{\text{DAPO}}(\theta) &= \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1 - \boxed{\varepsilon_{\text{low}}}, 1 + \boxed{\varepsilon_{\text{high}}}) \hat{A}_{i,t}) \right] \\ \text{s.t. } &0 < \left| \{o_i \mid \text{is_equivalent}(a, o_i)\} \right| < G, \\ r_{i,t}(\theta) &= \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)}. \end{aligned}$$

Concretely, when $\varepsilon = 0.2$ (the default value of most algorithms) and $\hat{A}_{i,t} > 0$ (the system tries to increase the probability),

consider two actions with probabilities $\pi_{\theta_{\text{old}}}(o_i \mid q) = 0.01$ and 0.9 .

The upper bounds of the increased probabilities $\pi_\theta(o_i \mid q)$ are 0.012 and 1.08 , respectively ($\pi_{\theta_{\text{old}}} \cdot (1 + \varepsilon)$).

This implies that ‘exploitation’ tokens with a higher probability (e.g., 0.9) are not constrained to get even extremely larger probabilities like 0.999

DAPO

Dynamic Sampling Policy Optimization (ByteDance — May 2025)

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q, a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | q)} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right) - \beta D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}}) \right) \right]$$

В GRPO никак не влияет длина генерации ризонинга. В общем случае это выливается в увеличении длины генерации размышлений, что не всегда хорошо

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}_{(q, a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | q)} \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}} \right) \hat{A}_{i,t} \right) \right]$$

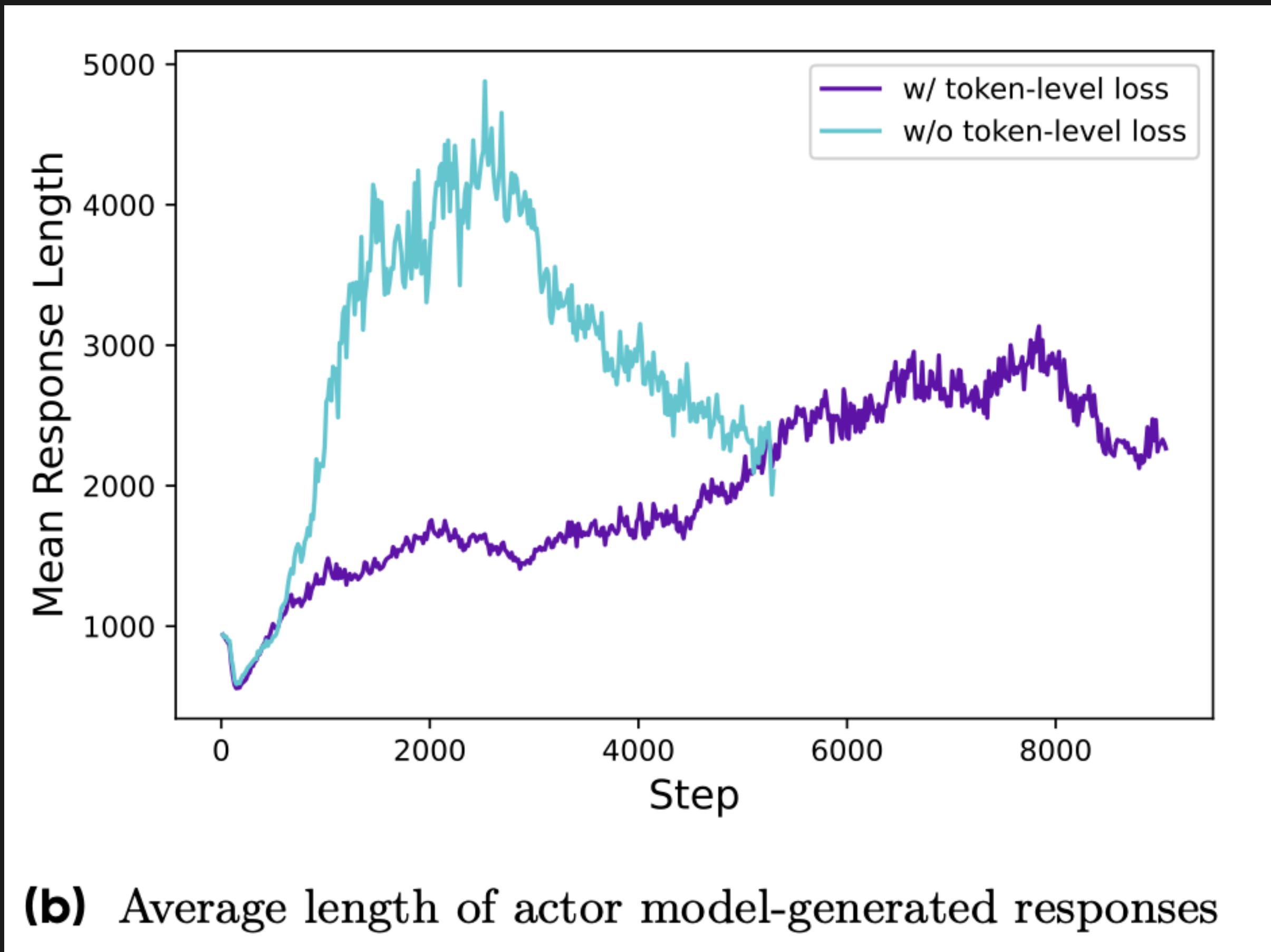
s.t. $0 < \left| \{o_i \mid \text{is_equivalent}(a, o_i)\} \right| < G,$

В DAPO у более длинных предложений будет меньший вес в результат

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)}.$$

DAPO

Dynamic Sampling Policy Optimization (ByteDance — May 2025)



В GRPO никак не влияет длина генерации ризонинга. В общем случае это выливается в увеличении длины генерации размышлений, что не всегда хорошо

В DAPO у более длинных предложений будет меньший вес в результат

DAPO

Dynamic Sampling Policy Optimization (ByteDance — May 2025)

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q, a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | q)} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right) - \beta D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}}) \right) \right]$$

$$\begin{aligned} \mathcal{J}_{\text{DAPO}}(\theta) = & \mathbb{E}_{(q, a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | q)} \\ & \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}} \right) \hat{A}_{i,t} \right) \right] \\ \text{s.t. } & 0 < \left| \{o_i \mid \text{is_equivalent}(a, o_i)\} \right| < G, \end{aligned}$$

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)}.$$

Нам не нужны такие семплы, где все output в группе равны: реворд относительный. Такие группы не будут влиять на обучение, но будут замедлять вычисления

Идея:
давайте фильтровать такие примеры

DAPO

Dynamic Sampling Policy Optimization (ByteDance — May 2025)

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) = & \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \\ & \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right) - \boxed{\beta D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}})} \right) \right]\end{aligned}$$

Сейчас это около стандарт обучения RL

$$\begin{aligned}\mathcal{J}_{\text{DAPO}}(\theta) = & \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \\ & \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \boxed{\varepsilon_{\text{high}}} \right) \hat{A}_{i,t} \right) \right] \\ \text{s.t. } & \boxed{0 < |\{o_i \mid \text{is_equivalent}(a, o_i)\}| < G},\end{aligned}$$

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)}.$$

DAPO

Dynamic Sampling Policy Optimization (ByteDance — May 2025)

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) = & \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \\ & \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right) - \boxed{\beta D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}})} \right) \right]\end{aligned}$$

Сейчас это около стандарт обучения RL
Но можно еще лучше....

$$\begin{aligned}\mathcal{J}_{\text{DAPO}}(\theta) = & \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \\ & \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \boxed{\varepsilon_{\text{high}}} \right) \hat{A}_{i,t} \right) \right] \\ \text{s.t. } & \boxed{0 < |\{o_i \mid \text{is_equivalent}(a, o_i)\}| < G},\end{aligned}$$

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)}.$$

RL подходы

01 DPO – Direct Preference Optimization

02 PPO – Proximal Preference Optimization

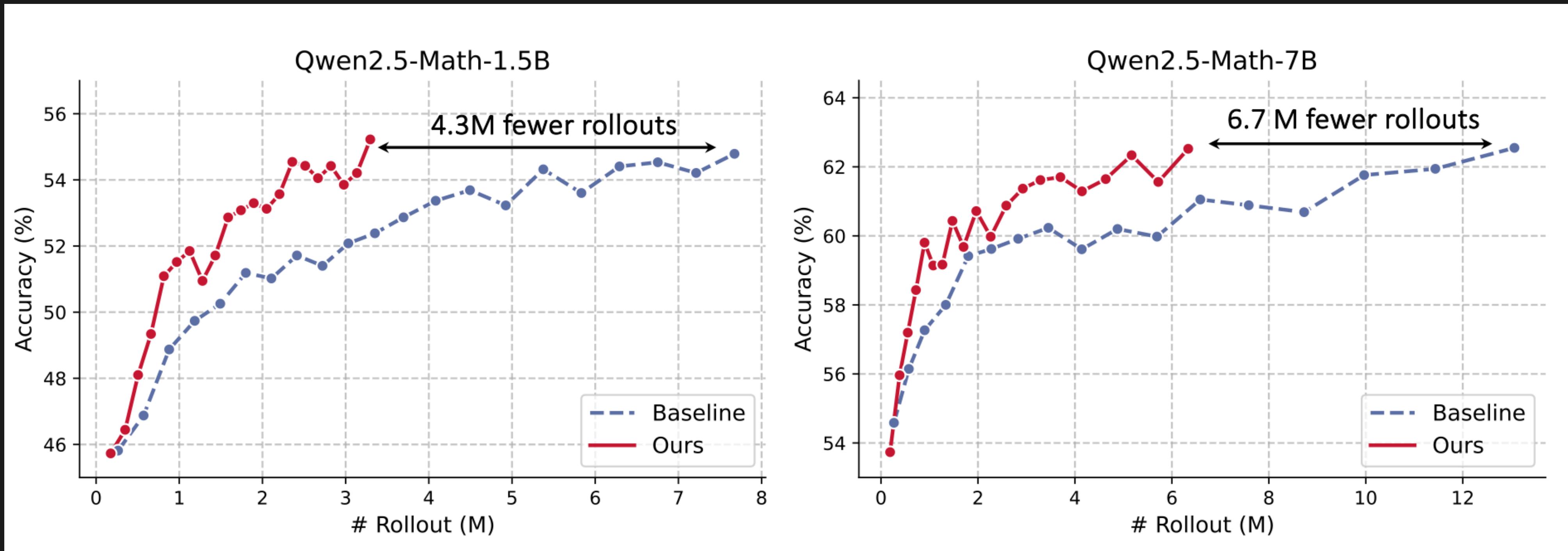
03 GRPO – Group Relative Preference Optimization

04 DAPO – Dynamic Sampling Policy Optimization

05 FINAL BOSS – GRESO

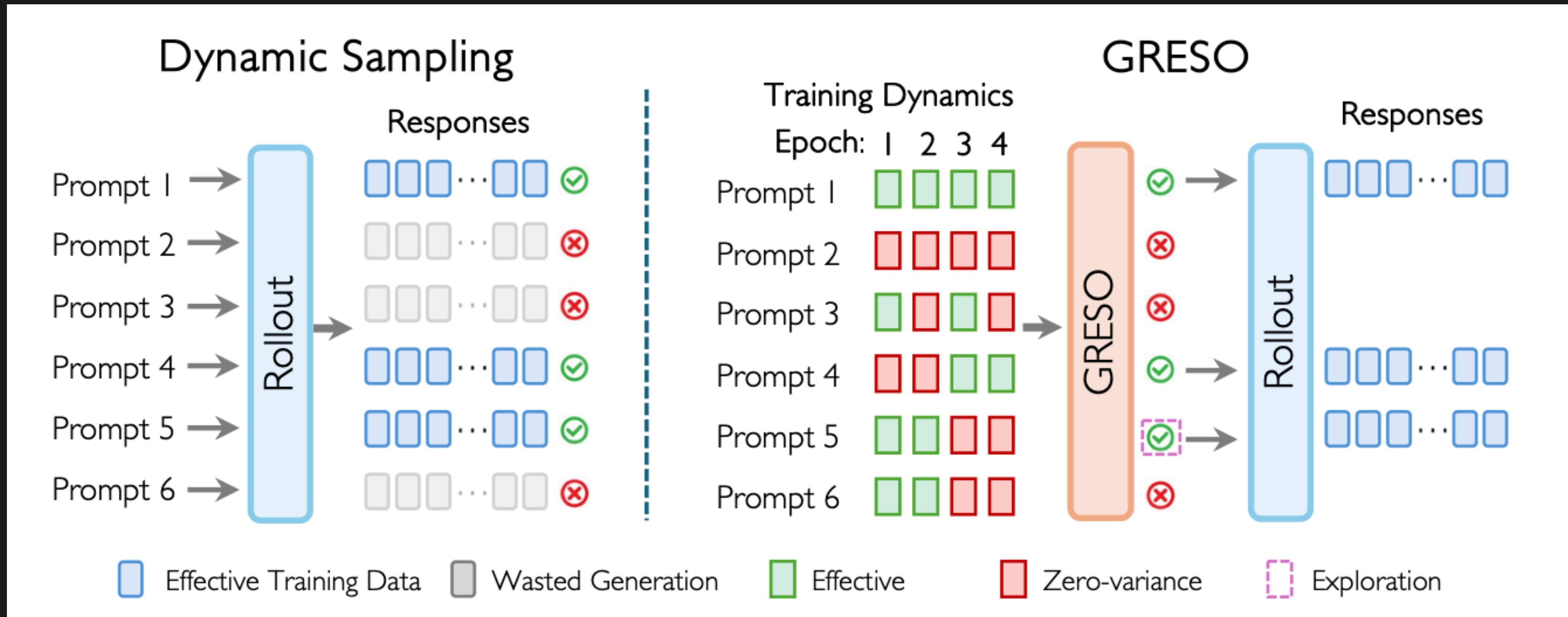
GRESO

GRPO with Effective Selective Rollout (Jun 2025)



GRESO

GRPO with Effective Selective Rollout (Jun 2025)



- При семплировании данных не берем те, на которых и так reward по прошлому обучению был 1 на всех output
- Однако оставляем часть простых/сложных данных для стабильности обучения

RL подходы

- 01 DPO – Direct Preference Optimization
- 02 PPO – Proximal Preference Optimization
- 03 GRPO – Group Relative Preference Optimization
- 04 DAPO – Dynamic Sampling Policy Optimization
- 05 GRESO – GRPO with Effective Selective Rollout – **SOTA RL**

VERL

Volcano Engine Reinforcement Learning for LLMs

01

SOTA фреймворк обучения RL

02

Активно развивается

03

github – <https://github.com/volcengine/verl>

04

Есть поддержка DAPO из коробки

05

Поддержка многошагового агентского обучения

06

Полезные статьи

Статьи, которые обсуждались в данном семинаре

Function calling

- 01** What is Function Calling? – <https://huggingface.co/learn/agents-course/bonus-unit1/what-is-function-calling>
- 02** Fine-tune + пример – <https://huggingface.co/learn/agents-course/bonus-unit1/fine-tuning>

RL Подходы

01 DPO – <https://arxiv.org/pdf/2305.18290>

02 PPO – <https://arxiv.org/pdf/1707.06347>

03 GRPO – <https://arxiv.org/pdf/2402.03300>

04 DAPO – <https://arxiv.org/pdf/2503.14476>

05 GRESO – <https://arxiv.org/pdf/2506.02177>

Ведущие модели

01 kimi k2 – https://github.com/MoonshotAI/Kimi-K2/blob/main/tech_report.pdf

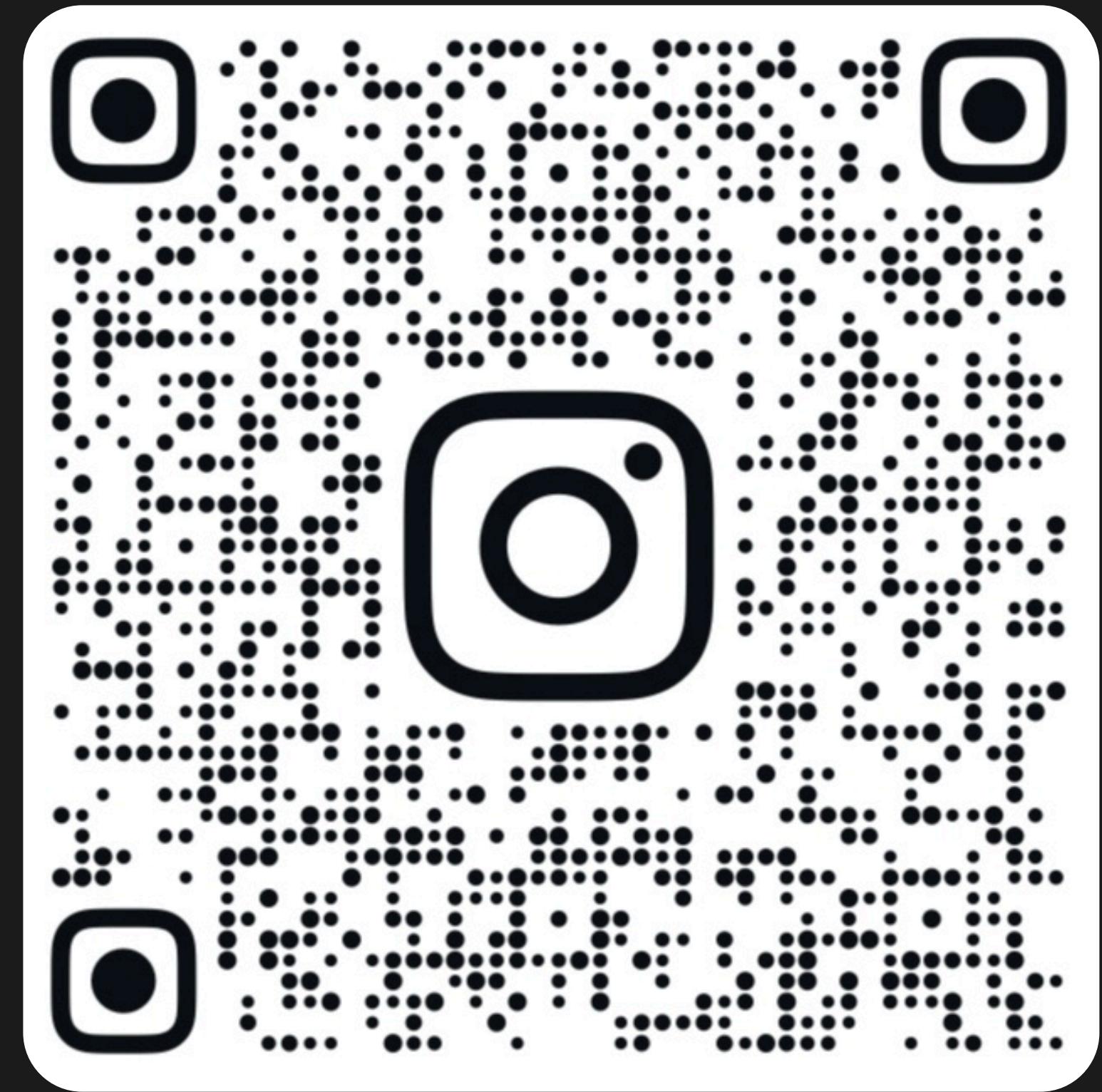
02 qwen 2.5 – <https://arxiv.org/pdf/2412.15115.pdf>

03 qwen 3 – <https://arxiv.org/pdf/2505.09388.pdf>

Контакты нанимаю



telegram



instagram*

*принадлежат компании Meta,
признанной экстремистской в России и
запрещённой