Weights & Biases

LLM Agent Fine-Tuning: Enhancing Task Automation

Anish Shah

Agenda

01 RAG: The current chatbot workflow

02 LLM Agents

03 Improving LLMs via Fine-tuning

04 Improving Agent Workflows via Fine-tuning

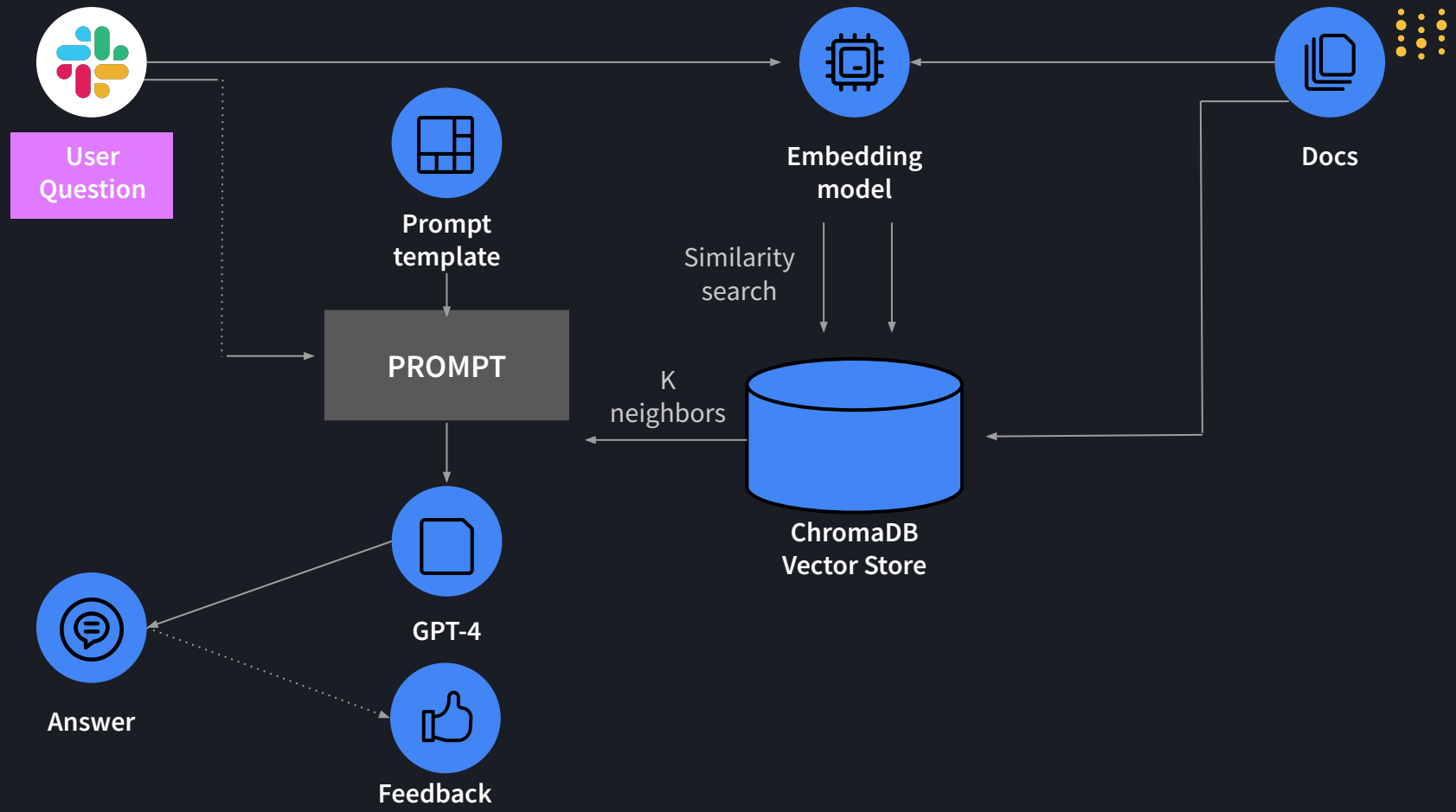
05 Demo: GPT3.5 and AgentInstruct

06 Demo: FireAct and Weights & Biases

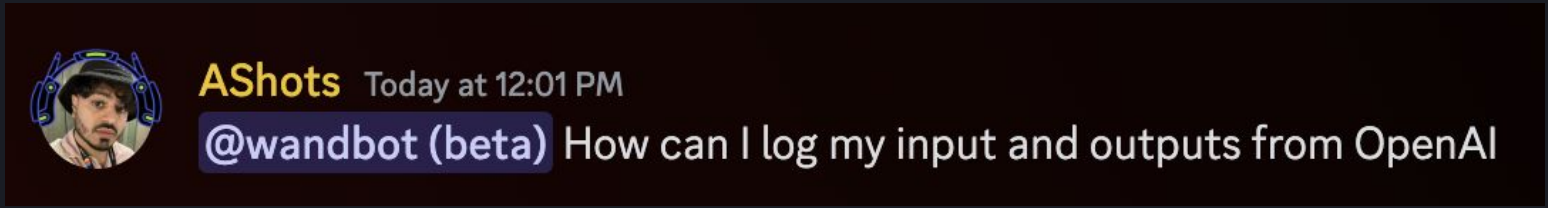
07 Standardized Agent Evaluation

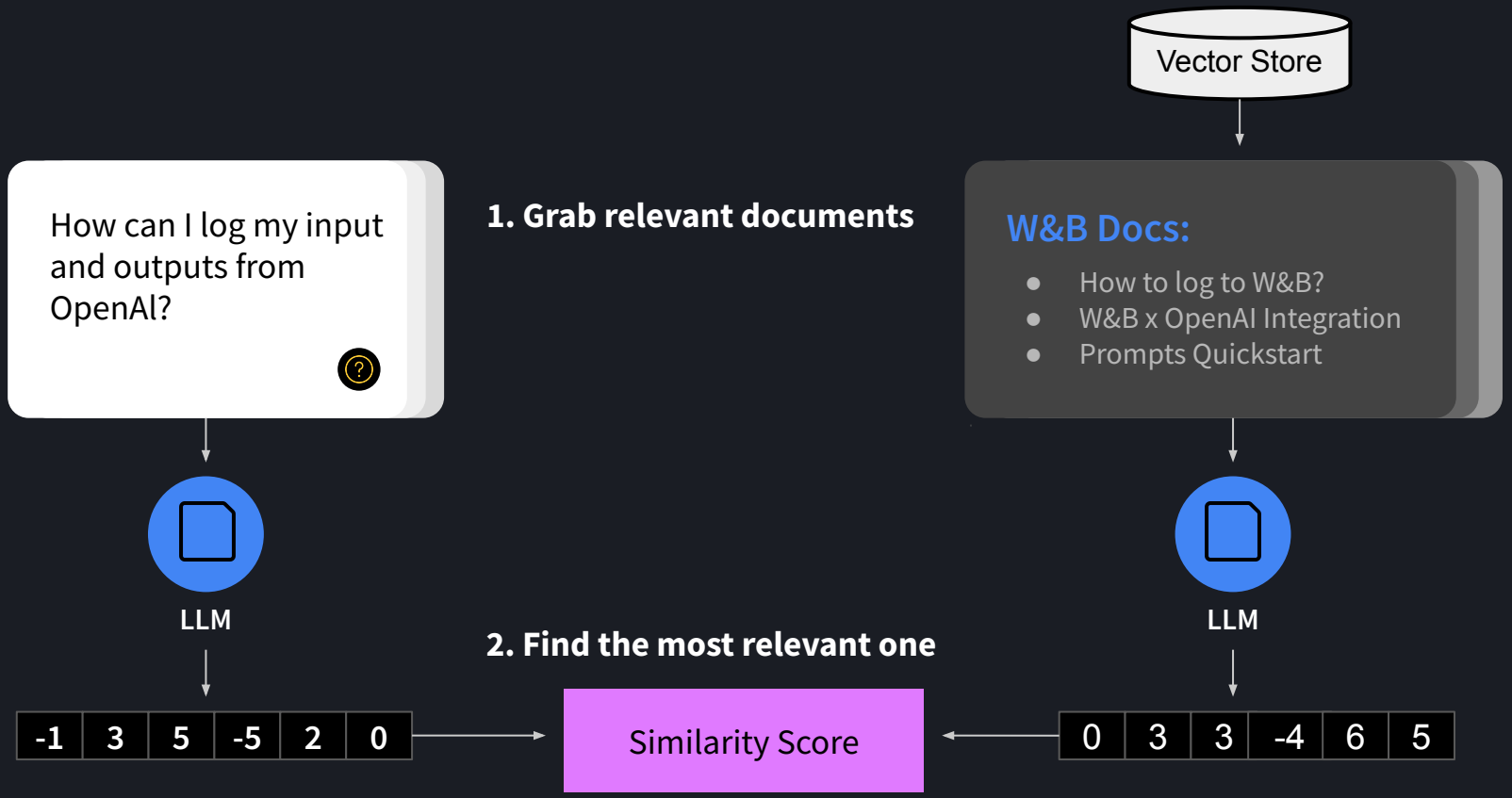
08 Demo: Agentboard Results

RAG: The Current Chatbot Workflow


User Question
Prompt template
Embedding model
Docs
Similarity search
PROMPT
K neighbors
ChromaDB Vector Store
GPT-4
Answer
Feedback

“


AShots Today at 12:01 PM
@wandbot (beta) How can I log my input and outputs from OpenAI


Vector Store
How can I log my input and outputs from OpenAI?
1. Grab relevant documents
W&B Docs:
How to log to W&B?
W&B x OpenAI Integration
Prompts Quickstart
LLM
LLM
2. Find the most relevant one
-1 3 5 -5 2 0
Similarity Score
0 3 3 -4 6 5

Add documents into the prompt

<!--Begin-->
<!--start-relevant-documents-->

How to log to Weights & Biases?

Log a dictionary of data to the current run's history.

log(

data: Dict[str, Any],

...

W&B x OpenAI Integration

1. Import autolog and initialize it

First, import autolog from wandb.integration.openai and initialise it.

import os

...

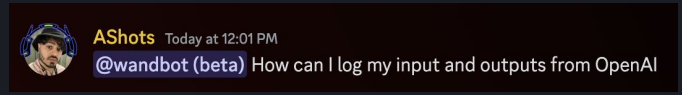
Prompts Quickstart

2. Log to a Trace

Now we will query OpenAI times and log the results to a W&B Trace.

...

<!end-relevant-documents-->
<!--Start-Question-->



<!--Final Answer in Markdown-->

Feed the prompt into GPT-4

How can I log my input and outputs from OpenAI?

W&B Docs:

How to log to Weights & Biases?

```
<!--start-relevant-documents→
<!end-relevant-documents-->
<!--Start-Question-->
```

Question

Relevant Docs

Prompt Template

GPT-4

GPT-4

wandbot (beta) BOT Today at 12:01 PM

Hi @AShots: Please note that **wandbot is currently in alpha testing** and will experience frequent updates.

Please do not share any private or sensitive information in your query at this time.

Please note that overly long messages (>1024 words) will be truncated!

Generating response...

You can log your input and output data from OpenAI using Weights & Biases by following these steps:

1. Install `wandb` and `openai` libraries:

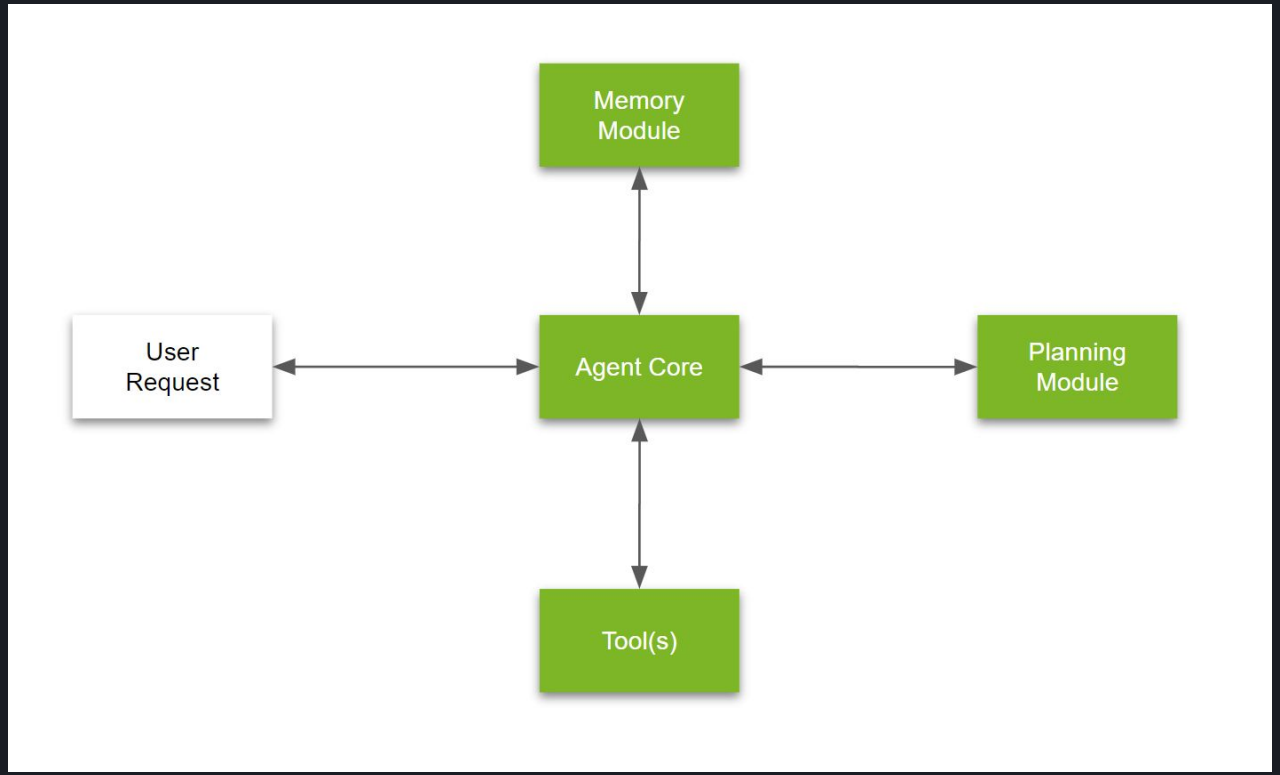
```
!pip install wandb openai -qU
```

2. Import and call `autolog`:

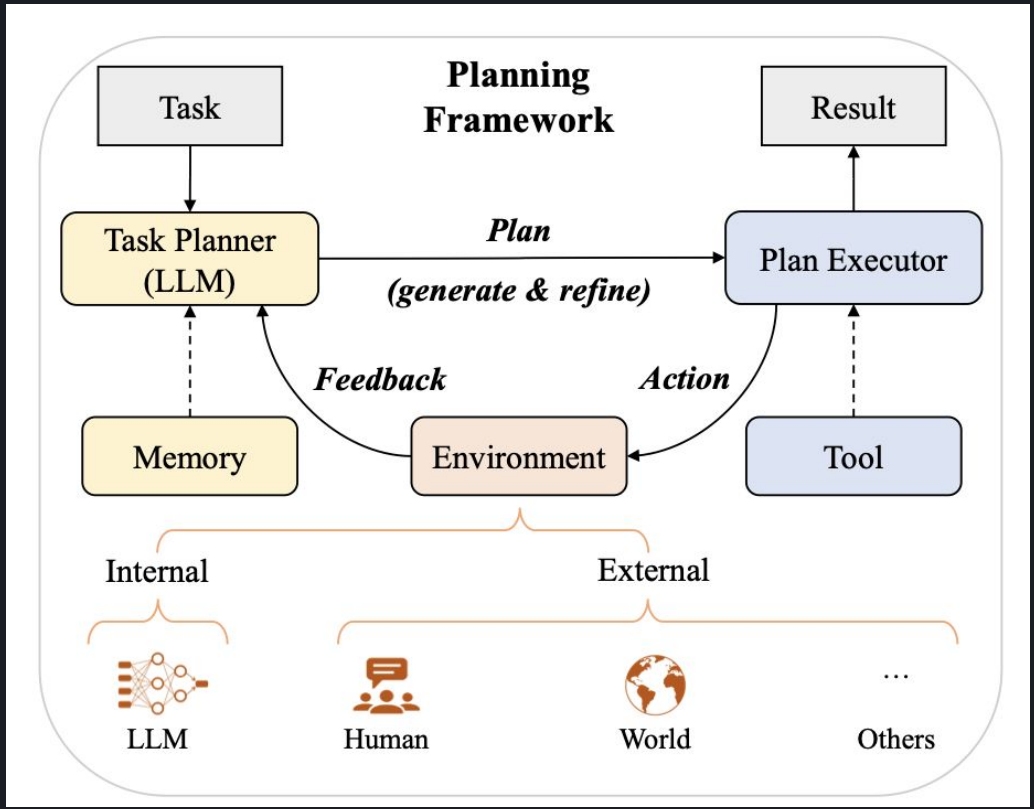
```
import openai
from wandb.integration.openai
import autolog

autolog({"project":"my_llm_project
"})
```

LLM Agents

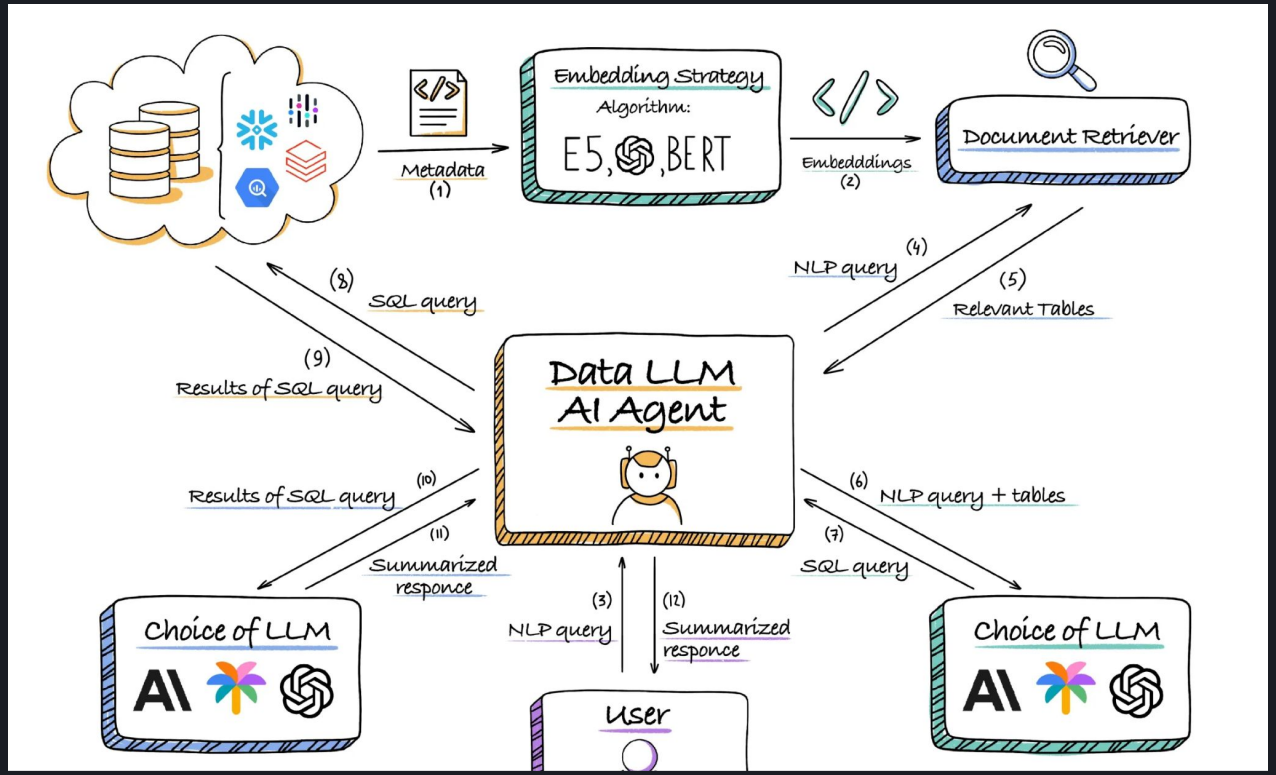
General Components of an Agent



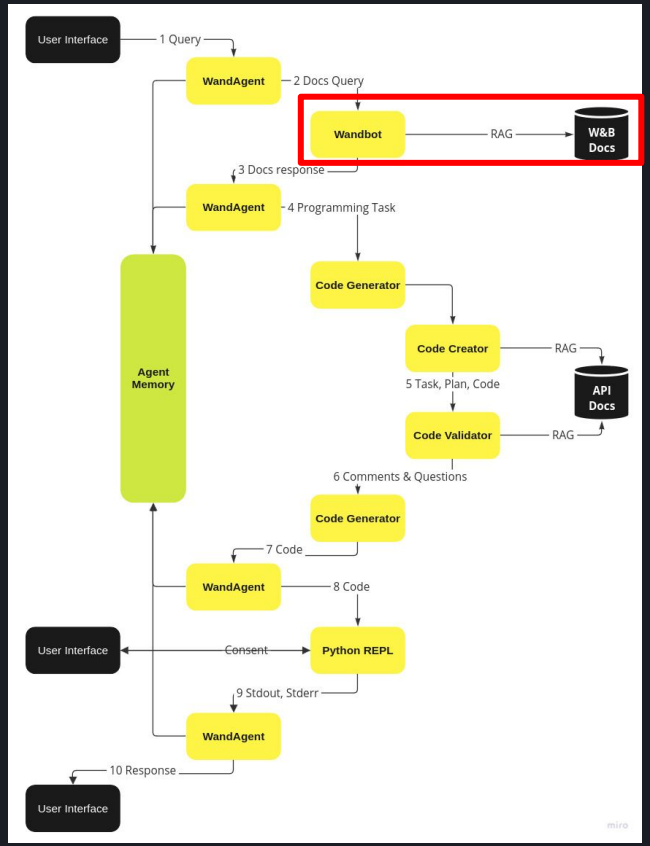
How Agent components interact



RAG can be considered a component of an Agent



We can expand our earlier RAG to be expanded as such:



We can expand the scope of Agents via MultiAgent usage

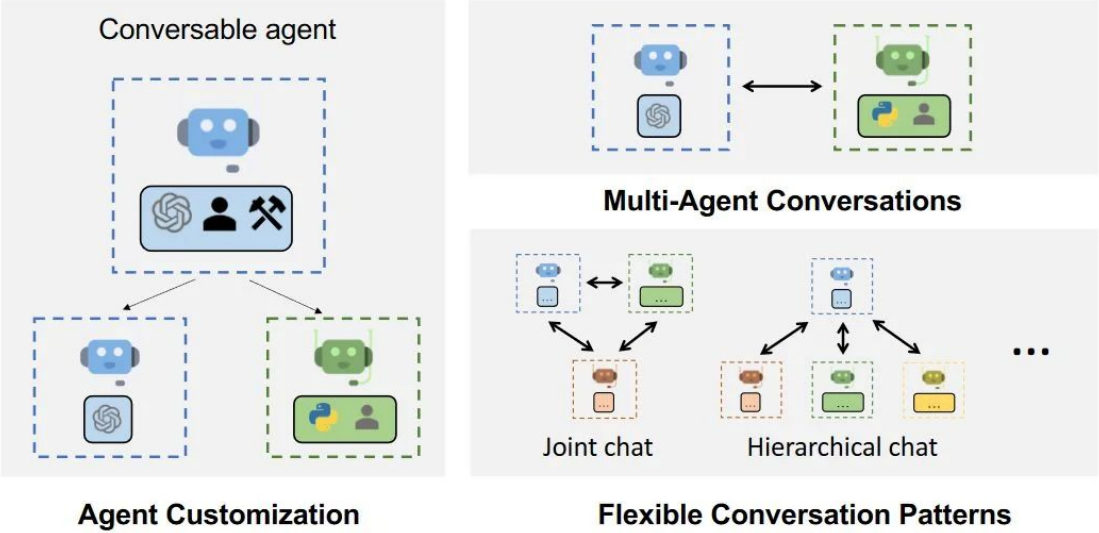


Figure 1: `AutoGen` enables complex LLM-based workflows using multi-agent conversations. (Left) `AutoGen` agents are customizable and can be based on LLMs, tools, humans and even a combination of them. (Top-right) Agents can converse to solve tasks. (Bottom-right) The framework supports many additional complex conversation patterns.

ChatDev is one such example

GPT-4 far outclasses all the other LLMs - proprietary and OSS as the underlying LLM for Agents

AgentBench

Models		Overall	OS	DB	KG	DCG	LTP	HH	WS	WB
API-based Model	gpt-4	4.41	36.81	33.67	52.14	50.00	17.60	78.00	58.60	22.59
	claude-v1.3	2.77	13.19	16.75	36.22	30.00	6.39	52.00	59.26	20.97
	gpt-3.5-turbo	2.55	32.64	15.00	27.20	30.00	14.85	14.00	67.21	15.69
	text-davinci-003	2.10	22.92	16.33	30.82	15.00	5.21	20.00	61.43	15.52
	claude-instant-v1.1	1.90	14.58	8.00	29.67	35.00	6.08	26.00	44.22	0.77
	text-davinci-002	1.46	4.86	13.67	18.87	25.00	2.32	14.00	60.15	1.11
	text-bison-001	1.39	4.17	12.75	17.12	20.00	0.12	4.00	46.06	20.46
	chatglm2-v0.2	1.31	14.58	13.67	6.85	10.00	12.62	6.00	19.35	12.87
	openchat_v3.2	1.15	9.72	5.33	6.84	0.00	9.54	8.00	50.17	14.92
	wizardlm-30b	0.83	14.58	12.67	2.33	10.00	4.47	6.00	10.60	3.07

Improving LLMs via Fine-tuning

LLMs can be improved via multiple ways

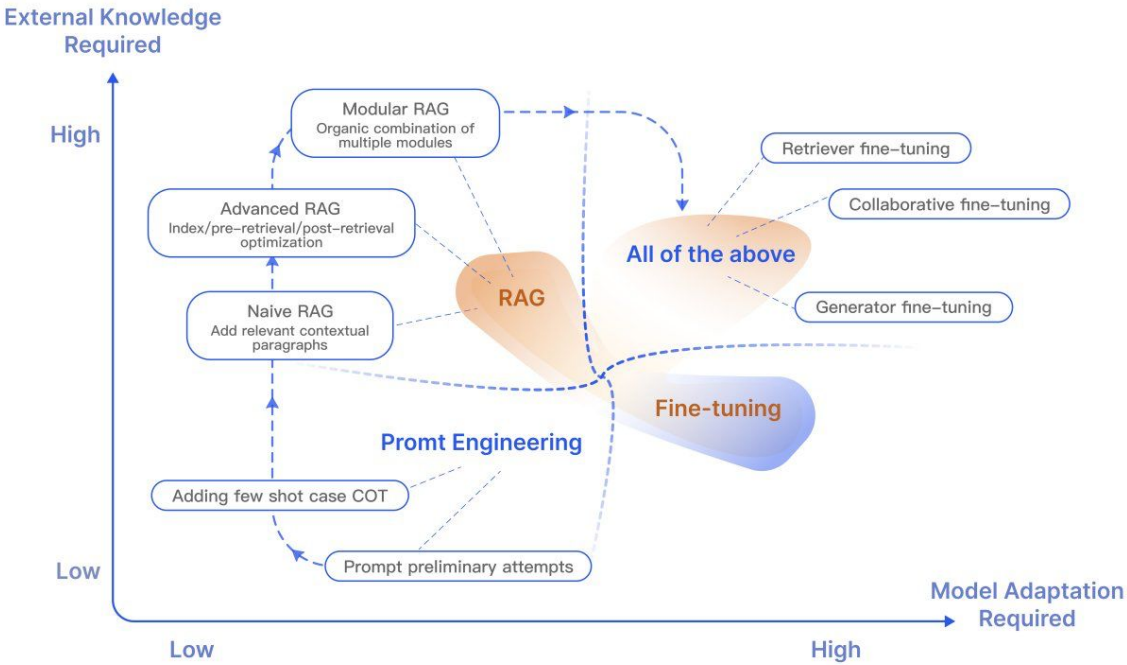
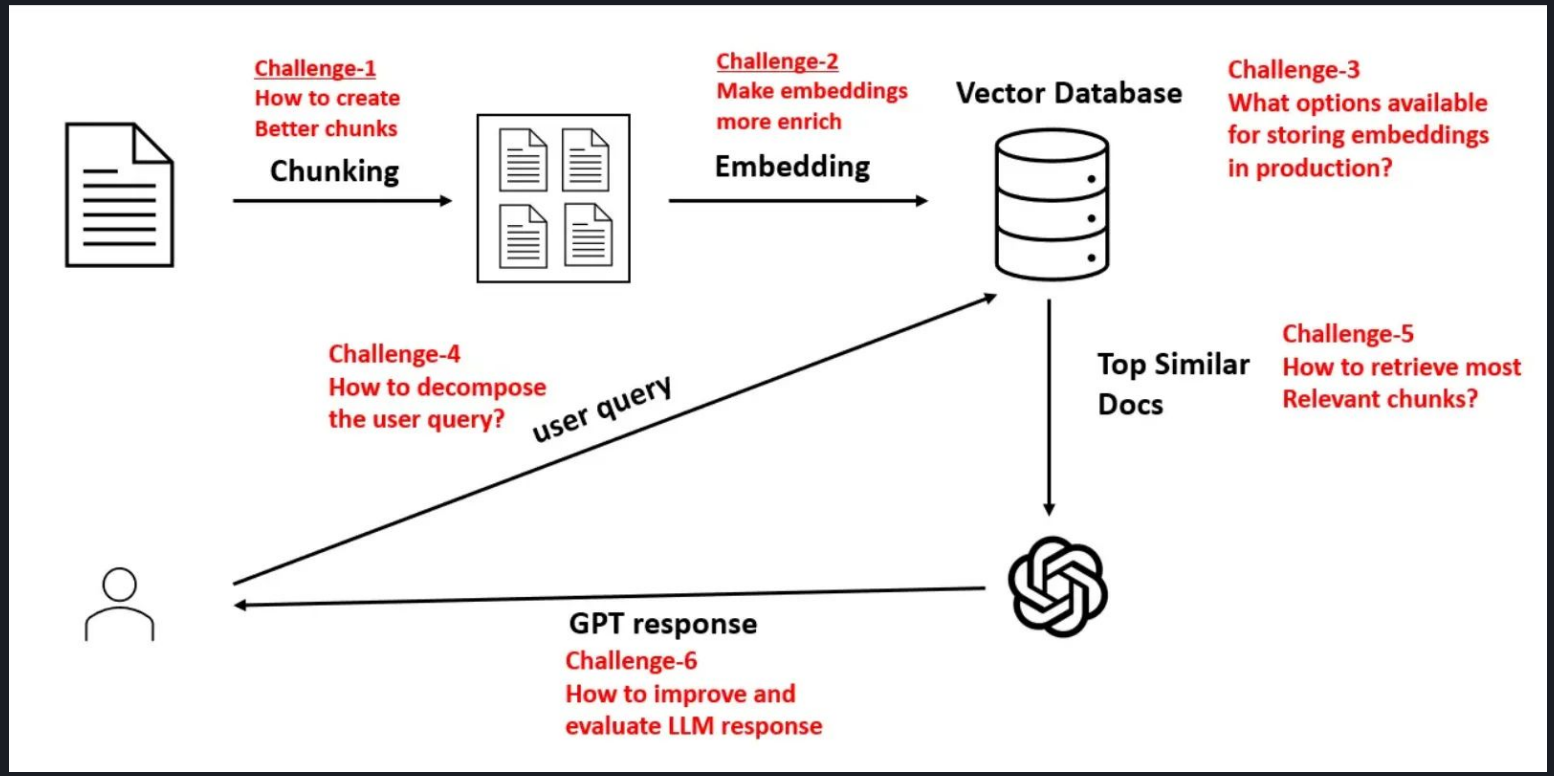
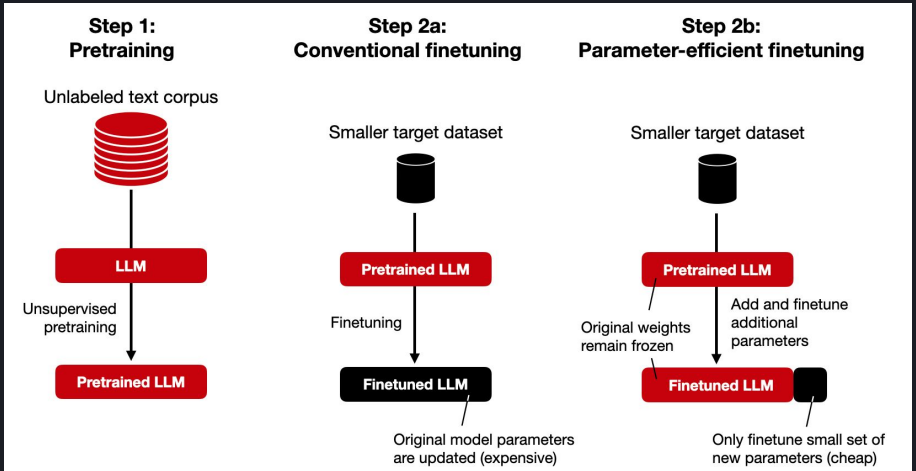


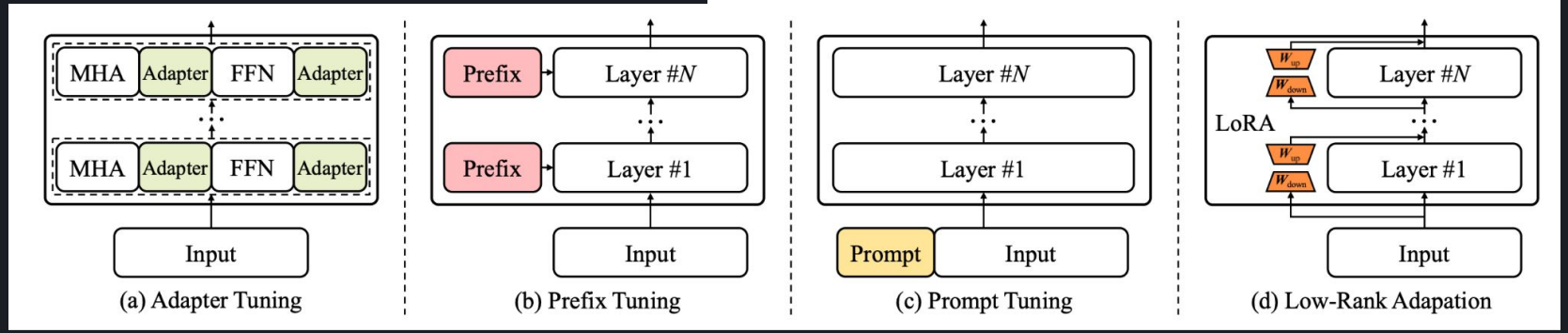
Figure 2: RAG compared with other model optimization methods

RAG has its own collection of challenges/experiments

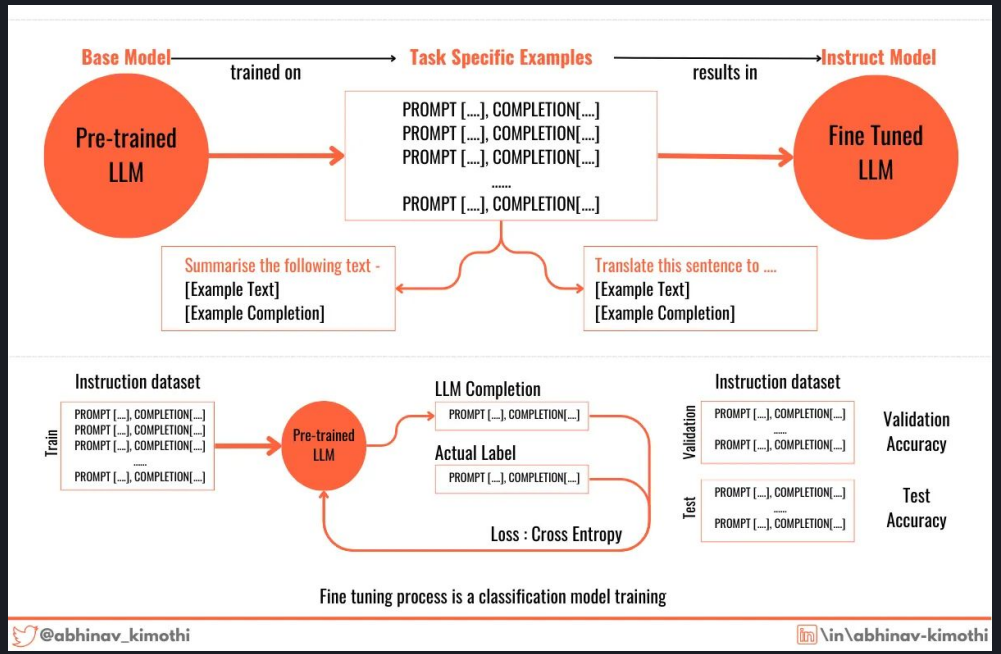


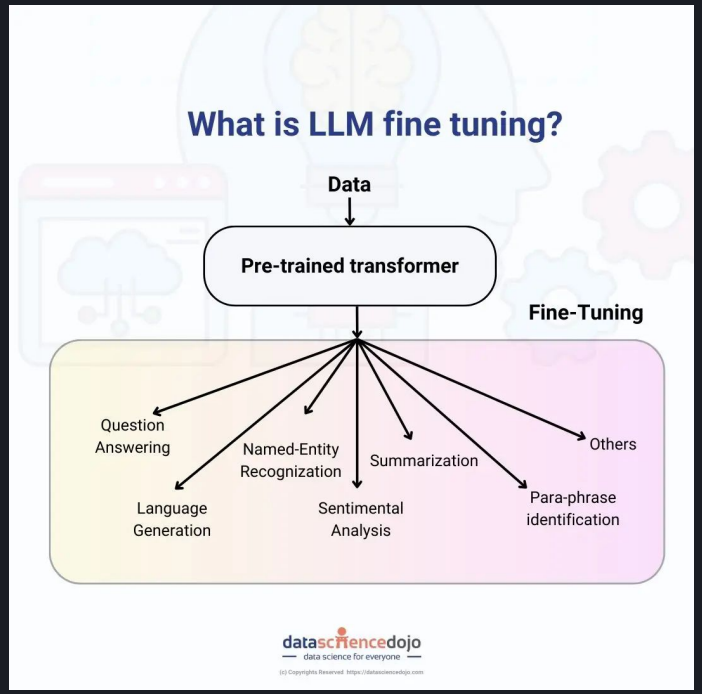
Many of these challenges can be mitigated by Fine Tuning (Parameter Efficient too!)





Why do we Fine-tune?





Top models on standard Evaluation data tend to come from Fine-tuning approaches

LLM Benchmark | Metrics through time | About | Submit here!

Search for your model (separate multiple queries with `;`) and press ENTER...

Select columns to show

- [x] Average
- [x] ARC
- [x] HellaSwag
- [x] MMLU
- [x] TruthfulQA
- [x] Winogrande
- [x] GSM8K
- [ ] Type
- [ ] Architecture
- [ ] Precision
- [ ] Merged
- [ ] Hub License
- [ ] #Params (B)
- [ ] Hub ❤️
- [ ] Model sha

Hide models

- [x] Private or deleted
- [x] Contains a merge/moerge
- [x] Flagged
- [ ] MoE

Model types

- [x] pretrained
- [x] fine-tuned on domain-specific datasets
- [x] chat models (RLHF, DPO, IFT, ...)
- [x] base merges and moerges
- [x] ?

Precision

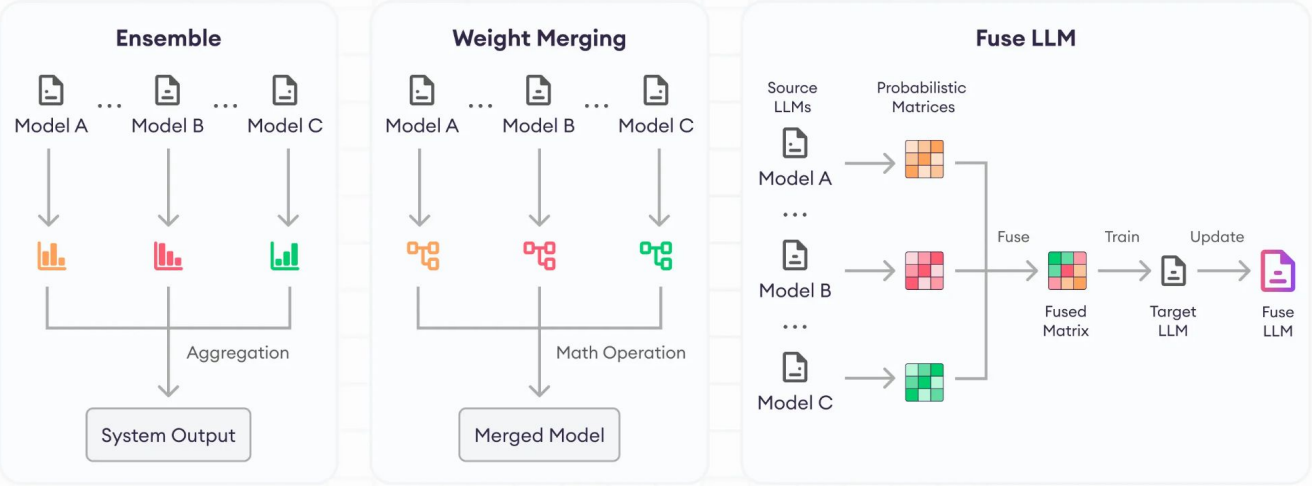
- [x] float16
- [x] bfloat16
- [x] 8bit
- [x] 4bit
- [x] GPTQ
- [x] ?

Model sizes (in billions of parameters)

- [x] ?
- [x] ~1.5
- [x] ~3
- [x] ~7
- [x] ~13
- [x] ~35
- [x] ~60
- [x] 70+

T	Model	Average	ARC	HellaSwag	MMLU	TruthfulQA
🔶	abacusai/Smaug-72B-v0.1	80.48	76.02	89.27	77.15	76.67
💬	moreh/MoMo-72B-lora-1.8.7-DPO	78.55	70.82	85.96	77.13	74.71
🔶	cloudyu/TomGrc_FusionNet_34Bx2_MoE_v0.1_DPO_f16	77.91	74.06	86.74	76.65	72.24
💬	yunconglong/Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B	77.44	74.91	89.3	64.67	78.02
🔶	TomGrc/FusionNet_34Bx2_MoE_v0.1	77.38	73.72	86.46	76.72	71.01
💬	moreh/MoMo-72B-lora-1.8.6-DPO	77.29	70.14	86.03	77.4	69
🔶	abacusai/Smaug-34B-v0.1	77.29	74.23	86.76	76.66	70.22
🔶	cloudyu/Truthful_DPO_TomGrc_FusionNet_34Bx2_MoE	77.28	72.87	86.52	76.96	73.28
🔶	yunconglong/DARE_TIES_13B	77.1	74.32	89.5	64.47	78.66

Model Merging has emerged to even take advantage of multiple successful trained/fine-tuned LLMs into one



Goal: Improving Agent Workflows via Fine-tuning for Smaller Language Models

Our best case Agent has certain properties similar to RL Agents that we want to optimize for

	RL Agent	Language Agent
Action space	Act	Act + Reason
Learning	Parameter update	Language update
Planning	Learned values	LM reasoning for both generation and evaluation

Example Action Space: ReAct

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1a) Standard

Answer: iPod

(1b) CoT (Reason Only)

Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

Answer: iPhone, iPad, iPod Touch

(1c) Act-Only

Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control …

Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: …

Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Act 4: Finish[yes]

(1d) ReAct (Reason + Act)

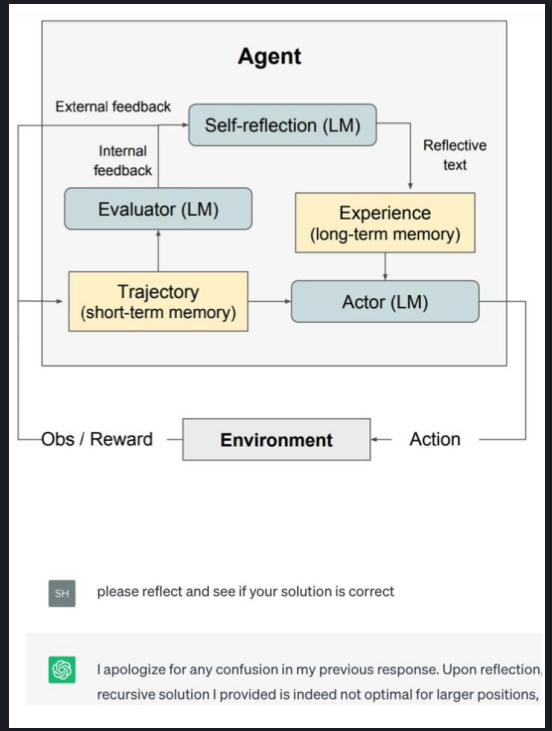
Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple … originally designed to control the Front Row media center program …

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software)', …]

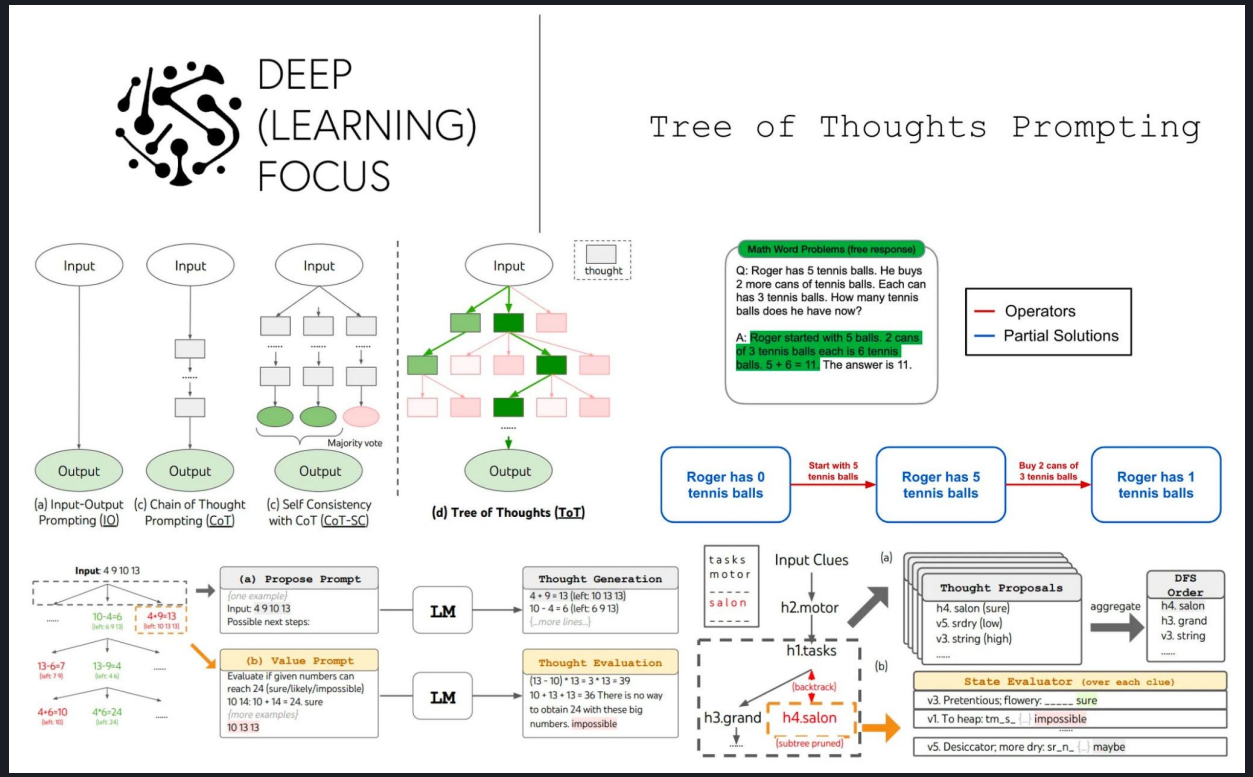
Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys]

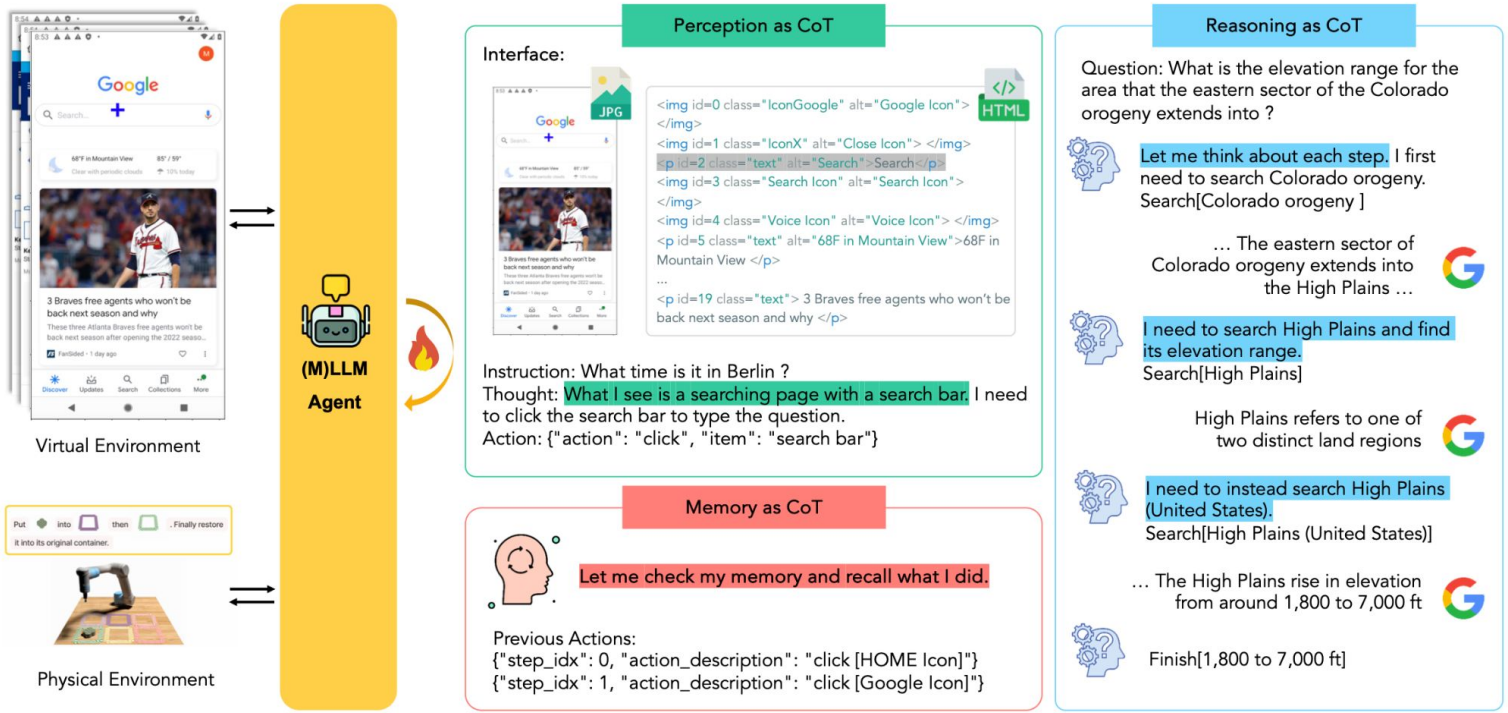
Example Learning: Reflexion



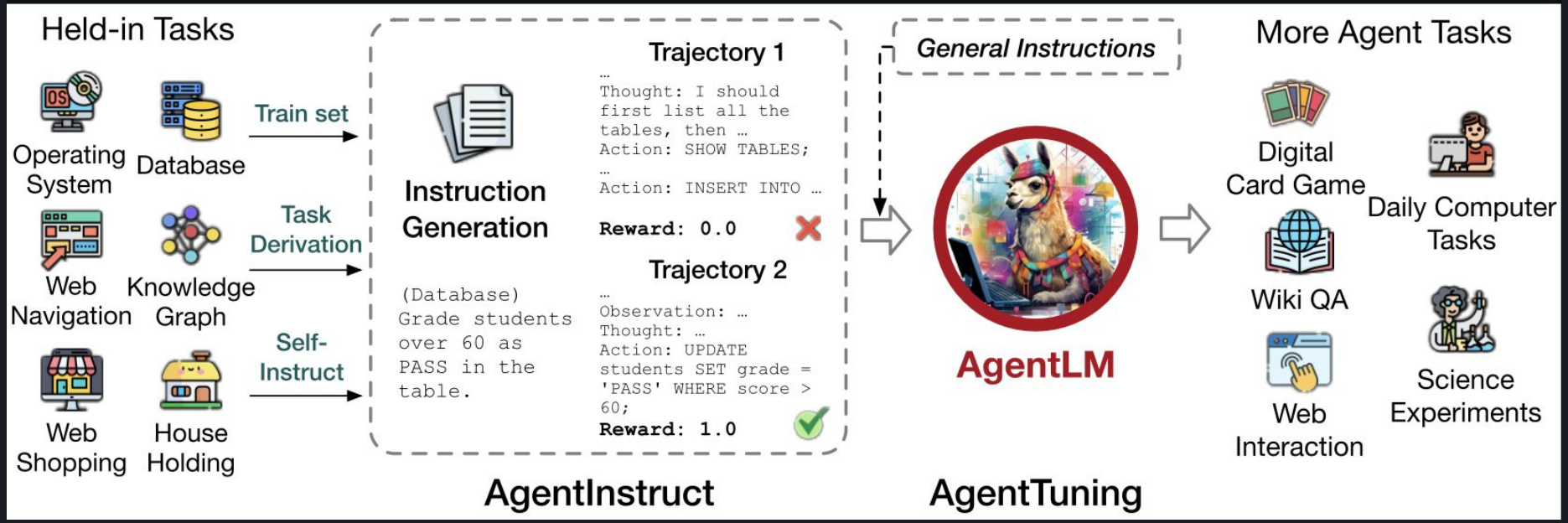
Example Planning: Thought Generation



We want to ensure consistent, parseable, and interpretable agent workflows when interacting with environments



Research such as AgentLM uses GPT4 generated Agent-interaction datasets to try to achieve this better

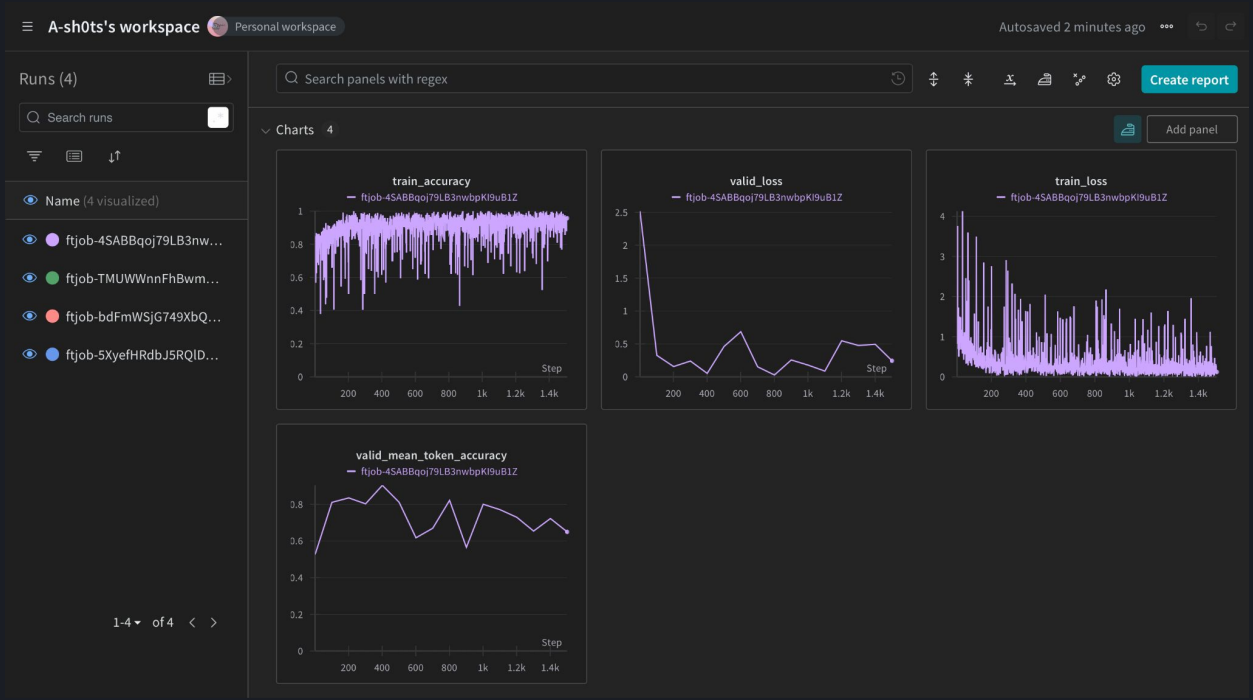


Split (6)
os · 195 rows

Search this dataset

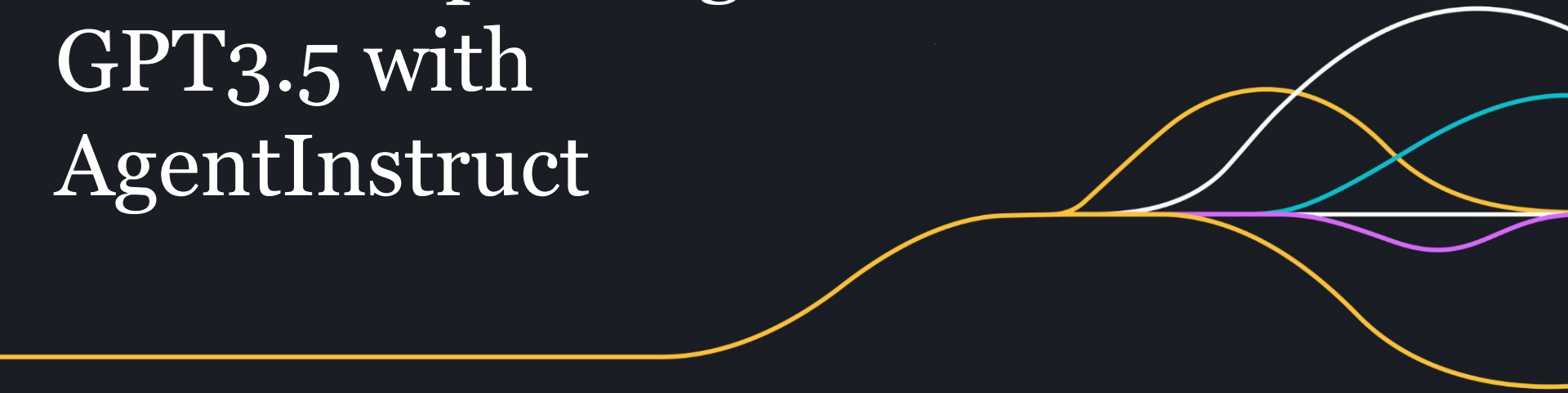
conversations list	id string · *lengths* 4 6
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_0
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_1
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_2
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_3
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_4
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_5
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_6
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_7
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_8
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_9
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_10
[{ "from": "human", "loss": null, "value": "You are an assistant that will act like a person, I'will play the role of linux(ubuntu) operating system. Your goal is to implement the operations required by me or answer to the question…	os_11

```python
from wandb.integration.openai.fine_tuning import WandbLogger

# Finetuning logic

WandbLogger.sync(fine_tune_job_id=FINETUNE_JOB_ID)
```

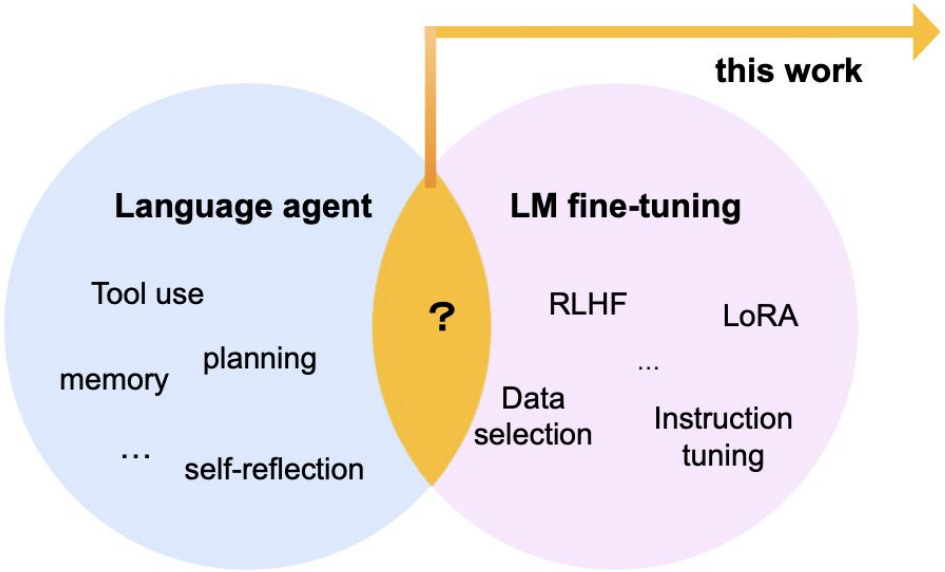


Weights & Biases

Demo: Improving GPT3.5 with AgentInstruct



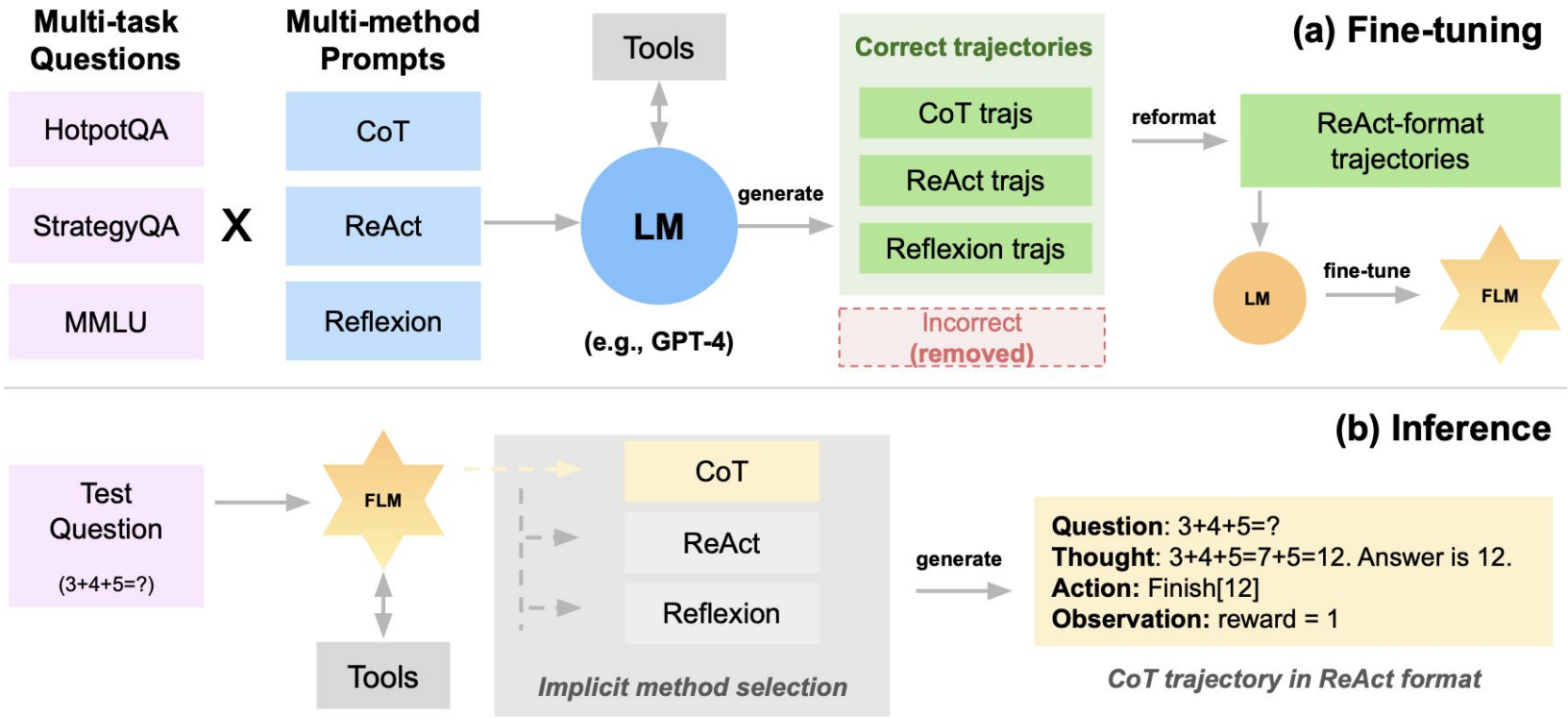
Case Study: FireAct - A different dataset than AgentInstruct



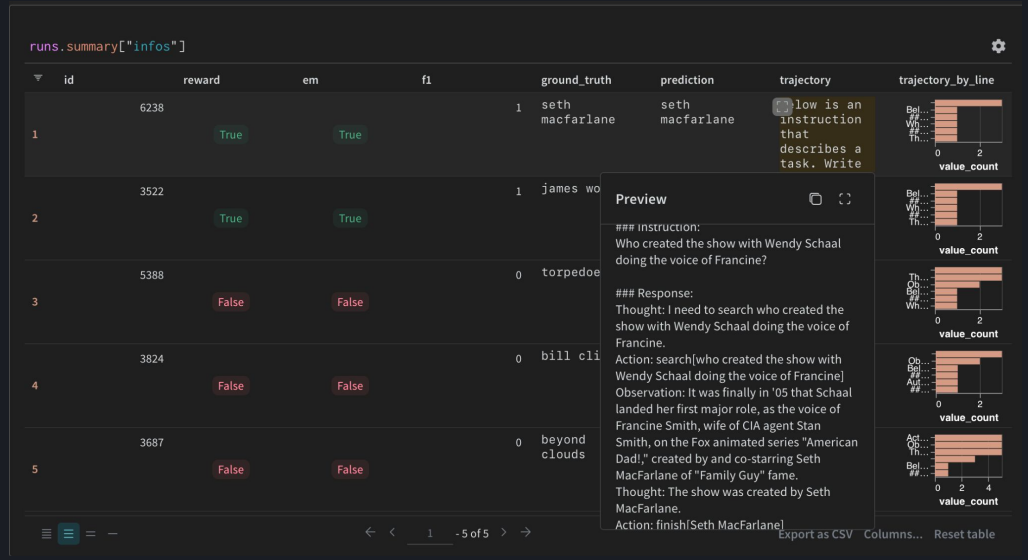
Language Agent Fine-tuning

- **Advantages over prompting**
  - More diverse learning support
  - Generalization and robustness
  - Cost and efficiency
  - Controllability
- **New research questions**
  - Scaling effect of fine-tune data?
  - Which LM type and size to use?
  - How to mix methods and tasks?
  - When to use tools or reflect?

How is FireAct tuned?

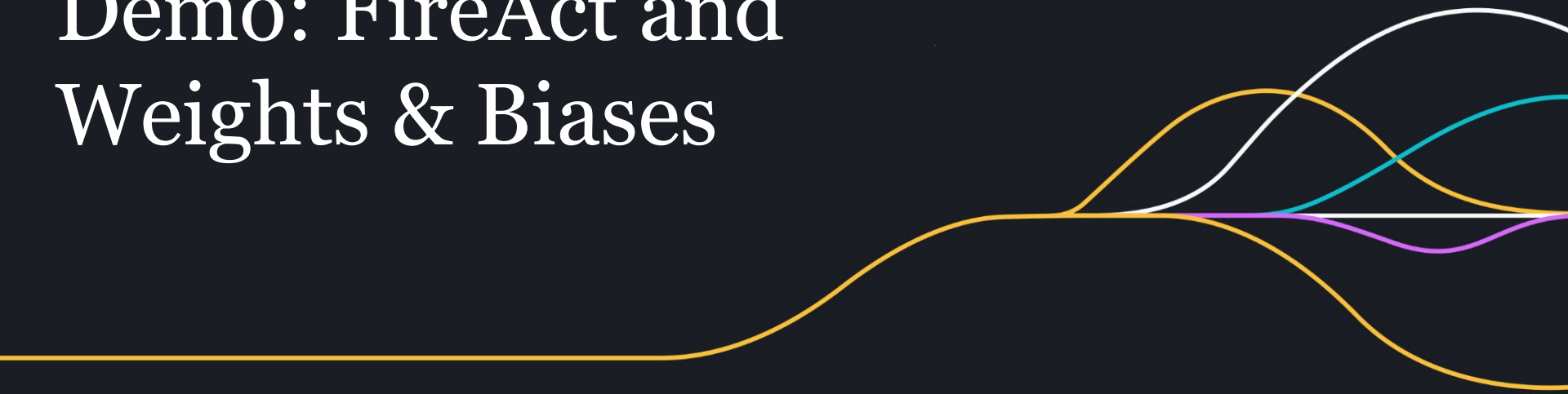


```python
os.environ["WANDB_PROJECT"] = "<my-amazing-project>"  # name your W&B projec
os.environ["WANDB_LOG_MODEL"] = "checkpoint"  # log all model checkpoints

from transformers import TrainingArguments, Trainer

args = TrainingArguments(..., report_to="wandb")  # turn on W&B logging
trainer = Trainer(..., args=args)
```



Weights & Biases

Demo: FireAct and Weights & Biases



How do we validate our LLMs actually improved?

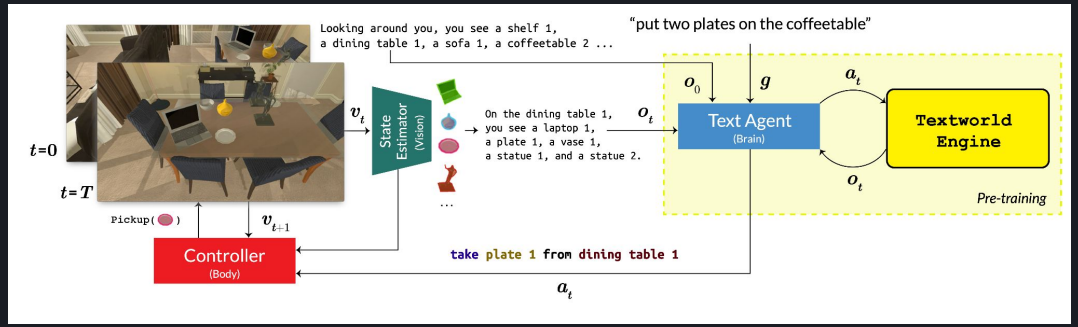
Standardized Agent Evaluation

Embodied AI

- Aims to bridge the gap between traditional AI, which often operates in abstract or purely digital spaces, and the physical world, by creating AI systems that can understand and interact with their environments in a more human-like way.
- This involves integrating sensory information, physical actions, and cognitive functions, enabling AI agents to perform tasks and solve problems in complex, real-world settings.

Alfworld

- Tests agents on how to do various tasks, like **making a cup of tea or cleaning a window**.
- It does this by combining two things:
  - a text-based game called ALFRED, where tasks are described in words,
  - and a 3D environment from AI2-THOR, which is like the virtual world or setting where these tasks take place.



Science World

- Science World is a simulated environment designed for AI agents to learn and understand scientific concepts and experiments through interaction and exploration.
- Providing a virtual laboratory where AI agents can perform scientific experiments, manipulate variables, and observe outcomes.

SCIENCEWORLD Task 3 (Test Electrical Conductivity Subtask)

Task Description
Your task is to determine if a metal fork is electrically conductive. The metal fork is located around the kitchen. First, focus on the metal fork. If conductive, place it in the blue box. If nonconductive, place it in the green box.

Agent starts in a random location, navigates to kitchen

Kitchen
This room is called the kitchen. In it, you see:
a fridge. The freezer door is closed.
a cupboard. The cupboard door is closed.
a table. On the table is: a glass cup (containing nothing).
a stove, which is turned off. On the stove is: nothing.
a sink, which is turned off. In the sink is: nothing.
a thermometer, currently reading a temperature of 10 degrees celsius
a metal fork
a counter. On the counter is: a drawer, a bowl (containing an orange, a banana, a potato, a red apple).
You also see:
A door to the outside (that is open)
A door to the bathroom (that is open)
A door to the hallway (that is open)

In room with task object → SCORE 0.1

Action: ***focus on metal fork***
You focus on the metal fork.

SCORE 0.5

Action: ***pick up metal fork***
You move the metal fork to the inventory.

Agent navigates to room with electrical components

Workshop
This room is called the workshop. In it, you see:
a electric buzzer, which is off
a electric motor, which is off
a solar panel, which is off
a table. On the table is: a blue wire, a battery, a red light bulb, which is off, a black wire, a switch, which is off, a orange wire.
a blue answer box
a green answer box
You also see:
A door to the hallway (that is open)

In room with components → SCORE 0.6

Action: ***connect battery cathode to orange wire terminal 1***
cathode on battery is now connected to terminal 1 on orange wire.

Action: ***connect metal fork terminal 1 to cathode in red light bulb***
terminal 1 on metal fork is now connected to cathode on red light bulb.

Agent continues wiring a circuit containing the metal fork

SCORE 0.7

Successfully made circuit

Action: ***examine red light bulb***
a red light bulb, which is on.

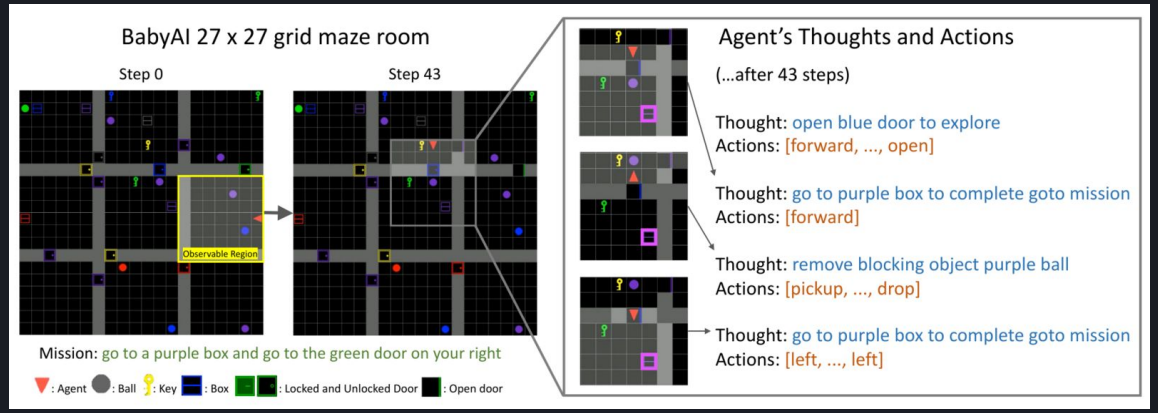
Action: ***move metal fork to blue answer box***
(disconnecting metal fork) You move the metal fork to the blue box.

SCORE 1.0

Task Completed.

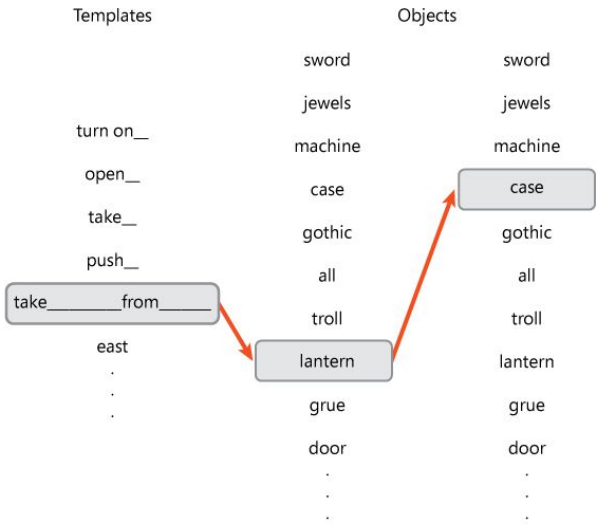
BabyAI

- BabyAI aims to study the development of artificial intelligence in an environment that simulates the learning stages of a human infant, focusing on understanding and executing simple language commands.
- The experimental design involves creating simplified, controlled environments where AI agents perform tasks based on verbal instructions, mirroring the way babies learn through interaction and repetition.



Jericho

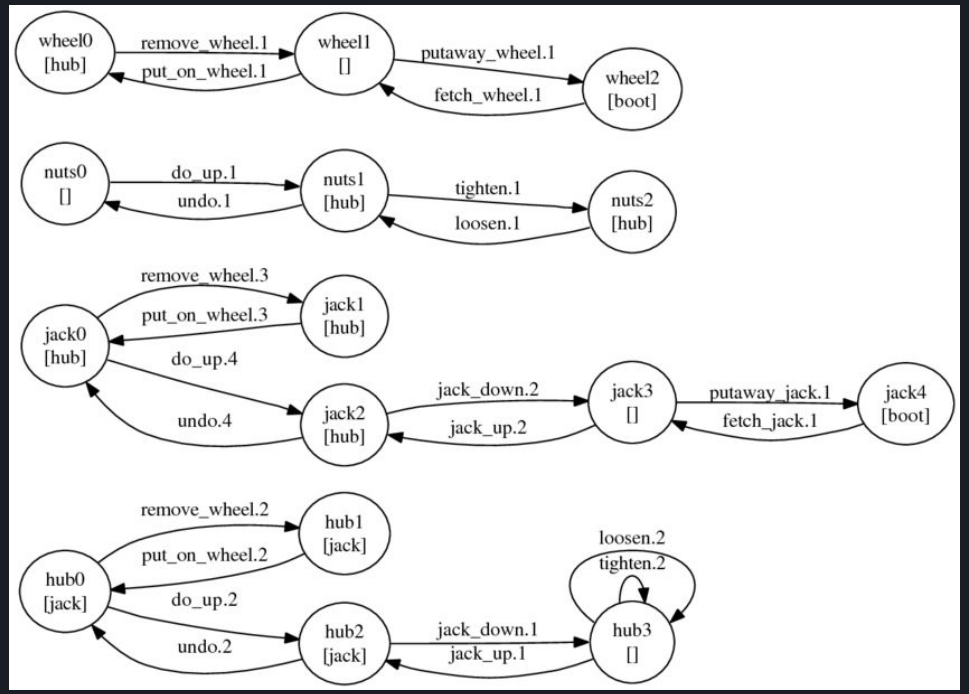
- interact with a variety of text-based adventure games, challenging them to understand narratives, follow storylines, and make strategic decisions based on textual information



Microsoft

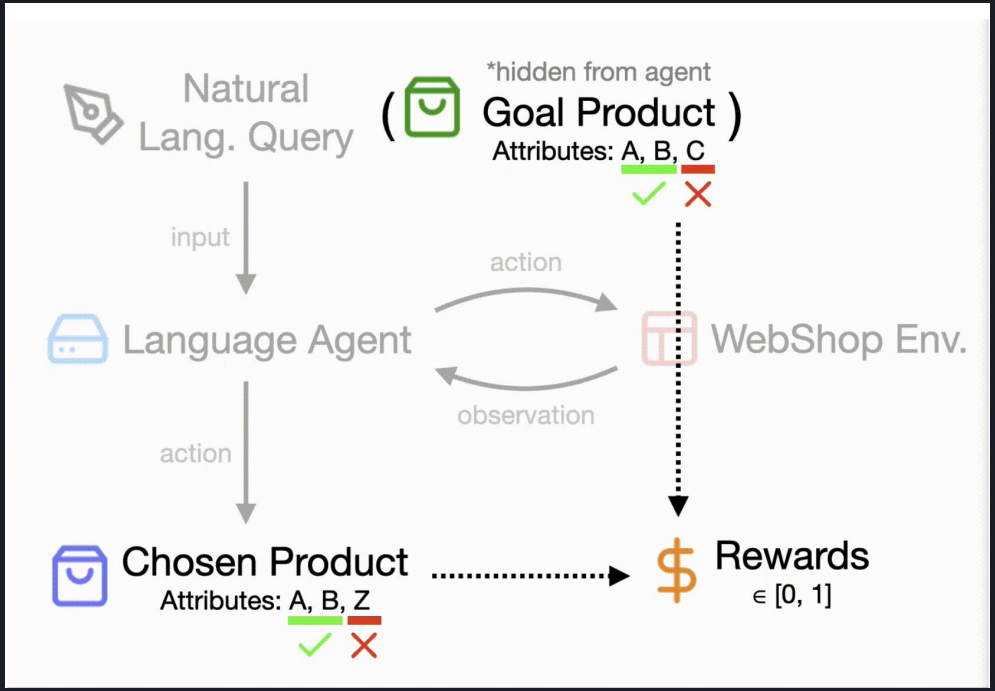
PDDL

- 4 representative game domains, Gripper, Barman, Blocksworld, Tyreworld to benchmark LLM agents in diverse scenarios, where the agent needs to move balls across rooms, make cocktails, rearrange blocks and pump up and install new tyres to cars.



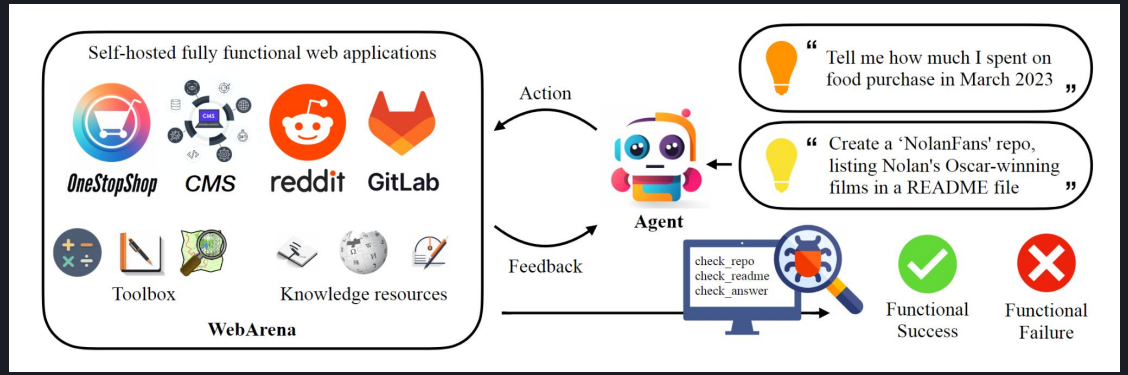
WebShop

- Online shopping experience with over a million real-world products and crowd-sourced text instructions. Agents navigate through various webpage types (search, results, item, item-detail) and perform actions (search, choose) based on given textual instructions, with the goal of finding and purchasing a specific item.



WebArena

- four applications:
  - online shopping,
  - discussion forums,
  - collaborative development,
  - and business content management.
- It supports 11 different web browsing actions. The observation space consists of structured web content.



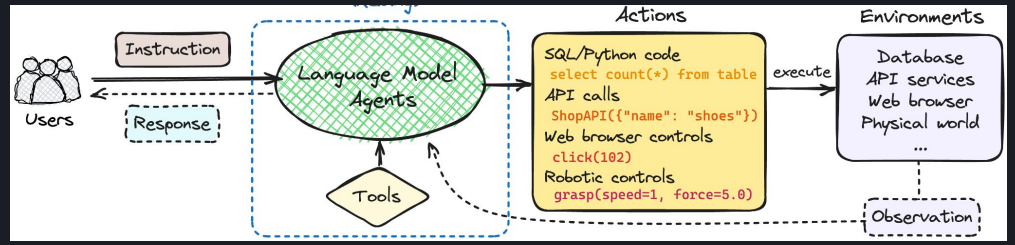
Tools Usage

Tool-Query

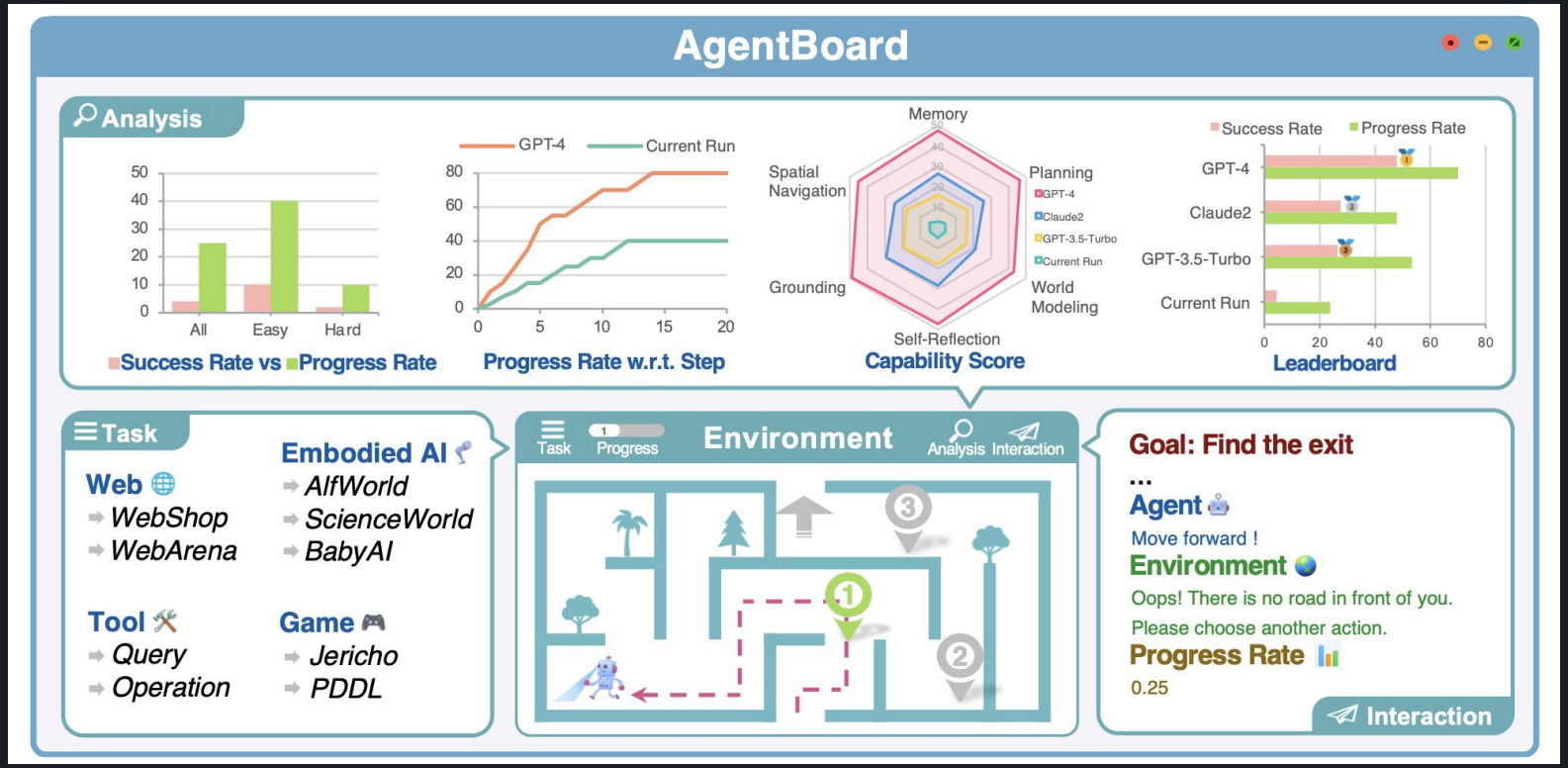
- Consists of three sub-environments: Weather Environment, Movie Environment and Academia Environment. Tools involved primarily serve the purpose of querying information about weather, movie and computer science academia.

Tool-Operation

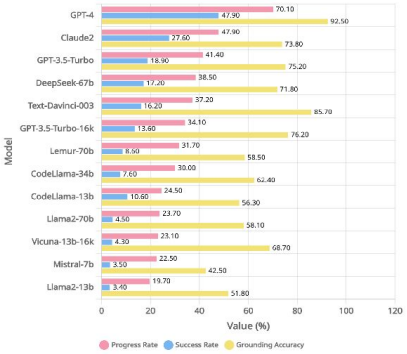
- Comprises two sub-environments: Todo Environment and Sheet Environment. Tools incorporated within these environments are designed for accessing and manipulating information about personal agendas and spreadsheets.

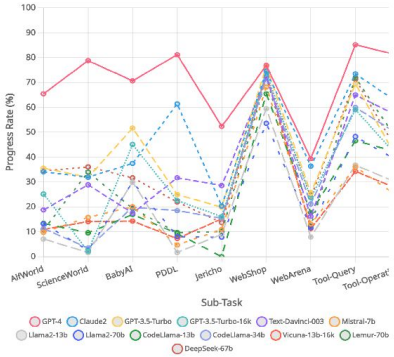


We want analytical evaluation for LLMs as generalist agents to perceive and act within various environments

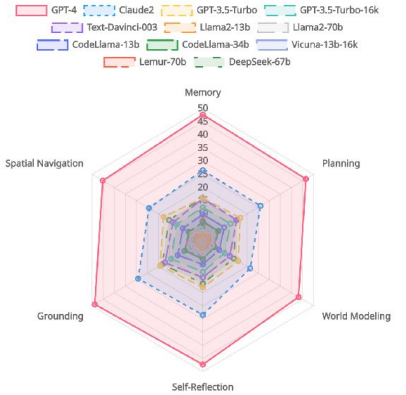


Main Performance

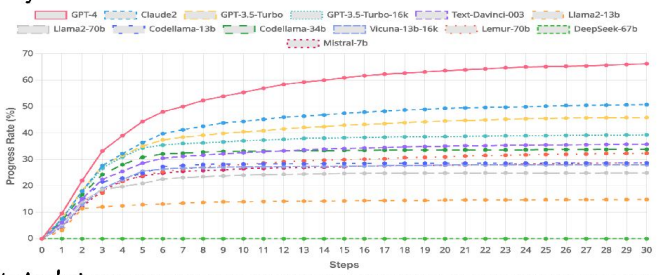




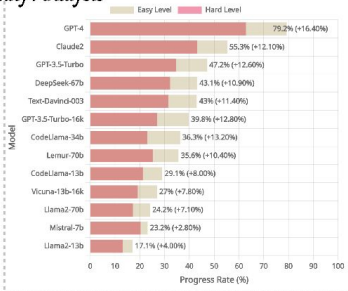
Dimensional Analysis

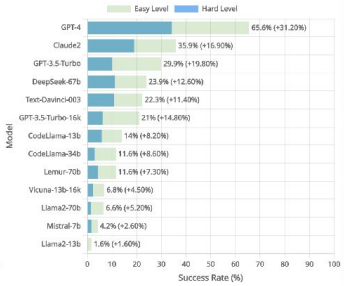


Steps Analysis



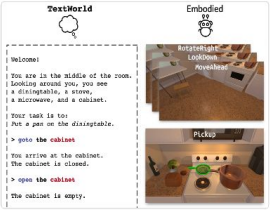
Difficulty Analysis





Trajectory Example

> Task Description



ALFWorld contains interactive TextWorld environments that parallel embodied worlds in the ALFRED dataset, divided into six categories of sub-tasks, encompassing a total of 134 environments.

> Example (By GPT-3.5)

Goal

heat some apple and put it in fridge.

Init observation: You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 3, a countertop 2, a countertop 1, a drawer 3, a drawer 2, a drawer 1, a

> go to fridge 1

The fridge 1 is closed.

> open fridge 1

You open the fridge 1. The fridge 1 is open. In it, you see a apple 1, a bowl 3, a cup 2, a cup 1, a egg 3, a lettuce 1, a potato 2, a potato 1, and a tomato 1. (reward: 0.25)

> take apple 1 from fridge 1

You pick up the apple 1 from the fridge 1. (reward: 0.5)

> go to microwave 1

The microwave 1 is closed.

> open microwave 1

You open the microwave 1. The microwave 1 is open. In it, you see a egg 2.

Weights & Biases

Demo: AgentBoard and Weights and Biases

Conclusion

- GPT4 far outclasses (Feb 2024) other models as the underlying LLM reasoning engine for Agent applications
- Smaller models can be fine-tuned such that they are better suited for specialized agent applications but tend to lose generalizability
- LORA did not serve as an effective method for fine-tuning for specific applications
- There still does not exist an effective dataset to improve LLM applications effectively for Agent usage
- Off the shelf top models from the Agent leaderboard also are not primed for Agent application usage
- Finetuning LLMs with Agent-specific data can improve the Agent behavior of specific applications against models of the same parameter size

The Weights & Biases platform helps you streamline your ML workflow from end to end

Experiments Experiment tracking	**Reports** Collaborative dashboards	**Artifacts** Dataset and model versioning	**Tables** Interactive data visualization	**Sweeps** Hyperparameter optimization
Launch Automate ML workflows	**Models** Model lifecycle management	**LLM Monitoring** Observability for production ML	**Prompts** LLMOps and prompt engineering	**Weave** Interactive ML app builder

ML Tools Ecosystem

OpenAI

Hugging Face

PyTorch Lightning

Keras

scikit learn

PyTorch

W&B Integration

```
args = TrainingArgs(report_to="wandb")
trainer = Trainer(args)
```

```
wandb_logger = WandbLogger()
trainer = Trainer(logger=wandb_logger)
```

Core Primitives

```
wandb.log({"wandb": train_loss})
wandb.Table()
wandb.Artifact()
```

wandb.me/courses

Want to Dive Deeper?

Weights & Biases | mosaicML

Training and Fine-tuning Large Language Models (LLMs)

Chief Scientist
Jonathan Frankle
MosaicML

PyTorch Engineer
Mark Saroufim
Meta

Principal SDE Manager
Weiwei Yang
Microsoft

MLE
Darek Kleczek
Weights & Biases

MLE
Ayush Thakur
Weights & Biases

Weights & Biases

LLM Engineering: Structured Outputs

Jason Liu

W&B Dashboards from Presentation

Resources

- Finetuning GPT3.5 on AgentInstruct:
  https://wandb.ai/a-sh0ts/AgentInstruct-OpenAI-Fine-Tune?nw=nwuserash0ts
- Finetuning LLama2 on Agent ReAct Trajectories - FireAct:
  https://wandb.ai/a-sh0ts/fireact_agent_finetuning?nw=nwuserash0ts
- Agentboard Report:
  https://wandb.ai/agentboard/llm-agent-eval-gpt-35-turbo-all/reports/Using-Wandb-to-Launch-AgentBoard--VmlldzO2MTg1Njc4
- Agentboard Evaluations:
  - GPT3.5 x AgentInstruct:
    https://wandb.ai/a-sh0ts/llm-agent-eval-ft-gpt-3.5-turbo-0613-weights-biases-8shLxWK0-all?nw=nwuserash0ts
  - AgentLM (Llama2 x AgentInstruct):
    https://wandb.ai/a-sh0ts/llm-agent-eval-thudm-agentlm-13b-all?nw=nwuserash0ts
  - Current Top 13B Model on OpenLLMLeaderboard:
    https://wandb.ai/a-sh0ts/llm-agent-eval-yunconglong-moe-13b-dpo-all?nw=nwuserash0ts
  - FireAct
    https://wandb.ai/a-sh0ts/llm-agent-eval-forestai-fireact-llama-2-7b-all?nw=nwuserash0ts

Stay in touch!

wandb.me/discord
wandb.me/courses

LinkedIn: Anish Shah