CS 4644-DL / 7643-A
ZSOLT KIRA

Agents

Slides by: Angel Chang

- **Projects**
    - Final project report due **July 26th**

RL: Sequential decision making in an environment with evaluative feedback.

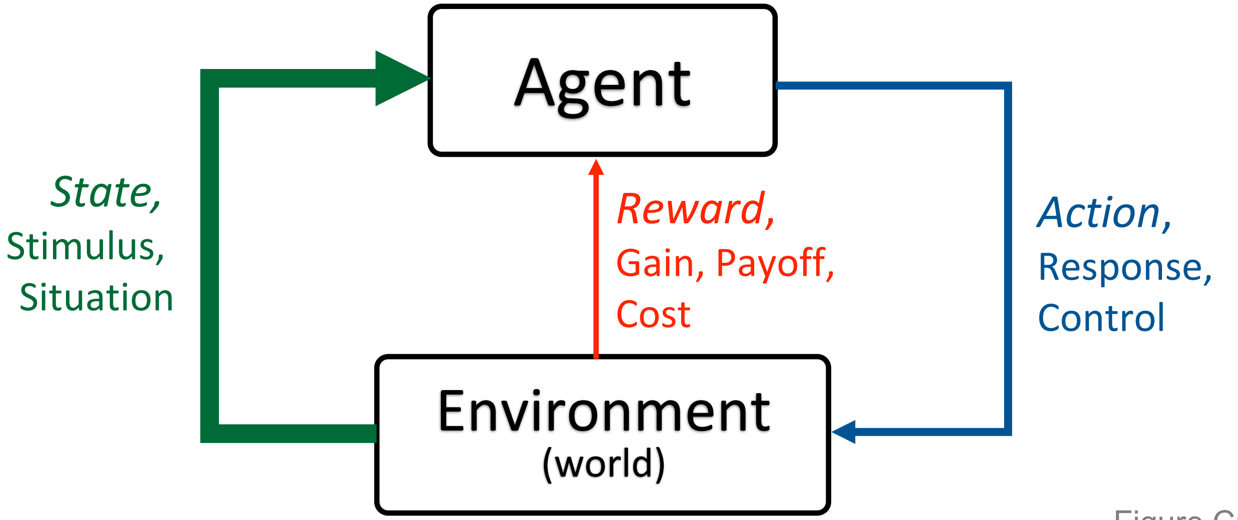
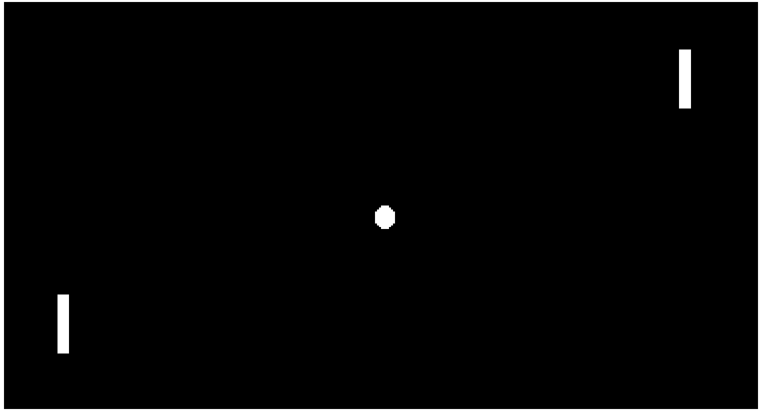
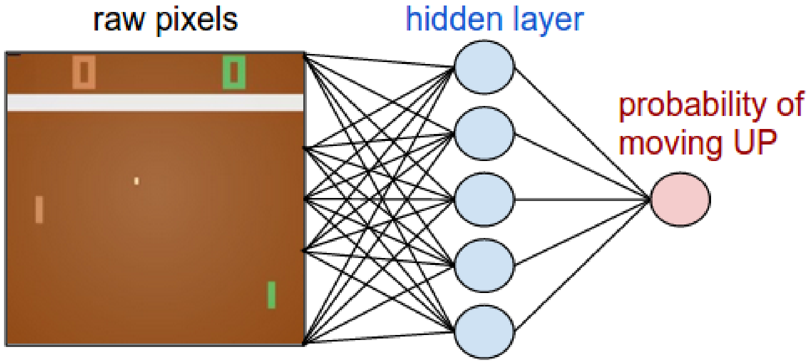


Figure Credit: Rich Sutton

- **Environment** may be unknown, non-linear, stochastic and complex.
- **Agent** learns a **policy** to map states of the environments to actions.
  - Seeking to maximize cumulative reward in the long run.

What is Reinforcement Learning?

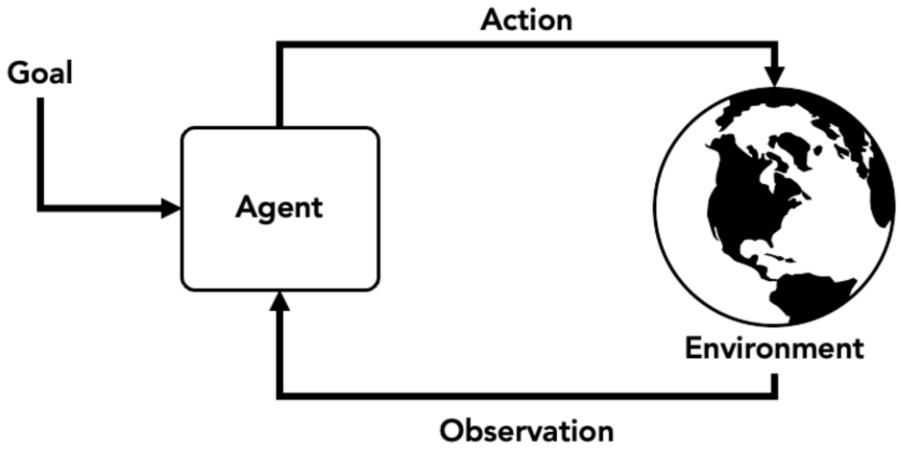




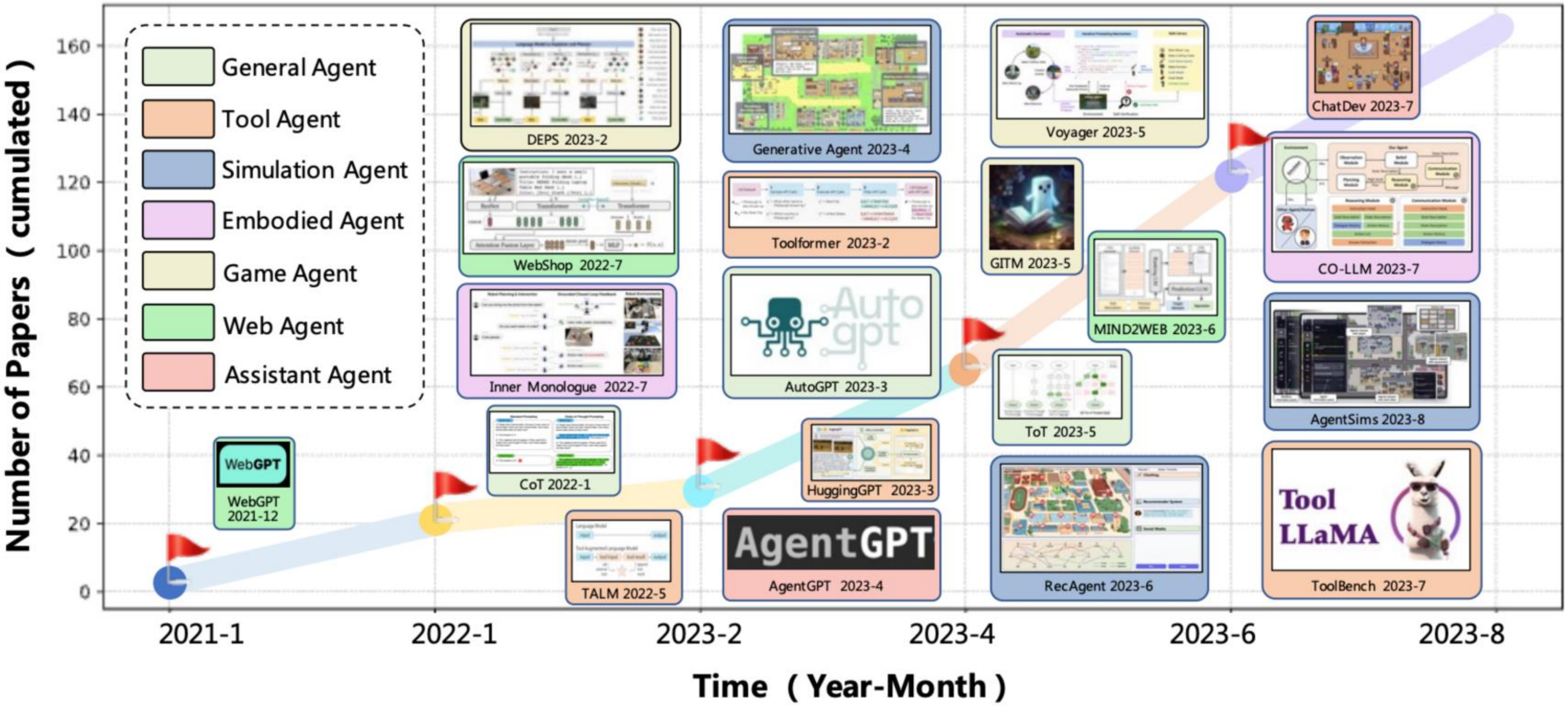
Pong from Pixels

LLMs as agents

- There is a lot of knowledge in LLMs
- But they can't "act"
- Can we leverage LLMs to build a "smart" agent that can interact with the environment to achieve given goals?

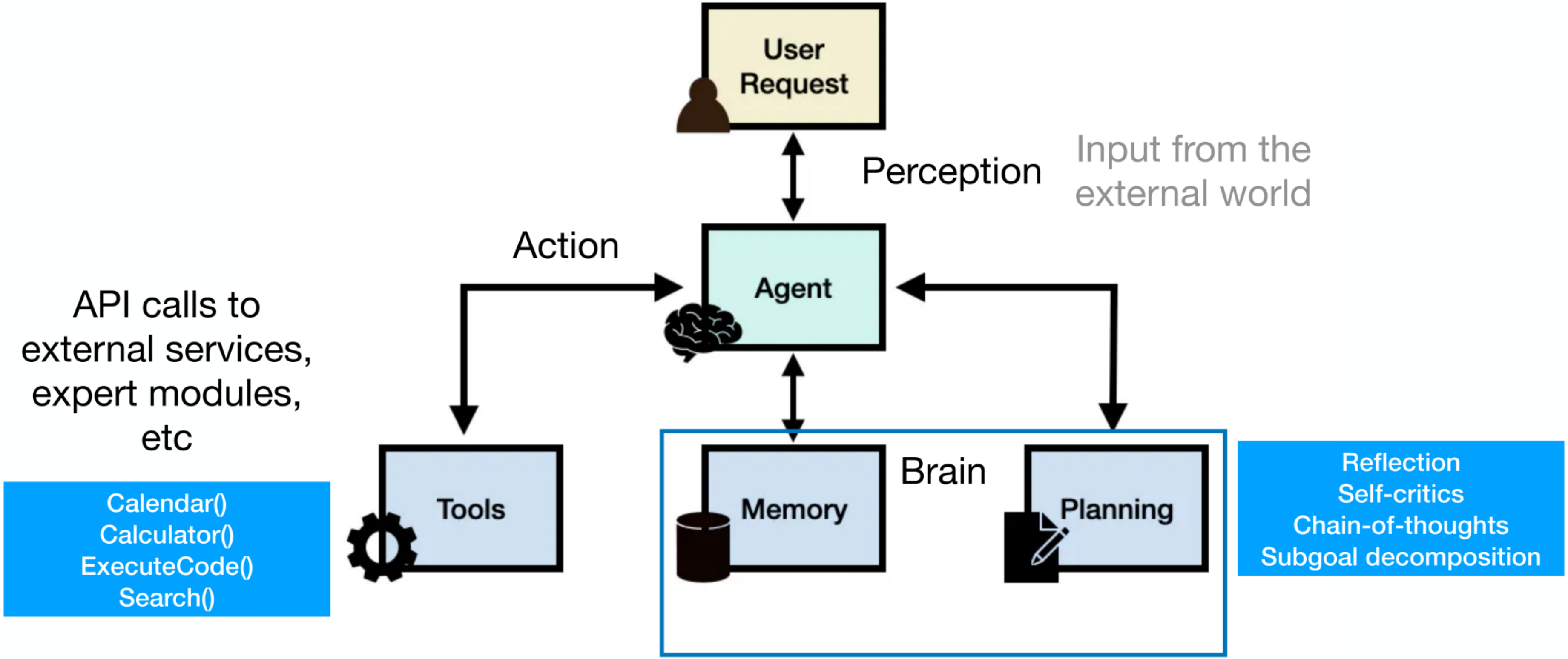


Lots of work!

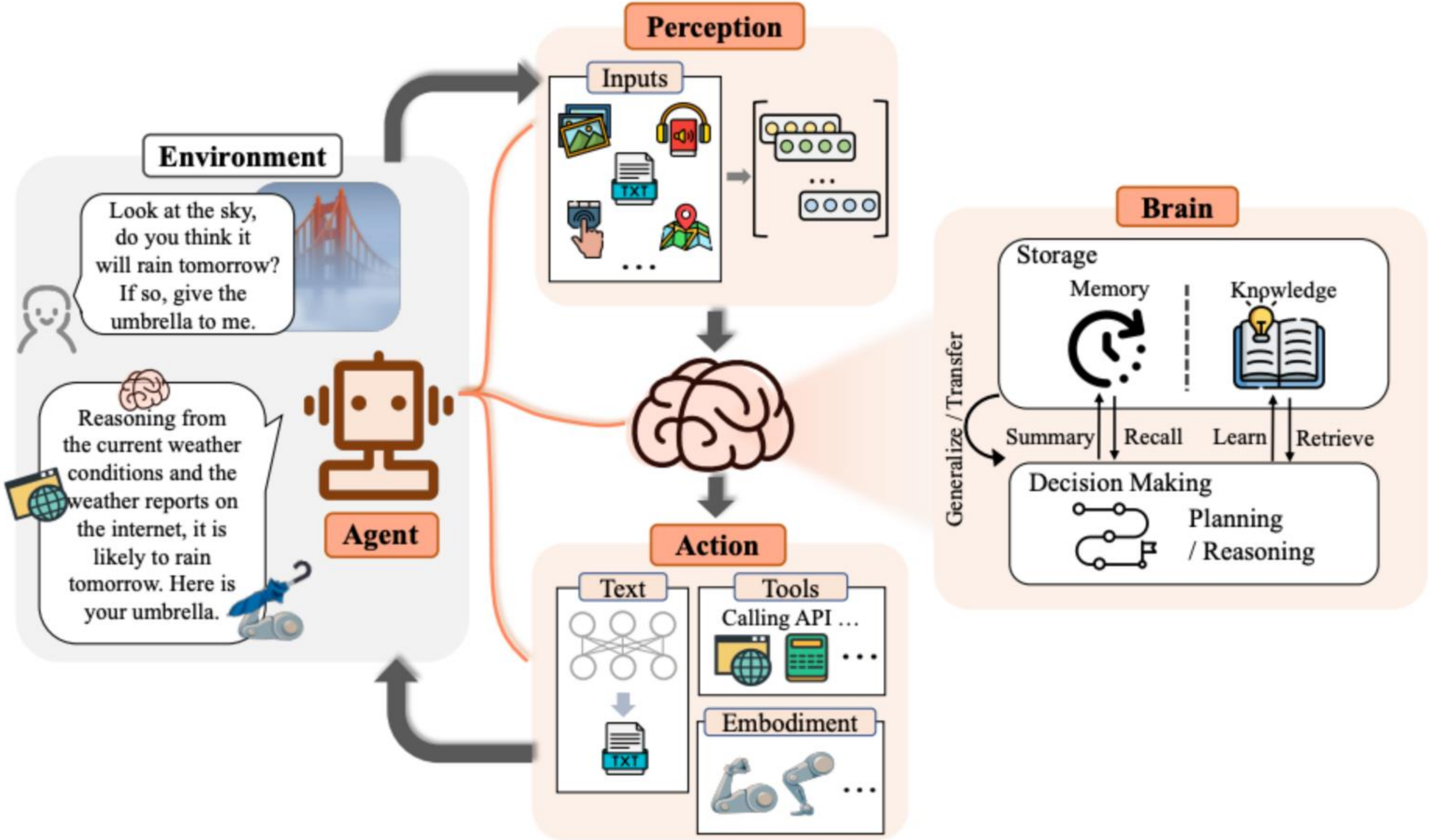


A Survey on Large Language Model based Autonomous Agents [Wang et al, 2023]

LLM Agent Framework



https://www.promptingguide.ai/research/llm-agents



The Rise and Potential of Large Language Model Based Agents: A Survey [Xi et al, 2023]

Planning

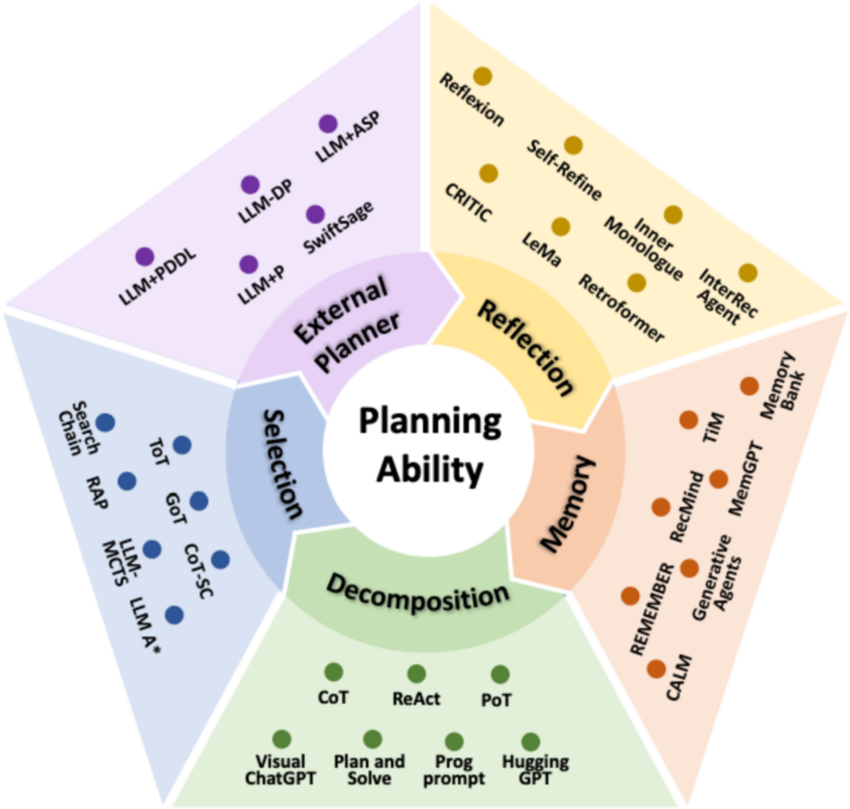
Planning

- What does the agent need to do to accomplish a specified goal?
- Low-level vs high-level planning
  - High-level plan: Identify subgoals for a long-horizon task
  - Low-level plan (sometime low-level control): Identify sequence of actions
- Traditionally use symbolic reasoning
  - Hard to recover from errors
  - Difficult to convert expert knowledge into planning languages such as PDDL (Planning Domain Definition Language)
- Use of LLM for planning
  - Lot of expert domain knowledge already encoded in language
  - Can we use LLMs to help us plan?

Taxonomy for planning with LLMs

Given environment and sequence of actions, and an overall task goal, identify subgoals and actions

- **Task decomposition** - figure out subgoals, do planning for subgoals if needed
- **Multi-plan selection** - generate multiple plans and then select one
- **External planner** - LLM used to formalized the problem which is passed to an external planner
- **Reflection and refinement** - After obtaining a plan, the LLM future reflects on the plan and refine it to fix any issues with the original plan
- **Memory-augmented planning** - Uses external memory to retrieve information (common sense knowledge, domain-specific knowledge, etc) and then determines plan based on that

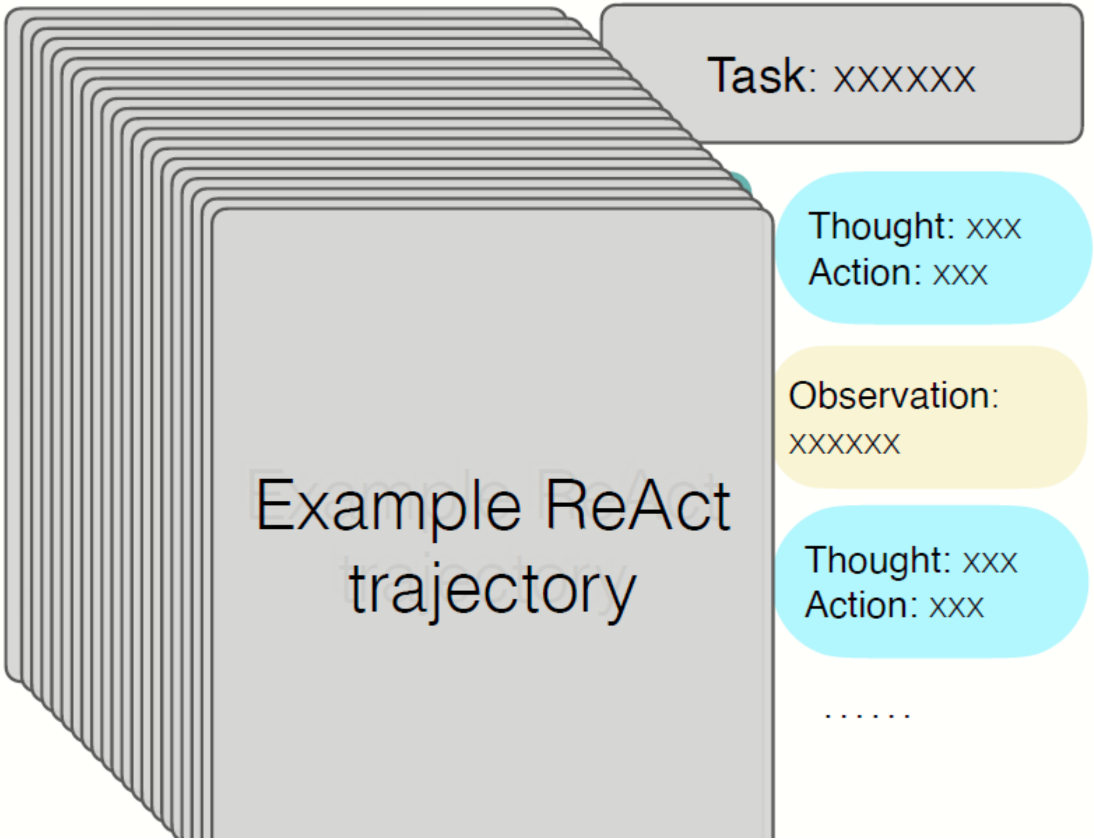


Understanding the planning of LLM agents: A survey [Huang et al, 2024]

ReAct is simple and intuitive to use

ReAct supports

- One-shot prompting
- Few-shot prompting
- Fine-tuning



Human prompt
LLM output
Environment feedback

Slide by Shunyu Yao

A zero-shot ReAct prompt

You are an agent that answers questions by using two actions:

- search[query]: Google search the query. You can also use it to calculate math.
- finish[answer]: return the answer

Your generation should have the following format:
Thought: to analyze the observation and inform the action.
Action: your action

Question: If I have seven trillion dollars today, can I buy Apple, Nvidia, and Microsoft? If not, how much more money do I need?

Slide by Shunyu Yao

Self-Reflection

- ReAct: interleave thought, action, and observation
- Prompting to combine reasoning with actions

(1) Hotspot QA

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

(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) ', …]

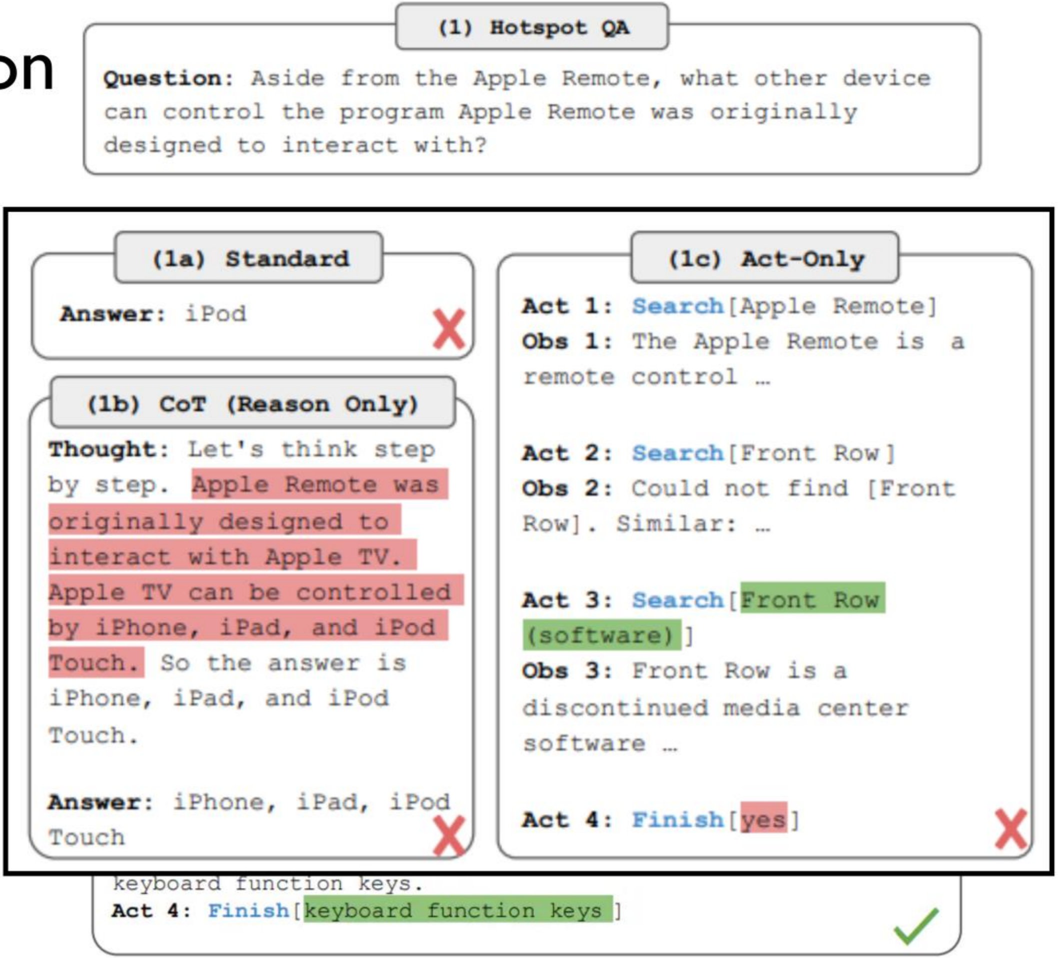
Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

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]

[ReAct: Synergizing Reasoning and Acting in Language Models](#) [Yao et al. 2022]

Self-Reflection

- Comparison with other ways of prompting



ReAct: Synergizing Reasoning and Acting in Language Models [Yao et al. 2022]

Memory

Types of memory

Memory

- Sensory memory
  few seconds
  - Iconic memory (visual)
  - Echoic memory (auditory)
  - Haptic memory (touch)

  Learn embeddings for raw inputs

- In-context learning
  Short-term memory (Working memory)
  20-30 seconds

- Long-term memory
  External vector store (fast retrieval for access)
  - Explicit / Declarative memory (conscious)
    Recalling events and facts
    - Episodic memory (life events)
    - Semantic memory (facts, concepts)
  - Implicit / Procedural memory (unconscious; skills)
    Example: Riding a bike

https://lilianweng.github.io/posts/2023-06-23-agent/

Different types of Approximate Nearest Neighbor

Vector transformation and encoding (focus on reduced memory)

- Quantization (FAISS, ScaNN)

Structures for searching (focus on fast search)

- Hashes (LSH)
- Trees (ANNOY)
- Graphs (HNSW)

Methods can be combined

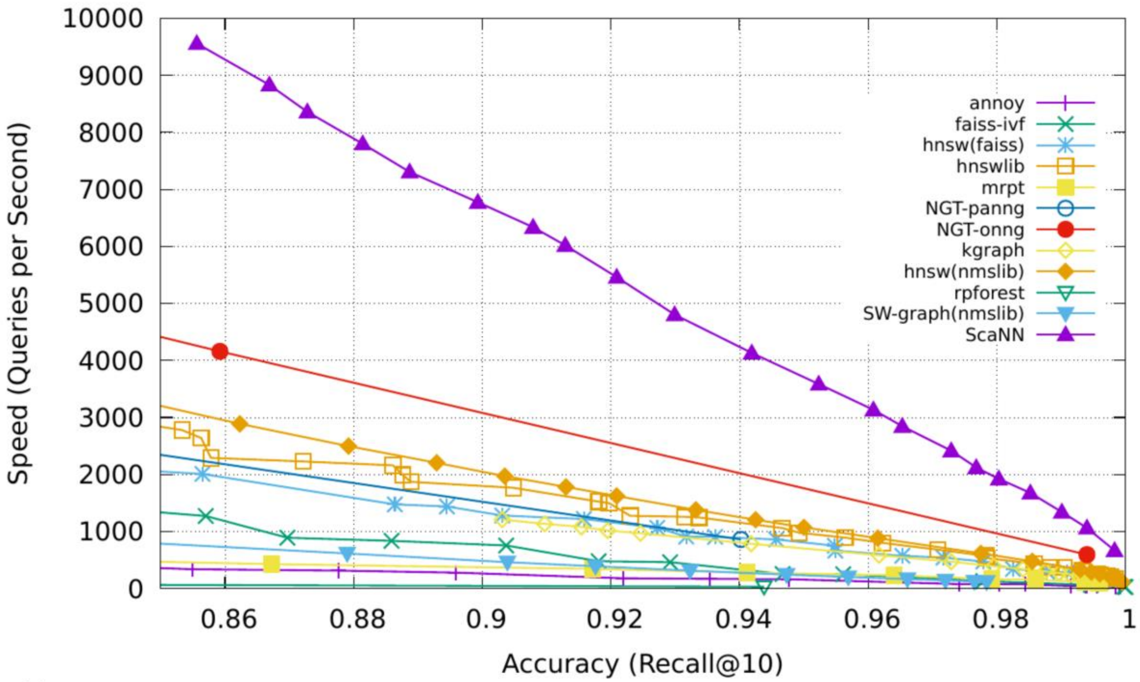
FAISS library: https://github.com/facebookresearch/faiss

Efficient retrieval from memory

Approximate nearest neighbours (ANN) algorithms to return top k nearest neighbours using maximum inner product search (MIPS)

- LSH (Locality Sensitive Hashing) - hashing function so that similar inputs are mapped to same buckets with high probability
- ANNOY (ANN Oh Yeah) - Random projection trees where nodes splits input space into half
- HNSW (Hierarchical Navigable Small World)
  - Hierarchical layers of small-world graphs (points in the bottom layers)
  - Can be used with FAISS
- FAISS (Facebook AI Similarity Search) - vector quantization - partition vector space into clusters
- ScaNN (Scalable Nearest Neighbours) - Anisotropic vector quantization (quantize points while maintaining distances)

https://lilianweng.github.io/posts/2023-06-23-agent/



https://blog.research.google/2020/07/announcing-scann-efficient-vector.html

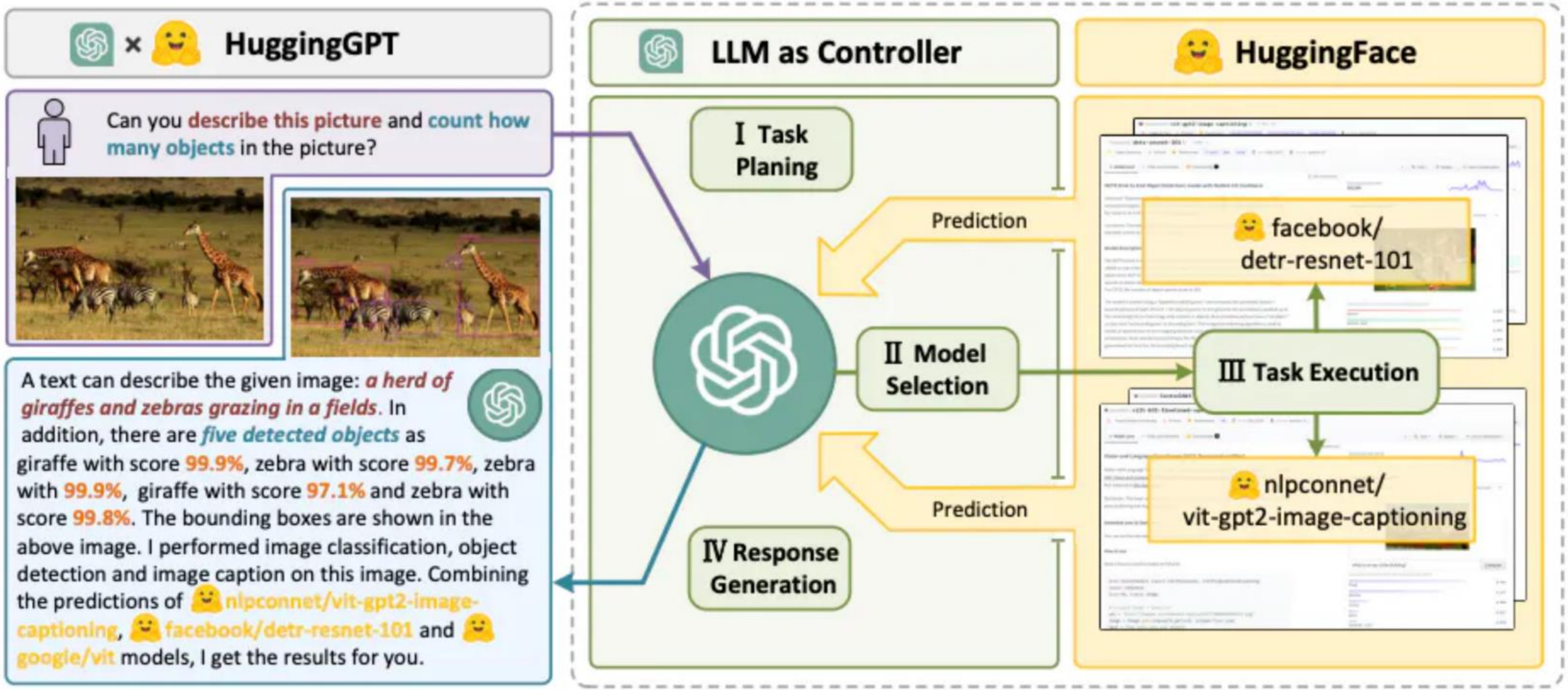
Note: nmslib focus is on non-metric spaces

Tool use

- API calls to external services (math calculator, currency converter, etc)
- Expert models that can be called

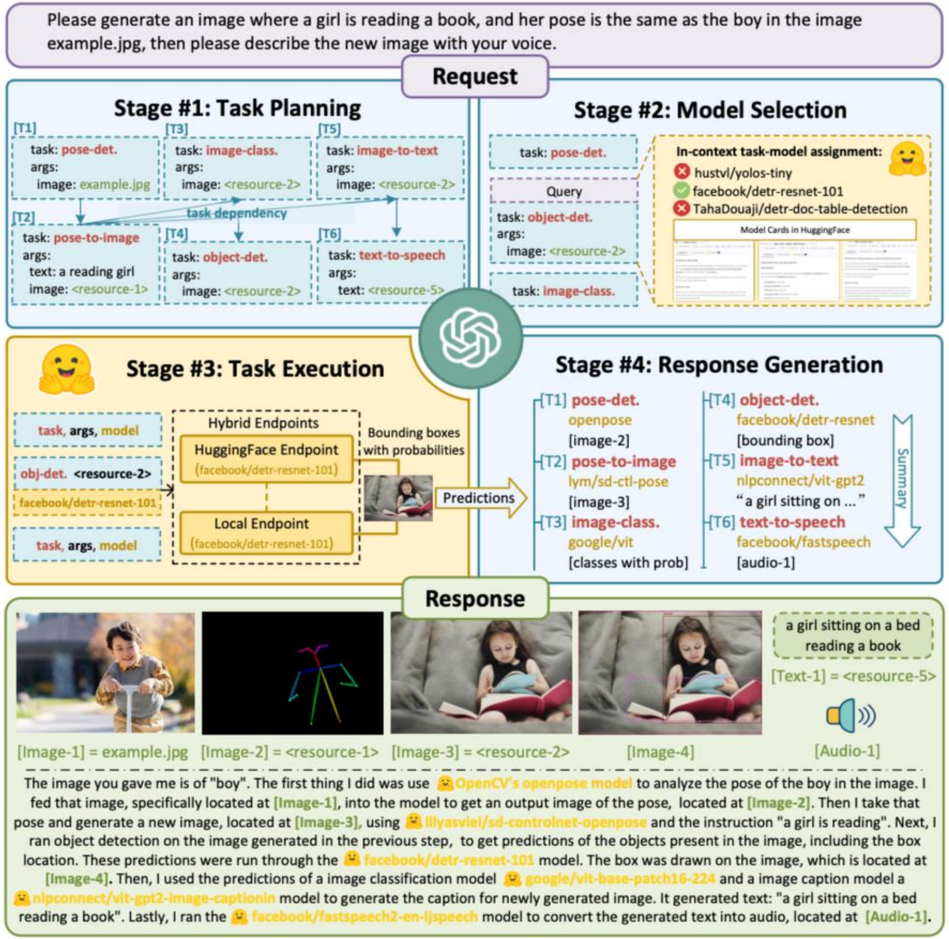
HuggingGPT: Task decomposition with model selection

1. Task planning
2. Model selection
3. Task execution
4. Response generation



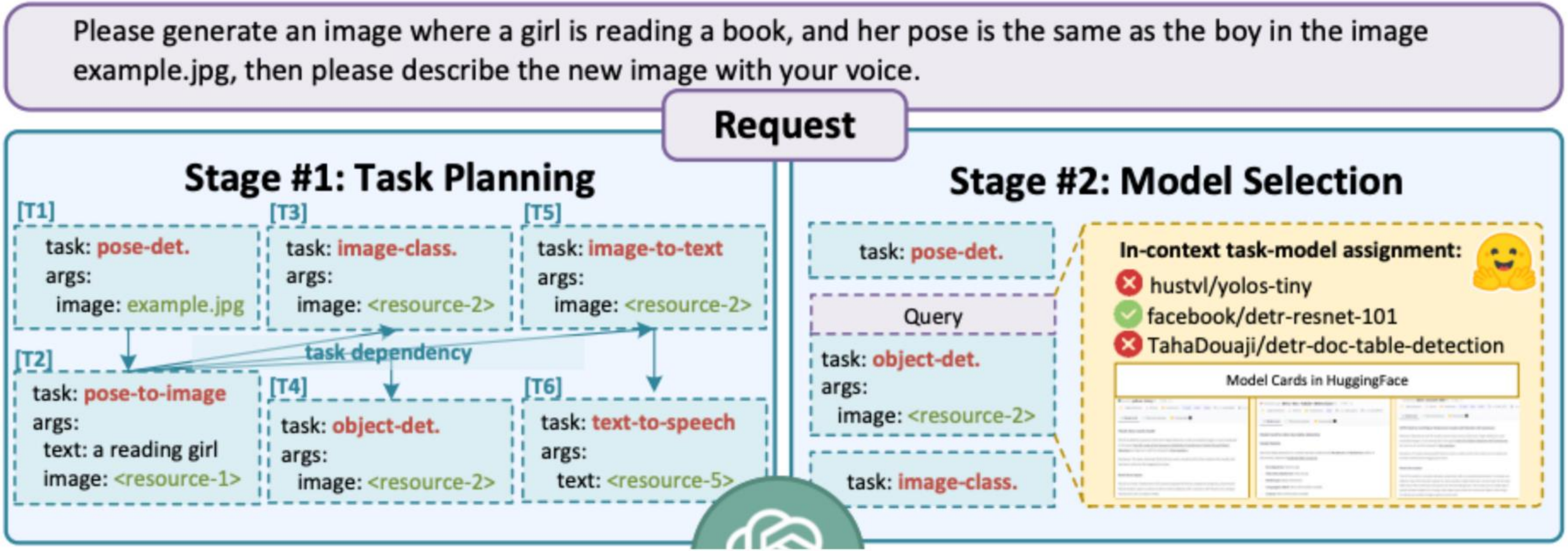
HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face [Shen et al, 2023]

HuggingGPT



[HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face [Shen et al, 2023]]

HuggingGPT



HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face [Shen et al, 2023]

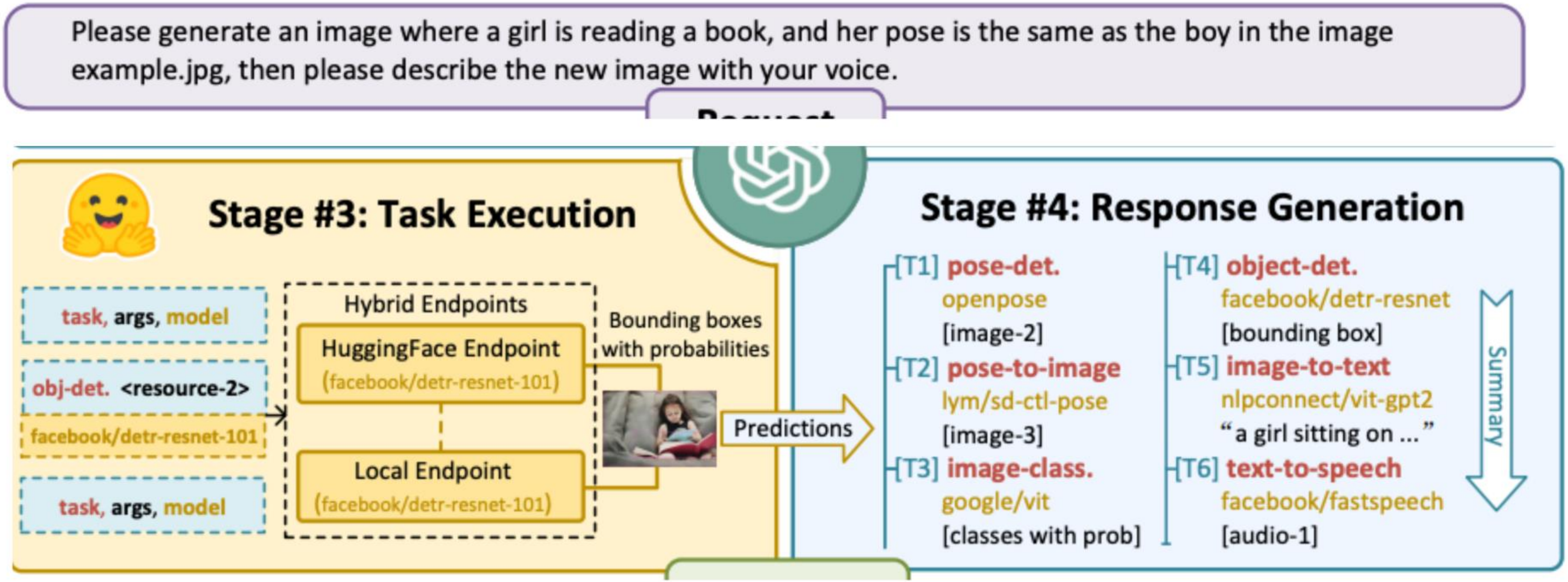
HuggingGPT

Task planning: figure out what task we want to solve, its id, **dependencies**, and **arguments** that are needed.

	Prompt	
Task Planning	#1 Task Planning Stage - The AI assistant performs task parsing on user input, generating a list of tasks with the following format: [{"task": task, "id", task_id, "dep": dependency_task_ids, "args": {"text": text, "image": URL, "audio": URL, "video": URL}}]. The "dep" field denotes the id of the previous task which generates a new resource upon which the current task relies. The tag "<resource>-task_id" represents the generated text, image, audio, or video from the dependency task with the corresponding task_id. The task must be selected from the following options: {{ *Available Task List* }}. Please note that there exists a logical connections and order between the tasks. In case the user input cannot be parsed, an empty JSON response should be provided. Here are several cases for your reference: {{ *Demonstrations* }}. To assist with task planning, the chat history is available as {{ *Chat Logs* }}, where you can trace the user-mentioned resources and incorporate them into the task planning stage.	
	Demonstrations	
	Can you tell me how many objects in e1.jpg?	[{"task": "object-detection", "id": 0, "dep": [-1], "args": {"image": "e1.jpg" }}]
	In e2.jpg, what's the animal and what's it doing?	[{"task": "image-to-text", "id": 0, "dep":[-1], "args": {"image": "e2.jpg" }}, {"task":"image-cls", "id": 1, "dep": [-1], "args": {"image": "e2.jpg" }}, {"task":"object-detection", "id": 2, "dep": [-1], "args": {"image": "e2.jpg" }}, {"task": "visual-quesrion-answering", "id": 3, "dep":[-1], "args": {"text": "what's the animal doing?", "image": "e2.jpg" }}]
	First generate a HED image of e3.jpg, then based on the HED image and a text "a girl reading a book", create a new image as a response.	[{"task": "pose-detection", "id": 0, "dep": [-1], "args": {"image": "e3.jpg" }}, {"task": "pose-text-to-image", "id": 1, "dep": [0], "args": {"text": "a girl reading a book", "image": "<resource>-0" }}]

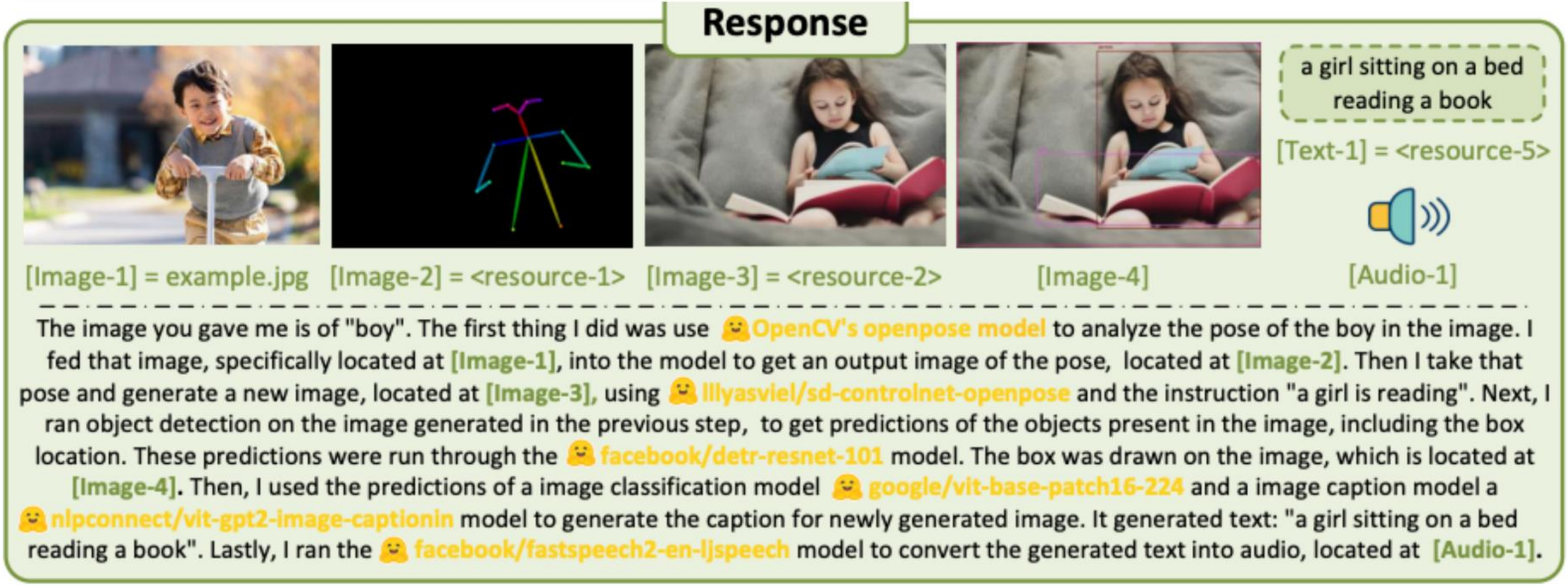
HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face [Shen et al, 2023]

HuggingGPT



HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face [Shen et al, 2023]

HuggingGPT



HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face [Shen et al, 2023]

HuggingGPT

Model selection: figure out what model to invoke

Model Selection	Prompt
	#2 Model Selection Stage - Given the user request and the call command, the AI assistant helps the user to select a suitable model from a list of models to process the user request. The AI assistant merely outputs the model id of the most appropriate model. The output must be in a strict JSON format: {`"id"`: "id", `"reason"`: "your detail reason for the choice"}. We have a list of models for you to choose from {{ *Candidate Models* }}. Please select one model from the list.
	Candidate Models
	{"model_id": model id #1, "metadata": meta-info #1, "description": description of model #1}
	{"model_id": model id #2, "metadata": meta-info #2, "description": description of model #2}
	... ...
	{"model_id": model id #K, "metadata": meta-info #K, "description": description of model #K}

HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face [Shen et al, 2023]

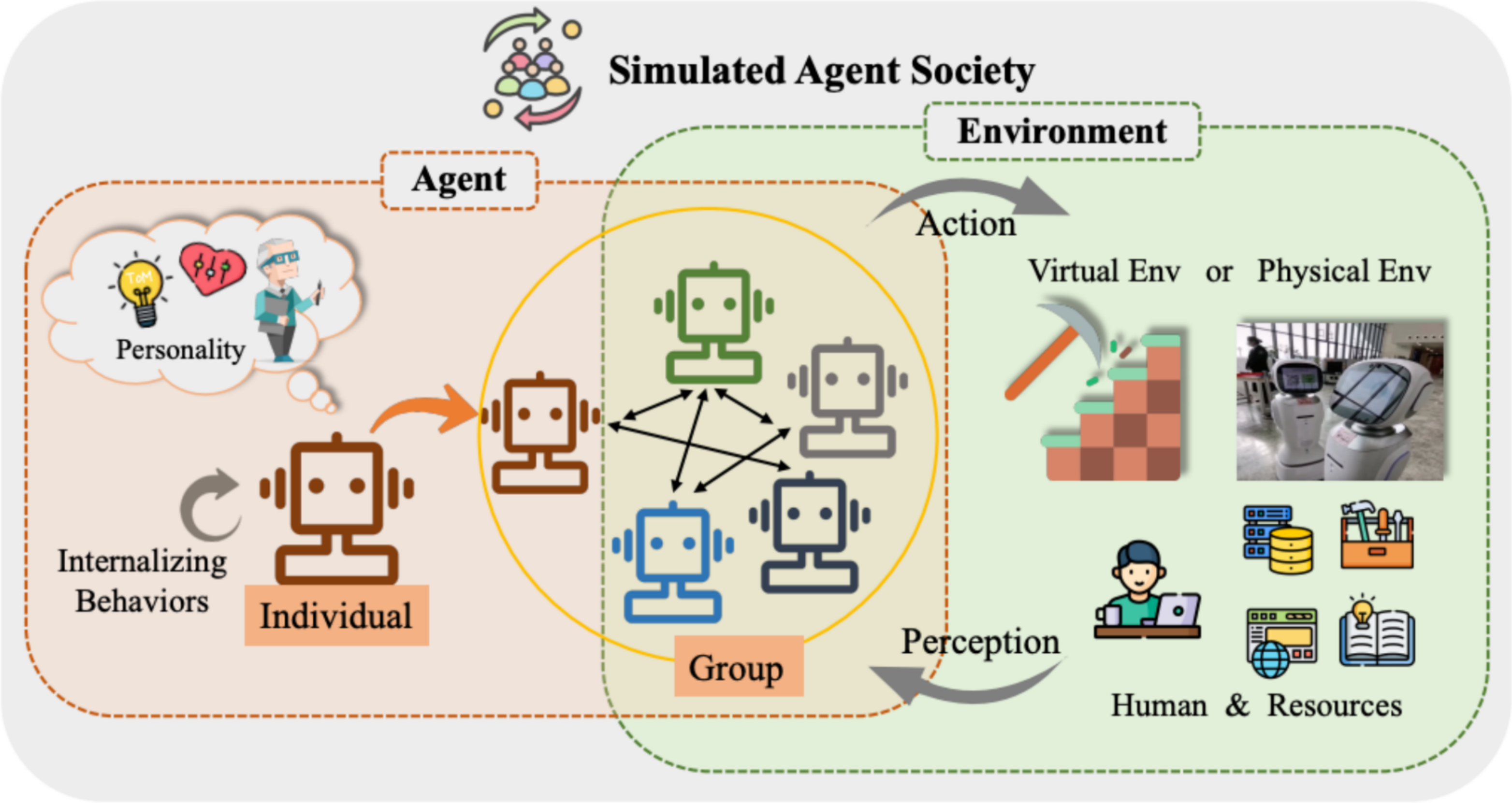
HuggingGPT

Response generation: respond to user the process and results

Response Generation	Prompt
	#4 Response Generation Stage - With the input and the inference results, the AI assistant needs to describe the process and results. The previous stages can be formed as - User Input: {{ *User Input* }}, Task Planning: {{ *Tasks* }}, Model Selection: {{ *Model Assignment* }}, Task Execution: {{ *Predictions* }}. You must first answer the user's request in a straightforward manner. Then describe the task process and show your analysis and model inference results to the user in the first person. If inference results contain a file path, must tell the user the complete file path. If there is nothing in the results, please tell me you can't make it.

HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face [Shen et al, 2023]

Virtual worlds



The Rise and Potential of Large Language Model Based Agents: A Survey [Xi et al, 2023]

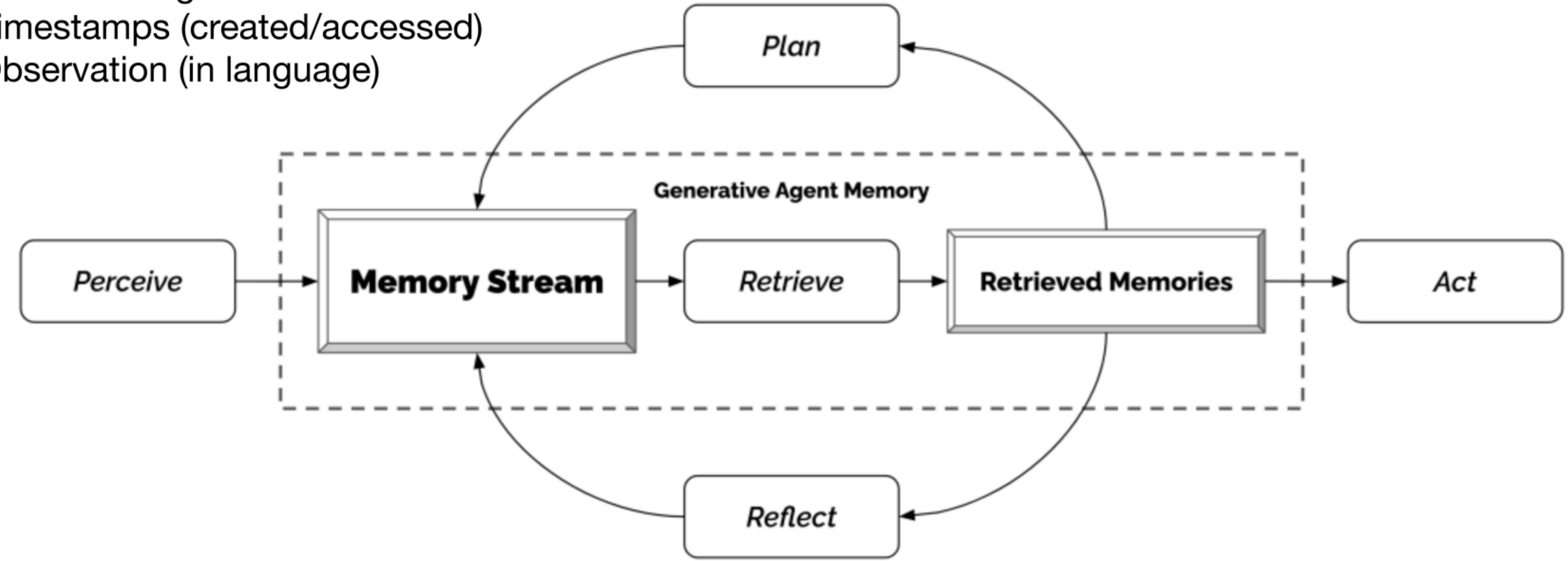
Generative Agents

Generative Agents: Interactive Simulacra of Human Behavior [Park et al, 2023]

Generative Agents

Memory Stream

- List of memory objects
- Each containing:
  - Timestamps (created/accessed)
  - Observation (in language)

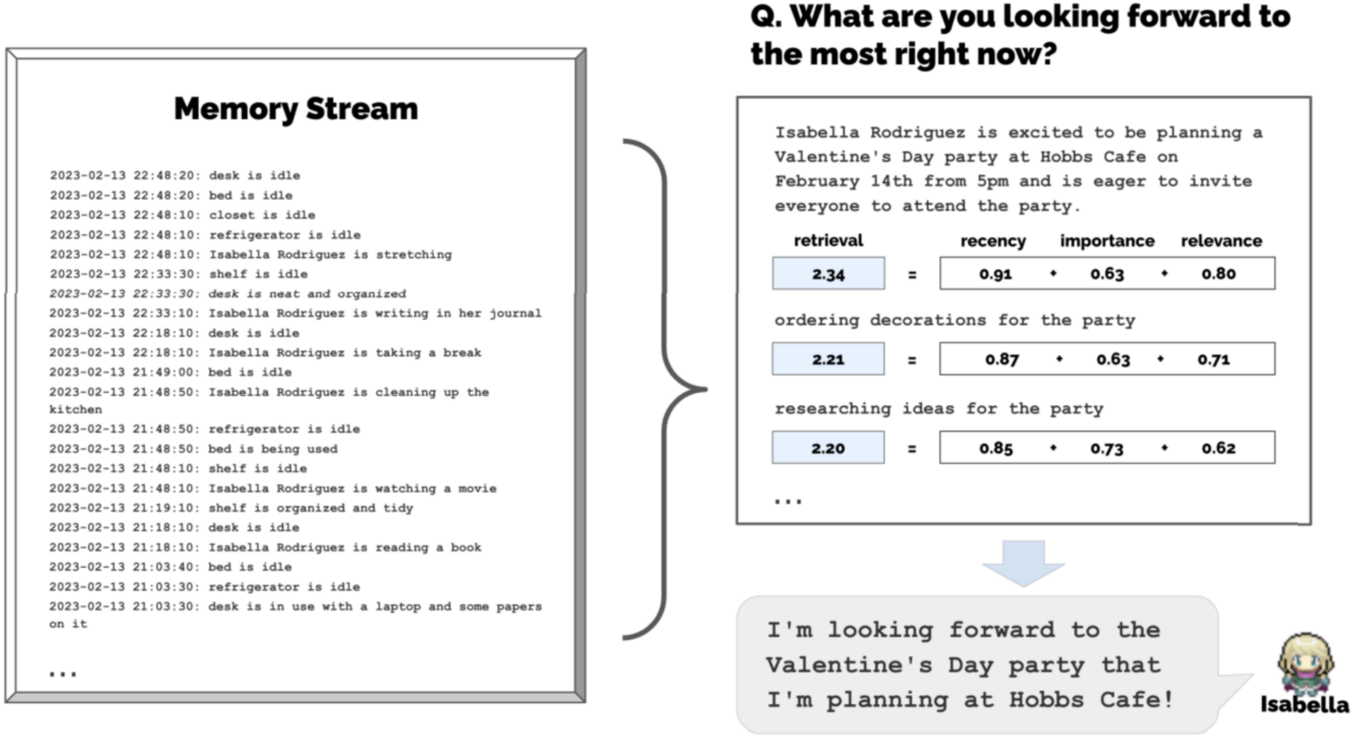


Generative Agents: Interactive Simulacra of Human Behavior [Park et al, 2023]

Generative Agents

Retrieval extracts subset of memories based on:

Recency: exponential decay
Importance: ask LLM to output score
Relevance: cosine similarity of memory and query embeddings



Generative Agents: Interactive Simulacra of Human Behavior [Park et al, 2023]

Generative Agents

Reflection: additional memory that is synthesized from previous memories (generated periodically)

- Prompt LLM with 100 most recent observations
- Use to to generate 3 questions from which relevant memories are extracted
- LLM then prompted to extract insights from the memories

```
Statements about Klaus Mueller
1. Klaus Mueller is writing a research paper
2. Klaus Mueller enjoys reading a book
on gentrification
3. Klaus Mueller is conversing with Ayesha Khan
about exercising [...]
What 5 high-level insights can you infer from
the above statements? (example format: insight
(because of 1, 5, 3))
```

Generative Agents: Interactive Simulacra of Human Behavior [Park et al, 2023]

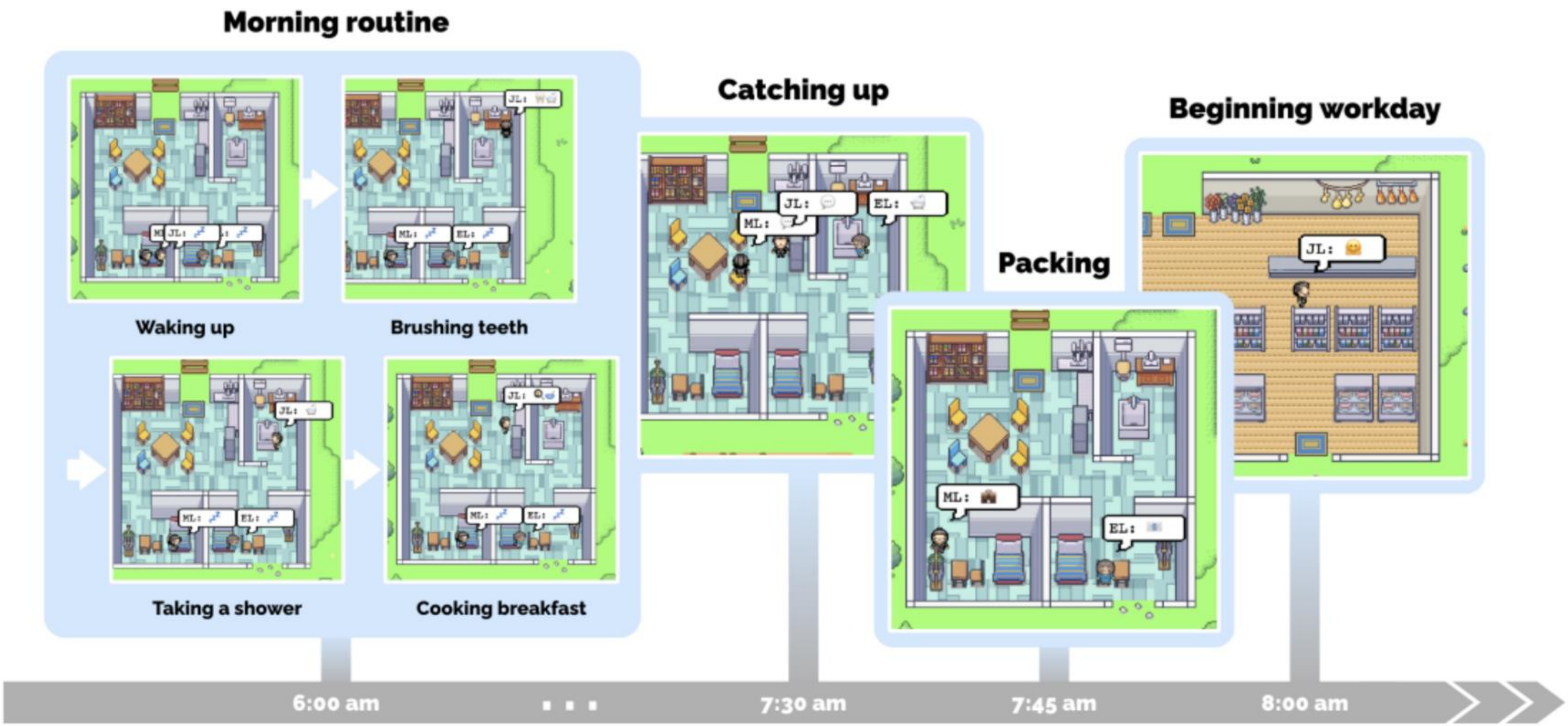
Generative Agents

Planning and reacting: converts memories and observations into actions

- Generates rough-plan from agent’s summary description and summary of previous day and has LLM complete is
- Converse as they interact with each other (conditioned on memories about each other)
- LLM then prompted to extract insights from the memories

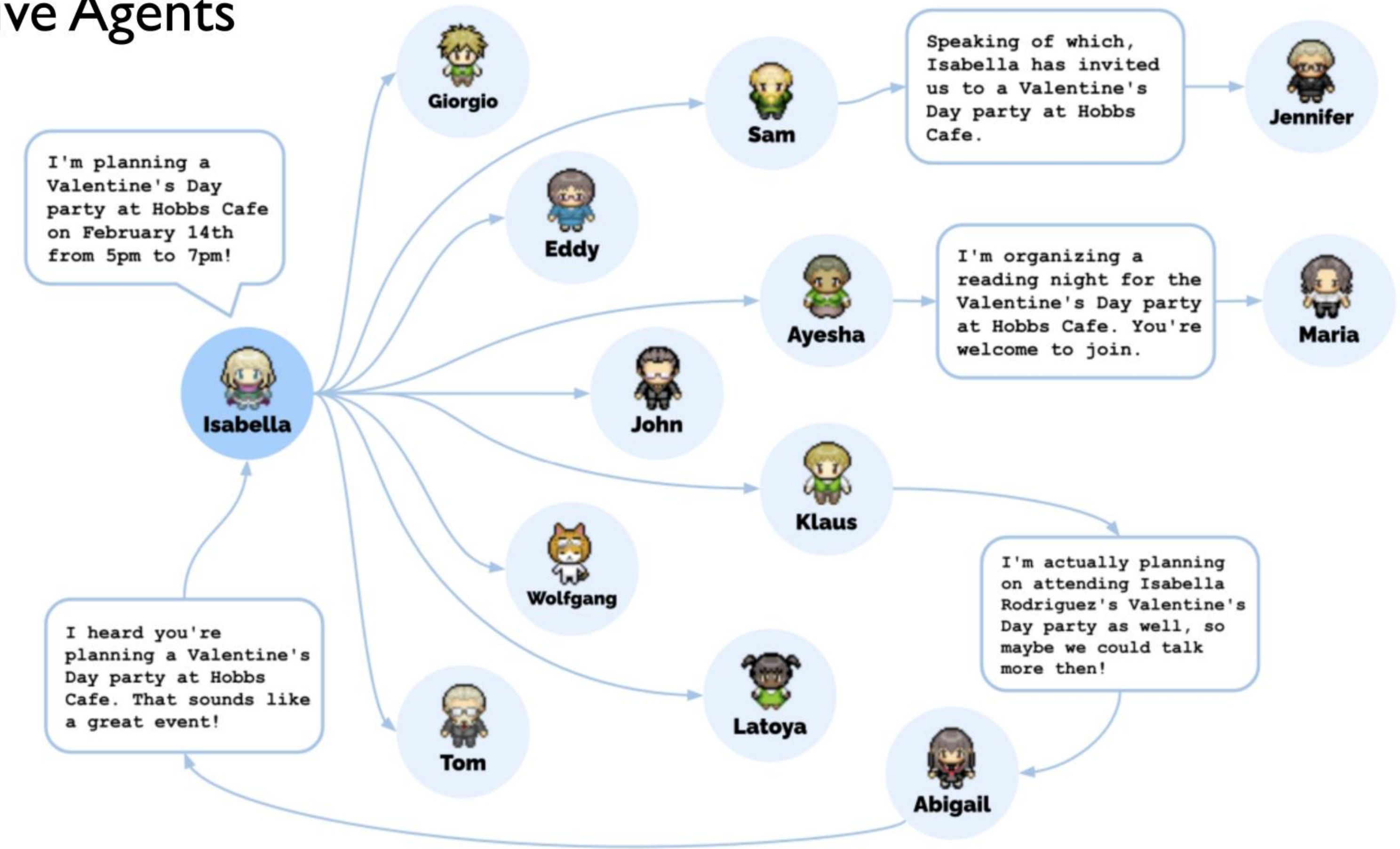
Generative Agents: Interactive Simulacra of Human Behavior [Park et al, 2023]

Generative Agents



Generative Agents: Interactive Simulacra of Human Behavior [Park et al, 2023]

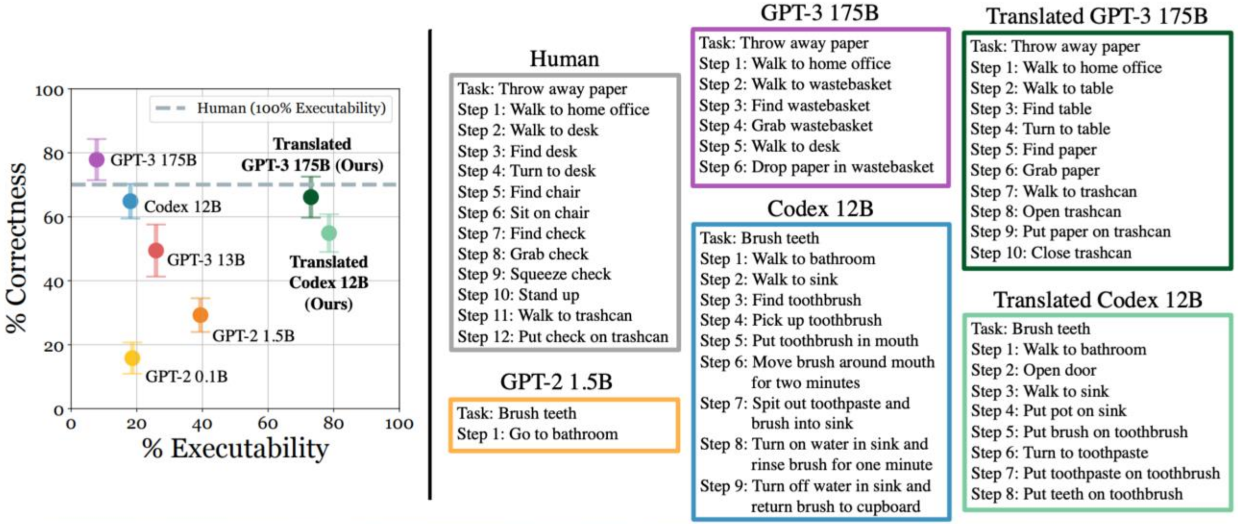
Generative Agents



Generative Agents: Interactive Simulacra of Human Behavior [Park et al, 2023]

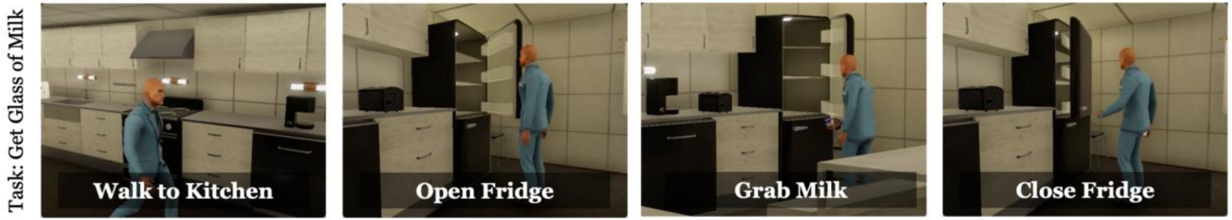
Agents in more realistic scenarios (robotics and embodied AI)

Planning via large language models



Throw away paper

Brush teeth

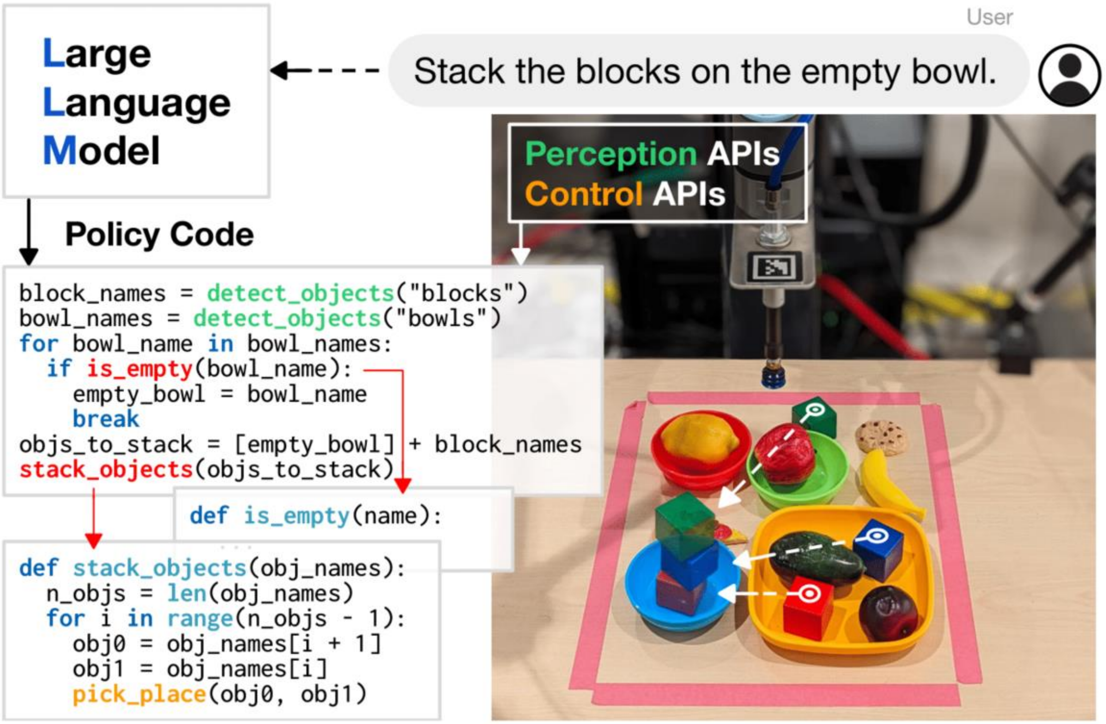


Get Glass of Milk

Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents [Huang et al. ICML 2022]

Berkeley UNIVERSITY OF CALIFORNIA   CMU   https://wenlong.page/language-planner/   Google Brain

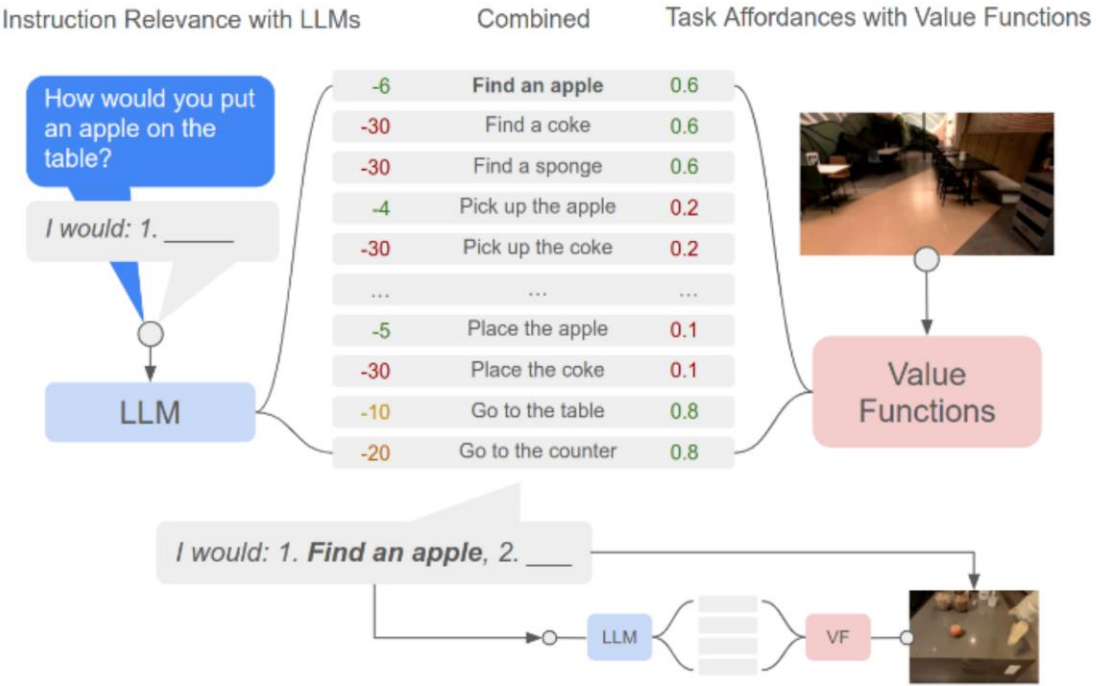
Control by code generation using LLMs



Code as Policies: Language Model Programs for Embodied Control [Liang et al. 2022]
https://code-as-policies.github.io/

Google

Combining perception with planning



Use perception to determine what is possible

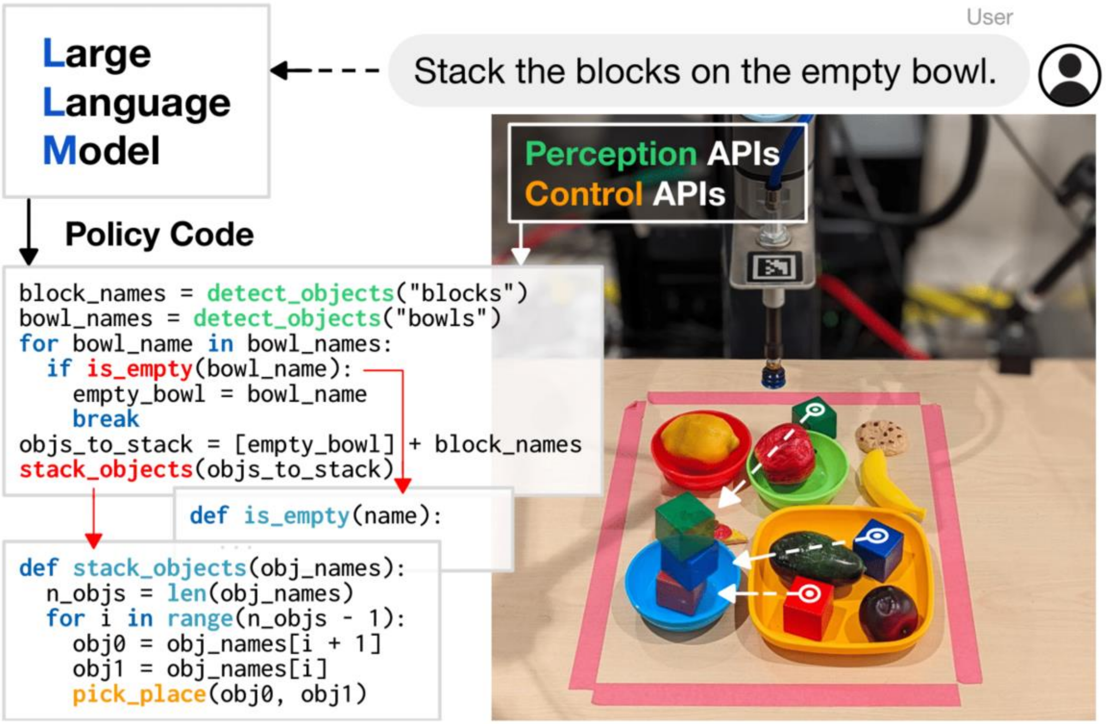
Do As I Can, Not As I Say: Grounding Language in Robotic Affordances [Ahn et al. CORL 2022]

https://say-can.github.io/

Everyday Robots

Google

Control by code generation using LLMs



Code as Policies: Language Model Programs for Embodied Control [Liang et al. 2022]
https://code-as-policies.github.io/

Practical applications

Web browsing

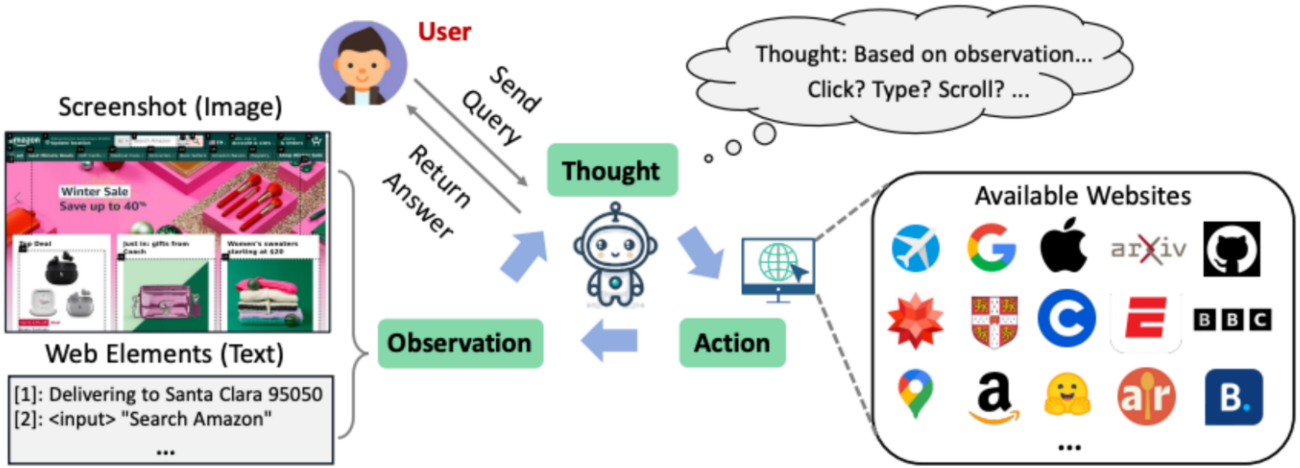


Figure 1: The overall workflow of WebVoyager. WebVoyager takes web tasks assigned by a human and automatically browses the web online. At each step, WebVoyager selects actions based on screenshots and text (the 'type' of the web element and its contents). Once the task is completed, the answers will be returned to the user. For example, for a user query: "Find the cost of a 2-year protection for PS4 on Amazon.", the agent interacts with Amazon online, locates the PS4, identifies the 2-year protection price, and returns "$30.99" to the user.

WebVoyager : Building an End-to-End Web Agent with Large Multimodal Models [He et. al. 2024]

Web browsing

- Browse web to find information (answer to question)
- Action: mouse + common keyboard actions
- Automated browsing of open web using Selenium
- Challenges: ads, popup-windows, constant updates

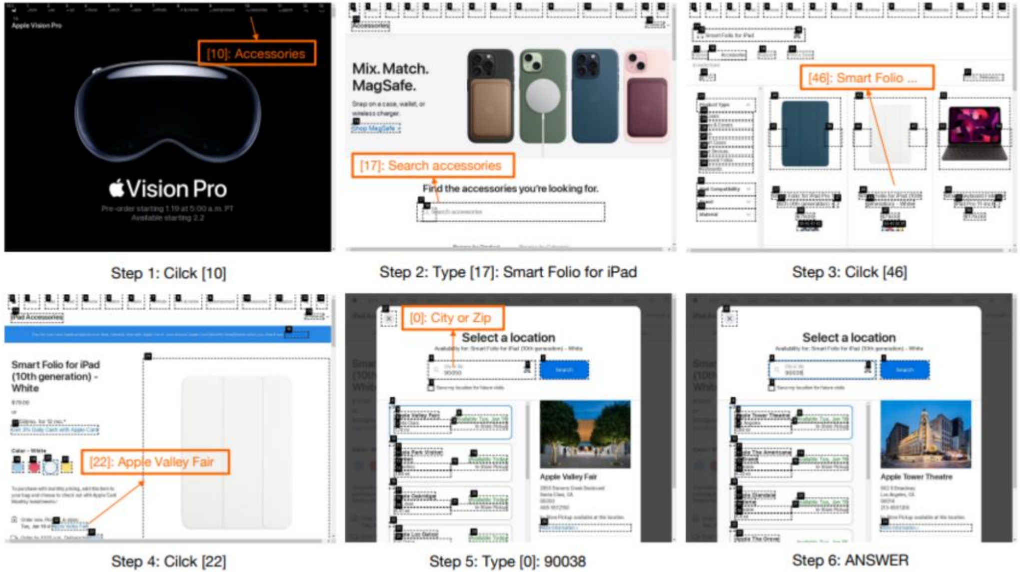


Figure 4: Screenshots of a complete trajectory of online web browsing. Given the task: 'Search Apple for the accessory Smart Folio for iPad and check the closest pickup availability next to zip code 90038.' The agent interacts with the Apple website and obtains the answer: 'Apple Tower Theatre.'

WebVoyager : Building an End-to-End Web Agent with Large Multimodal Models [He et. al. 2024]

Web browsing

Evaluate task success rate on
- New data set created using self-instruct
- 90 tasks from GAIA dataset (level 1 and 2)
- 50 tasks from SeeAct

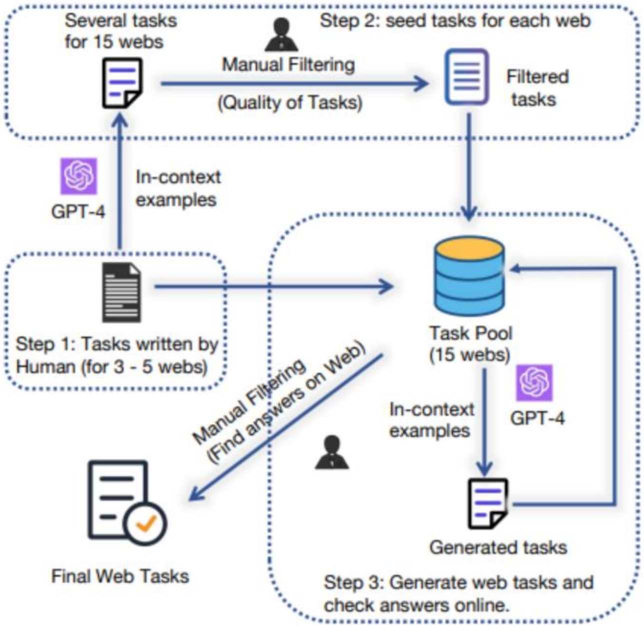


Figure 3: Data creation process using self-instruct.

WebVoyager : Building an End-to-End Web Agent with Large Multimodal Models [He et. al. 2024]

GAIA Benchmark

Question answering tasks inspired by real world
- straightforward but tedious for humans
- finding / transforming information from different data sources (including attached documents)

Requires:
- reasoning
- multi-modal understanding
- tool use

Levels
- 1: 0-1 tool, <= 5 steps
- 2: 5-10 tools, more steps
- 3: general assistant

Level 1

Question: What was the actual enrollment count of the clinical trial on H. pylori in acne vulgaris patients from Jan-May 2018 as listed on the NIH website?
Ground truth: 90

Level 2

Question: If this whole pint is made up of ice cream, how many percent above or below the US federal standards for butterfat content is it when using the standards as reported by Wikipedia in 2020? Answer as + or - a number rounded to one decimal place.
Ground truth: +4.6

Level 3

Question: In NASA's Astronomy Picture of the Day on 2006 January 21, two astronauts are visible, with one appearing much smaller than the other. As of August 2023, out of the astronauts in the NASA Astronaut Group that the smaller astronaut was a member of, which one spent the least time in space, and how many minutes did he spend in space, rounded to the nearest minute? Exclude any astronauts who did not spend any time in space. Give the last name of the astronaut, separated from the number of minutes by a semicolon. Use commas as thousands separators in the number of minutes.
Ground truth: White; 5876

Figure 1 Sample GAIA questions. Completing the tasks requires fundamental abilities such as reasoning, multi-modality handling, or tool use proficiency. Answers are unambiguous and by design unlikely to be found in plain text in training data. Some questions come with additional evidence, such as images, reflecting real use cases and allowing better control on the questions.

GAIA: A Benchmark for General AI Assistants [Mialon et. al. 2023]

Software engineering

https://github.com/SWE-agent/SWE-agent

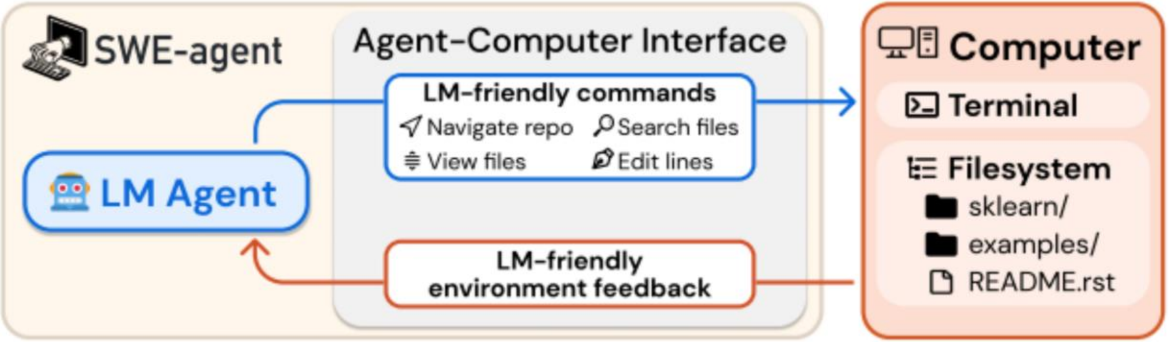


Figure 1: SWE-agent is an LM interacting with a computer through an agent-computer interface (ACI), which includes the commands the agent uses and the format of the feedback from the computer.

SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering [Yang et. al. 2024]

Software engineering

https://github.com/SWE-agent/SWE-agent

Observation (File Viewer)

```
[File: solvers/diophantine.py (3093 lines total)]
(400 lines above)
401:    elif total_degree == 3:
402:        diop_type = "cubic_thue"
403:
404:    elif (total_degree > 3):
405:        if all(coeff[k] == 1):
406:            diop_type = 'general_sum_even'
407:
408:    if diop_type is not None:
409:        return var, coeff, diop_type
(2684 lines below)
```

(a) Observation from the file viewer.

Response (Thought + Edit Action)

We need to add a check to ensure that all powers are the same. Let's edit the code to include this check.

```
edit 404:407
    elif (total_degree > 3 and
          len(set(k.exp[k.is_Pow])) == 1):
        if all(coeff[k] == 1):
            diop_type = 'general_sum_even'
end_of_edit
```

(b) Action using the edit interface.

Figure 3: The file viewer and edit command are integrated. (a) The file viewer shows the agent the open file's content with line numbers. (b) The agent invokes the edit function to replace lines 404-407 in the open file. After the edit, the file viewer shows the agent the now updated version of the file.

SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering [Yang et. al. 2024]

Software engineering

https://github.com/All-Hands-AI/OpenHands

- Docker sandbox
- API server to execute actions
  - Bash shell
  - Jupyter Python server
  - Chromium browser
- Extensible
  - Agent skills: add tools
  - Agent hub: community contributed agents

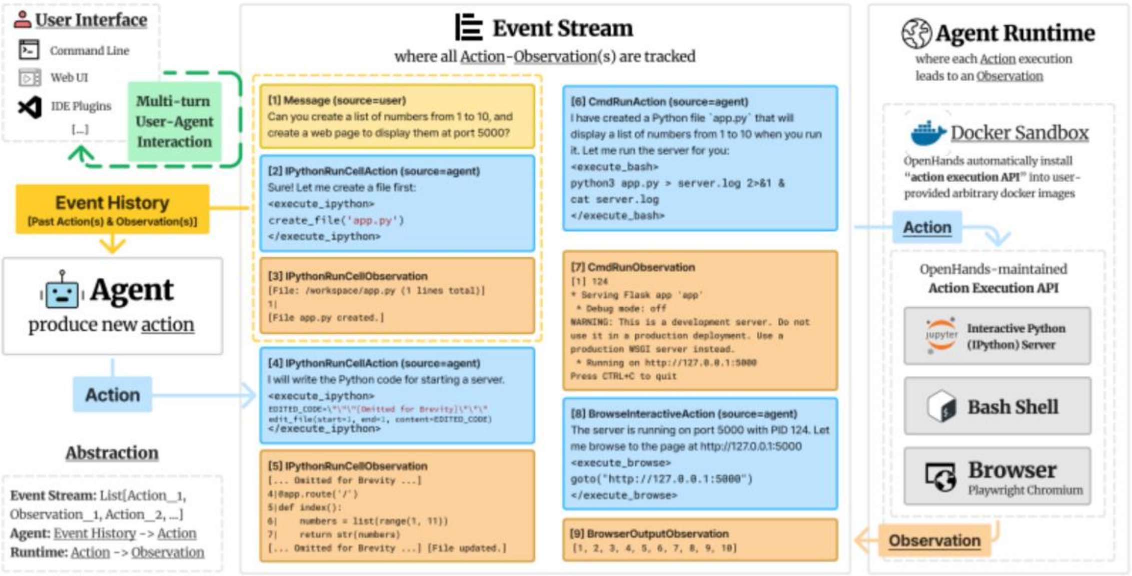


Figure 2: OpenHands consists of 3 main components: 1) **Agent abstraction** where community can contribute different implementation of agents (§2.1) into agenthub (§3); 2) **Event stream** for tracking history of actions and observations; 3) **Runtime** to execute all actions into observations (§2.2).

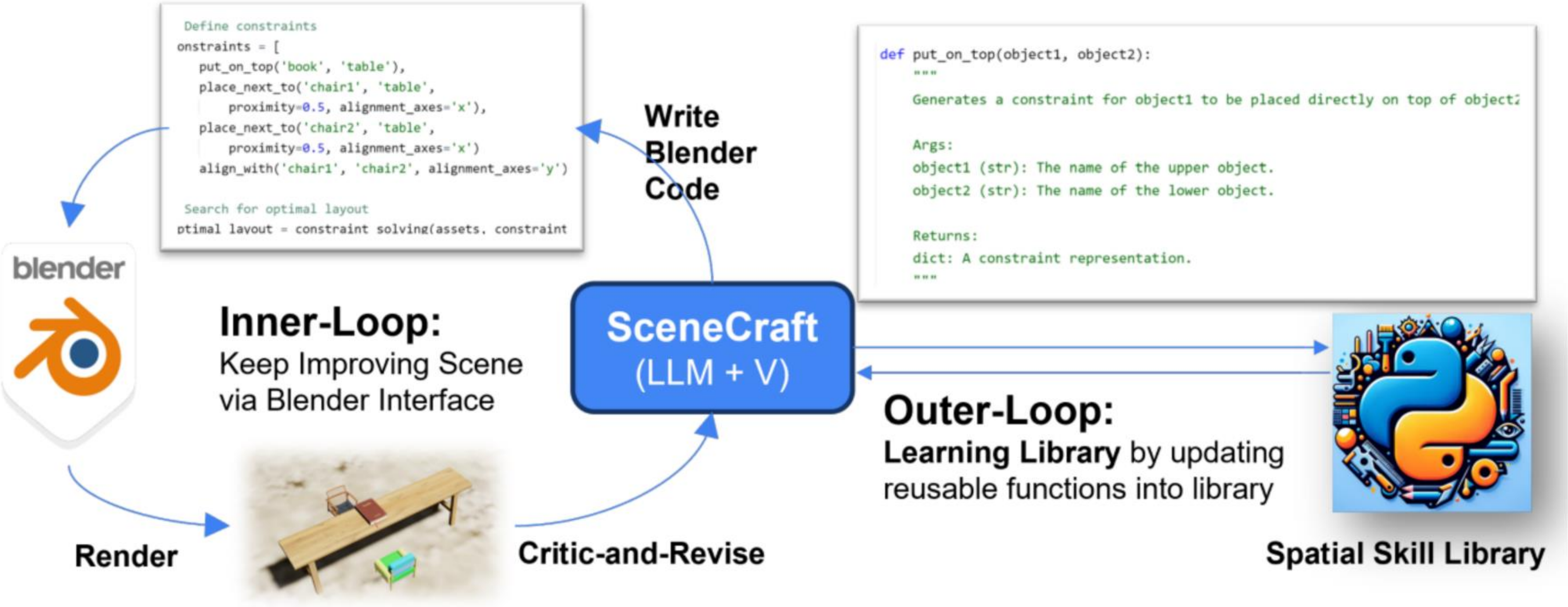
OpenHands: An Open Platform for AI Software Developers as Generalist Agents [Wang et. al. 2024]

Development Agents

- For coding (e.g. SWE-Agent, Aider)
- For broader development (e.g. Devin, OpenHands)

YouTube lecture by Graham Neubig

Scene Generation



SceneCraft: An LLM Agent for Synthesizing 3D Scene as Blender Code [Hu et al, 2024]

Scene Generation

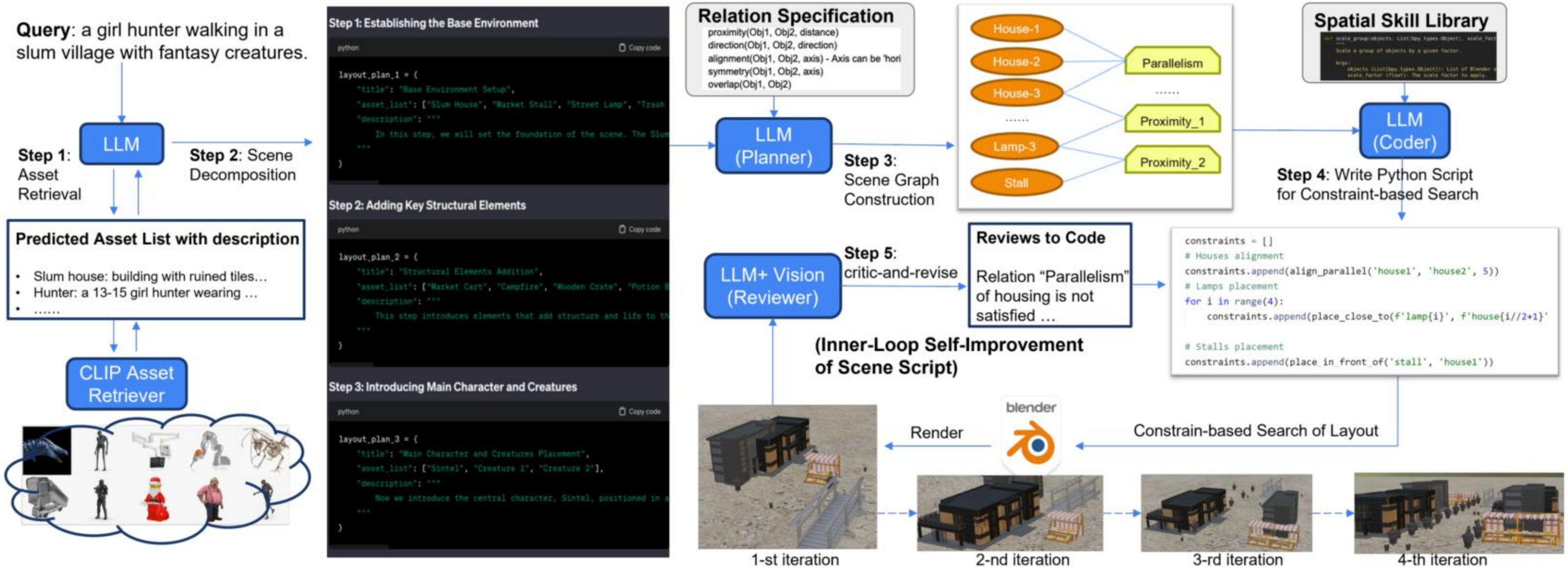
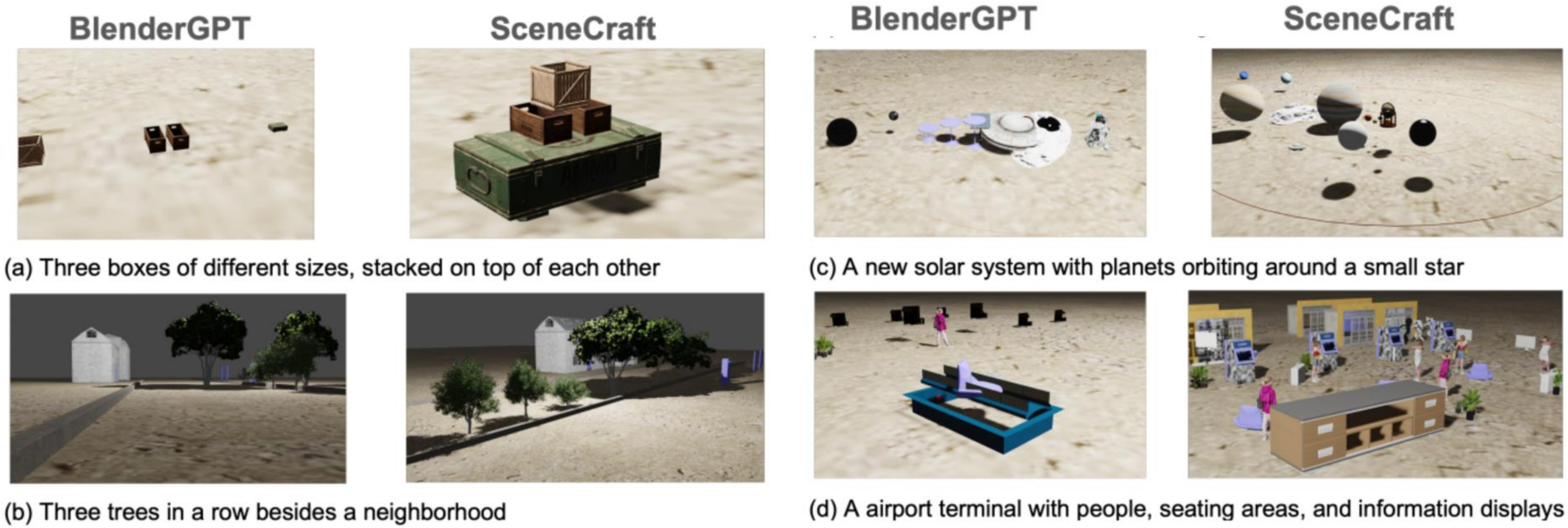


Figure 3. **The workflow of SceneCraft's inner-loop improvement of each scene.** 1) given query, a LLM writes a list of assets descriptions, then use CLIP retriever to fetch assets; 2) then LLM decomposes the full query into a sequence of sub-scene, each associated with a subset of assets and a text description; 3) a LLM-Planner generate a relational graph linking assets to spatial relationship; 4) Based on the graph, LLM-Coder writes python codes to get a list of numerical constraints, which can be executed to search optimal layout, and render into image using Blender; 5) LLM-Reviewer with vision perception capability criticize the rendered image, and update the script accordingly. This critic-and-revise procedure can be done multiple times to iteratively improve the script and scene.

SceneCraft: An LLM Agent for Synthesizing 3D Scene as Blender Code [Hu et al, 2024]

Scene Generation



SceneCraft: An LLM Agent for Synthesizing 3D Scene as Blender Code [Hu et al, 2024]

LLM Agents for Research

Agent Laboratory

https://agentlaboratory.github.io/

- AI agents for ML research
- Collaborate with humans
- Humans focus on ideation and critical thinking
- AI agent help automate repetitive and time-intensive tasks

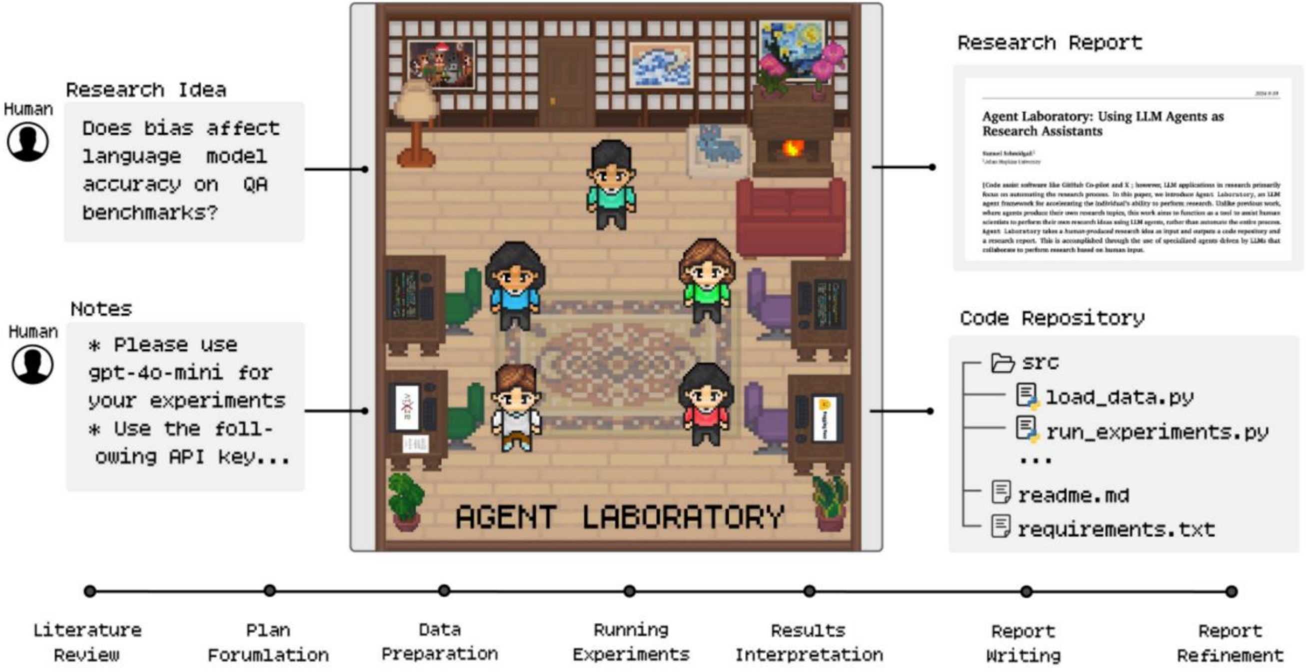


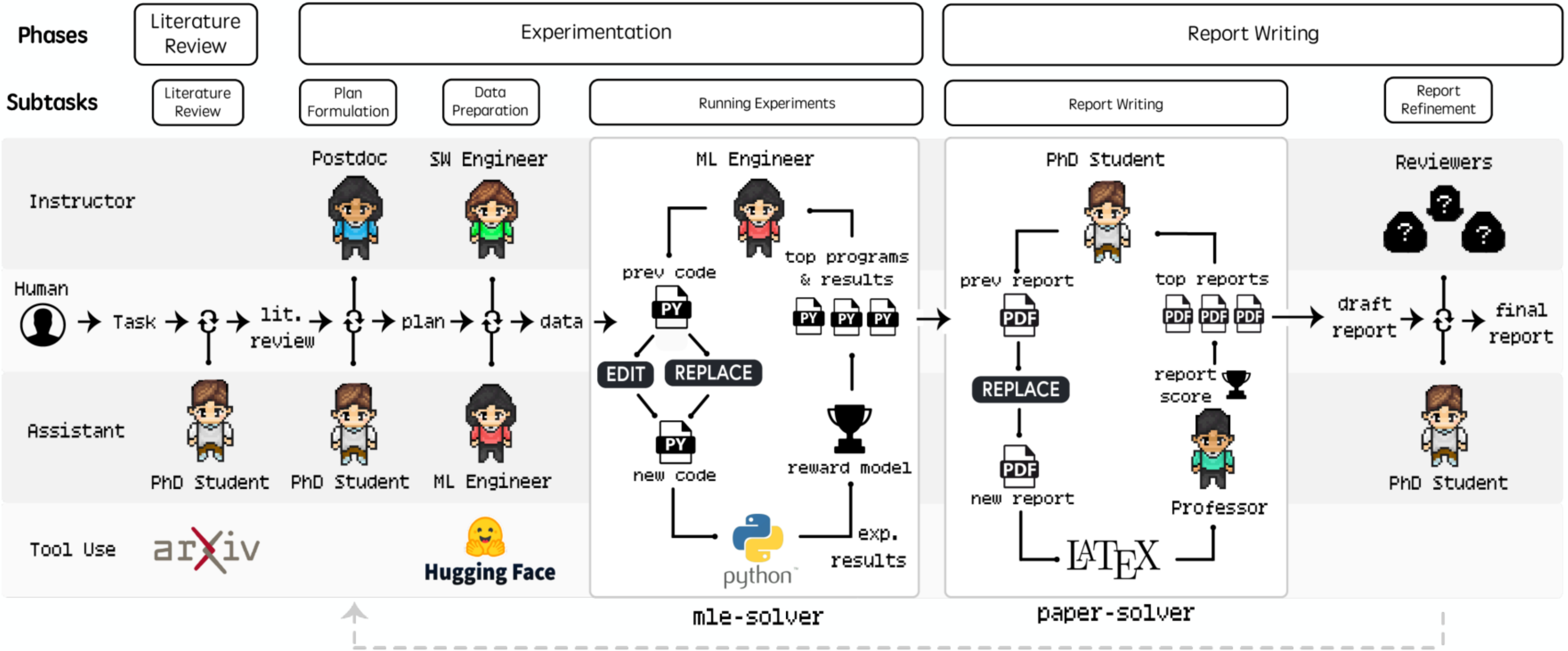
Figure 1 | Agent Laboratory takes as input a human research idea and a set of notes, provides this to a pipeline of specialized LLM-driven agents, and produces a research report and code repository.

Agent Laboratory: Using LLM Agents as Research Assistants [Schmidgall et al, 2025]

Agent Laboratory

https://agentlaboratory.github.io/

- AI agent help with 1) **Literature review**, 2) **Experimentation**, and 3) **Report Writing**



Agent Laboratory: Using LLM Agents as Research Assistants [Schmidgall et al, 2025]

Agent Laboratory

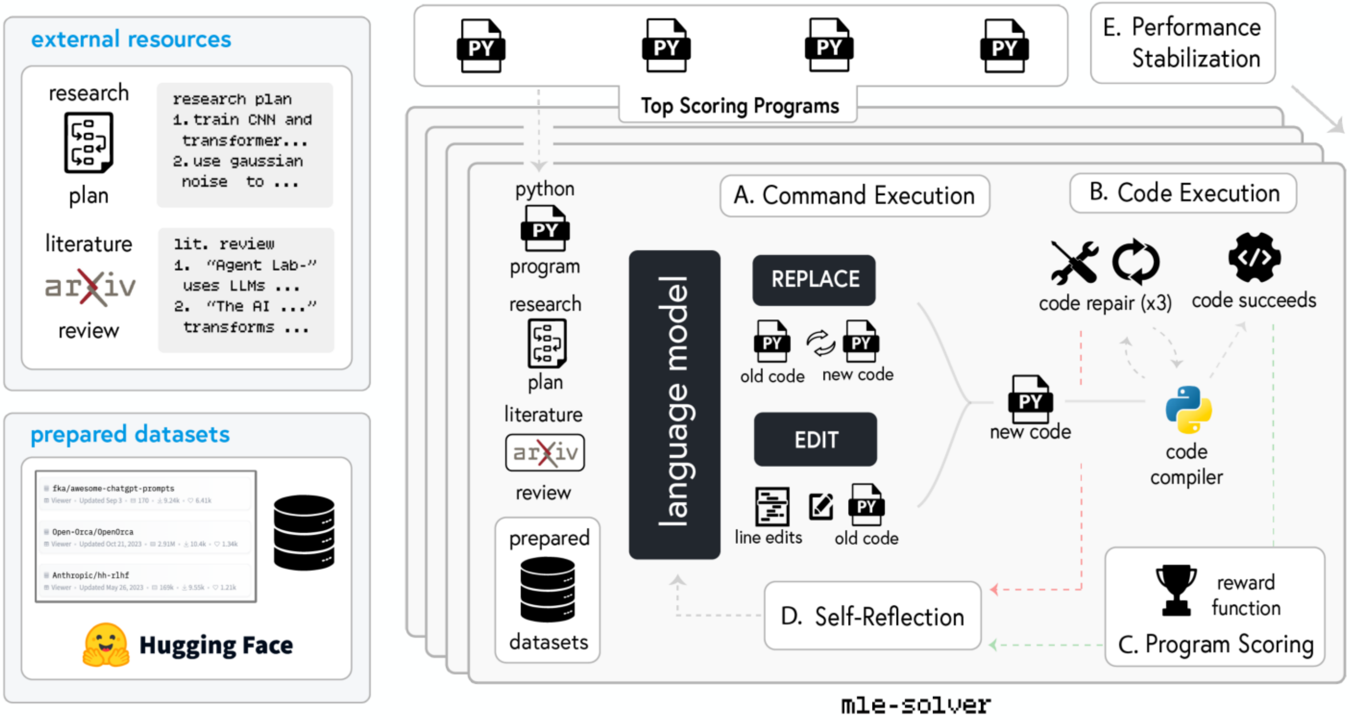
https://agentlaboratory.github.io/

Literature review

- Use arXiv API to summary, full text, add paper

Experiments

- Plan
- Data preparation
- Run experiments
  - mle-solver to write code, test, and refine code
  - Code refined using REPLACE/EDIT
  - Tries to run / fix code up to 3 times
  - Reward model to score effectiveness of ML code
  - Self-Reflection
- Results discussion



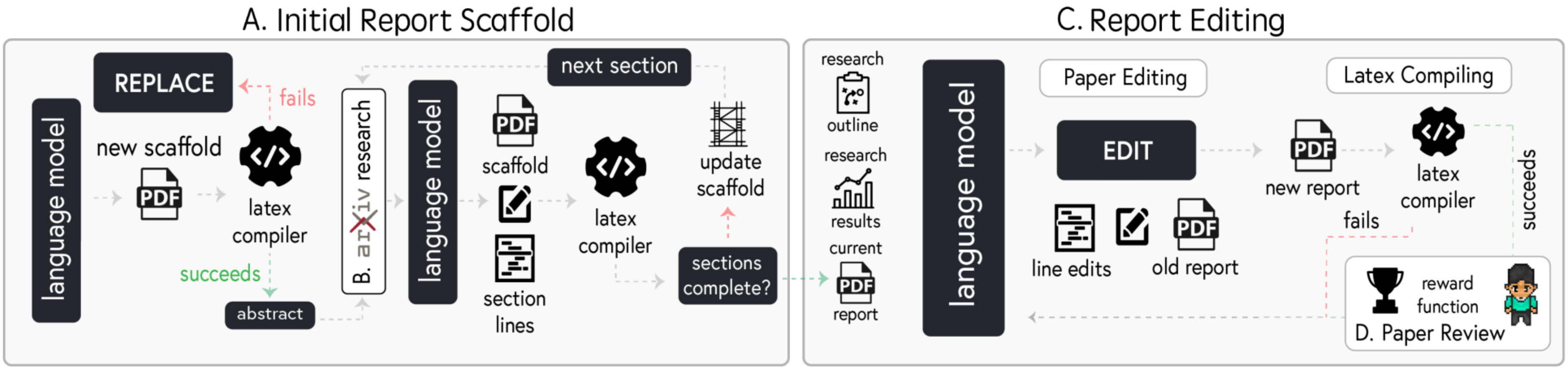
Agent Laboratory: Using LLM Agents as Research Assistants [Schmidgall et al, 2025]

Agent Laboratory

https://agentlaboratory.github.io/

Report writing

- Takes as input research plan, literature review, experimental results, insights and outputs research paper
- Steps: A. Initial report scaffold, B. Arxiv Research, C. Report Editing, D. Paper Review, E. Paper Refinement



paper-solver workflow

Agent Laboratory: Using LLM Agents as Research Assistants [Schmidgall et al, 2025]

Agent Laboratory

Human evaluation of generated papers (15 papers over 5 research questions)

Average human evaluated score by `Agent Laboratory` base LLM

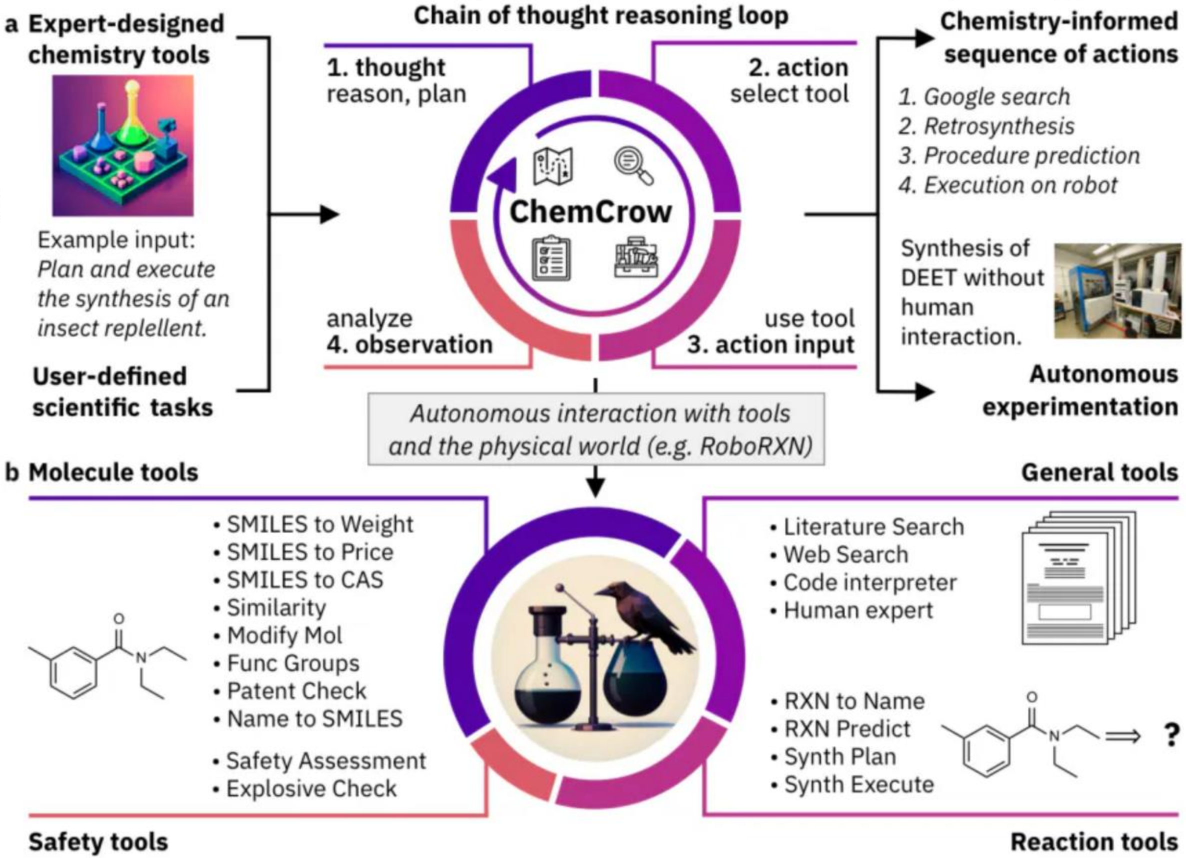
		gpt-4o			o1-mini			o1-preview		
Research Question	Research Type	Experiment Quality	Report Quality	Usefulness	Experiment Quality	Report Quality	Usefulness	Experiment Quality	Report Quality	Usefulness
Are image transformers more or less sensitive to noise than convolutional networks?	Computer Vision	**1.5** / 5	**2.5** / 5	**2.5** / 5	**4.0** / 5	**3.0** / 5	**4.0** / 5	**2.5** / 5	**3.5** / 5	**4.5** / 5
Does gender affect the accuracy on of language models on answering gsm8k questions?	NLP [Social Sci]	**3.0** / 5	**3.0** / 5	**4.0** / 5	**3.0** / 5	**3.5** / 5	**4.0** / 5	**3.0** / 5	**3.5** / 5	**5.0** / 5
Do language models improve accuracy on MedQA when asked to perform differential diagnosis?	NLP [Medical]	**3.0** / 5	**3.5** / 5	**4.5** / 5	**2.5** / 5	**2.5** / 5	**4.5** / 5	**3.5** / 5	**3.5** / 5	**4.0** / 5
Do language models exhibit cognitive biases similar to humans, such as anchoring bias?	NLP [Cog Sci]	**2.5** / 5	**2.5** / 5	**4.5** / 5	**4.0** / 5	**3.5** / 5	**4.5** / 5	**3.0** / 5	**2.0** / 5	**4.0** / 5
Are language models sensitive to word order in multiple choice benchmarks?	NLP [Core]	**3.0** / 5	**3.5** / 5	**4.5** / 5	**2.5** / 5	**3.5** / 5	**4.5** / 5	**2.5** / 5	**4.5** / 5	**4.5** / 5
	Average	**2.6** / 5	**3.0** / 5	**4.0** / 5	**3.2** / 5	**3.2** / 5	**4.3** / 5	**2.9** / 5	**3.4** / 5	**4.4** / 5

Agent Laboratory: Using LLM Agents as Research Assistants [Schmidgall et al, 2025]

ChemCrow: LLM powered chemistry engine

List of tools with name, description, expected input and output

Use ReAct style prompting: thought, action, action input, observation



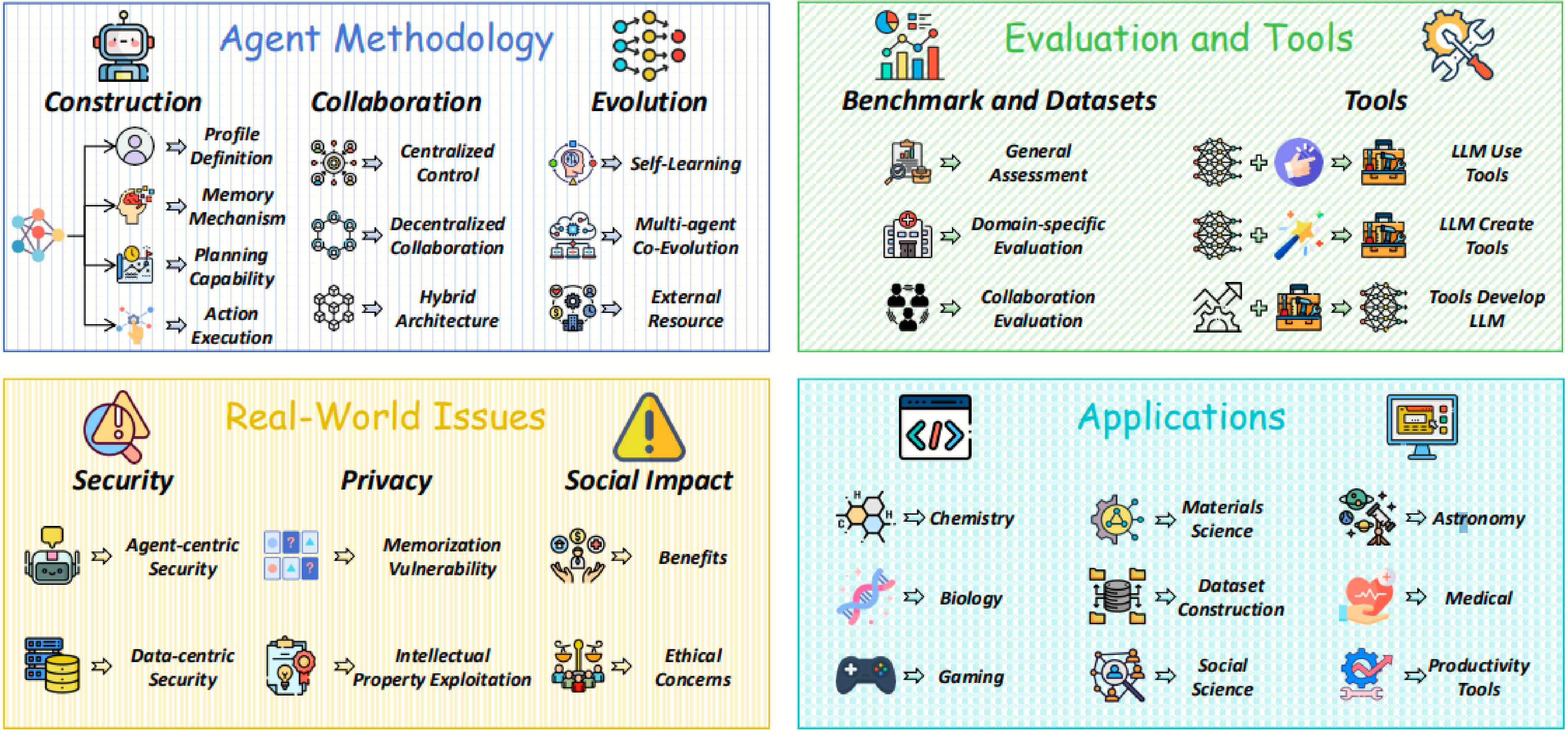
ChemCrow: Augmenting large-language models with chemistry tools [Bran et al, 2023]

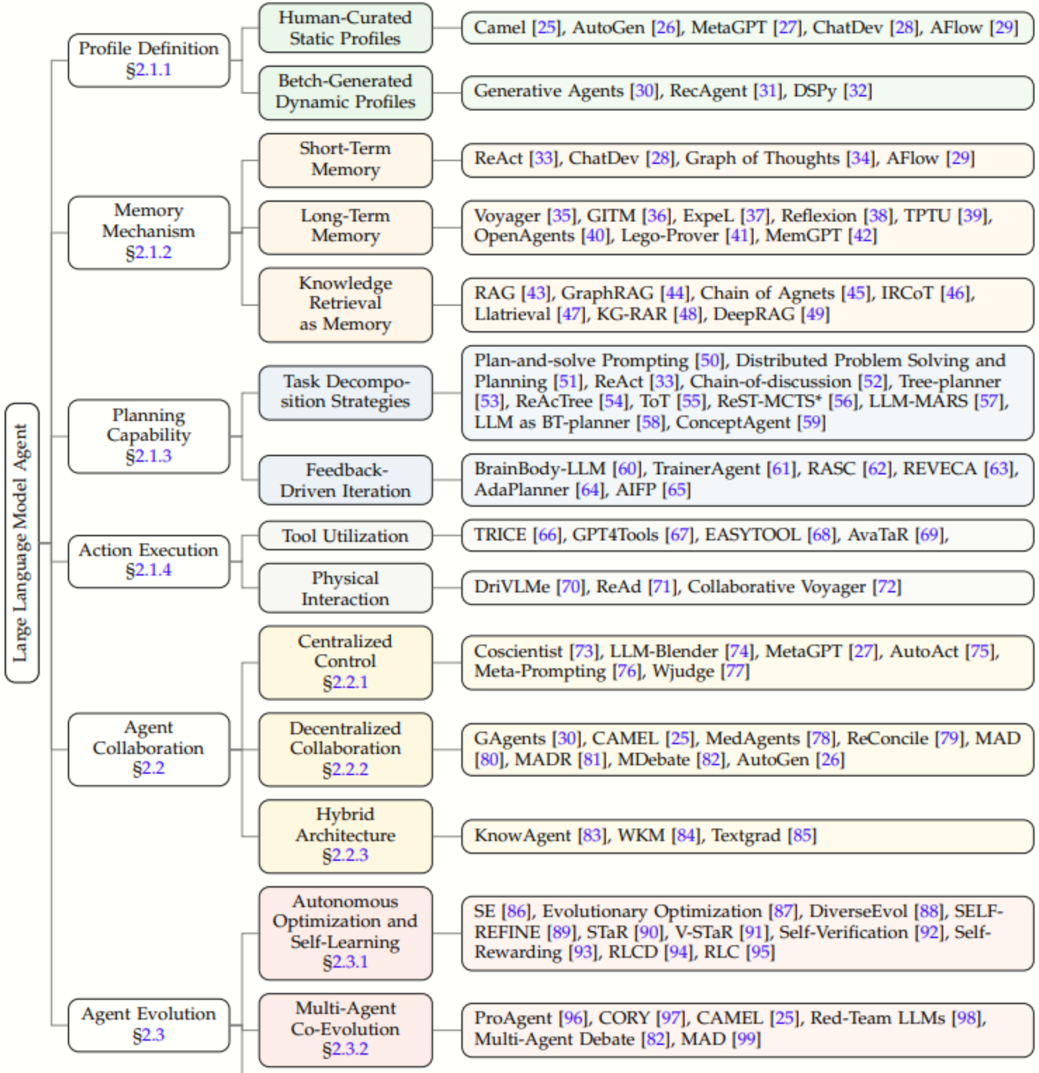
Large Language Model Agent: A Survey on Methodology, Applications and Challenges

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, Yiqiao Jin, Fan Zhang, Xian Wu, Hanqing Zhao, Dacheng Tao, *Fellow, IEEE*, Philip S. Yu, *Fellow, IEEE* and Ming Zhang

Abstract—The era of intelligent agents is upon us, driven by revolutionary advancements in large language models. Large Language Model (LLM) agents, with goal-driven behaviors and dynamic adaptation capabilities, potentially represent a critical pathway toward artificial general intelligence. This survey systematically deconstructs LLM agent systems through a methodology-centered taxonomy, linking architectural foundations, collaboration mechanisms, and evolutionary pathways. We unify fragmented research threads by revealing fundamental connections between agent design principles and their emergent behaviors in complex environments. Our work provides a unified architectural perspective, examining how agents are constructed, how they collaborate, and how they evolve over time, while also addressing evaluation methodologies, tool applications, practical challenges, and diverse application domains. By surveying the latest developments in this rapidly evolving field, we offer researchers a structured taxonomy for understanding LLM agents and identify promising directions for future research. The collection is available at https://github.com/luo-junyu/Awesome-Agent-Papers.

Index Terms—Large language model, LLM agent, AI agent, intelligent agent, multi-agent system, LLM, literature survey


Agent Methodology
Construction
Profile Definition
Memory Mechanism
Planning Capability
Action Execution
Collaboration
Centralized Control
Decentralized Collaboration
Hybrid Architecture
Evolution
Self-Learning
Multi-agent Co-Evolution
External Resource
Evaluation and Tools
Benchmark and Datasets
General Assessment
Domain-specific Evaluation
Collaboration Evaluation
Tools
LLM Use Tools
LLM Create Tools
Tools Develop LLM
Real-World Issues
Security
Agent-centric Security
Data-centric Security
Privacy
Memorization Vulnerability
Intellectual Property Exploitation
Social Impact
Benefits
Ethical Concerns
Applications
Chemistry
Biology
Gaming
Materials Science
Dataset Construction
Social Science
Astronomy
Medical
Productivity Tools



Large Language Model Agent
Profile Definition §2.1.1
Human-Curated Static Profiles
Camel [25], AutoGen [26], MetaGPT [27], ChatDev [28], AFlow [29]
Betch-Generated Dynamic Profiles
Generative Agents [30], RecAgent [31], DSPy [32]
Memory Mechanism §2.1.2
Short-Term Memory
ReAct [33], ChatDev [28], Graph of Thoughts [34], AFlow [29]
Long-Term Memory
Voyager [35], GITM [36], ExpeL [37], Reflexion [38], TPTU [39], OpenAgents [40], Lego-Prover [41], MemGPT [42]
Knowledge Retrieval as Memory
RAG [43], GraphRAG [44], Chain of Agnets [45], IRCoT [46], Llatrieval [47], KG-RAR [48], DeepRAG [49]
Planning Capability §2.1.3
Task Decomposition Strategies
Plan-and-solve Prompting [50], Distributed Problem Solving and Planning [51], ReAct [33], Chain-of-discussion [52], Tree-planner [53], ReAcTree [54], ToT [55], ReST-MCTS* [56], LLM-MARS [57], LLM as BT-planner [58], ConceptAgent [59]
Feedback-Driven Iteration
BrainBody-LLM [60], TrainerAgent [61], RASC [62], REVECA [63], AdaPlanner [64], AIFP [65]
Action Execution §2.1.4
Tool Utilization
TRICE [66], GPT4Tools [67], EASYTOOL [68], AvaTaR [69],
Physical Interaction
DriVLMe [70], ReAd [71], Collaborative Voyager [72]
Agent Collaboration §2.2
Centralized Control §2.2.1
Coscientist [73], LLM-Blender [74], MetaGPT [27], AutoAct [75], Meta-Prompting [76], Wjudge [77]
Decentralized Collaboration §2.2.2
GAgents [30], CAMEL [25], MedAgents [78], ReConcile [79], MAD [80], MADR [81], MDebate [82], AutoGen [26]
Hybrid Architecture §2.2.3
KnowAgent [83], WKM [84], Textgrad [85]
Agent Evolution §2.3
Autonomous Optimization and Self-Learning §2.3.1
SE [86], Evolutionary Optimization [87], DiverseEvol [88], SELF-REFINE [89], STaR [90], V-STaR [91], Self-Verification [92], Self-Rewarding [93], RLCD [94], RLC [95]
Multi-Agent Co-Evolution §2.3.2
ProAgent [96], CORY [97], CAMEL [25], Red-Team LLMs [98], Multi-Agent Debate [82], MAD [99]

Memory Mechanisms

Short-Term Memory. Short-term memory retains agentinternal dialog histories and environmental feedback to support context-sensitive task execution

Long-Term Memory. Long-term memory systematically archives agents' intermediate reasoning trajectories and synthesizes them into reusable tools for future invocation

- Skill libraries
- Experience repositories

Knowledge Retrieval as Memory. This paradigm diverges from agent-internal memory generation by integrating external knowledge repositories into generation processes, effectively expanding agents' accessible information boundaries.

Planning Capabilities

Task Decomposition Strategies. Task decomposition represents a basic approach to enhancing LLM planning capabilities by breaking down complex problems into more manageable subtasks.

1. Trees, chains, etc.

Feedback-Driven Iteration. Feedback-driven iteration is a crucial aspect of LLM planning capabilities, enabling the agent to learn from the feedback and enhance its performance over time

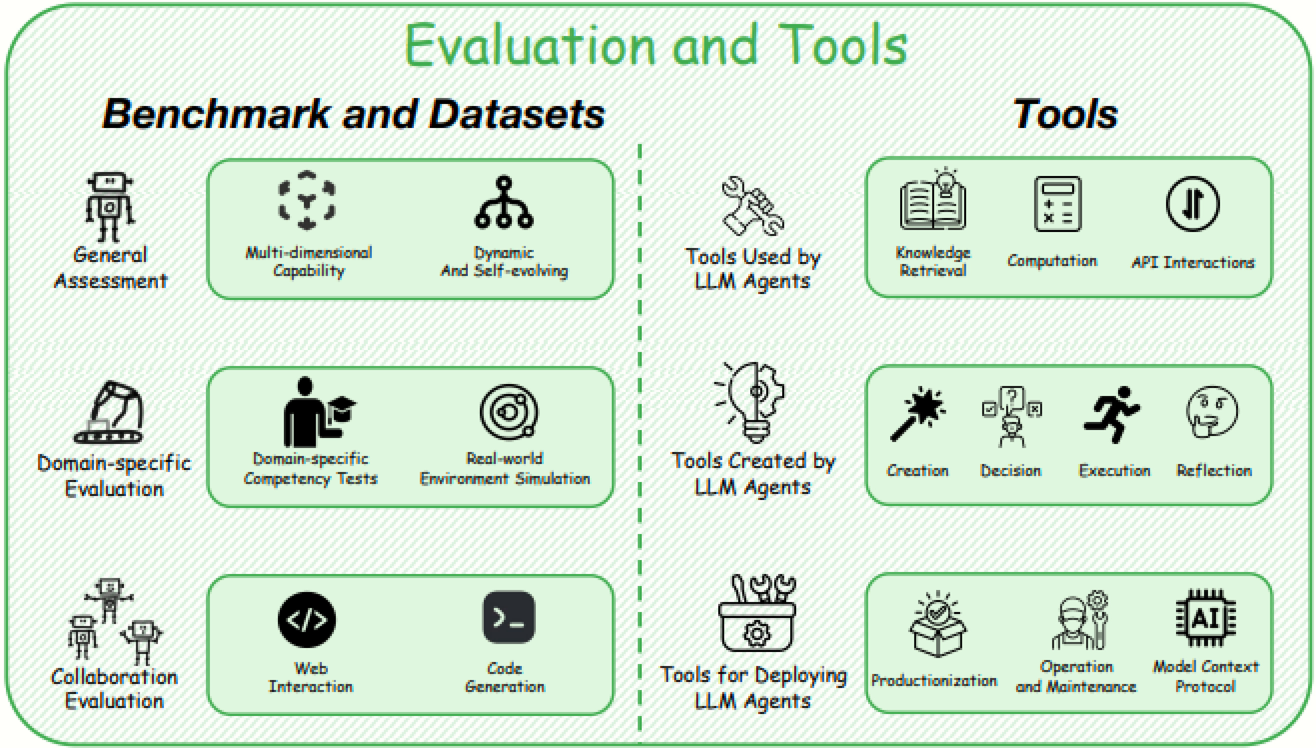
Knowledge Retrieval as Memory. This paradigm diverges from agent-internal memory generation by integrating external knowledge repositories into generation processes, effectively expanding agents’ accessible information boundaries.

Collaboration

TABLE 1: A summary of agent collaboration methods.

Category	Method	Key Contribution
Centralized Control	Coscientist [73]	Human-centralized experimental control
	LLM-Blender [74]	Cross-attention response fusion
	MetaGPT [27]	Role-specialized workflow management
	AutoAct [75]	Triple-agent task differentiation
	Meta-Prompting [76]	Meta-prompt task decomposition
	WJudge [77]	Weak-discriminator validation
Decentralized Collaboration	MedAgents [78]	Expert voting consensus
	ReConcile [79]	Multi-agent answer refinement
	METAL [115]	Domain-specific revision agents
	DS-Agent [116]	Database-driven revision
	MAD [80]	Structured anti-degeneration protocols
	MADR [81]	Verifiable fact-checking critiques
	MDebate [82]	Stubborn-collaborative consensus
	AutoGen [26]	Group-chat iterative debates
Hybrid Architecture	CAMEL [25]	Grouped role-play coordination
	AFlow [29]	Three-tier hybrid planning
	EoT [117]	Multi-topology collaboration patterns
	DiscoGraph [118]	Pose-aware distillation
	DyLAN [119]	Importance-aware topology
	MDAgents [120]	Complexity-aware routing

Evaluation and Tools



Applications

TABLE 7: Overview of Applications in LLM Agents.

Method	Domain	Core Idea
Scientific Discovery		
SciAgents [266]	General Sciences	Collaborative hypothesis generation
Curie [267]	General Sciences	Automated experimentation
ChemCrow [269]	Chemistry	Tool-augmented synthesis planning
AtomAgents [270]	Materials Science	Physics-aware alloy design
D. Kostunin el al [271]	Astronomy	Telescope configuration management
BioDiscoveryAgent [273]	Biology	Genetic perturbation design
GeneAgent [274]	Biology	Self-verifying gene association discovery
RiGPS [275]	Biology	Biomarker identification
BioRAG [211]	Biology	Biology-focused retrieval augmentation
PathGen-1.6M [276]	Medical Dataset	Pathology image dataset generation
KALIN [277]	Biology Dataset	Scientific question corpus generation
GeneSUM [278]	Biology Dataset	Gene function knowledge maintenance
AgentHospital [281]	Medical	Virtual hospital simulation
ClinicalLab [282]	Medical	Multi-department diagnostics
AIPatient [283]	Medical	Patient simulation
CXR-Agent [284]	Medical	Chest X-ray interpretation
MedRAX [285]	Medical	Multimodal medical reasoning
Gaming		
ReAct [33]	Game Playing	Reasoning and acting in text environments
Voyager [35]	Game Playing	Lifelong learning in Minecraft
ChessGPT [287]	Game Playing	Chess gameplay evaluation
GLAM [288]	Game Playing	Reinforcement learning in text environments
CALYPSO [289]	Game Generation	Narrative generation for D&D
GameGPT [290]	Game Generation	Automated game development
Sun et al. [291]	Game Generation	Interactive storytelling experience
Social Science		
Econagent [292]	Economy	Economic decision simulation
TradingGPT [293]	Economy	Financial trading simulation
CompeteAI [294]	Economy	Market competition modeling
Ma et al. [295]	Psychology	Mental health support analysis
Zhang et al. [296]	Psychology	Social behavior simulation
TE [297]	Psychology	Psychological experiment simulation
Generative agents [30]	Social Simulation	Human behavior emulation
Liu et al. [298]	Social Simulation	Learning from social interactions
S^3 [299]	Social Simulation	Social network behavior modeling
Productivity Tools		
SDM [300]	Software Development	Self-collaboration for code generation
ChatDev [301]	Software Development	Chat-powered development framework
MetaGPT [27]	Software Development	Meta-programming for collaboration
Agent4Rec [302]	Recommender Systems	User behavior modeling
AgentCF [303]	Recommender Systems	User-item interaction modeling
MACRec [304]	Recommender Systems	Multi-agent recommendation
RecMind [305]	Recommender Systems	Knowledge-enhanced recommendation

Future Trends

Scaling LLM-based multi-agent systems remains challenging due to high computational demands, inefficiencies in coordination, and resource utilization

Maintaining coherence across multi-turn dialogues and the longitudinal accumulation of knowledge requires effective memory mechanisms

LLMs, while knowledge-rich, are **neither comprehensive nor up-to-date**, thus potentially unsuitable as standalone replacements for structured databases.

Traditional AI evaluation frameworks, designed for static datasets and single-turn tasks, **fail to capture the complexities of LLM agents** in dynamic, multi-turn, and multi-agent environments

As agentic AI systems gain autonomy**, regulatory frameworks must evolve to ensure accountability, transparency, and safety**

LLM agents can simulate roles such as researchers, debators, and instructors [307], [327], **but their effectiveness is constrained by training data limitations and an incomplete understanding of human cognition**

Future Trends

Others?

Other Comments?