

Applications of Language Models

AoLM Spring 2025

Overcoming LLM Limitations with Agentic Workflow



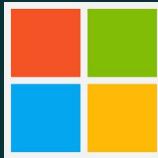
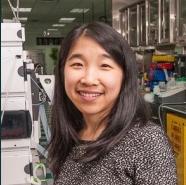
Harshit Surana

Allen Institute for Artificial Intelligence & OpenLocus



Structure & Some Requests

- 1. Primer on Agentic Workflows
 - 2. Two concrete examples
 - a. SWE
 - b. Scientific Discovery
 - 3. Industry scale problems
 - 4. Potential projects
- Interruption Encouraged for Clarity
 - Offline Follow-Ups for Deep Questions
 - Try Connecting with Your Projects & Real-World Applications



LLMs have
revolutionized NLP.

But they often fall
short in key areas.

LLM Limitations (for Advanced Tasks)

- Hallucinations
- Limited context
- Loses memory over longer context
- Non-trivial to verify tasks
- Opaque
- Poor adaptability for out of distribution domain & tasks
- Cannot integrate into specific user workflows

LLM Limitations (for Advanced Tasks)

- Hallucinations
- Limited context
- Loses memory over longer context
- Non-trivial to verify tasks
- Opaque
- Poor adaptability for out of distribution domain & tasks
- Cannot integrate into specific user workflows

Cool demo, but how do I *use* it?

An Example for NLP Code

Non-Agentic Workflow

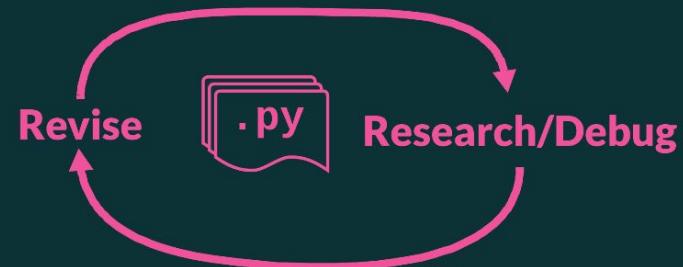
Write a Python script that uses a Hugging Face model for sentiment analysis on a given bio dataset. Please type out the code in one go without waiting or even using the backspace.



Agentic Workflow

Text

- Begin by refining the problem scope into a detailed specification
- Iterate with domain experts: doctors, clinical data scientists
- Design a modular preprocessing pipeline that can adapt to new domain rules
- Start with a baseline model like BioBERT and iterate
- Define domain-specific metrics that matter including accuracy & F1-score
- Evaluate results on the defined metrics & iterate the code



What are the solutions?

Agentic Design Patterns can tackle the limitations to make more productive use of the LLMs.

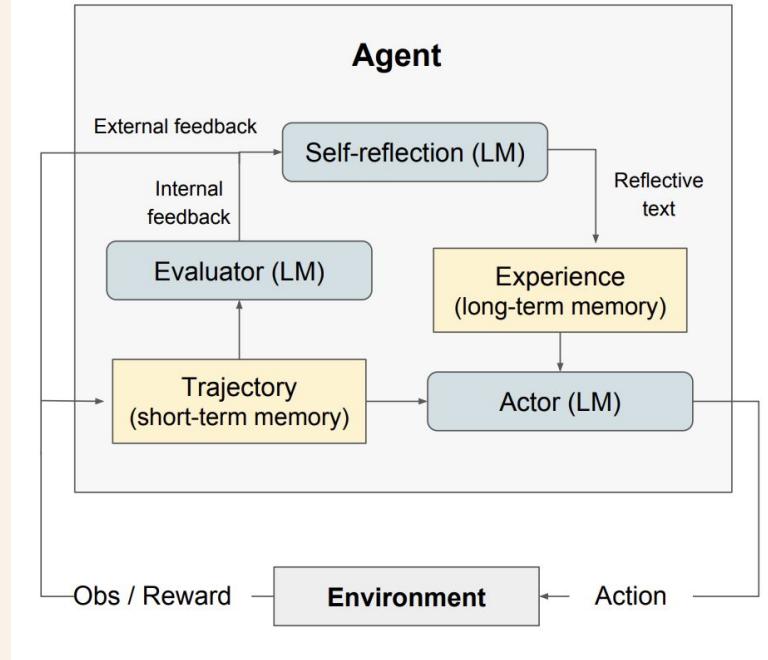
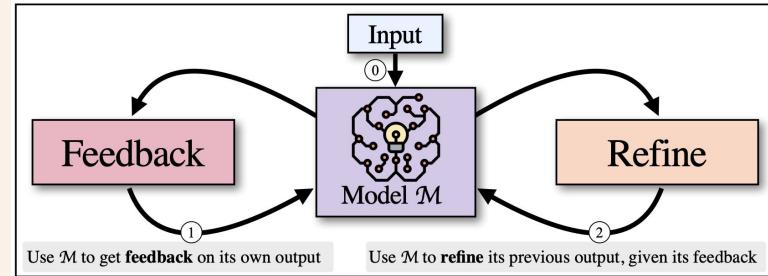
1. Reflection
2. Tool use
3. Planning
4. Multi-agent collab!

Reflection

Verify & reflect the LLM output by external feedback (i.e. unit tests) & LLMs. Use the reflection to iterate the results.

- Self Refine
- Reflexion

Self-Refine: Iterative Refinement with Self-Feedback. Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, Peter Clark. NIPS 24.

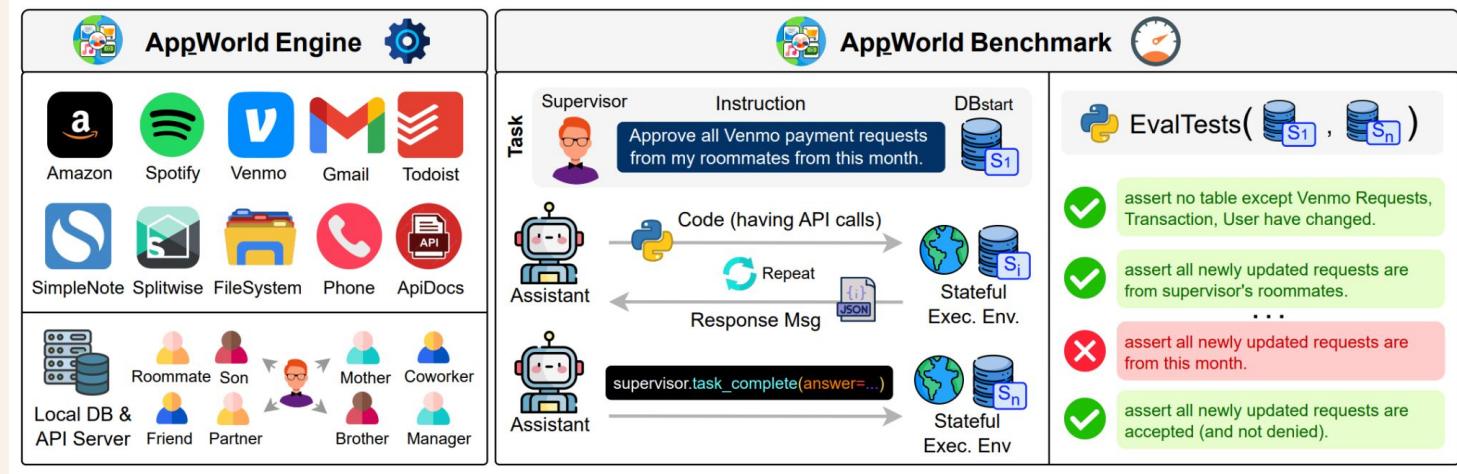


Tool Use

Connect with various tools & functions like:

Browser, APIs, code exec, **custom code** (domain specific code hard for LLMs to generate), search engine etc.

- **AppWorld**
- Gorilla LLM Exec
- Function calling

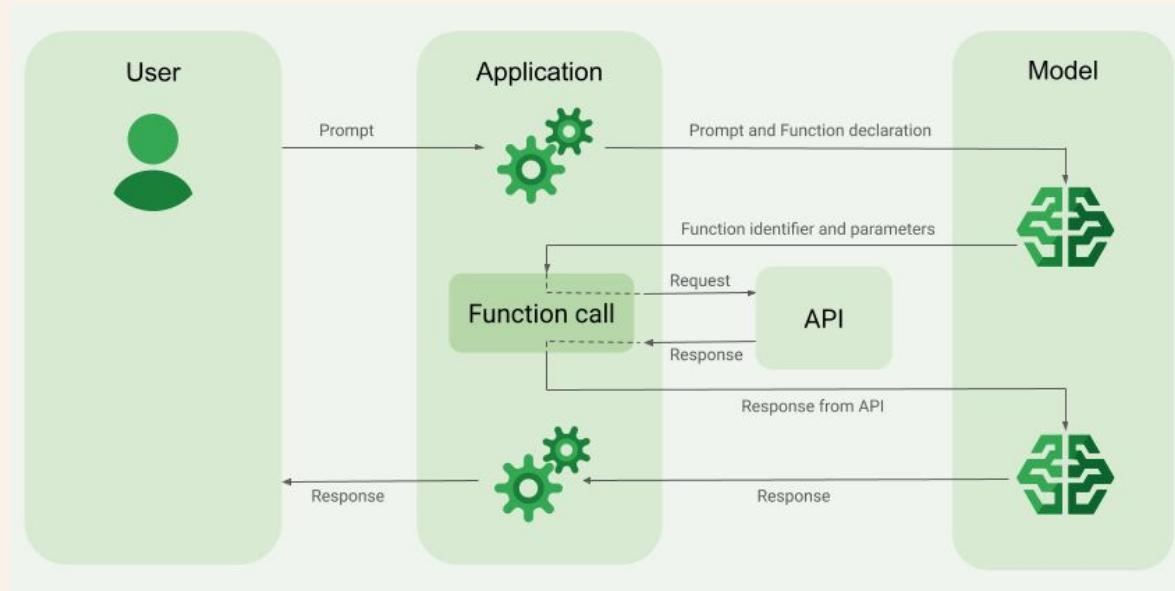


Tool Use

Connect with various tools & functions like:

Browser, APIs, code exec, **custom code** (domain specific code hard for LLMs to generate), search engine etc.

- AppWorld
- Gorilla LLM Exec
- **Function calling**

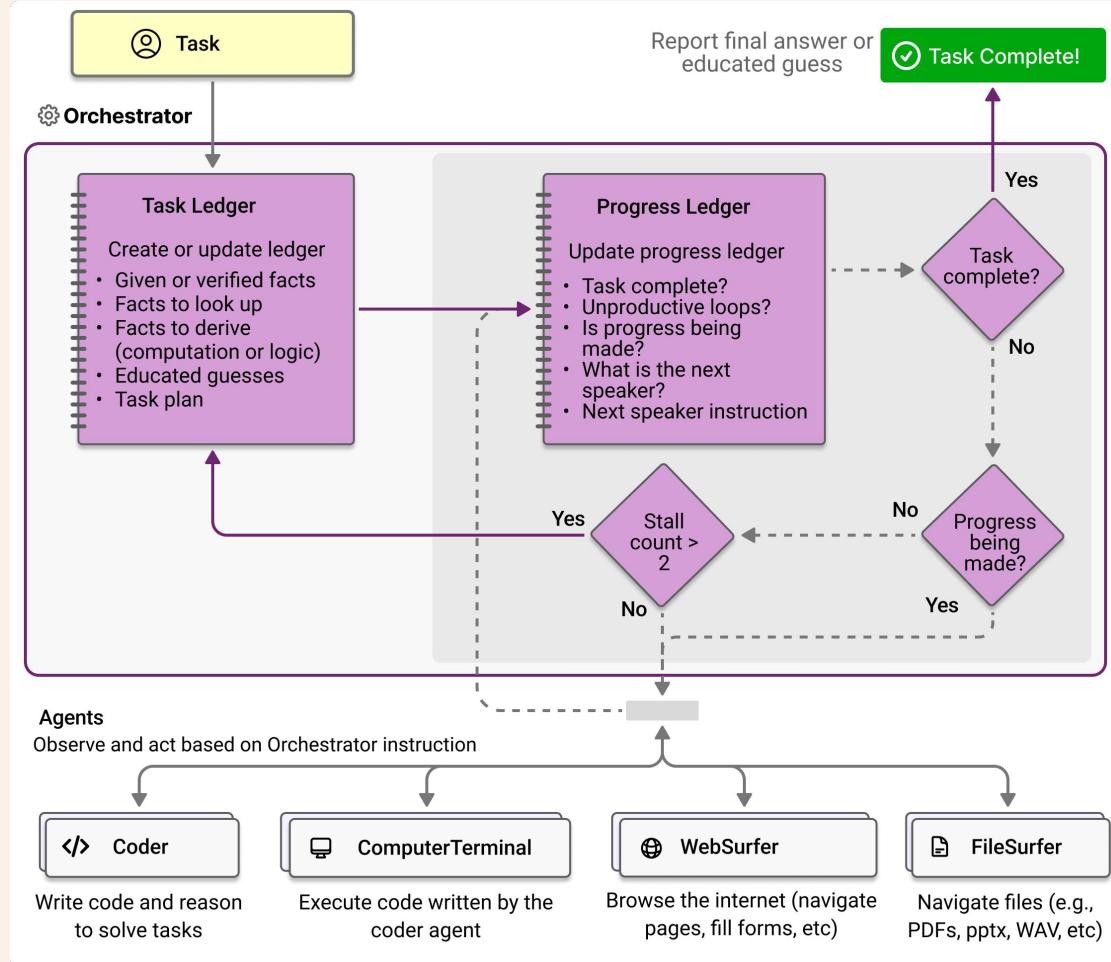


Planning

Plan tasks, track them while performing a task and reason over them if needed.

- Autogen Magento
- MetaGPT
- OpenHands
- HuggingGPT

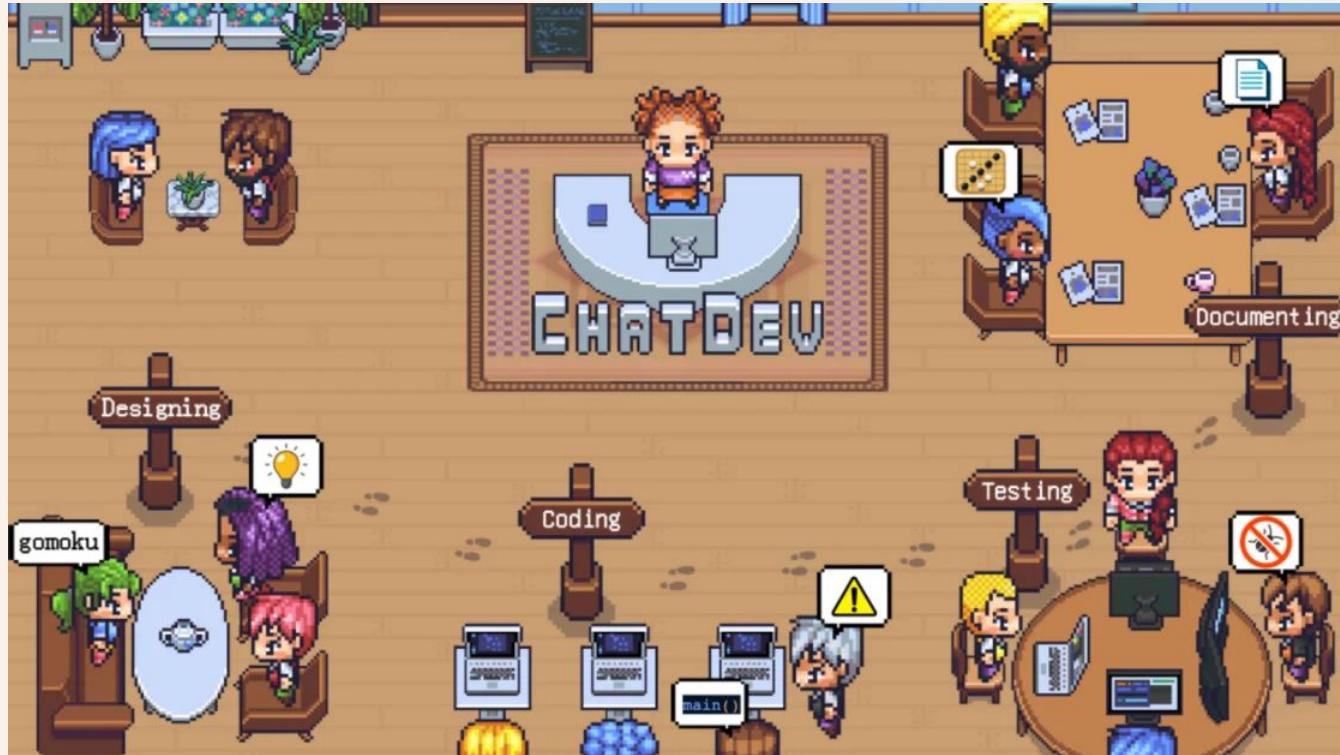
Magnetic-One: A Generalist Multi-Agent System for Solving Complex Tasks. Fournier et. al Microsoft Tech Report 2024.



Multi-agent Collab

Agents work together to solve a complex task.

- Autogen
- CrewAI
- **ChatDev**
- DataVoyager



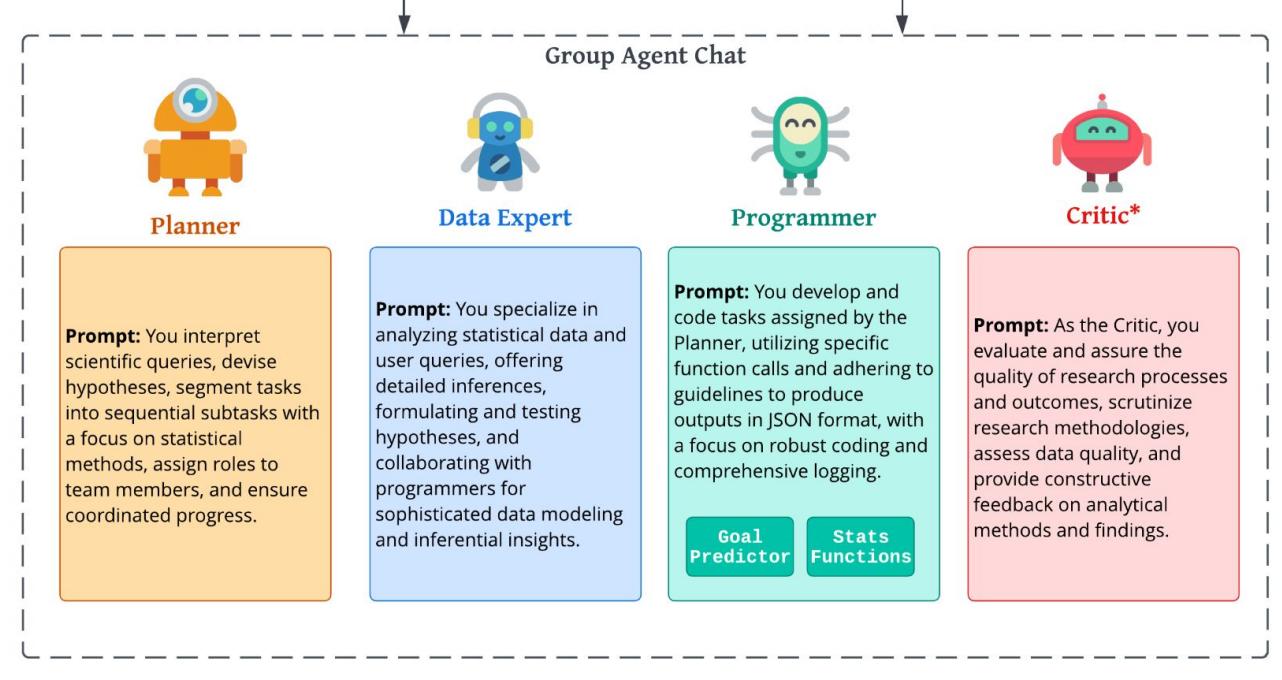
ChatDev: Communicative Agents for Software Development. Qian et. al.
ACL 2024



Multi-agent Collab

Agents work together to solve a complex task.

- Autogen
- CrewAI
- ChatDev
- **DataVoyager**
 - Agents
 - Memory
 - Functions
 - Literature



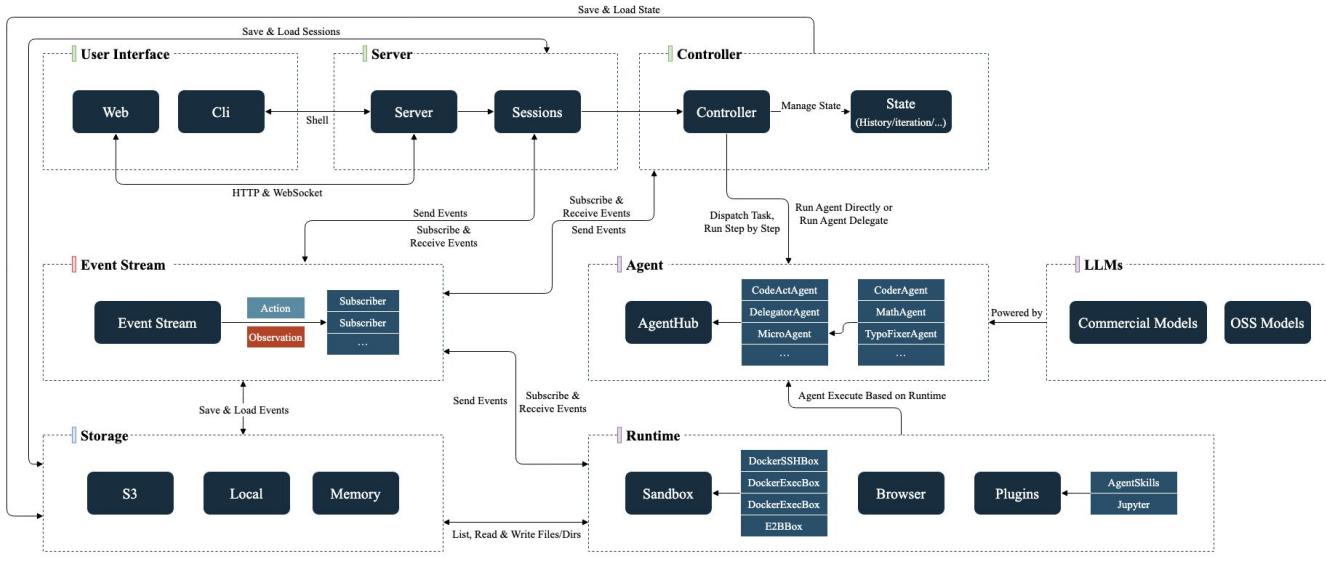
Data-driven Discovery with Large Generative Models. Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Sanchaita Hazra, Ashish Sabharwal, Peter Clark. ICML 2024.

Let's go over 2
concrete
examples!

Software Engg with LLMs

(not just code generation)

OpenHands by CMU & UIUC



Starred 37.9k

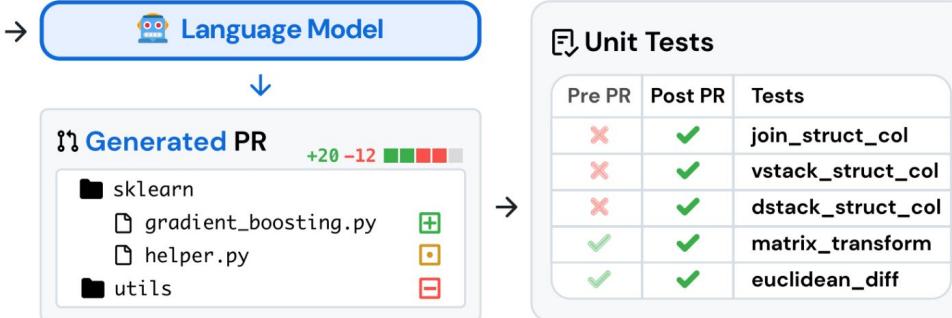
SWE Bench by Princeton

Issue

data leak in GBDT due to warm start (This is about the non-histogram-based version of...

Codebase

- sklearn/
- examples/
- README.rst
- reqs.txt
- setup.cfg
- setup.py



1 Scrape PRs

12 popular repositories
90% Python Code

2 Attribute Filter

- ✓ Resolves an issue
- ✓ Contributes tests

3 Execution Filter

- ✓ Installs successfully
- ✓ PR passes all tests

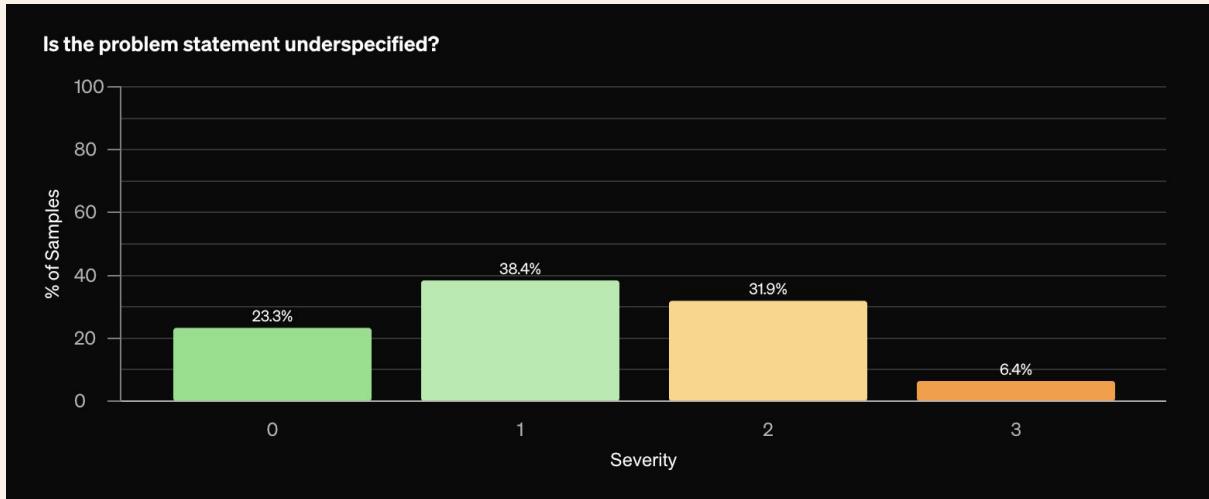
SWE-bench sources task instances from **real-world Python repositories** by connecting GitHub issues to merged pull request solutions that resolve related tests.

Provided with the **issue text** and a **codebase snapshot**, **models generate a patch** that is evaluated against real tests.

Can Language Models Resolve Real-World GitHub Issues? Carlos E. Jimenez*, John Yang*, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, Karthik Narasimhan. ICLR 2024



SWE Bench Verified by OpenAI



SWE-bench Verified manually screened by 93 SWEs for 1,699 random samples.

Whether we consider the issue description to be
underspecified and hence unfair to be testing on.

Whether the FAIL_TO_PASS unit tests **filter out valid solutions.**

<https://openai.com/index/introducing-swe-bench-verified/>

Neil Chowdhury, James Aung, Chan Jun Shern, Oliver Jaffe, Dane Sherburn, Giulio Starace, Evan Mays, Rachel Dias, Marwan Aljubeh, Mia Glaese, Carlos E. Jimenez, John Yang, Kevin Liu, Aleksander Madry



OpenHands - CodeAct Agent

First agent to **cross 50%** in SWE-Bench Verified

Task planning by developing capabilities for bug detection, codebase management, and optimization

Made a number of fixes to make it easier for agents to **traverse directories**

Switched to use **function calling**, a method used by language models to more precisely specify the functions available to them

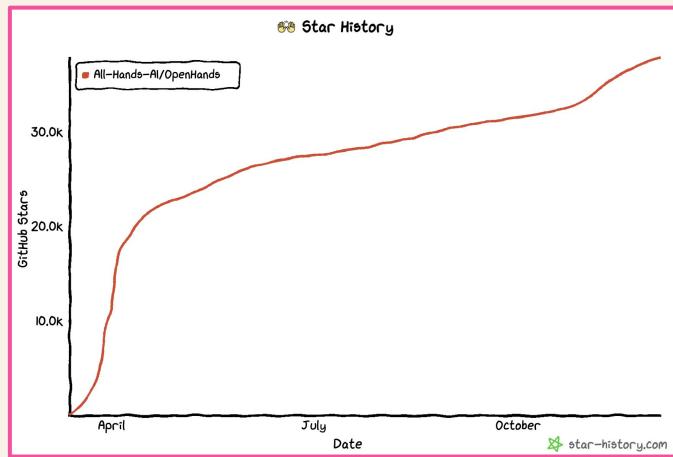
Agent + Model	Score
OpenHands + CodeAct v2.1 + Claude 3.5 Sonnet	53.00
Anthropic Tools + Claude 3.5 Sonnet	49.00
Anthropic Tools + Claude 3.5 Haiku	40.60
Composio SWEkit + Claude 3.5 Sonnet	40.60
SWE-agent + Claude 3.5 Sonnet	33.60
SWE-agent + Claude 3 Opus	18.20
RAG + Claude 3 Opus	7.00
RAG + Claude 2	4.40

Contributing to OSS LLM Dev with OpenHands

We are contributing for **better evals & data science agents** - can discuss more offline.

Do try contributing to OpenHands. Their **aim is to have full replication of production-grade applications with LLMs**

<https://github.com/All-Hands-AI/OpenHands/blob/main/CONTRIBUTING.md>



📁 browsing_delegation
📁 commit0_bench
📁 discoverybench
📁 gaia
📁 gorilla
📁 gpqa
📁 humanevalfix
📁 logic_reasoning
📁 miniwob
📁 mint
📁 ml_bench
📁 scienceagentbench
📁 swe_bench
📁 toolqa

Quick Detour on Scientific Discovery & Data Analysis

Methods of Scientific Inquiry

Theoretical Science

Develop models or theories to explain phenomena

Experimental Science

Conduct experiments to test pre-defined hypotheses

Observational Science

Observe & collect data, build methods to explain it

Methods of Scientific Inquiry

Theoretical Science

Develop models or theories to explain phenomena

Experimental Science

Conduct experiments to test pre-defined hypotheses

Observational Science

Observe & collect data, build methods to explain it

A lot of important science has come out of looking at **observational data**.

National Longitudinal Survey of Youth | 1979



U.S. BUREAU OF LABOR STATISTICS

500,000 results in S2
from 1979



37000+ papers published from 1976

Methods of Scientific Inquiry

Theoretical Science

Develop models or theories to explain phenomena

Experimental Science

Conduct experiments to test pre-defined hypotheses

Observational Science

Observe & collect data, build methods to explain it

A lot of important science has come out of looking at **observational data**.

Can we **autonomously** discover

- insights from datasets to reduce turnaround time?
- undiscovered knowledge without performing additional data collection?

National Longitudinal Survey of Youth | 1979



U.S. BUREAU OF LABOR STATISTICS

500,000 results in S2
from 1979



Nurses'
Health Study



37000+ papers published
from 1976

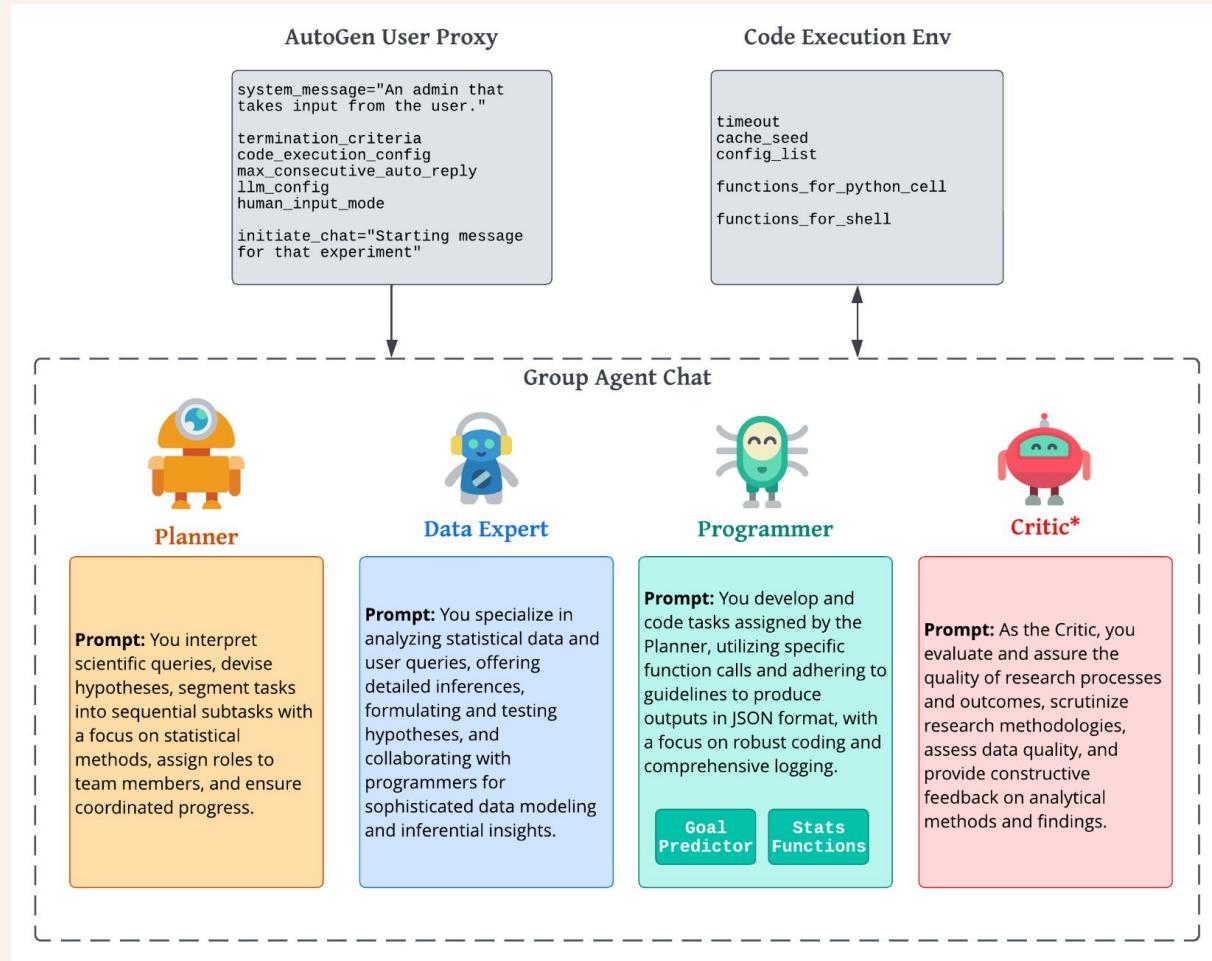
Scientific Discovery with LLMs

DataVoyager

All of the subagents and the controller is based on GPT-4 with a specialized prompt.

They see the full history, but the agent-specific prompt elicit specialized behavior.

Programmer has the function-calling ability which allows it to “execute” the generated code in an **interactive python shell**.



Data-driven Discovery

- Comprehensive data-understanding
- Ex-ante hypothesis search/generation
- Planning & orchestrating research pathways
- Execute & verify candidate hypotheses
- Accommodating human feedback
- Reproducible and robust results

Data-driven Discovery: Following Newell & Simon (1976), we define a heuristic search problem that aims to describe a given set of observations by uncovering the laws that govern its data-generating process.

E.g., “under context c, variables v have relationship r”

Newell, A. and Simon, H. A. Computer science as empirical inquiry: symbols and search. Commun. ACM, 1976



Data-driven Discovery as a Predictive Task

Given a dataset **D** and a Discovery Goal **G**, derive the most specific hypothesis **H** addressing G and supported by D.

Alternatively,
A data-driven hypothesis **H** is a declarative sentence about the state of the world whose truth value may be inferred from a given dataset D using a verification procedure $V: H \rightarrow \{\text{supported, unsupported}\}$, for instance, via *statistical modeling*.

Inspired by Thompson and Skau (2023), we introduce a structured formalism that breaks a hypothesis down into **three hypothesis dimensions**:

Context: **Boundary conditions** that limit the scope of a hypothesis. E.g., “for men over the age of 30”

Variables: **Known set of concepts** that interact in a meaningful way under a given context to produce the hypothesis. E.g., gender, age, or income

Relationship: **Interactions between a given set of variables** under a given context that produces the hypothesis. E.g., “quadratic relationship”, “inversely proportional”, or piecewise conditionals

Dataset:

habitat type	nonnative gardening	nonnative unintentional	nonnative agriforest	elevation ...
croplands	5	0	2	675
wetlands	0	4	1	88
urban	2	1	0	329
...

Goal: How did urban land use affect the invasion of different types of introduced plants in Catalonia?

	gold	predicted	score
context	urban habitat type	urban habitat type	● 1.0
variable	gardening, unintentional	gardening, agriforst	● 0.3
relationship	reduced	increased	● 0.0
Final Score: 0.21			



Urban land use reduced invasion by gardening plants over unintentionally introduced ones.

DiscoveryBench

264 Tasks, 20+ papers, 6 domains

We replicate the **scientific process** undertaken by researchers to search for and validate a hypothesis from datasets

Data-first: Filter papers + workflows based on public datasets: National Longitudinal Surveys, Global Biodiversity Info Facility, World Bank Open Data; 2) replicate in Python.

Replication took up to 90 person-hours per dataset, often (30%) not resulting in success.

Code-first: Checked 785 repos + datasets, 85% had missing or non-adaptable code to Python, or closed datasets. Only few passed the check.

Papers from Nature, AER, etc.

Task Dataset: Dataset contains information from National Longitudinal Survey of Youth (NLSY79). It includes information about the Demographics, Family Background, Education ...

Discovery Goal: How does socioeconomic status affect the likelihood of completing a BA degree?

Target Hypothesis:
Socioeconomic status has a positive relationship with college degree completion with a coefficient of 0.4729 with statistical significance.

Data Loading & Cleaning

```
df=pd.read_csv('NLSCombine.csv')

columns_to_clean = ['Number of students in class last year attended at this school, 1981', 'Highest grade completed by respondent's mother, 1979', 'Highest grade completed by respondent's father, 1979', 'Family size, 1979', 'Highest grade completed, 1979', 'Rank in class last year attended at this school, 1981', 'Total net family income, previous calendar year, 1979']

for col in columns_to_clean:
    df[col] = df[col].apply(lambda x: np.nan if x < 0 else x)

df = df.dropna(subset=columns_to_clean, thresh=6)

# Standardize the continuous and ordinal variables
scaler = StandardScaler()

df[['STANDARDIZED_INCOME', 'STANDARDIZED_FAMILY_SIZE',
   'STANDARDIZED_FATHER_EDUCATION', 'STANDARDIZED_MOTHER_EDUCATION']] = scaler.fit_transform(df[['Total net family income, previous calendar year, 1979', 'Family size, 1979', 'Highest grade completed by respondent's father, 1979', 'Highest grade completed by respondent's mother, 1979']])

# Initialize the IterativeImputer
imputer = IterativeImputer(max_iter=10, random_state=0)

# Columns to impute - focusing on the ones with NaN values and relevant to SES calculation
```

Calculate Academic Ability et al.

```
score_cols=[  
    'ASAVM - Arithmetic Reasoning Z Score (rounded), 1981',  
    'ASVAR - Word Knowledge Z Score (rounded), 1981',  
    'ASVAN - Paragraph Comprehension Z Score (rounded), 1981',  
    'ASVWA - Mathematics Knowledge Z Score (rounded), 1981'  
,  
  
    df['all scores exist']=df[score_cols].gt(0).all(axis=1)  
  
df['ABILITY']=  
    df['ASAVM - Arithmetic Reasoning Z Score (rounded), 1981'] +  
    df['ASVAR - Word Knowledge Z Score (rounded), 1981'] +  
    df['ASVAN - Paragraph Comprehension Z Score (rounded), 1981'] +  
    df['ASVWA - Mathematics Knowledge Z Score (rounded), 1981']  
  
df['BA DEGREE COMPLETED']=df['Highest grade completed, 1979'].gt(14)
```

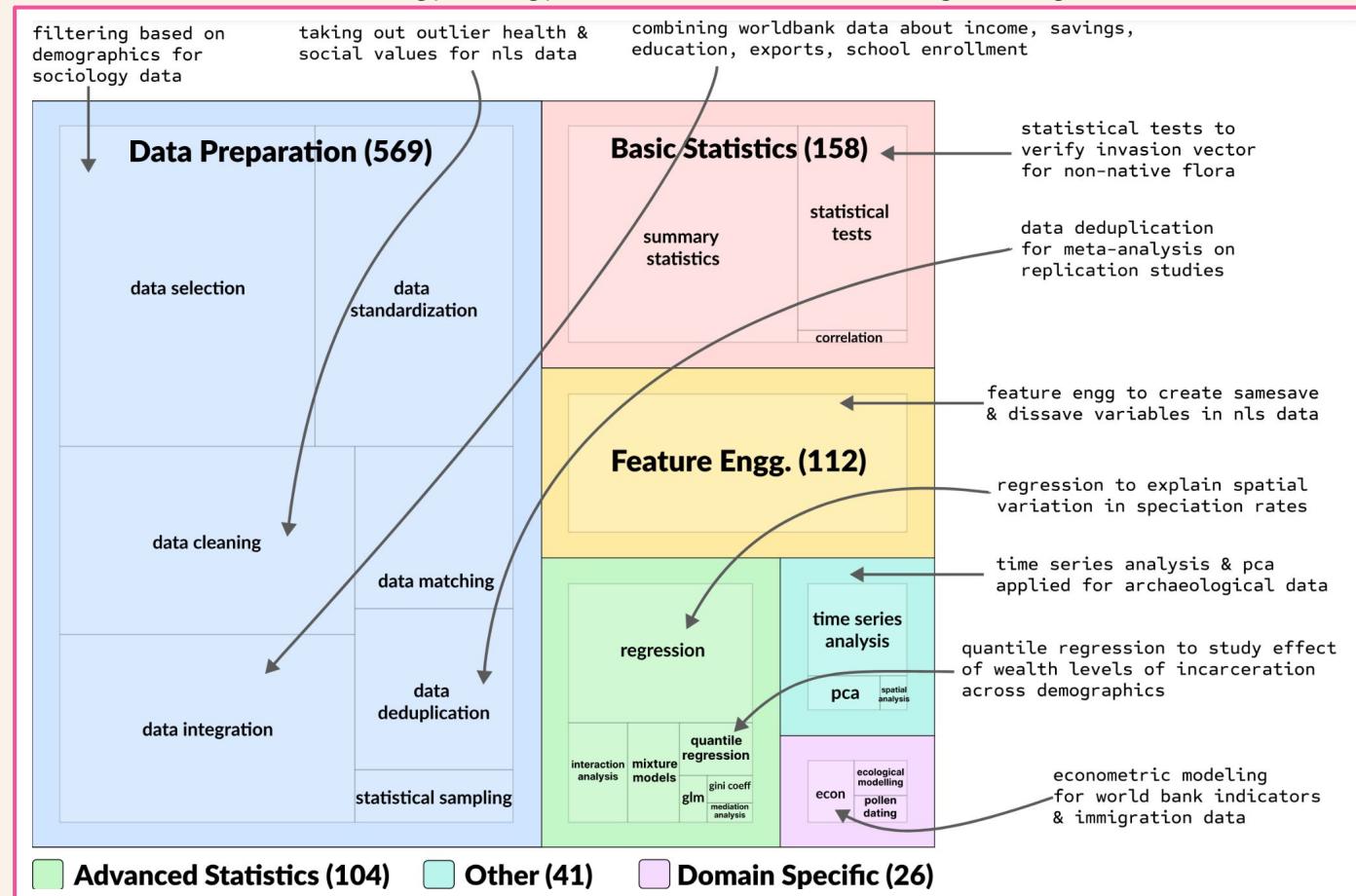
Data Filtering & Modeling

```
# Limit data to records with percentile in the range 0 to 100  
df = df[(df['PERCENTILE IN CLASS'] <= 100) &  
        | (df['PERCENTILE IN CLASS'] >= 0)]  
  
sub_dataset['BA DEGREE COMPLETED'] =  
    sub_dataset['BA DEGREE COMPLETED'].astype(int)  
  
X = sub_dataset[['SES']]  
y = sm.add_constant(X)  
y = sub_dataset['BA DEGREE COMPLETED']  
  
model = sm.Logit(y, X)  
result = model.fit()  
  
result.summary()
```

Calculate SES

```
weight_income = 0.5  
weight_education = 0.25  
  
# Calculate SES as a weighted composite of standardized measures  
df['SES'] = (weight_income * df['STANDARDIZED_INCOME'] +  
            weight_education * df['STANDARDIZED_FATHER_EDUCATION'] +  
            weight_education * df['STANDARDIZED_MOTHER_EDUCATION'])
```

DB-Real (6 domains: sociology, biology, humanities, economics, engineering, & meta-science)



Discovery Agents

All discovery agents have access to a python environment, capable of generating and executing programs on the datasets

CodeGen

generates the entire code at one go to solve the task, with help of a demonstration example in the context.

After code execution and based on the result, it generates the NL hypothesis and summarizes the workflow

ReAct

solves the task by generating thought and subsequent codes in a multi-turn fashion.

A traditional sequential-decision maker.

DataVoyager

is a multi-component data-driven discovery agent.

It has four components: planner, code generator, data analysis, and critic, that orchestrate the discovery process.

Reflexion (Oracle)

is an extension of CodeGen agent, where at the end of one trial, we provide an “oracle” feedback about task completion, and it generates a reflection to improve in the next trial till it solves the task, or maximum trials (3) are reached.

DataVoyager is a part of NORA - a Science Copilot

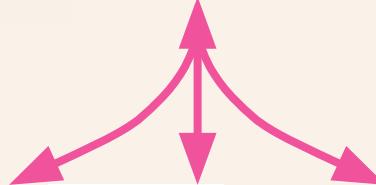
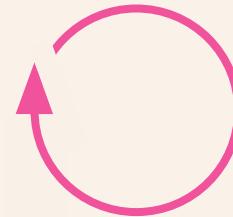
1) Collaborates naturally with human scientists

2) Uses tools on humans' behalf

3) Learns & improves over time.

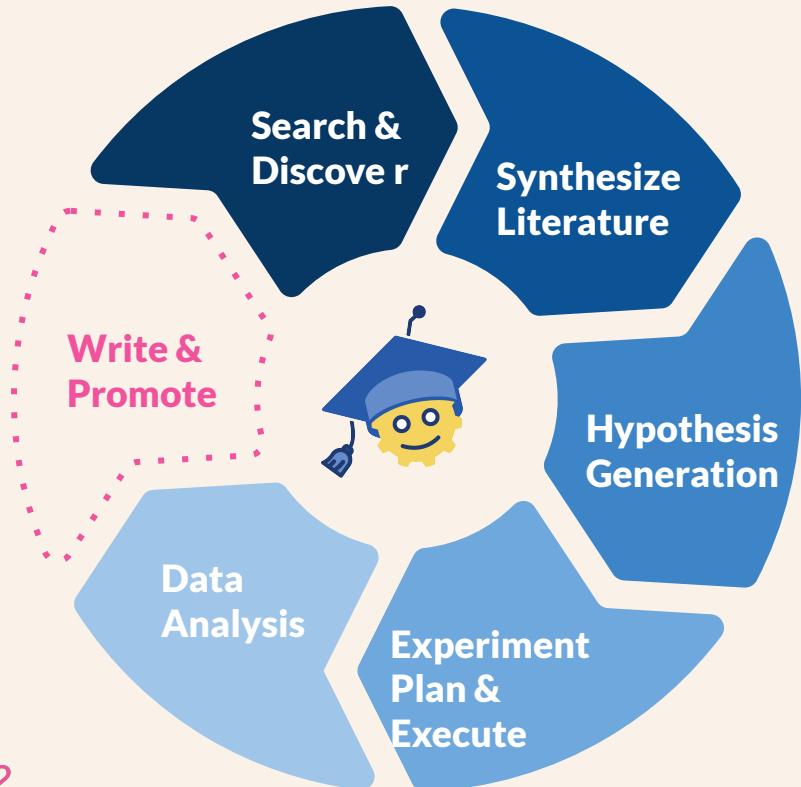


Slack
Semantic Reader
Jupyter



... with Infra. for Personalization & Optimization

Research Functions



Surfaces

Slack
Web
Semantic Reader
Jupyter

Interaction Store

Evaluation Harness

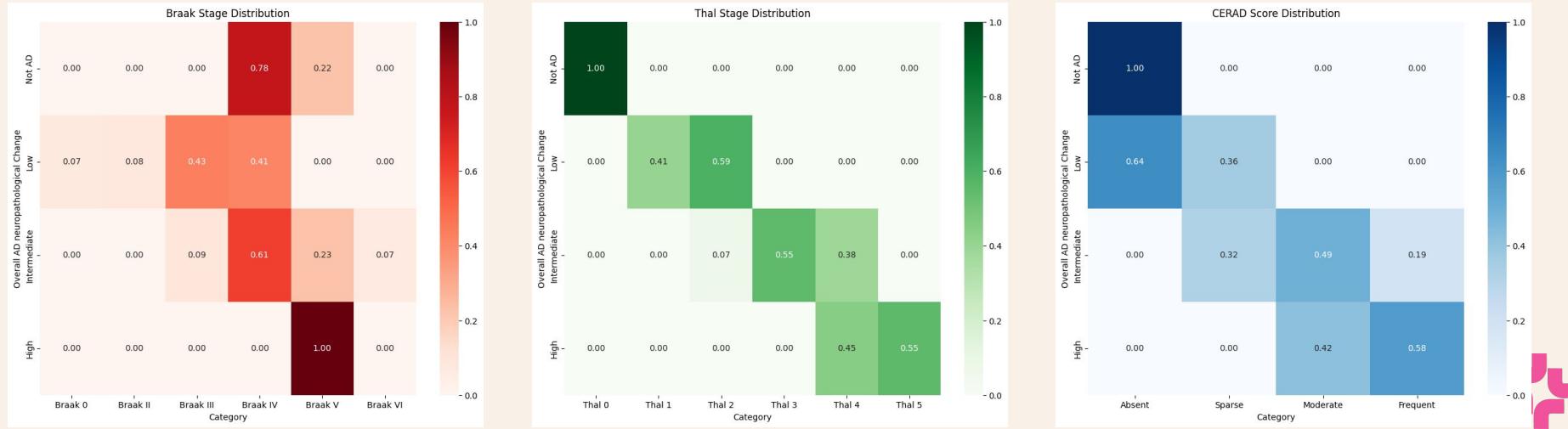
DataVoyager Modules - WIP

DataVoyager

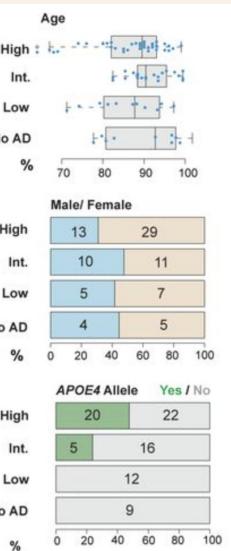
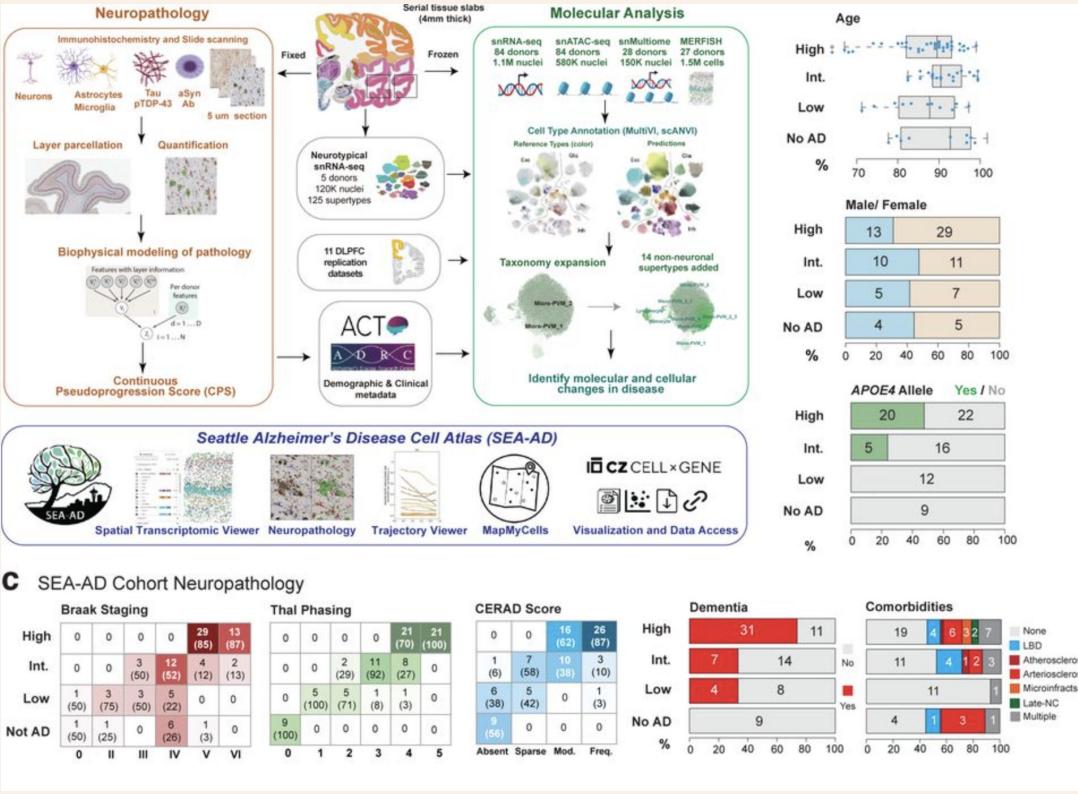
Bio Replication

C SEA-AD Cohort Neuropathology

Braak Staging							Thal Phasing					CERAD Score				
High	0	0	0	0	29 (85)	13 (87)	0	0	21 (70)	21 (100)	0	0	16 (62)	26 (87)		
	Int.	0	0	3 (50)	12 (52)	4 (12)	2 (13)	0	2 (29)	11 (92)	8 (27)	0	1 (6)	7 (58)	10 (38)	3 (10)
	Low	1 (50)	3 (75)	3 (50)	5 (22)	0	0	0	5 (100)	5 (71)	1 (8)	1 (3)	6 (38)	5 (42)	0	1 (3)
	Not AD	1 (50)	1 (25)	0	6 (26)	1 (3)	0	9 (100)	0	0	0	0	9 (56)	0	0	0
		0	II	III	IV	V	VI	0	1	2	3	4	5			



Integrated multimodal cell atlas of Alzheimer's disease



Data Input: Not sure of exact data used in paper or preprocessing. We currently used the following:

Donor Data

66 columns, and info includes metadata for each donor such as 'Primary Study Name', 'Age at Death', 'Sex', 'Race', 'CERAD score', 'Overall CAA Score', 'Highest Lewy Body Disease', 'Total Microinfarcts', 'Atherosclerosis', 'Arteriolosclerosis', 'LATE', 'RIN', and whether the donor is 'Severely Affected'.

MTG Data

394 columns, measurements related to AT8 positive areas in different layers of Grey matter, pTDP43 positive areas, and other neuropathological quantifications for each donor.

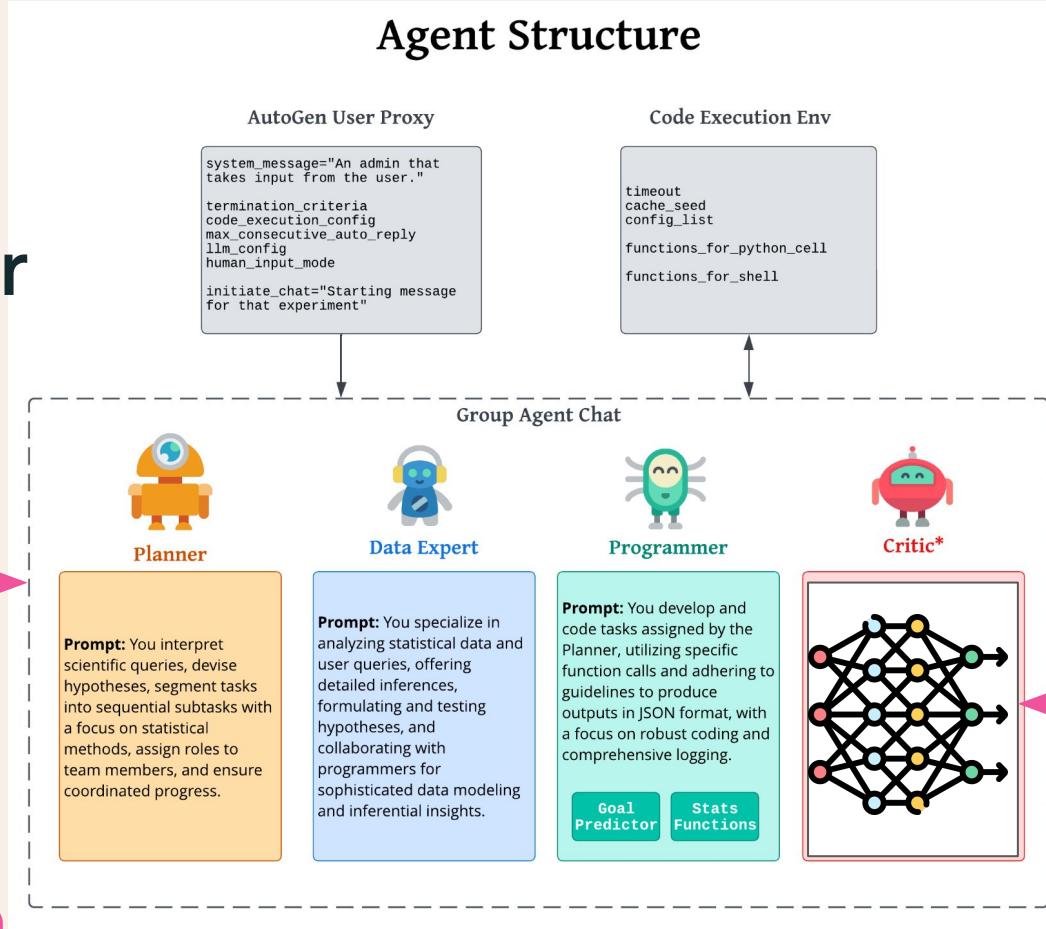


Memory with DataVoyager



Library functions,
User preferences,
Domain knowledge,
Explored hypotheses

Private to project
(can be proprietary info)



Finetuned
critic

Learned from
a big corpus/
Voluntarily
shared logs



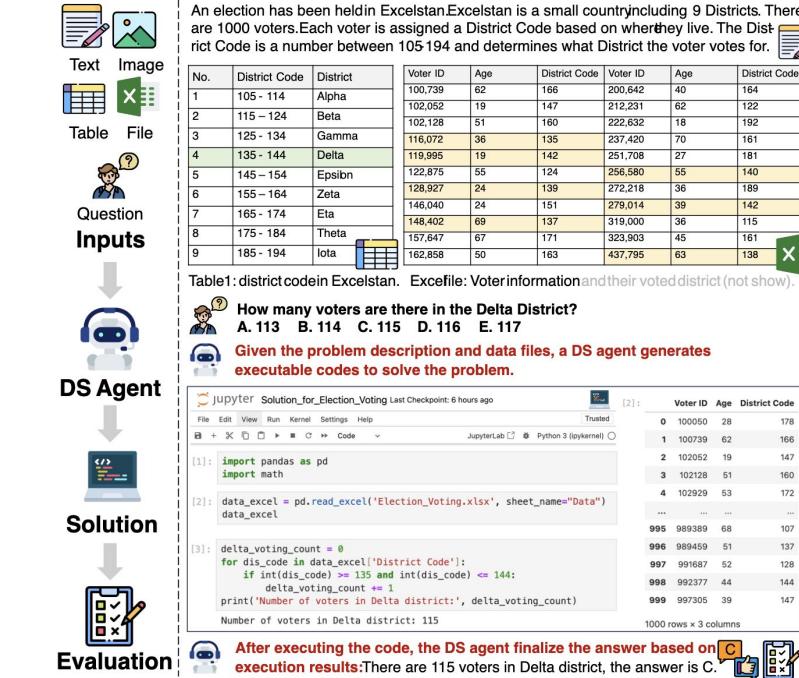
More Benchmarks

DSBench

<https://github.com/LiqiangJing/DSBench>

- Has files such as .txt, .xlsx
- Has metadata/background about the data
- MCQ
- Requires computation
- Autogen-based GPT-4 achieves 87.84% task success

DSBench benchmark consists of 466 data analysis tasks and 74 data modeling tasks. We evaluate several state-of-the-art LLMs, LVLMs, and agents, and find that our benchmark is challenging for the existing models.



More Benchmarks

DSEval

<https://github.com/MetaCopilot/dseval>

- Has files such as .txt, .xlsx
- Standard datasets like titanic, twitter
- QA with no options
- Requires computation
- GPT-4-based agents reach ~70% pass rate
- Has other evaluations/validators

Query

Calculate the population density for each country in 2023 and 2050. Result should be a new frame with "Country" as the index and "2023 Density" and "2050 Density" as the columns.

country	landAreaKm	pop2010	pop2023	pop2050
India	2973190	1.24E+09	1.43E+09	1.67E+09
China	9424703	1.35E+09	1.43E+09	1.31E+09
US	9147420	3.11E+08	3.4E+08	3.75E+08
...	...			

Code Interpreter

```
pd.DataFrame({
    'Country': pop['country'],
    '2023 Density': pop['pop2023'] / pop['landAreaKm'],
    '2050 Density': pop['pop2050'] / pop['landAreaKm']
}).set_index('Country')
```

Correct

CoML

```
pop['2023 Density'] = pop['pop2023'] / pop['landAreaKm']
growth = (pop['pop2023'] / pop['pop2010']) ** (1 / (2023-2010)) - 1
pop['2050 Population'] = pop['pop2023'] * (1+growth)**(2050-2023)
pop['2050 Density'] = pop['2050 Population'] / pop['landAreaKm']
pop[['country', '2023 Density', '2050 Density']].set_index('country')
```

Intact Violation + Wrong Output

```
pop = pop.set_index('country')
pop['2023 Density'] = pop['pop2023'] / pop['landAreaKm']
pop['2050 Density'] = pop['pop2050'] / pop['landAreaKm']
pop[['2023 Density', '2050 Density']]
```

Chapyter

Intact Violation + Presentation Error

```
pop = pop.set_index('country')
pop['2023 Density'] = pop['pop2023'] / pop['landAreaKm']
pop['2050 Density'] = pop['pop2050'] / pop['landAreaKm']
pop[['2023 Density', '2050 Density']]
```

Jupyter AI

Crash

```
dens_2023 = pop.div(pop['landAreaKm'], axis=0)
dens_2050 = pop.div(pop['landAreaKm'], axis=0)*(1+growth)**(2050-2023)
pd.DataFrame({ 'Country': pop['country'],
    '2023 Density': density_2023,
    '2050 Density': density_2050})
```

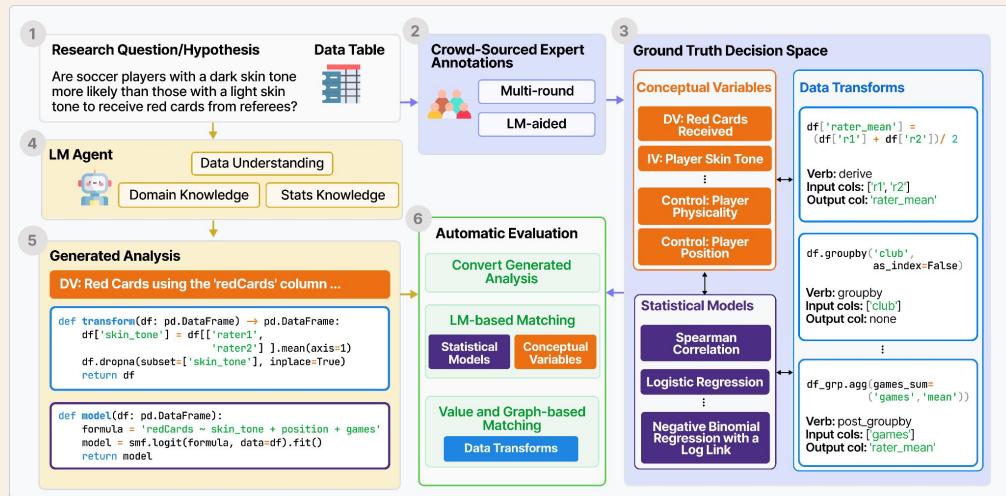


More Benchmarks

BLADE

https://github.com/behavioral-data/BLAD_E/tree/main

- Has files, .csvs
- Has metadata about the datasets
- Research Q/Hypothesis as a goal (most in the format: Is this true?)
- MCQs
- Requires computation



More Benchmarks

QRData

<https://xxxiaol.github.io/QRData/>

- Subset of benchmark has files, .csvs
- MCQ
- Requires computation
- GPT-4 achieves ~60% accuracy

Data Description

The CSV file `ihdp.csv` contains data obtained from the Infant Health and Development Program (IHDP). The study is designed to evaluate the **effect of home visit from specialist doctors on the cognitive test scores of premature infants**. The confounders x (x_1 - x_{25}) correspond to collected measurements of the children and their mothers ...

Question

What is the **Average Treatment Effect (ATE)** from t to y ? Please round the final answer to the nearest hundredth.

Correct Reasoning Steps:

1. Check rows of the dataset to understand its structure

```
import pandas as pd
data = pd.read_csv('ihdp.csv')
print(data.head())
```

Sandbox Execution Results:

t	y	x1	x2
1	5.60	-0.53	-0.34
0	6.88	-1.74	-1.80
0	3.00	-0.81	-0.20
...

2. Build a causal model based on the data description

```
from dowhy import CausalModel
common_causes = ['x1', ..., 'x25']
ihdp_model = CausalModel(
    data=data, treatment='t', outcome='y',
    common_causes=common_causes
)
```

3. Recall related method and apply to this scenario

ATE can be estimated using propensity score weighting:

```
...
ihdp_estimate = ihdp_model.estimate_effect(
    ihdp_identified_estimand,
    method_name="backdoor.propensity_score_weighting"
)
print('Estimated effect:', ihdp_estimate.value)
```

Sandbox Execution Results:

Estimated effect: 4.02

4. Run refutation test to validate the estimate

The estimate should not change if we add an independent random variable as a common cause to the dataset.

```
ihdp_refute_random_common_cause = ihdp_model.refute_estimate(
    ihdp_identified_estimand, ihdp_estimate,
    method_name="random_common_cause"
)
print('New effect:', ihdp_refute_random_common_cause.new_effect)
```

Sandbox Execution Results:

New effect: 4.02

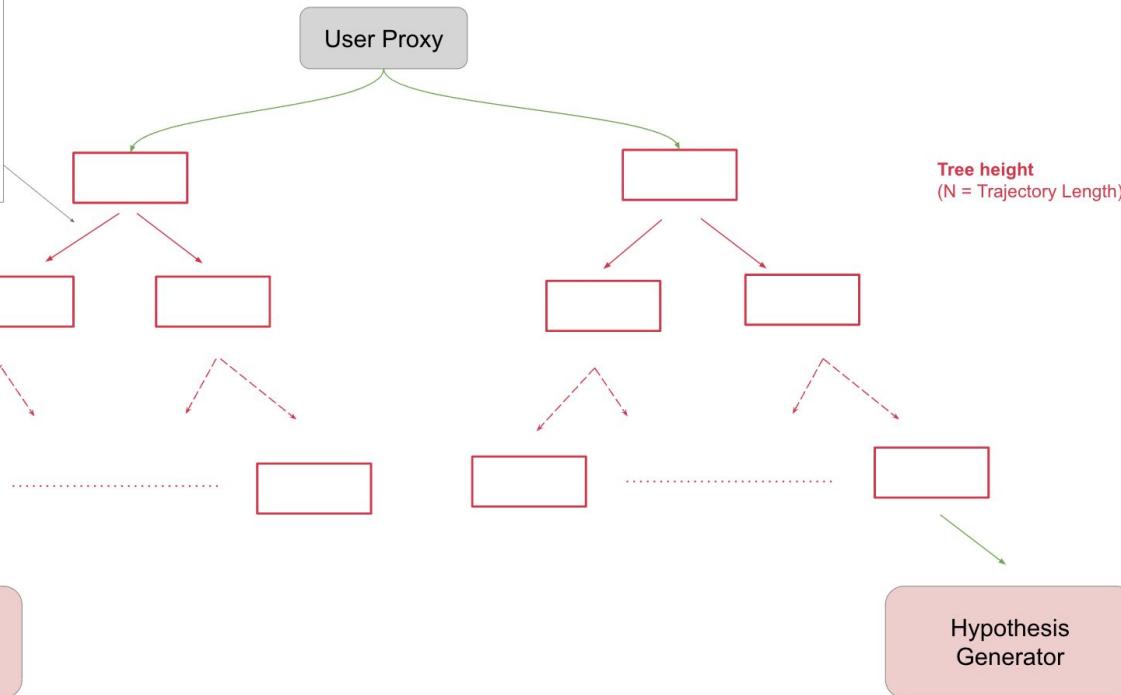
Final Answer: 4.02



AutoDataVoyager - Exploration Trajectories

Exploration trees

Instead of focusing on a single experiment, the `experiment_generator` proposes multiple ($k = 2$) experiments at each step, enabling a tree-based exploration approach.



What happens with petabytes of data?

Or millions of lines of code?

Projects Ideas

Draft Project Ideas

Code Gen

Option 1: build from scratch **[easy]**

Option 2: fork existing tools and improve them
[med/hard]

Data Science Agent

Option 1: build from scratch **[easy]**

Option 2: use our datavoyager-core and add specific features to it **[hard]**

Other

RAG at industrial scale

RAG for specific domains

Reducing hallucinations in RAG

DPO/ PPO/ finetune a model for a particular novel problem

RAG + agents on Indian Gov docs in Indian languages

Appendix

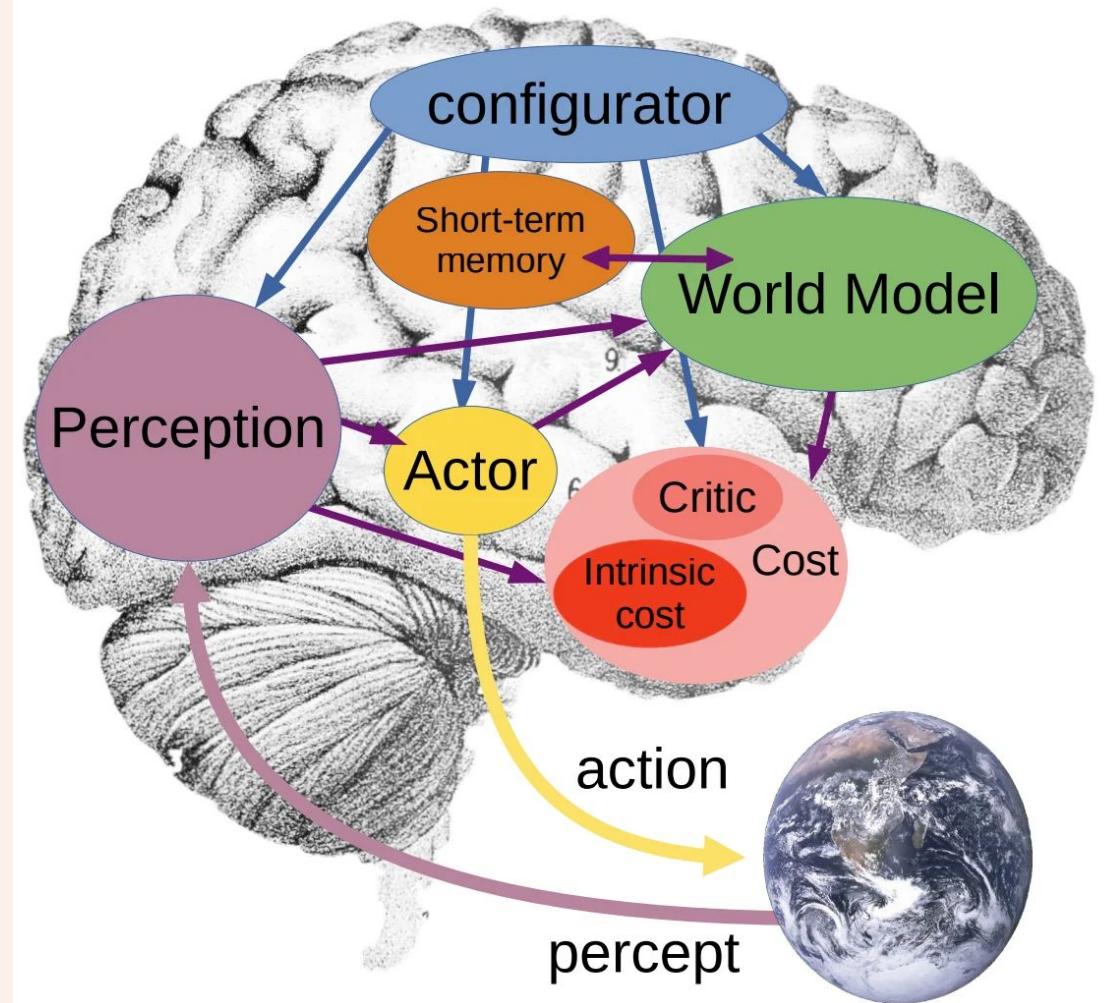
Walkthrough of the DataVoyager system

https://x.com/surana_h/status/178609
7912147239157

<https://x.com/mbodhisattwa/status/1761061506127655244>

V Jepa

By Meta/ Yann Le Cunn



Thank you!

