

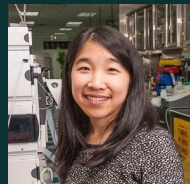
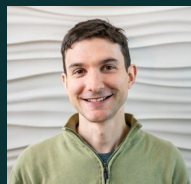
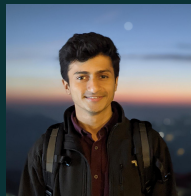
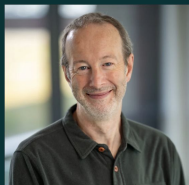
FOSS India

DiscoveryBench: Data-Driven Scientific Discovery with LLM Agents



Harshit Surana

Allen Institute for Artificial Intelligence & OpenLocus



Structure & Some Requests

1. Scientific Discovery with LLMs
 2. DiscoveryBench/ Data Driven Discovery
 3. Best Practices in Building AI at Scale
- Important to have interactivity
 - Feel free to interrupt for quick clarifications.
 - Will stop at the end of each section for **QnA**
 - If audio issues arise, **please drop messages** in the chat.

Scientific Discovery with LLMs

“Superintelligent tools could
massively accelerate scientific
discovery and innovation well
beyond what we are capable of
doing on our own, and in turn
massively increase abundance
and prosperity.”

- *Sam Altman, OpenAI*



NOBELPRISET I KEMI 2024 THE NOBEL PRIZE IN CHEMISTRY 2024



KUNGL.
VETENSKAPS-
AKADEMIEN

THE ROYAL SWEDISH ACADEMY OF SCIENCES



Photo: University of Washington

David Baker
University of Washington
USA

"för datorbaserad proteindesign"

"for computational protein design"



Photo: The Royal Society

Demis Hassabis
Google DeepMind
United Kingdom

"för proteinstrukturprediktion"

"for protein structure prediction"



Photo: IBM Research

John M. Jumper
Google DeepMind
United Kingdom

Broad Areas of Scientific Discovery where LLMs are Helping!

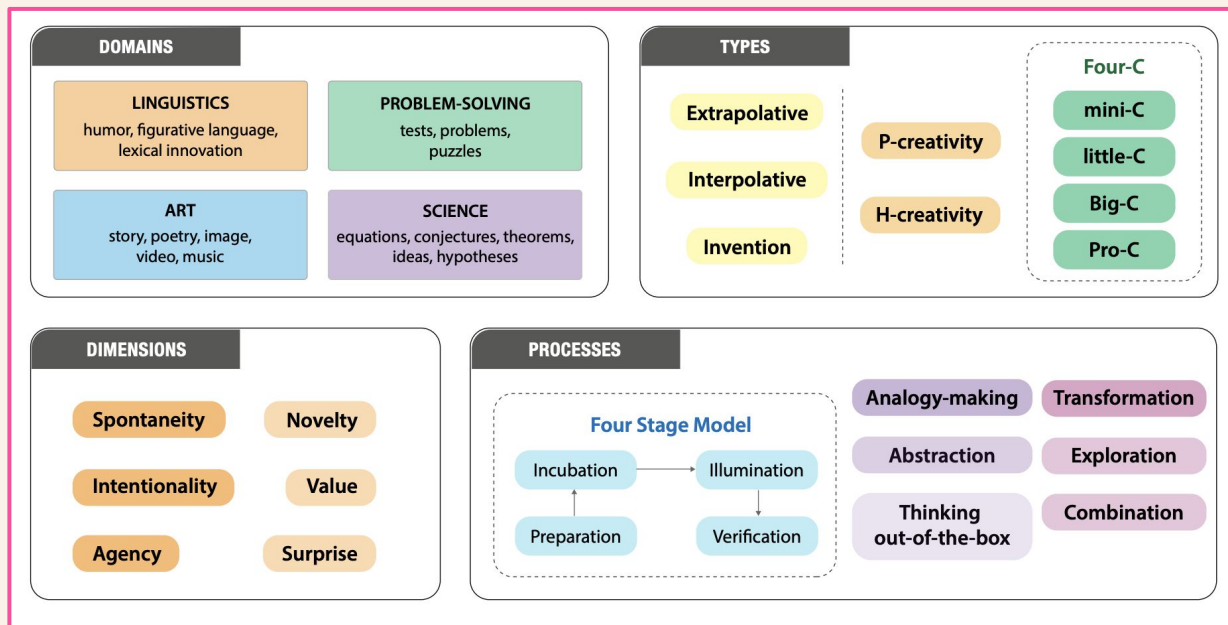
Laboratory Tools

Exp. Chemistry

Equation Discovery

Theorem Proving

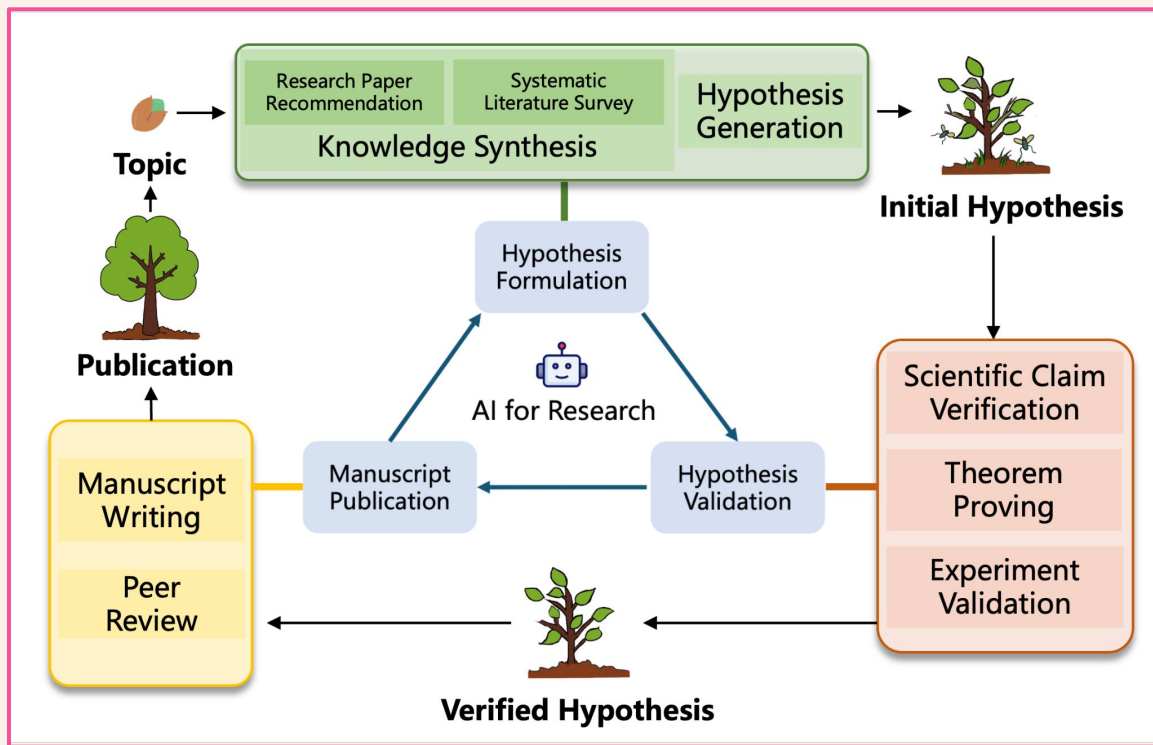
Observational Data*



What are the steps in any
Research Process?

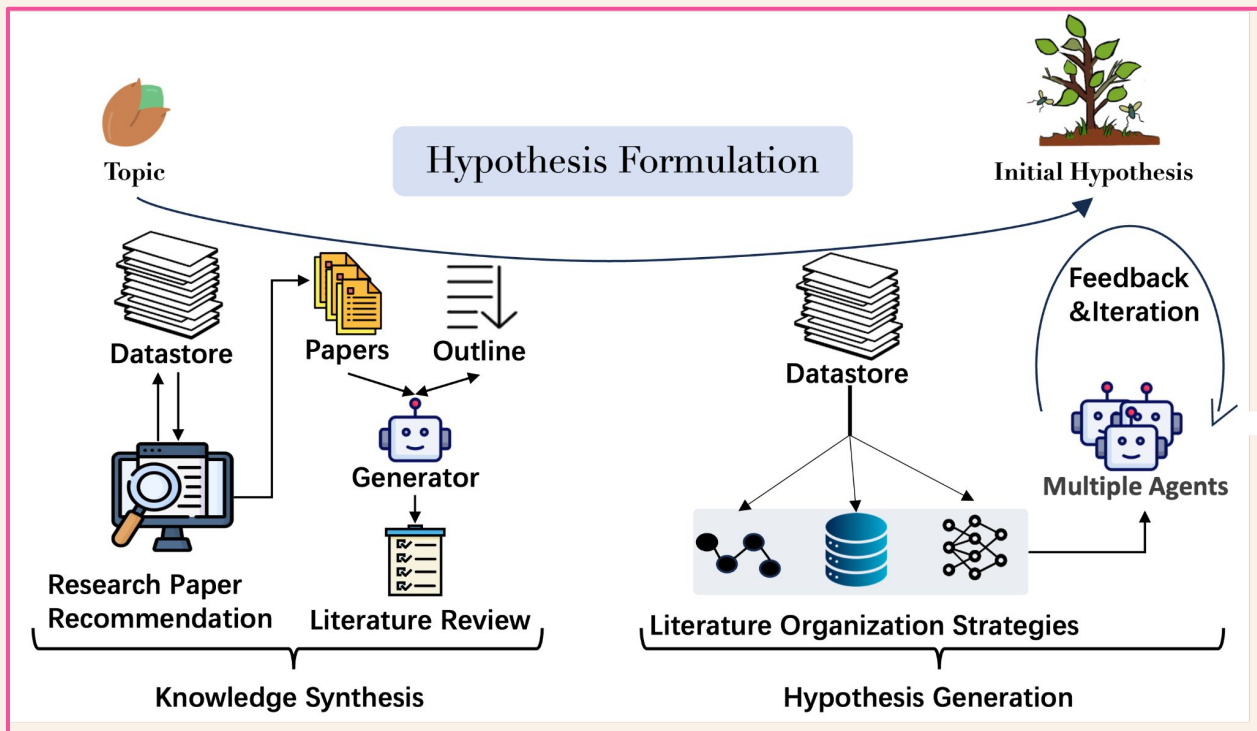
What are different
methods of verifying a
Hypothesis?

Topic → Hypothesis → Paper



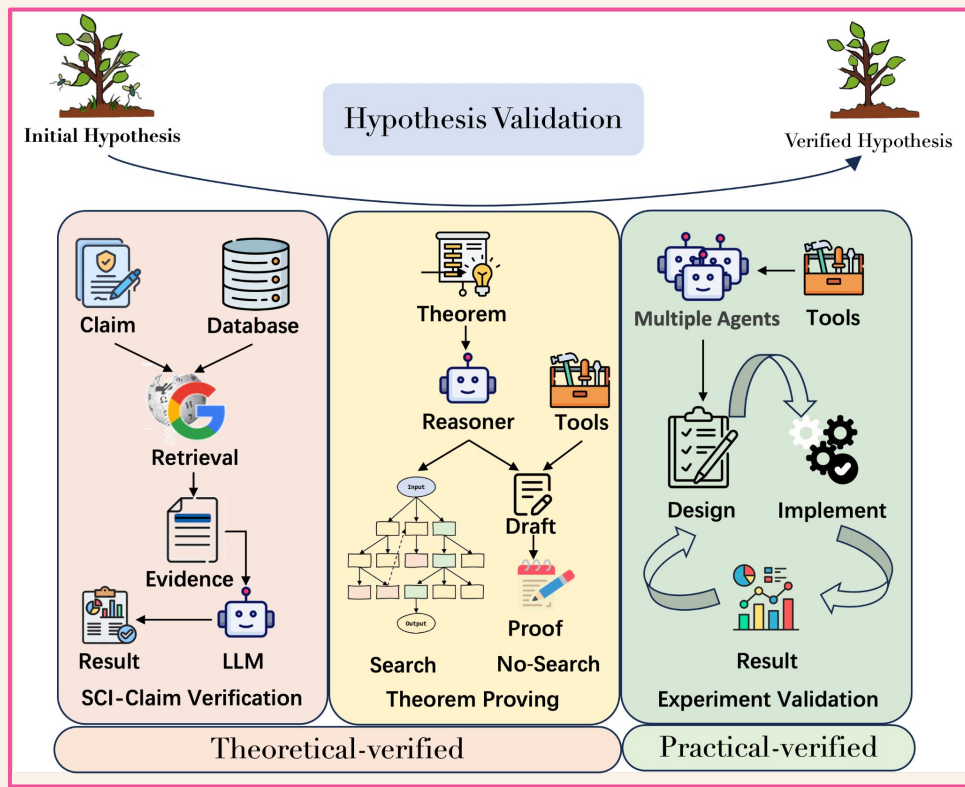
From Hypothesis to Publication: A Comprehensive Survey of AI-Driven Research Support Systems, Arxiv 2025

Hypothesis Formulation



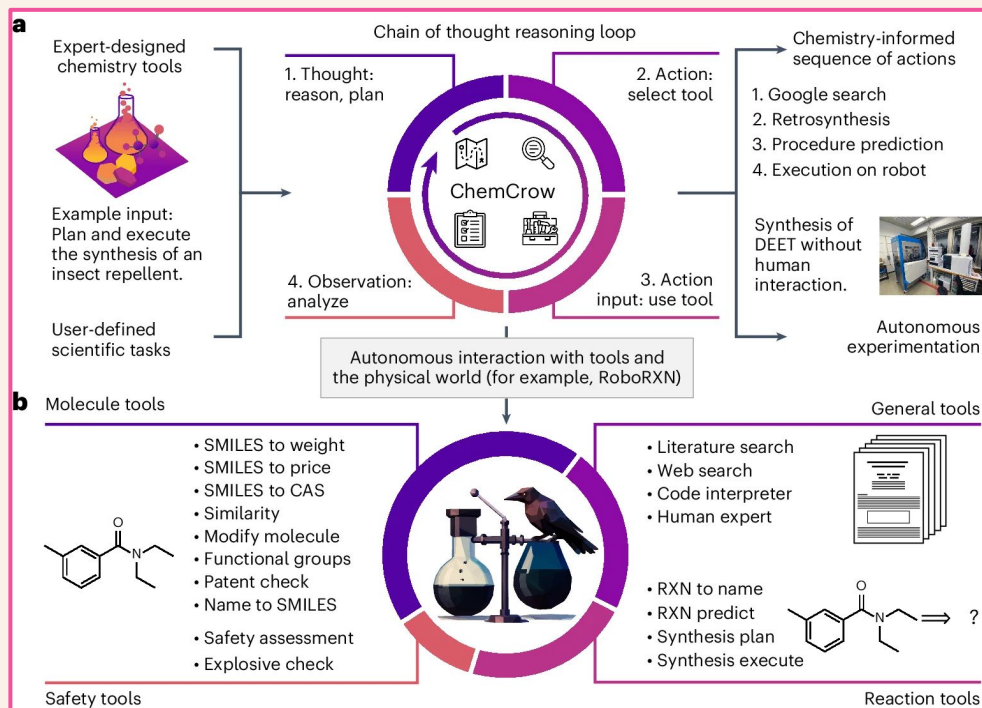
From Hypothesis to Publication: A Comprehensive Survey of AI-Driven Research Support Systems, Arxiv 2025

Hypothesis Validation



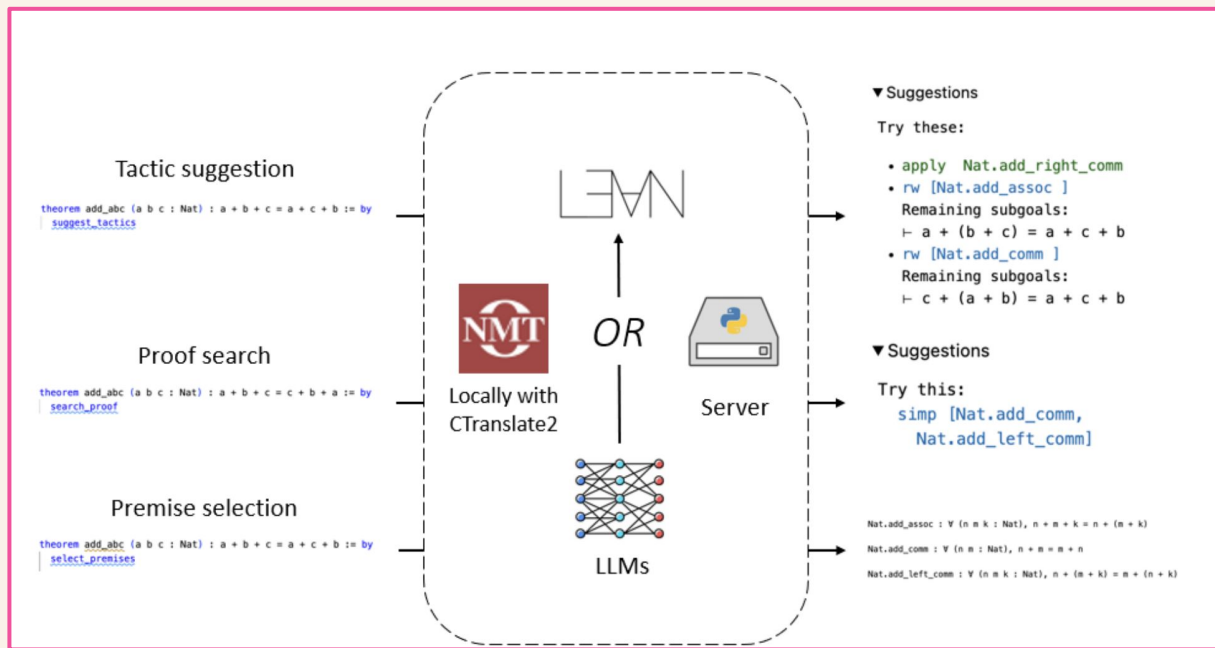
From Hypothesis to Publication: A Comprehensive Survey of AI-Driven Research Support Systems, Arxiv 2025

LLMs, Lab Tools & Beyond



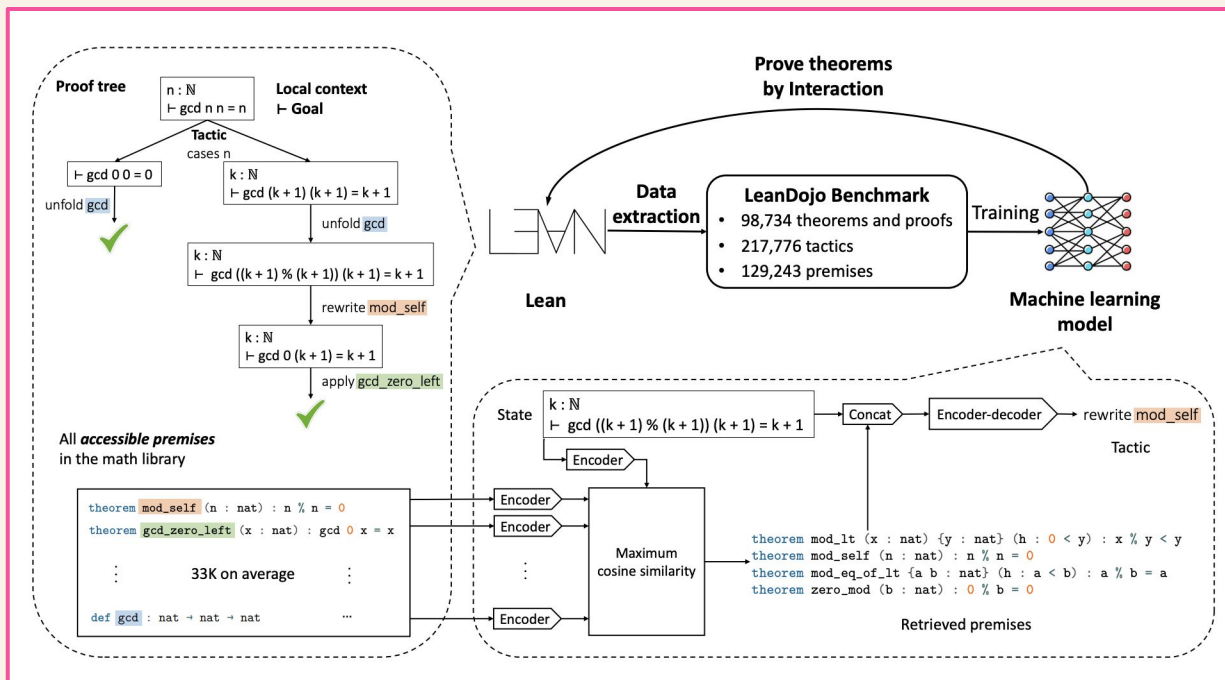
ChemCrow: Augmenting large language models with chemistry tools, Nature 2024

Theorem Proving Copilot



Towards Large Language Models as Copilots for Theorem Proving in Lean, NuerIPS 2023

Theorem Proving Env with LLMs + Retrieval



LeanDojo: Theorem Proving with Retrieval-Augmented Language Models, NuerIPS 2023

Methods of Science

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



Nurses'
Health Study —



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

Data Driven Scientific Discovery with LLMs

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, agriforest	● 0.3
relationship	reduced	increased	● 0.0
Final Score: 0.21			

W. H. Thompson and S. Skau. On the scope of scientific hypotheses. Royal Society Open Science, 2023



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',
                    'ABILITY', '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=[
    'ASVAB - Arithmetic Reasoning Z Score (rounded), 1981',
    'ASVAB - Word Knowledge Z Score (rounded), 1981',
    'ASVAB - Paragraph Comprehension Z Score (rounded), 1981',
    'ASVAB - Mathematics Knowledge Z Score (rounded), 1981'
]

df['all scores exist']=df[score_cols].gt(0).all(axis=1)

df['ABILITY'] =
df['ASVAB - Arithmetic Reasoning Z Score (rounded), 1981'] +
df['ASVAB - Word Knowledge Z Score (rounded), 1981'] +
df['ASVAB - Paragraph Comprehension Z Score (rounded), 1981'] +
df['ASVAB - Mathematics Knowledge Z Score (rounded), 1981']

df['BA DEGREE COMPLETED'] = df['Highest grade completed, 1979'].gt(14)
```

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']
)
```

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)]

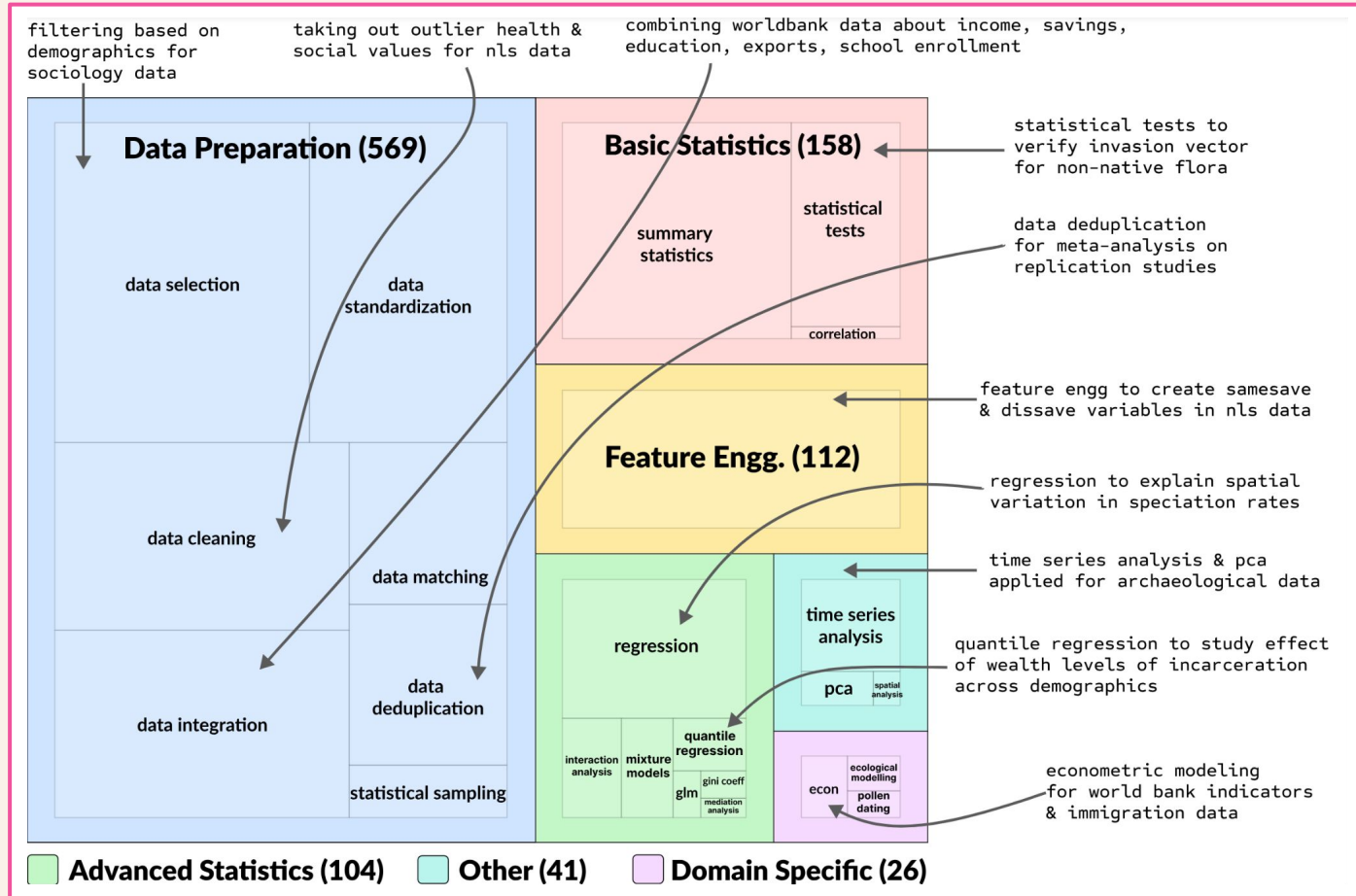
sub_dataset['BA DEGREE COMPLETED'] =
    sub_dataset['BA DEGREE COMPLETED'].astype(int)

X = sub_dataset[['SES']]
X = sm.add_constant(X)
y = sub_dataset['BA DEGREE COMPLETED']

model = sm.Logit(y, X)
result = model.fit()

result.summary()
```

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.

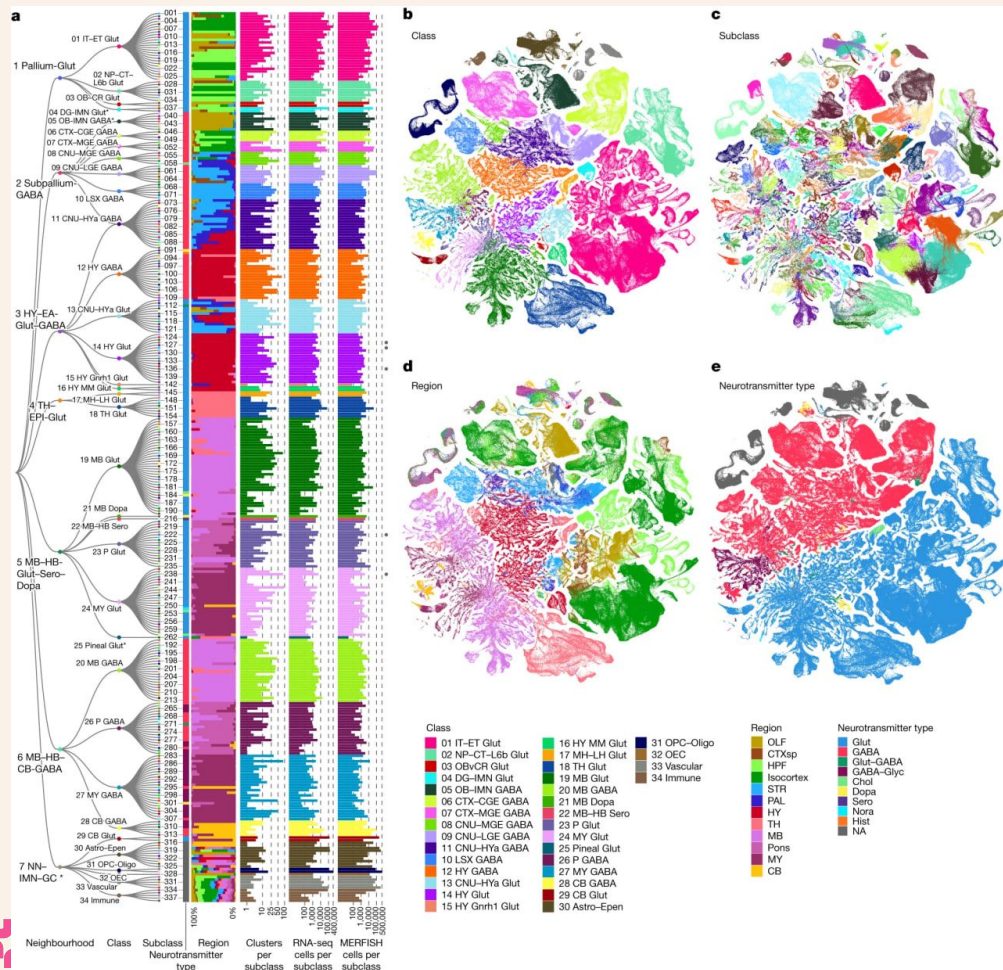
It has four components: planner, code generator, data analysis, and critic, that orchestrate the discovery process.

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.

A NIMHANS (neuro) Example?

Culmination of decades of work summarized in an image



Landmark Mouse Brain Paper (Ai1)

DataVoyager Prompt:

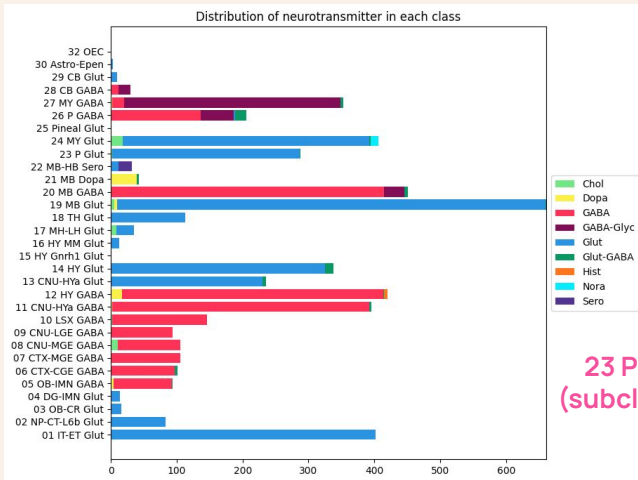
1. Find the unique classes present in the data.
2. Select the "20 MB GABA" class and plot the distribution of subclasses and neurotransmitters.
3. Select the "23 P Glut" class and plot the distribution of subclasses and neurotransmitters.
4. Plot the supertypes and neurotransmitters distribution for the same filtered class.
5. Interpret the unique supertypes for this filtered class.

A high-resolution transcriptomic and spatial atlas of cell types in the whole mouse brain

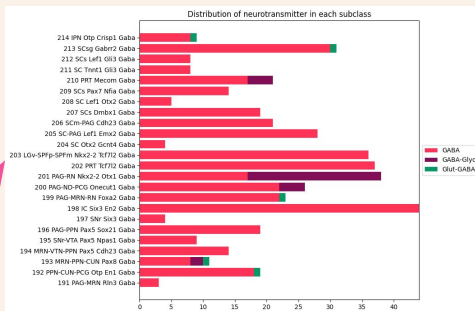
<https://www.nature.com/articles/s41586-023-06812-z>

DataVoyager Traces for WMB Data

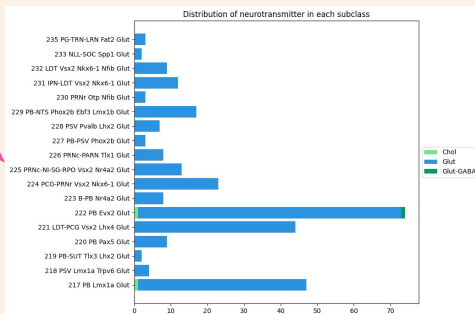
WMB Classes by nt



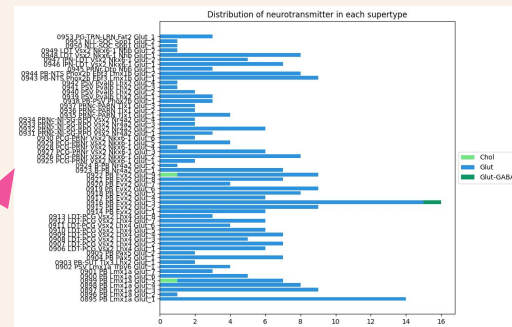
20 MB GABA (subclasses)



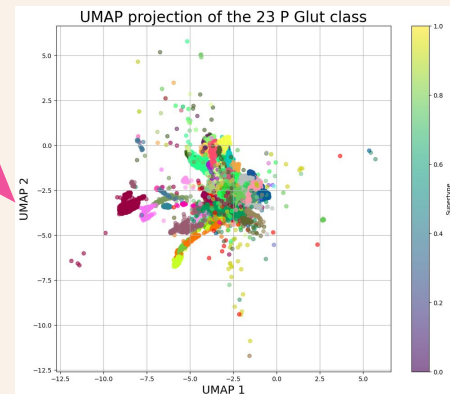
23 P Glut (subclasses)



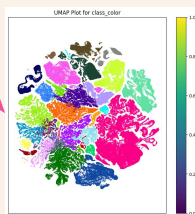
Supertypes



Projection



Projection



Integrated multimodal cell atlas of Alzheimer's disease

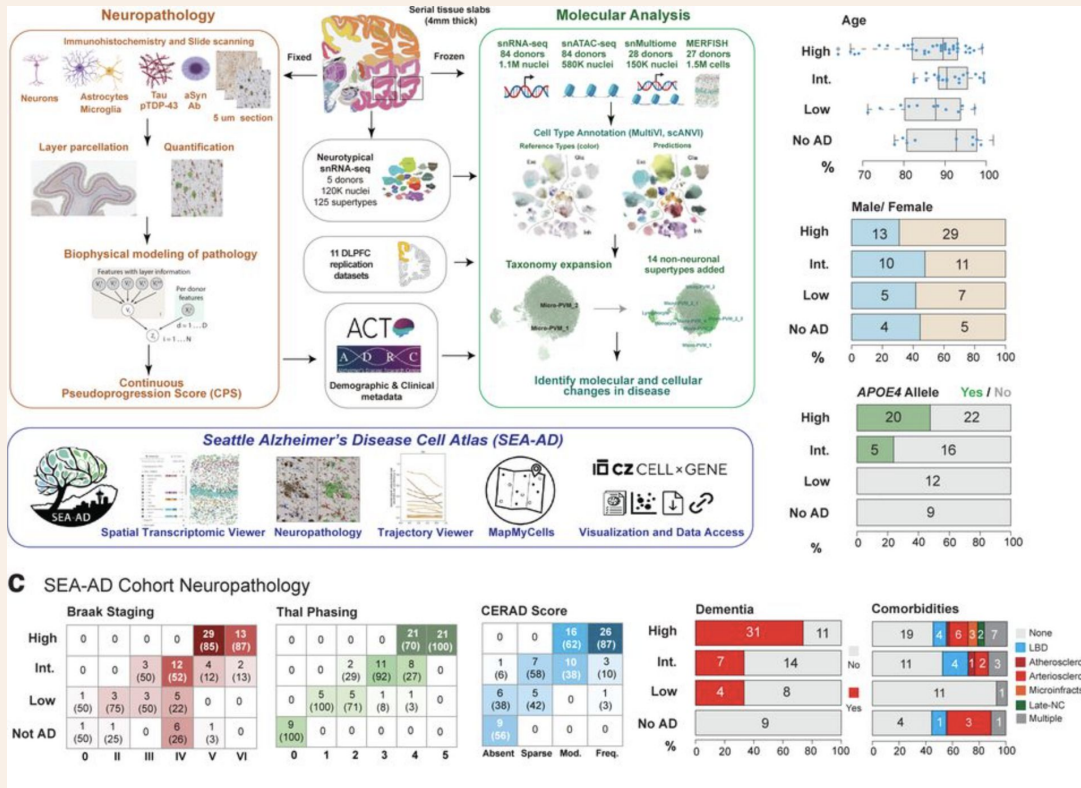
Data Input: Not sure of exact data used in paper or preprocessing. We currently used the following:

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.



DataVoyager is a part of Asta - a Science Copilot

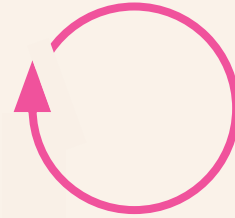
1) Collaborates naturally with human scientists

2) Uses tools on humans' behalf

3) Learns & improves over time.

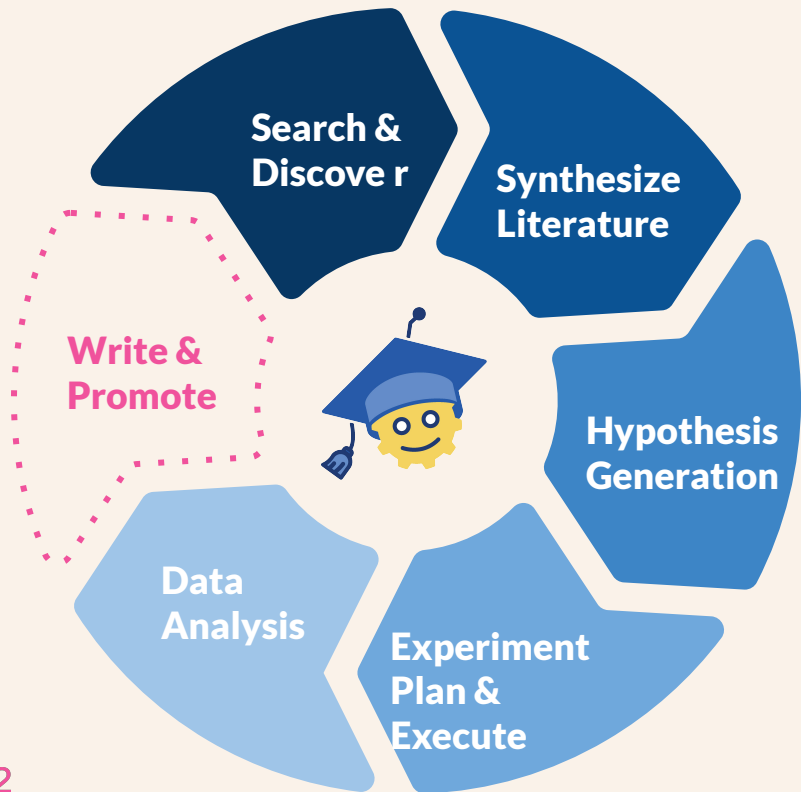


Slack
Semantic Reader
Jupyter



... with Infra. for Personalization & Optimization

Research Functions



Surfaces

Slack
Web
Semantic Reader
Jupyter

Interaction Store

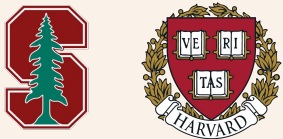
Evaluation Harness

Some Accolades

New Directions in the Area Influenced by our Work



BLADE: Benchmarking Language Model Agents for Data-Driven Science



POPPER: Automated Hypothesis Validation with Agentic Sequential Falsifications



ScienceAgentBench: Toward Rigorous Assessment of Language Agents for Data-Driven Scientific Discovery



QR Data: Are LLMs Capable of Data-based Statistical and Causal Reasoning? Benchmarking Advanced Quantitative Reasoning with Data

Recognized with other major works including the aforementioned Nobel Prize in 2024

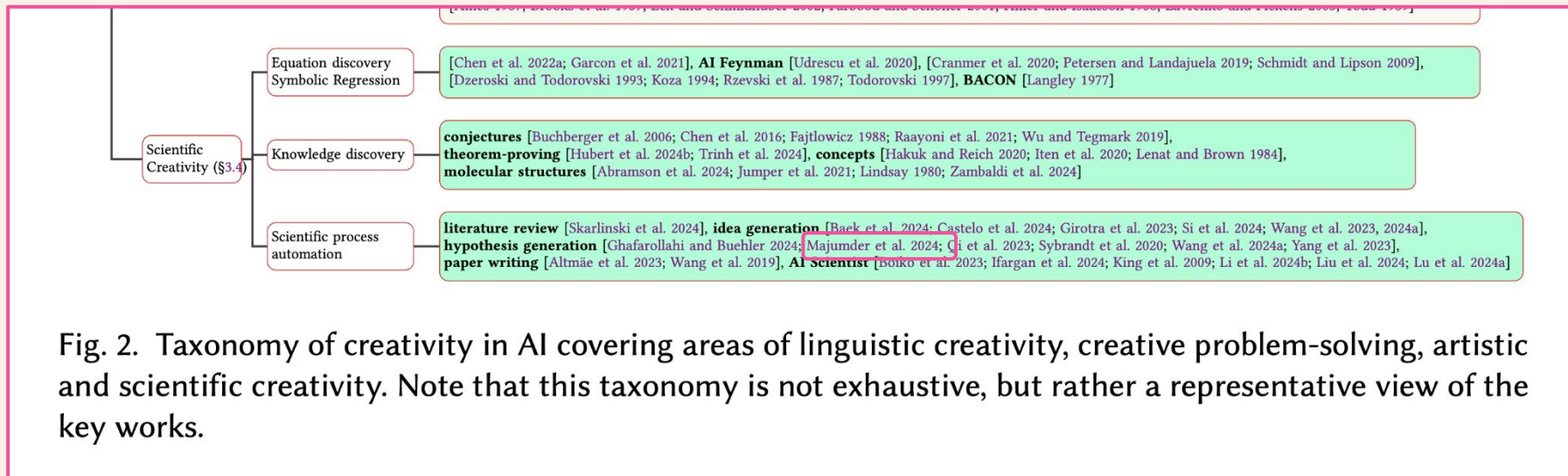


Fig. 2. Taxonomy of creativity in AI covering areas of linguistic creativity, creative problem-solving, artistic and scientific creativity. Note that this taxonomy is not exhaustive, but rather a representative view of the key works.

Learnings

Learnings for in LLMs + Open Source

Instrument the **Data Pipeline**, Not Just the Model

Building **Evaluations** that Actually Matter

Adopting & **Extending Frameworks** (LangChain, AutoGen, etc.)

Treat **Prompts** and Pipelines as **Code Assets**

Exploit Small Models First, Big Models Last

Exploit Big Models First, Small Models Last

A/B (or **Ablations**) Everything Behind a Feature Flag/ Config

How to contribute to fundamental AI from India?

What happens with
petabytes of data?

Or millions of lines
of code?

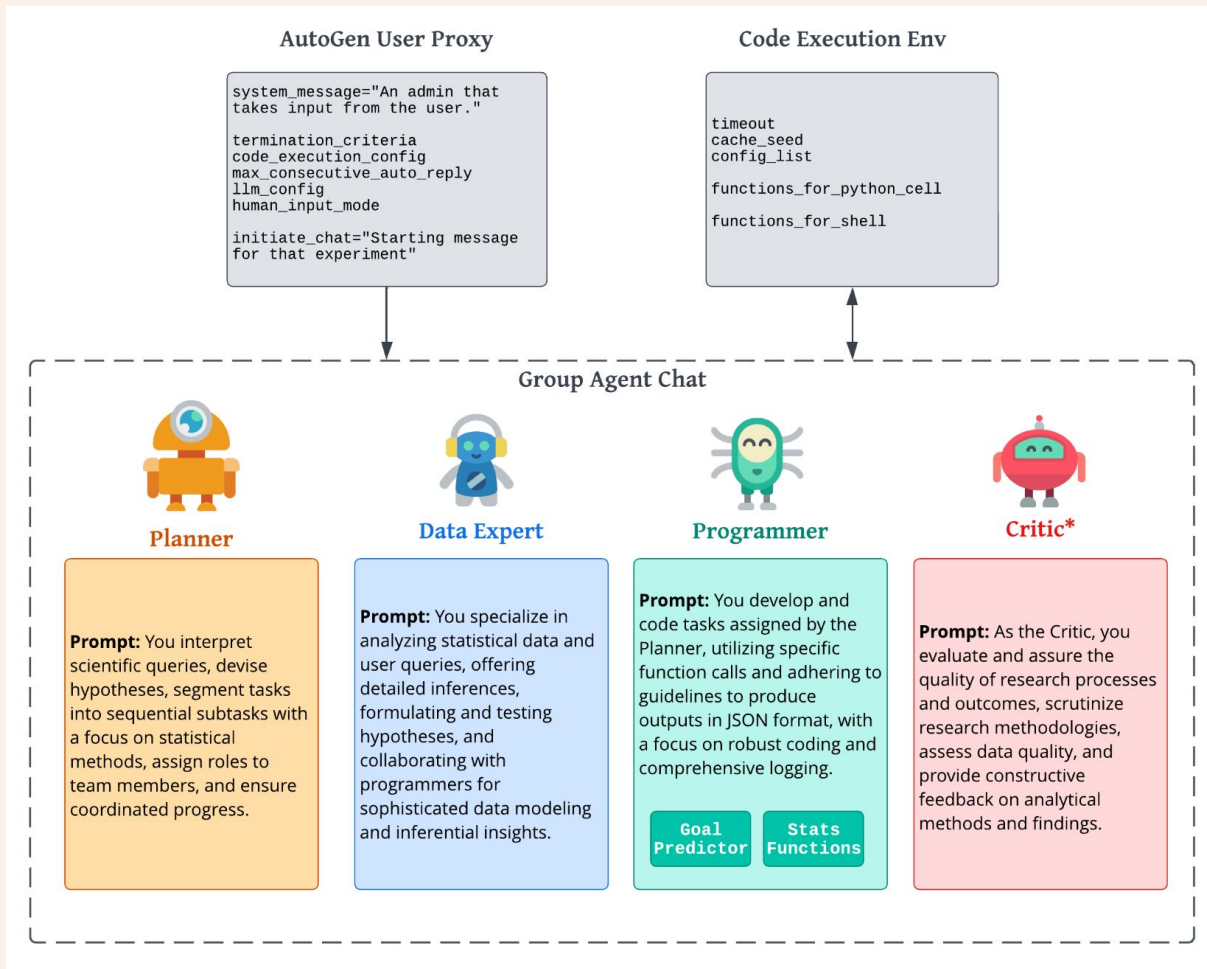
Appendix

DataVoyager

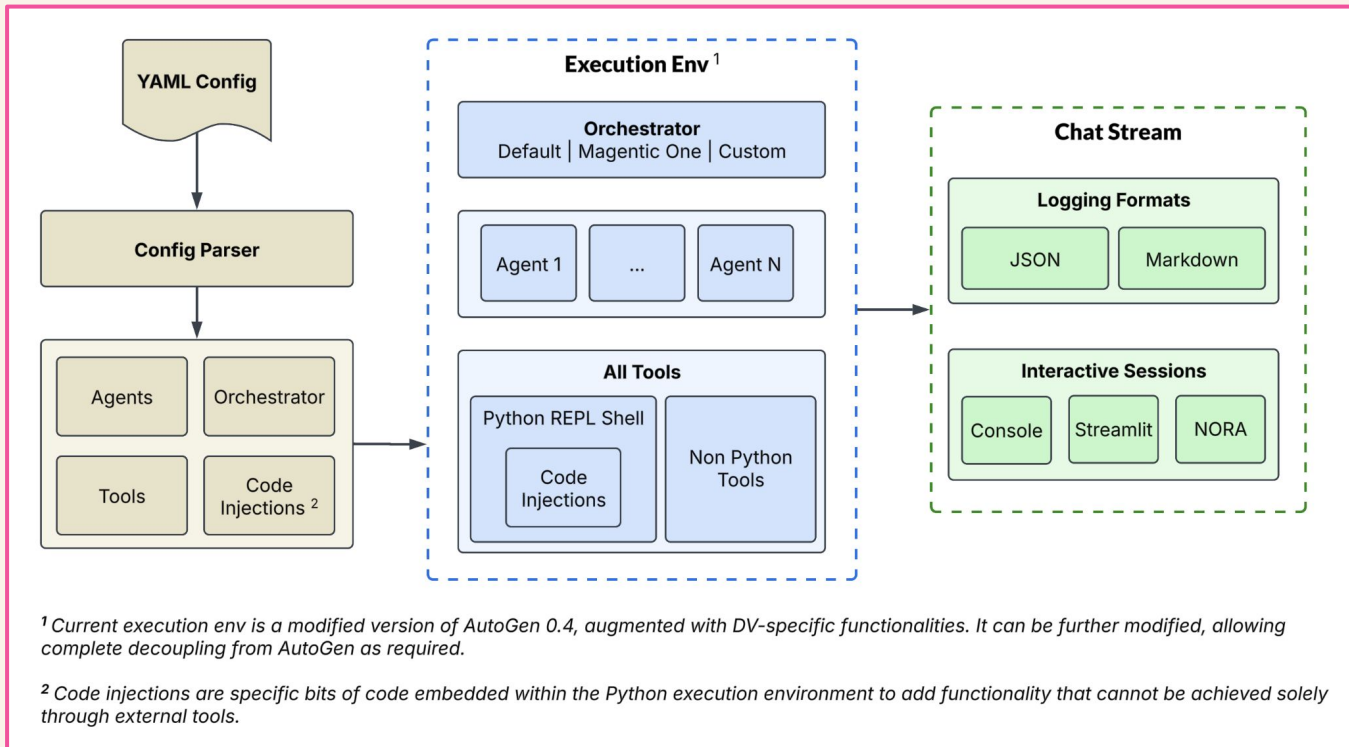
All of the subagents and the controller is based on with a specialized prompt.

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**.



DataVoyager Core Architecture



LLM Benchmarks

- DataGLUE
- NORA - End to End Science
- Bio - Neuro/ Immunology

Better Agents/ Models for

- Data Driven Discovery
- End to End Science
- Neuro/ Immunology/ Genetics

Collabs

Auto-exploration with UMass/
Andrew McCollum group

Better memory for coding/
agents with Zurich U

Better Cost Efficiency

Pareto optimal model usage/
model routing for DDD

100X+ difference b/w cheaper
& advanced models

Related Projects List

Augmenting code editing with AI features

- Fork open-source AI code editors like melty, zed or pearai or aider
- Add new features like AI diagramming for architecture etc. in them
- <https://aider.chat/docs/leaderboards/>

Enterprise heavy requirements

- Add guardrails to prevent certain data leakage
- Serve data while making sure it is not hallucinated
- Can be RAG or database query driven

Other Ideas

- AutoRAG
- RAG + agents on domain specific areas like medicine
- RAG + agents for indian docs
 - Analyze data
 - Make it available in Indian languages

More Benchmarks

DSBench

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

- Has files such as .txt, .xlxs
- 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.



Text Image



Table File



Question Inputs



DS Agent



Solution



Evaluation

An election has been held in Excelstan. Excelstan is a small country including 9 Districts. There are 1000 voters. Each voter is assigned a District Code based on where they live. The District Code is a number between 105-194 and determines what District the voter votes for.

No.	District Code	District	Voter ID	Age	District Code	Voter ID	Age	District Code
1	105 - 114	Alpha	100,739	62	166	200,642	40	164
2	115 - 124	Beta	102,052	19	147	212,231	62	122
3	125 - 134	Gamma	102,128	51	160	222,632	18	192
4	135 - 144	Delta	116,072	36	135	237,420	70	161
5	145 - 154	Epsilon	119,995	19	142	251,708	27	181
6	155 - 164	Zeta	122,875	55	124	256,580	55	140
7	165 - 174	Eta	128,927	24	139	272,218	36	189
8	175 - 184	Theta	146,040	24	151	279,014	39	142
9	185 - 194	Iota	148,402	69	137	319,000	36	115
			157,647	67	171	323,903	45	161
			162,858	50	163	437,795	63	138

Table1: district code in Excelstan. Excefile: Voter information and their voted district (not show).

How many voters are there in the Delta District?
A. 113 B. 114 C. 115 D. 116 E. 117

Given the problem description and data files, a DS agent generates executable codes to solve the problem.

```
jupyter Solution_for_Election_Voting Last Checkpoint: 6 hours ago
File Edit View Run Kernel Settings Help
JupyterLab Python 3 (ipykernel)

[1]: import pandas as pd
import math

[2]: data_excel = pd.read_excel('Election_Voting.xlsx', sheet_name='Data')
data_excel

[3]: delta_voting_count = 0
for dis_code in data_excel['District Code']:
    if int(dis_code) >= 135 and int(dis_code) <= 144:
        delta_voting_count += 1
print('Number of voters in Delta district:', delta_voting_count)

Number of voters in Delta district: 115
```

After executing the code, the DS agent finalizes the answer based on execution results: There are 115 voters in Delta district, the answer is C.



More Benchmarks

DSEval

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

- Has files such as .txt, .xlxs
- 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	landArea Km	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

Correct

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

CoML

Intact Violation + Wrong Output

```
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')
```

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': dens_2023,
    '2050 Density': dens_2050})
```

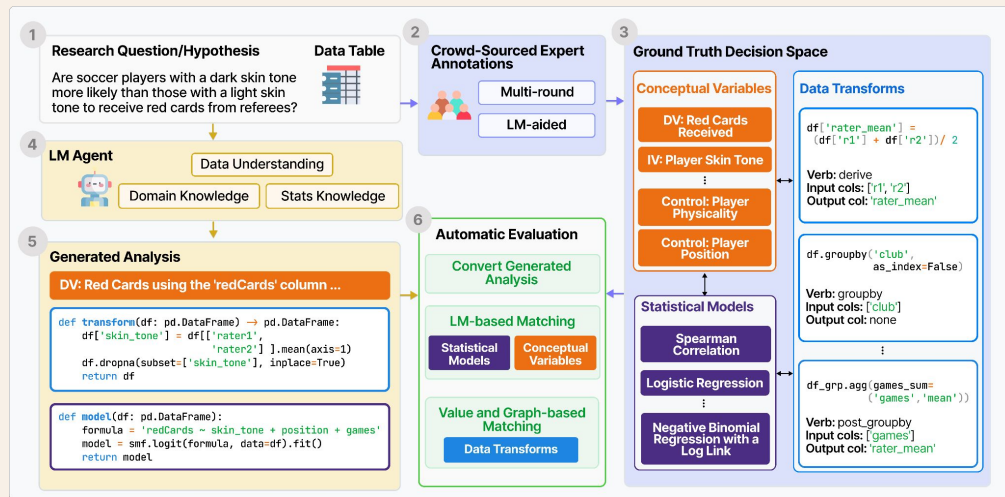


More Benchmarks

BLADE

<https://github.com/behavioral-data/BLADE/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://xxxiao.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 (x1-x25)` 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

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)
```

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)
```

New effect: 4.02

Final Answer: 4.02



Memory with DataVoyager

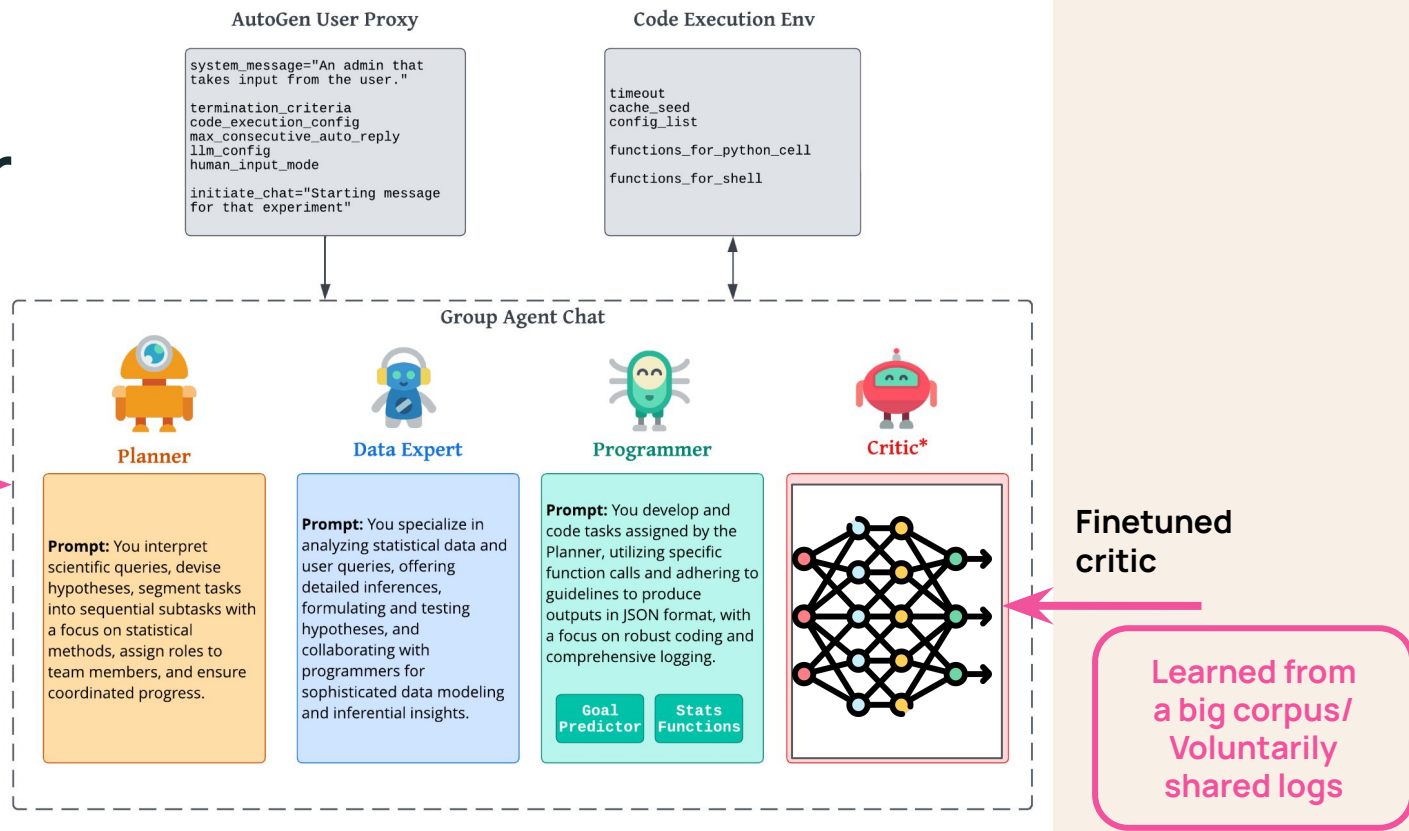
Project Memory



Library functions,
User preferences,
Domain knowledge,
Explored hypotheses

Private to project
(can be proprietary info)

Agent Structure



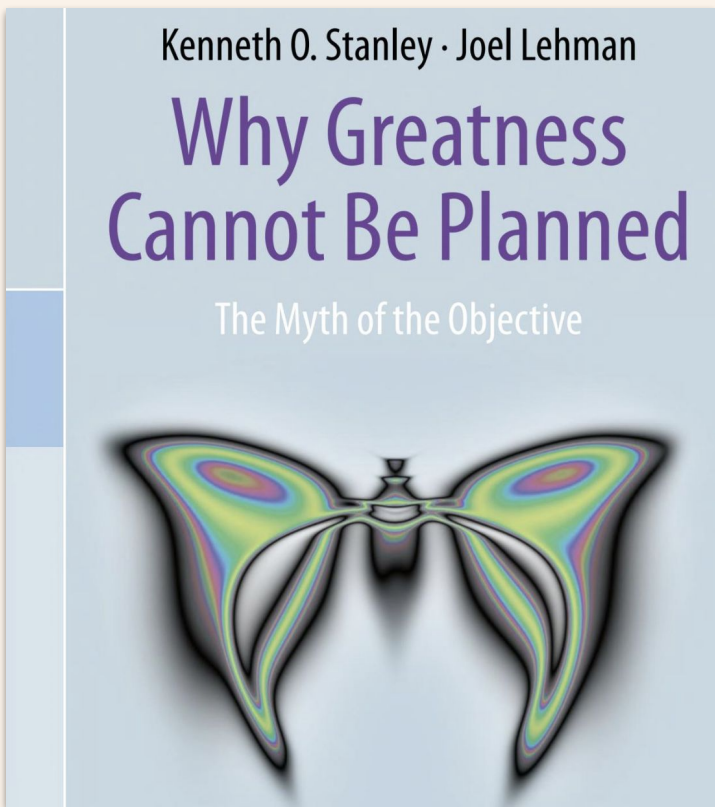
But, what if the goal is **unknown**?

Can open-ended exploration lead to useful discoveries?

- No explicit research **question** or discovery **goal**
- Only given a dataset
- Optionally, a set of known hypotheses

- Should align with human notions of “**interestingness**”
 - Discoveries made may answer human-posed research queries in the future
- Serves as a **building block for future discoveries**

Not **just** a philosophical point



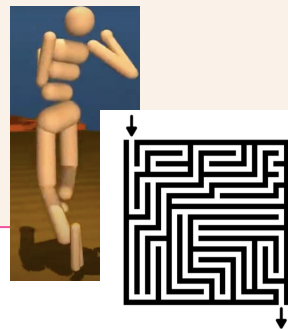
Could address **practical limitations** in **goal-driven discovery**:

- **Low dataset coverage:** Performs shallow analyses; Struggles to sufficiently cover the dataset; misses promising experiment designs.
- **Low diversity:** Repeated execution does not produce diverse outputs. Inference-time scaling laws do not seem to hold in our discovery setting.

Open-endedness in data-driven discovery

Benefits:

- **Immediate feedback:** Closed loop of proposing and executing experiments on a given dataset allows for a continual, autonomous process to collect verifiable hypotheses.
- **Expanding search space:** As hypotheses are collected, new opportunities arise to further explore known hypotheses, combine them, or continue sampling new ones.
 - Solutions quickly plateau in static search spaces. E.g., maze solving, bipedal walking.



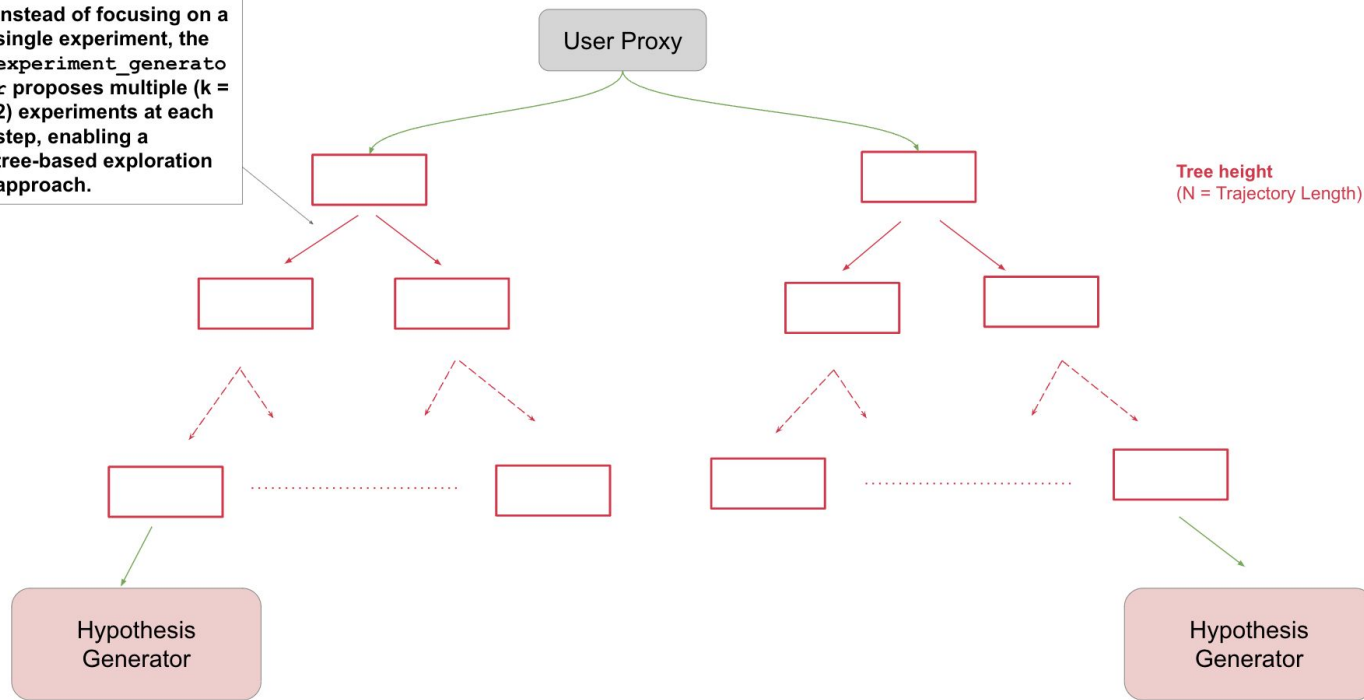
3 main challenges:

1. Repeated sampling is insufficient. **Need a method to sample diverse trajectories.**
2. Is diversity/novelty a sufficient metric? **What reward function should be used to guide exploration?**
 - Interestingness? Utility? **Hard to define!**
3. Even with useful reward functions, **how should models be steered to continually generate interesting hypotheses**, especially given the dynamical nature of “interestingness”?
 - **Is prompting enough? Can we do more?**

AutoDataVoyager - Exploration Trajectories

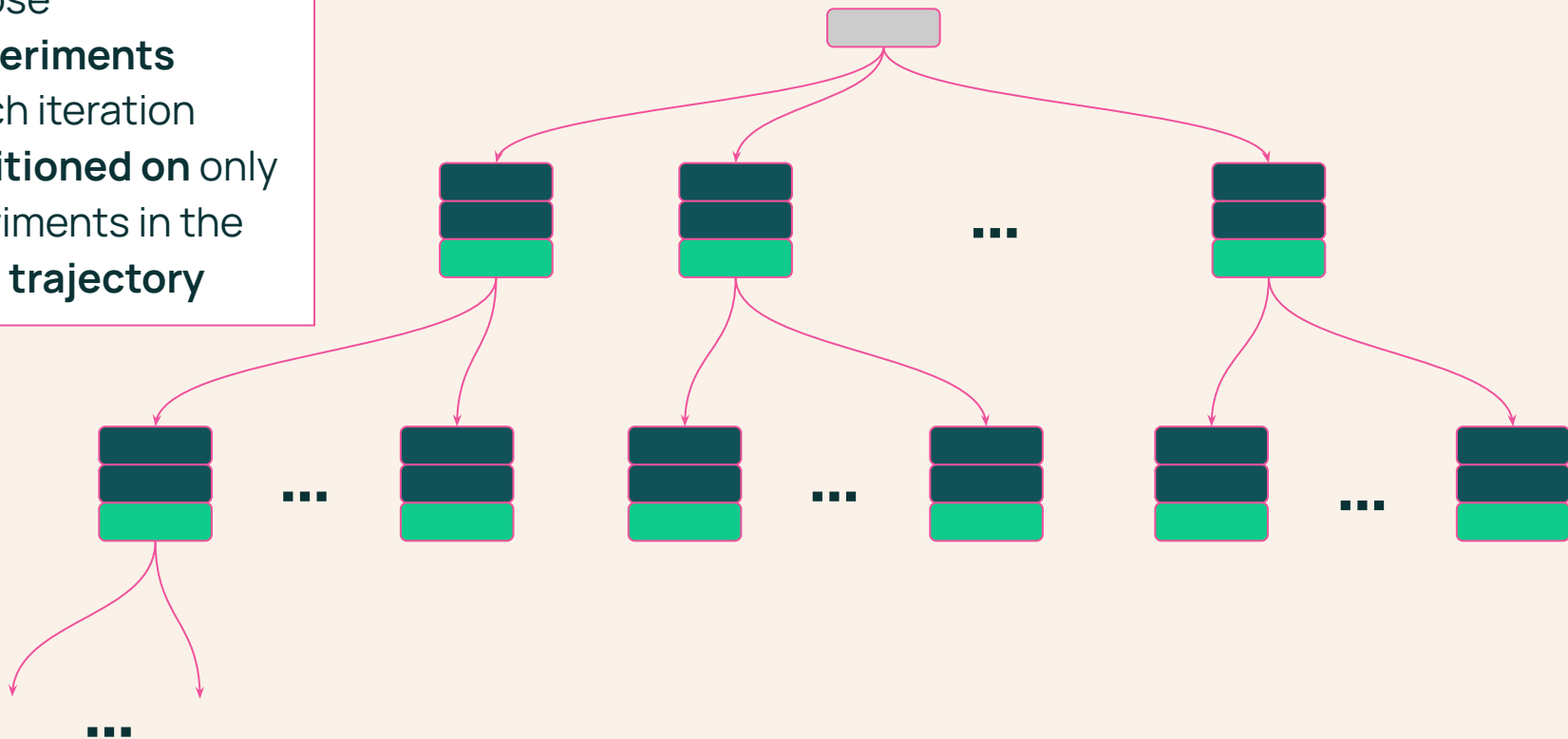
Exploration trees

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



AutoDV: Experiment Tree

Propose
 k experiments
in each iteration
conditioned on only
experiments in the
same trajectory



Experiment Tree: Samples

E: Analyze the **influence** of **family size** on **educational achievement**, specifically focusing on the **completed BA degree status**.

R: Significant difference in family size between those who completed a BA degree and those who did not, with smaller family sizes associated with degree completion.

H: Individuals from smaller families are more likely to complete a BA degree compared to those from larger families, as evidenced by the significant difference in mean family sizes between the two groups.

E: Investigate potential **nonlinear relationships** between **socioeconomic status (SES)** and academic performance indicators such as **ASVAB scores** and **class percentile**.

R: Slight nonlinear relationship with SES, but linear relationship remains dominant pattern.

H: There is a slight nonlinear relationship between socioeconomic status (SES) and academic performance indicators, with a quadratic term indicating diminishing returns at higher SES levels. However, the linear relationship remains the dominant pattern, suggesting that SES consistently predicts academic performance across its range.

E: Conduct a **clustering analysis** to identify distinct profiles of students based on their **ASVAB scores**, **class percentile**, **SES**, **race**, and **gender**.

R: Three distinct clusters identified, showing varying academic performance and SES levels.

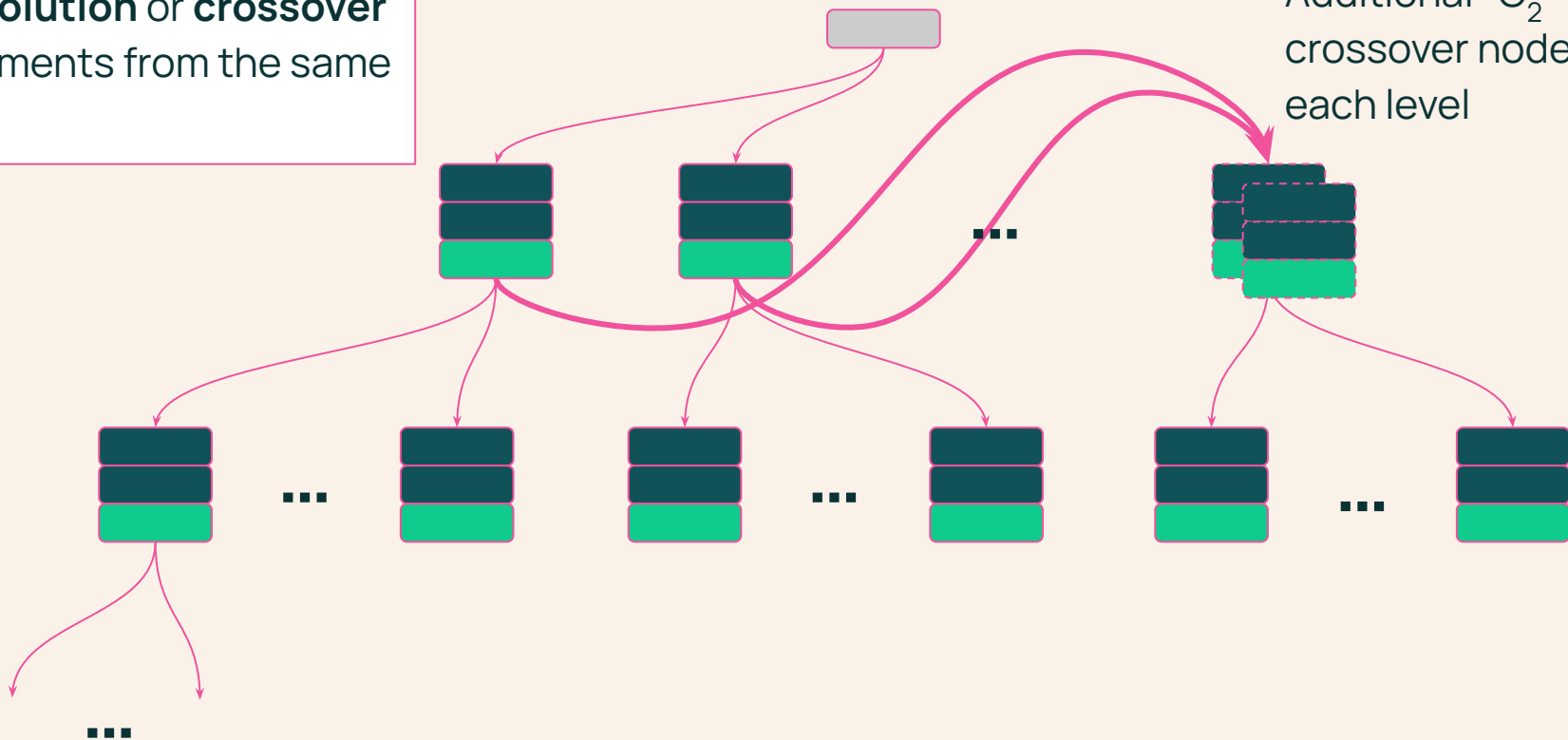
H: Distinct profiles of students can be identified based on ASVAB scores, class percentile, and SES, with higher SES associated with better academic performance. These profiles may also reflect racial and gender disparities, with certain groups more likely to belong to clusters with lower academic performance.

Higher diversity and higher complexity in experiment design.

AutoDV: Crossovers

Co-evolution or crossover experiments from the same level

Additional nC_2 crossover nodes in each level



Walkthrough of the DataVoyager system

https://x.com/surana_h/status/1786097912147239157

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

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.



Single Shot

Agentic Workflow

- 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

Text



What are the solutions?

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

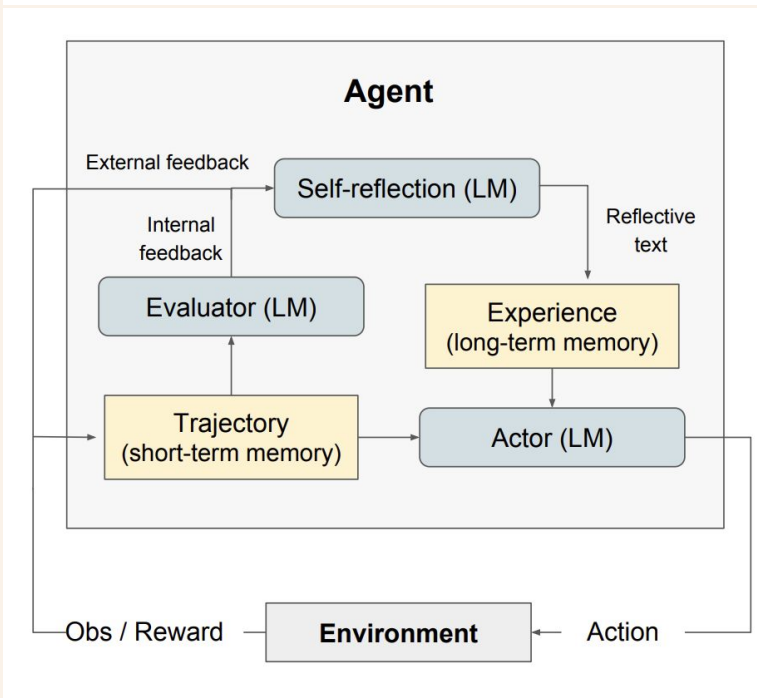
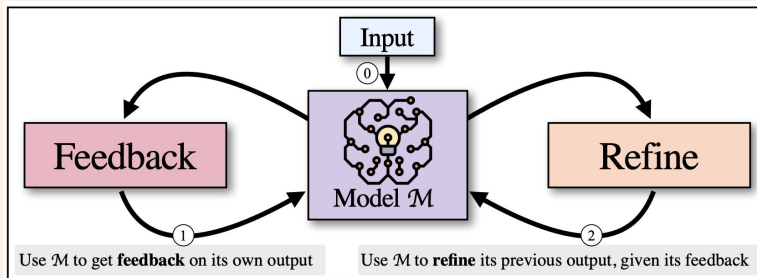
1. Reflection
2. Tool use
3. Planning
4. Multi-agent collab!

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 Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, Peter Clark. NIPS 24.

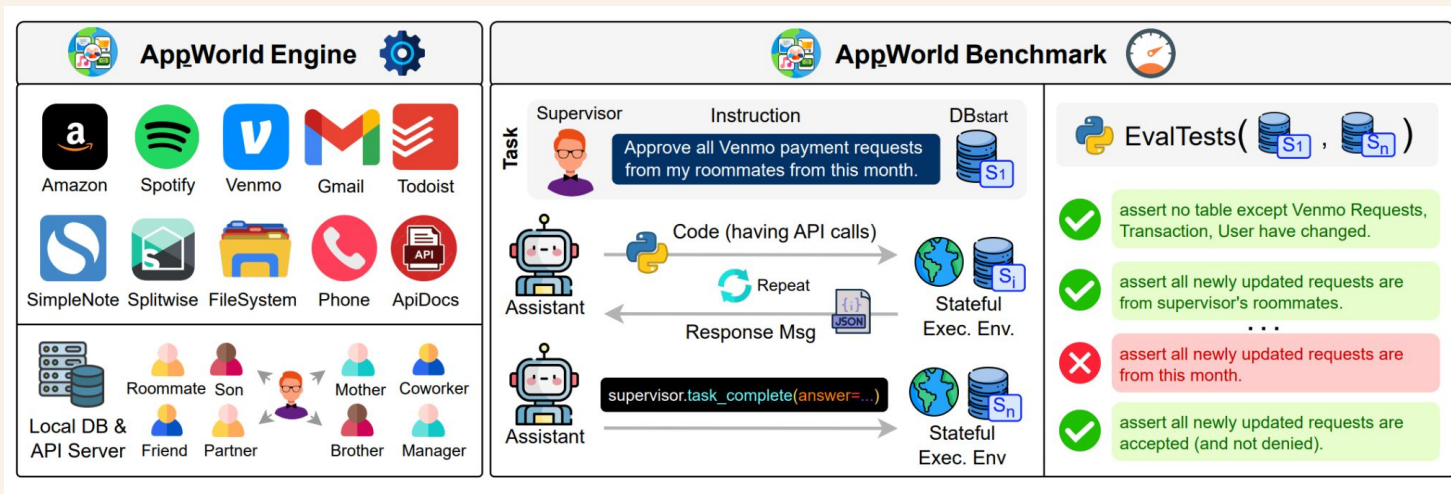


Tool Use

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

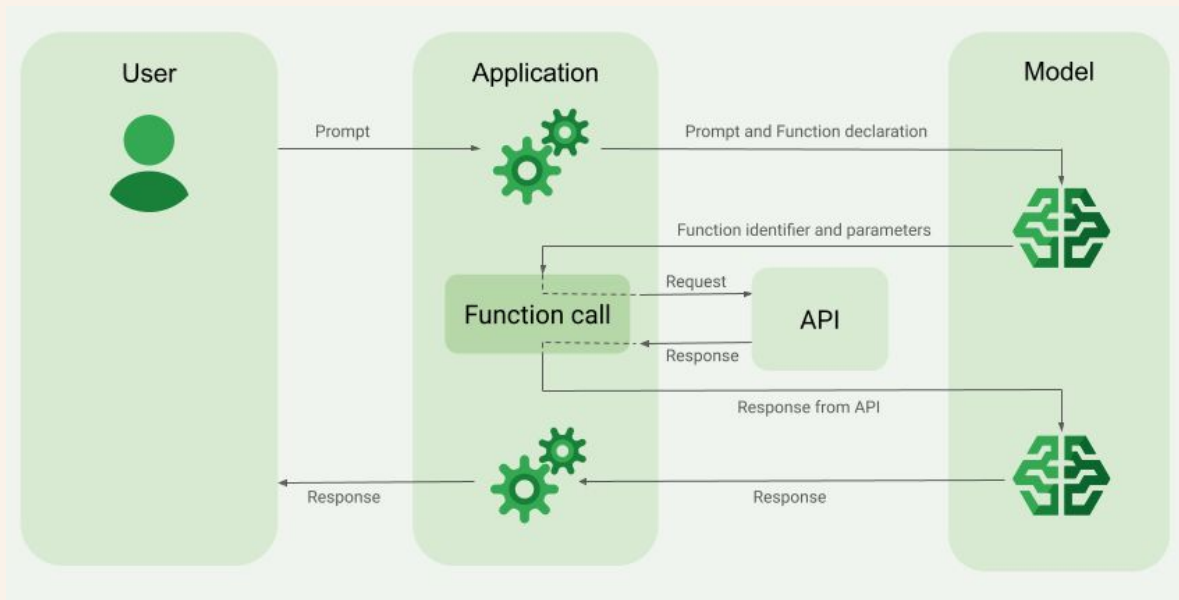


Tool Use

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**

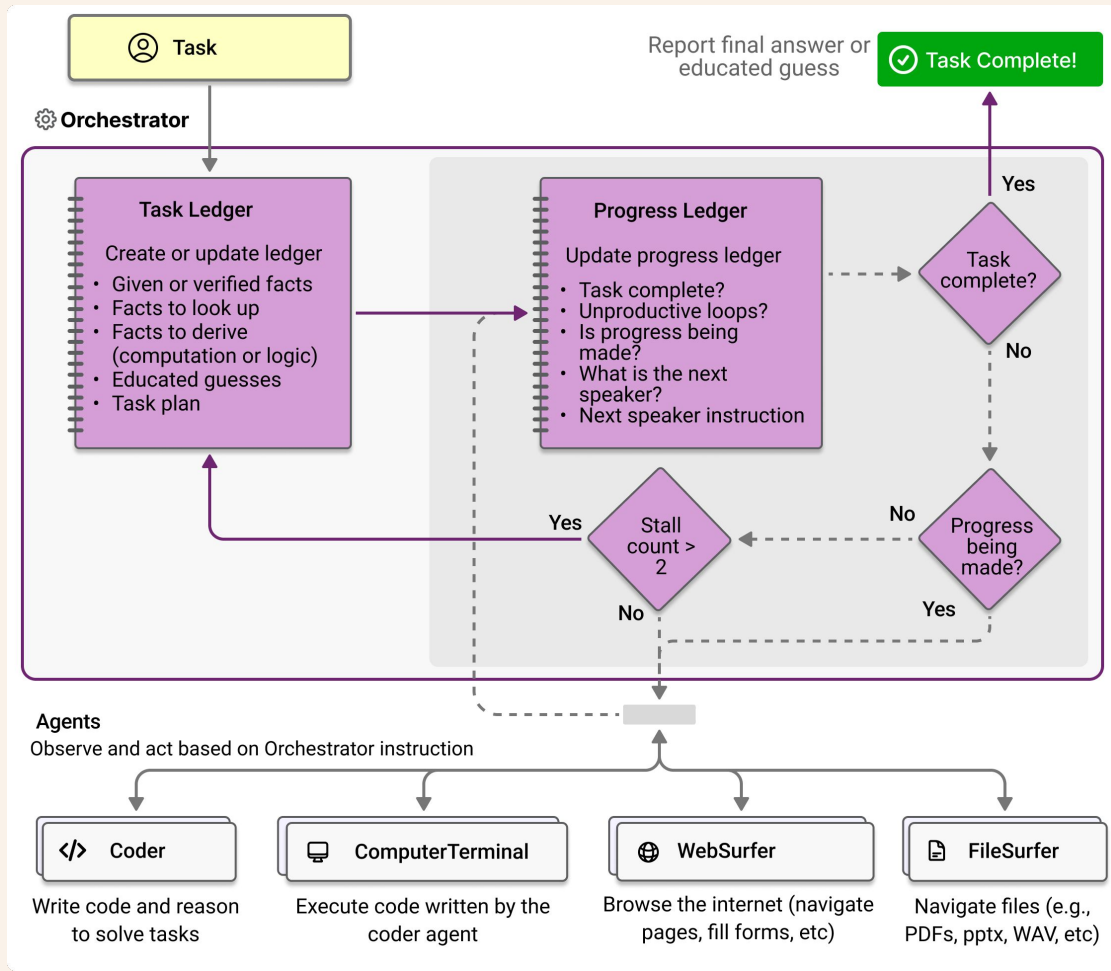


Planning

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

- Autogen Magento
- MetaGPT
- OpenHands
- HuggingGPT

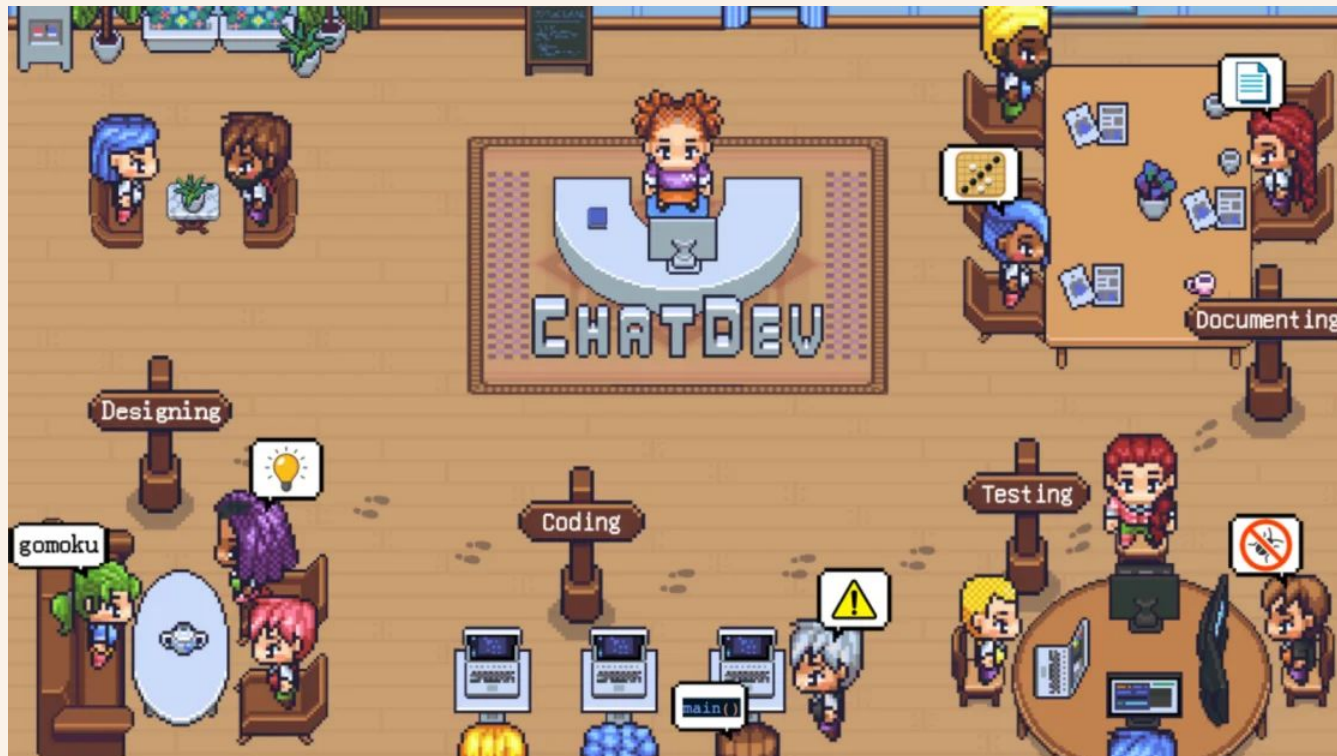
Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks. Fourney et. al Microsoft Tech Report 2024.



Multi-agent Collab

Agents work together to solve a complex task.

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



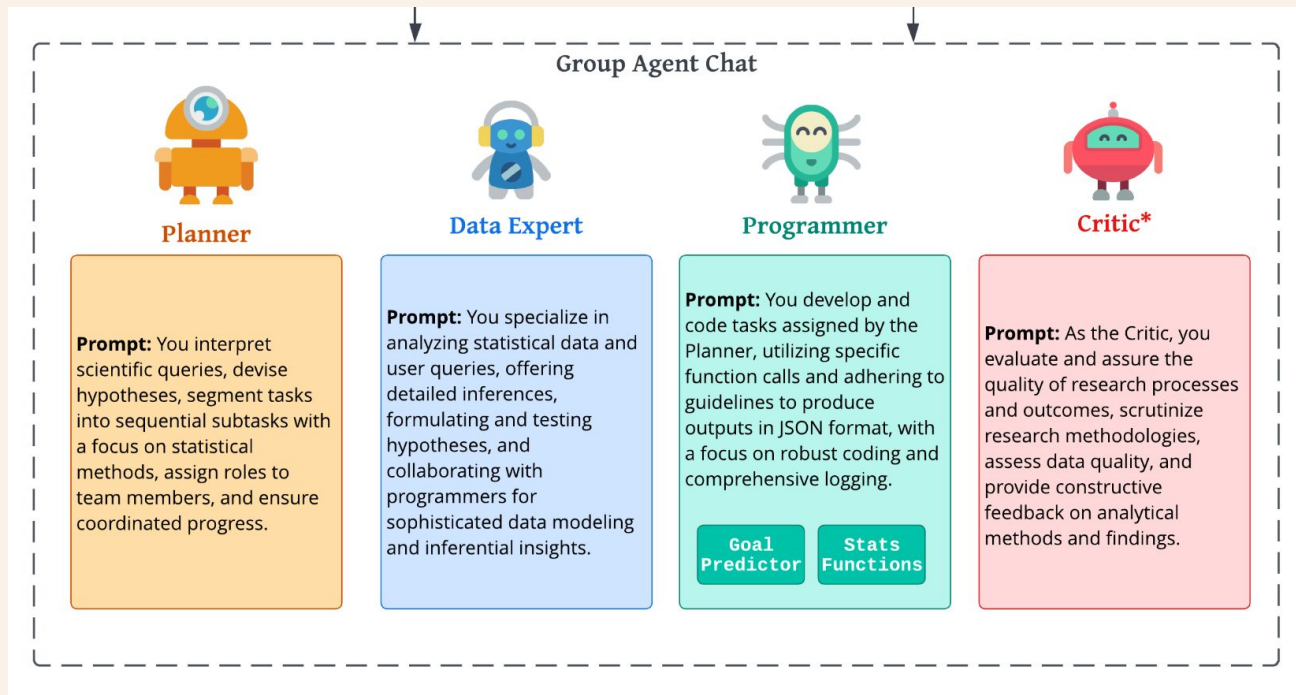
ChatDev: Communicative Agents for Software Development. Qian et. al.
ACL 2024



Multi-agent Collab

Agents work together to solve a complex task.

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



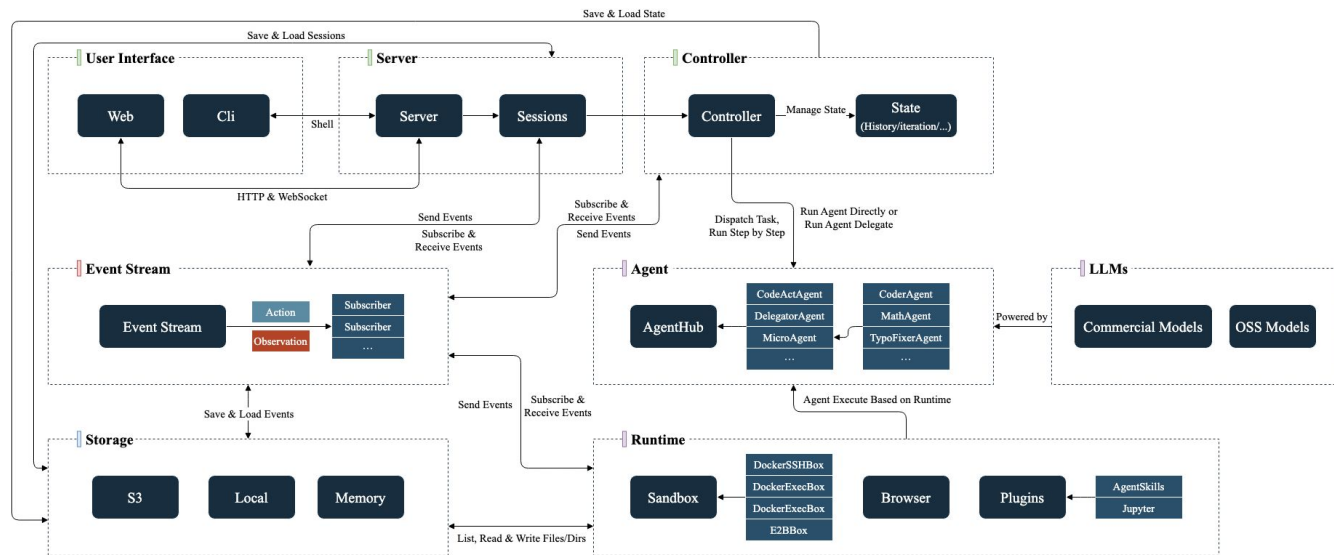
Data-driven Discovery with Large Generative Models. Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Sanchaita Hazra, Ashish Sabharwal, Peter Clark. ICML 2024.

Let's go over 2
concrete
examples!

Software Engg with LLMs

(not just code generation)

OpenHands by CMU & UIUC



Full coding env; **not just code gen**

Access to agents

Access to runtime

Access to tools

Pluggable LLMs

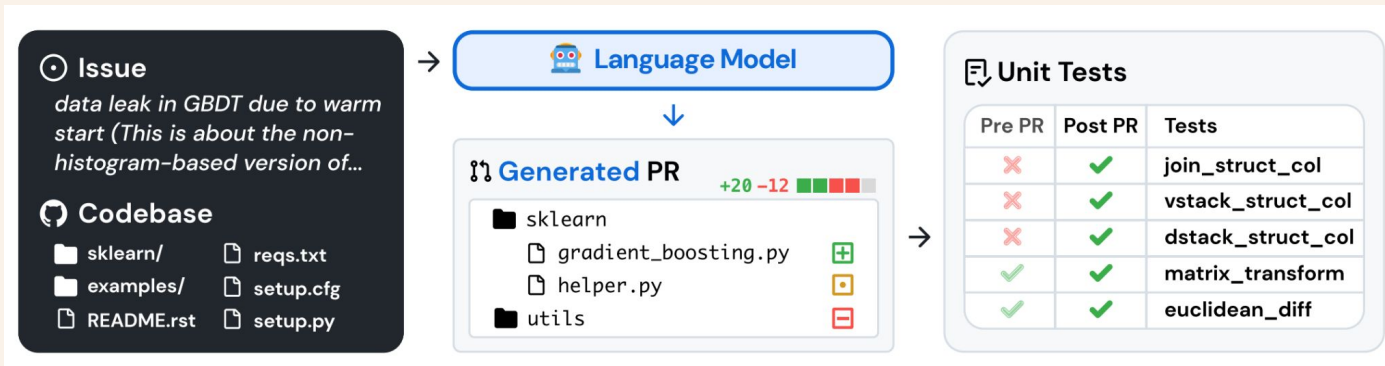
Top LLM repo on Github



Starred 37.9k



SWE Bench by Princeton



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.

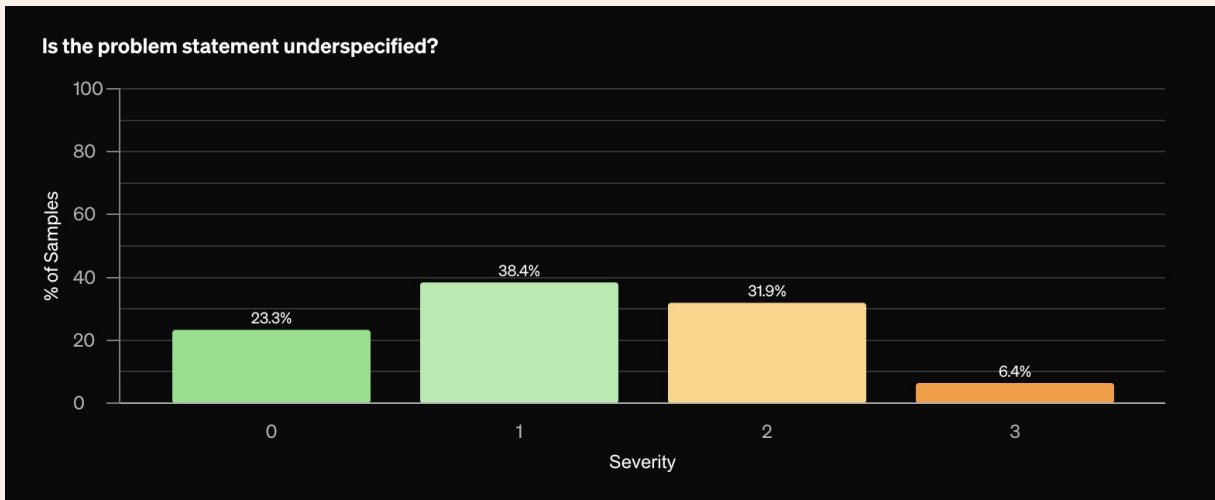
1 Scrape PRs
12 popular repositories
>90% Python Code

2 Attribute Filter
✓ Resolves an issue
✓ Contributes tests

3 Execution Filter
✓ Installs successfully
✓ PR passes all tests

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 Aljubeih, 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

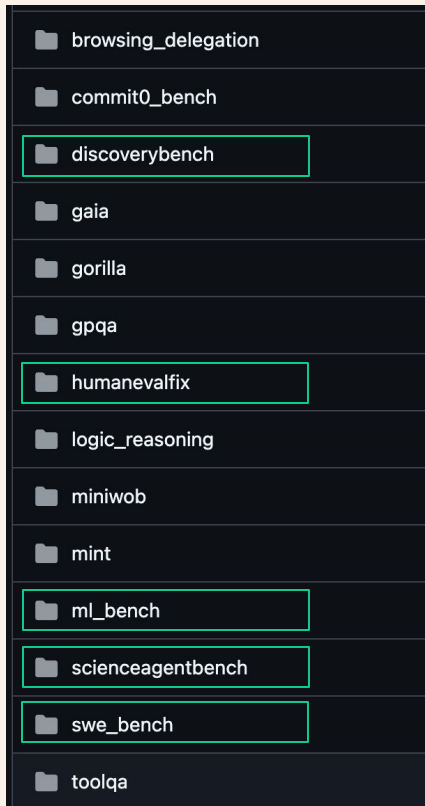
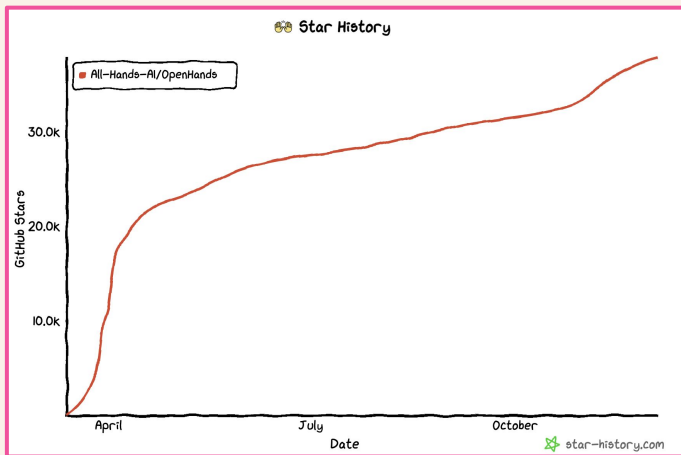
Agent + Model	Score
OpenHands + CodeAct v2.1 + Claude 3.5 Sonnet	53.00
Anthropic Tools + Claude 3.5 Sonnet	49.00
Anthropic Tools + Claude 3.5 Haiku	40.60
Composio SWEkit + Claude 3.5 Sonnet	40.60
SWE-agent + Claude 3.5 Sonnet	33.60
SWE-agent + Claude 3 Opus	18.20
RAG + Claude 3 Opus	7.00
RAG + Claude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

Contributing to OSS LLM Dev with OpenHands

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

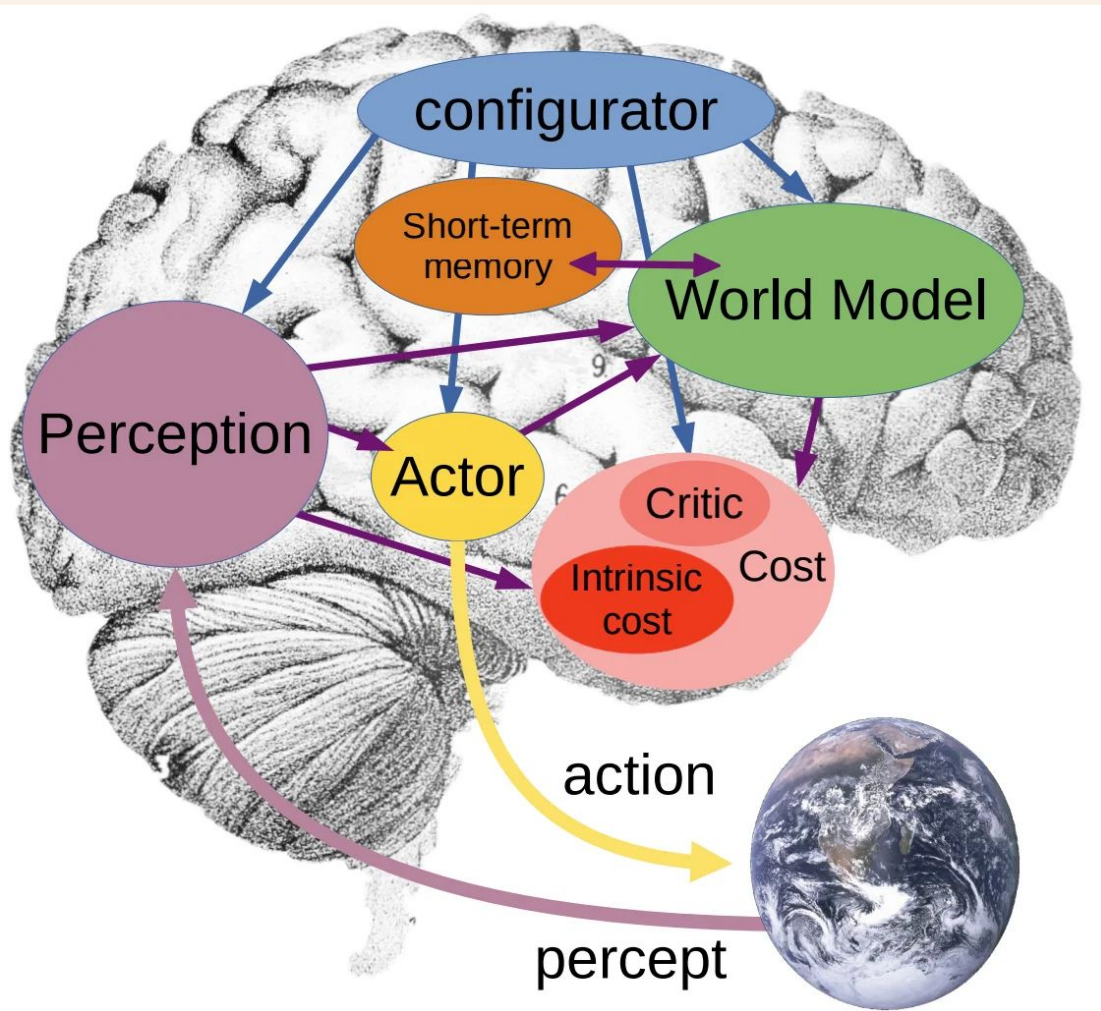
Do try contributing to OpenHands. Their **aim is to have full replication of production-grade applications with LLMs**

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



V Jepa

By Meta/ Yann Le Cunn



Thank you!

