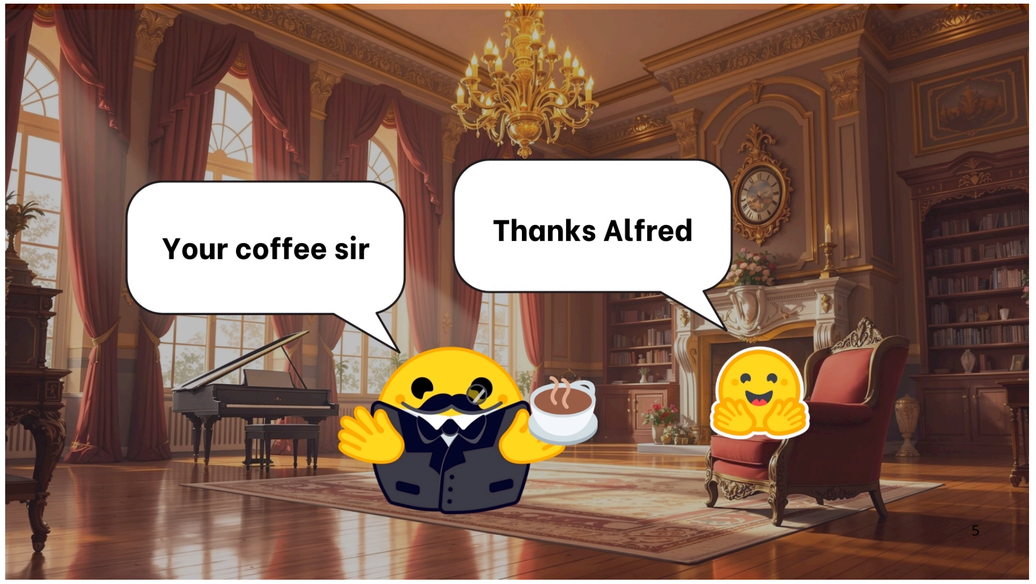
LLM Agents & Neuro-symbolic models

LMARL Session 4, February 20th 2025, Polina Tsvilodub

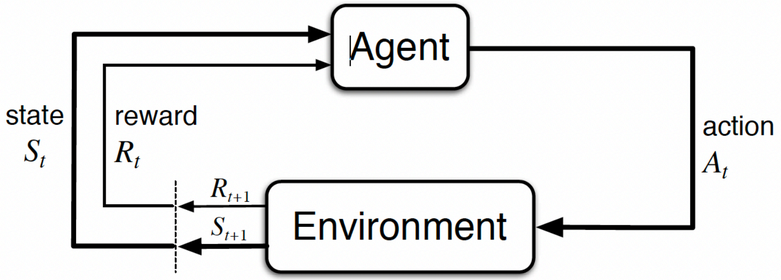

Alfred, I would like a coffee please
2

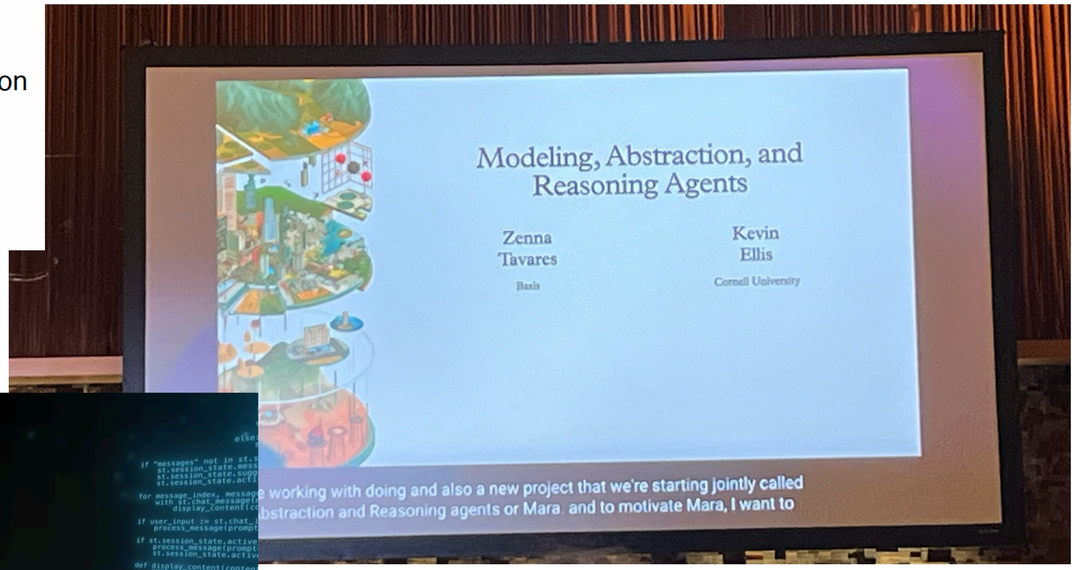

Alfred's Toolbox
Reason and plan
3

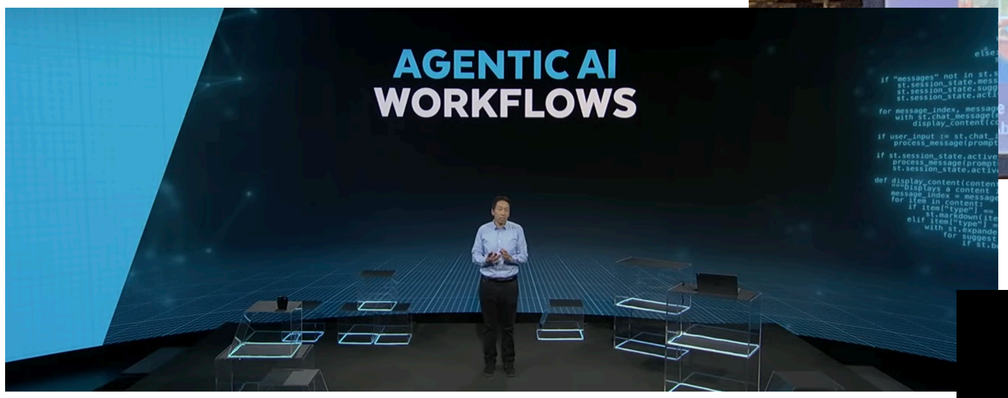

Execute
4

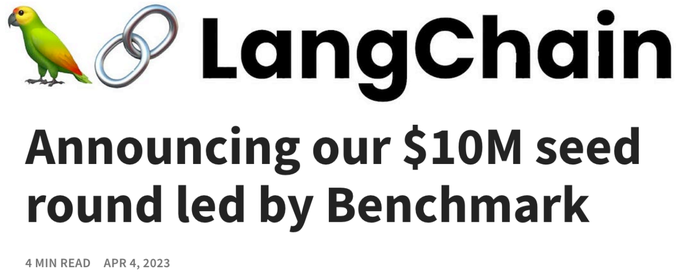

Your coffee sir
Thanks Alfred
5

Course on agents: Why now?









Cognitive architecture

Overview of general components:

- **Perception**: transform raw sensory input into representations
- **Attention Mechanisms**: allocate cognitive resources to certain information
- **Action Selection**: decision-making processes as to which actions to undertake
- **Memory Systems**:
  - Short-term memory
  - Working Memory: "temporary storage" for active tasks
  - Long-Term Memory: “permanent” storage for knowledge, rules, and experiences
    - episodic memory
    - procedural memory
    - semantic memory
- **Learning**:
  - adaptation, generalization based on experiences
- **Reasoning & Metacognition**:
  - Logical inference, probabilistic models, DPT, self-reflective thinking

Laird et al. (2017)

Human feedback in RL

RLHF

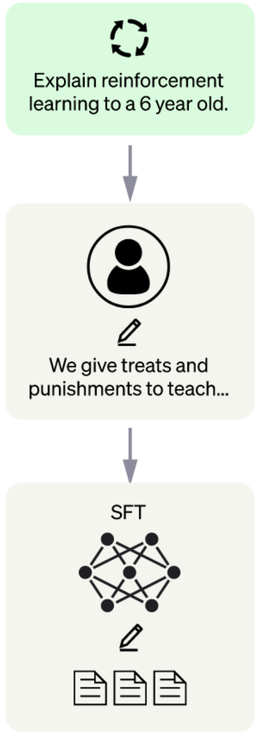
Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.



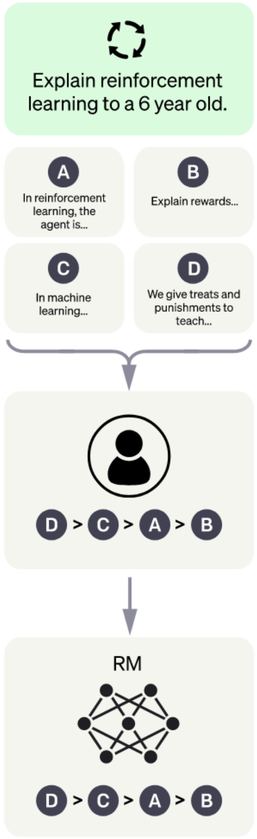
Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

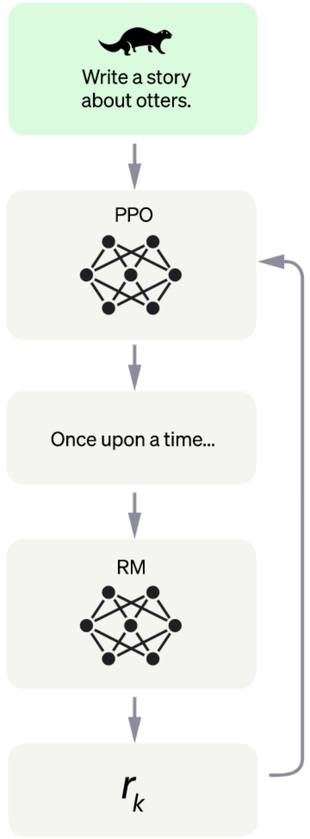
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

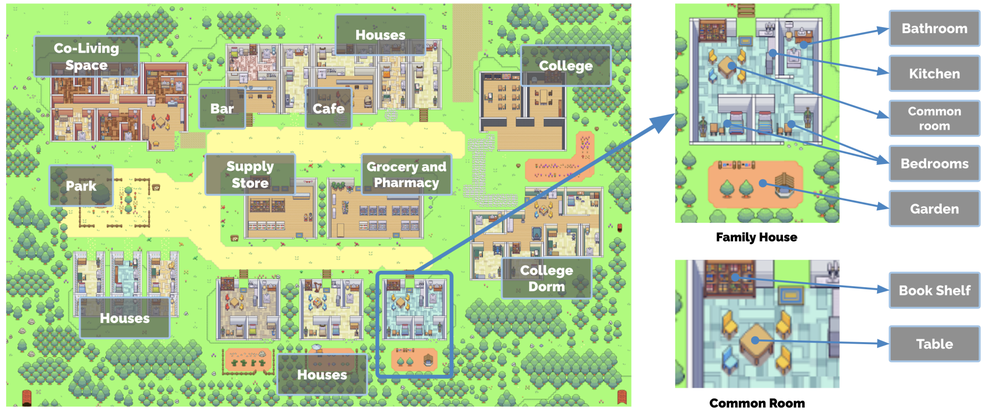
The reward is used to update the policy using PPO.

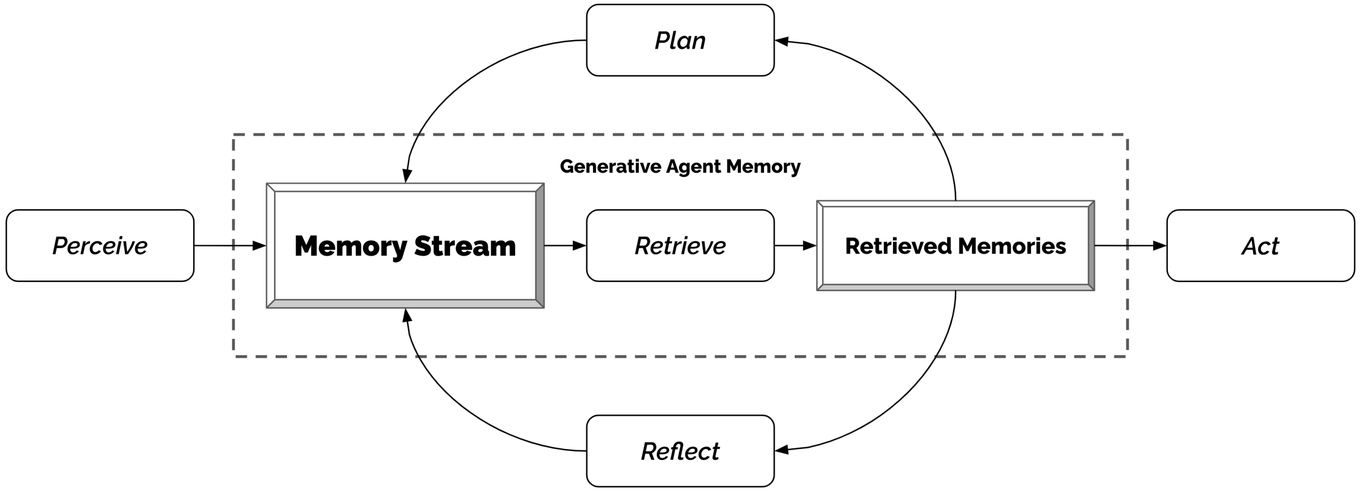


OpenAI (2022)

Generative agents

- The Sims-style environment Smallville in which LLM based agents dynamically simulate human behavior
- based on 25 agents (initialized with text bio)
  - interaction with environment via descriptions of actions
  - (emergent) social behavior between agents
  - user intervention via conversation or direct instruction
  - game sandbox movements computed based on LLM output

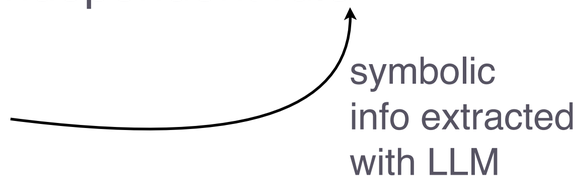




Park et al. (2023)

Synergistic Integration of Large Language Models and Cognitive Architectures for Robust AI

- Modular approach:
  - LLMs partially enhance the performance of certain modules and components of a CA
  - a CA augments an LLM by injecting reasoning traces and contents from memories into the prompting process.
- Agency approach:
  - specialized agents process information in parallel, competing for resources like attention and memory
  - coordinated in a global workspace
  - single agents could be LLMs or symbolic
- Neuro-symbolic approach:
  - CLARION: action-centered sub-system operating on:
    - top level (symbolic), responsible for encoding explicit knowledge -> fixed rules, independent rule-learning, rule-extraction-refinement
    - bottom level (connectionist), tasked with encoding implicit knowledge -> LLMs



  - synergistically engage in action selection, reasoning, and learning processes

Romero et al. (2023)

Eureka: Human-Level Reward Design via Coding Large Language Models

Evolution-driven Universal REward Kit for Agent

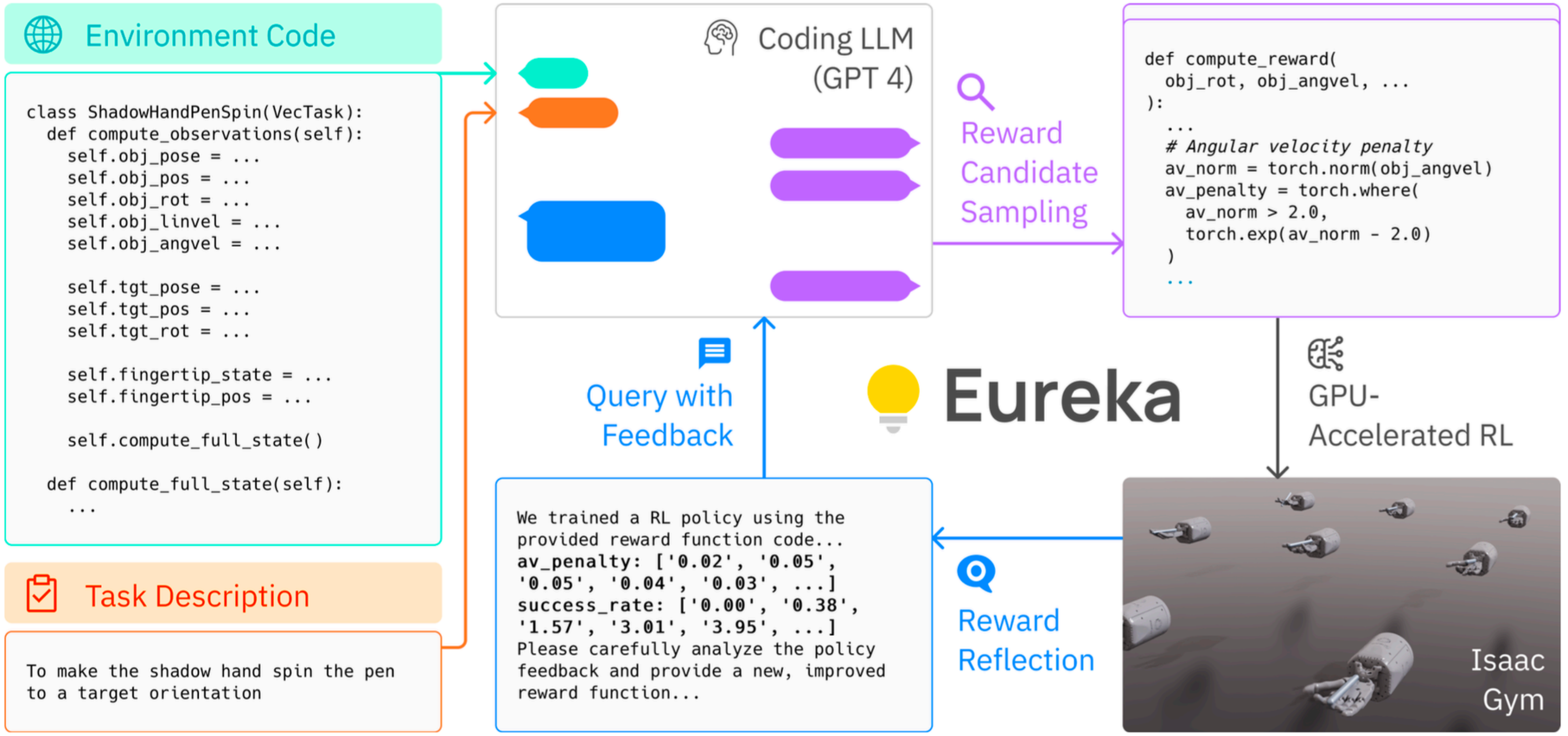
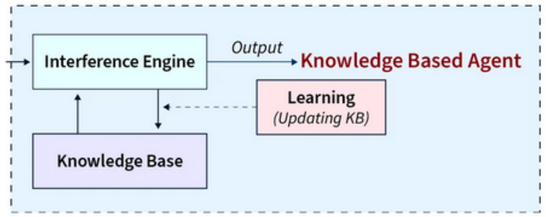


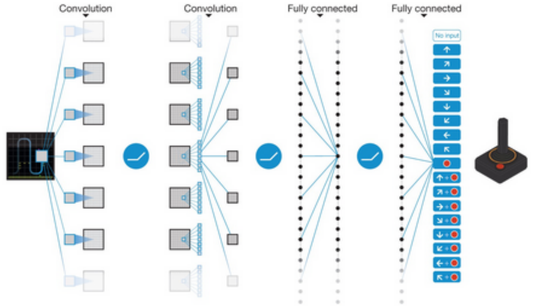
Figure 2: EUREKA takes unmodified environment source code and language task description as context to zero-shot generate executable reward functions from a coding LLM. Then, it iterates between reward sampling, GPU-accelerated reward evaluation, and reward reflection to progressively improve its reward outputs.

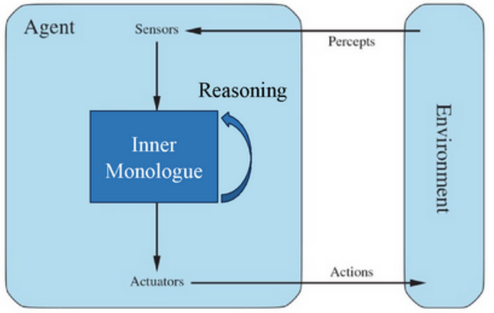
Ma et al. (2024, ICLR)

Evolution of AI Agents

CSP-Subheading







Logical Agent | **Neural Agent** | **Language Agent**

Expressiveness	Low bounded by the logical language	Medium anything a (small) NN can encode	High almost anything, esp. verbalizable parts of the world
Reasoning	Logical inferences sound, explicit, rigid	Parametric inferences stochastic, implicit, rigid	Language-based inferences fuzzy, semi-explicit, flexible
Adaptivity	Low bounded by knowledge curation	Medium data-driven but sample inefficient	High strong prior from LLMs + language use

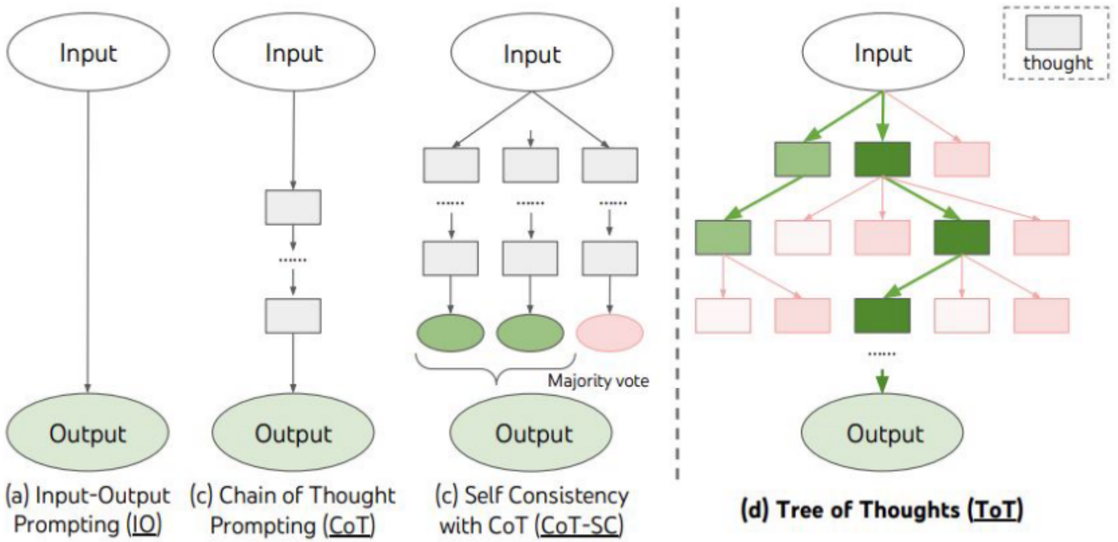
Image sources: https://www.scaler.com/topics/artificial-intelligence-tutorial/knowledge-based-agent/,
Mnih et al., “Human-level control through deep reinforcement learning.” Nature (2015)

Citation (2002), Citation 2 (2050)

LLMs & complex tasks

Chain-of-thought prompting & generation

CSP-Subheading



Tree of Thoughts, 2023

Math Word Problems (free response)

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Math Word Problems (multiple choice)

Q: How many keystrokes are needed to type the numbers from 1 to 500? Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788

A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. 9 + 90(2) + 401(3) = 1392. The answer is (b).

StrategyQA

Q: Yes or no: Would a pear sink in water?

A: The density of a pear is about 0.6 g/cm^3, which is less than water. Thus, a pear would float. So the answer is no.

Date Understanding

Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?

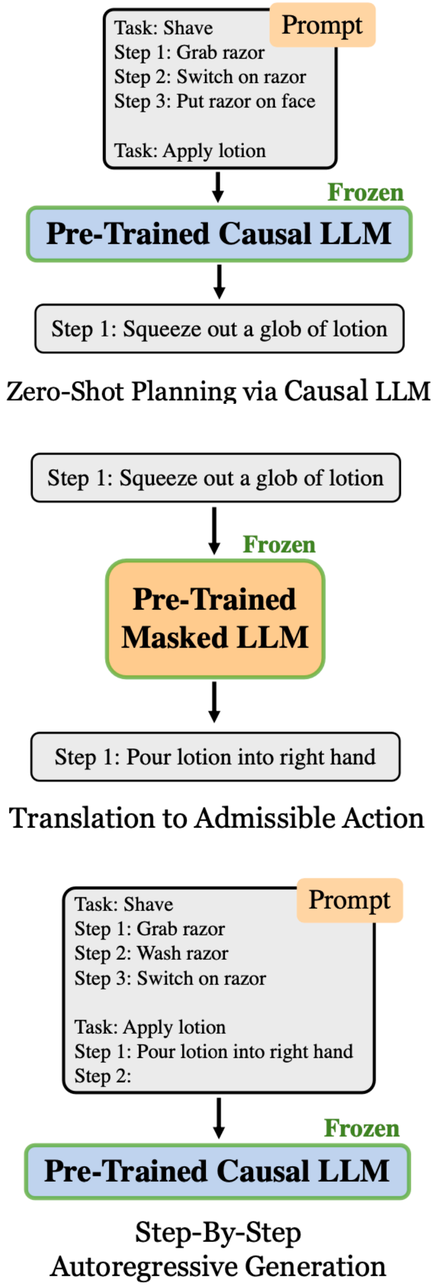
A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.

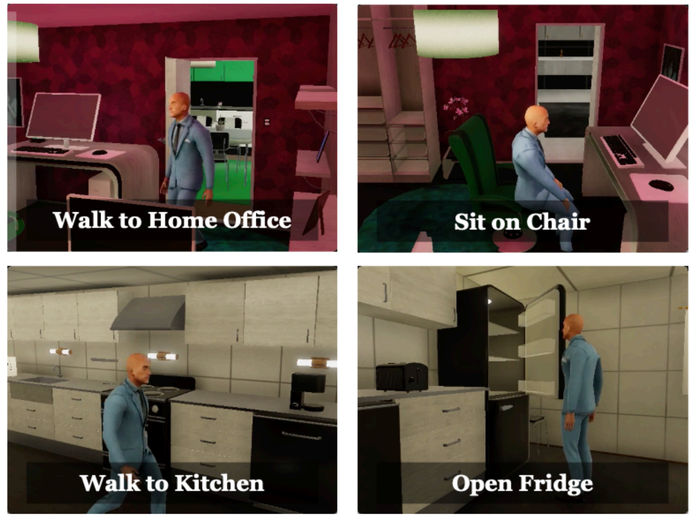
*(Wei et al., 2022), also see (Ling et al., 2017; Cobbe et al., 2021; Nye et al., 2021)

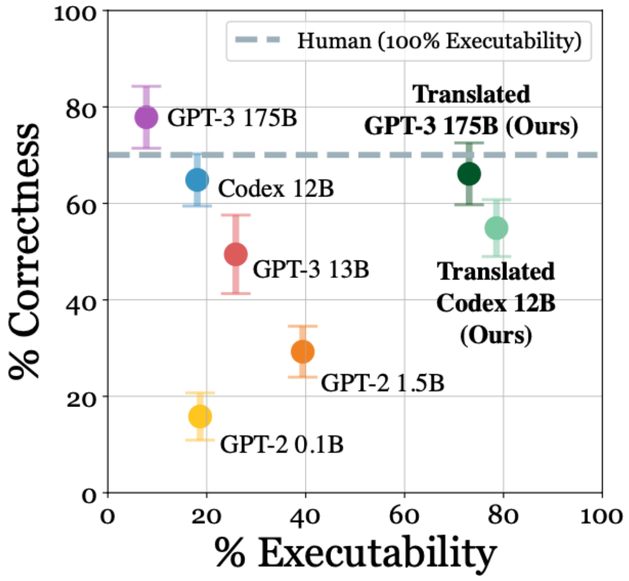
Citation (2002), Citation 2 (2050)

LLMs as planners

- LLMs can produce structured action sequences for a given goal
  - e.g., given a few examples / in-context learning
- problem:
  - map them onto a set of primitive actions that we can execute
- solution:
  - generate next action in sequence
  - use compare similarity of generation to list of executable actions (embeddings, BERT)
  - insert text of known action in list
  - loop







Huang (Wenlong) et al. (2022) “LMs as zero-shot planners”

Structured reasoning

ChatGPT (Dec 2023)

You

Do the numbers of letters in all words starting with a vowel from the following list sum up to 42?

Polina, Michael, eggplant, cheese, oyster, imagination, elucidation, induce

Please answer just 'yes' or 'no'

ChatGPT

No.

You

Are you sure?

Analysis

python — Copy code

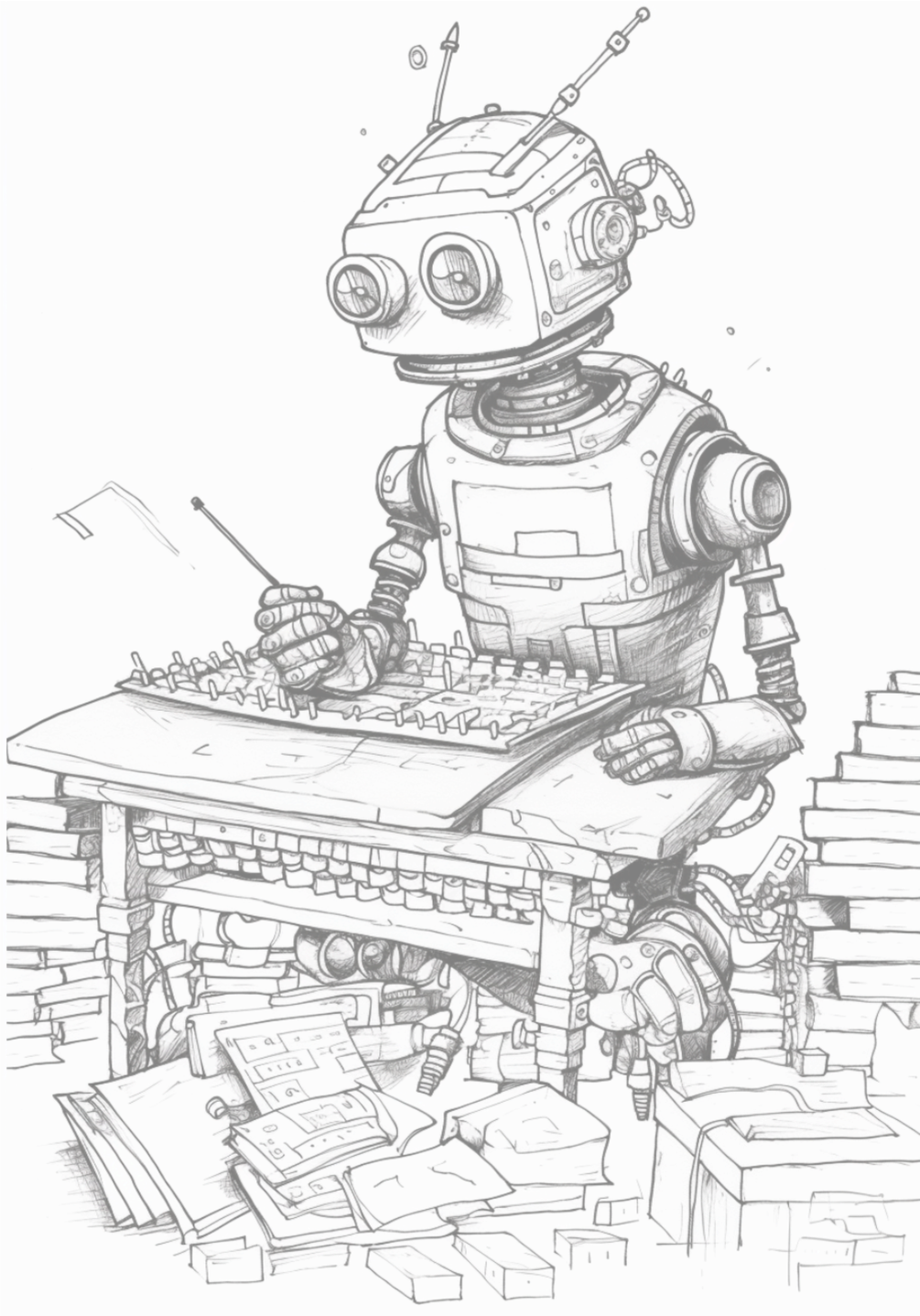
```
# List of words starting with a vowel
words = ["eggplant", "oyster", "imagination", "elucidation"

# Calculating the sum of the number of letters in each word
total_letters = sum(len(word) for word in words)
total_letters == 42
```

Result

```
True
```

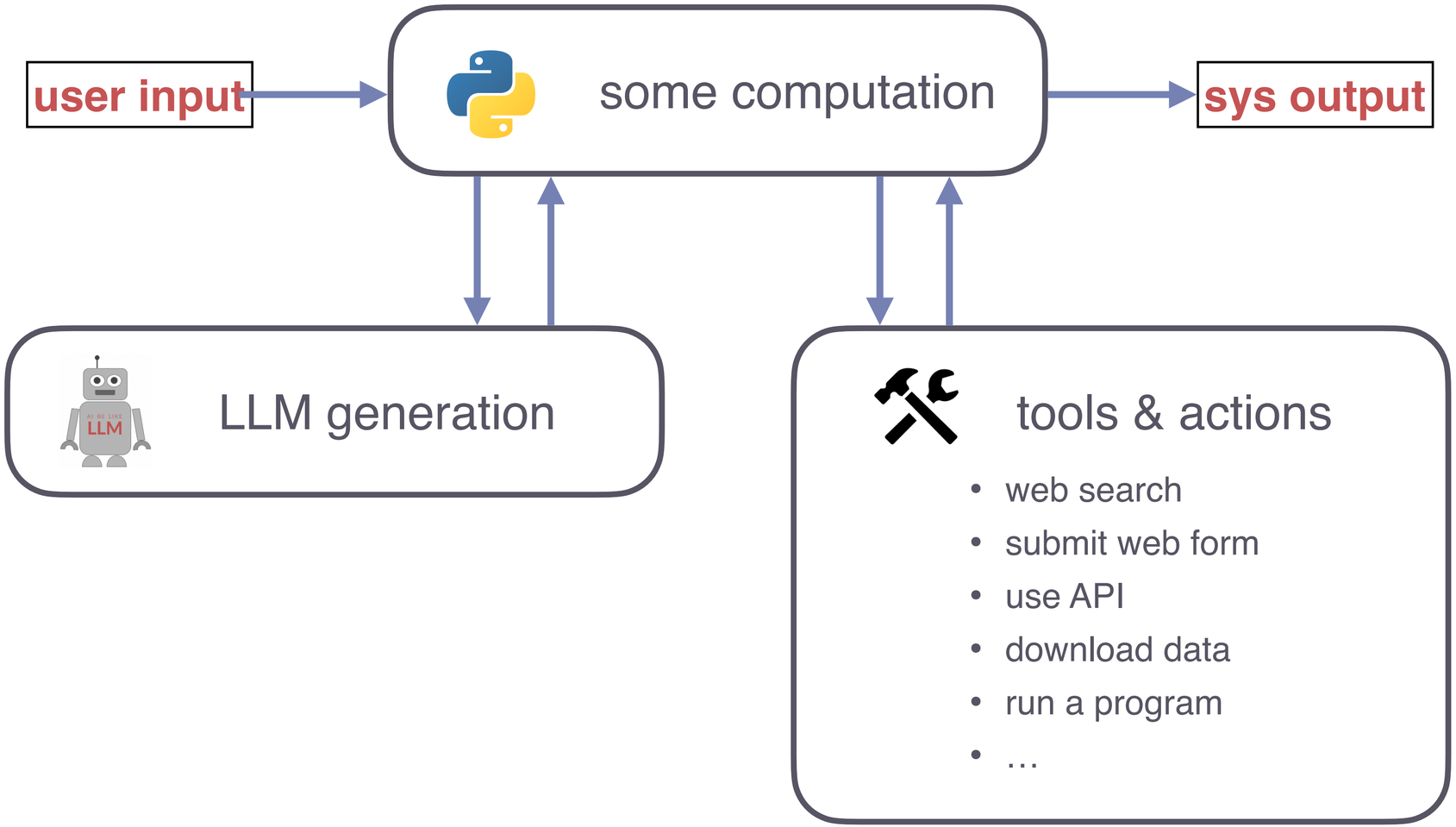
ChatGPT

Yes, the numbers of letters in all words starting with a vowel from the list do sum up to 42.



(Commercial) LLM agents

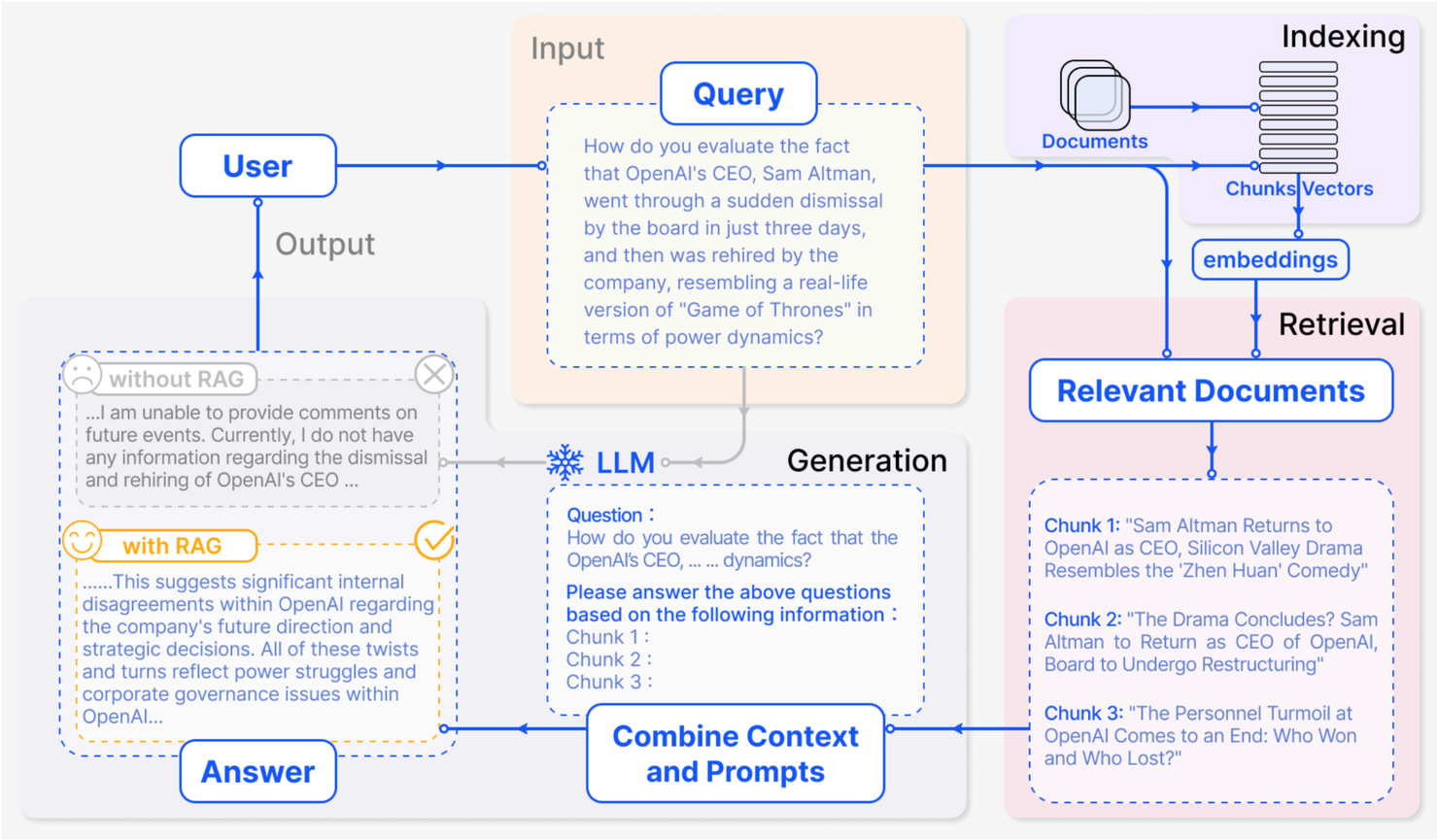
LLM-based applications



Retrieval-augmented generation

naive architecture

- indexing
  - pre-process relevant information into chunks of plain text
  - compute text embeddings for each chunk (e.g., BERT)
- retrieval
  - compute embedding for user prompt
  - retrieve k most similar chunks
    - e.g., (cosine) similarity of embeddings
- generation
  - supply retrieved chunks in prompt



Gao (Yunfan) et al. (2024) ”RAG for LLMs: A Survey”

Components of LLM agents

overview

Profile


Profile contents

- Demographic information
- Personality information
- Social information

Generation strategy

- Handcrafting method
- LLM -Generation method
- Dataset Alignment method

Memory


Memory structure

- Unified memory
- Hybrid memory

Memory formats

- Languages
- Databases
- Embeddings
- Lists

Memory operation

- Memory reading
- Memory writing
- Memory reflection

Planning

Planning w/o feedback

- Single-path reasoning
- Multi-path reasoning
- External planner

Planning w/ feedback

- Environment feedback
- Human feedback
- Model feedback

Action

Action target

- Task Completion
- Exploration
- Communication

Action production

- Memory recollection
- Plan Following

Action space

- Tools
- Self-Knowledge

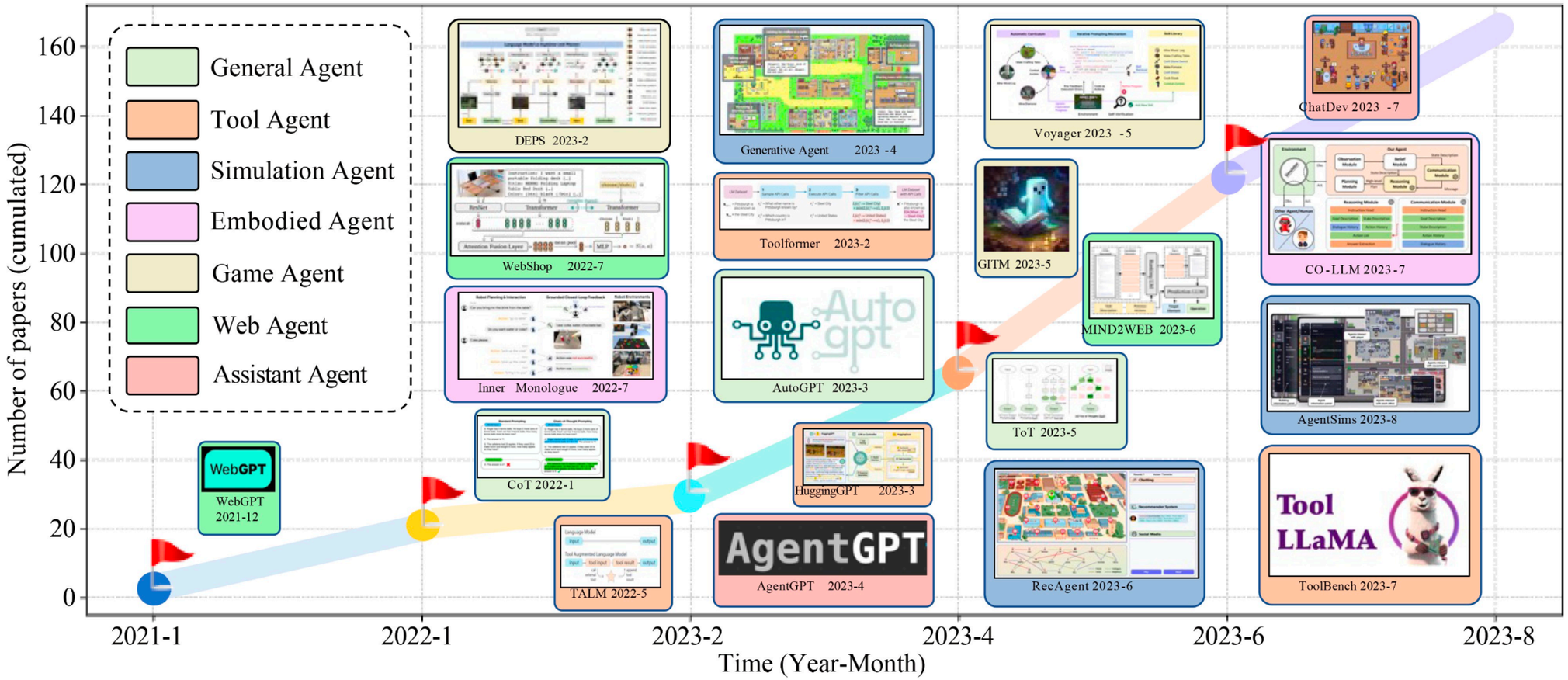
Action impact

- Environments
- New actions
- Internal States

Wang (Lei) et al. (2023)

LLM Agents overview

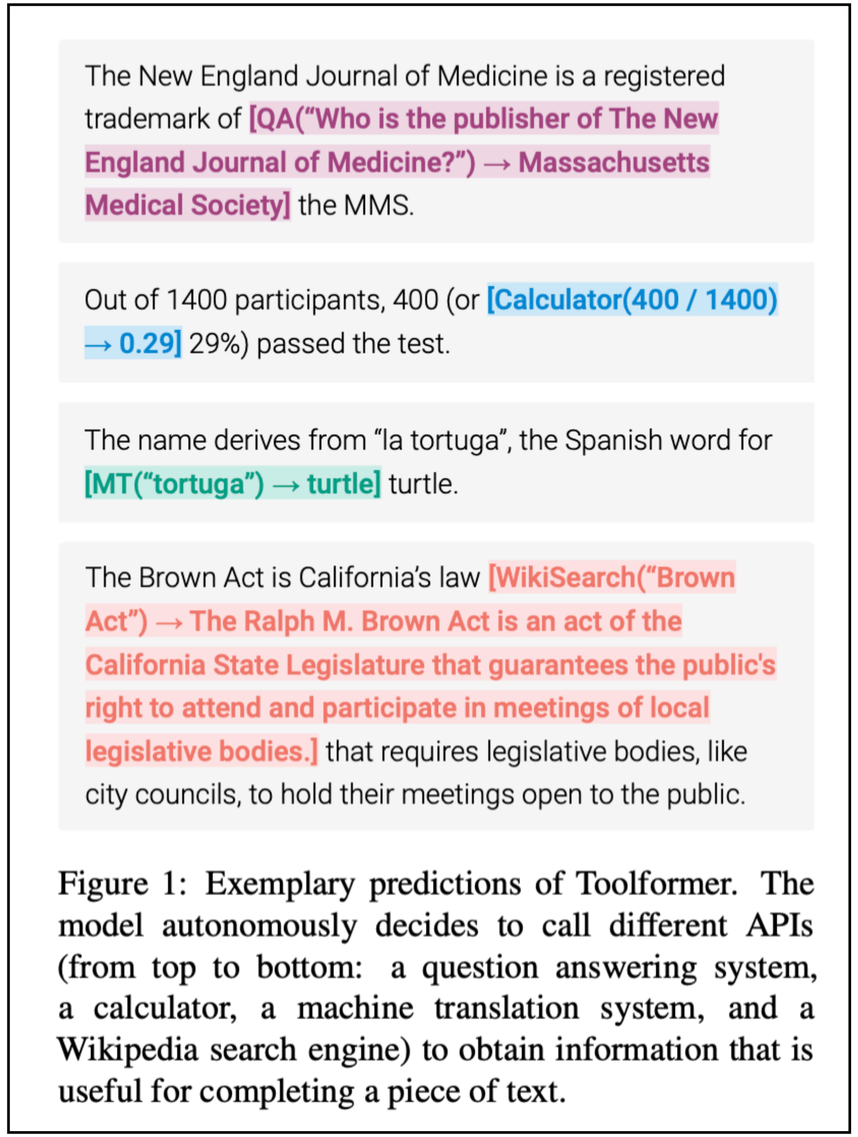
overview



Wang (Lei) et al. (2023)

LLMs can (learn to) **use tools**

- common problems of foundation / prepped LLMs
  - arithmetic
  - hallucination / factual lookup
- addressed by teaching LLMs to use external tools:
  - calculator
  - search engine
  - specialized machine translation system
  - calendar
- new model **Toolformer**:
  - decides which tool to use / API to call
  - how to call it (which arguments to pass)
  - how to condition future predictions based on outcome



The New England Journal of Medicine is a registered trademark of **[QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society]** the MMS.

Out of 1400 participants, 400 (or **[Calculator(400 / 1400) → 0.29]** 29%) passed the test.

The name derives from "la tortuga", the Spanish word for **[MT("tortuga") → turtle]** turtle.

The Brown Act is California's law **[WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.]** that requires legislative bodies, like city councils, to hold their meetings open to the public.

Figure 1: Exemplary predictions of Toolformer. The model autonomously decides to call different APIs (from top to bottom: a question answering system, a calculator, a machine translation system, and a Wikipedia search engine) to obtain information that is useful for completing a piece of text.

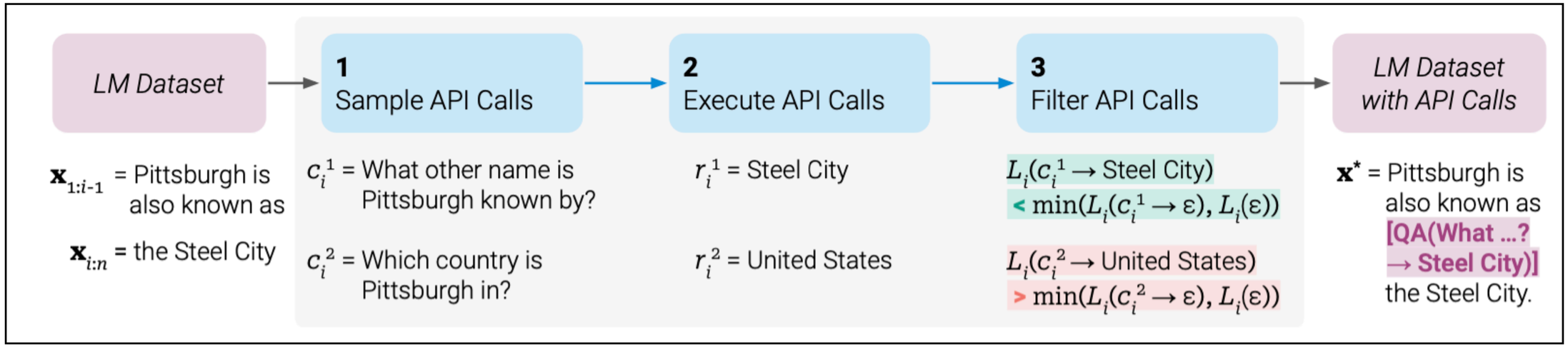
Schick et al. (2023) "Toolformer"

LLMs can (learn to) **use tools**

- general approach
  - fine-tune pre-trained LM on data set that has text and API calls with there results
    - API calls are at the “right” place
    - results are computed

The New England Journal of Medicine is a registered trademark of **[QA(“Who is the publisher of The New England Journal of Medicine?”) → Massachusetts Medical Society]** the MMS.

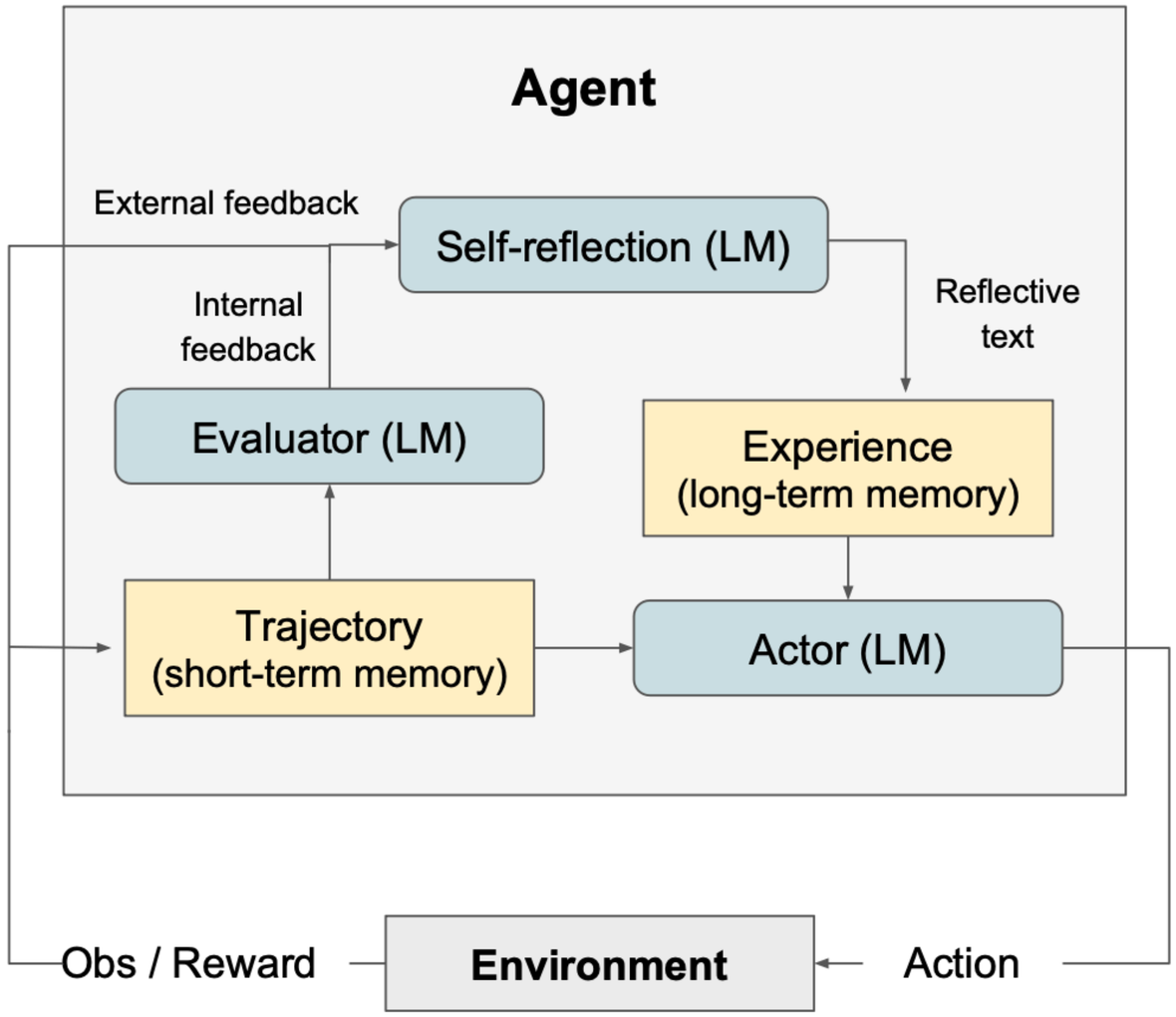
- build this data set by
  - using LLMs to sample API calls at random places
  - execute the calls and store the results
  - check whether the downstream predictions of the LLM get better (reduce generation uncertainty) if API call and result are inserted
  - if so, keep that datum and add it to the data set



Reflexion agents

learning from observations based on linguistic feedback

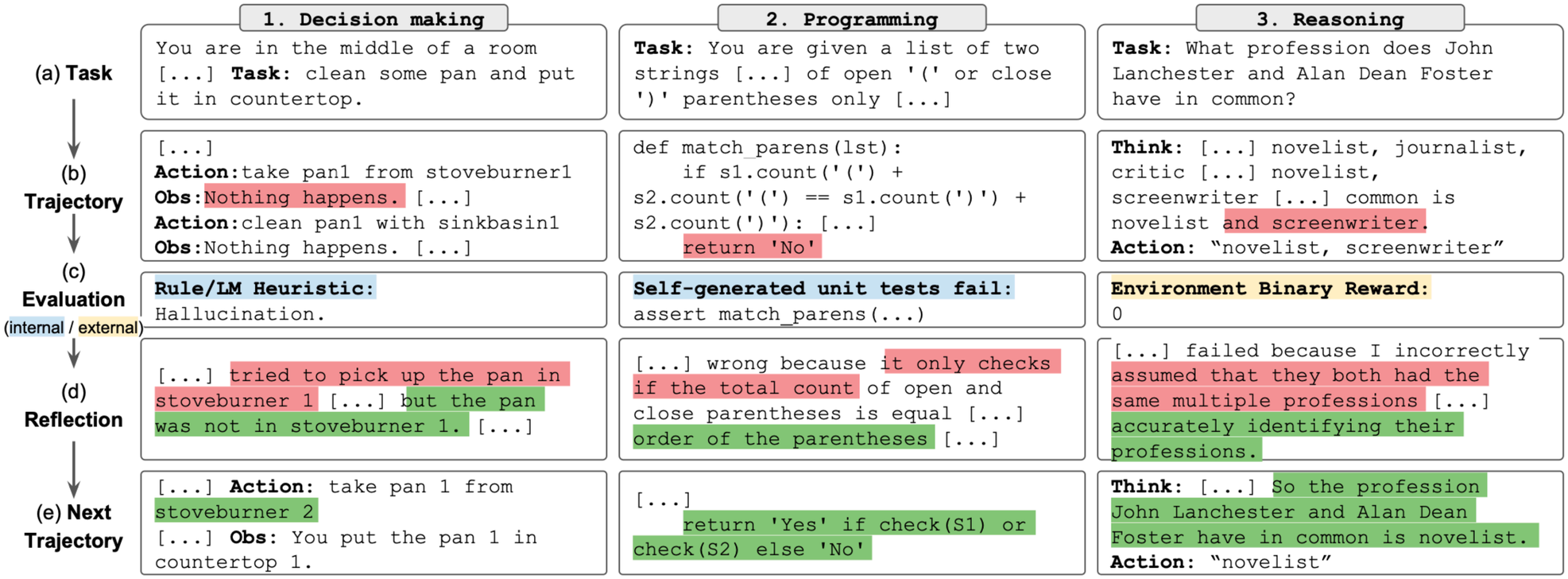
- actor
  - chooses actions, produces text/code
    - uses CoT or ReAct
- evaluator
  - scores the outcome
- self-reflection
  - generates verbal reinforcement cues
  - input:
    - sparse reward signal
    - current trajectory
    - memory
  - output:
    - nuanced and specific feedback
    - more informative than scalar rewards



Shinn et al. (2023)

Reflexion agents

learning from observations based on linguistic feedback



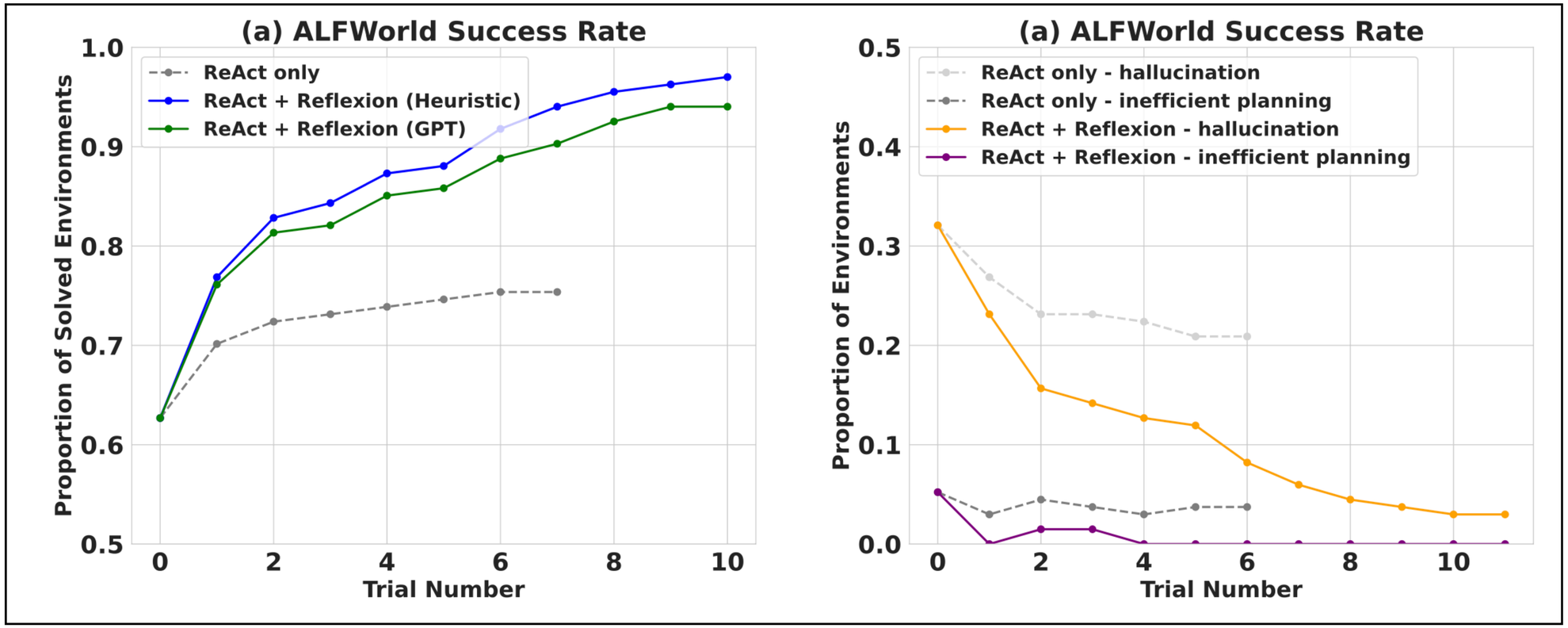
Shinn et al. (2023)

Reflexion agents

learning from observations based on linguistic feedback

increases task performance

reduces hallucinations and inefficient planning



Shinn et al. (2023)

LangChain

LangChain

Sophisticated prompting

- single call to LLM
- call predefined
- performance optimized via smart prompt engineering

LangChain chain

- multiple calls to LLM
- calls predefined
- performance optimized via repeated use of LLM to complete different tasks
  - I/O
  - calls based on own results & CoT

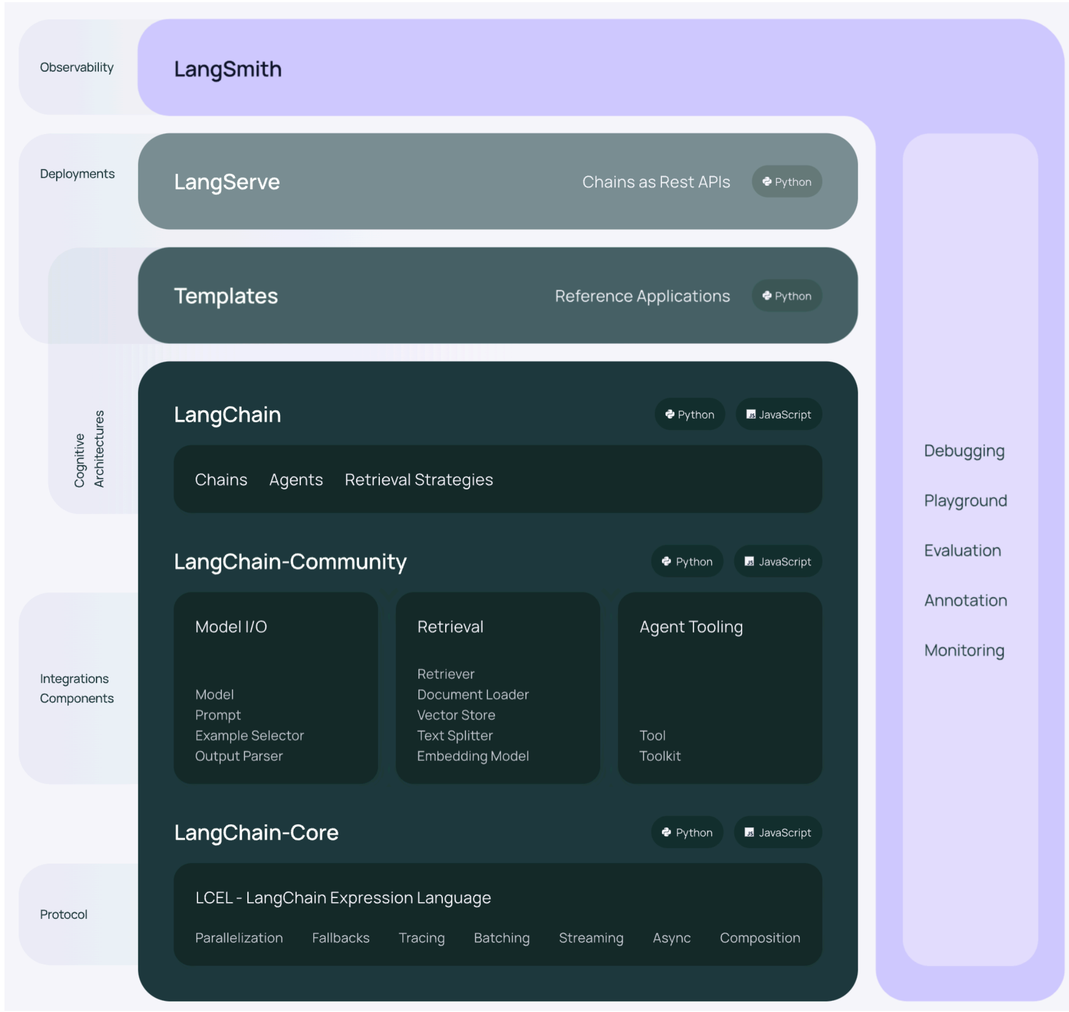
LangChain agent

- multiple calls to LLM
- calls defined online based on input & results
- performance optimized via flexible online tool selection
  - agent controller online reasoning about results from different tools
  - calls based on own results, CoT and thoughts (observations)

LangChain website

LangChain

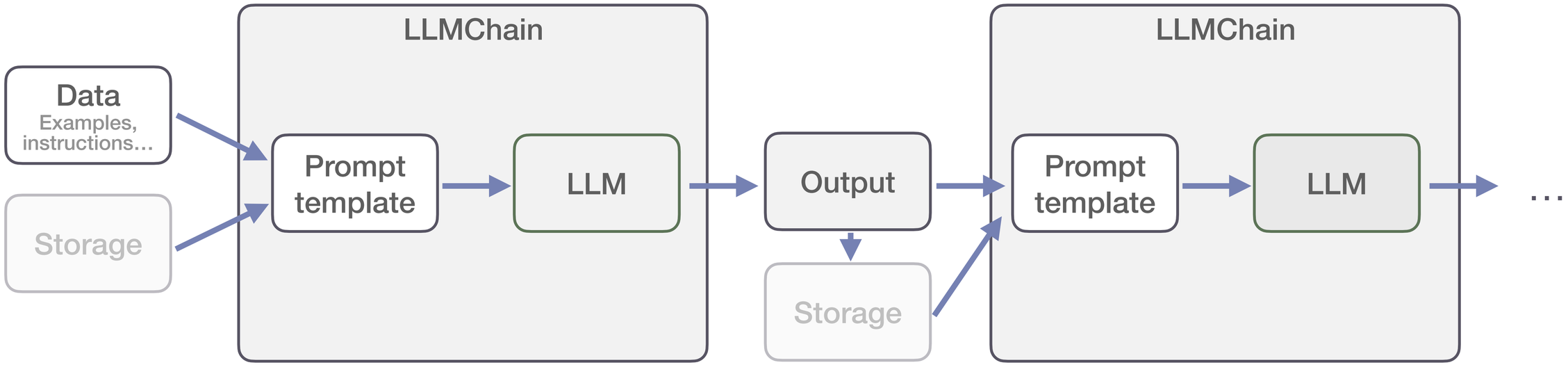
- framework for developing LLM applications
- supplies coding elements
  - building blocks
  - components
  - third-party integrations
- supplies dev structure
  - debugging, monitoring
  - serving (as API)
- supports Python and JS



LangChain website

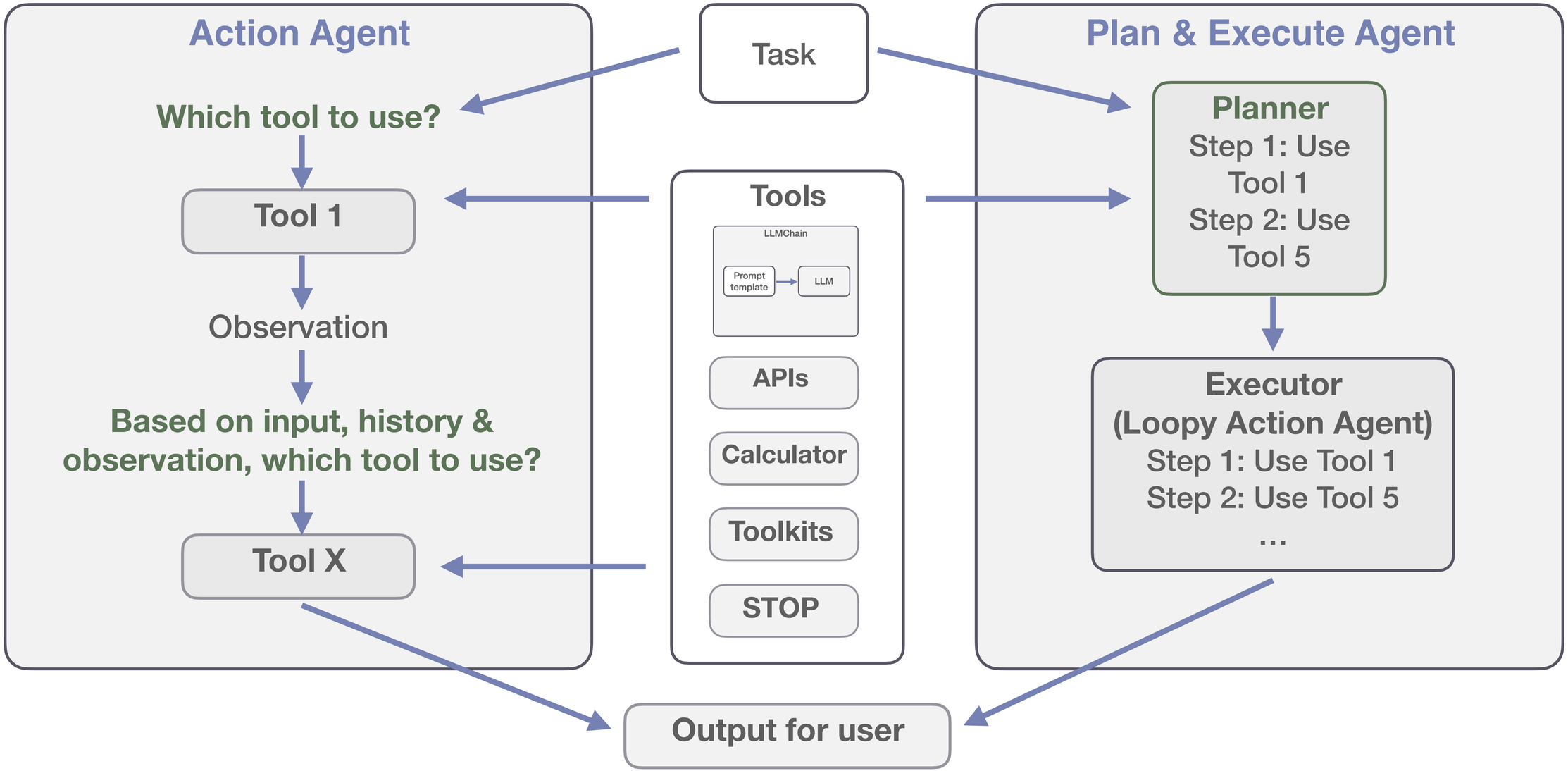
LangChain

simple linear prompt-generation chain



LangChain website

LangChain
agent models



LangChain website

LLM MODULO

POV: LLMs cannot plan / act as verifiers

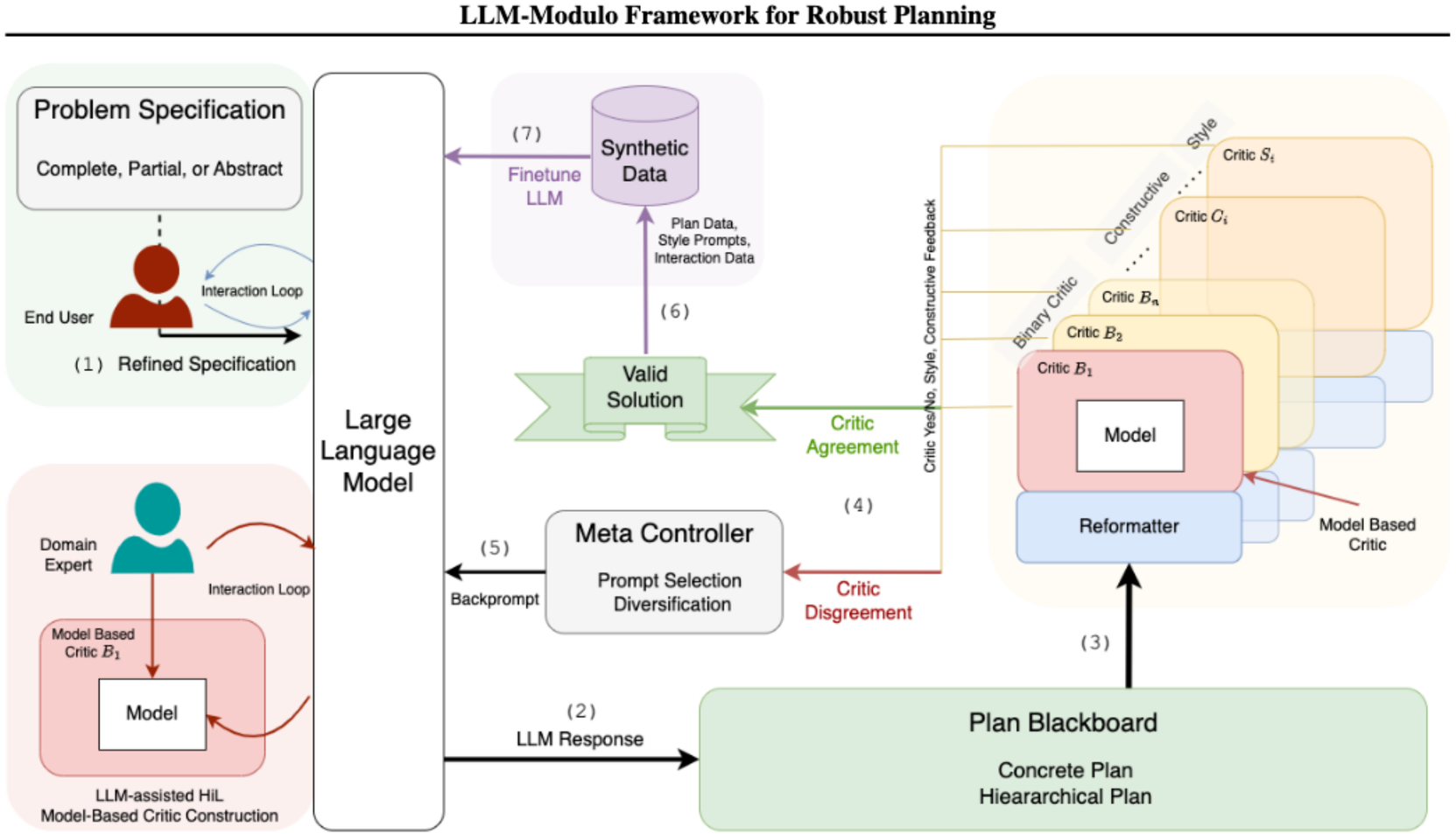


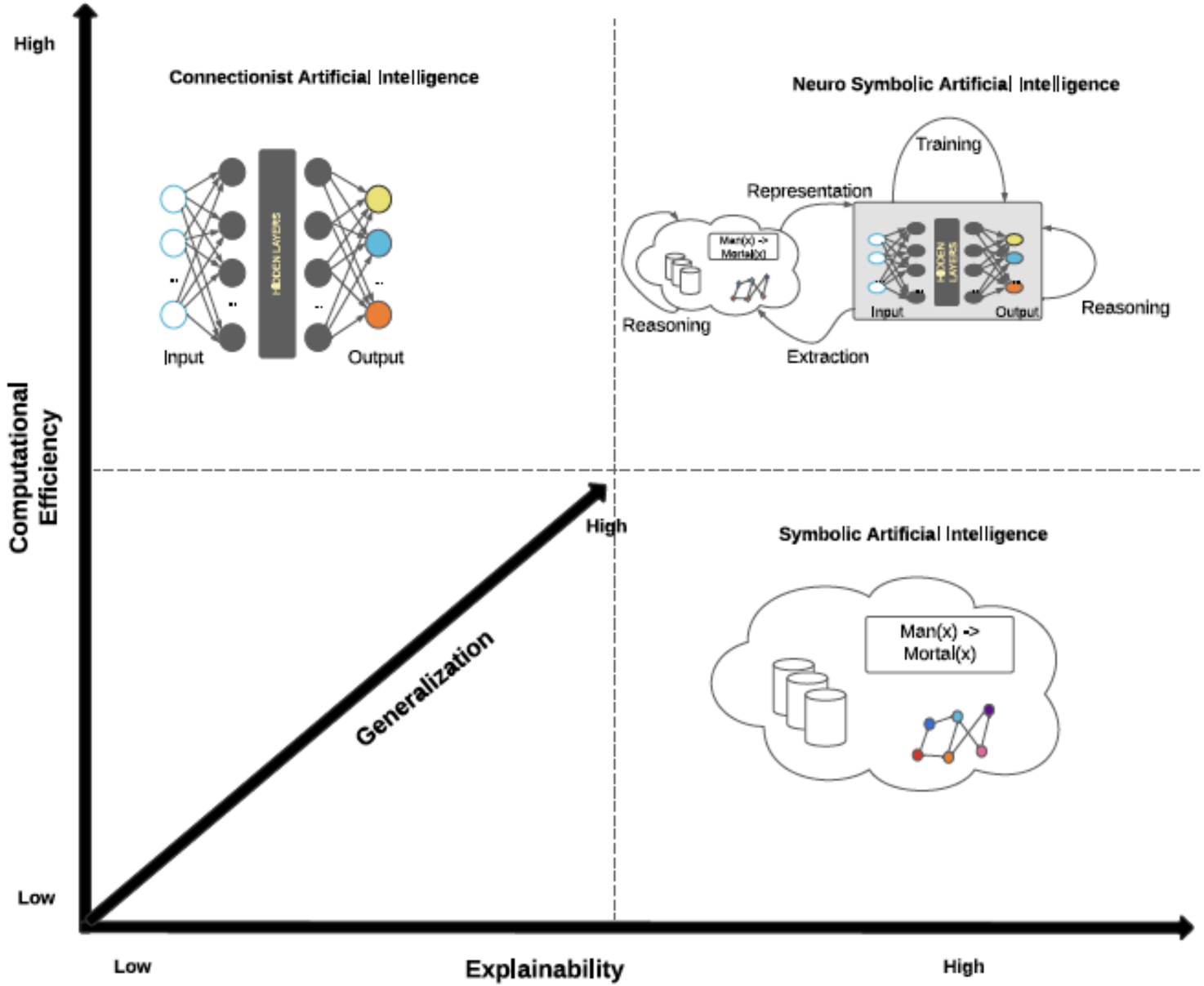
Figure 3. The proposed LLM-Modulo framework where LLMs act as idea generators and various external critics that specialize in different aspects, critique the candidate plan.

Kambaphati et al. (2024)

Neuro-symbolic (cognitive) modeling

w/ LLMs

Why neuro-symbolic modeling?



Bhuyan et al (2024)

Integrating neural models with symbolic reasoning

- idea: integration of neural networks with powerful reasoning mechanisms, learning mechanisms, decision mechanisms, knowledge representations / structures

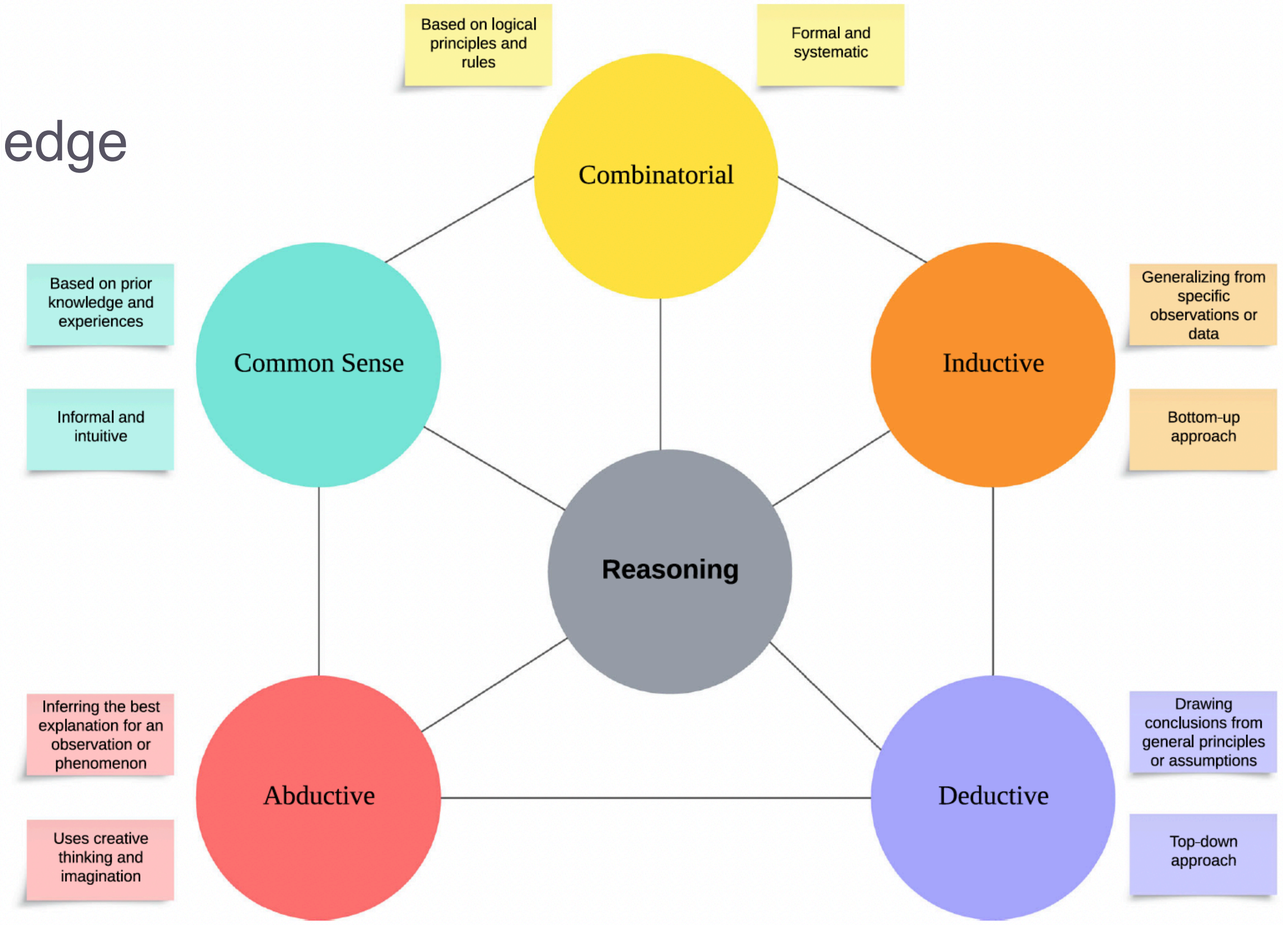
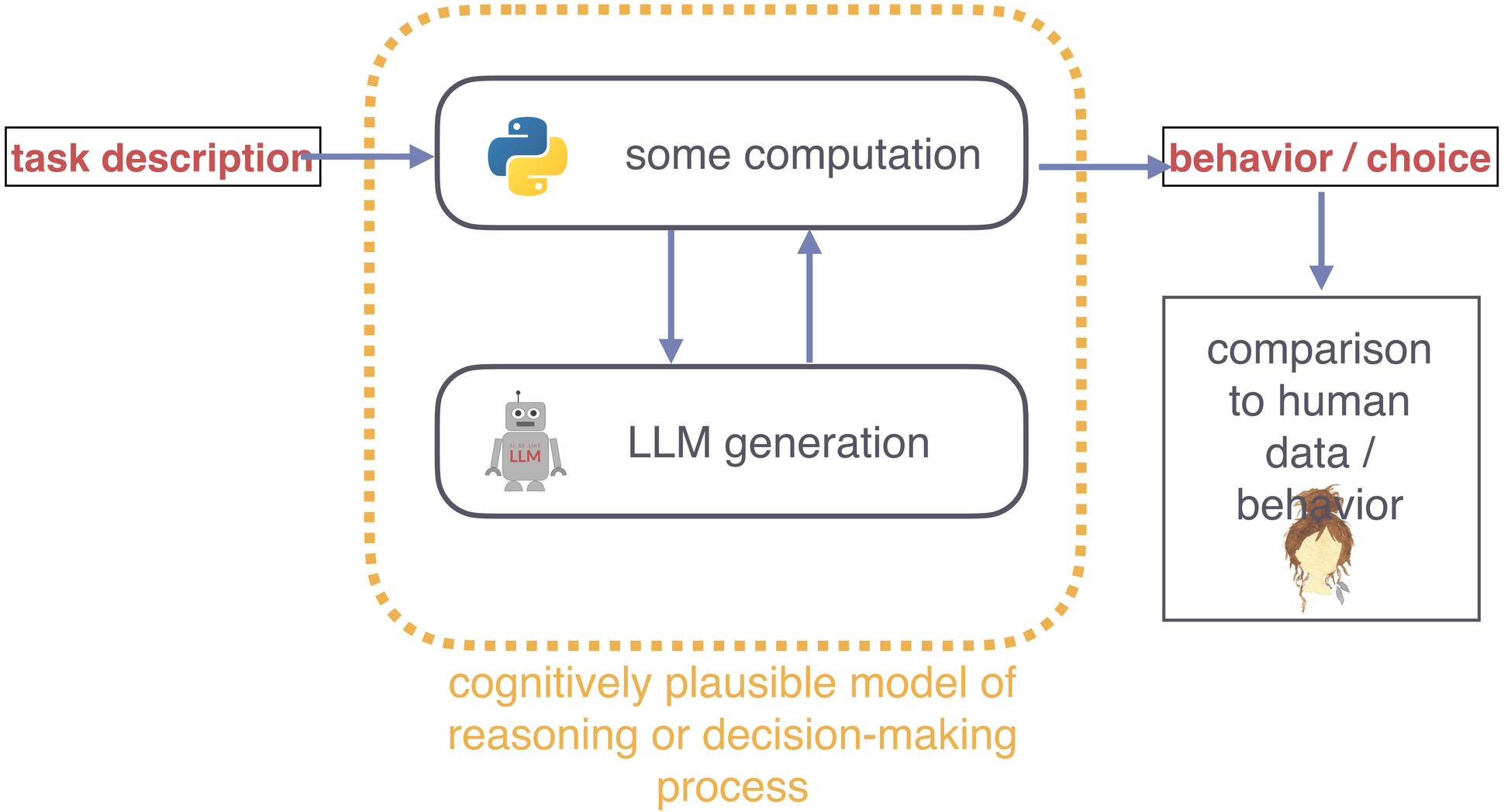


Fig. 6 Different types of reasoning which are not mutually exclusive and can often be used in combination with one another

Bhuyan et al (2024)

Neuro-symbolic cognitive models

:= LLM-based agent models to explain human reasoning / behavior



Neuro-symbolic modeling

LLM-generated world knowledge

- neuro-symbolic model
  - implemented in Gen (Julia)
  - PPL execution with LLM samples
- application: structured reasoning
- LLM inclusion
  - XLNet (masked LM)
  - fill-in-blank task

proposer / generator

- suggests contingencies

evaluator / scorer / ranker

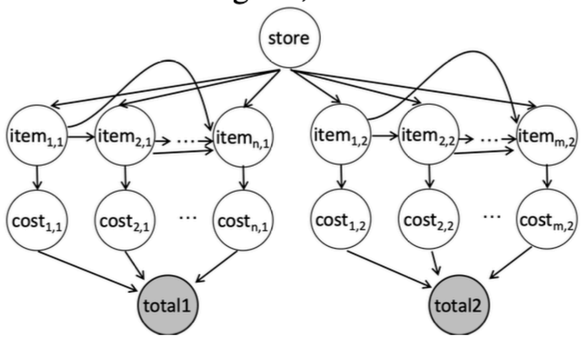
- evaluates / compares

reasoner / selector / decision maker

- select, rank or rule out options

Friends Going Shopping

"My friend and I went to the same store. I bought n items for a total of $total_1$, and my friend bought m items for a total of $total_2$. What store did we go to, and what did we each buy?"




```
query(friends_shopping(n=2, m=1),
      observe(:total_1 => 30,
              :total_2 => 150))
```

Sample:
corner (store),
sandwich & cake,
bicycle

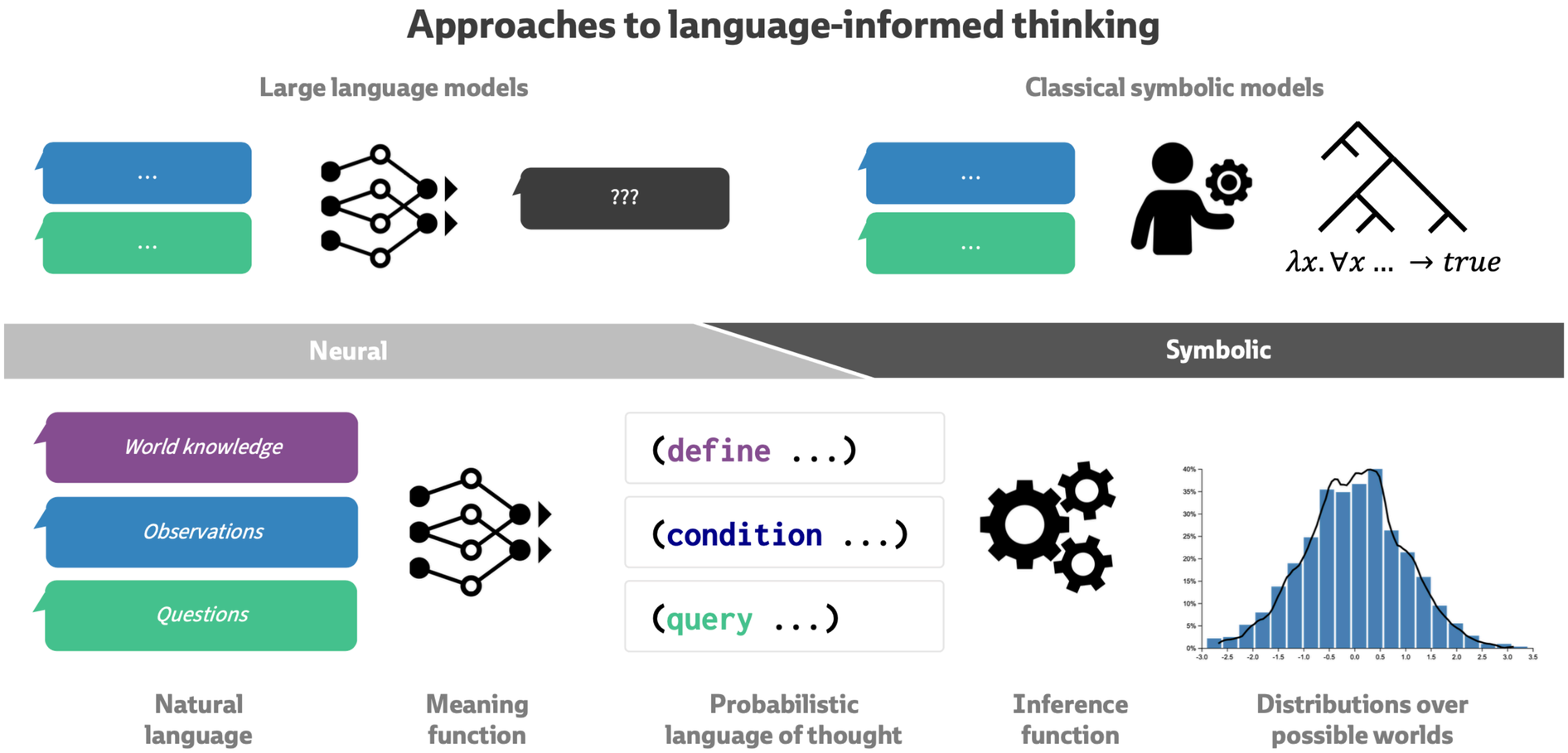
Program 2: A Structured Model of Shopping with Unstructured Statistical Knowledge

```
function go_shopping()
 store = noun("I went to the [?] store.")
 num_items = poisson(3)

 for i=1:num_items
  items[i]  = noun("I bought this [?]
                    at the $(store) store.")
  prices[i] = associated_quantity(
                items[i], "dollars")
 end

 total = sum(prices)
 return store, items, prices, total
end

query(go_shopping, observe(:store => "grocery"))
query(go_shopping, observe(:total => 500))
```


Lew et al. (2020), Tsvilodub, Carcassi & Franke (2024, in prep)

Rational meaning construction

computational framework for language-informed thinking



Wong et al (2023)

Integrating neural models with symbolic reasoning

Example NeSy model types

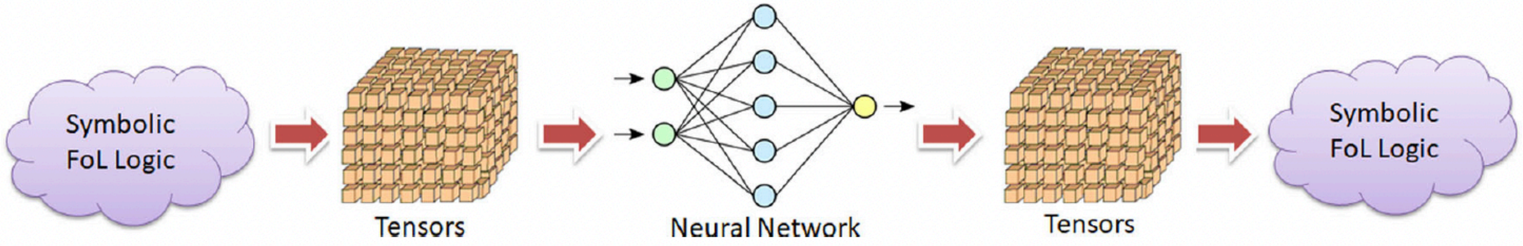


Fig. 12 Type 5 neuro-symbolic AI with tensor-based transformation. This visualization presents the conversion of symbolic first-order logic (FoL) into tensors, processed by a neural network, and then re-converted into symbolic FoL, highlighting a system where symbolic logic is seamlessly integrated with tensorial neural computation

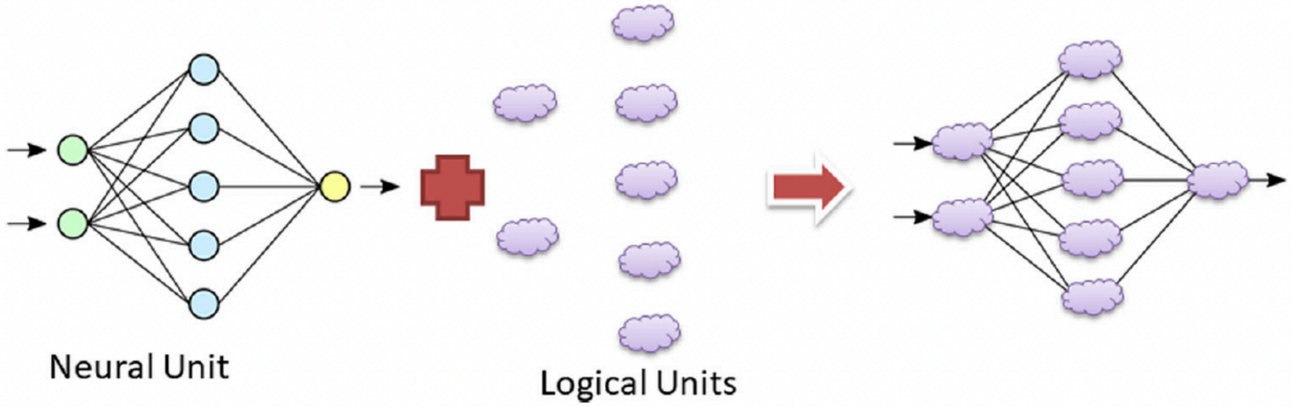
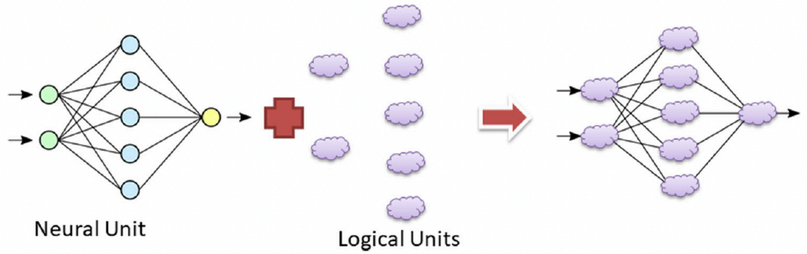


Fig. 13 Type 6 neuro-symbolic AI integration model. The process begins with a neural unit that feeds into a series of logical units, symbolizing the transition from sub-symbolic neural processing to higher-level logical reasoning. This represents an advanced form of integration where the neural network output is not just interpreted but also informs and shapes logical unit operations. This illustration conceptualizes the ideal of a fully integrated system, embedding a symbolic reasoning engine within a neural framework. As proposed by Kautz, it symbolizes the aspiration for a comprehensive AI model capable of both Kahneman's intuitive (System 1) and deliberate (System 2) thinking processes

Bhuyan et al (2024)

Integrating neural models with symbolic reasoning

Example NeSy model types



Question:
A revolving door is convenient for two direction travel, but it also serves as a security measure at a what?
Answer choices:
A. Bank*; B. Library; C. Department Store; D. Mall; E. New York;

Table 2. An example from the `CommonsenseQA` dataset; the asterisk (*) denotes the correct answer.

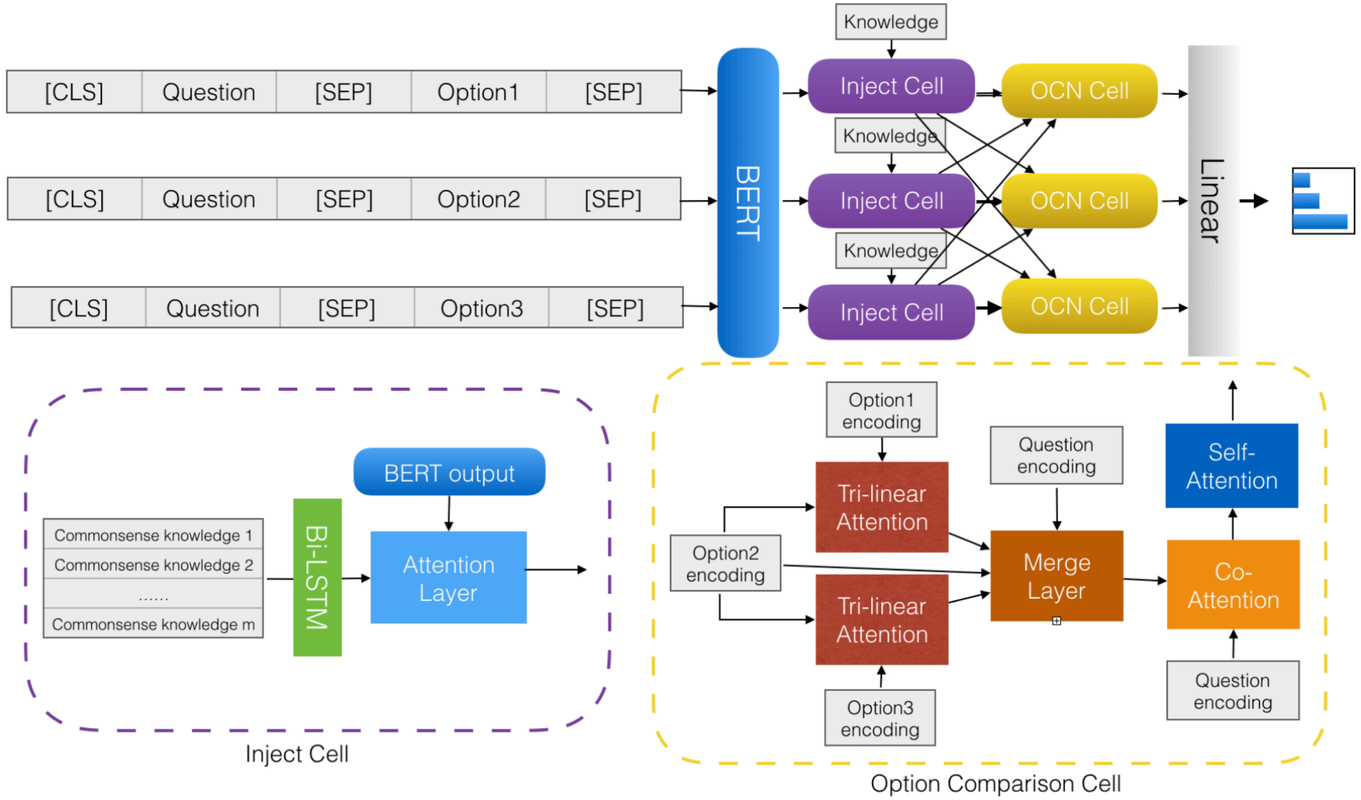


Figure 5. Option Comparison Network with Knowledge Injection

Models	Dev Acc
BERT + OMCS pre-train(*)	68.8
RoBERTa + CSPT(*)	**76.2**
OCN	64.1
OCN + CN injection	67.3
OCN + OMCS pre-train	65.2
OCN + ATOMIC pre-train	61.2
OCN + OMCS pre-train + CN inject	**69.0**

Table 4. Results on `CommonsenseQA`; the asterisk (*) denotes results taken from leaderboard.

Bhuyan et al (2024), Oltramari et al (2020)

Evaluation of LLM agents

CSP-Subheading

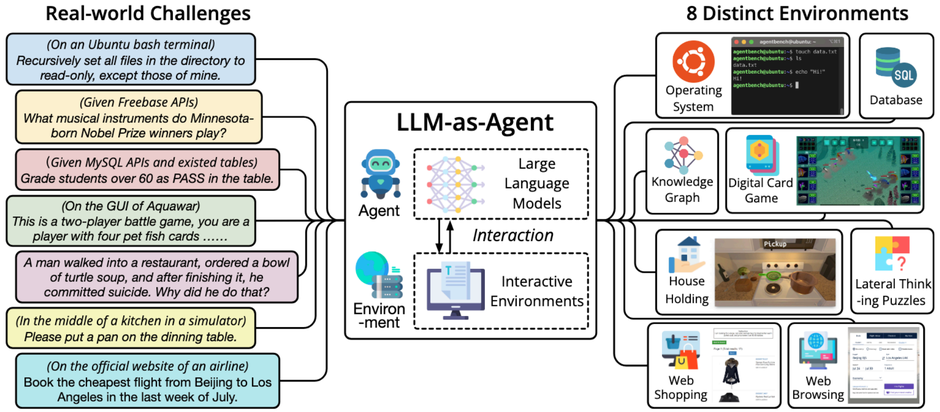


Figure 2: AGENTBENCH is the first systematic benchmark to evaluate LLM-as-Agent on a wide array of real-world challenges and 8 distinct environments. In total, 27 LLMs are examined in this edition. concepts of artificial intelligence (AI) historically. Notwithstanding substantial advancements in deep learning algorithms applied in both computer vision and natural language processing (NLP), their potential for developing efficient and practically usable assisting agents remains largely unexplored.

SWE-bench

Can Language Models Resolve Real-World GitHub Issues?

ICLR 2024

Carlos E. Jimenez*, John Yang*,
Alexander Wettig, Shunyu Yao, Kexin Pei,
Ofir Press, Karthik Narasimhan

Paper | Code | Submit | Analysis

Sotopia

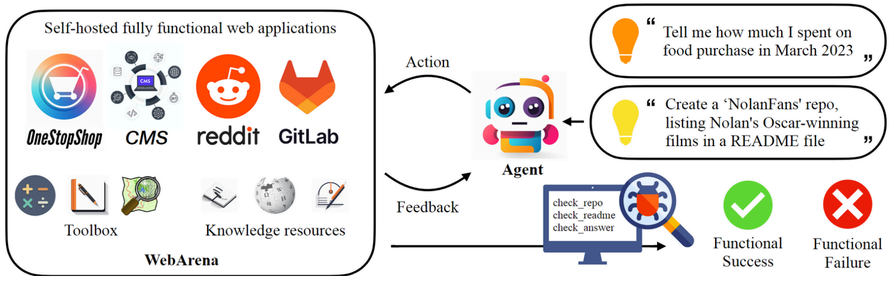
A research platform for realistic social simulations,
& an incubator for social intelligence.

WebArena: A Realistic Web Environment for Building Autonomous Agents

Shuyan Zhou[1*], Frank F. Xu[1*],
Hao Zhu[1+], Xuhui Zhou[1+], Robert Lo[1+], Abishek Sridhar[1+],
Xianyi Cheng[1], Tianyue Ou[1], Yonatan Bisk[1], Daniel Fried[1], Uri Alon[1], Graham Neubig[1,2].
[1]Carnegie Mellon University, [2]Inspired Cognition
[*]Lead contributors. [+]Equal contribution.
{shuyanzh,fangzhex,gneubig}@cs.cmu.edu

Paper | Code | Data | Docker Environment | Leaderboard

Our new benchmark TheAgentComanpy!



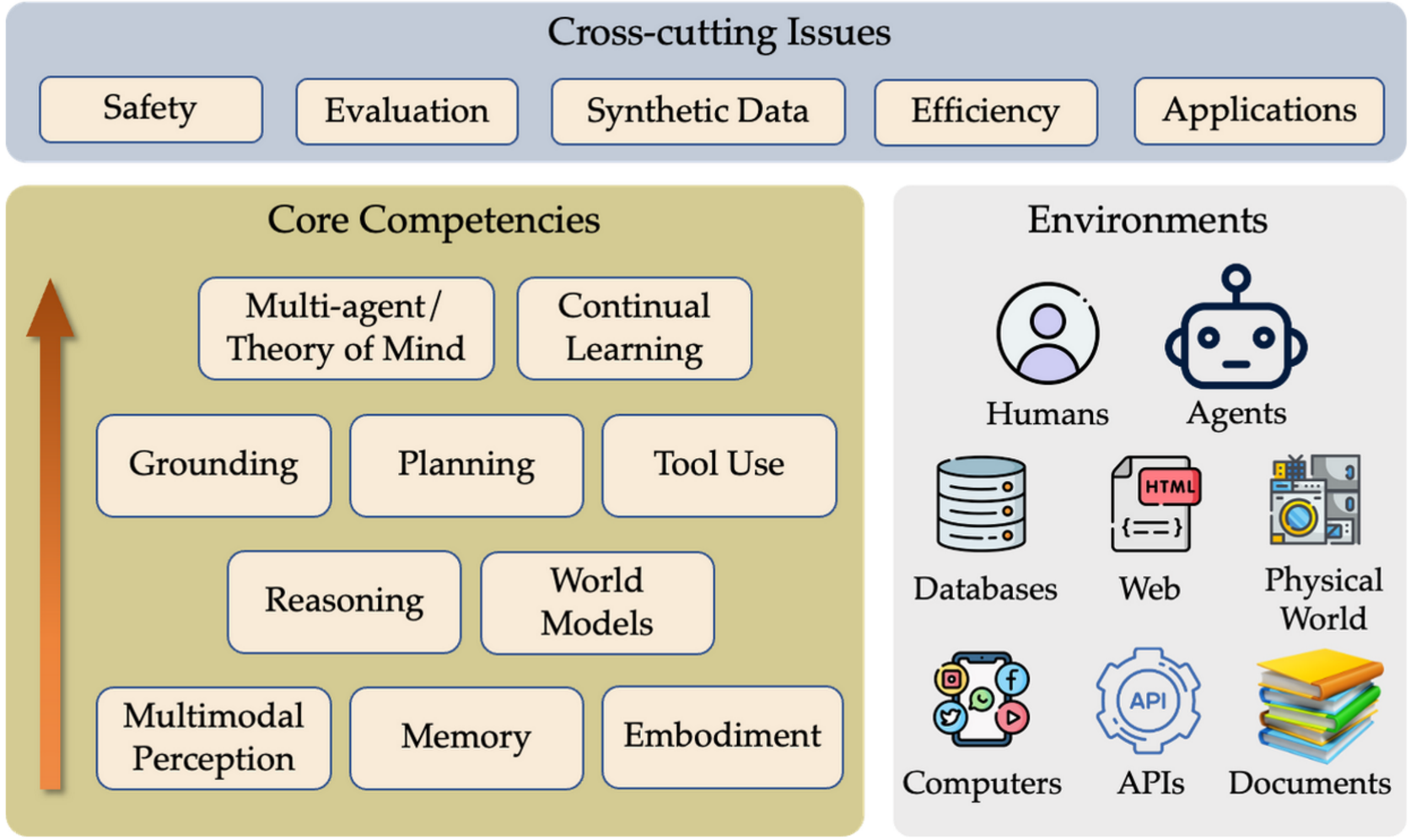
Paper discussion logistics

- **First discuss questions from Moodle!**
- How are the “cognitive” terms defined and used in the paper? To what extend do they correspond to concepts we have been discussing?
- What are limitations of this approach? How could we conduct targeted assessment of whether its components work as ‘advertised’?

1. split in two groups, half of the experts in each group
2. discuss the paper (e.g., start with questions of non-expert participants)
3. experts: responsible for adding key points, insights, new questions of the group to shared Google slides: https://docs.google.com/presentation/d/1BH53A2ipfzrix9C0gR39Cd1uUIHZaIZ2_CseZmhe81U/edit?usp=sharing
   a. maximally 3 slides!
   b. make slides such that they will be helpful for exam!
4. joint discussion

Wrap-up

A conceptual framework for language agents



Sumers et al. (2023)

Outlook

- There are many open questions, some of them reviewed, e.g., here
- Evaluation of agent system components remains an open question
- If you are interested in learning more about active research in related domains, check out, e.g., proceedings of workshops in EMNLP, NeurIPS, follow researchers you like
- Reflect, think, and engage in research!
  - If you want to talk more about your own ideas and questions, feel free to reach out to me any time!

Junior Group Leader
NLP / CogSci

- **4 years E14 (100%)**
- VW-Momentum + ML Excellence Cluster
- theoretical / CogSci perspective on language modeling
  - behavioral assessment
  - mechanistic interpretability
  - euro-symbolic models
  - ….
- **apply by March 15th**
- starting date ~ June ‘25
- job description

PostDoc
pragmatics / CogSci

- **2.5 year E13 (100%)**
- DFG-AHRC project
  - joint project w/ Daniel Lassiter (Edinburgh)
- experiments & computational modeling on pragmatic language use to communicate information about causality
- **apply by March 15th**
- starting date ~ June ’25
- job description

PhD
pragmatics / CogSci

- **3 year E13 (65%)**
- DFG-AHRC project
  - joint project w/ Daniel Lassiter (Edinburgh)
- experiments & computational modeling on pragmatic language use to communicate information about causality
- **apply by March 15th**
- starting date ~ June ’25
- job description