Reasoning about Actions with Large Multimodal Models

Committee

Professors Raymond Mooney (Advisor), David Harwath, Elias Stengel-Eskin, and Jacob Andreas

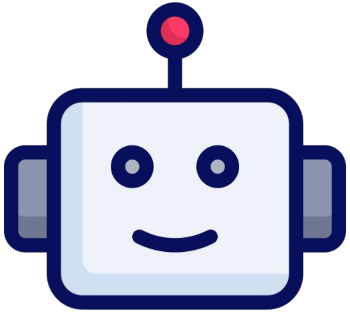
Vanya Cohen

PhD Candidate, The University of Texas at Austin

Contents

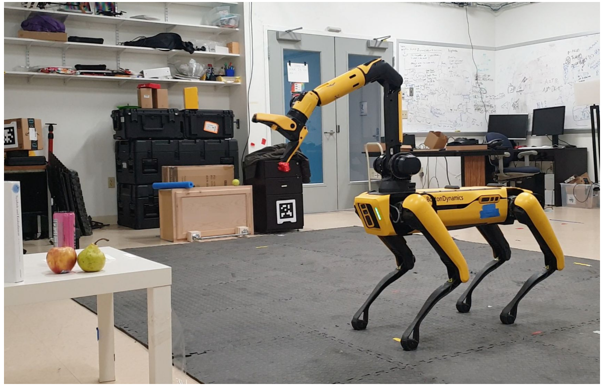
- Introduction and Background
- Completed Work
- Future Work

Contents



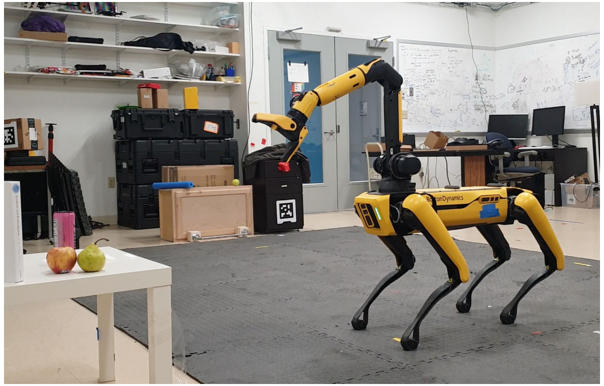
Natural language is an accessible means of specifying tasks for AI agents.

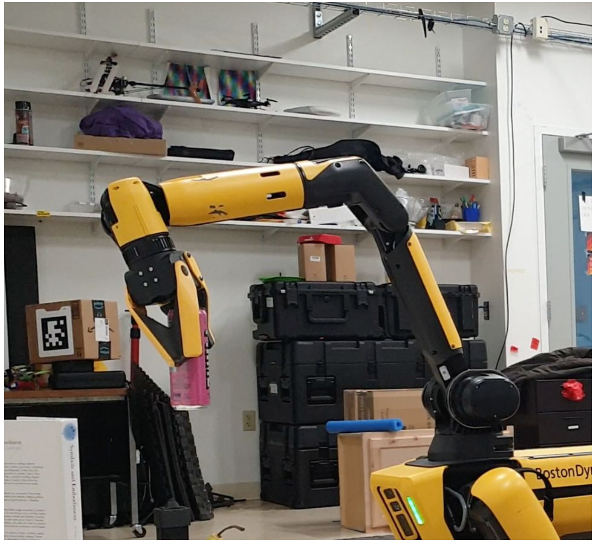
“Pick up my energy drink and place it by my running shoes.”



“Pick up my energy drink and place it by my running shoes.”

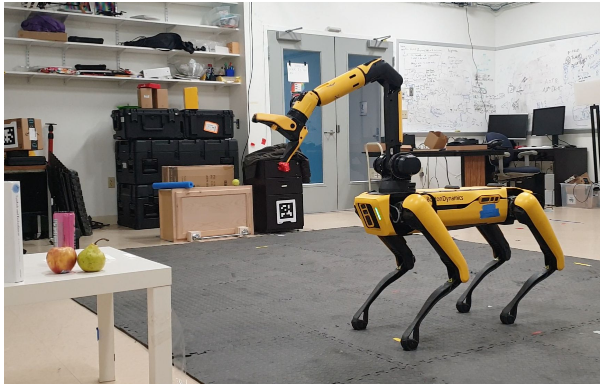
CAPE: Corrective Actions from Precondition Errors using Large Language Models
Shreyas Sundara Raman, **Vanya Cohen,** Ifrah Idrees, Eric Rosen, Ray Mooney, Stefanie Tellex, David Paulius.
ICRA 2024.

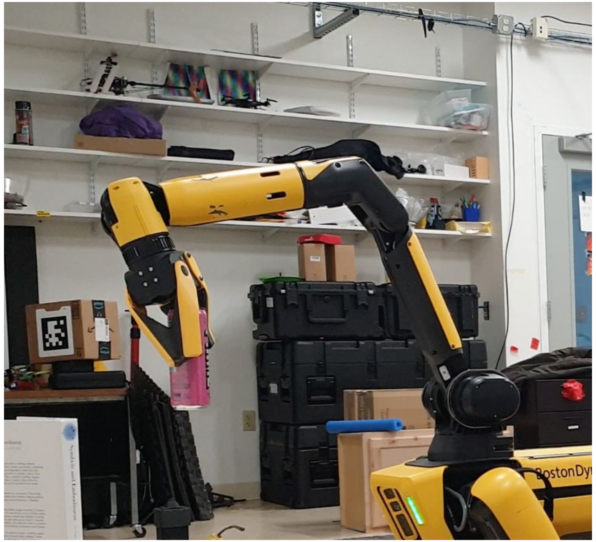




“Pick up my energy drink and place it by my running shoes.”

CAPE: Corrective Actions from Precondition Errors using Large Language Models
Shreyas Sundara Raman, **Vanya Cohen,** Ifrah Idrees, Eric Rosen, Ray Mooney, Stefanie Tellex, David Paulius.
ICRA 2024.

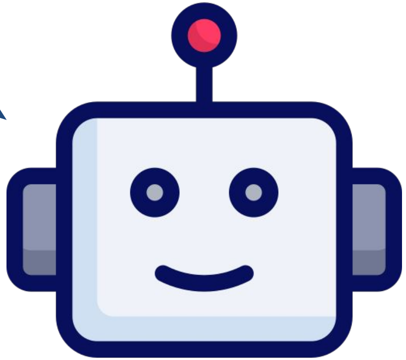


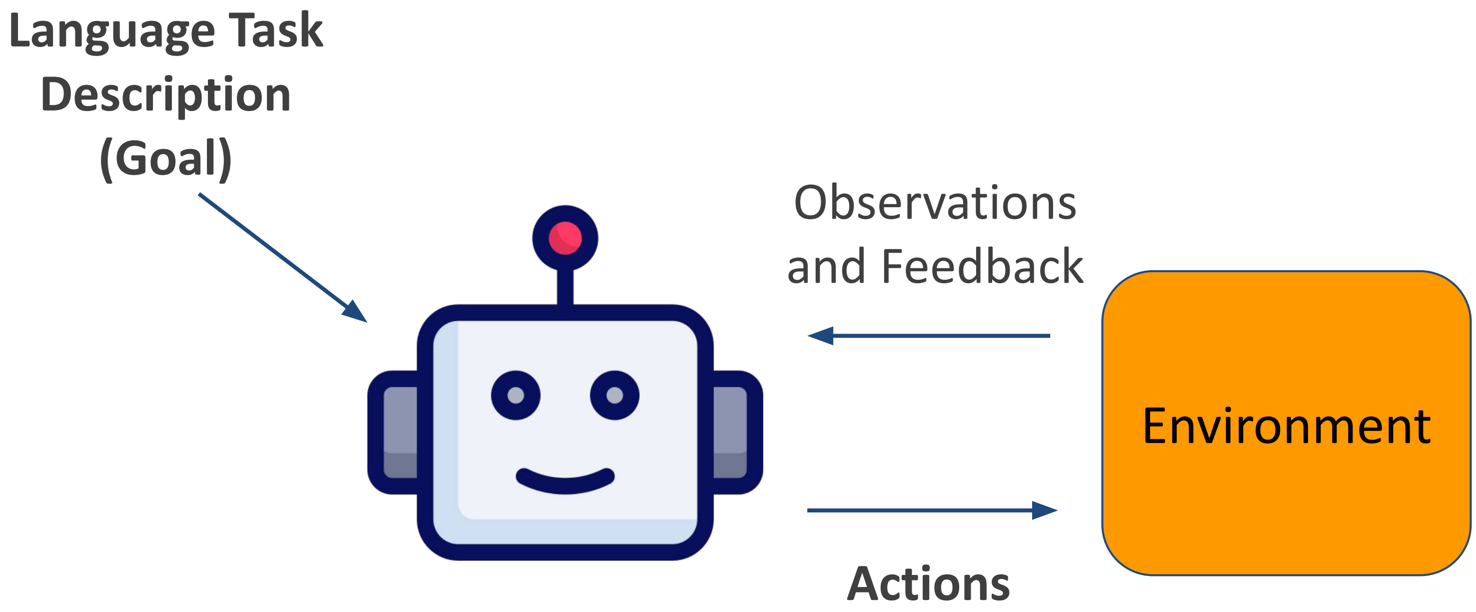


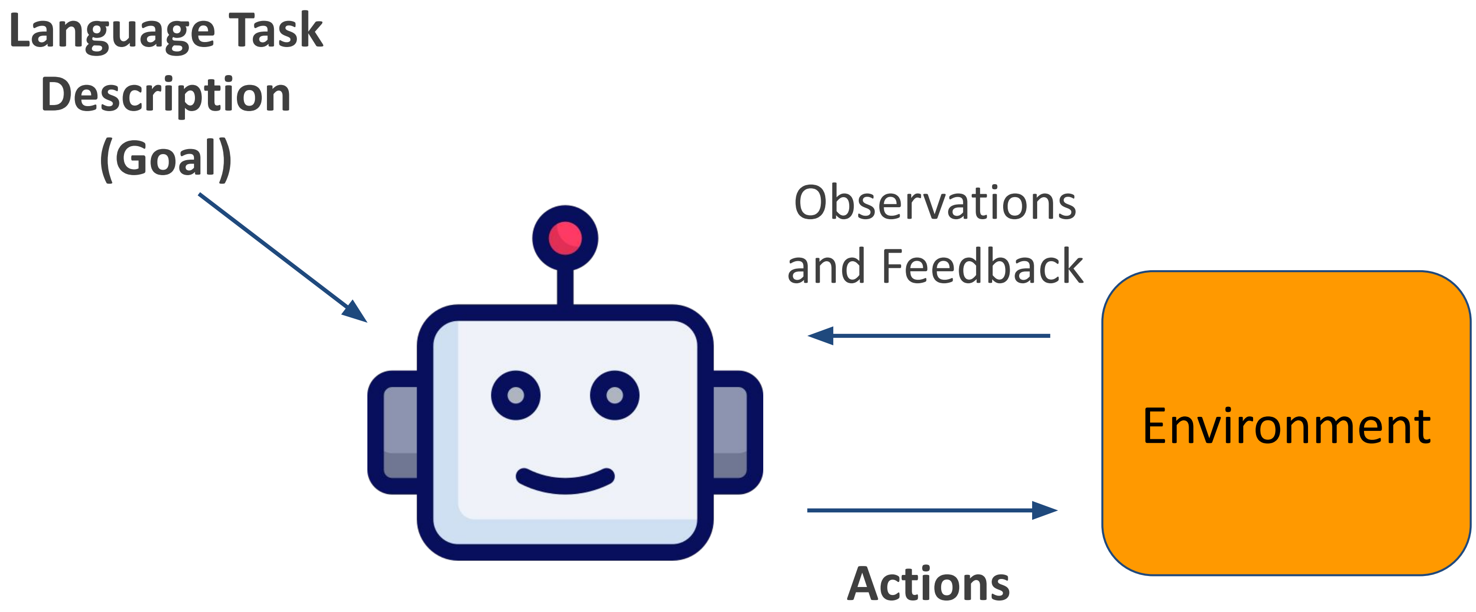
"Pick up my energy drink and place it by my running shoes."

CAPE: Corrective Actions from Precondition Errors using Large Language Models
Shreyas Sundara Raman, **Vanya Cohen,** Ifrah Idrees, Eric Rosen, Ray Mooney, Stefanie Tellex, David Paulius.
ICRA 2024.

Language Task Description (Goal)

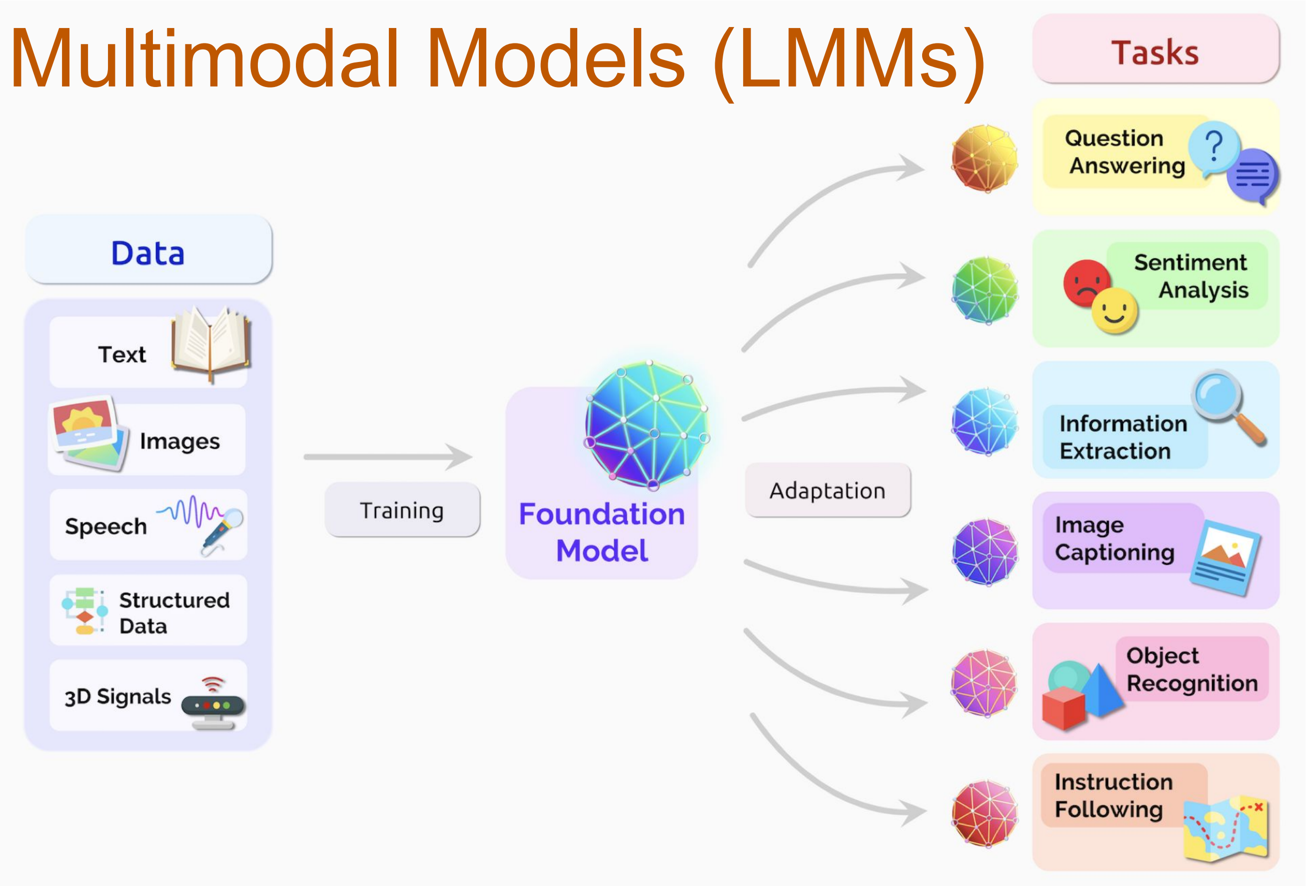



Language Task
Description
(Goal)
Observations
and Feedback
Environment
Actions



Produce **valid actions** to **accomplish a task** specified in **natural language**.

Large Multimodal Models (LMMs)



Center for Research on Foundation Models 2022

How can we **improve** and **evaluate** action reasoning in LMMs?

What is action reasoning?

1. Relate **goals to action sequences.**

What is action reasoning?

1. Relate **goals to action sequences.**
2. **Preconditions:** What must the world state be for the action to be afforded (executable)?

What is action reasoning?

1. Relate **goals to action sequences.**
2. **Preconditions:** What must the world state be for the action to be afforded (executable)?
3. **Postconditions:** What is the world state after executing this action? I.e. World Modeling.

Contents

- Introduction and Background
- **Completed Work**
- Future Work

Completed Work

Improvement

Planning-Augmented Parsing *ACL NLRSE 2023*

Using Planning to Improve Semantic Parsing of Instructional Texts

Planning constraints rank/correct candidate action sequences.

Compositional Instruction Following *TMLR 2024 / RLC 2025*

Compositional Instruction Following with LMs and RL

Learns language-to-policy composition from environment feedback.

Evaluation

CaT-Bench *EMNLP 2024*

Causal and Temporal Dependencies in Plans

Tests whether LMs understand preconditions, step ordering, and dependence.

MET-Bench *ICML 2026*

Multimodal Entity State Tracking

Tests postconditions and entity state tracking across text and images.

Completed Work

Improvement

Planning-Augmented Parsing *ACL NLRSE 2023*

Using Planning to Improve Semantic Parsing of Instructional Texts

Planning constraints rank/correct candidate action sequences.

Compositional Instruction Following *TMLR 2024 / RLC 2025*

Compositional Instruction Following with LMs and RL

Learns language-to-policy composition from environment feedback.

Evaluation

CaT-Bench *EMNLP 2024*

Causal and Temporal Dependencies in Plans

Tests whether LMs understand preconditions, step ordering, and dependence.

MET-Bench *ICML 2026*

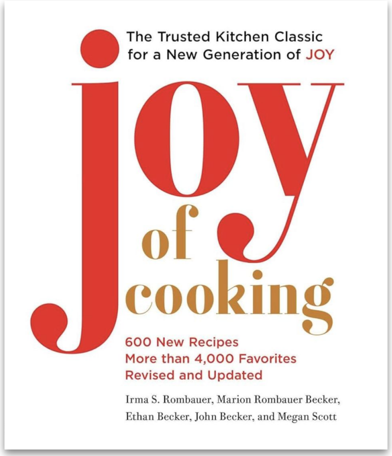
Multimodal Entity State Tracking

Tests postconditions and entity state tracking across text and images.

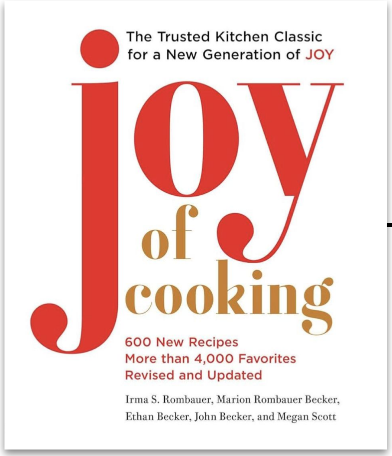
Using Planning to Improve Semantic Parsing of Instructional Texts

Vanya Cohen and Raymond Mooney

Instructional Texts



Instructional Texts



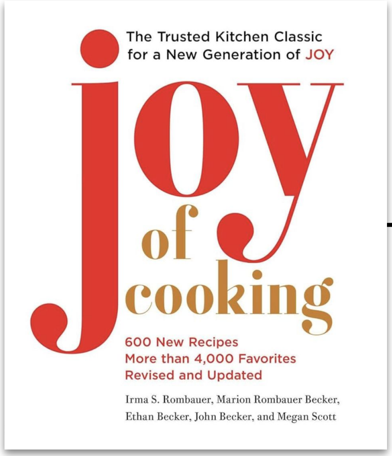
Large Pretrained Model

Executable Plan?

(get-ingredient flour)
(get-ingredient sugar)
(get-ingredient butter)
(get-ingredient eggs)
(get-ingredient chips)
(preheat-oven)
(scoop-dough)
(mix-dough)
(bake-cookies)
(cool-cookies)

May not know or respect domain constraints.

Instructional Texts



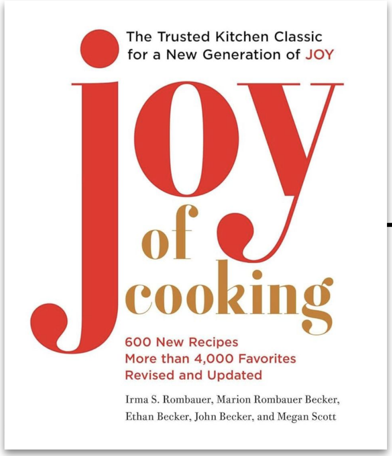
Large Pretrained Model

Executable Plan?

(get-ingredient flour)
(get-ingredient sugar)
(get-ingredient butter)
(get-ingredient eggs)
(get-ingredient chips)
(preheat-oven)
(scoop-dough)
(mix-dough)
(bake-cookies)
(cool-cookies)

May not know or respect domain constraints.

Instructional Texts



Large Pretrained Model

Executable Plan?

(get-ingredient flour)
(get-ingredient sugar)
(get-ingredient butter)
(get-ingredient eggs)
(get-ingredient chips)
(preheat-oven)
(scoop-dough)
(mix-dough)
(bake-cookies)
(cool-cookies)

Introduction

- Few-shot semantic parsing[1] of long-form instructional texts poses unique challenges.
  - Long context dependencies.
  - Ambiguous, domain-specific language.
    - Omitted and implied steps.

1. Richard Shin, Benjamin Van Durme. Few-Shot Semantic Parsing with Language Models Trained on Code. NAACL 2022

Planning Domain

```
(:action preheat-oven
   :parameters ()
   :precondition (and (not
(preheated)))
   :effect (preheated)
 )

 (:action mix-dough
   :parameters ()
   :precondition (and
…
```

Problem (Constraints)

```
(define (problem
make-choc-chip-cookies)
 (:domain cookie-making)
 (:objects)
 (:init
   (needed flour)
   (needed sugar)
   (needed butter)
   (needed eggs)
   (needed chips)
…
```

Planner

Executable Plan

```
(get-ingredient flour)
(get-ingredient sugar)
(get-ingredient butter)
(get-ingredient eggs)
(get-ingredient chips)
(preheat-oven)
(mix-dough)
(scoop-dough)
(bake-cookies)
(cool-cookies)
```

Related Work

- Prior work on long-form instruction parsing and few-shot parsing of individual commands.

Shin and Van Durme. Few-Shot Semantic Parsing with Language Models Trained on Code. NAACL 2022.
Papadopoulos et al. Learning Program Representations for Food Images and Cooking Recipes. CVPR 2022.

Method

- Utilize ***planning domain information*** to improve the quality of generated **semantic parses (plans*).***
- Ensure generated plans are **executable**: each action has **satisfied preconditions**.

Method

- Planning-Augmented Semantic Parsing
  - Symbolic-planning-based decoder.
  - Ranks and corrects candidate parses.
  - Combines the strengths of LLMs and classical AI planning.

Datasets

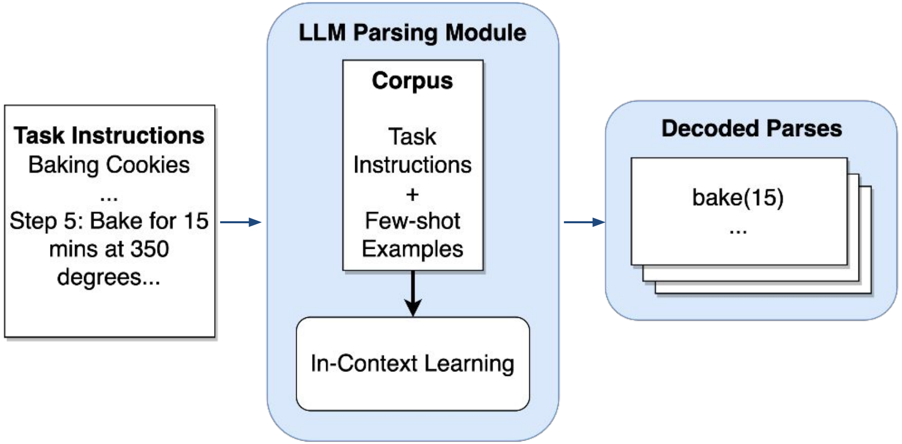
- Cooking recipes with ground-truth plans
  - Describe the steps needed to make the recipe.
  - **Bollini et al. 2013** and **Tasse and Smith 2008**.

Method

Task Instructions
Baking Cookies
...
Step 5: Bake for 15 mins at 350 degrees...

Pipeline starts with input instructions

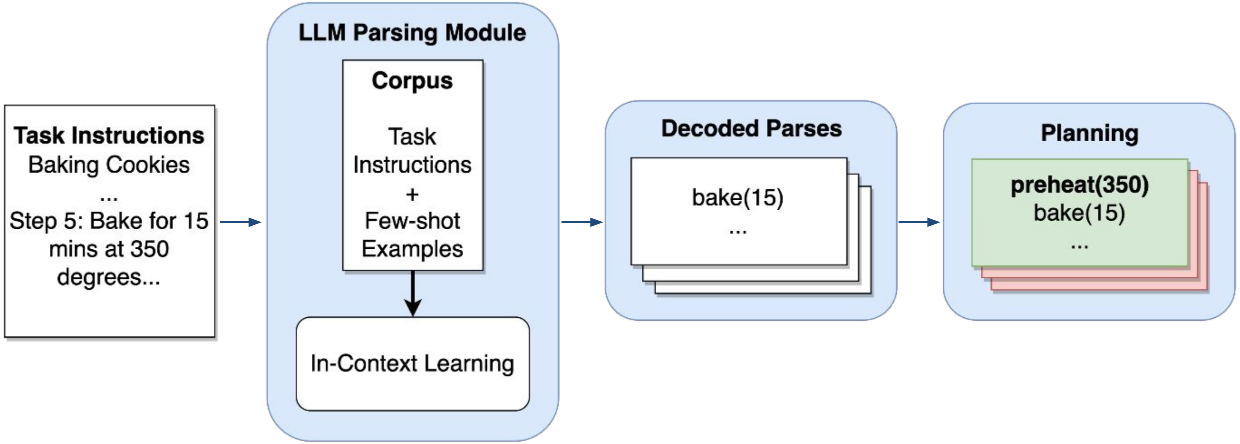
Method



Few-shot semantic parsing to translate the instructions into a sequence of operators. Produce ten candidates

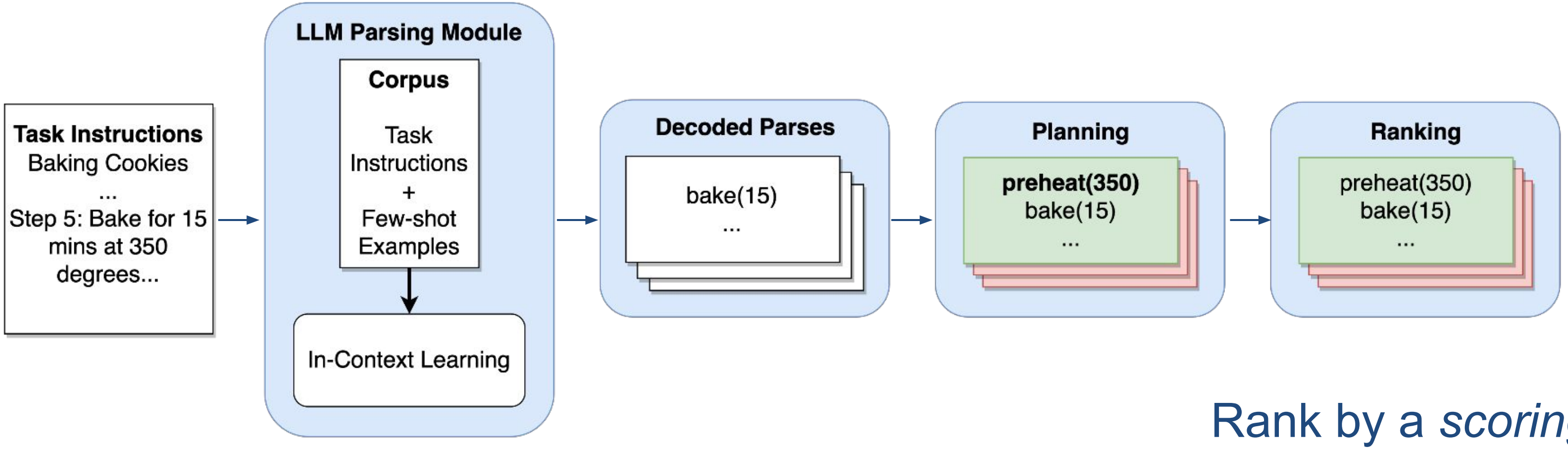
OpenAI Davinci Codex (Chen et al. 2021)

Method



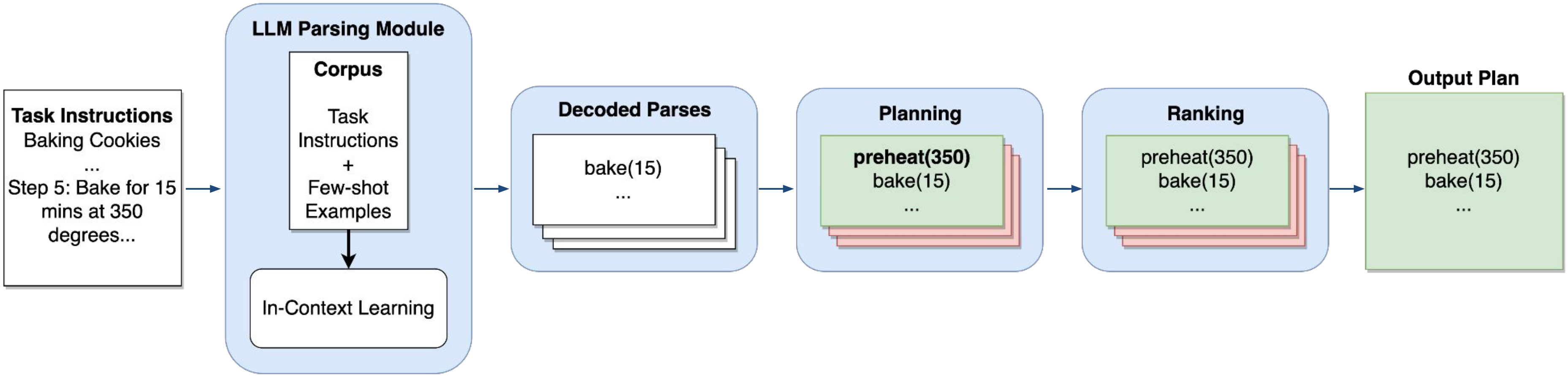
bake() preconditions not met, planning adds a preheat step

Method



Rank by a *scoring function* that prioritizes executable plans

Method


Task Instructions
Baking Cookies
...
Step 5: Bake for 15 mins at 350 degrees...
LLM Parsing Module
Corpus
Task Instructions
+
Few-shot Examples
In-Context Learning
Decoded Parses
bake(15)
...
Planning
preheat(350)
bake(15)
...
Ranking
preheat(350)
bake(15)
...
Output Plan
preheat(350)
bake(15)
...

Method

- Rank candidate parses (ten candidates)
  - **Minimize** syntax errors ***(SE)***, precondition errors ***(PE)***
  - **Minimize** the number of steps that need to be added to make the parsed plan executable ***(AS)***
  - **Maximize** the probability of all the plan steps **(*ln* P_t)**
- Output the highest scoring plan with added steps.

Experiments

- Evaluation Metrics
  - Longest Common Subsequence (LCS)
  - Plan Steps F1: harmonic mean of precision and recall of generated steps
  - Precondition Errors (PE) and Syntax Errors (SE)

Experiments

- Evaluation Metrics
  - Longest Common Subsequence (LCS)
  - Plan Steps F1: harmonic mean of precision and recall of generated steps
  - Precondition Errors (PE) and Syntax Errors (SE)

- Experimental Settings
  - **Rank (PPL):** Selects the plan with the lowest perplexity
  - **Rank:** Ranks the plans by the scoring function without correcting precondition errors
  - **Rank + Plan:** Our full ranking method with planning to correct errors

Results (Bollini et al. 2013)

Models	(Bollini et al., 2013)			
	LCS↑	PE↓	SE↓	F1 ↑
Rank (PPL)				
Davinci Codex, E=1	0.897 ± 0.008	0.962 ± 0.685	0.042 ± 0.008	0.784 ± 0.004
Davinci Codex, E=5	0.949 ± 0.005	0.198 ± 0.009	0.002 ± 0.004	0.863 ± 0.003
Rank				
Davinci Codex, E=1	0.901 ± 0.008	0.382 ± 0.037	0.025 ± 0.008	0.798 ± 0.002
Davinci Codex, E=5	0.952 ± 0.005	0.120 ± 0.015	0.002 ± 0.004	$\mathbf{0.868 \pm 0.002}$
Rank + Plan				
Davinci Codex, E=1	0.903 ± 0.008	0.143 ± 0.033	0.025 ± 0.008	0.807 ± 0.002
Davinci Codex, E=5	0.952 ± 0.005	$\mathbf{0.033 \pm 0.000}$	0.002 ± 0.004	$\mathbf{0.870 \pm 0.002}$

Results (Bollini et al. 2013)

Models	(Bollini et al., 2013)			
	LCS↑	PE↓	SE↓	F1 ↑
Rank (PPL)				
Davinci Codex, E=1	0.897 ± 0.008	0.962 ± 0.685	0.042 ± 0.008	0.784 ± 0.004
Davinci Codex, E=5	0.949 ± 0.005	0.198 ± 0.009	0.002 ± 0.004	0.863 ± 0.003
Rank				
Davinci Codex, E=1	0.901 ± 0.008	0.382 ± 0.037	0.025 ± 0.008	0.798 ± 0.002
Davinci Codex, E=5	0.952 ± 0.005	0.120 ± 0.015	0.002 ± 0.004	$\mathbf{0.868 \pm 0.002}$
Rank + Plan				
Davinci Codex, E=1	0.903 ± 0.008	0.143 ± 0.033	0.025 ± 0.008	0.807 ± 0.002
Davinci Codex, E=5	0.952 ± 0.005	$\mathbf{0.033 \pm 0.000}$	0.002 ± 0.004	$\mathbf{0.870 \pm 0.002}$

Results (Bollini et al. 2013)

Models	(Bollini et al., 2013)			
	LCS↑	PE↓	SE↓	F1 ↑
Rank (PPL)				
Davinci Codex, E=1	0.897 ± 0.008	0.962 ± 0.685	0.042 ± 0.008	0.784 ± 0.004
Davinci Codex, E=5	0.949 ± 0.005	0.198 ± 0.009	0.002 ± 0.004	0.863 ± 0.003
Rank				
Davinci Codex, E=1	0.901 ± 0.008	0.382 ± 0.037	0.025 ± 0.008	0.798 ± 0.002
Davinci Codex, E=5	0.952 ± 0.005	0.120 ± 0.015	0.002 ± 0.004	$\mathbf{0.868 \pm 0.002}$
Rank + Plan				
Davinci Codex, E=1	0.903 ± 0.008	0.143 ± 0.033	0.025 ± 0.008	0.807 ± 0.002
Davinci Codex, E=5	0.952 ± 0.005	$\mathbf{0.033 \pm 0.000}$	0.002 ± 0.004	$\mathbf{0.870 \pm 0.002}$

Results (Tasse and Smith 2008)

Models	(**Tasse and Smith, 2008**)			
	LCS↑	PE↓	SE↓	F1 ↑
Rank (PPL)				
Davinci Codex, E=1	0.692 ± 0.003	0.827 ± 0.086	0.875 ± 0.199	0.443 ± 0.002
Rank				
Davinci Codex, E=1	0.695 ± 0.004	0.293 ± 0.016	$\mathbf{0.226 \pm 0.024}$	0.446 ± 0.001
Rank + Plan				
Davinci Codex, E=1	0.695 ± 0.003	$\mathbf{0.000 \pm 0.000}$	$\mathbf{0.237 \pm 0.018}$	0.446 ± 0.001

Results (Tasse and Smith 2008)

Models	(Tasse and Smith, 2008)			
	LCS↑	PE↓	SE↓	F1 ↑
Rank (PPL)				
Davinci Codex, E=1	0.692 ± 0.003	0.827 ± 0.086	0.875 ± 0.199	0.443 ± 0.002
Rank				
Davinci Codex, E=1	0.695 ± 0.004	0.293 ± 0.016	$\mathbf{0.226 \pm 0.024}$	0.446 ± 0.001
Rank + Plan				
Davinci Codex, E=1	0.695 ± 0.003	$\mathbf{0.000 \pm 0.000}$	$\mathbf{0.237 \pm 0.018}$	0.446 ± 0.001

Results (Tasse and Smith 2008)

Models	(**Tasse and Smith, 2008**)			
	LCS↑	PE↓	SE↓	F1 ↑
Rank (PPL)				
Davinci Codex, E=1	0.692 ± 0.003	0.827 ± 0.086	0.875 ± 0.199	0.443 ± 0.002
Rank				
Davinci Codex, E=1	0.695 ± 0.004	0.293 ± 0.016	$\mathbf{0.226 \pm 0.024}$	0.446 ± 0.001
Rank + Plan				
Davinci Codex, E=1	0.695 ± 0.003	$\mathbf{0.000 \pm 0.000}$	$\mathbf{0.237 \pm 0.018}$	0.446 ± 0.001

Conclusion

- Neuro-symbolic approach that leverages planning domain information to improve few-shot semantic parsing.
- Reduces precondition errors while maintaining content similarity to ground-truth plans.

Improvement

Planning-Augmented Parsing *ACL NLRSE 2023*

Using Planning to Improve Semantic Parsing of Instructional Texts

Planning constraints rank/correct candidate action sequences.

Compositional Instruction Following *TMLR 2024 / RLC 2025*

Compositional Instruction Following with LMs and RL

Learns language-to-policy composition from environment feedback.

Evaluation

CaT-Bench *EMNLP 2024*

Causal and Temporal Dependencies in Plans

Tests whether LMs understand preconditions, step ordering, and dependence.

MET-Bench *ICML 2026*

Multimodal Entity State Tracking

Tests postconditions and entity state tracking across text and images.

Improvement	Evaluation
Planning-Augmented Parsing *ACL NLRSE 2023* Using Planning to Improve Semantic Parsing of Instructional Texts Planning constraints rank/correct candidate action sequences.	**CaT-Bench** *EMNLP 2024* Causal and Temporal Dependencies in Plans Tests whether LMs understand preconditions, step ordering, and dependence.
Compositional Instruction Following *TMLR 2024 / RLC 2025* Compositional Instruction Following with LMs and RL Learns language-to-policy composition from environment feedback.	**MET-Bench** *ICML 2026* Multimodal Entity State Tracking Tests postconditions and entity state tracking across text and images.

Compositional Instruction Following with Language Models and Reinforcement Learning

Vanya Cohen*, Geraud Nangue Tasse*, Nakul Gopalan, Steven James, Matthew Gombolay, Raymond Mooney, Benjamin Rosman

Reinforcement Learning Conference 2025; Transactions on Machine Learning Research 2024

Motivation

- Formal planning domains are unavailable in many environments, so agents must learn action effects through interaction.

Motivation

- Learning a separate policy for every language instruction is prohibitively sample-inefficient.

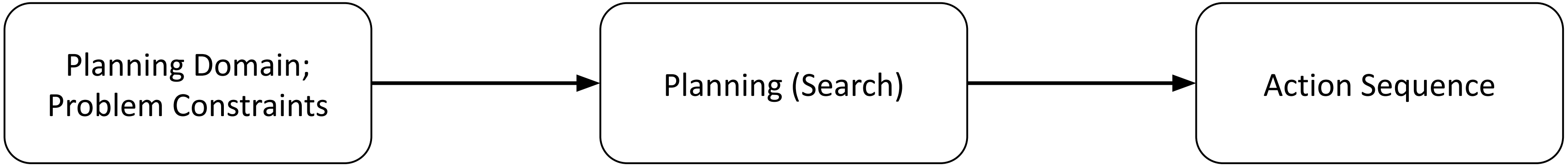
Motivation

- Language instructions and task policies often share compositional structure.

Motivation

- Use this to enable sample efficient learning and generalization.

Planning


Planning Domain;
Problem Constraints
Planning (Search)
Action Sequence

Reinforcement Learning (RL)

Markov Decision Process → RL: Learning by trial and error → **Policy**: Function describing what actions to take in what situations.

Low sample efficiency

Reinforcement Learning (RL)

Markov Decision Process

RL: Learning by trial and error

Policy: Function describing what actions to take in what situations.

Reinforcement Learning

Domain

Markov Decision Process

States: S
Actions: A
Reward(s, a) ⟶ scalar
Transition(s, a) ⟶ s'

RL Algorithm

E.g. Policy Gradient[1]; Deep Q Networks[2]

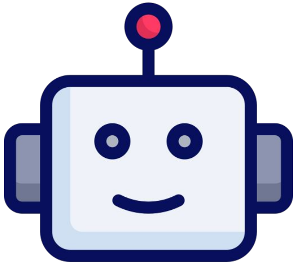
Policy Representation

Policy Function

$$\pi(a|s) = P(A_t = a \mid S_t = s)$$

Value Function

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, S_0 = s, A_0 = a\right]$$



State Observation and Reward

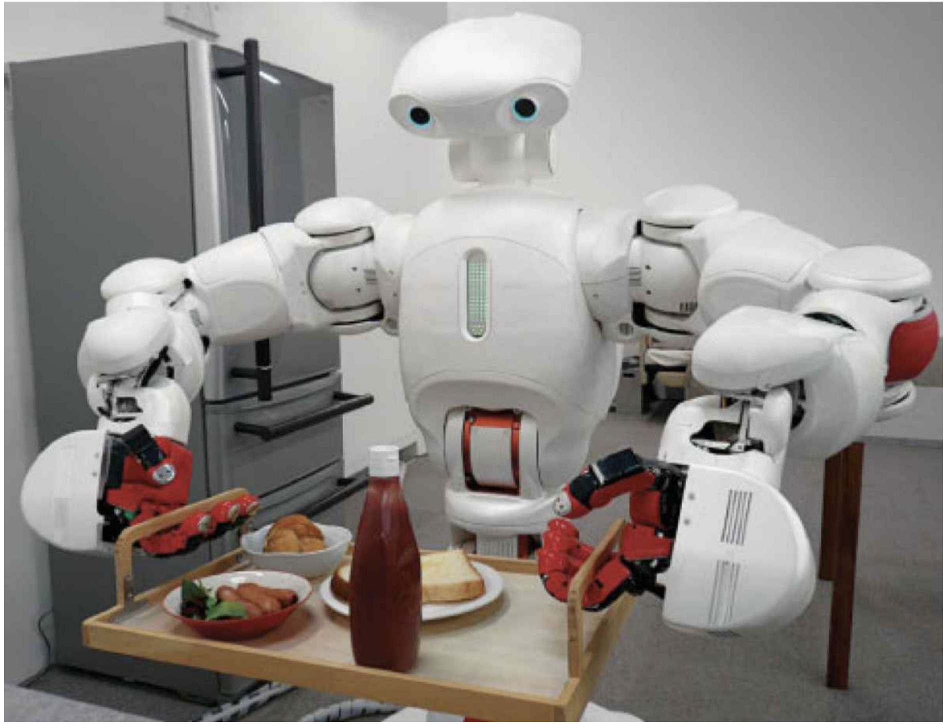
Environment

Actions

1. Richard Sutton and Andrew Barto. Reinforcement Learning: An Introduction. 2018.
2. Mnih et al. Playing Atari with Deep Reinforcement Learning. 2013.

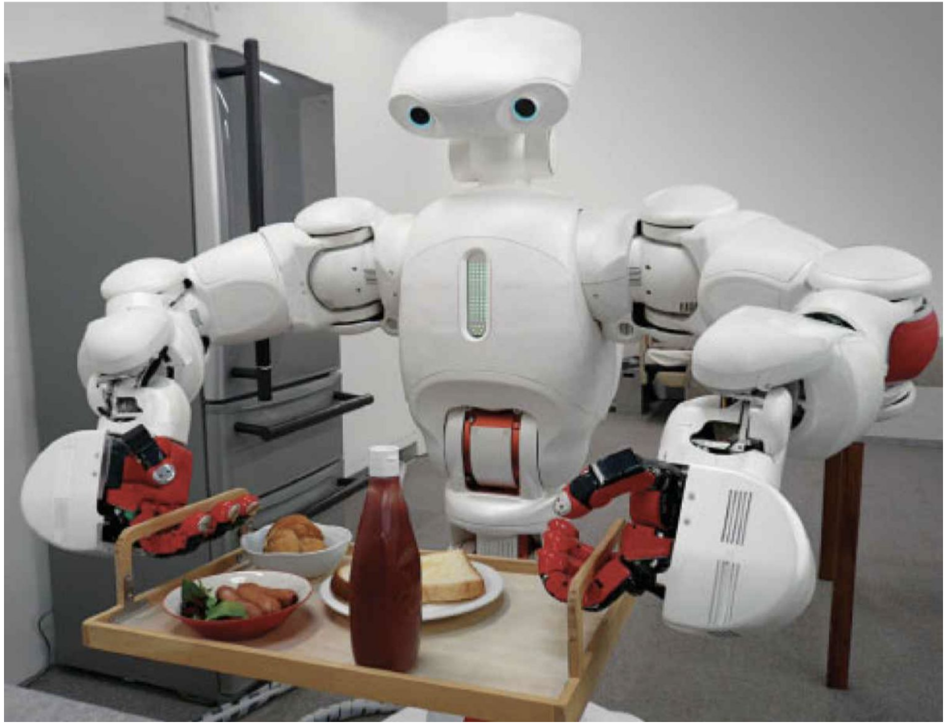
Background: Language and RL Tasks Share Compositional Structure

- "**Serve breakfast** with **plain toast** *and* **ketchup**…"



Background: Language and RL Tasks Share Compositional Structure

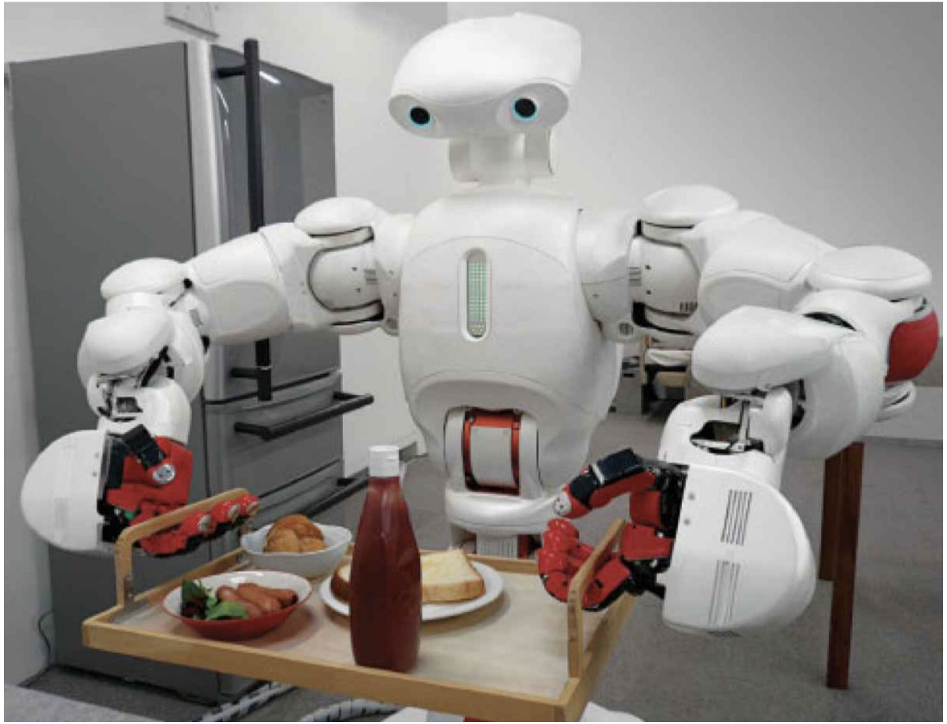
- "**Serve breakfast** with **plain toast** *and* **ketchup**…"
- Compose existing policies to perform tasks with minimal training.



1. Lake, B. M., & Baroni, M. (2018). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. Proceedings of the 35th International Conference on Machine Learning.

Background: Language and RL Tasks Share Compositional Structure

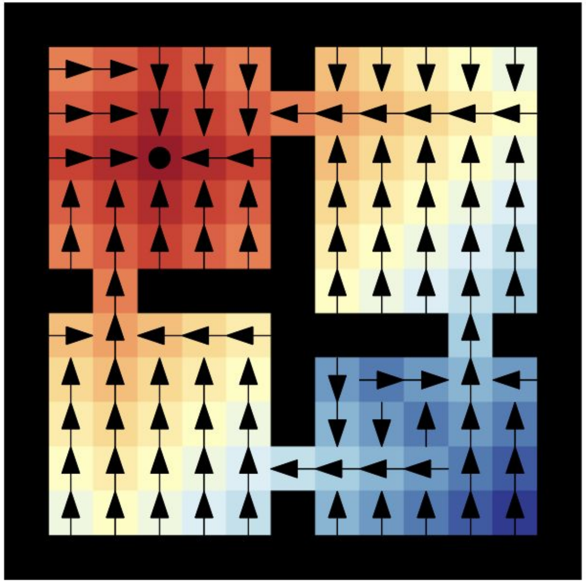
- "**Serve breakfast** with **plain toast** *and* **ketchup**…"
- Compose existing policies to perform tasks with minimal training.
- Neural networks **struggle to generalize compositionally**[1].



1. Lake, B. M., & Baroni, M. (2018). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. Proceedings of the 35th International Conference on Machine Learning.

State-Action Value Function

$$Q_\pi(s, a) = \mathbb{E}_s^\pi\left[\sum_{t=0}^{\infty} r(s_t, a_t)\right]$$

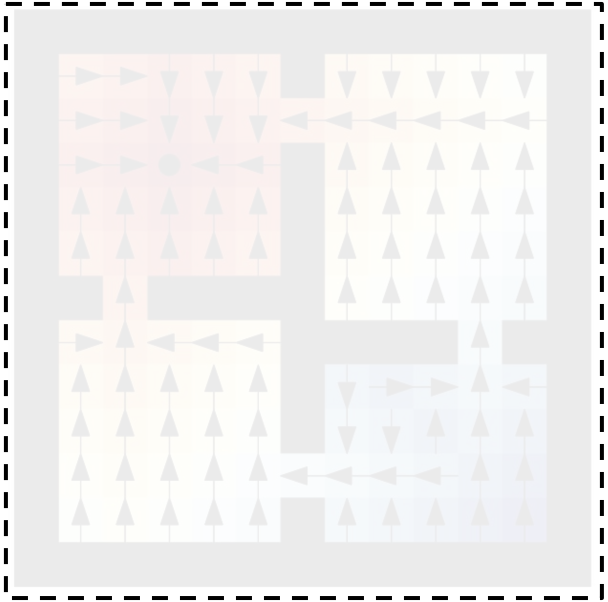


"Go to the top left room."

1. Nangue Tasse, G., James, S., & Rosman, B. (2020). A Boolean task algebra for reinforcement learning. Advances in Neural Information Processing Systems, 33, 17279–17290.
2. Nangue Tasse, G., James, S., & Rosman, B. (2022, June). World value functions: Knowledge representation for multitask reinforcement learning. Paper presented at the 5th Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM).

State-Action Value Function

$$Q_\pi(s,a) = \mathbb{E}_s^\pi\left[\sum_{t=0}^{\infty} r(s_t, a_t)\right]$$

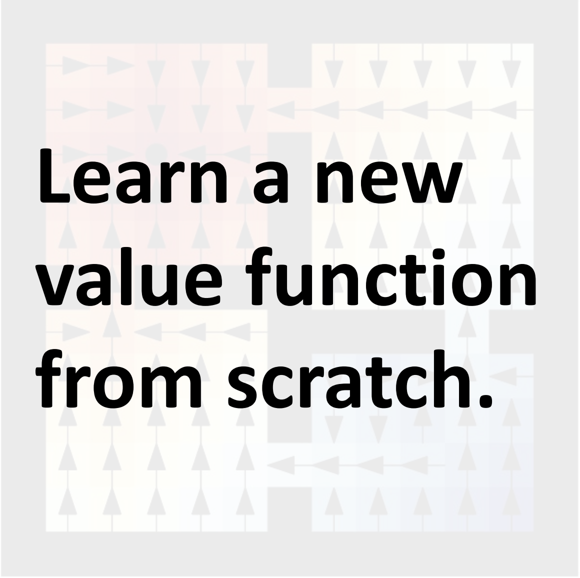


"Go to the top right room."

1. Nangue Tasse, G., James, S., & Rosman, B. (2020). A Boolean task algebra for reinforcement learning. Advances in Neural Information Processing Systems, 33, 17279–17290.
2. Nangue Tasse, G., James, S., & Rosman, B. (2022, June). World value functions: Knowledge representation for multitask reinforcement learning. Paper presented at the 5th Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM).

State-Action Value Function

$$Q_\pi(s, a) = \mathbb{E}_s^\pi\left[\sum_{t=0}^{\infty} r(s_t, a_t)\right]$$



"Go to the top right room."

1. Nangue Tasse, G., James, S., & Rosman, B. (2020). A Boolean task algebra for reinforcement learning. Advances in Neural Information Processing Systems, 33, 17279–17290.
2. Nangue Tasse, G., James, S., & Rosman, B. (2022, June). World value functions: Knowledge representation for multitask reinforcement learning. Paper presented at the 5th Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM).

World Value Functions (Tasse et al. 2020, 2022)

- Add a goal *g* to the Q function.
- WVF represents how to achieve all goals **and** their values.
- Learn one WVF for each task in the environment we wish to compose.

$$Q_\pi(s, g, a) = \mathbb{E}_s^\pi\left[\sum_{t=0}^{\infty} \bar{r}(s_t, g, a_t)\right]$$

1. Nangue Tasse, G., James, S., & Rosman, B. (2020). A Boolean task algebra for reinforcement learning. Advances in Neural Information Processing Systems, 33, 17279–17290.
2. Nangue Tasse, G., James, S., & Rosman, B. (2022, June). World value functions: Knowledge representation for multitask reinforcement learning. Paper presented at the 5th Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM).

World Value Functions (Tasse et al. 2020, 2022)

- Add a goal *g* to the Q function.
- WVF represents how to achieve all goals **and** their values.
- Learn one WVF for each task in the environment we wish to compose.
- Train by **penalizing** the agent for entering a terminal state for another goal.

$$Q_\pi(s, g, a) = \mathbb{E}_s^\pi\left[\sum_{t=0}^{\infty} \bar{r}(s_t, g, a_t)\right]$$

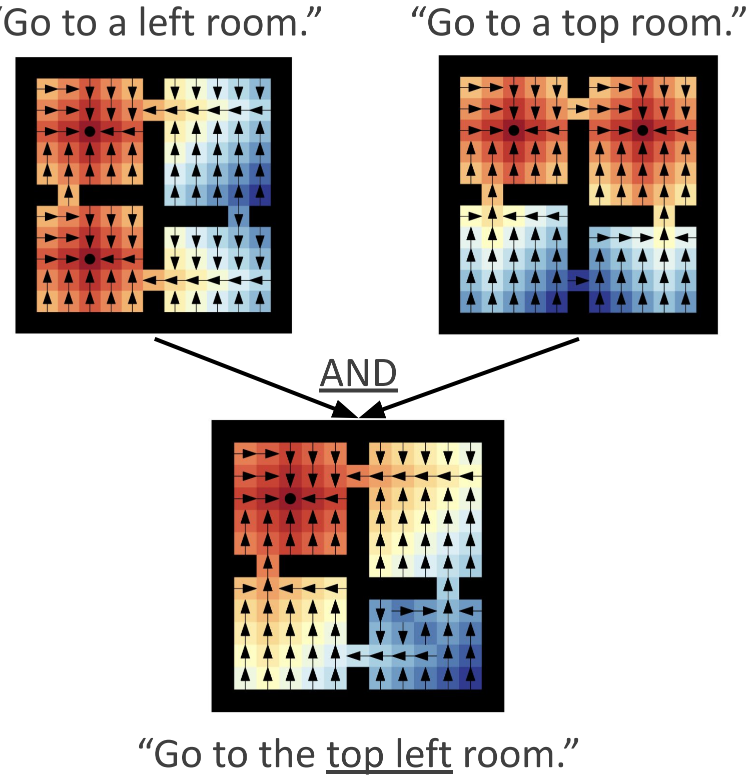
$$\bar{r}(s, g, a) = \begin{cases} \bar{r}_{MIN} & \text{if } g \neq s \in G \\ r(s, a) & \text{otherwise} \end{cases}$$

1. Nangue Tasse, G., James, S., & Rosman, B. (2020). A Boolean task algebra for reinforcement learning. Advances in Neural Information Processing Systems, 33, 17279–17290.
2. Nangue Tasse, G., James, S., & Rosman, B. (2022, June). World value functions: Knowledge representation for multitask reinforcement learning. Paper presented at the 5th Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM).

World Value Functions (Tasse et al. 2020, 2022)

For AND (conjunction) the composed WVF is given by:

$$\bar{Q}_1^* \wedge \bar{Q}_2^* : \mathcal{S} \times \mathcal{G} \times \mathcal{A} \to \mathbb{R}$$
$$(s, g, a) \mapsto \min\{\bar{Q}_1^*(s, g, a), \bar{Q}_2^*(s, g, a)\}$$

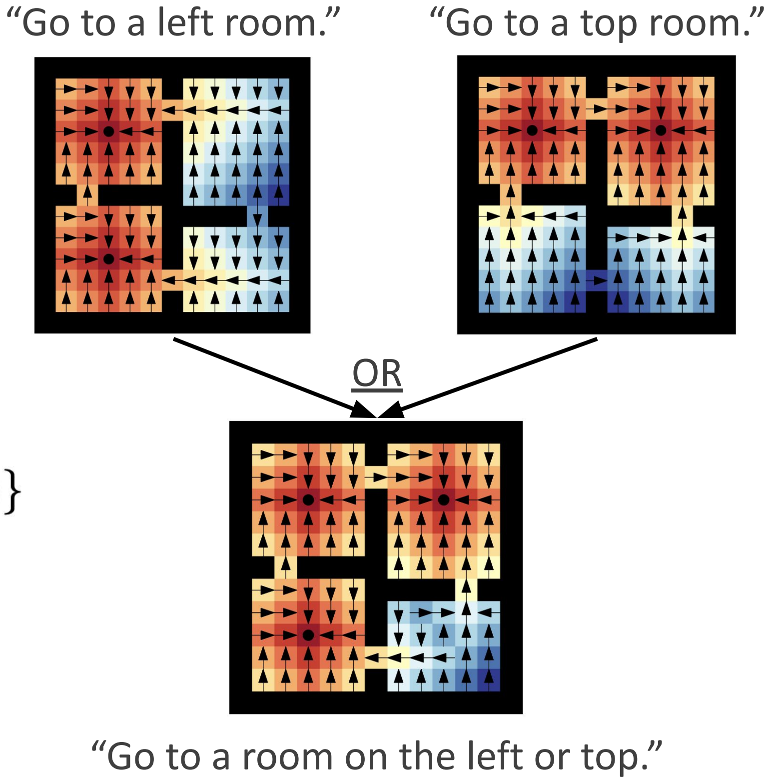


1. Nangue Tasse, G., James, S., & Rosman, B. (2020). A Boolean task algebra for reinforcement learning. Advances in Neural Information Processing Systems, 33, 17279–17290.
2. Nangue Tasse, G., James, S., & Rosman, B. (2022, June). World value functions: Knowledge representation for multitask reinforcement learning. Paper presented at the 5th Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM).

World Value Functions (Tasse et al. 2020, 2022)

For OR (disjunction) the composed WVF is given by:

$$\bar{Q}_1^* \vee \bar{Q}_2^* : \mathcal{S} \times \mathcal{G} \times \mathcal{A} \to \mathbb{R}$$
$$(s, g, a) \mapsto \max\{\bar{Q}_1^*(s, g, a), \bar{Q}_2^*(s, g, a)\}$$



1. Nangue Tasse, G., James, S., & Rosman, B. (2020). A Boolean task algebra for reinforcement learning. Advances in Neural Information Processing Systems, 33, 17279–17290.
2. Nangue Tasse, G., James, S., & Rosman, B. (2022, June). World value functions: Knowledge representation for multitask reinforcement learning. Paper presented at the 5th Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM).

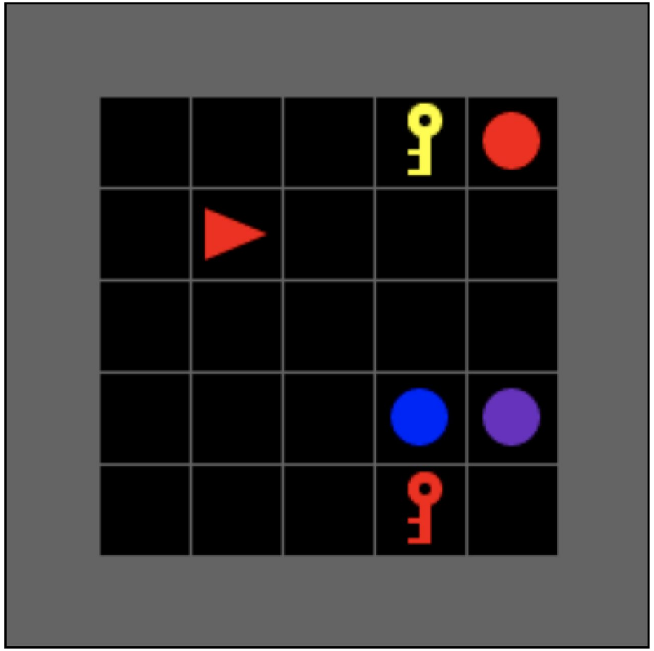
World Value Functions (WVF) (Tasse et al. 2020, 2022)

Composing WVFs

- Arbitrary expressions of **AND**, **OR**, and **NOT**.
- Can now solve a **combinatorial number** of goal reaching tasks

BabyAI (Chevalier-Boisvert et al. 2019)

- Gridworld domain consisting of language instruction tasks.

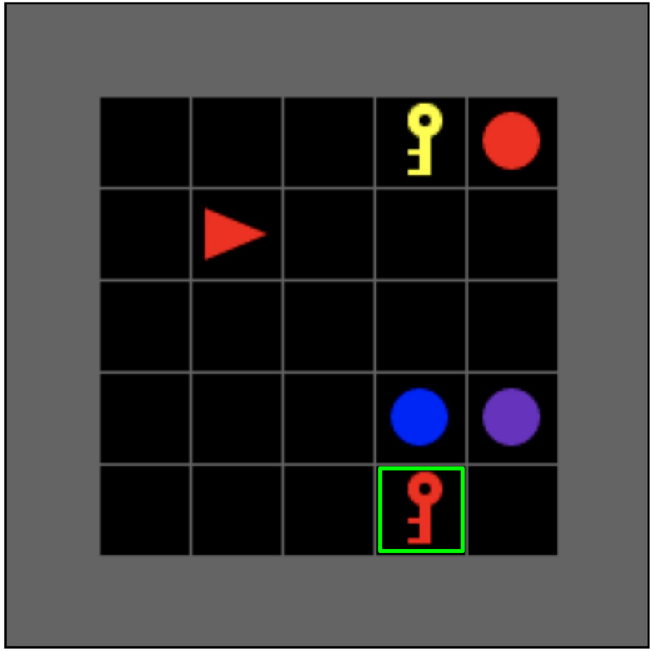


BabyAI (Chevalier-Boisvert et al. 2019)

- Gridworld domain consisting of language instruction tasks.

“Pick up a red object” $\wedge$ “Pick up a key”

$\neg$“Pick up a blue object” $\vee$ “Pick up a ball”



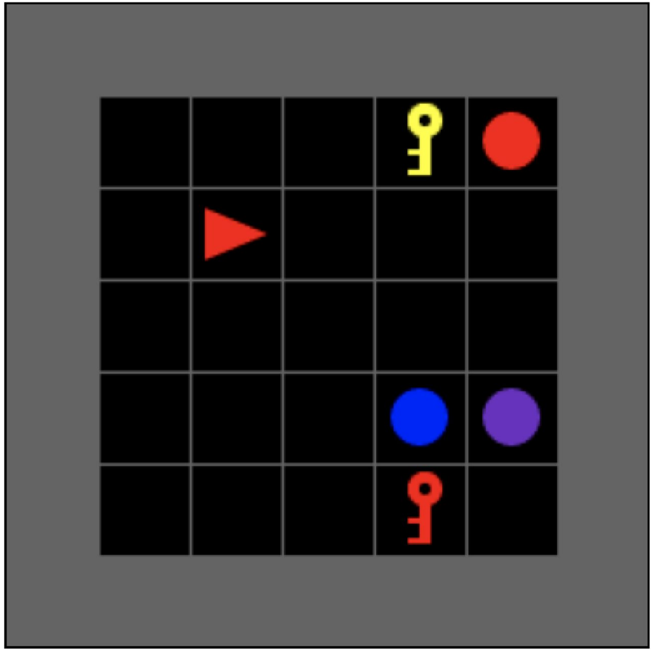
BabyAI (Chevalier-Boisvert et al. 2019)

- Gridworld domain consisting of language instruction tasks.

"Pick up a red object" ∧ "Pick up a key"

¬"Pick up a blue object" ∨ "Pick up a ball"

- Modified task set to include 162 goal reaching tasks that can be solved through **AND**, **OR**, and **NOT** expressions over object attributes.



Related Work

- Prior work in compositional policies either does not include language, or assumes access to demonstrations or separately trained parsers.
- Few-shot semantic parsing from environment feedback.

Mendez et al. CompoSuite: A compositional reinforcement learning benchmark. CoLLAs 2022.
Mendez and Eaton. How to reuse and compose knowledge for a lifetime of tasks: A survey on continual learning and functional composition. TMLR 2023.
Kuo et al. Compositional RL agents that follow language commands in temporal logic. Frontiers in Robotics and AI 2021.
Schick et al. Toolformer: Language models can teach themselves to use tools. NeurIPS 2023.

Compositionally-Enabled RL and Language Agent (CERLLA)

- How can we use compositionality of language + value functions to generalize better?

Compositionally-Enabled RL and Language Agent (CERLLA)

- How can we use compositionality of language + value functions to generalize better?
- Must learn mapping from natural language to WVF composition.

Compositionally-Enabled RL and Language Agent (CERLLA)

- How can we use compositionality of language + value functions to generalize better?
- Must learn mapping from natural language to WVF composition.
- Idea: Use language models to translate instructions into formal language / boolean symbols (e.g. semantic parsing).

Compositionally-Enabled RL and Language Agent (CERLLA)

- But these symbols are arbitrary (just an index over WVFs) - how do we know translation is correct?

Compositionally-Enabled RL and Language Agent (CERLLA)

- But these symbols are arbitrary (just an index over WVFs) - how do we know translation is correct?
- Idea: Use environment feedback to learn the translation!

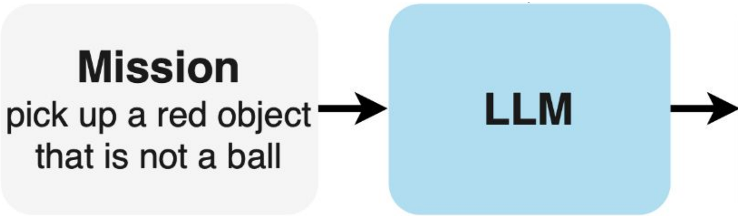
Compositionally-Enabled RL and Language Agent (CERLLA)

Core challenge: CERLLA learns to parse input commands to **arbitrary symbols** representing WVFs with **unknown semantics**, using **environment rollouts**, a much noisier form of supervision than is typical for weakly supervised parsing methods.

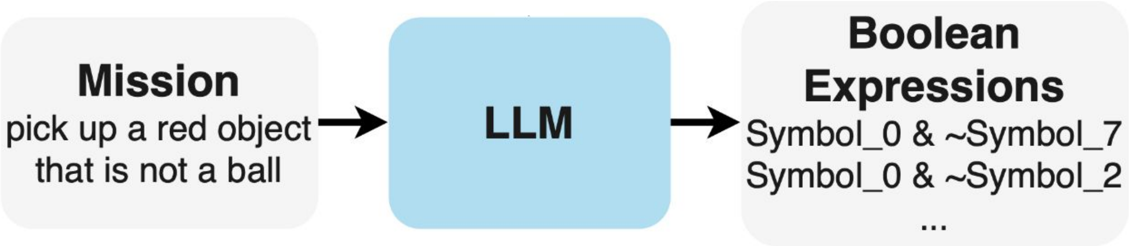
Compositionally-Enabled RL and Language Agent (CERLLA)

"Pick up the red ball" → WVF_1 **AND** WVF_8 → Reward: Agent picked up a red ball?

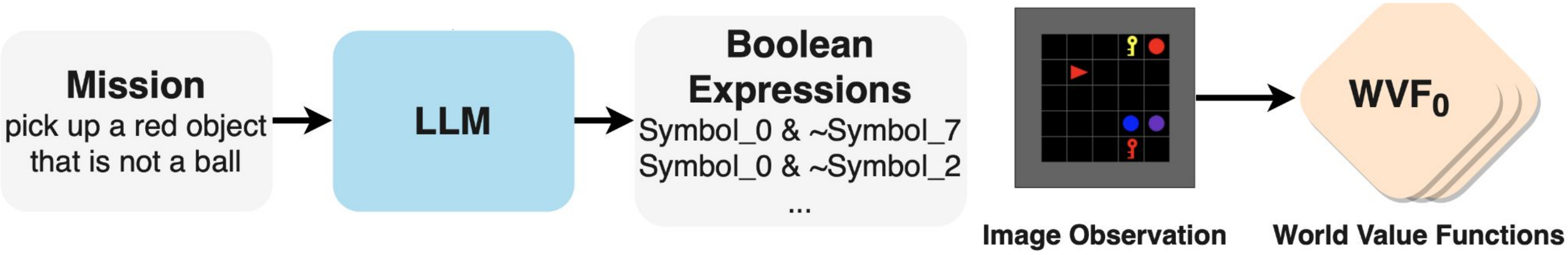
Compositionally-Enabled RL and Language Agent (CERLLA)



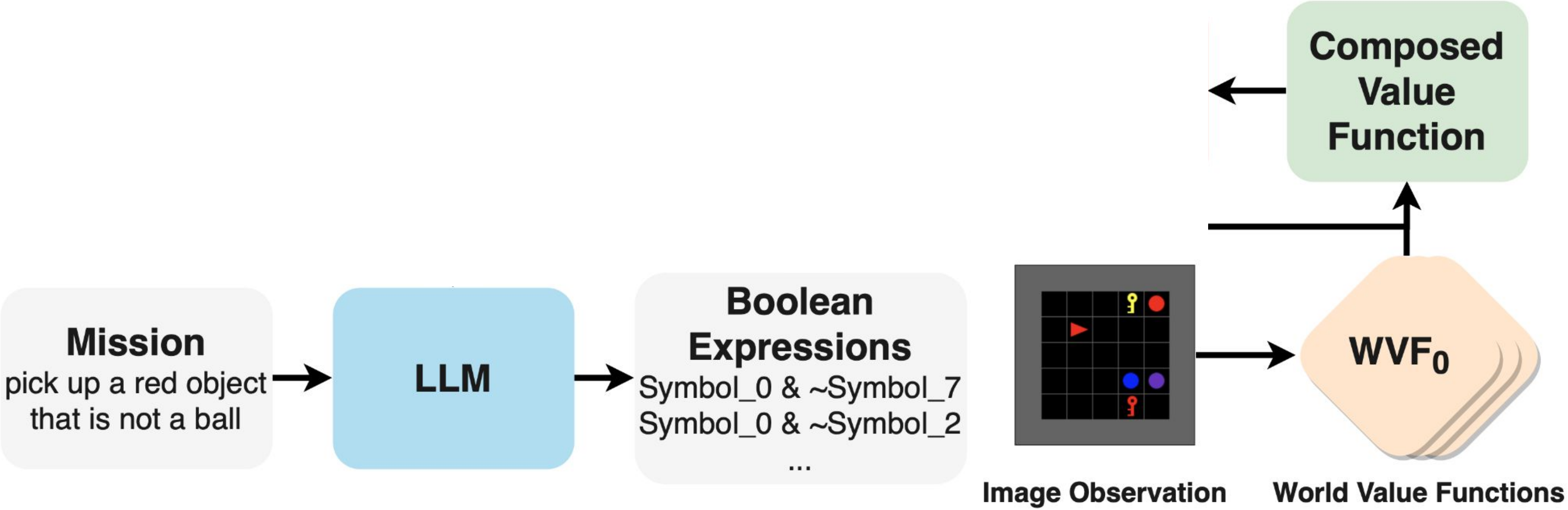
Compositionally-Enabled RL and Language Agent (CERLLA)



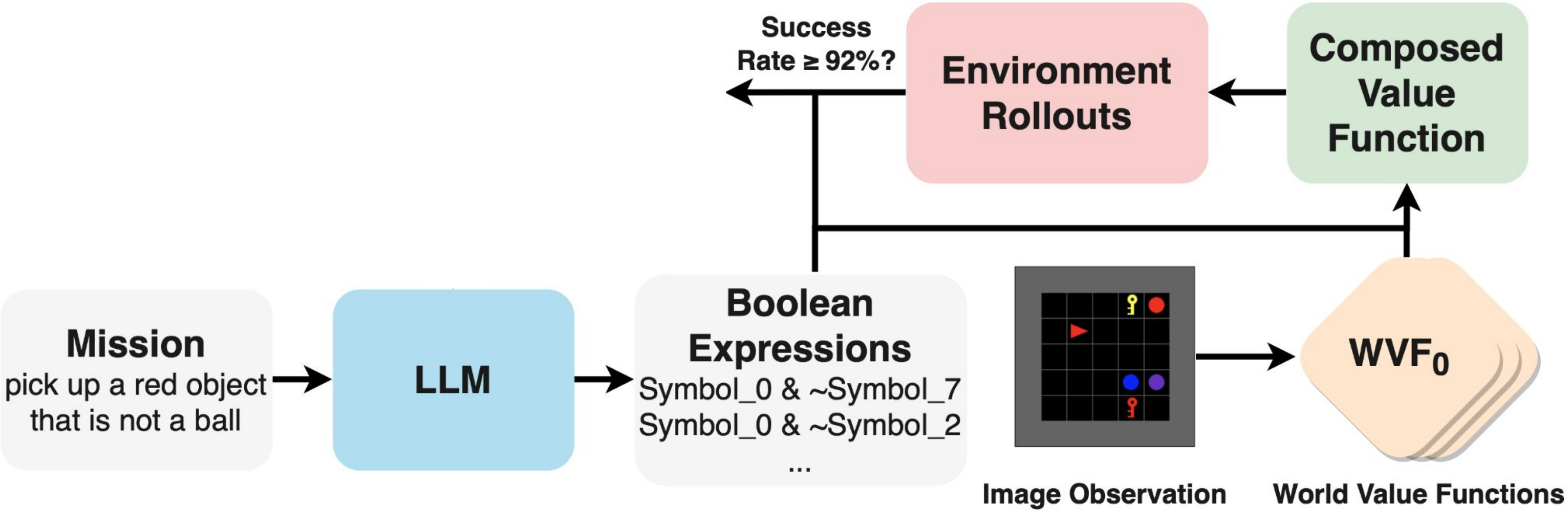
Compositionally-Enabled RL and Language Agent (CERLLA)



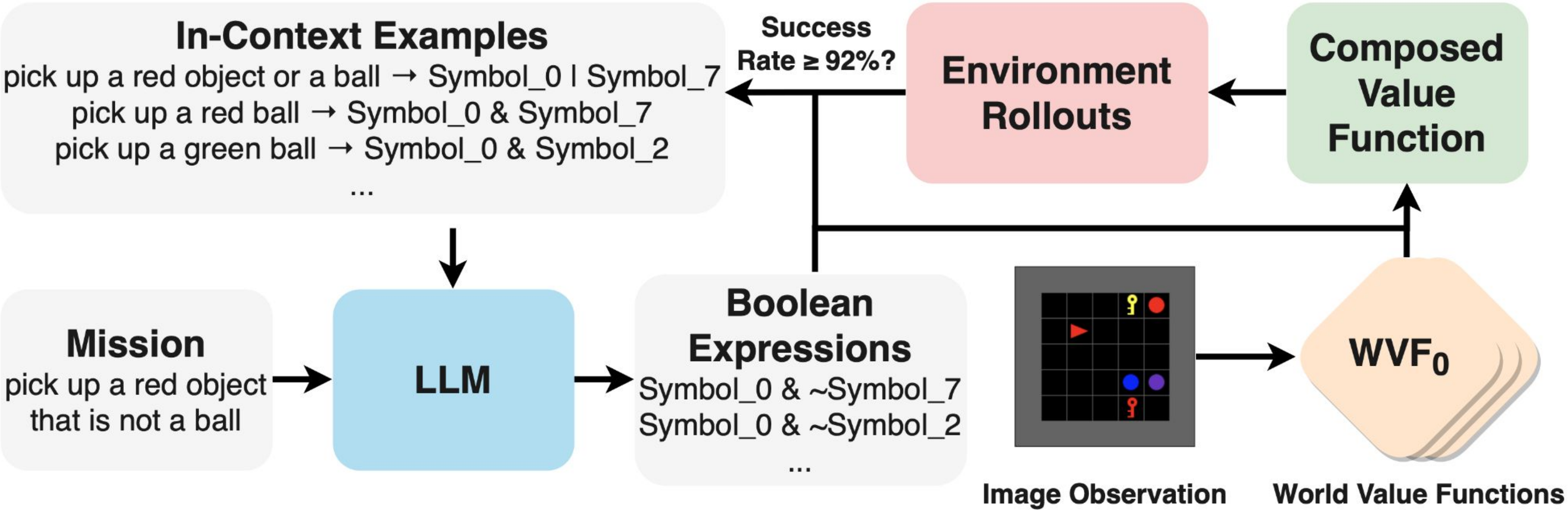
Compositionally-Enabled RL and Language Agent (CERLLA)



Compositionally-Enabled RL and Language Agent (CERLLA)



Compositionally-Enabled RL and Language Agent (CERLLA)



Experiments

- 162 tasks, learned simultaneously from vision and language.

Experiments

- 162 tasks, learned simultaneously from vision and language.
- Evaluate **sample efficiency**, and **generalization**, comparing:

Experiments

- 162 tasks, learned simultaneously from vision and language.
- Evaluate **sample efficiency**, and **generalization**, comparing:
  - CERLLA (Ours): using OpenAI's GPT-4 LM
    - CERLLA GPT-3.5

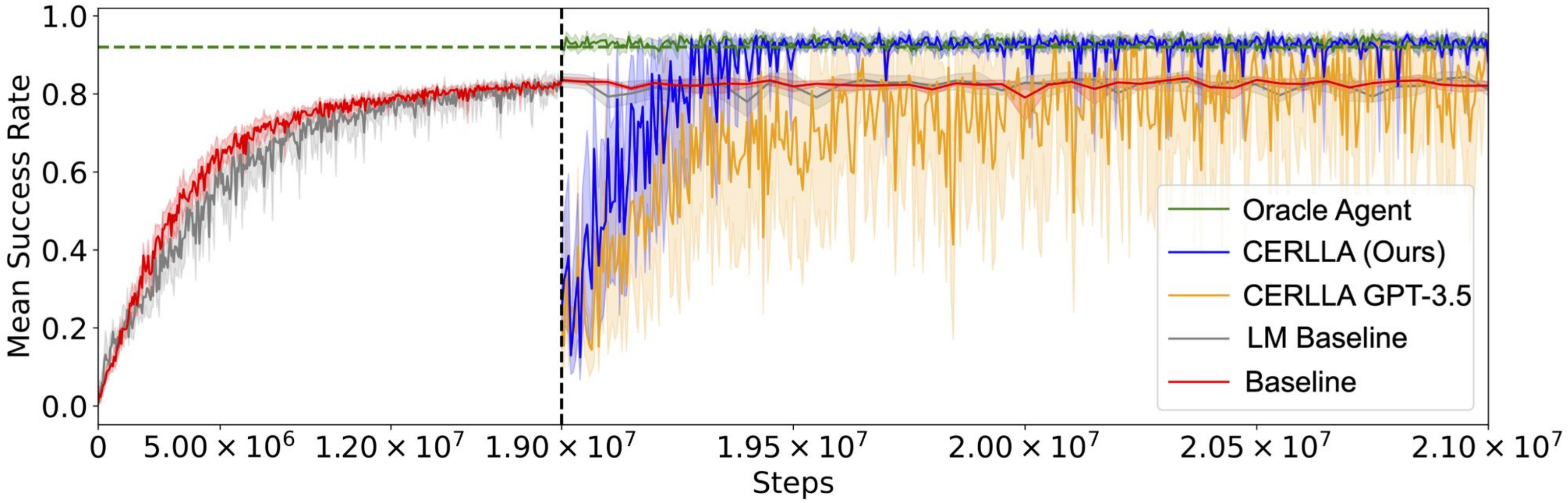
Experiments

- 162 tasks, learned simultaneously from vision and language.
- Evaluate **sample efficiency**, and **generalization**, comparing:
  - CERLLA (Ours): using OpenAI's GPT-4 LM
    - CERLLA GPT-3.5
  - Two non-compositional baseline DQNs
    - Baseline: RNN + CNN
    - LM Baseline: pretrained sentence embedding language model + CNN

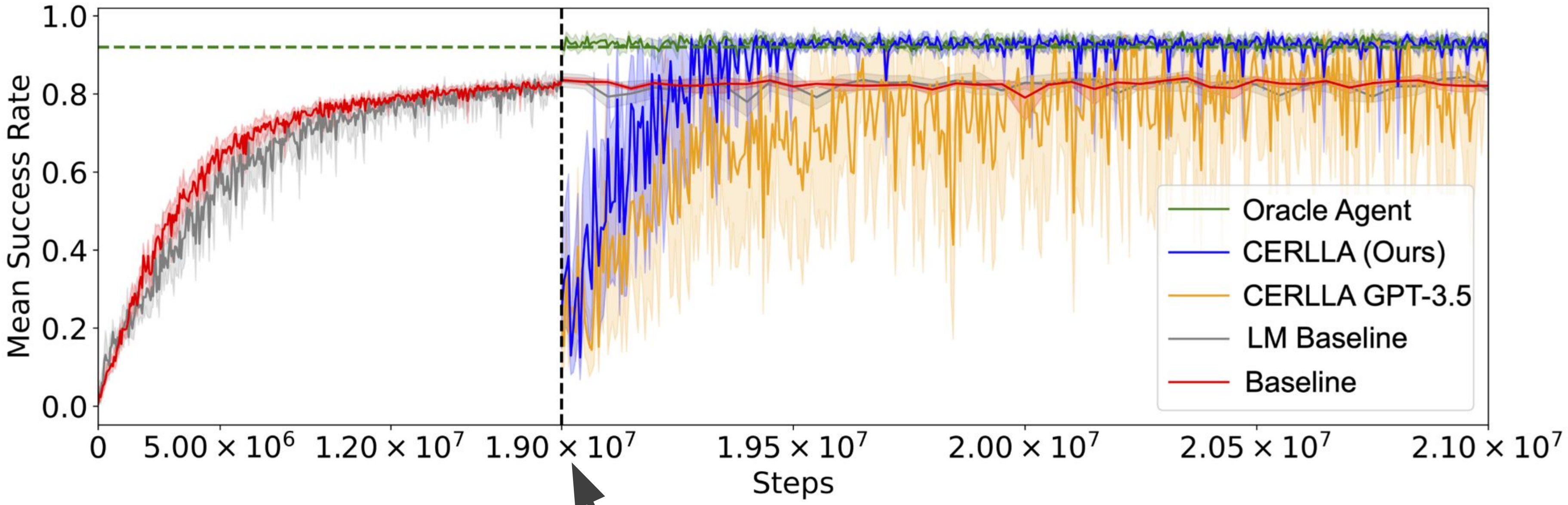
Experiments

- 162 tasks, learned simultaneously from vision and language.
- Evaluate **sample efficiency**, and **generalization**, comparing:
  - CERLLA (Ours): using OpenAI's GPT-4 LM
    - CERLLA GPT-3.5
  - Two non-compositional baseline DQNs
    - Baseline: RNN + CNN
    - LM Baseline: pretrained sentence embedding language model + CNN
  - Oracle Agent with access to the ground-truth compositional expressions for each task.

Sample Efficiency

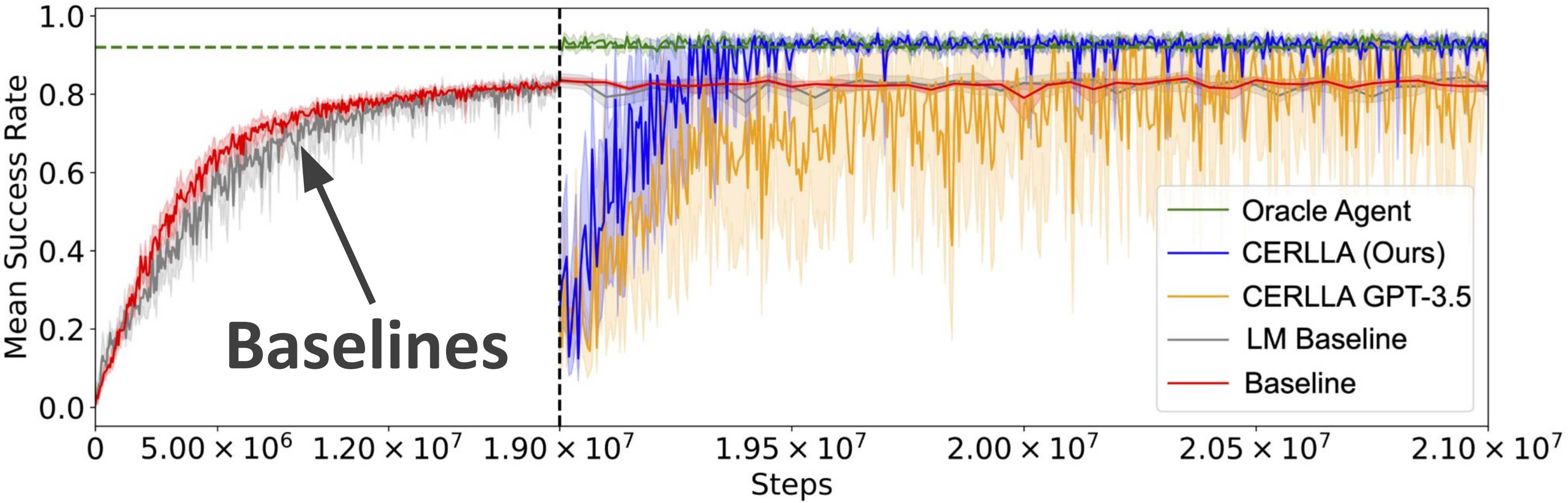


Sample Efficiency

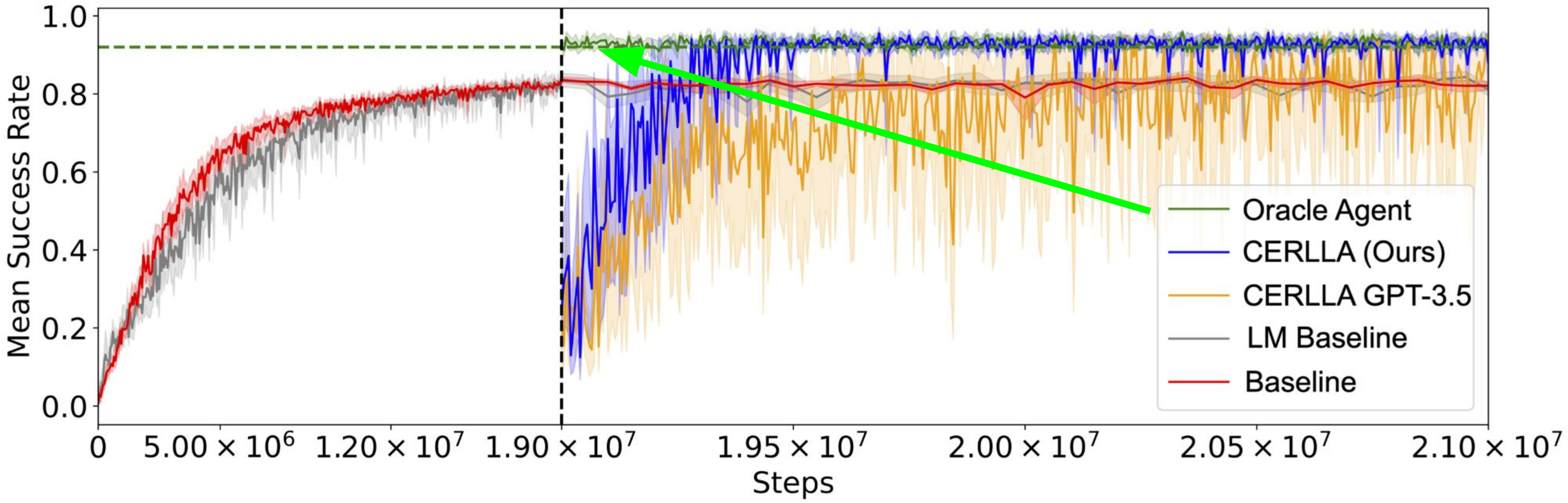


Equalize environment steps

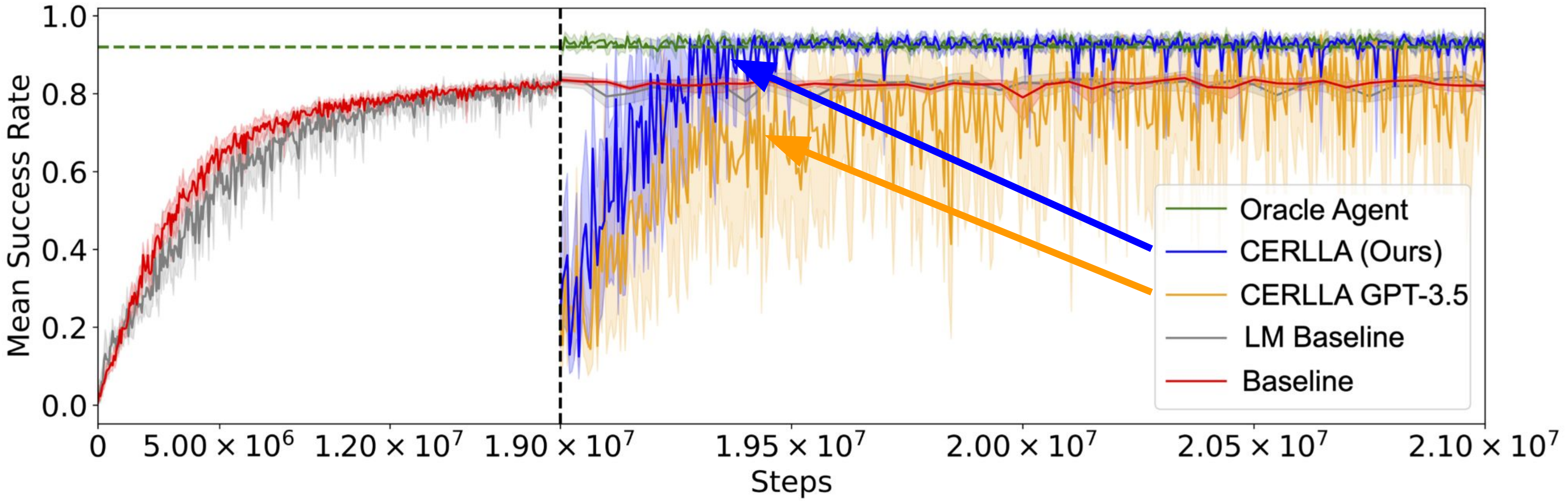
Sample Efficiency



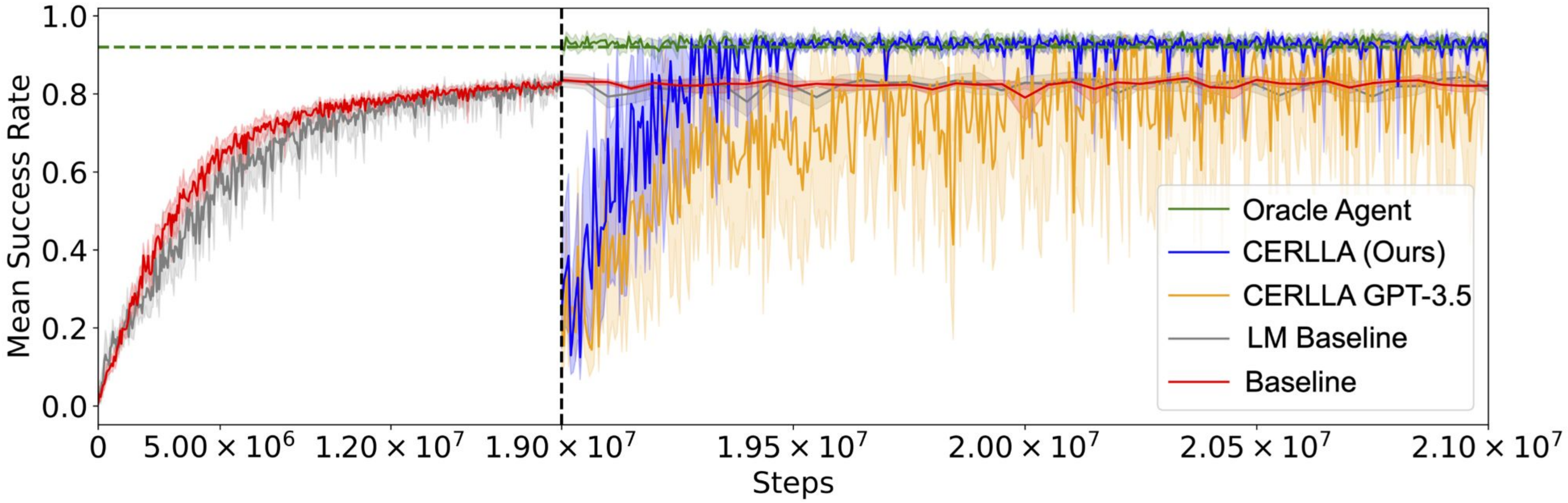
Sample Efficiency



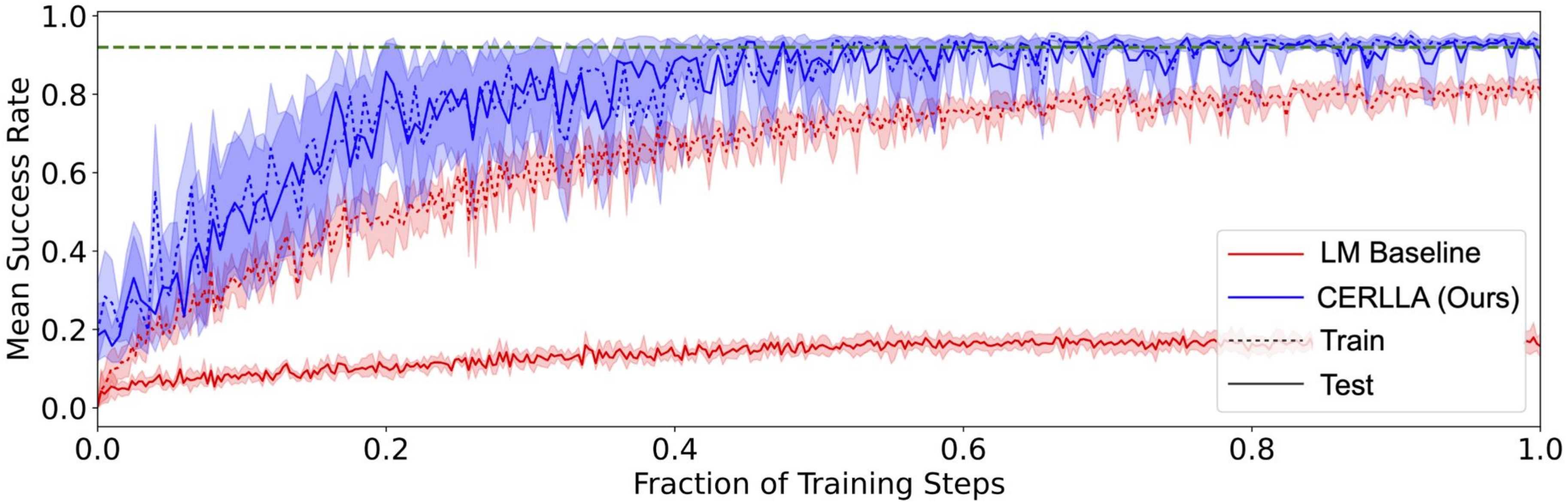
Sample Efficiency



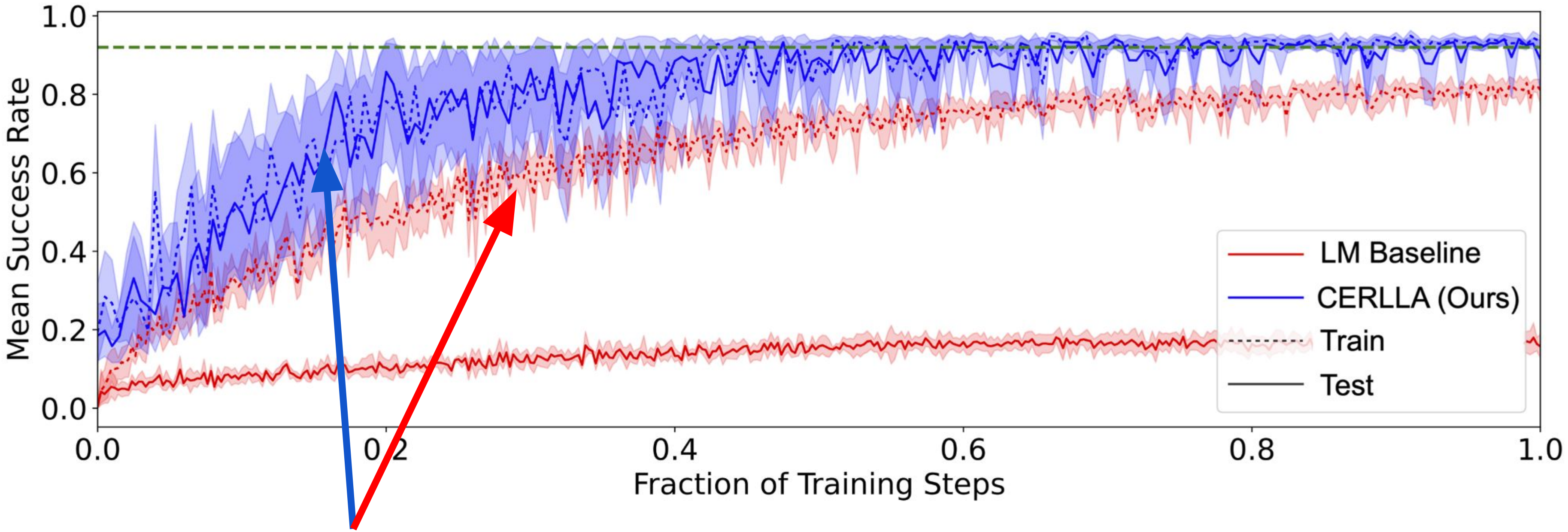
Sample Efficiency



Generalization

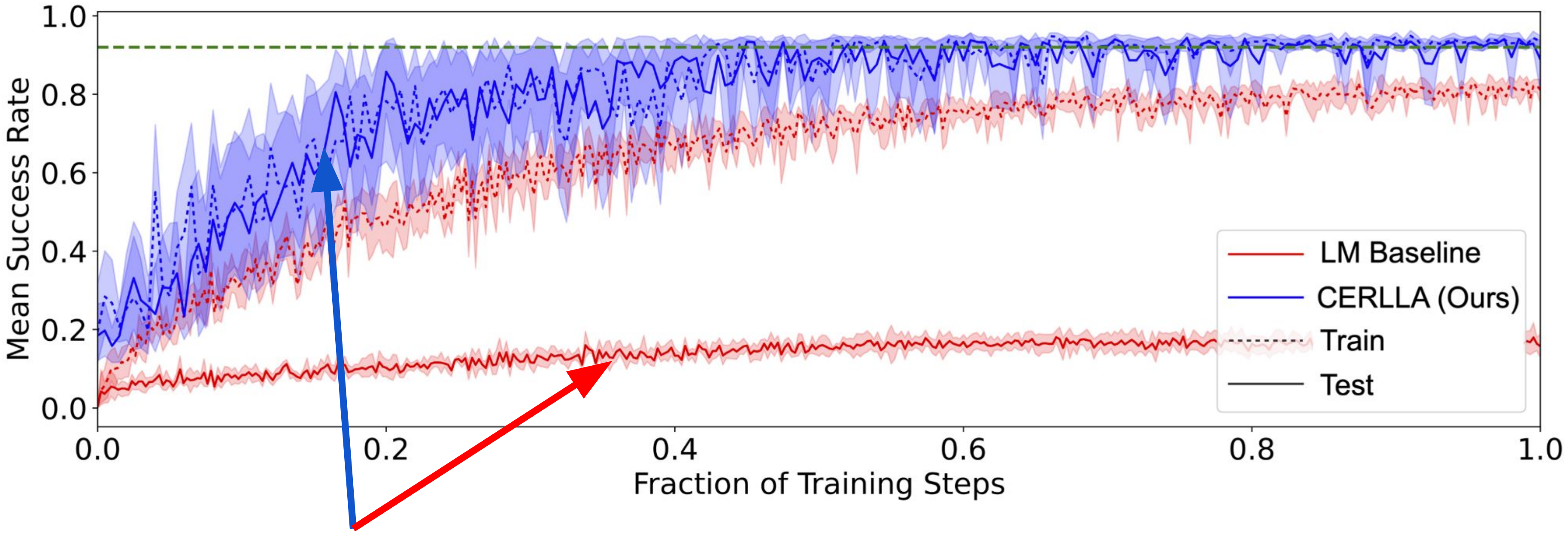


Generalization



Train Success Rates

Generalization



Test Success Rates

Conclusion

- Introduces CERLLA, a **novel semantic parsing** method based on **in-context learning** that learns from **environment feedback**.

Conclusion

- Simultaneously learns and solves a large collection of **162 compositional vision-language-RL tasks**.

Conclusion

- Outperforms non-compositional baselines with respect to sample efficiency and generalization to held-out tasks.

Improvement

Planning-Augmented Parsing *ACL NLRSE 2023*

Using Planning to Improve Semantic Parsing of Instructional Texts

Planning constraints rank/correct candidate action sequences.

Compositional Instruction Following *TMLR 2024 / RLC 2025*

Compositional Instruction Following with LMs and RL

Learns language-to-policy composition from environment feedback.

Evaluation

CaT-Bench *EMNLP 2024*

Causal and Temporal Dependencies in Plans

Tests whether LMs understand preconditions, step ordering, and dependence.

MET-Bench *ICML 2026*

Multimodal Entity State Tracking

Tests postconditions and entity state tracking across text and images.

Improvement	Evaluation
Planning-Augmented Parsing *ACL NLRSE 2023* Using Planning to Improve Semantic Parsing of Instructional Texts Planning constraints rank/correct candidate action sequences.	**CaT-Bench** *EMNLP 2024* Causal and Temporal Dependencies in Plans Tests whether LMs understand preconditions, step ordering, and dependence.
Compositional Instruction Following *TMLR 2024 / RLC 2025* Compositional Instruction Following with LMs and RL Learns language-to-policy composition from environment feedback.	**MET-Bench** *ICML 2026* Multimodal Entity State Tracking Tests postconditions and entity state tracking across text and images.

Motivation: Evaluation

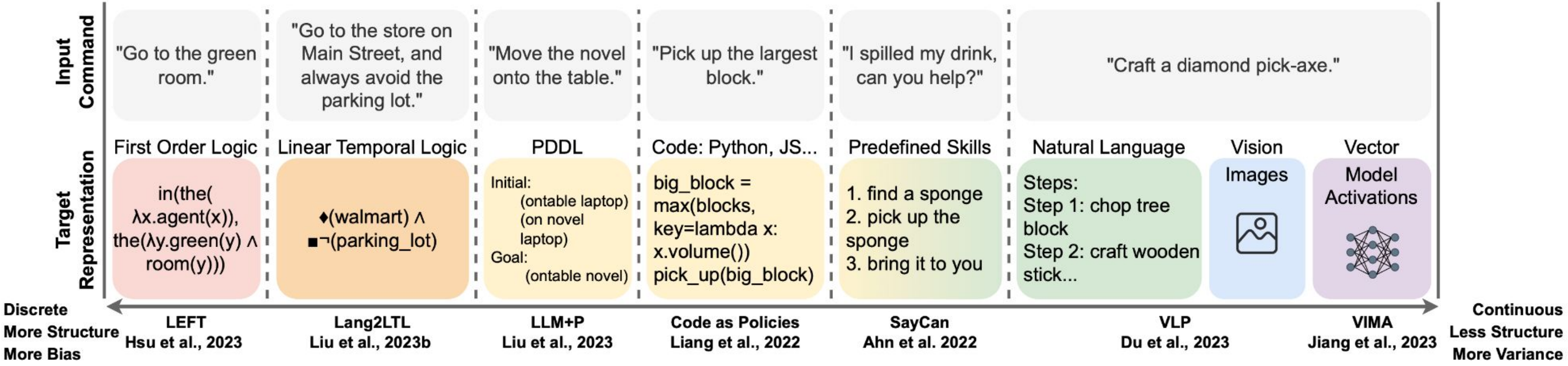
- The previous works improve action reasoning using external planning domains and a curriculum, but constructing these is costly and domain-specific.

Motivation: Evaluation

- Practical agents may instead rely on the pretrained model’s internal knowledge.

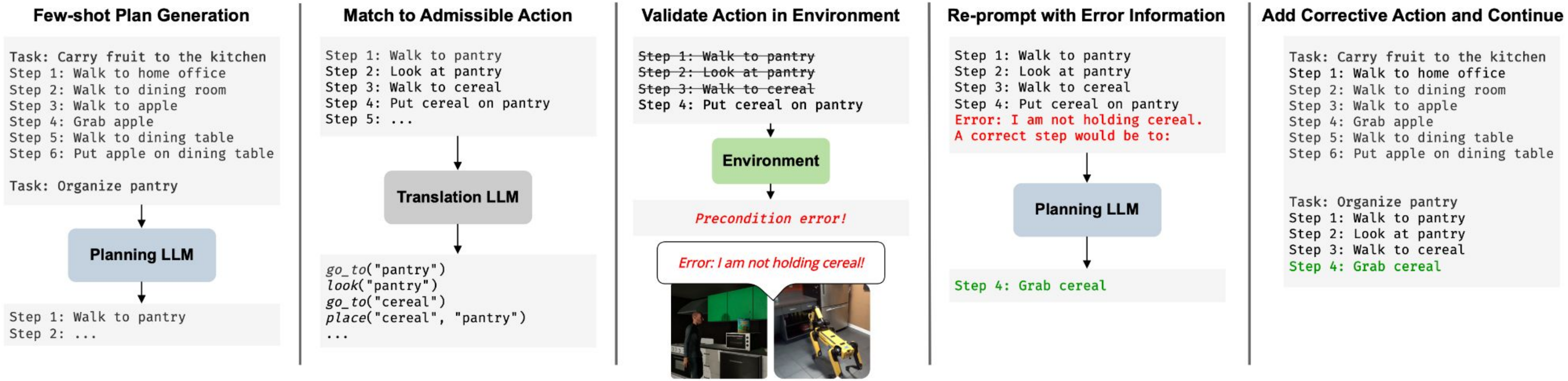
A Survey of Robotic Language Grounding: Tradeoffs Between Symbols and Embeddings (IJCAI 2024: Survey Track)

Vanya Cohen*, Jason Xinyu Liu*, Raymond Mooney*, Stefanie Tellex*, David Watkins*



CAPE: Corrective Actions from Precondition Errors using Large Language Models (ICRA 2024)

Shreyas Sundara Raman, **Vanya Cohen**, Ifrah Idrees, Eric Rosen, Ray Mooney, Stefanie Tellex, David Paulius.



Improvement	Evaluation
Planning-Augmented Parsing *ACL NLRSE 2023* Using Planning to Improve Semantic Parsing of Instructional Texts Planning constraints rank/correct candidate action sequences.	**CaT-Bench** *EMNLP 2024* Causal and Temporal Dependencies in Plans Tests whether LMs understand preconditions, step ordering, and dependence.
Compositional Instruction Following *TMLR 2024 / RLC 2025* Compositional Instruction Following with LMs and RL Learns language-to-policy composition from environment feedback.	**MET-Bench** *ICML 2026* Multimodal Entity State Tracking Tests postconditions and entity state tracking across text and images.

CaT-Bench: Benchmarking Language Model Understanding of Causal and Temporal Dependencies in Plans

Yash Kumar Lal*, **Vanya Cohen***, Nathanael Chambers, Niranjan Balasubramanian, Raymond Mooney

LLMs Generate Fluent Plans

plan a day and a half trip to yellowstone national park from salt lake city in early may

Day 1: Leave Salt Lake City early in the morning to make the most of your time at Yellowstone National Park. The drive is approximately 320 miles (515 km) and takes about 5-6 hours, depending on traffic and road conditions.

Here's a suggested itinerary for your day and a half trip:

Morning:

- Depart from Salt Lake City around 6:00 AM.
- Drive through scenic Idaho, passing by Craters of the Moon National Monument and Preserve if time allows.

But can they reason about preconditions in plans?

Plans Contain Dependencies

Goal: Bake Almond and Chocolate Cake
… Step 6: Add in ground almonds. Step 7: Add half flour and half milk. Step 8: Use wooden spoon to stir. … Step 12: Whip cream till stiff peaks …

Plans Contain Dependencies

Goal: Bake Almond and Chocolate Cake

…
Step 6: Add in ground almonds.
Step 7: Add half flour and half milk.
Step 8: Use wooden spoon to stir.
…
Step 12: Whip cream till stiff peaks
…

Dependent Steps

Goal: Bake Almond and Chocolate Cake

…
Step 6: Add in ground almonds.
Step 7: Add half flour and half milk.
Step 8: Use wooden spoon to stir.
…
Step 12: Whip cream till stiff peaks
…

Q: Must Step 6 happen before Step 8?

Preconditions

A: Yes, all ingredients have to be in the bowl before stirring

Questions about dependent steps

Goal: Bake Almond and Chocolate Cake

…
Step 6: Add in ground almonds.
Step 7: Add half flour and half milk.
Step 8: Use wooden spoon to stir.
…
Step 12: Whip cream till stiff peaks
…

Q: Must Step 6 happen before Step 8?

Preconditions

A: Yes, all ingredients have to be in the bowl before stirring

Questions about dependent steps

Q: Must Step 7 happen after Step 6?

Parallel Steps

A: No, almonds can be added after flour and milk

Questions about non-dependent steps

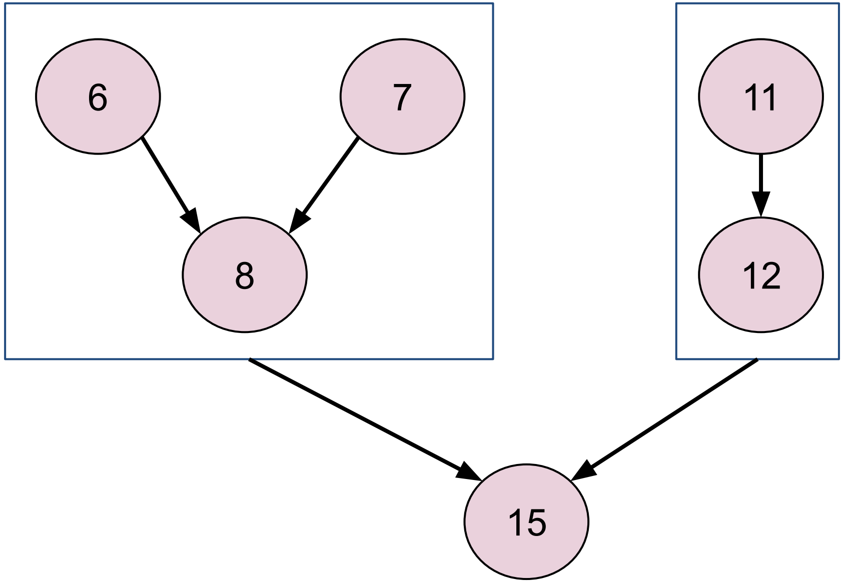
Creating 🐱 CaT-Bench

Goal: Bake Almond and Chocolate Cake
… Step 6: Add in ground almonds. Step 7: Add half flour and half milk. Step 8: Use wooden spoon to stir. … Step 12: Whip cream till stiff peaks …

Recipe Flow Graph Corpus (Yamakata et al. 2020)

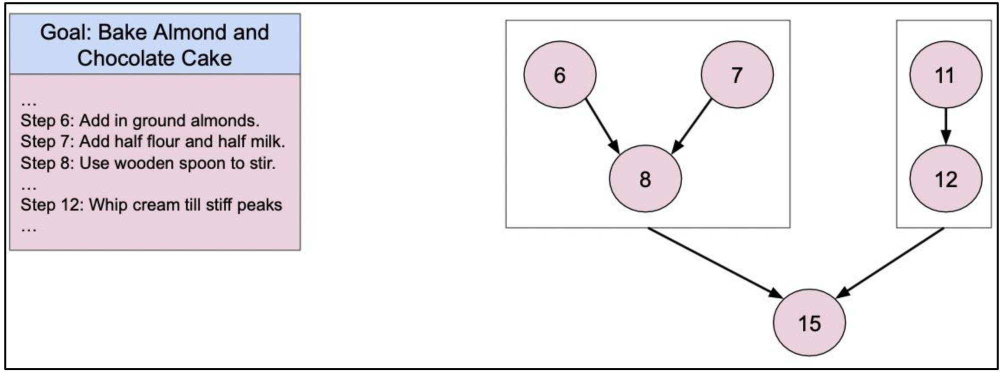
Creating 🐱 CaT-Bench

Goal: Bake Almond and Chocolate Cake
… Step 6: Add in ground almonds. Step 7: Add half flour and half milk. Step 8: Use wooden spoon to stir. … Step 12: Whip cream till stiff peaks …



Recipe Flow Graph Corpus (Yamakata et al. 2020)

Creating 🐱 CaT-Bench



710 Q: Must Step 6 happen before Step 8?

710 Q: Must Step 8 happen after Step 6?

Questions about dependent steps

Q: Must Step 12 happen after Step 6? 710

Q: Must Step 6 happen before Step 12? 710

Questions about non-dependent steps 1420

Dataset Size: 1420

Related Work

- Prior works focus on understanding plans, including step relationships (XPAD: Dalvi et al. 2019; MM-ReS: Pan et al. 2020; Wu et al. 2024)

Dalvi et al. Everything Happens for a Reason: Discovering the Purpose of Actions in Procedural Text. EMNLP 2019.
Pan et al. Multi-modal cooking workflow construction for food recipes. ACM International Conference on Multimedia 2020.
Wu et al. Understanding multimodal procedural knowledge by sequencing multimodal instructional manuals. 2024.

Related Work

- CaT-Bench evaluates LLMs through the prediction and explanation of precondition relationships in instructional texts.

Benchmarking Models

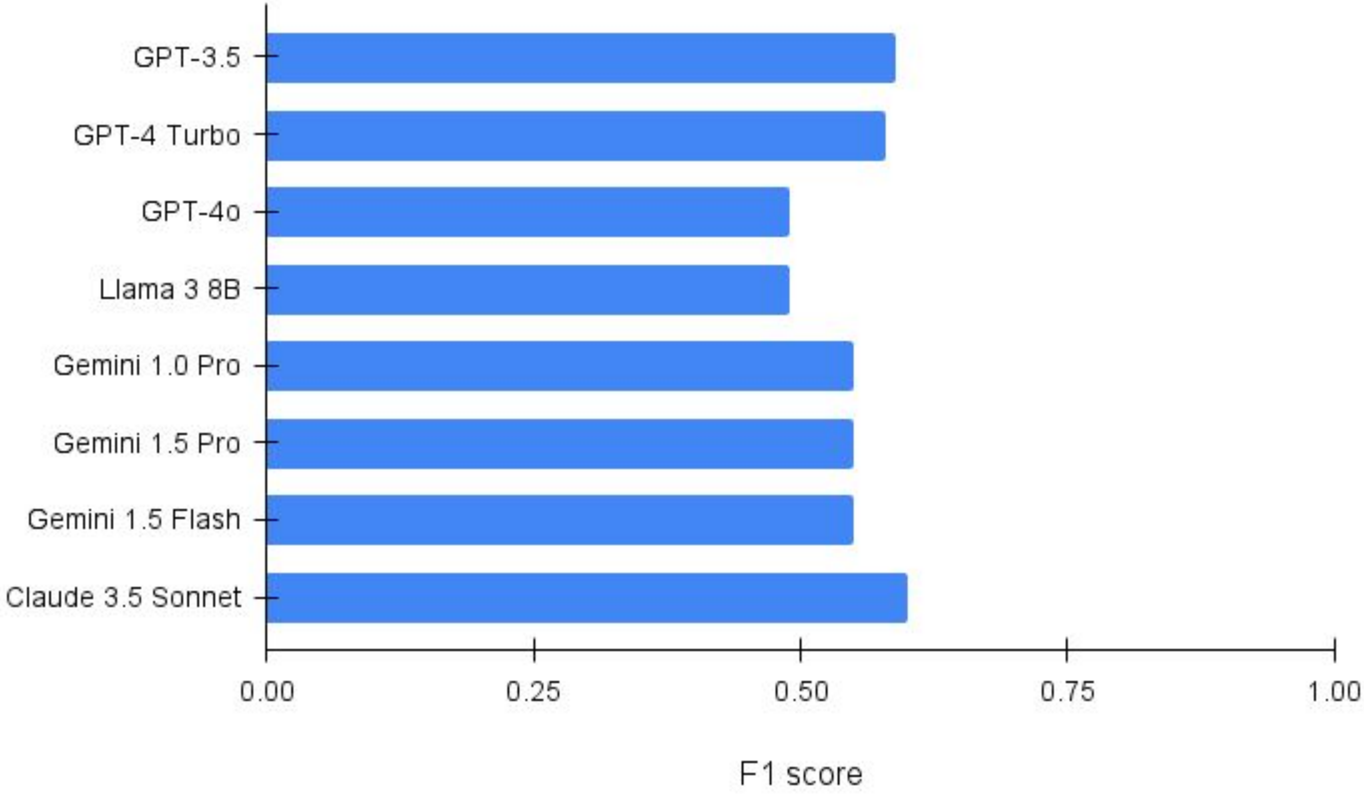
- GPT-3.5
- GPT-4-Turbo
- GPT-4o
- Llama-3-8B
- Claude-3.5-Sonnet
- Gemini-1.0-Pro
- Gemini-1.5-Pro
- Gemini-1.5-Flash

Experiments - Answer Only (A)

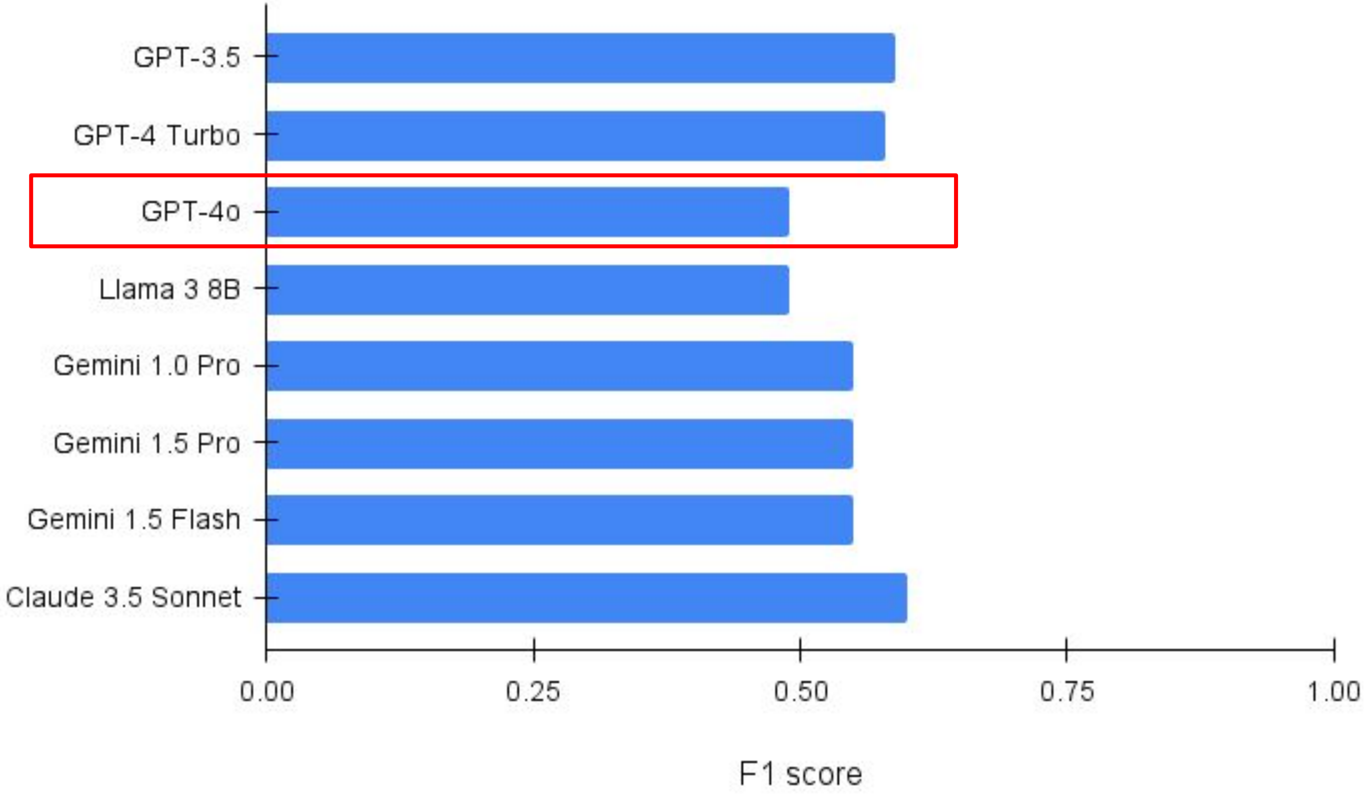
- Binary Prediction

Answer-only (A)	Given a goal, a procedure to achieve that goal and a question about the steps in the procedure, you are required to answer the question in one sentence. Goal: {title} Procedure: {procedure} Must Step {i} happen before Step {j}? Select between yes or no

Models Struggle at Predicting Step Order



Models Struggle at Predicting Step Order

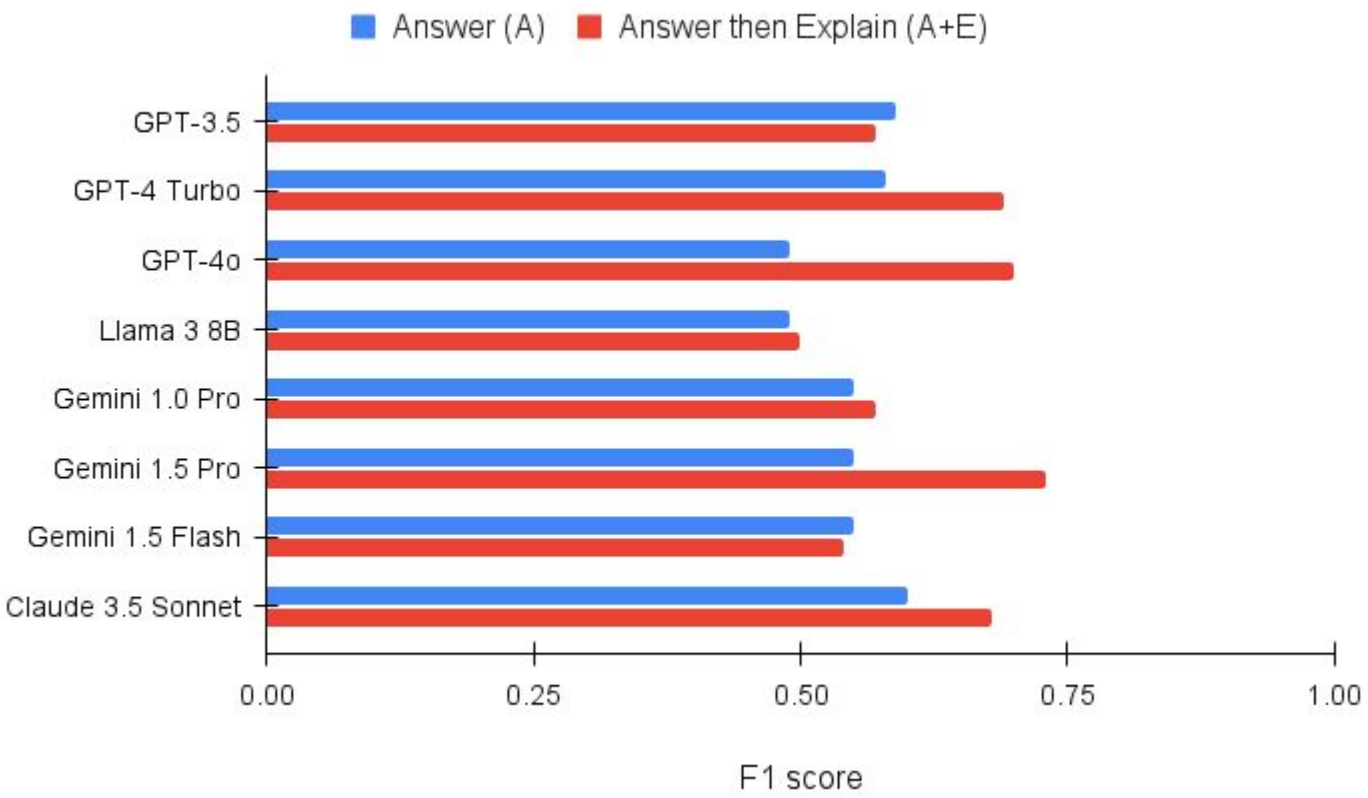


Experiments - Answer + Explanation (A+E)

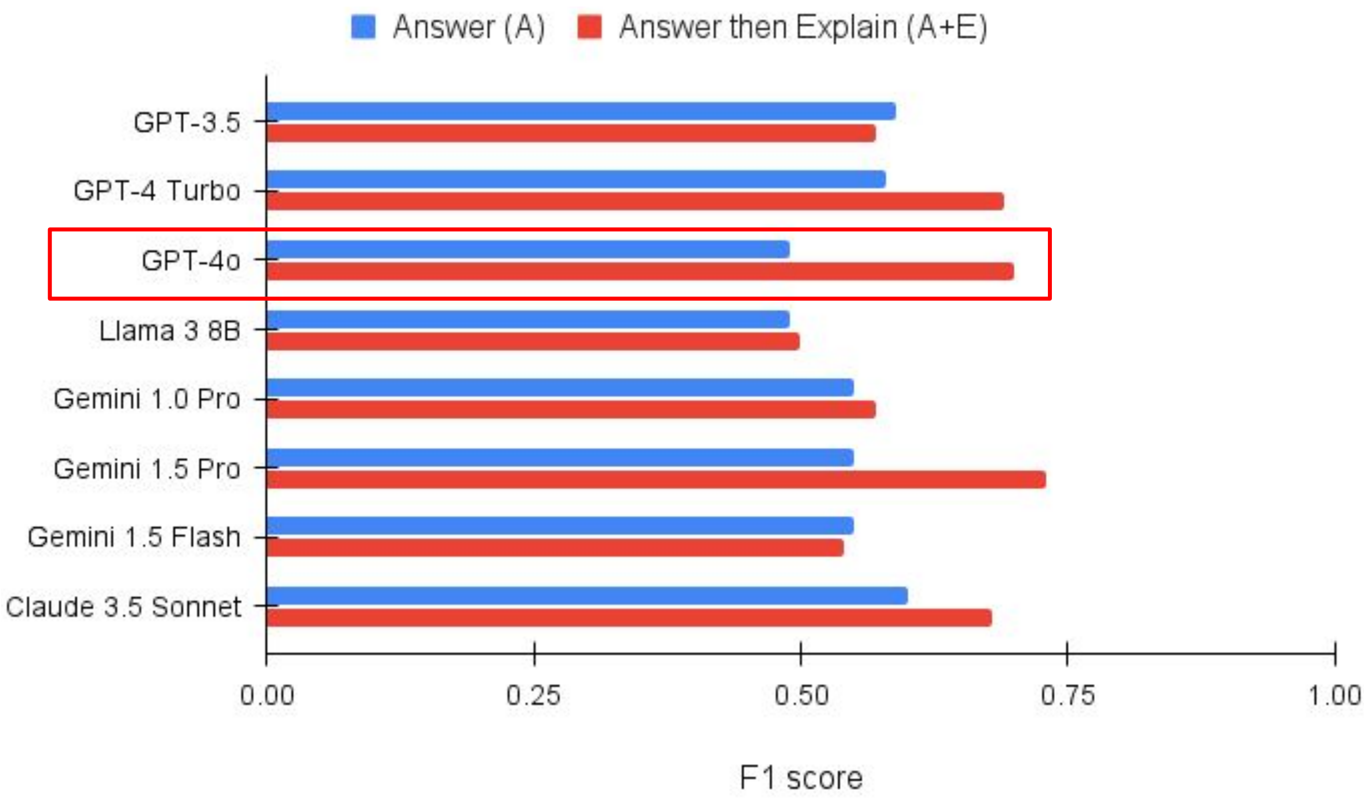
- Binary Prediction and Post-hoc Explanation

Answer + Explanation (A+E)	Given a goal, a procedure to achieve that goal and a question about the steps in the procedure, you are required to answer the question in one sentence. Goal: {title} Procedure: {procedure} 1. Must Step {i} happen before Step {j}? Select between yes or no 2. Explain why or why not. Format your answer as JSON with the key value pairs "binary_answer": "yes/no answer to Q1", "why_answer": "answer to Q2"

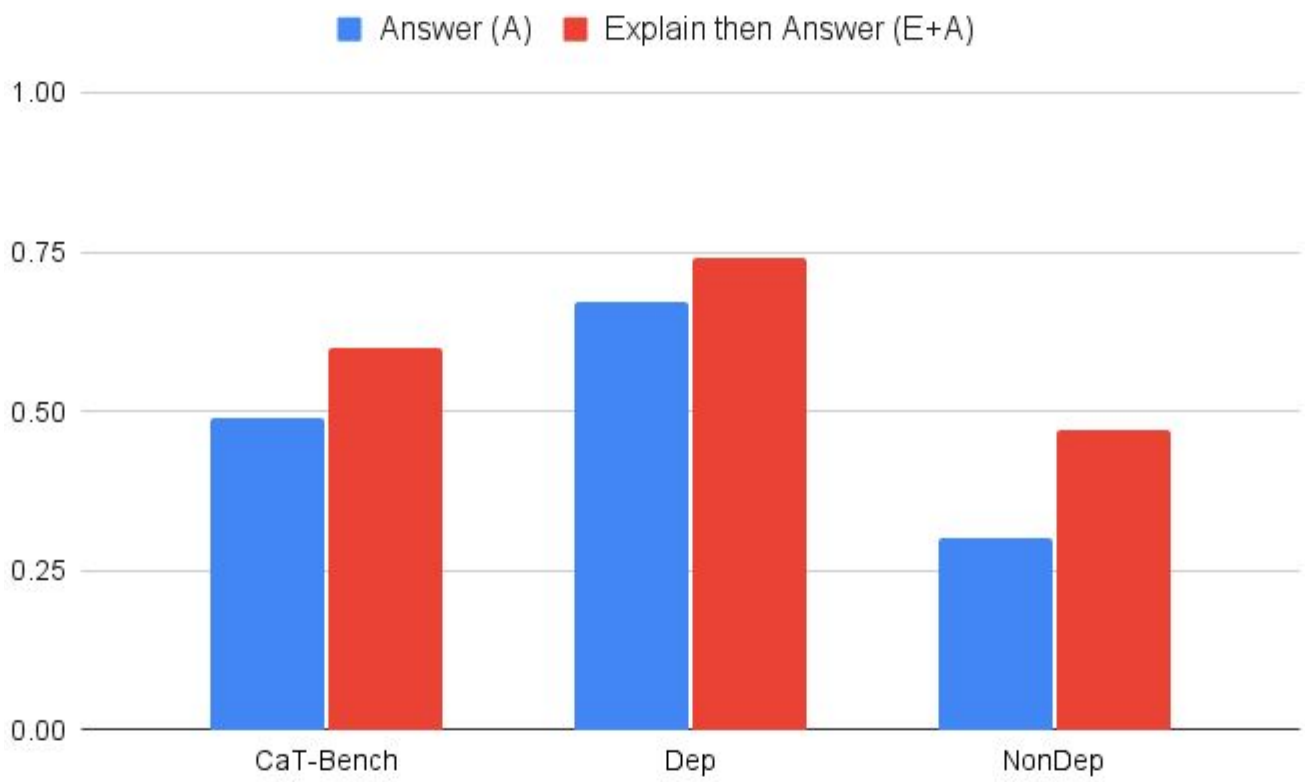
Generating Explanations Helps



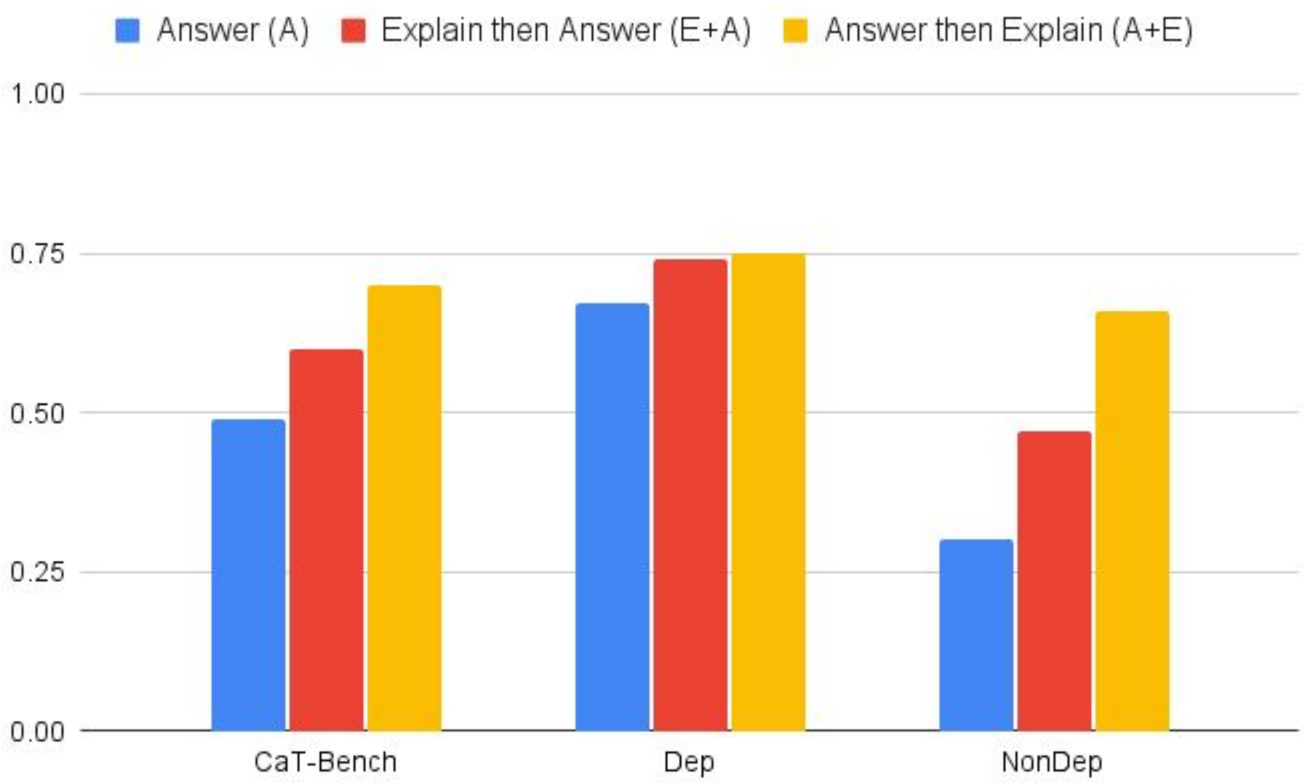
Generating Explanations Helps



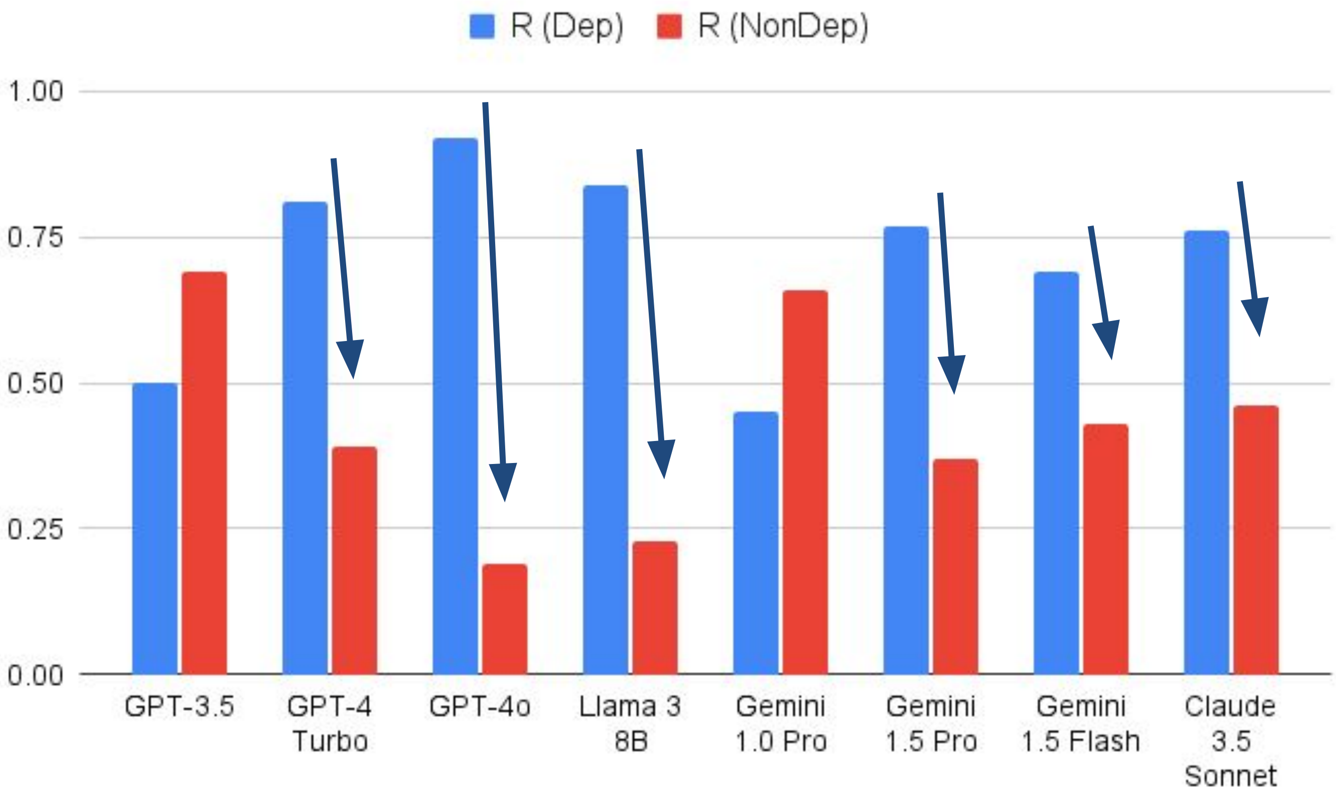
GPT-4o: Chain-of-Thought



GPT-4o: Post-Hoc Explanation Beats Chain-of-Thought



Models are Biased Towards Dependence



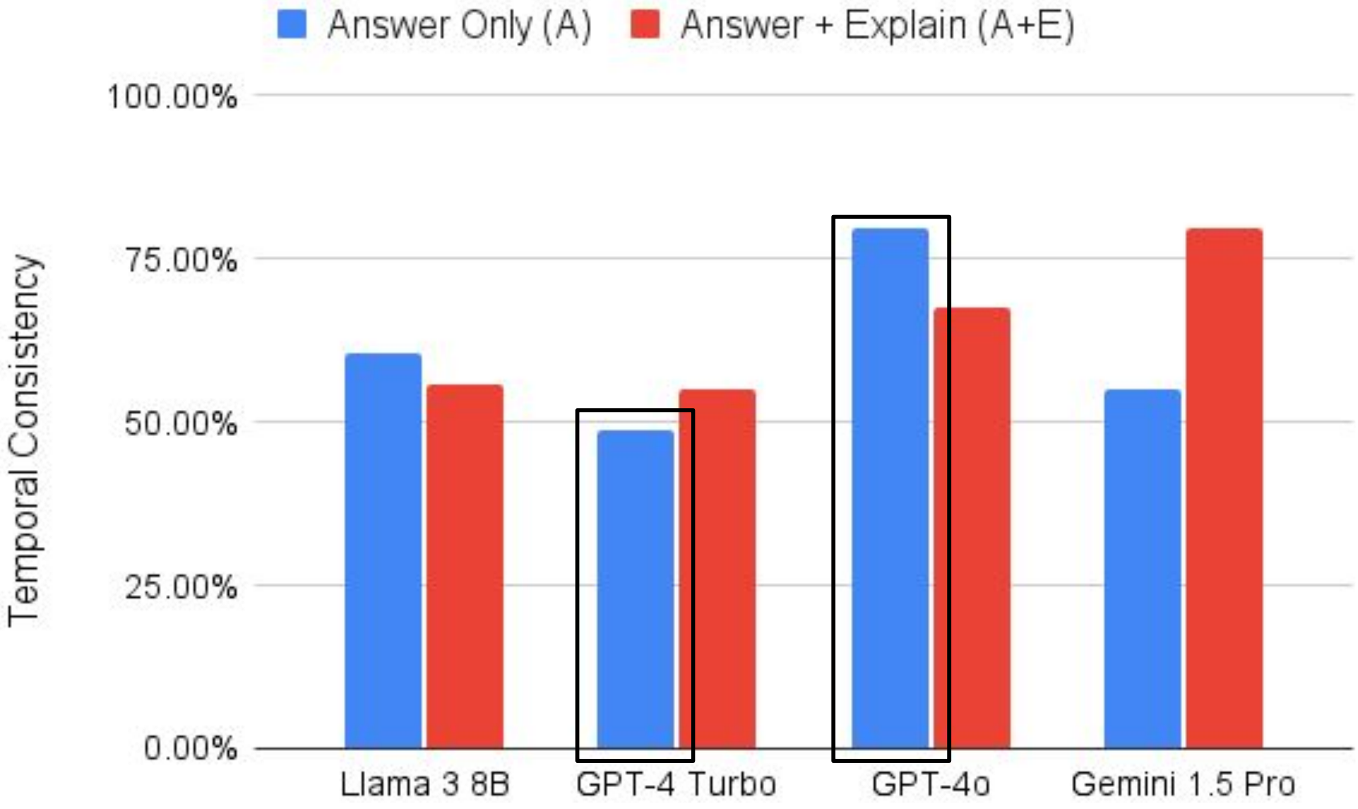
Robustness: Temporal Consistency

Goal: Bake Almond and Chocolate Cake
… Step 6: Add in ground almonds. Step 7: Add half flour and half milk. Step 8: Use wooden spoon to stir. … Step 12: Whip cream till stiff peaks …

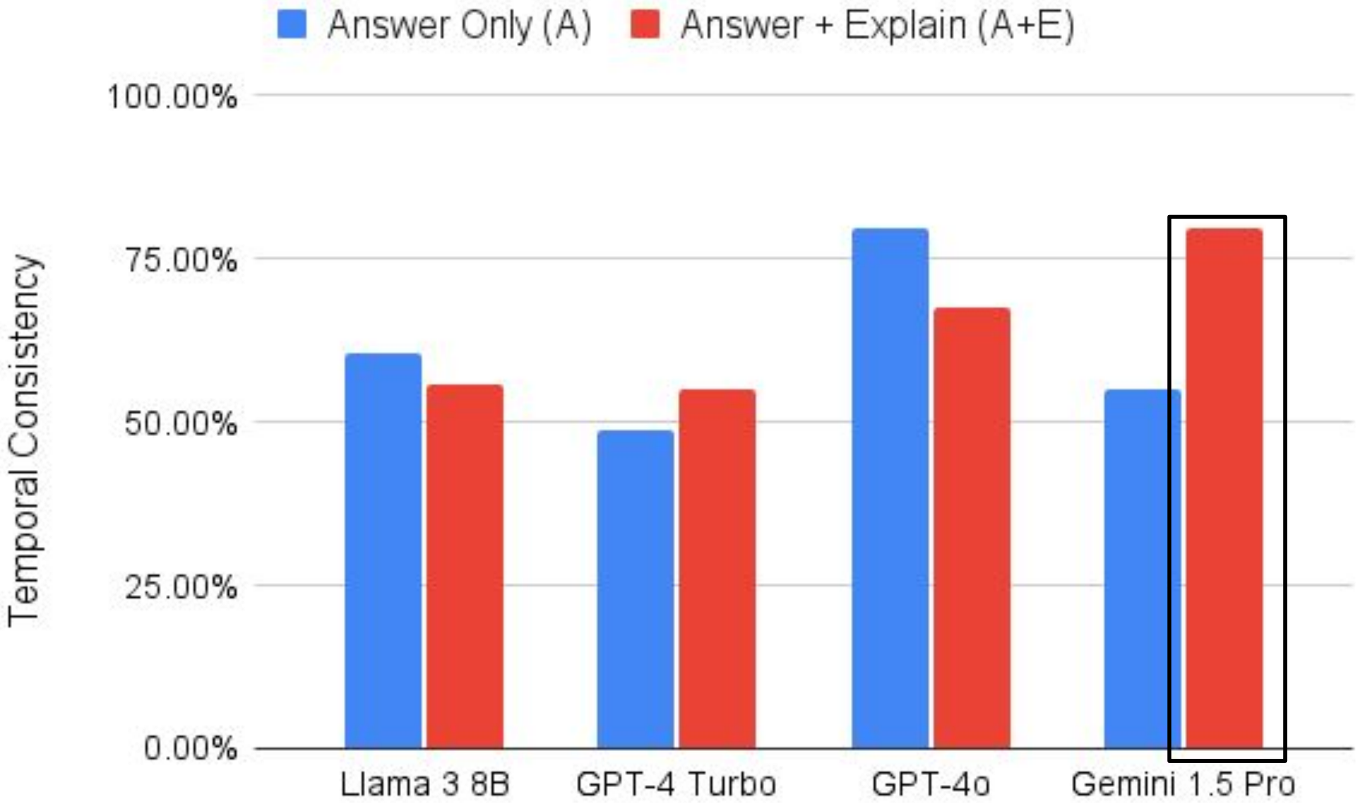
Q: Must Step 6 happen before Step 8?	Before
Q: Must Step 8 happen after Step 6?	After

The model's answers to the 'before' and 'after' questions should be the same.

How Robust are Models?



How Robust are Models?



Conclusion

- We introduce an easy-to-evaluate plan based reasoning benchmark.
- SOTA LLMs struggle with this simple task but post-hoc explanations help.
- Models are inconsistent and biased towards predicting step dependence.

Improvement	Evaluation
Planning-Augmented Parsing *ACL NLRSE 2023* Using Planning to Improve Semantic Parsing of Instructional Texts Planning constraints rank/correct candidate action sequences.	**CaT-Bench** *EMNLP 2024* Causal and Temporal Dependencies in Plans Tests whether LMs understand preconditions, step ordering, and dependence.
Compositional Instruction Following *TMLR 2024 / RLC 2025* Compositional Instruction Following with LMs and RL Learns language-to-policy composition from environment feedback.	**MET-Bench** *ICML 2026* Multimodal Entity State Tracking Tests postconditions and entity state tracking across text and images.

Improvement	Evaluation
Planning-Augmented Parsing *ACL NLRSE 2023* Using Planning to Improve Semantic Parsing of Instructional Texts Planning constraints rank/correct candidate action sequences.	**CaT-Bench** *EMNLP 2024* Causal and Temporal Dependencies in Plans Tests whether LMs understand preconditions, step ordering, and dependence.
Compositional Instruction Following *TMLR 2024 / RLC 2025* Compositional Instruction Following with LMs and RL Learns language-to-policy composition from environment feedback.	**MET-Bench** *ICML 2026* Multimodal Entity State Tracking Tests postconditions and entity state tracking across text and images.

MET-Bench: Multimodal Entity Tracking for Evaluating the Limitations of Vision-Language and Reasoning Models

Vanya Cohen and Raymond Mooney

Motivation

- Agents must also model postconditions: how entities and the world state change after each action to do forward planning.

Motivation

- Agents need to track the state of the world in text and visual modalities.
  - Entity state tracking evaluated extensively in text tasks[1].
- Paired text and image tracking tasks to measure the visual tracking performance gap.

1. Shubham Toshniwal, Sam Wiseman, Karen Livescu, and Kevin Gimpel. Chess as a Testbed for Language Model State Tracking. In Proceedings of the AAAI 2022
   Najoung Kim and Sebastian Schuster. Entity Tracking in Language Models. ACL 2023
   Erwan Fagnou, Paul Caillon, Blaise Delattre, and Alexandre Allauzen. Chain and Causal Attention for Efficient Entity Tracking. EMNLP 2024

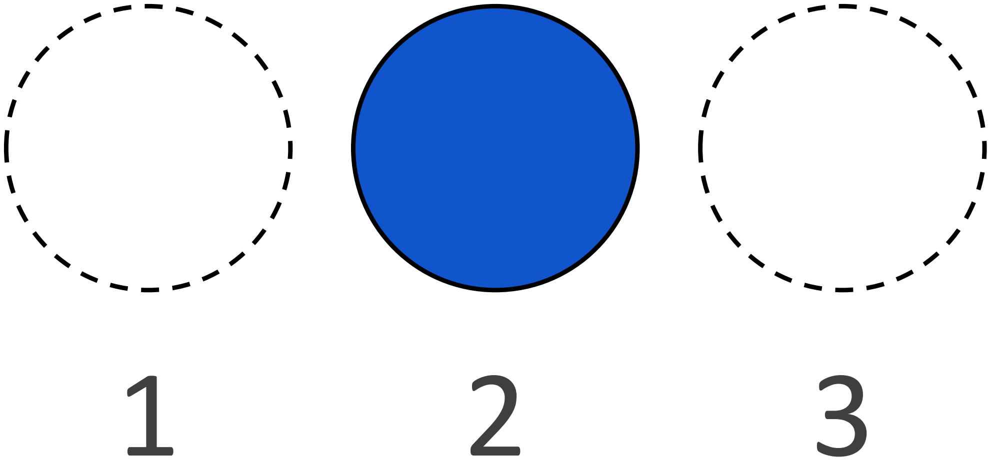
Entity State Tracking

Example: Shell Game

- Ball placed under one of three shells.
- Series of swaps change the hidden ball position.
- Predict the final position of the ball.

Example: Text Shell Game

Ball Position: 2

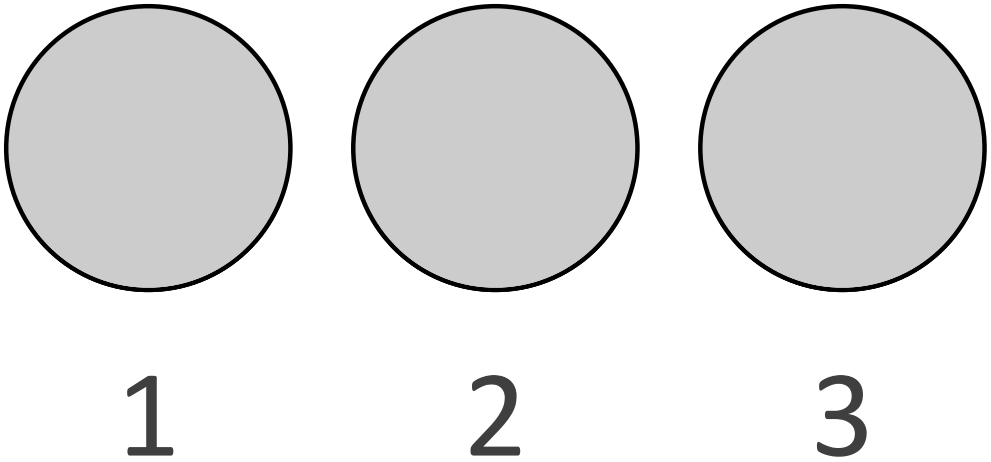


Example: Text Shell Game

Swap 1, 2

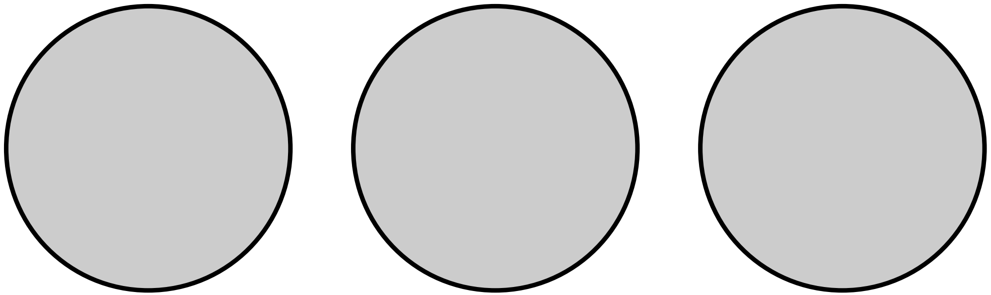
Swap 2, 3

Swap 1, 3



Example: <u>Text</u> Shell Game

Q: Final State?

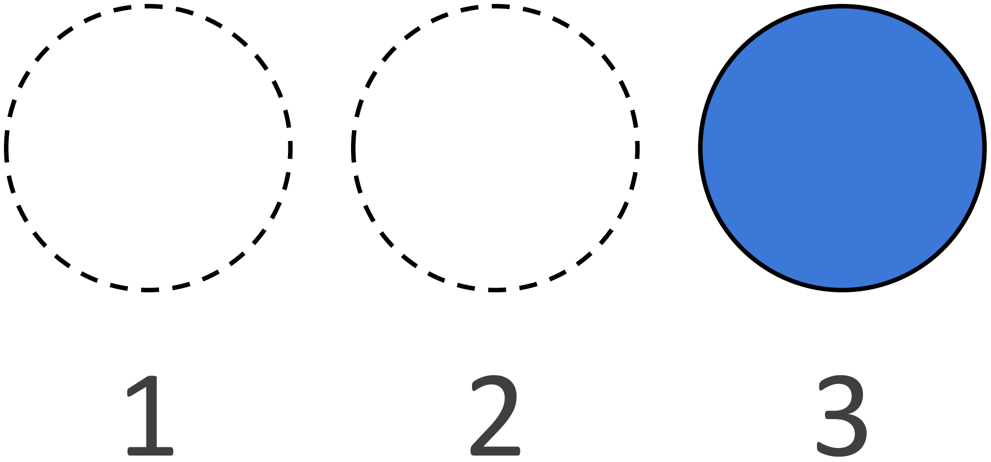


1 2 3

Example: Text Shell Game

Q: Final State?

A: 3



Example: Multimodal Shell Game

Text Input:

Ball Position: 2

Example: Multimodal Shell Game

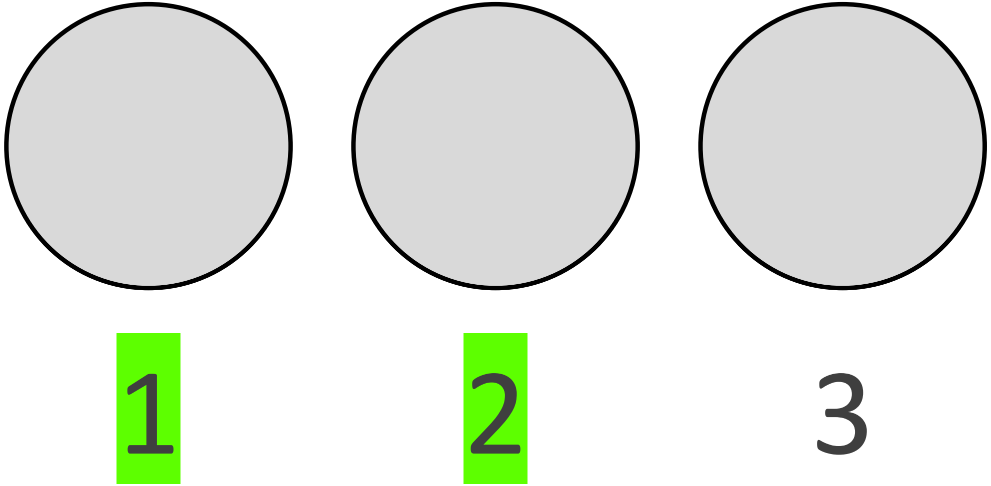
Model receives a text description of the initial state

Text Input:

Ball Position: 2

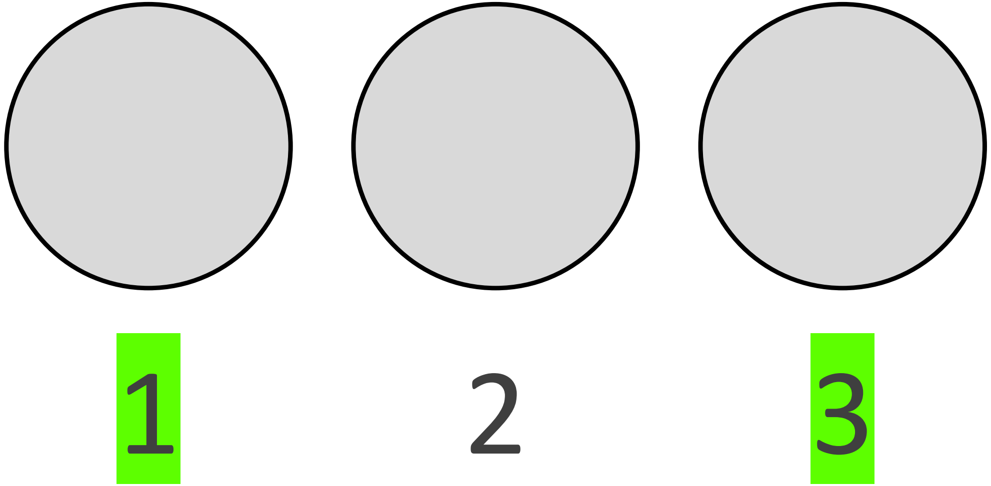
Example: Multimodal Shell Game

Model receives an image depiction of the action



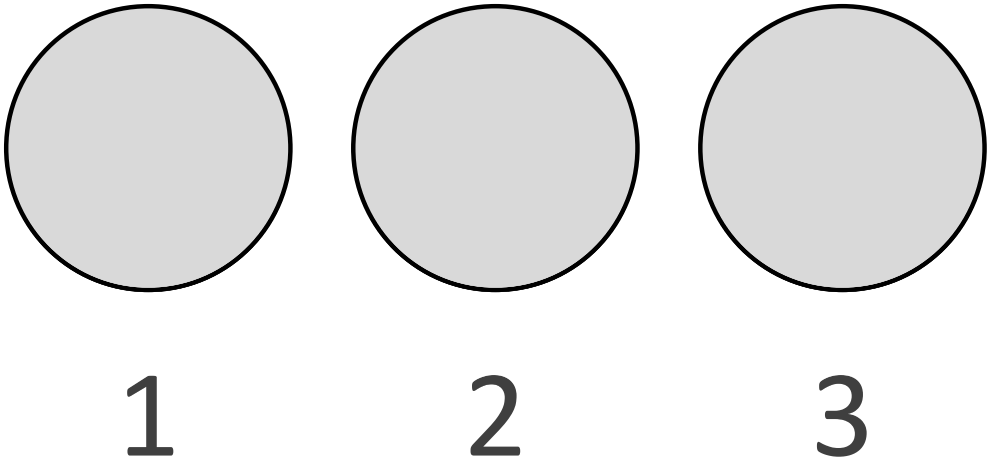
Example: Multimodal Shell Game

Model receives an image depiction of the action



Example: Multimodal Shell Game

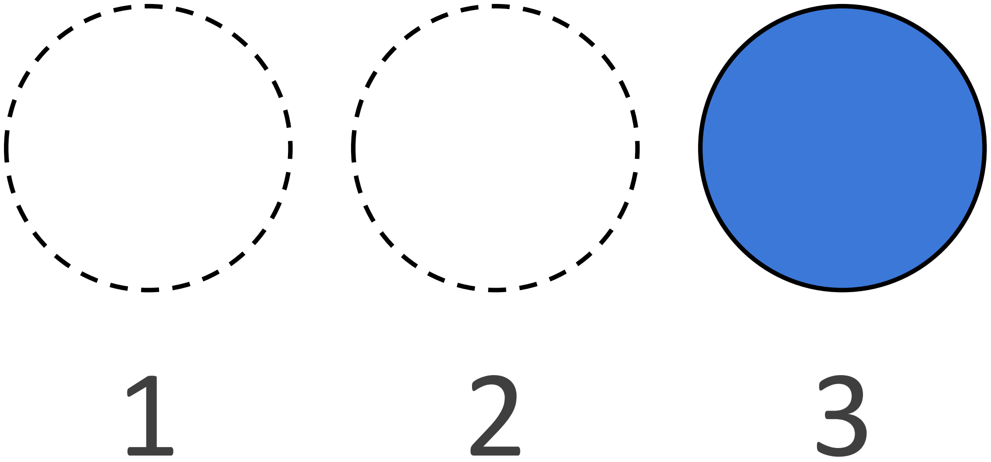
Q: Final State?



Example: Multimodal Shell Game

Q: Final State?

A: 3



Example: Multimodal Chess

Text Input:
Forsyth–Edwards Notation (FEN)

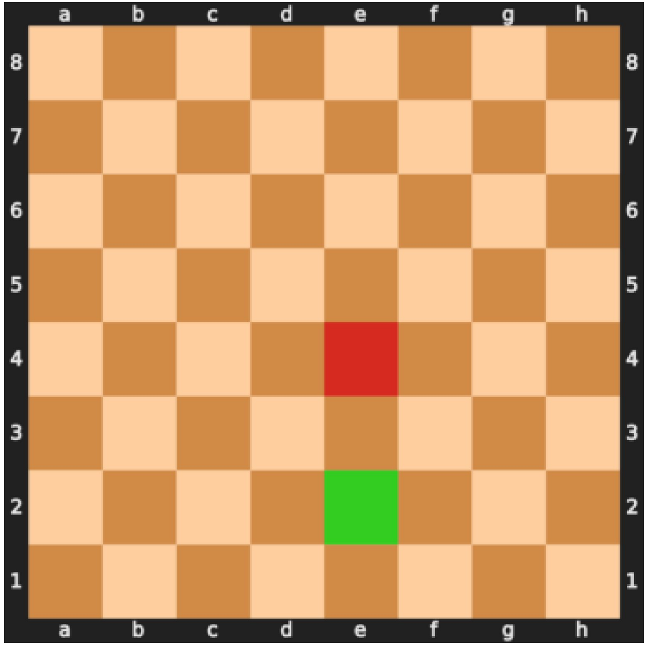
"rnbqkbnr/pppppppp/8/8/8/8/
PPPPPPPP/RNBQKBNR w KQkq -
0 1"

Example: Multimodal Chess

Universal Chess Interface Notation

e2e4

"The piece at e2 moves to e4."



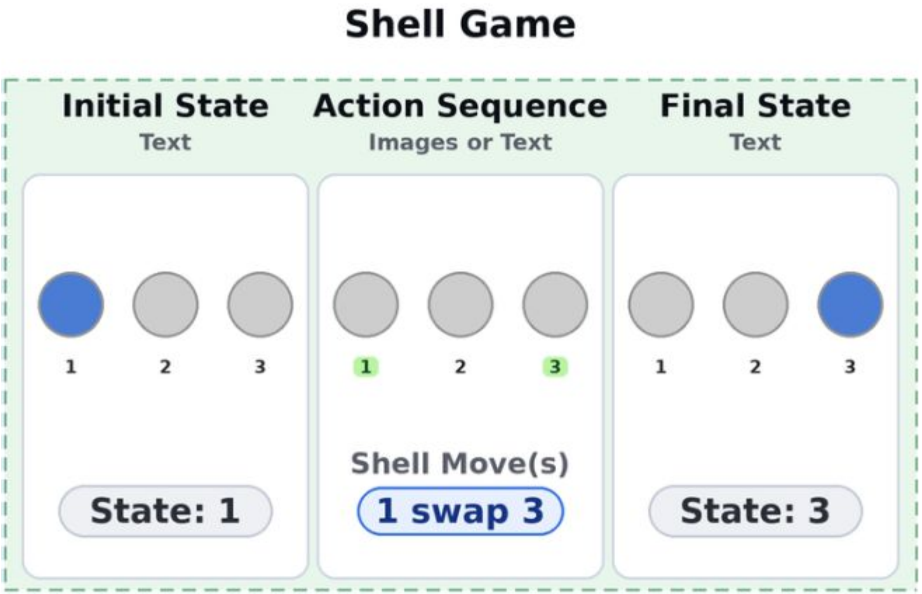
Example: Multimodal Chess

Q: Final State?

A:
rnbqkbnr/pppppppp/8/8/4P
3/8/PPPP1PPP/RNBQKBNR b
KQkq e3 0 1

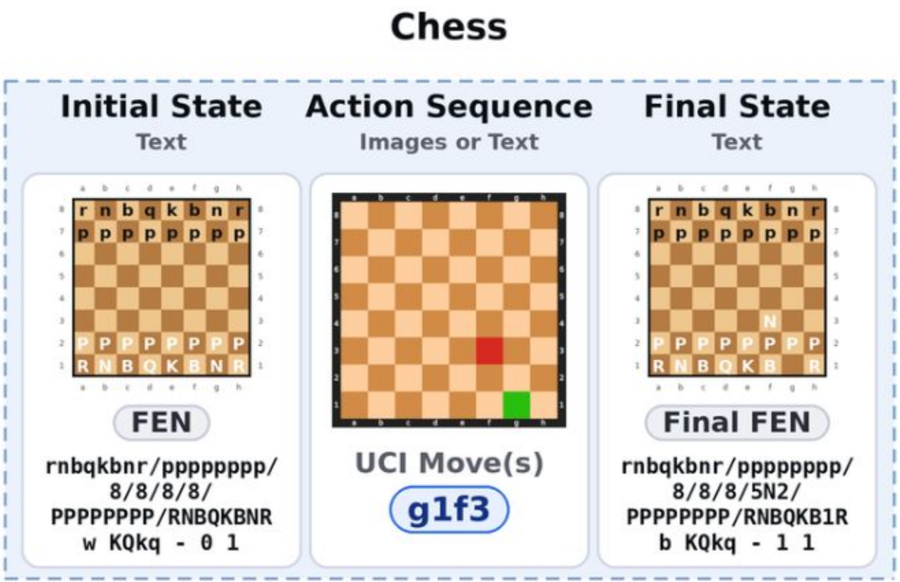
MET-Bench: Shell Game

- 56k tasks randomly generated.



MET-Bench: Chess

- 105k tasks from real chess games.



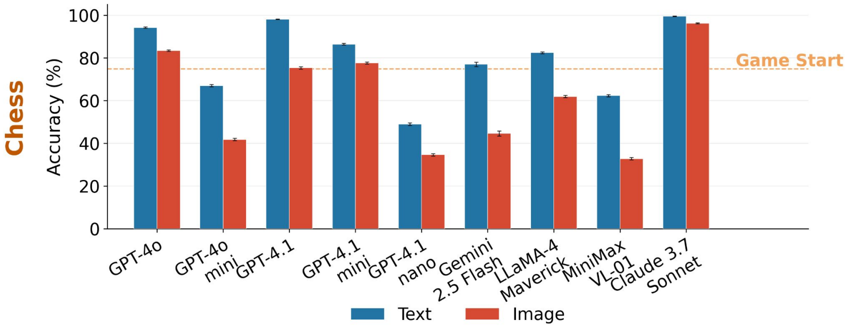
MET-Bench: Minecraft

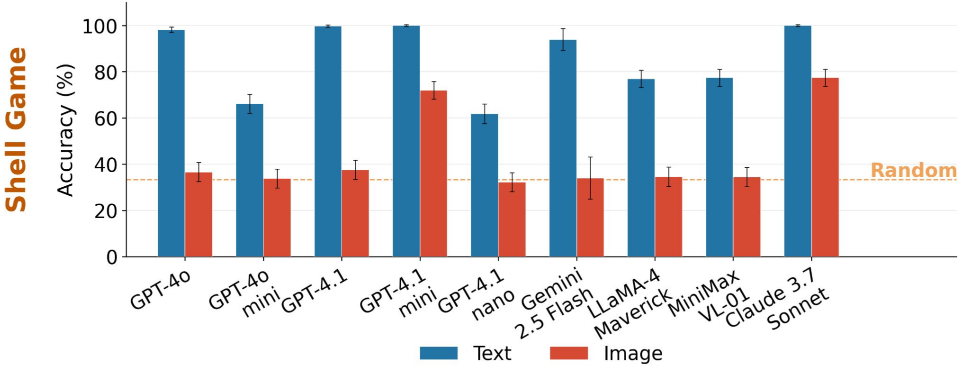
- 7,000 MCQ tasks.
- Initial and Final State: Text/Image

- Gathering wood, exploring, and building.
- Distractor views selected from nearby segments within the same trajectory.

Tracking in Text Outperforms Images

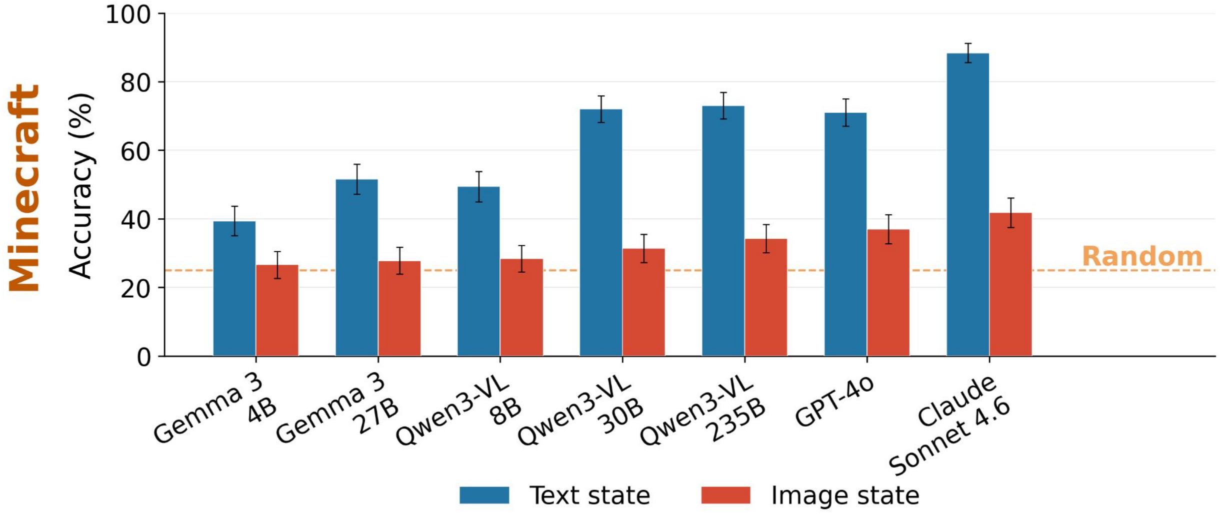
Chain-of-Thought, 10 actions, 500 examples



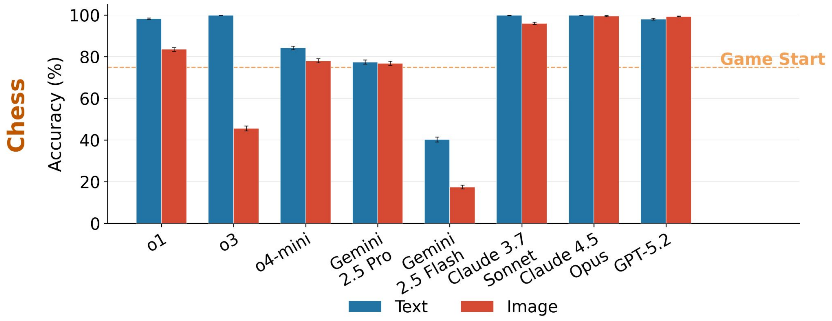


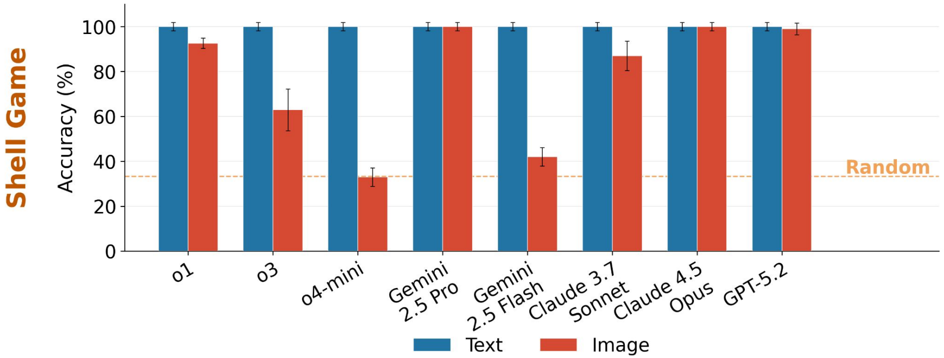
Tracking in Text Outperforms Images

Chain-of-Thought / Reasoning Models, 500 examples

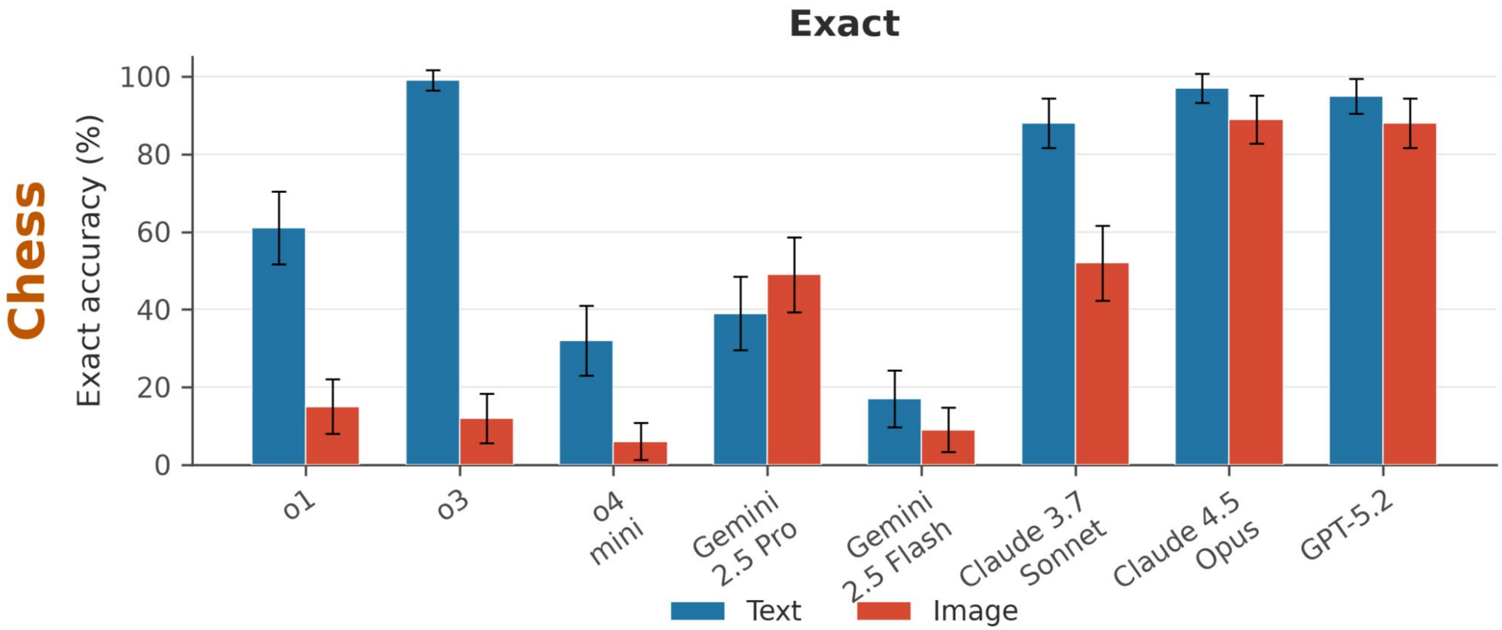


Reasoning Training Improves Performance



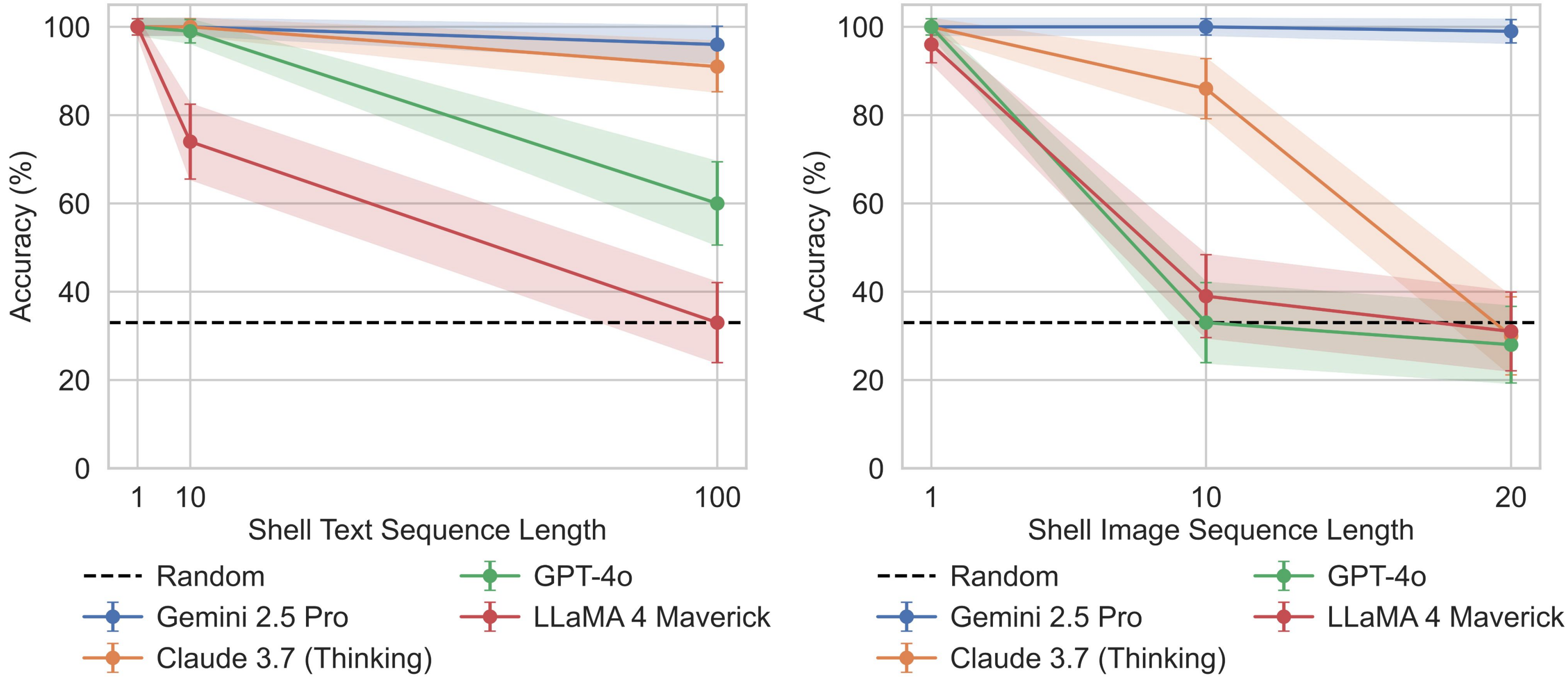


Stronger Metrics Reveal Limitations



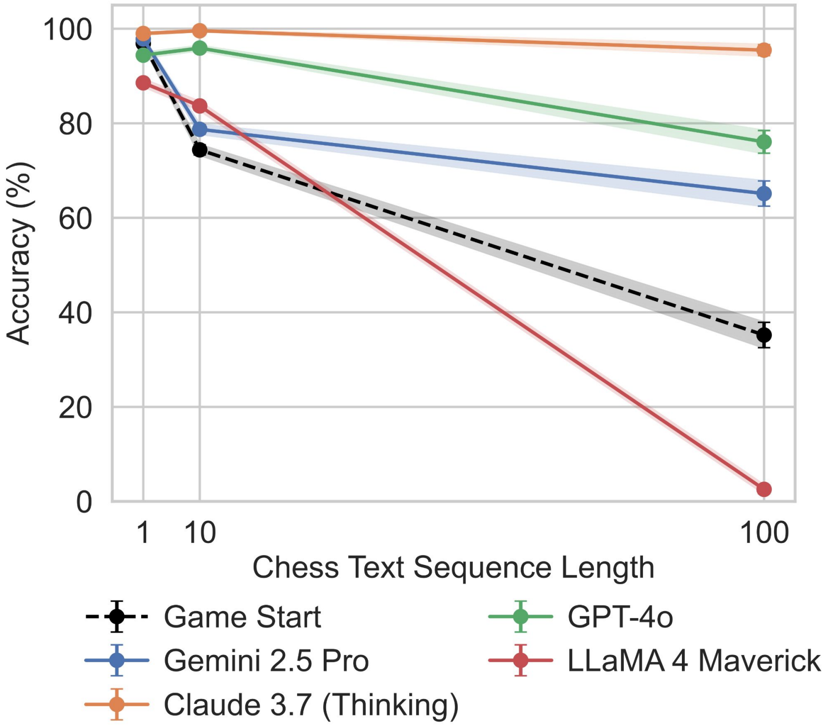
Models Struggle with Long Sequences

Shell Game



Models Struggle with Long Sequences

Chess



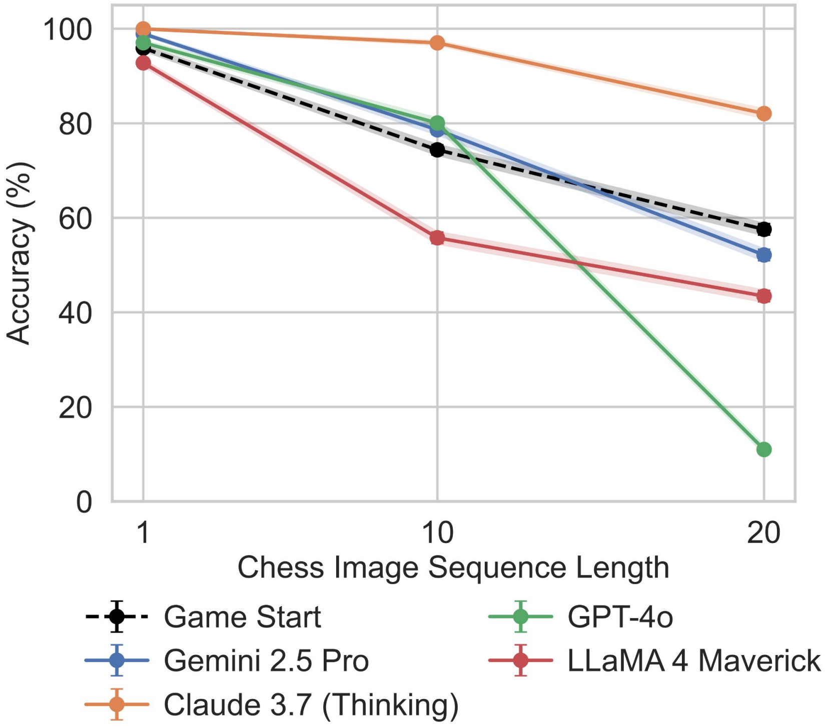
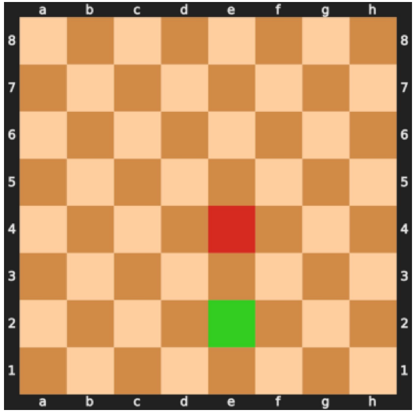
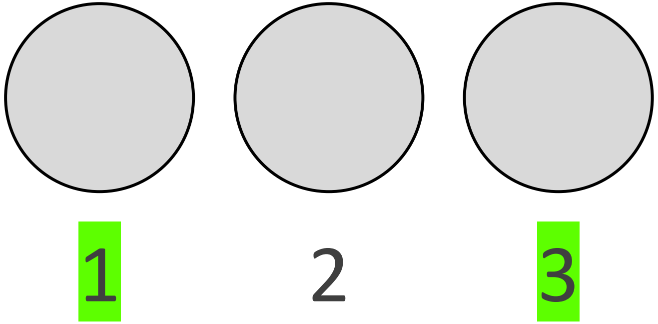


Image Action Perception

Q: In UCI notation what move does the arrow on the chessboard represent? The move is from the green square to the red square. (e.g., 'g2f3').



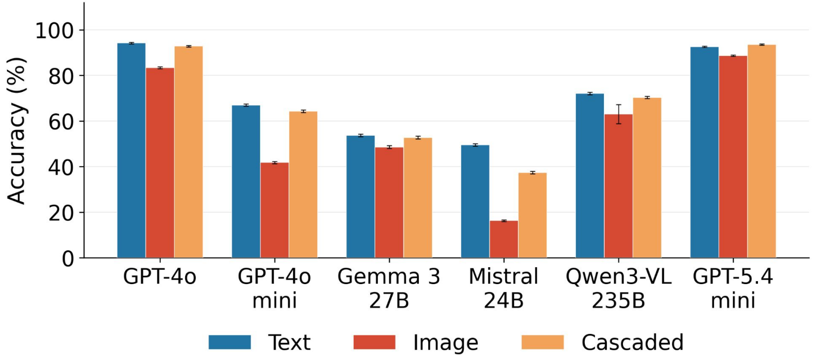
Q: Which shells are being swapped in the image? Shells are labeled '1', '2', '3' and the shells being swapped have their numbers highlighted in green. Only output a move in the format 'X swap Y' and nothing else.



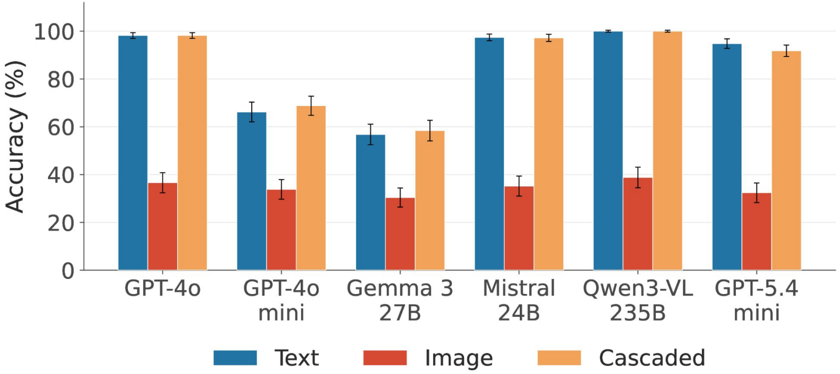
Performance is Limited by Visual Reasoning

Cascaded: Image Actions → Text Actions → State Prediction

Chess

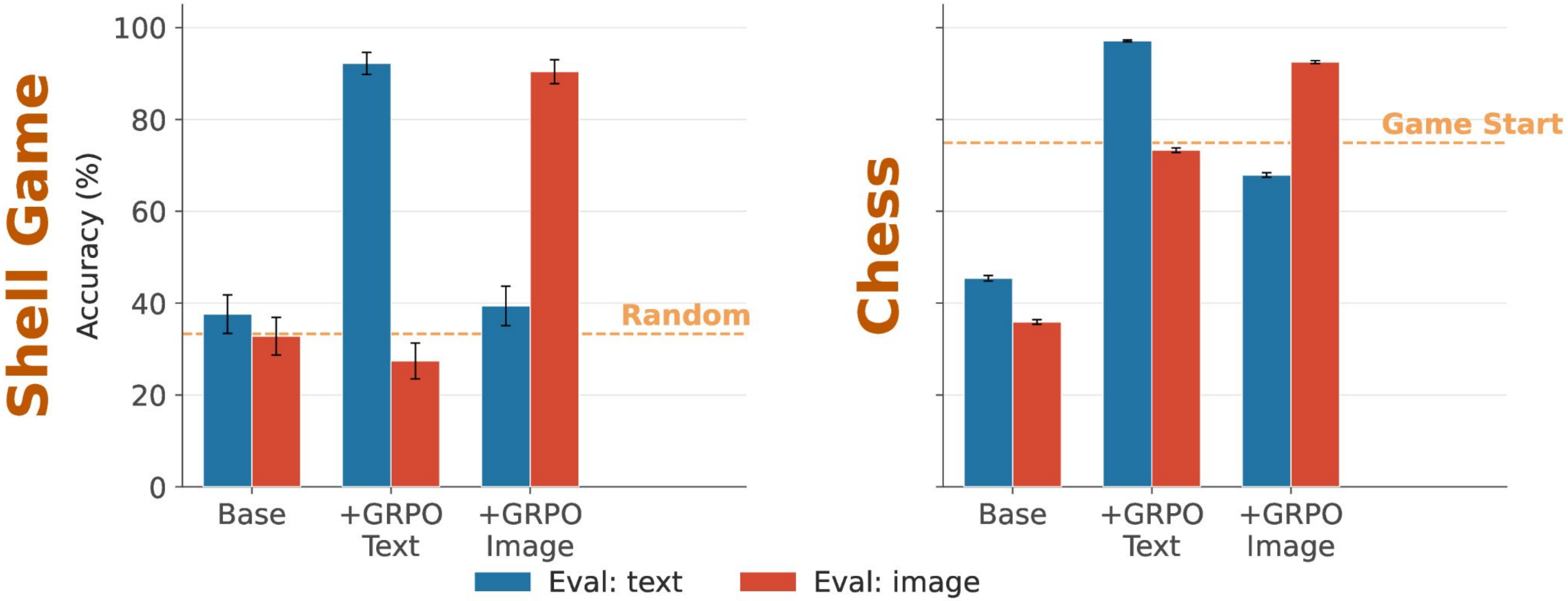


Shell Game



RL Improves Tracking Performance

Gemma 3 4B IT



GRPO: Shao, Zhihong, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." (2024).

Conclusion

- Need to improve the visual state reasoning capabilities of VLMs to match their text performance.
- RL reasoning training offers one path, but multimodal representations remain a limitation.

Contents

- Introduction and Background
- Completed Work
- **Future Work**
  - Short-Term Extensions
  - Long-Term Themes

Short-Term Extensions

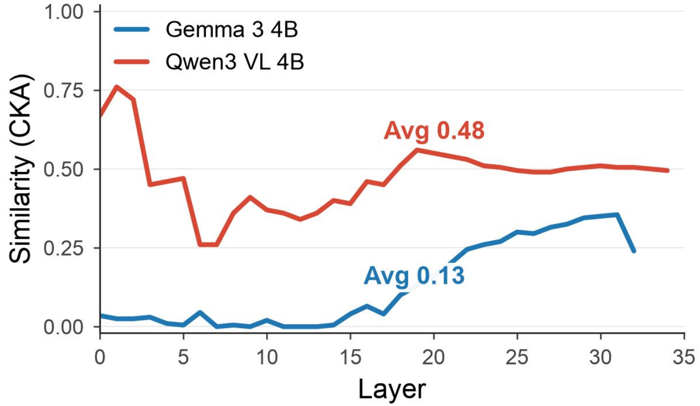
- **Reduce manual structure:** automatically induce planning domains and learn compositional policy bases without fixed curricula.
- **Broaden evaluation:** extend dependency and state-tracking benchmarks to richer procedural and physically embodied domains.

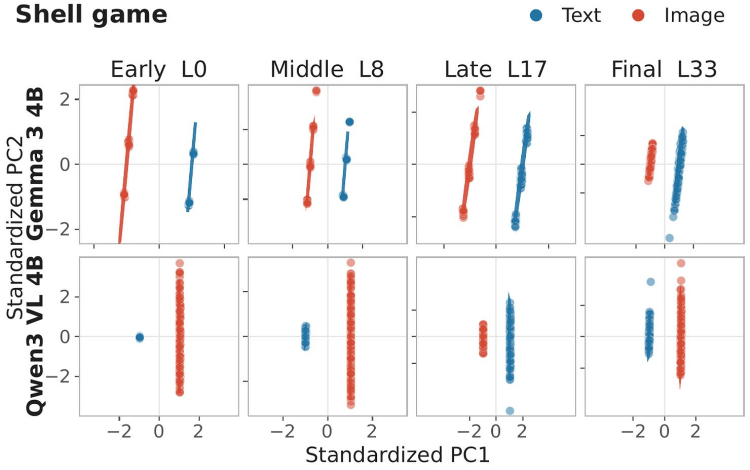
Long-Term Themes

- **Understand VLM failures:** identify why equivalent text and image inputs produce different internal representations and predictions.

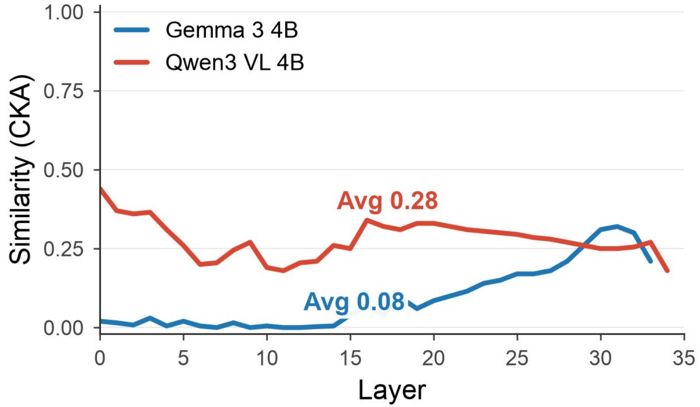
Liu et al. Unraveling and Mitigating Safety Alignment Degradation of Vision-Language Models. In Findings of the Association for Computational Linguistics: ACL 2025
Yaras et al. Explaining and Mitigating the Modality Gap in Contrastive Multimodal Learning. 2025.

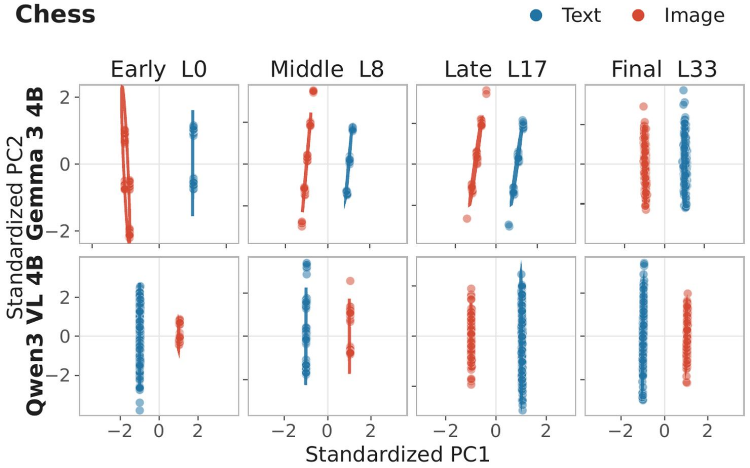
Shell game





Chess



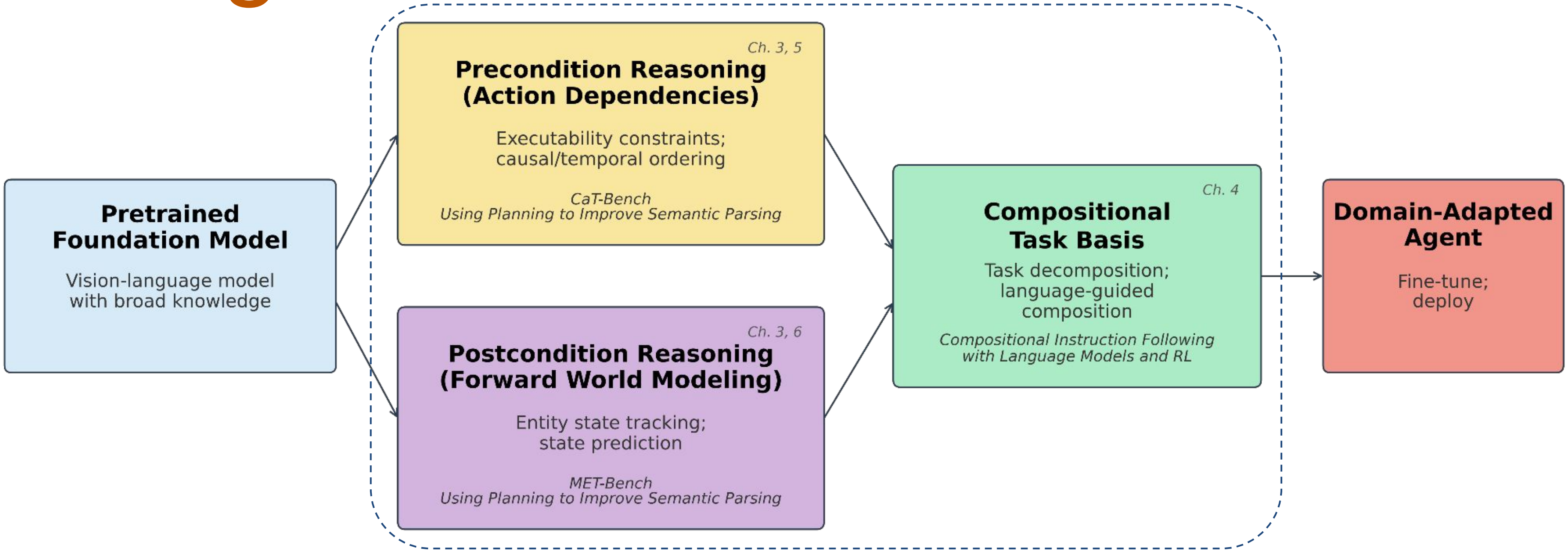


CKA: Kornblith, Simon, et al. "Similarity of neural network representations revisited." International conference on machine learning. 2019.

Long-Term Themes

- Build a unified agent combining precondition reasoning, forward world modeling, reusable skills, and domain adaptation.

Long-Term Themes



RL Action-Reasoning Training

Conclusions

- Leveraging compositional and domain structure improves action reasoning in LMMs, yielding better sample efficiency, plan executability, and zero-shot generalization to novel tasks.

Conclusions

- New benchmarks reveal that LMMs struggle with causal and temporal plan reasoning and show significantly weaker visual state tracking compared to equivalent text inputs.

Conclusions

- RL reasoning training and structured generation offer paths toward closing these gaps, pointing to unified agents that combine precondition reasoning, world modeling, and reusable compositional skills.

Acknowledgements