ICAPS 2025 Workshop on Planning in the Era of LLMs

PDDL-Instruct: Enhancing Symbolic Planning Capabilities in LLMs through Logical Chain-of-Thought Instruction Tuning

Pulkit Verma[*]

Ngoc La[*]

Anthony Favier[*]

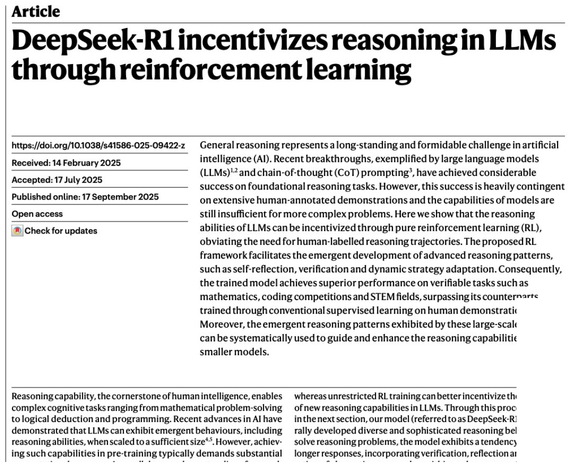
Swaroop Mishra[†]

Julie A. Shah[*]

[*] MIT CSAIL Computer Science & Artificial Intelligence Laboratory

[†] MAI Microsoft AI

LLMs are good at Reasoning



Article

DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning

https://doi.org/10.1038/s41586-025-09422-z

Received: 14 February 2025

Accepted: 17 July 2025

Published online: 17 September 2025

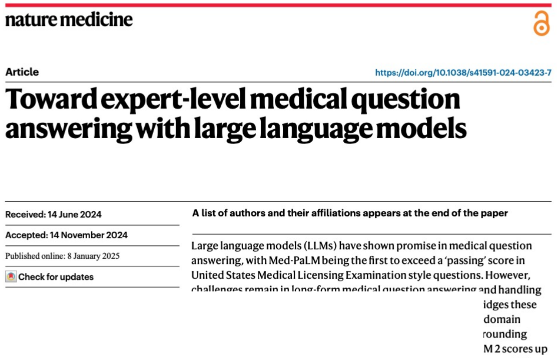
Open access

Check for updates

General reasoning represents a long-standing and formidable challenge in artificial intelligence (AI). Recent breakthroughs, exemplified by large language models (LLMs)[1,2] and chain-of-thought (CoT) prompting[3], have achieved considerable success on foundational reasoning tasks. However, this success is heavily contingent on extensive human-annotated demonstrations and the capabilities of models are still insufficient for more complex problems. Here we show that the reasoning abilities of LLMs can be incentivized through pure reinforcement learning (RL), obviating the need for human-labelled reasoning trajectories. The proposed RL framework facilitates the emergent development of advanced reasoning patterns, such as self-reflection, verification and dynamic strategy adaptation. Consequently, the trained model achieves superior performance on verifiable tasks such as mathematics, coding competitions and STEM fields, surpassing its counterparts trained through conventional supervised learning on human demonstrati
Moreover, the emergent reasoning patterns exhibited by these large-scal
can be systematically used to guide and enhance the reasoning capabiliti
smaller models.

Reasoning capability, the cornerstone of human intelligence, enables complex cognitive tasks ranging from mathematical problem-solving to logical deduction and programming. Recent advances in AI have demonstrated that LLMs can exhibit emergent behaviours, including reasoning abilities, when scaled to a sufficient size[4,5]. However, achieving such capabilities in pre-training typically demands substantial

whereas unrestricted RL training can better incentivize th
of new reasoning capabilities in LLMs. Through this proc
in the next section, our model (referred to as DeepSeek-R
rally developed diverse and sophisticated reasoning be
solve reasoning problems, the model exhibits a tendenc
longer responses, incorporating verification, reflection a



nature medicine

Article

https://doi.org/10.1038/s41591-024-03423-7

Toward expert-level medical question answering with large language models

Received: 14 June 2024

Accepted: 14 November 2024

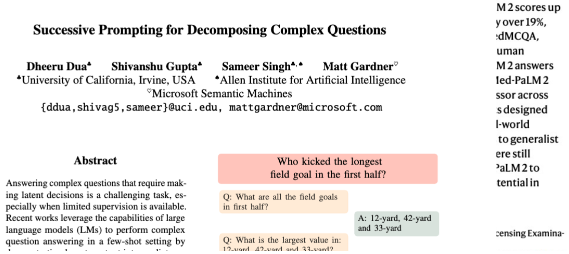
Published online: 8 January 2025

Check for updates

A list of authors and their affiliations appears at the end of the paper

Large language models (LLMs) have shown promise in medical question answering, with Med-PaLM being the first to exceed a 'passing' score in United States Medical Licensing Examination style questions. However, challenges remain in long-form medical question answering and handling
idges these
domain
rounding
M 2 scores up
y over 19%,
dMCQA,
uman
M 2 answers
ed-PaLM 2
ssor across
s designed
l-world
to generalist
ere still
aLM 2 to
tential in

censing Examina-



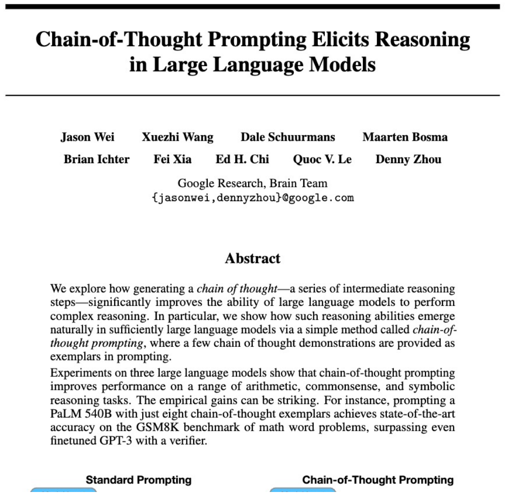
Successive Prompting for Decomposing Complex Questions

Dheeru Dua[♣] Shivanshu Gupta[♣] Sameer Singh[♣,♠] Matt Gardner[♡]
[♣]University of California, Irvine, USA [♠]Allen Institute for Artificial Intelligence
[♡]Microsoft Semantic Machines
{ddua,shivag5,sameer}@uci.edu, mattgardner@microsoft.com

Abstract

Answering complex questions that require making latent decisions is a challenging task, especially when limited supervision is available. Recent works leverage the capabilities of large language models (LMs) to perform complex question answering in a few-shot setting by

Who kicked the longest field goal in the first half?
Q: What are all the field goals in first half?
A: 12-yard, 42-yard and 33-yard
Q: What is the largest value in: 12-yard, 42-yard and 33-yard?



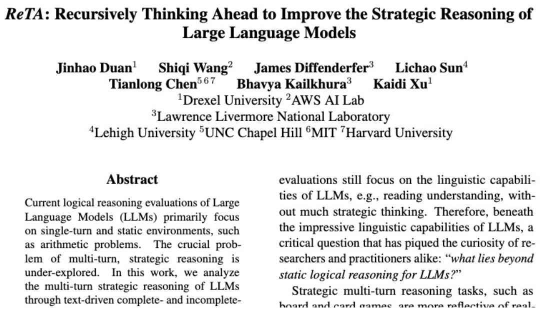
Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei Xuezhi Wang Dale Schuurmans Maarten Bosma
Brian Ichter Fei Xia Ed H. Chi Quoc V. Le Denny Zhou
Google Research, Brain Team
{jasonwei,dennyzhou}@google.com

Abstract

We explore how generating a *chain of thought*—a series of intermediate reasoning steps—significantly improves the ability of large language models to perform complex reasoning. In particular, we show how such reasoning abilities emerge naturally in sufficiently large language models via a simple method called *chain-of-thought prompting*, where a few chain of thought demonstrations are provided as exemplars in prompting.

Experiments on three large language models show that chain-of-thought prompting improves performance on a range of arithmetic, commonsense, and symbolic reasoning tasks. The empirical gains can be striking. For instance, prompting a PaLM 540B with just eight chain-of-thought exemplars achieves state-of-the-art accuracy on the GSM8K benchmark of math word problems, surpassing even finetuned GPT-3 with a verifier.



ReTA: Recursively Thinking Ahead to Improve the Strategic Reasoning of Large Language Models

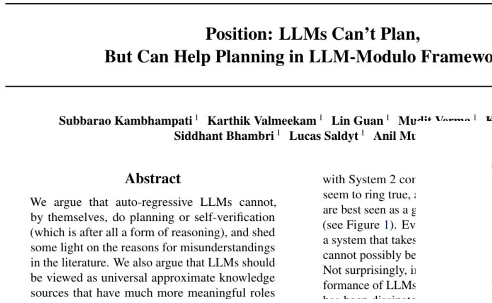
Jinhao Duan[1] Shiqi Wang[2] James Diffenderfer[3] Lichao Sun[4]
Tianlong Chen[5 6 7] Bhavya Kailkhura[3] Kaidi Xu[1]
[1]Drexel University [2]AWS AI Lab
[3]Lawrence Livermore National Laboratory
[4]Lehigh University [5]UNC Chapel Hill [6]MIT [7]Harvard University

Abstract

Current logical reasoning evaluations of Large Language Models (LLMs) primarily focus on single-turn and static environments, such as arithmetic problems. The crucial problem of multi-turn, strategic reasoning is under-explored. In this work, we analyze the multi-turn strategic reasoning of LLMs through text-driven complete- and incomplete- evaluations still focus on the linguistic capabilities of LLMs, e.g., reading understanding, without much strategic thinking. Therefore, beneath the impressive linguistic capabilities of LLMs, a critical question that has piqued the curiosity of researchers and practitioners alike: "*what lies beyond static logical reasoning for LLMs?*"

Strategic multi-turn reasoning tasks, such as

LLMs are bad at Planning



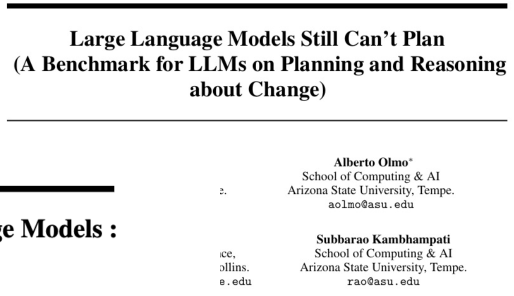
Position: LLMs Can't Plan,
But Can Help Planning in LLM-Modulo Frameworks

Subbarao Kambhampati[1] Karthik Valmeekam[1] Lin Guan[1] Mudit Verma[1] Kaya Stechly[1]
Siddhant Bhambri[1] Lucas Saldyt[1] Anil Mu

Abstract

We argue that auto-regressive LLMs cannot, by themselves, do planning or self-verification (which is after all a form of reasoning), and shed some light on the reasons for misunderstandings in the literature. We also argue that LLMs should be viewed as universal approximate knowledge sources that have much more meaningful roles

with System 2 co
seem to ring true,
are best seen as a
(see Figure 1). Ev
a system that take
cannot possibly be
Not surprisingly, i
formance of LLM

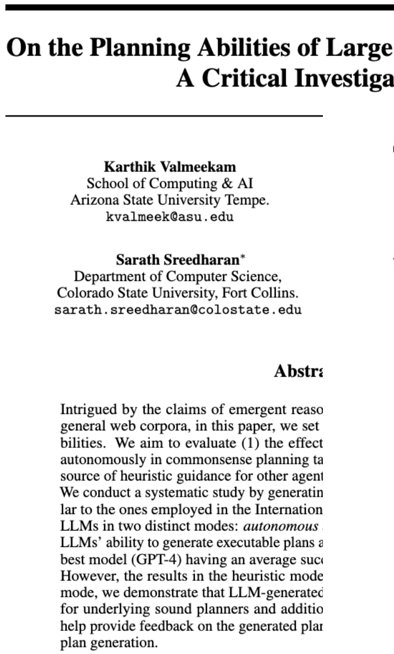


Large Language Models Still Can't Plan
(A Benchmark for LLMs on Planning and Reasoning about Change)

e.
ce,
ollins.
e.edu

Alberto Olmo*
School of Computing & AI
Arizona State University, Tempe.
aolmo@asu.edu

Subbarao Kambhampati
School of Computing & AI
Arizona State University, Tempe.
rao@asu.edu



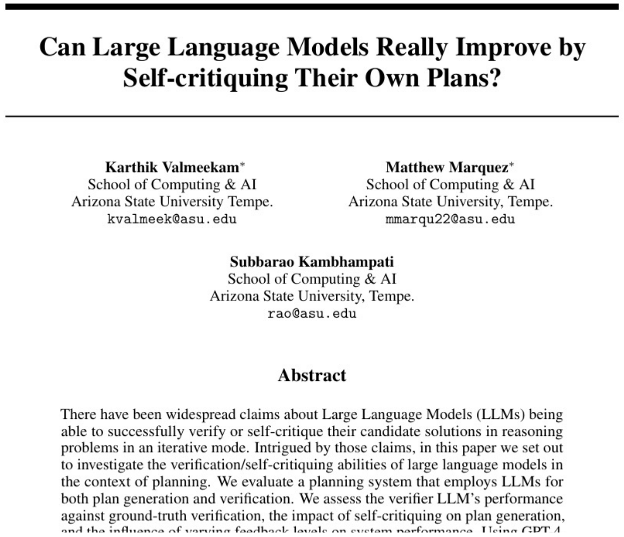
On the Planning Abilities of Large Language Models :
A Critical Investigation

Karthik Valmeekam
School of Computing & AI
Arizona State University Tempe.
kvalmeek@asu.edu

Sarath Sreedharan*
Department of Computer Science,
Colorado State University, Fort Collins.
sarath.sreedharan@colostate.edu

Abstra

Intrigued by the claims of emergent reaso
general web corpora, in this paper, we set
bilities. We aim to evaluate (1) the effect
autonomously in commonsense planning ta
source of heuristic guidance for other agent
We conduct a systematic study by generatin
lar to the ones employed in the Internation
LLMs in two distinct modes: *autonomous*
LLMs' ability to generate executable plans a
best model (GPT-4) having an average suc
However, the results in the heuristic mode
mode, we demonstrate that LLM-generated
for underlying sound planners and additio
help provide feedback on the generated pla
plan generation.



Can Large Language Models Really Improve by Self-critiquing Their Own Plans?

Karthik Valmeekam*
School of Computing & AI
Arizona State University Tempe.
kvalmeek@asu.edu

Matthew Marquez*
School of Computing & AI
Arizona State University, Tempe.
mmarqu22@asu.edu

Subbarao Kambhampati
School of Computing & AI
Arizona State University, Tempe.
rao@asu.edu

Abstract

There have been widespread claims about Large Language Models (LLMs) being able to successfully verify or self-critique their candidate solutions in reasoning problems in an iterative mode. Intrigued by those claims, in this paper we set out to investigate the verification/self-critiquing abilities of large language models in the context of planning. We evaluate a planning system that employs LLMs for both plan generation and verification. We assess the verifier LLM's performance against ground-truth verification, the impact of self-critiquing on plan generation,

Can we leverage LLMs' reasoning capabilities for Planning?

- What works for reasoning in LLMs?
- How to leverage it for planning?

Planning Domain Definition Language (PDDL)

```
(:action pickup

  :parameters (?ob)

  :precondition (and
     (handempty)
     (ontable ?ob))

  :effect (and
     (not (handempty))
     (not (ontable ?ob))
     (holding ?ob))
)
```

Precondition: This condition must be true for this action to execute

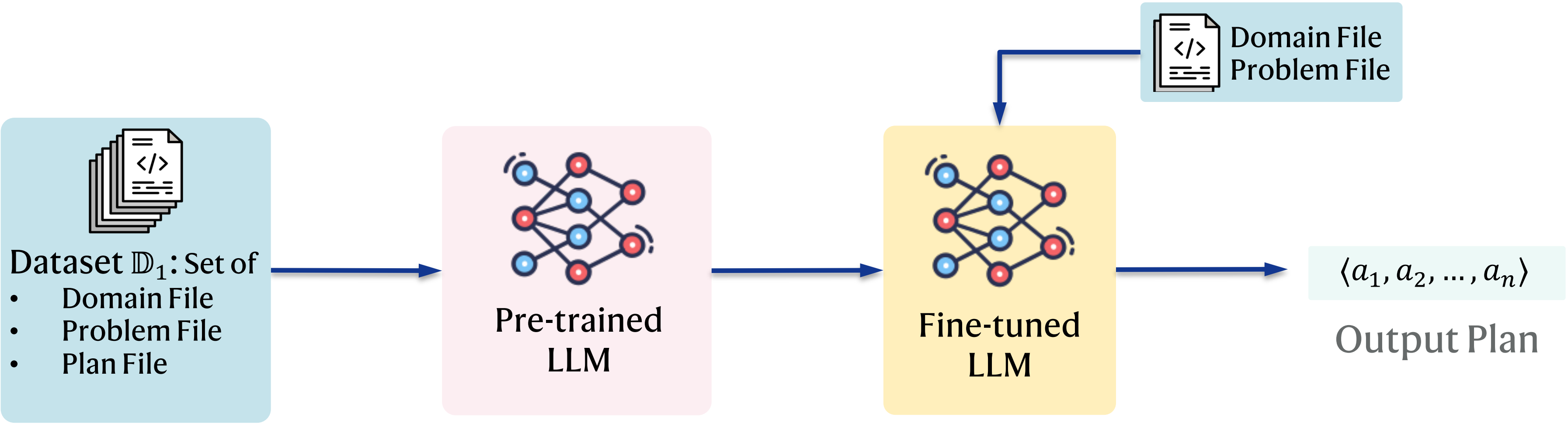
Effect: This is a set of conditions, one of which becomes true when this action is executed

What works for reasoning in LLMs?

- Finetuning
- Instruction tuning (finetuning with instructions)
- Chain-of-Thought prompting

Finetuning

Adapt a pre-trained general LLM to excel at a specific task (planning) by training on domain examples.



Dataset $\mathbb{D}_1$: Set of

- Domain File
- Problem File
- Plan File

Pre-trained LLM

Fine-tuned LLM

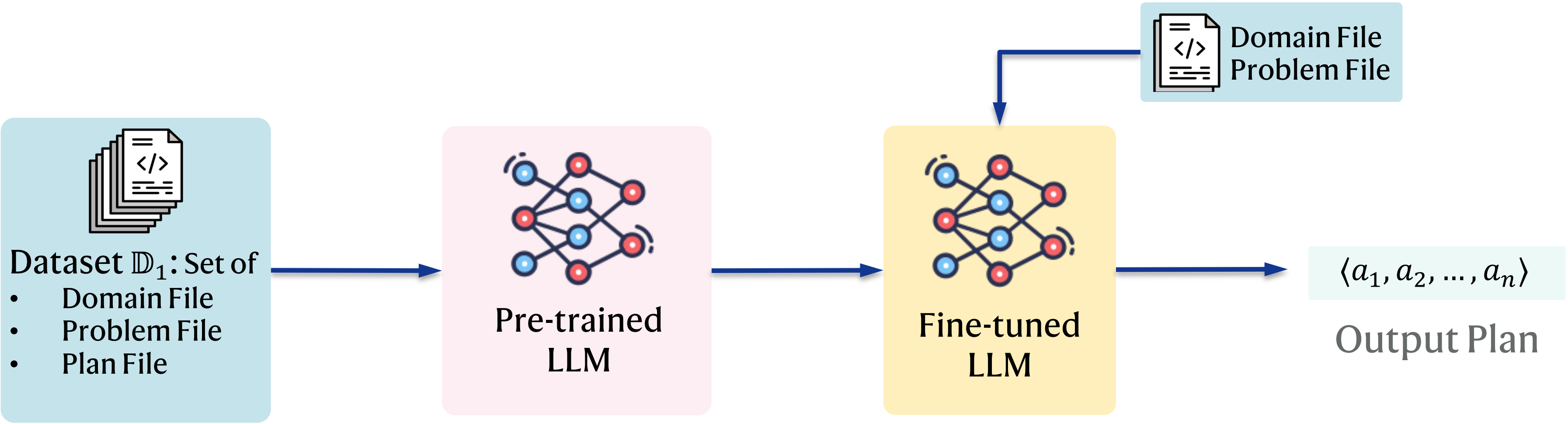
Domain File
Problem File

$\langle a_1, a_2, \dots, a_n \rangle$

Output Plan

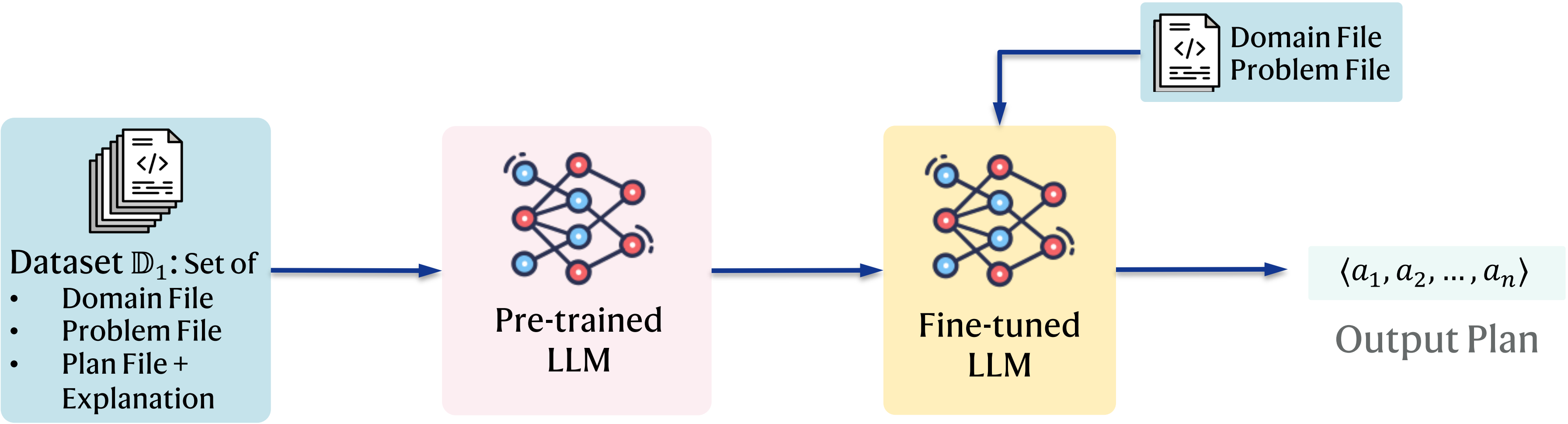
Finetuning with Negative Examples

Add some failing plans, label them as incorrect, and add them to the finetuning data.

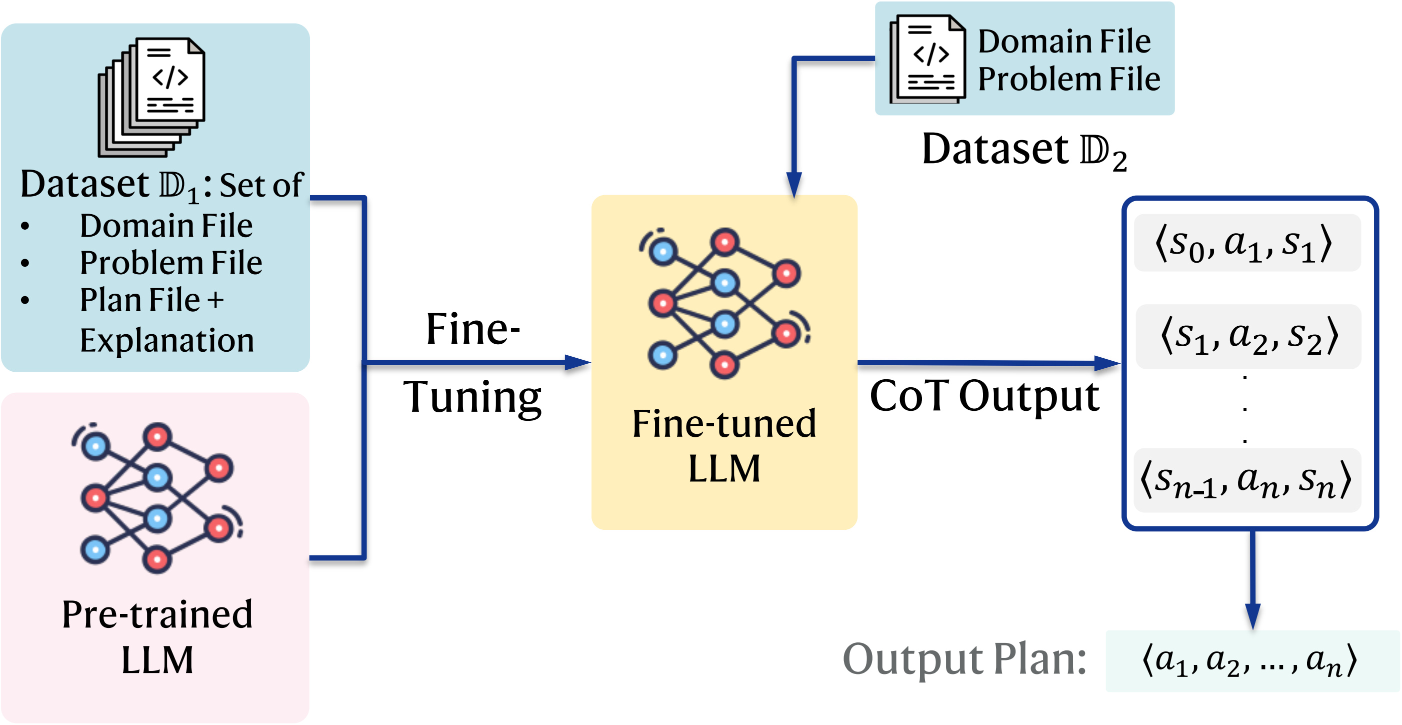


Instruction Finetuning

Add Explanations: Instructions teach the model WHAT planning means, not just PATTERNS in data. Tell it to check preconditions, apply effects, and verify goals.


Dataset $\mathbb{D}_1$: Set of
- Domain File
- Problem File
- Plan File + Explanation
Pre-trained LLM
Fine-tuned LLM
Domain File
Problem File
$\langle a_1, a_2, \dots, a_n \rangle$
Output Plan

Augment finetuned LLM with Chain-of-Thought Prompting

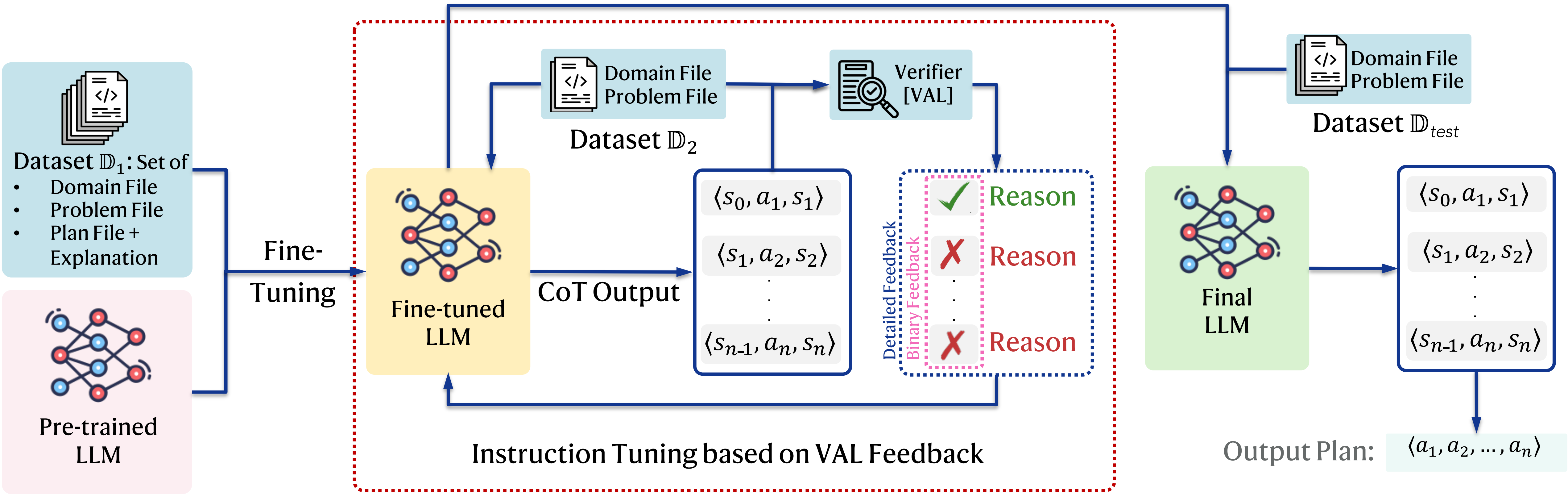


Dataset $\mathbb{D}_1$: Set of
- Domain File
- Problem File
- Plan File + Explanation

Pre-trained LLM
Fine-Tuning
Fine-tuned LLM
Domain File
Problem File
Dataset $\mathbb{D}_2$
CoT Output
$\langle s_0, a_1, s_1 \rangle$
$\langle s_1, a_2, s_2 \rangle$
$\langle s_{n-1}, a_n, s_n \rangle$
Output Plan: $\langle a_1, a_2, \ldots, a_n \rangle$

Making the model show intermediate reasoning steps for planning instead of jumping to the final answer.

PDDLInstruct


Dataset $\mathbb{D}_1$: Set of
- Domain File
- Problem File
- Plan File + Explanation
Pre-trained LLM
Fine-Tuning
Fine-tuned LLM
Domain File
Problem File
Dataset $\mathbb{D}_2$
Verifier [VAL]
CoT Output
$\langle s_0, a_1, s_1 \rangle$
$\langle s_1, a_2, s_2 \rangle$
$\langle s_{n\text{-}1}, a_n, s_n \rangle$
Detailed Feedback
Binary Feedback
✓ Reason
✗ Reason
✗ Reason
Instruction Tuning based on VAL Feedback
Domain File
Problem File
Dataset $\mathbb{D}_{test}$
Final LLM
$\langle s_0, a_1, s_1 \rangle$
$\langle s_1, a_2, s_2 \rangle$
$\langle s_{n\text{-}1}, a_n, s_n \rangle$
Output Plan: $\langle a_1, a_2, \ldots, a_n \rangle$

Reasoning Chain Optimization

optimize the model parameters θ_t to
improve the generation of high-quality reasoning chains

$$\theta_t^r = \theta_t - \delta_1 \nabla_{\theta_t} \mathcal{L}_{reasoning} (\theta_t, \mathbb{D}_{reasoning}^t)$$

$\{(s_{i-1}, a_i, s_i, f_i)$: ∀ steps in CoT plans generated at iteration $t\}$

loss function that measures the quality of the generated reasoning chains

Reasoning Chain Optimization: $\theta_t^r = \theta_t - \delta_1 \nabla_{\theta_t} \mathcal{L}_{reasoning}(.)$

This objective encourages the model to produce step-by-step reasoning that correctly:

1. checks all necessary preconditions before applying actions;
2. tracks state changes resulting from action effects;
3. verifies that invariants are maintained throughout the plan; and
4. detects logical inconsistencies in proposed plans.

Reasoning Chain Optimization

$$\mathcal{L}_{reasoning}\left(\theta_t, \mathbb{D}^t_{reasoning}\right) =$$

$$\frac{1}{\left|\mathbb{D}^t_{reasoning}\right|} \sum_{(s_{i-1}, a_i, s_i, f_i) \in \mathbb{D}^t_{reasoning}} d\left(s_i, s_i^{expected}\right) + \lambda_{feedback}\, \mathcal{L}_{feedback}$$

$$\mathcal{L}_{feedback} = \begin{cases} 0 & \text{if action } a_i \text{ is valid} \\ \alpha_{pre} & \text{if precondition violation detected} \\ \alpha_{eff} & \text{if incorrect effect applied} \\ \alpha_{goal} & \text{if goal not achieved} \end{cases}$$

End-Task (Final) Performance Optimization

optimize from the reasoning-improved parameters θ_t^r to enhance overall planning

$$\theta_{t+1} = \theta_t^r - \delta_2 \nabla_{\theta_t^r} \mathcal{L}_{final}(\theta_t^r, \mathbb{D}_{final}^t)$$

$\{(d_j, p_j, \pi_i^t, v_j^t)$: ∀ problems j at iteration $t\}$

loss function that measures measures how well the final outputs match the expected answers in the training data

End-Task Performance Optimization: $\theta_{t+1} = \theta_t^r - \delta_2 \nabla_{\theta_t^r} \mathcal{L}_{final}(.)$

This objective ensures that improvements in logical reasoning translate to practical planning capability of producing accurate plans.

Empirical Evaluation: Objectives

RQ1: Does logical CoT instruction tuning improve plan validity compared to standard approaches?

RQ2: How does the quality of feedback (binary vs. detailed) affect planning performance?

RQ3: How well does the approach generalize across different planning domains?

Empirical Evaluation: Dataset and Models

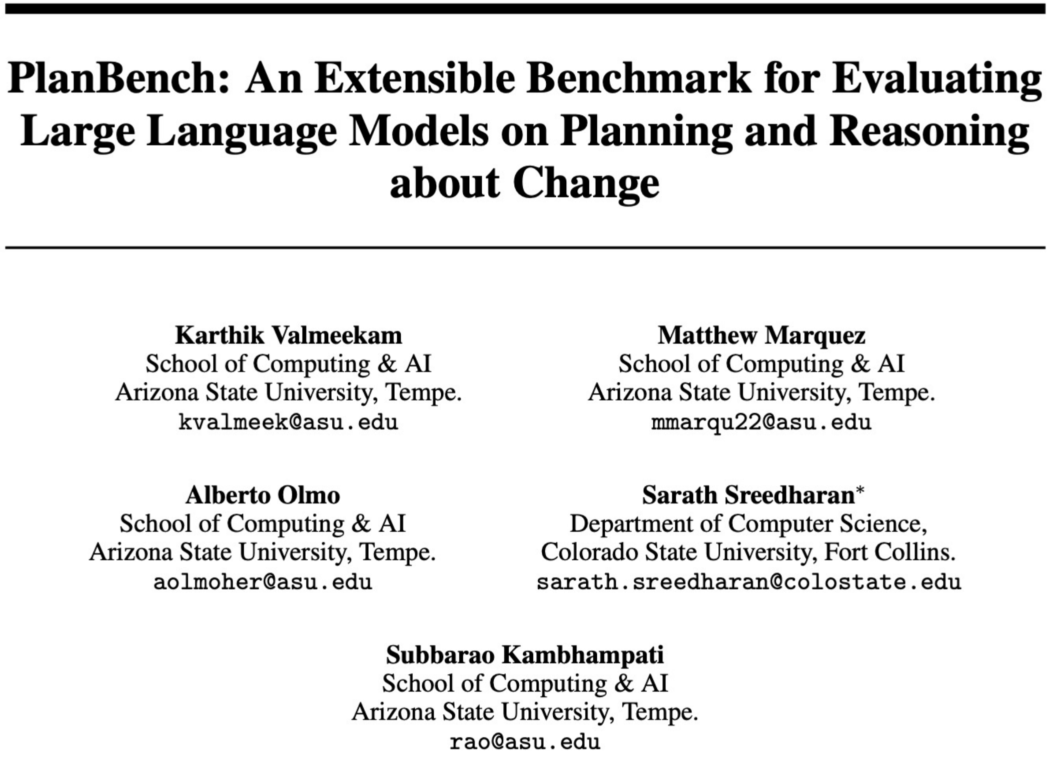
Three Domains:

- Blockworld
- Logistics
- Mystery Blocksworld

Three Models:

- Llama-3-8B
- GPT-4
- Gemma-3-270M

Benchmark



PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change

Karthik Valmeekam
School of Computing & AI
Arizona State University, Tempe.
kvalmeek@asu.edu

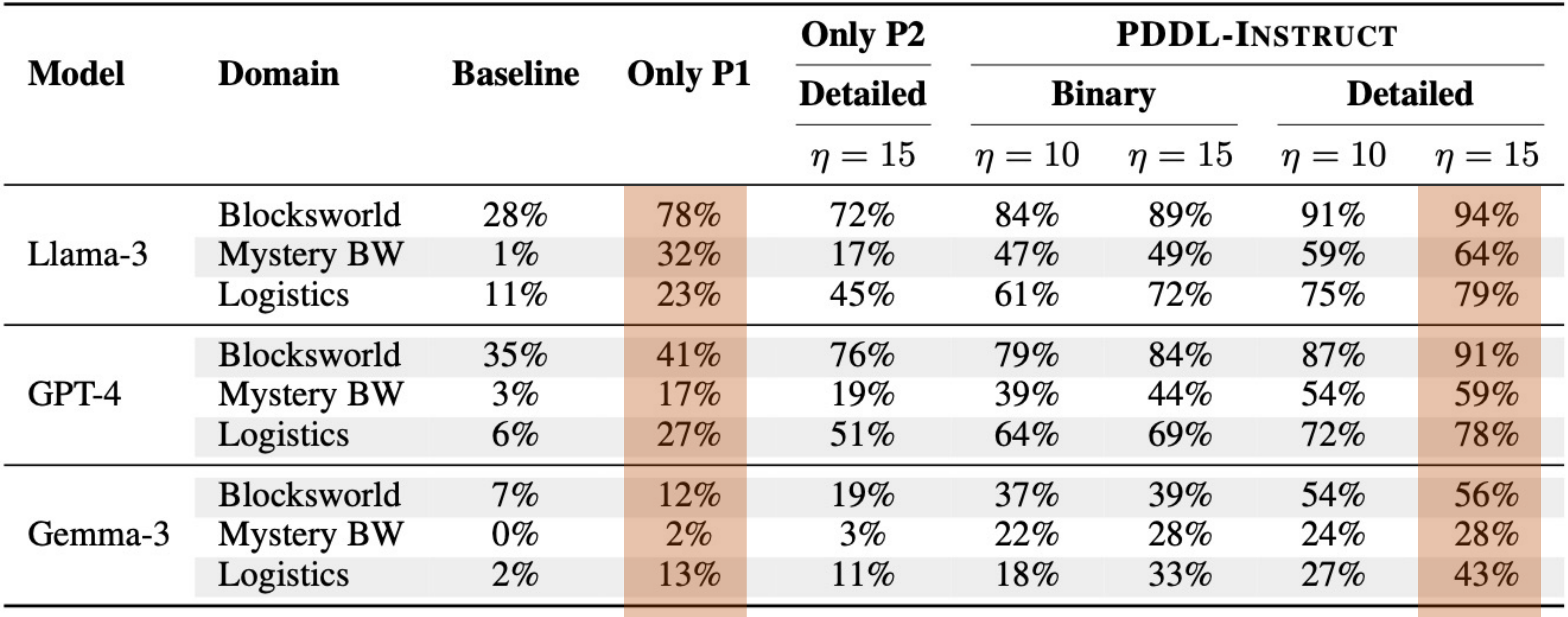
Matthew Marquez
School of Computing & AI
Arizona State University, Tempe.
mmarqu22@asu.edu

Alberto Olmo
School of Computing & AI
Arizona State University, Tempe.
aolmoher@asu.edu

Sarath Sreedharan*
Department of Computer Science,
Colorado State University, Fort Collins.
sarath.sreedharan@colostate.edu

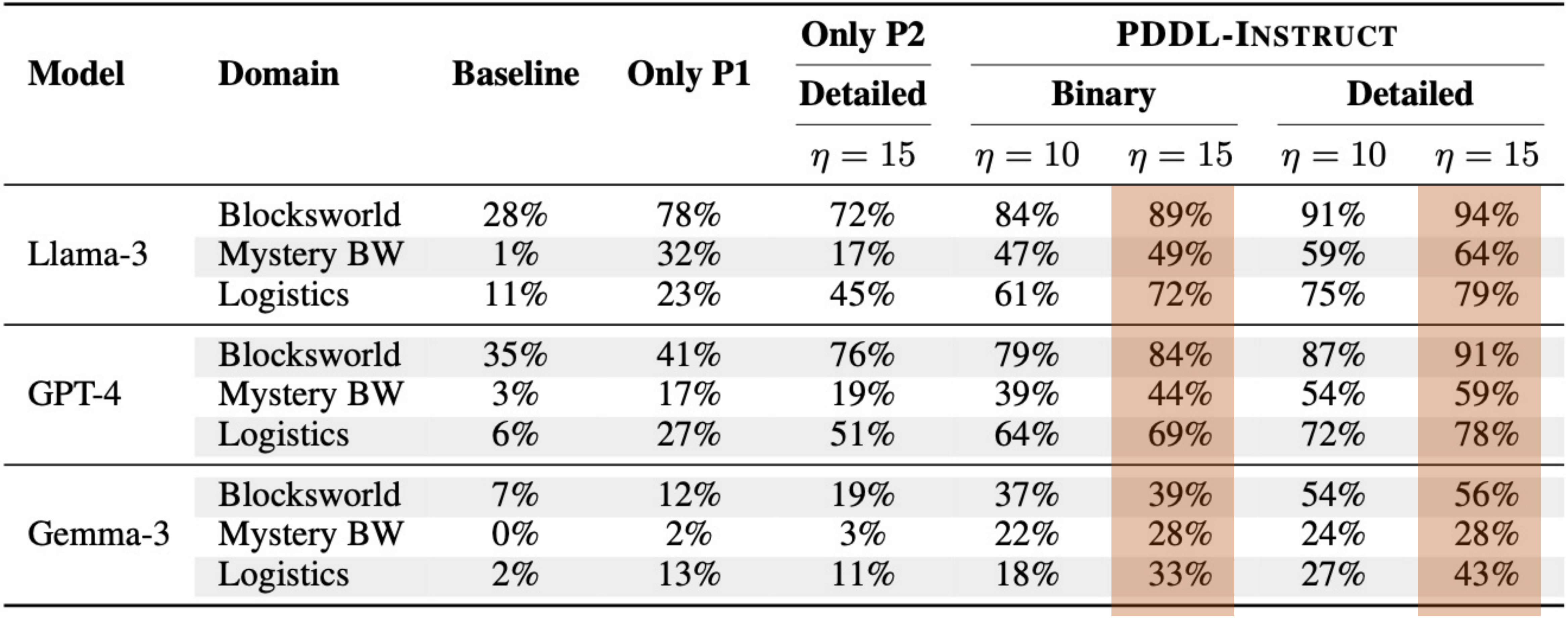
Subbarao Kambhampati
School of Computing & AI
Arizona State University, Tempe.
rao@asu.edu

Logical CoT instruction tuning improves Plan Validity



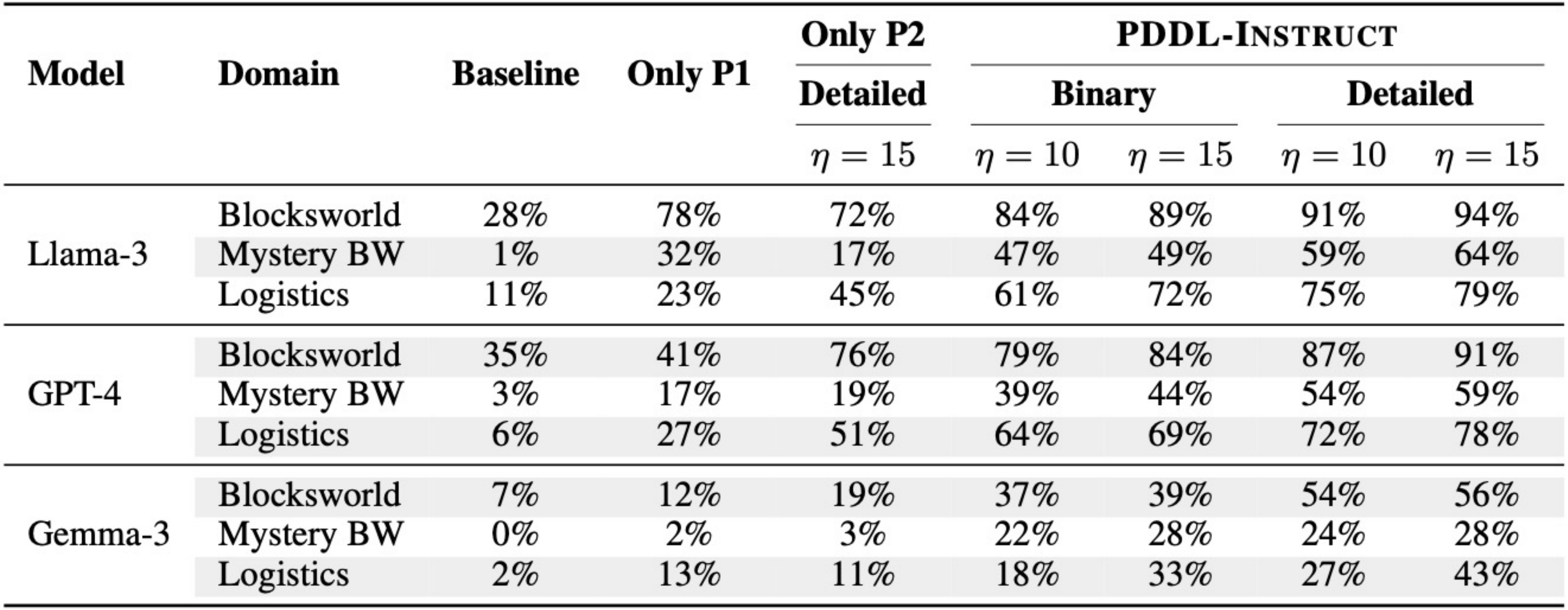
Model	Domain	Baseline	Only P1	Only P2	PDDL-INSTRUCT			
				Detailed	Binary		Detailed	
				$\eta = 15$	$\eta = 10$	$\eta = 15$	$\eta = 10$	$\eta = 15$
Llama-3	Blocksworld	28%	78%	72%	84%	89%	91%	94%
	Mystery BW	1%	32%	17%	47%	49%	59%	64%
	Logistics	11%	23%	45%	61%	72%	75%	79%
GPT-4	Blocksworld	35%	41%	76%	79%	84%	87%	91%
	Mystery BW	3%	17%	19%	39%	44%	54%	59%
	Logistics	6%	27%	51%	64%	69%	72%	78%
Gemma-3	Blocksworld	7%	12%	19%	37%	39%	54%	56%
	Mystery BW	0%	2%	3%	22%	28%	24%	28%
	Logistics	2%	13%	11%	18%	33%	27%	43%

Detailed feedback is better than Binary Feedback



Model	Domain	Baseline	Only P1	Only P2	PDDL-INSTRUCT			
				Detailed	Binary		Detailed	
				$\eta = 15$	$\eta = 10$	$\eta = 15$	$\eta = 10$	$\eta = 15$
Llama-3	Blocksworld	28%	78%	72%	84%	89%	91%	94%
	Mystery BW	1%	32%	17%	47%	49%	59%	64%
	Logistics	11%	23%	45%	61%	72%	75%	79%
GPT-4	Blocksworld	35%	41%	76%	79%	84%	87%	91%
	Mystery BW	3%	17%	19%	39%	44%	54%	59%
	Logistics	6%	27%	51%	64%	69%	72%	78%
Gemma-3	Blocksworld	7%	12%	19%	37%	39%	54%	56%
	Mystery BW	0%	2%	3%	22%	28%	24%	28%
	Logistics	2%	13%	11%	18%	33%	27%	43%

PDDLInstruct’s improved performance generalizes across domains



Model	Domain	Baseline	Only P1	Only P2	PDDL-Instruct			
				Detailed	Binary		Detailed	
				$\eta = 15$	$\eta = 10$	$\eta = 15$	$\eta = 10$	$\eta = 15$
Llama-3	Blocksworld	28%	78%	72%	84%	89%	91%	94%
	Mystery BW	1%	32%	17%	47%	49%	59%	64%
	Logistics	11%	23%	45%	61%	72%	75%	79%
GPT-4	Blocksworld	35%	41%	76%	79%	84%	87%	91%
	Mystery BW	3%	17%	19%	39%	44%	54%	59%
	Logistics	6%	27%	51%	64%	69%	72%	78%
Gemma-3	Blocksworld	7%	12%	19%	37%	39%	54%	56%
	Mystery BW	0%	2%	3%	22%	28%	24%	28%
	Logistics	2%	13%	11%	18%	33%	27%	43%

Conclusion

- Novel framework leveraging CoT-based instruction tuning to significantly enhance LLM-based planning.
- Performance of CoT-based instruction tuning depends on the feedback type.

Limitations:

- Optimizing instruction tuning data.
- Finegrained analysis of planning performance.
- Comparison with SoTA symbolic planners.
- Extending domain coverage.

pulkitverma.net | pulkitv@csail.mit.edu