Lecture 33

CS 6260 Automated Planning and Learning

LLMs for Planning

Department of Computer Science and Engineering

Indian Institute of Technology Madras

Some material based on LLM Planning Tutorial br Subbarao Kambhampati.

What are LLMs?

- Text is a long sequence of words (including spaces, punctuations)

- An n-gram model of language learns to predict n-th word given the preceding n-1 words
  - $Pr(W_n | W_1 ... W_{n-1})$
  - Unigram predicts each word independently (no preceding context)
  - Bigram predicts each word given the previous word
  - A 3001-gram model learns to predict the next word given the previous 3000 words

- The power of an n-gram model depends on
  - How much text it trains on
  - How big is the n (context) and
  - How high-capacity is the function learning $Pr(W_n | W_1 ... W_{n-1})$

- ChatGPT trains on ~600 gigabytes of text on the Web
  - It learns a very high capacity function that has 175 billion parameters

IIT Madras is the best engineering college in India

- LLM uses its current function to guess the next word
  - IIT Kharagpur

- Predicted: Kharagpur; Correct: Madras

- Error = {Madras - Kharagpur}
  - To the LLMs, all vocabulary tokens are just vectors in some high-dimensional embedding space; so the difference is well defined as the vector difference

- Propagate this error back through the function, and change the parameters so the error is reduced.

LLMs are good at Reasoning

LLMs are bad at Planning

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 demonstratio
Moreover, the emergent reasoning patterns exhibited by these large-scale
can be systematically used to guide and enhance the reasoning capabilitie
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-R1
rally developed diverse and sophisticated reasoning be
solve reasoning problems, the model exhibits a tendency
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
ounding
M 2 scores up
y over 19%,
dMCQA,
uman
M 2 answers
led-PaLM 2
ssor across
s designed
l-world
to generalist
ere still
PaLM 2 to
tential in
censing Examina-

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.

Standard Prompting　Chain-of-Thought Prompting

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:

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

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

Subbarao Kambhampati[1] Karthik Valmeekam[1] Lin Guan[1] Mu
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 con
seem to ring true,
are best seen as a g
(see Figure 1). Ev
a system that takes
cannot possibly be
Not surprisingly, i
formance of LLMs

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

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 plan
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?

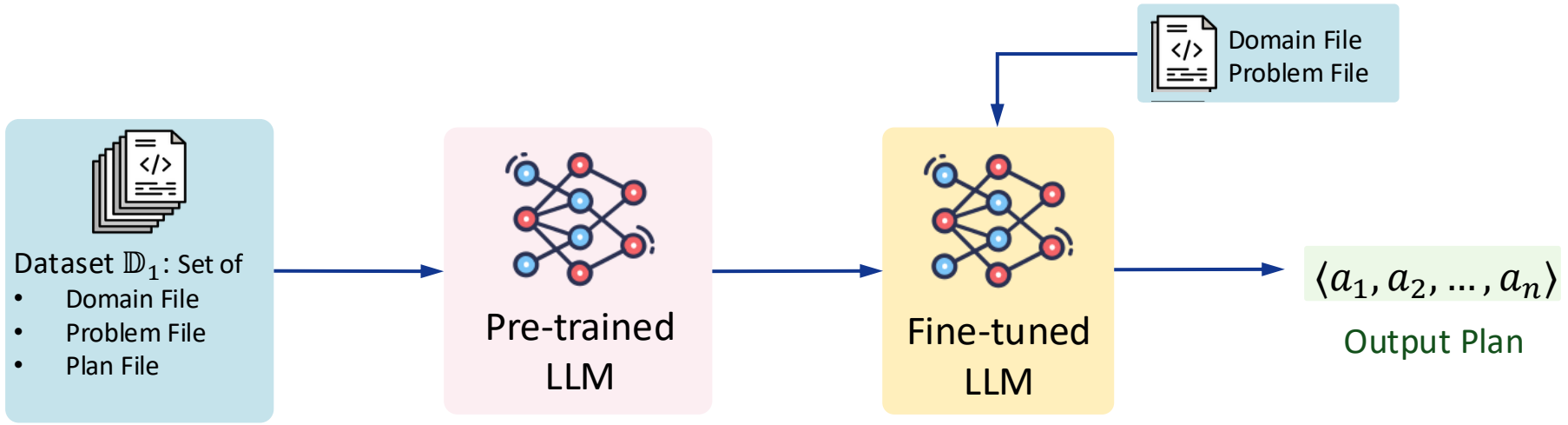
- What works for reasoning in LLMs?
- How to leverage it for planning?

What works for reasoning in LLMs?

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

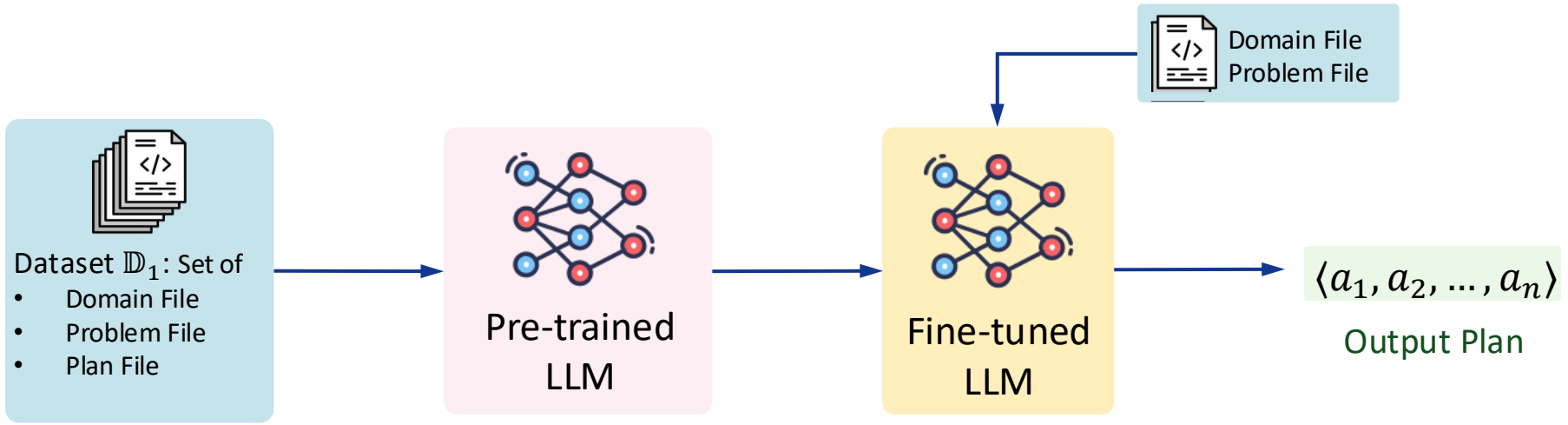
Finetuning

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



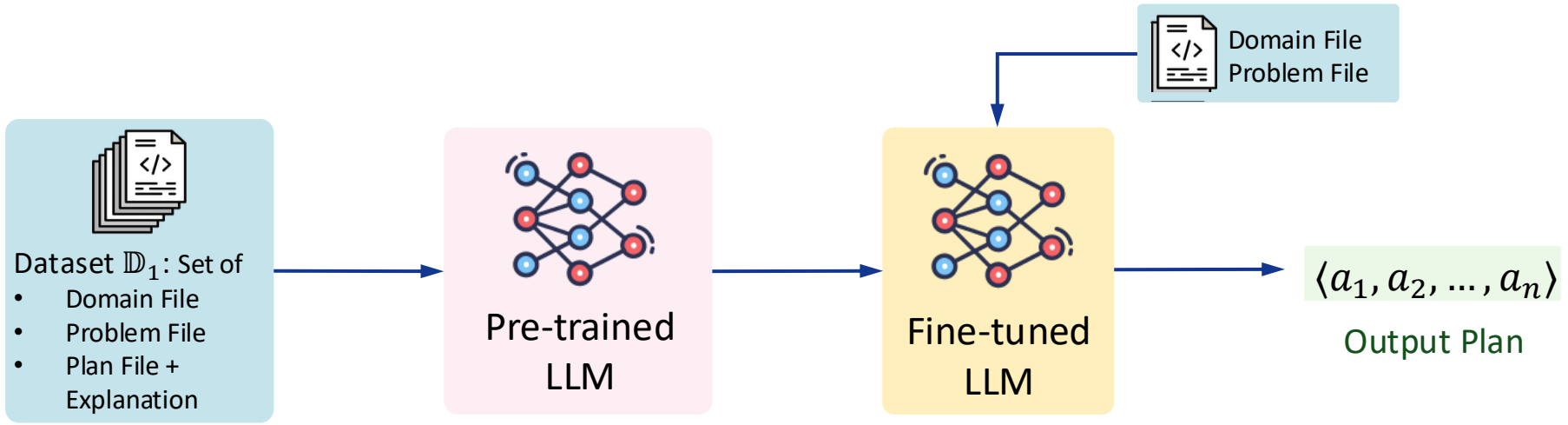
Finetuning with Negative Examples

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

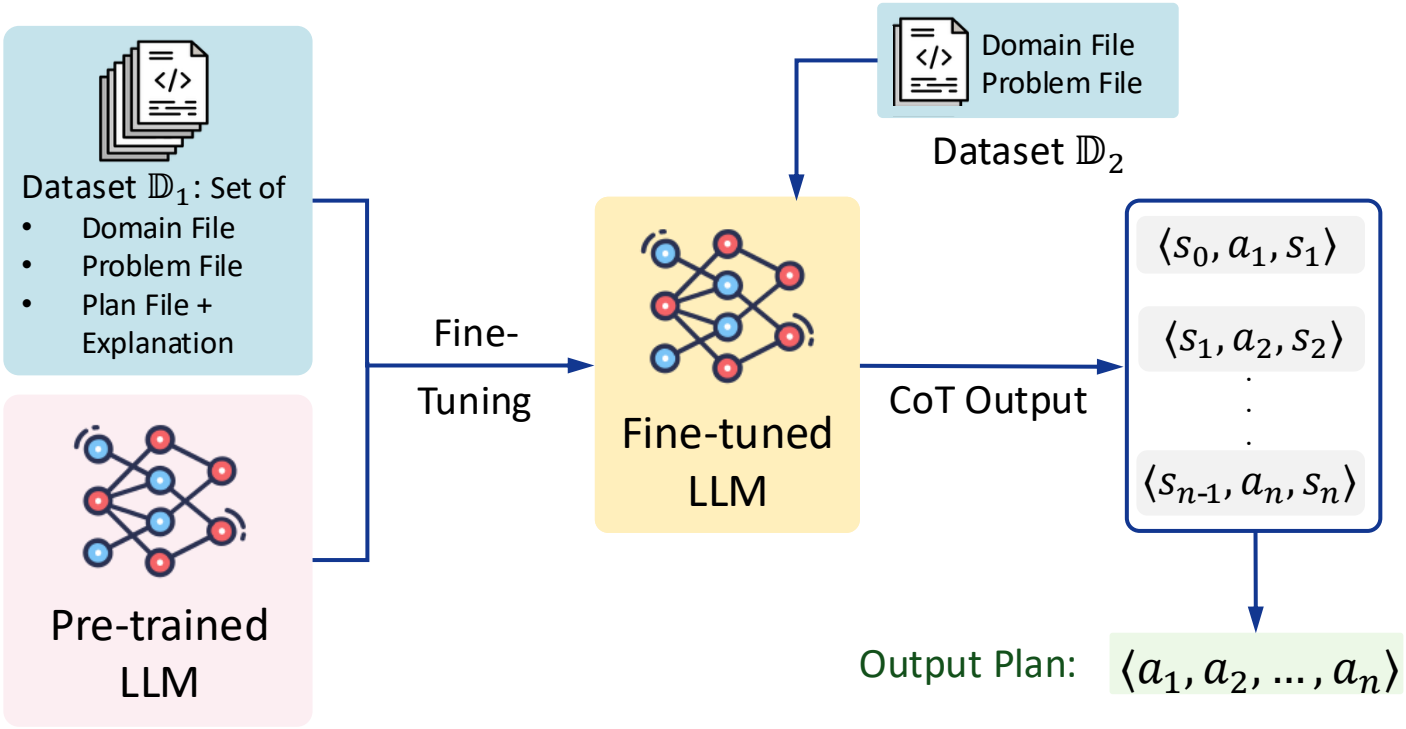


Instruction Finetuning

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

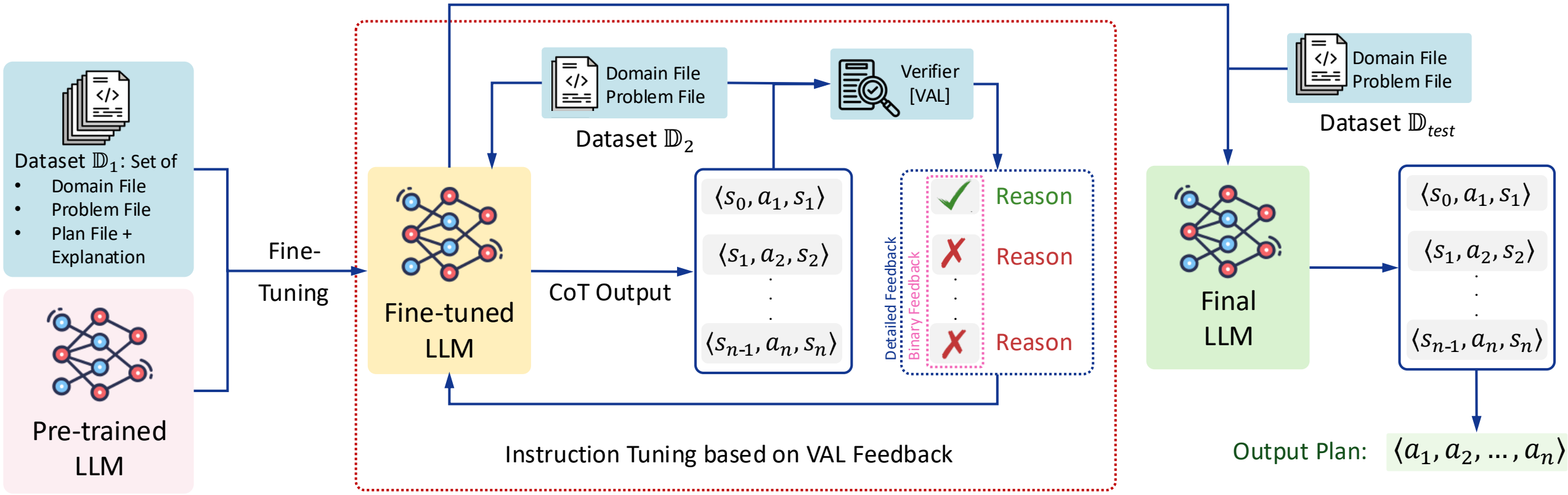


Augment finetuned LLM with Chain-of-Thought Prompting



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

PDDLInstruct



Verma et al., PDDL-Instruct: Enhancing Symbolic Planning Capabilities in LLMs through Logical Chain-of-Thought Instruction Tuning. In ICAPS 2025 LM4Plan Workshop.

How else can we leverage LLMs for Planning

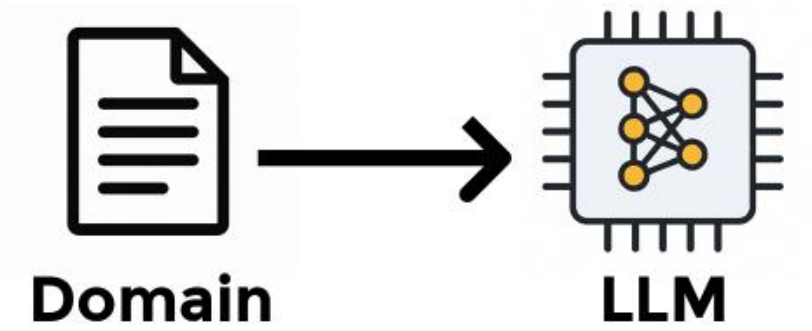
Learn heuristics!!

How else can we leverage LLMs for Planning

Learn heuristics!!

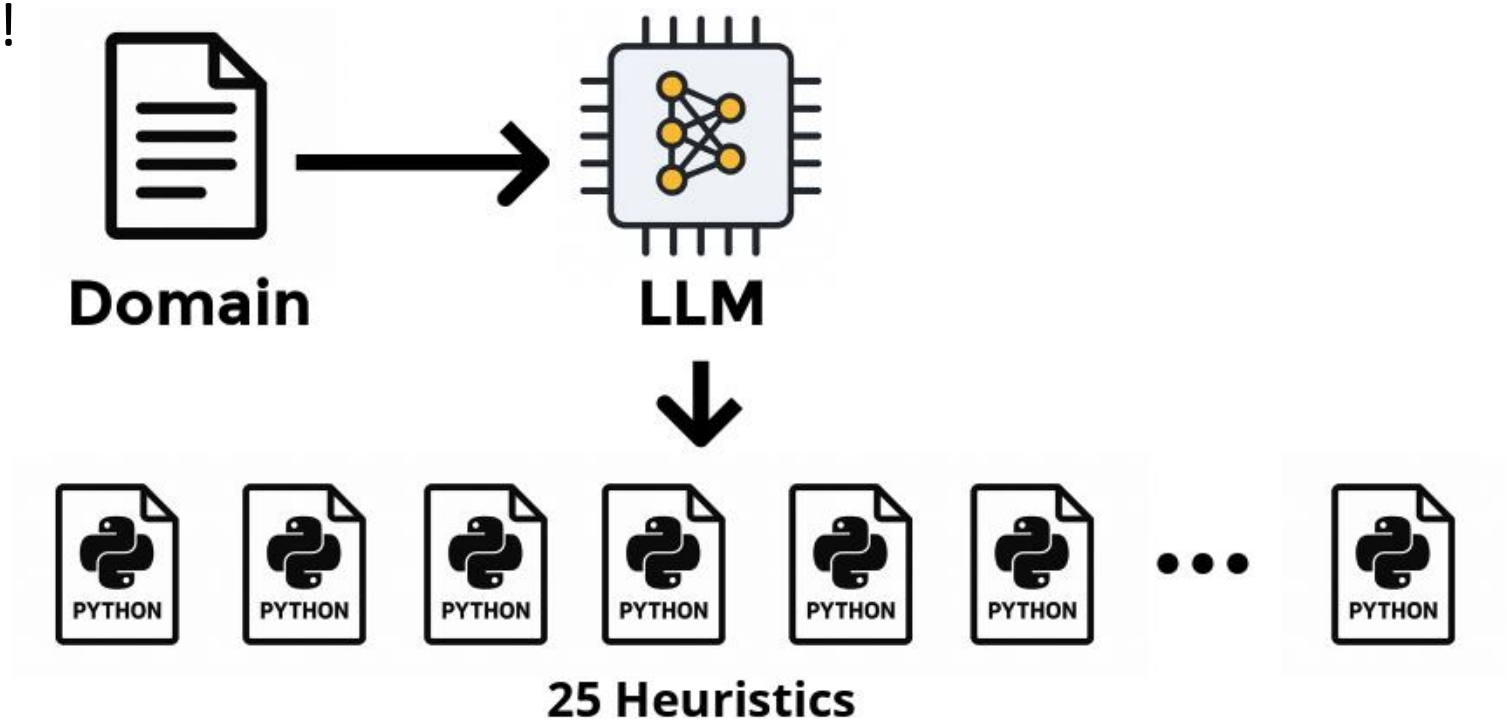
How else can we leverage LLMs for Planning

Learn heuristics!!



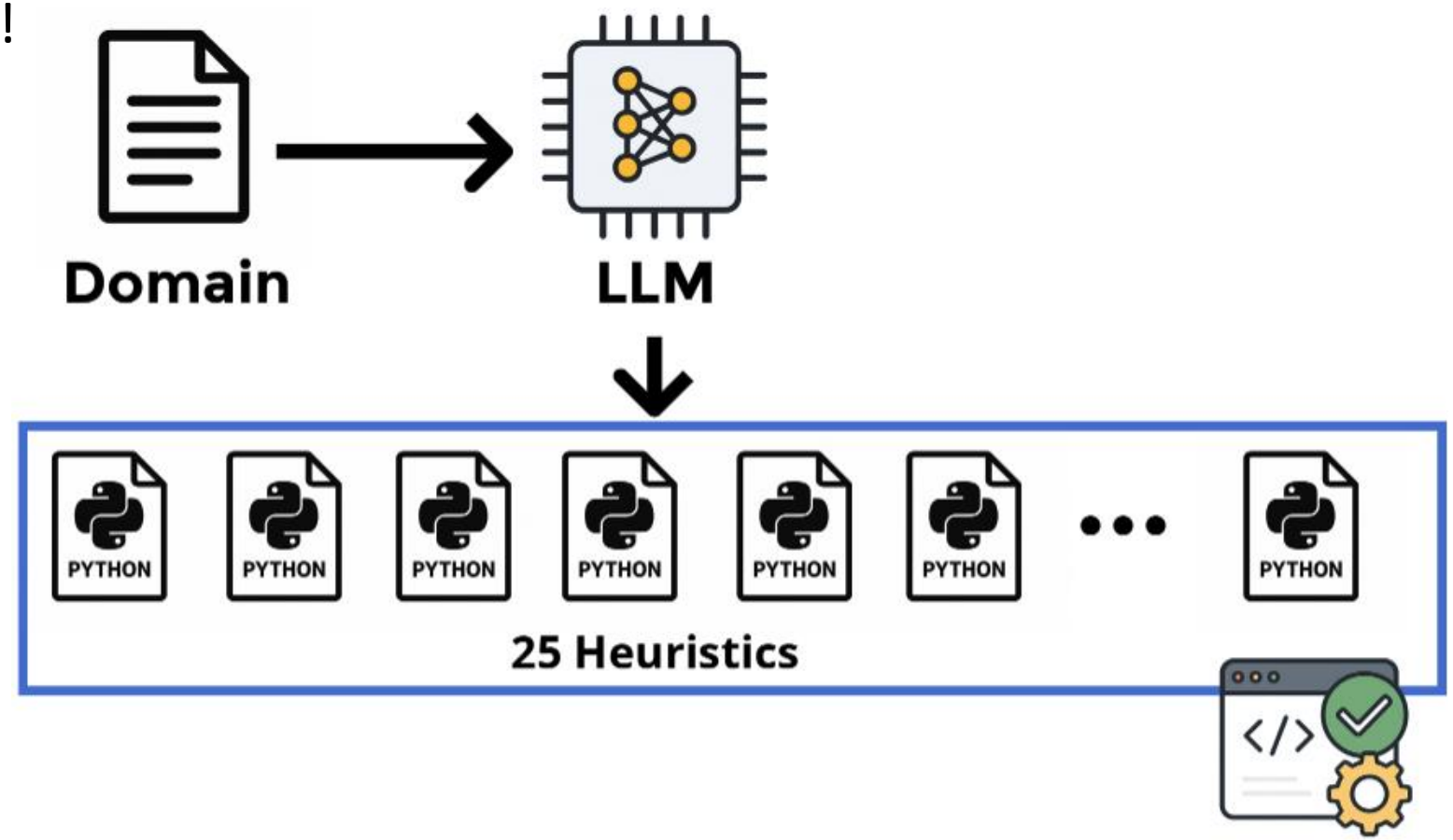
How else can we leverage LLMs for Planning

Learn heuristics!!



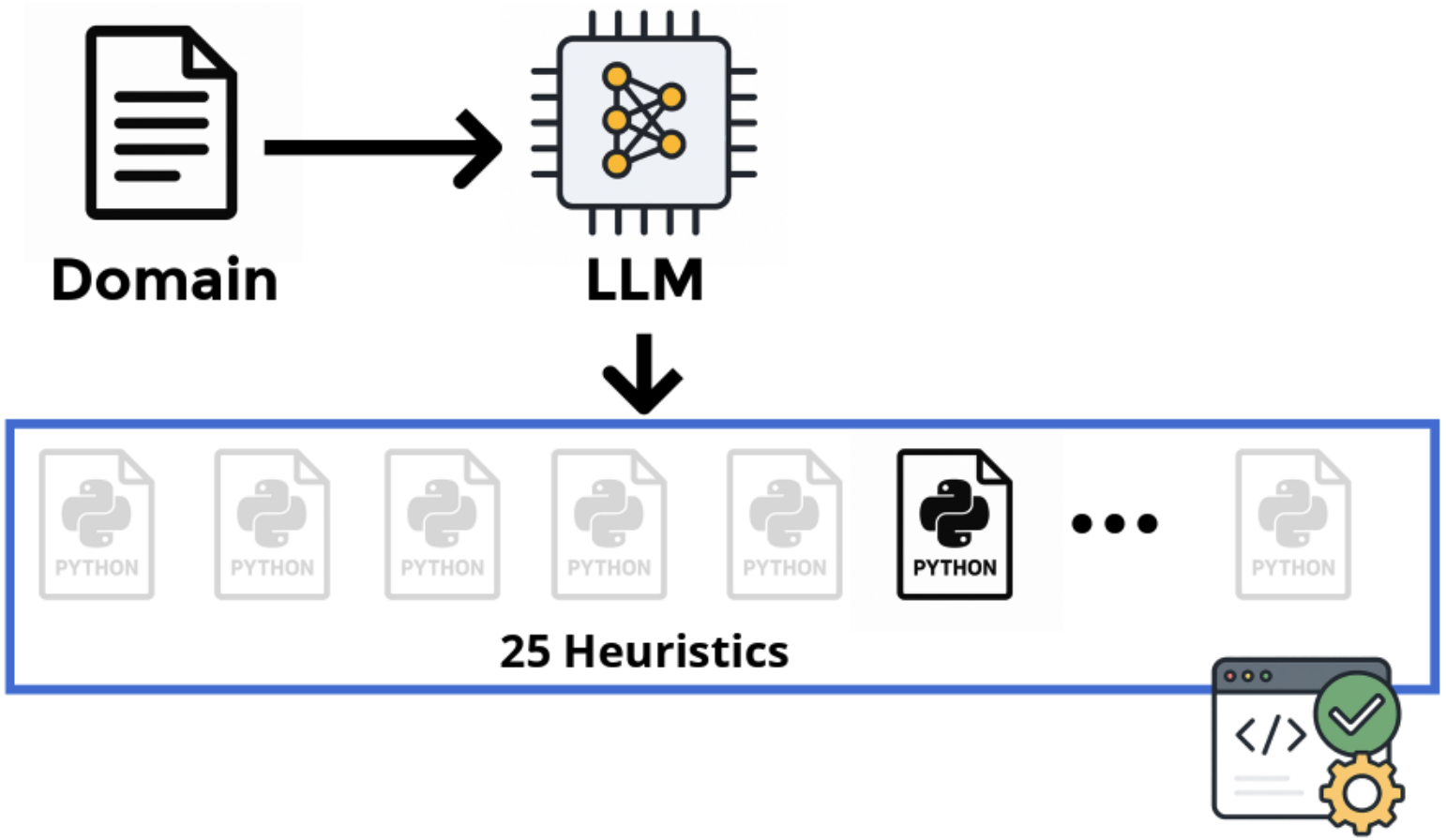
How else can we leverage LLMs for Planning

Learn heuristics!!



How else can we leverage LLMs for Planning

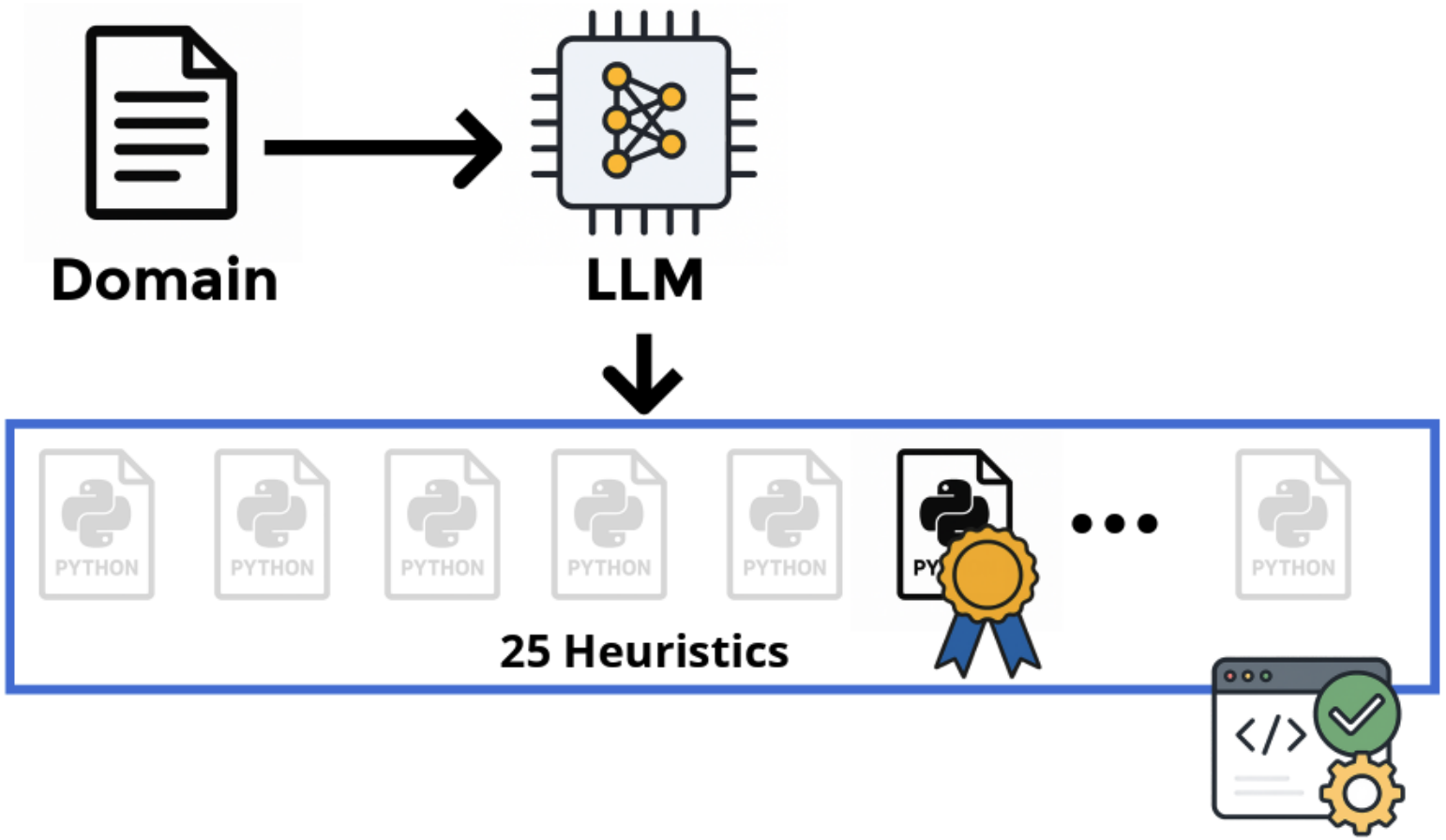
Learn heuristics!!



Augusto B Corrêa, André G Pereira, Jendrik Seipp. Classical Planning with LLM-Generated Heuristics: Challenging the State of the Art with Python Code. In Proc. NeurIPS 2025

How else can we leverage LLMs for Planning

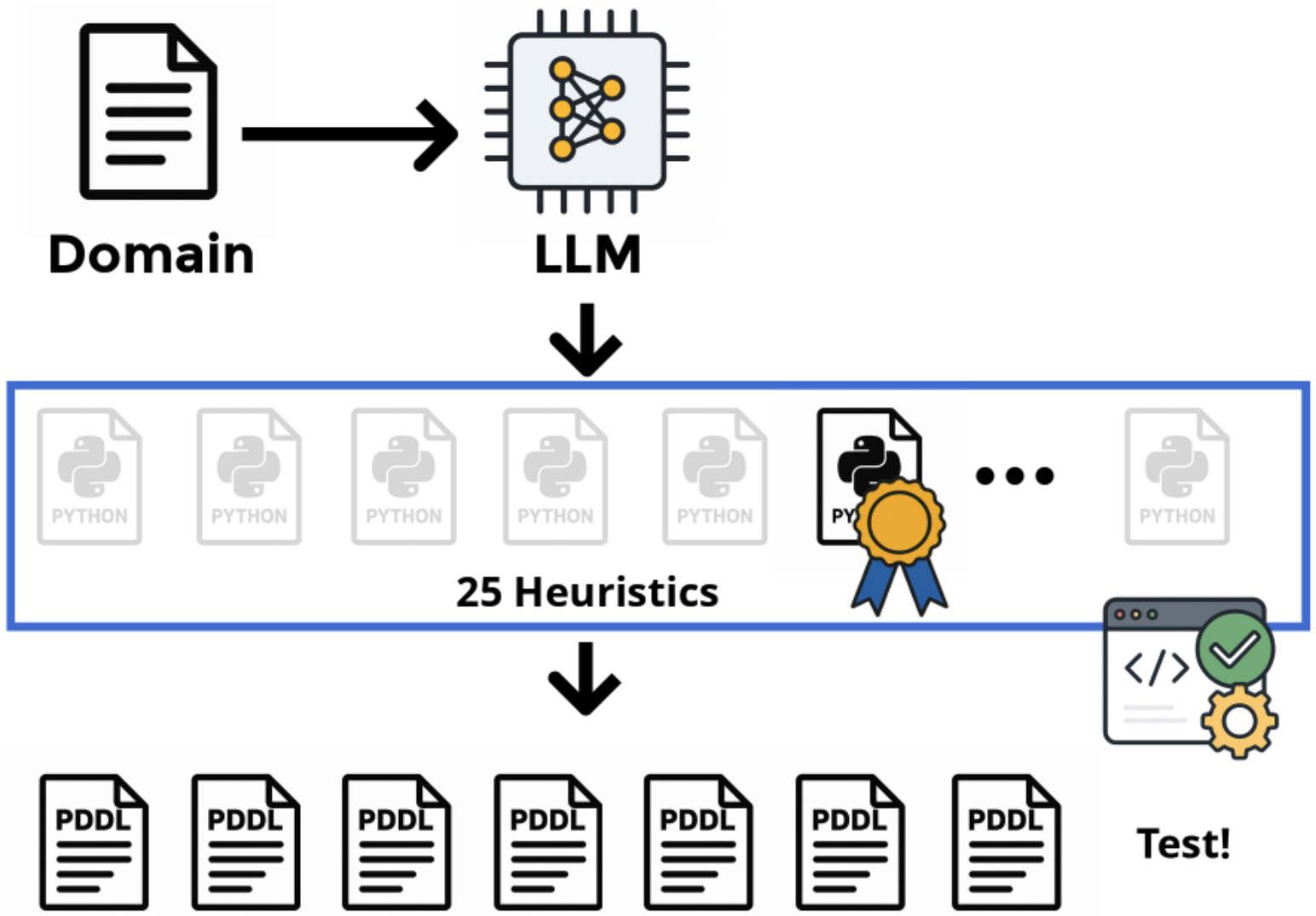
Learn heuristics!!



Augusto B Corrêa, André G Pereira, Jendrik Seipp. Classical Planning with LLM-Generated Heuristics: Challenging the State of the Art with Python Code. In Proc. NeurIPS 2025

How else can we leverage LLMs for Planning

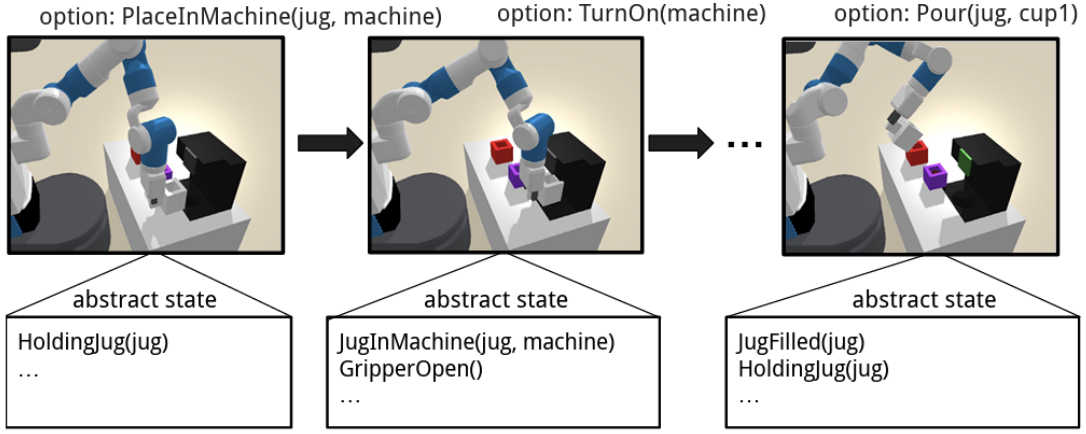
Learn heuristics!!



Augusto B Corrêa, André G Pereira, Jendrik Seipp. Classical Planning with LLM-Generated Heuristics: Challenging the State of the Art with Python Code. In Proc. NeurIPS 2025

Learning End-to-End World Models

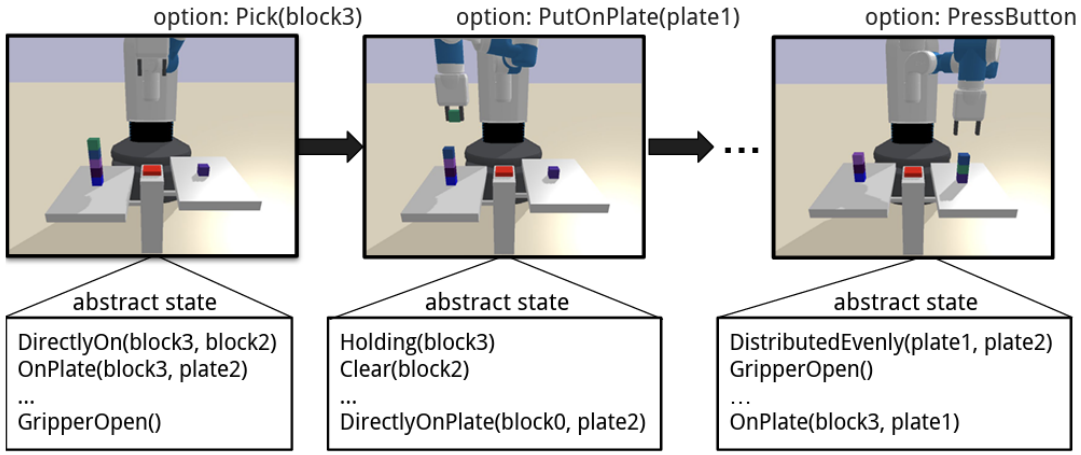
(A) making and pouring coffee into a pair of cups



example learned predicate+planning operator

```
;; PDDL operator definition, TurnOn option.
(def-op (?x0:coffee_machine ?x1:jug ?x2:robot)
 :option (TurnOn ?x2:robot ?x0:coffee_machine)
 :precondition (JugInMachine ?x1 ?x0)
 :postcondition (JugFilled ?x1:jug))
# Neuro-Symbolic Predicate.
def JugInMachine(state, objects):
 jug, machine = objects
 # If the jug is held, it cannot be in the machine.
 if Holding(state,[state.objects(robot_type)[0],jug]):
   return False
 # Crop to focus on jug and coffee machine.
 attention_img = state.crop_to_objects([jug,machine])
 return state.query_VLM(
 f"{jug.id_name} is placed inside {machine.id_name}.",
   attention_img)
```

(B) using a balance beam to activate a machine that requires balanced platters



example learned predicate+planning operator

```
;; PDDL operator definition, PressButton option.
(def-op (?x0:machine ?x1:plate ?x2:plate ?x3:robot)
 :option (PressButton ?x3:robot ?x1:plate ?x2:plate)
 :precondition (DistributedEvenly ?x2 ?x1)
 :postcondition (MachineOn ?x0:machine))
# Neuro-Symbolic Predicate.
def DistributedEvenly(atoms, objects):
 plate1, plate2 = objects
 if plate1 == plate2: return False
 count1, count2 = 0, 0
 for atom in atoms:
   if atom.predicate == OnPlate:
     if atom.objects[1] == plate1: count1 += 1
     elif atom.objects[1] == plate2: count2 += 1
 return count1 == count2
```

Liang et al.. VisualPredicator: Learning Abstract World Models with Neuro-Symbolic Predicates for Robot Planning. In Proc. ICLR 2025