ExoPredicator: Learning Abstract Models of Dynamic Worlds for Robot Planning

Yichao Liang[1 2], Dat Nguyen[2], Cambridge Yang[2], Tianyang Li[2], Joshua B. Tenenbaum[5], Carl Rasmussen[1], Adrian Weller[1 6], Zenna Tavares[2], Tom Silver[4]*, Kevin Ellis[3]*

[1]University of Cambridge, [2]Basis, [3]Cornell University, [4]Princeton University, [5]Massachusetts Institute of Technology, [6]The Alan Turing Institute, *equal-advising

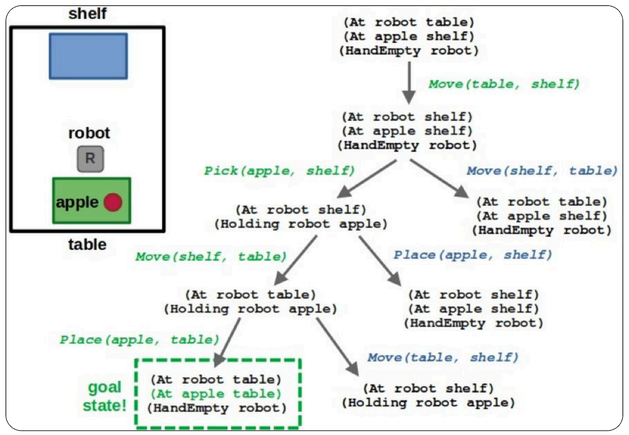
1. Motivation

Goal: Efficient learning for long-horizon planning in *dynamic* and novel environments

Example:
- Learn to use a new oven

Desiderata:
- Efficient to learn
- Generalizable to related tasks
- Interpretable knowledge

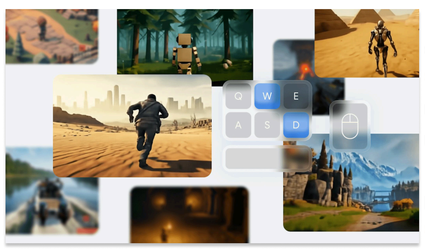
2. Prior Work -- Task and Motion Planning



Example search tree

💪 Effective Long horizon planning + Generalizable to different tasks (mature planning algorithms)
☹️ Hard to hand-code models + unreliable when model is wrong (predicates, operators, low-level simulators)

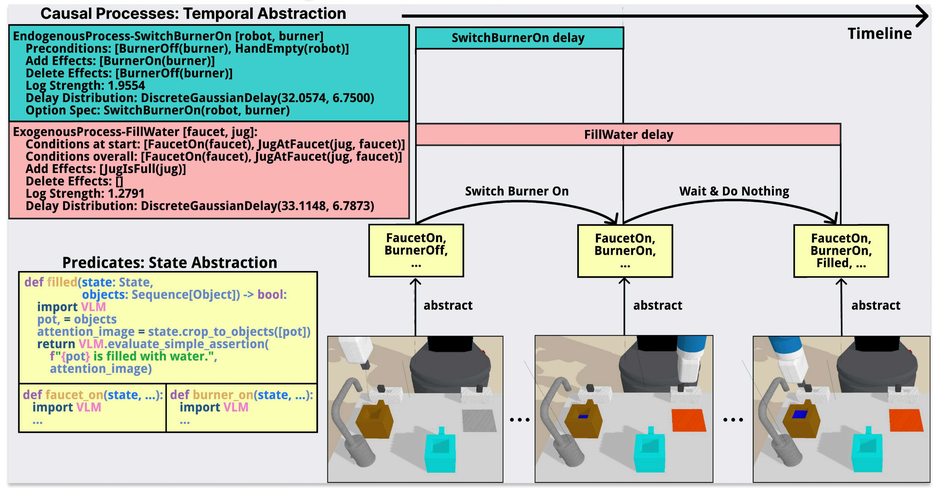
3. Prior Work -- Low Level World Models



Examples: Genie, JEPA, cosmos, Dreamer

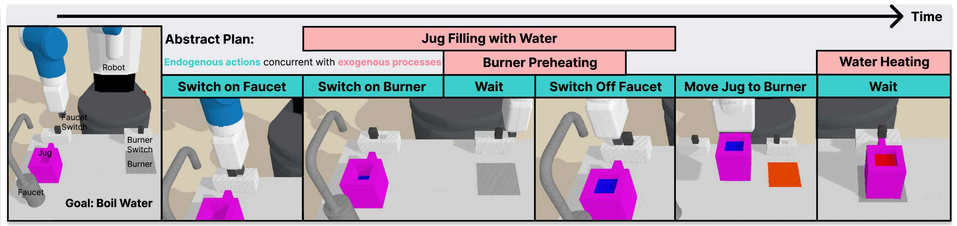
💪 Applicable to diverse domains
☹️ Expensive and unreliable to adapt to unseen domains (stemmed from neural nets)
☹️ Hard to plan with (no temporal abstraction and fine-grained controllability).

4. Abstracting State, Time, And Causal Processes



- Raw input maps to a state abstraction via predicates: short Python programs detecting binary features.
- Temporal dynamics of abstract states are governed by *causal processes*: either *endogenous processes* (actions), or *exogenous processes* in the outside world.
- Causes realize their effects only after a delay, and can be interleaved.

5. Planning with Causal Processes

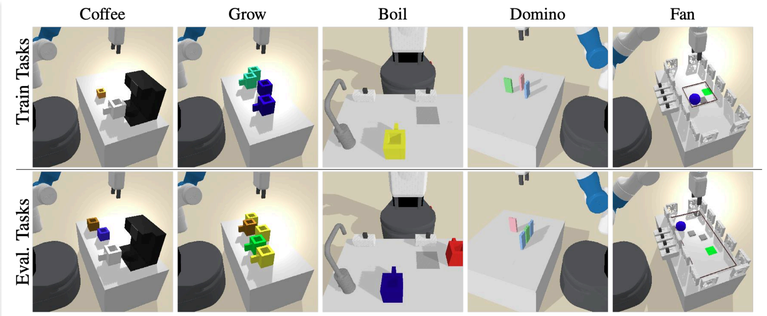


- *Causal processes* allows planning by breaking frame-by-frame dynamics into discrete jumps between abstract states

6. Process Learning and Predicate Invention

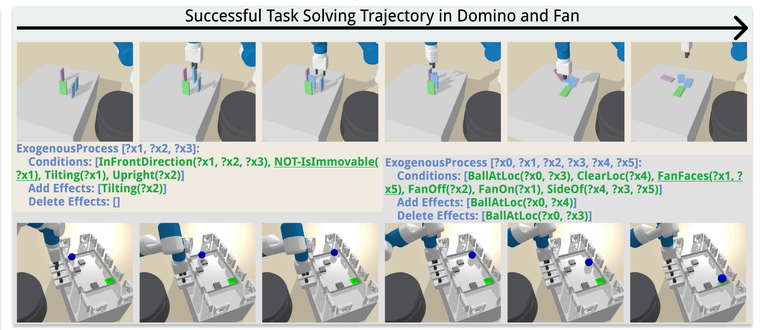
- Agent uses its current model to plan and interact with the world, then refines that model by learning new predicates and causal processes from the experience.
- Example:
  - The agent's initial model in iteration *i* leads to a failed plan (Water Spilled!).
  - Observing this failure and updating its knowledge ("Diff Learned Model"),
  - Creates a successful plan in iteration *i*+1 ("Diff plan after learning")

7. Environments

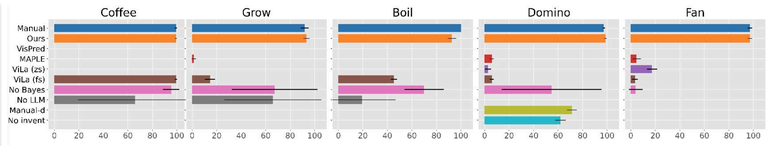


Coffee: make & pour coffee | Grow: water plants w. jugs of matching color | Boil: boil water w. jugs | Domino: topple the pink dominoes | Fan: blow the ball the target w. fans

8. Experiments



- the key learned exogenous processes, describing how dominoes cascade and how the fan's wind moves the ball.
- These processes incorporate predicates invented by the agent



- Ours consistently outperforms the VLM planning (ViLa), HRL (MAPLE) and STRIPS-style operator learning and planning (VisPred) approaches
- Ours achieves the same or better performance as Manual which uses manually engineered abstractions
- Both the Bayesian model learning and the LLM guidance play a critical role in efficient, effective, and robust model structure learning.

9. Takeaways

- Real world has *exogenous processes* and stochastic delays -- crucial for efficient and effective planning.
- VLM program synthesis + Bayesian reasoning offers an scalable and reliable way of model learning