34th International Conference on Automated Planning and Scheduling

Epistemic Exploration for Generalizable Planning and Learning in Non-Stationary Settings

Rushang Karia*, **Pulkit Verma***, Alberto Speranzon, Siddharth Srivastava


Rushang


Pulkit


Alberto


Siddharth

AAI Autonomous Agents and Intelligent Robots
ASU Arizona State University
LOCKHEED MARTIN

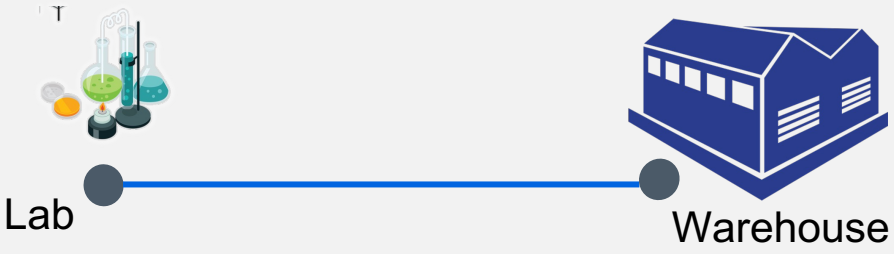
* Equal Contribution

Motivation: Learn Environment Dynamics Efficiently

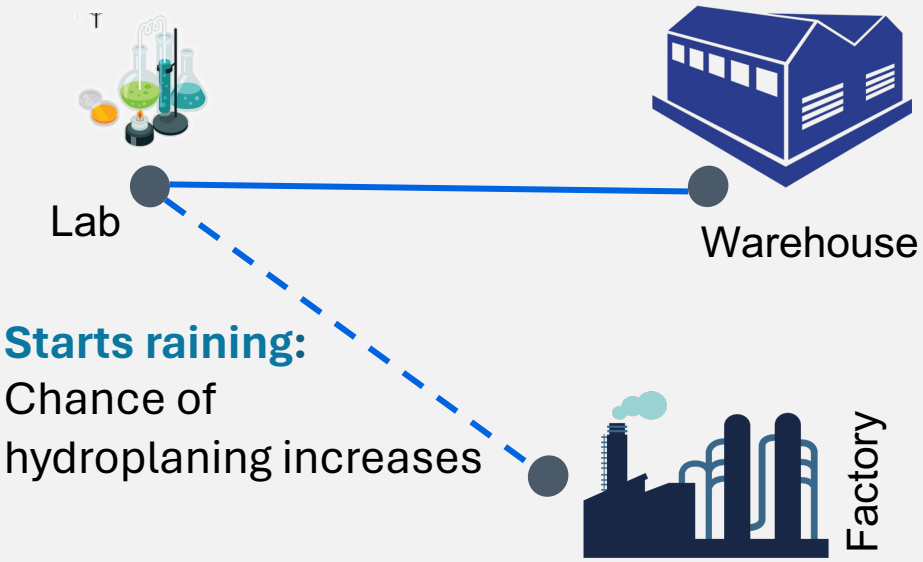
- Stream of tasks not known in advance
- Unknown, non-stationary environment dynamics (action models)
- Limited budget per task

Actively Conduct Experiments to Learn Environment Dynamics Efficiently

Task 1: Deliver goods from warehouse to lab



Task 2: Deliver goods to the factory



Problem Setting

Setting

- A stream of tasks $M_1, \ldots, M_n$ with
  - different initial states and goals
  - different state/action spaces
- A simulator whose transition function changes in an arbitrary fashion at unknown intervals.
- Reward for reaching a goal is +1 and is 0 otherwise.

Objective

- Maximize #tasks completed within a fixed budget.
- Need to adapt fast (minimize adaptive delay), and compute good solutions (minimize regret).

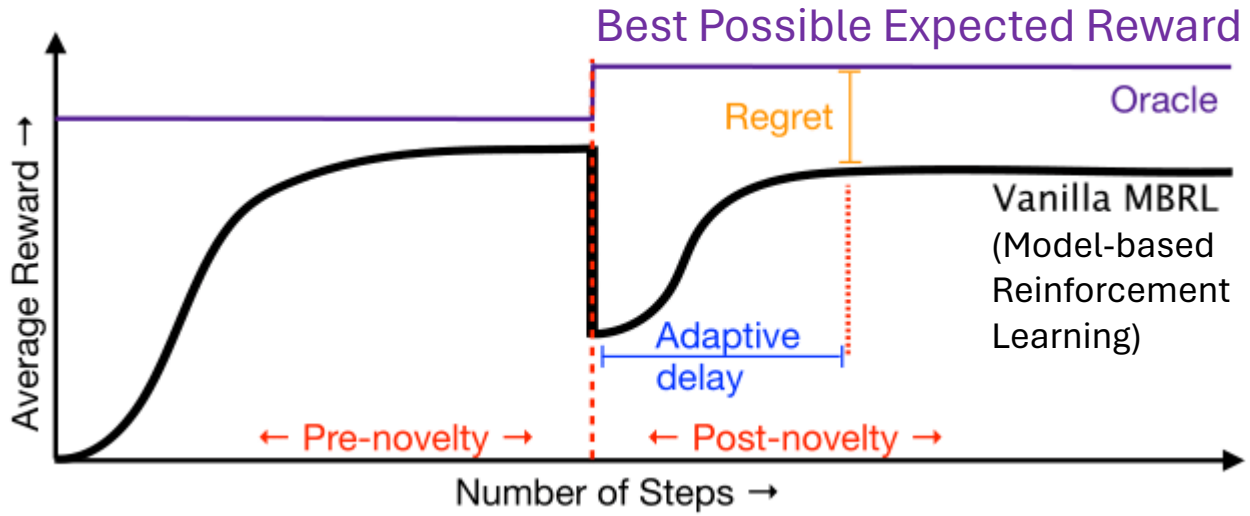


Image Source: J. Balloch et al., NovGrid: A Flexible Grid World for Evaluating Agent Response to Novelty, AAAI'22 Spring Symposium on Designing AI for Open Worlds

Research Questions

Q1

Can we develop a method to conduct experiments in the environment actively?

Q1

Can we develop a method to conduct experiments in the environment actively?

Q2

Can the exploration be more deliberative to learn more about the environment?

Q1

Can we develop a method to conduct experiments in the environment actively?

Q2

Can the exploration be more deliberative to learn more about the environment?

Q3

Would it be useful to learn a STRIPS-like model to direct exploration?

Is it even feasible?

Our Solution:
Continual Learning and Planning (CLaP)

Challenges in the Learning and Planning Paradigm

1. How do we generate sample-efficient strategies for learning?
   - Need to explore the state space
   - Need to be systematic

2. Is learning a full model worth it?
   - Might learn irrelevant actions wrt the current task.
   - Non-stationarity might waster a lot of the computational effort.

Continual Learning and Planning (CLaP)

Algorithm 1: Continual Learning and Planning

Input : RMDP M, Simulator Δ, Simulator Budget $\Delta_{\mathcal{S}}$, Learned Model $\mathcal{M}^{\uparrow}$, Horizon H, Sampling Count η, Threshold θ, Failure Threshold β

Output: $\mathcal{M}^{\uparrow}$

$s \leftarrow s_0$; $h \leftarrow 0$; $f \leftarrow 0$

$\pi \leftarrow$ modelBasedSolver$(S, A, s_0, g, \mathcal{M}^{\uparrow}, R, \gamma, H)$

while $|\Delta| < \Delta_{\mathcal{S}}$ **do**

  if $f > \beta$ *or* *unreachableGoal*$(s_0, g, \mathcal{M}^{\uparrow}, \pi)$ **then**

    explore$(\mathcal{M}^{\uparrow}, \Delta)$

  if *needsLearning*$(\mathcal{M})$ **then**

    $\mathcal{M}^{\uparrow} \leftarrow$ learnModel$(\Delta, \mathcal{M}^{\uparrow})$

    $\pi \leftarrow$ modelBasedSolver$(S, A, s_0, g, \mathcal{M}^{\uparrow}, R, \gamma, H)$

  $a \leftarrow \pi(s)$

  $s' \leftarrow \Delta(s, a)$

  $h \leftarrow h + 1$

  if $(s, a, s') \rightleftharpoons \mathcal{M}^{\uparrow}$ **then**

    $\mathcal{M} \leftarrow$ goodnessOfFitTest$(s, a, s', \Delta, \mathcal{M}^{\uparrow}, \theta, \mathit{Freq})$

  else

    $\mathcal{M}^{\uparrow} \leftarrow$ addInconsistentPredicates$(s, a, s', \mathcal{M}^{\uparrow})$

  if $s \models g$ *or* $h \geq H$ **then**

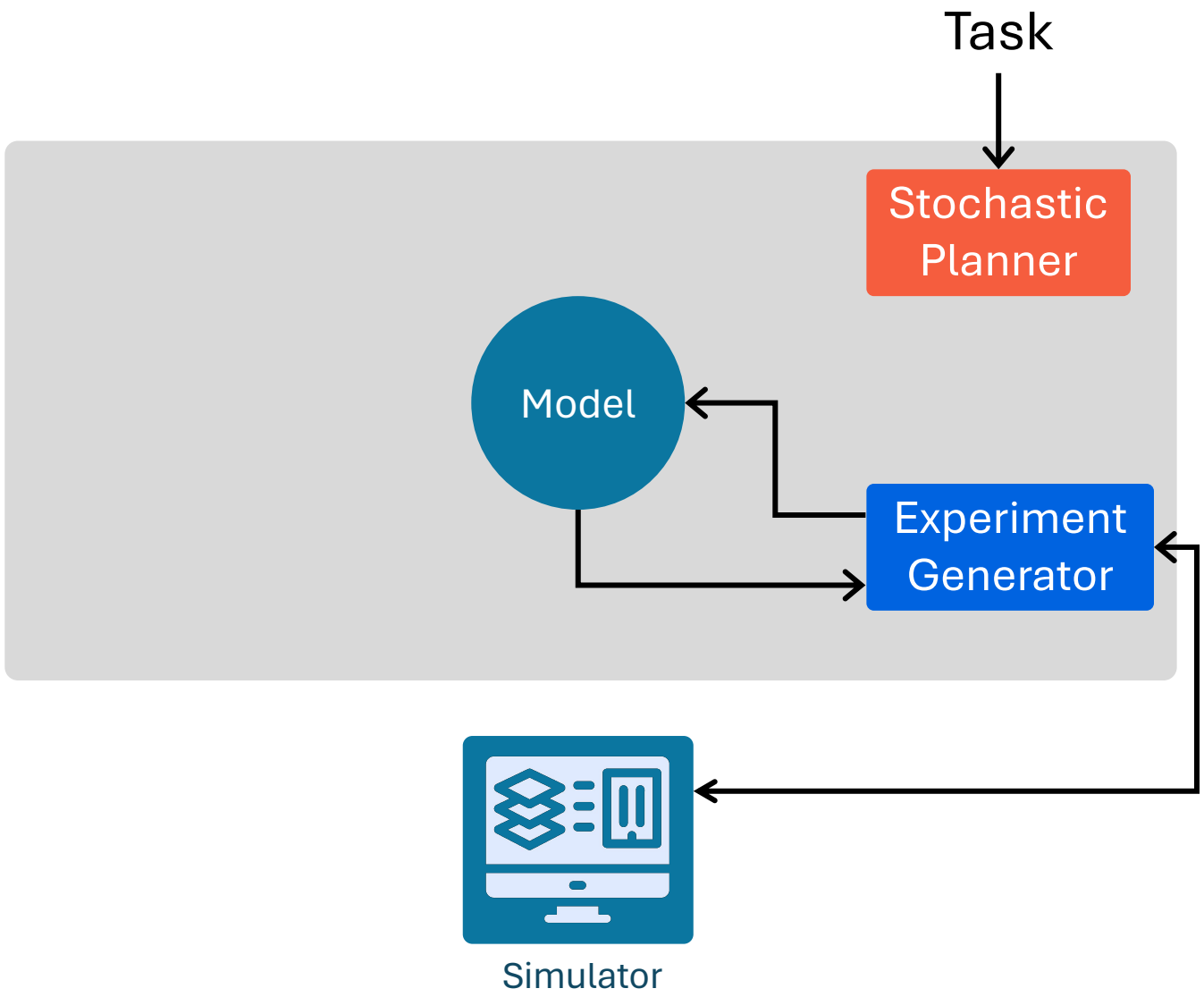
    $s \leftarrow s_0$; $f \leftarrow f + 1$ iff $s \not\models g$

  else

    $s \leftarrow s'$

return $\mathcal{M}^{\uparrow}$

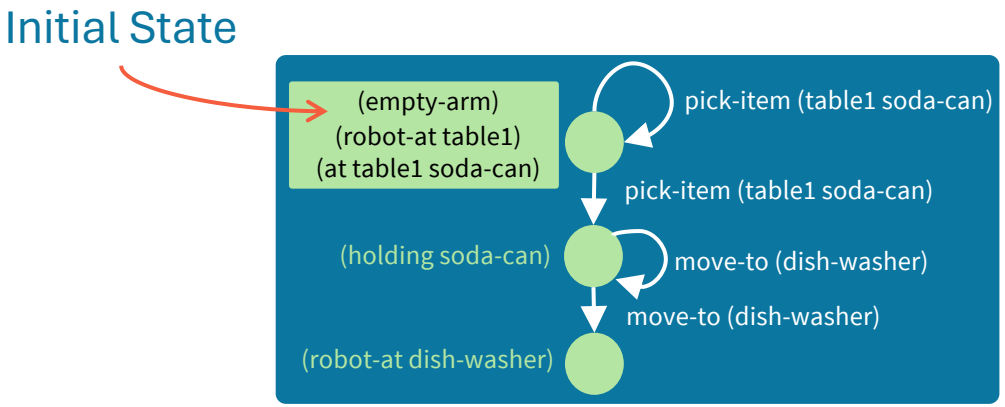
Continual Learning and Planning (CLaP)



Experiment Generation using Active Querying

[**Verma**, Karia, Srivastava; NeurIPS '23]

- Automated query generation to help fix each (+/−/∅) to a correct form.

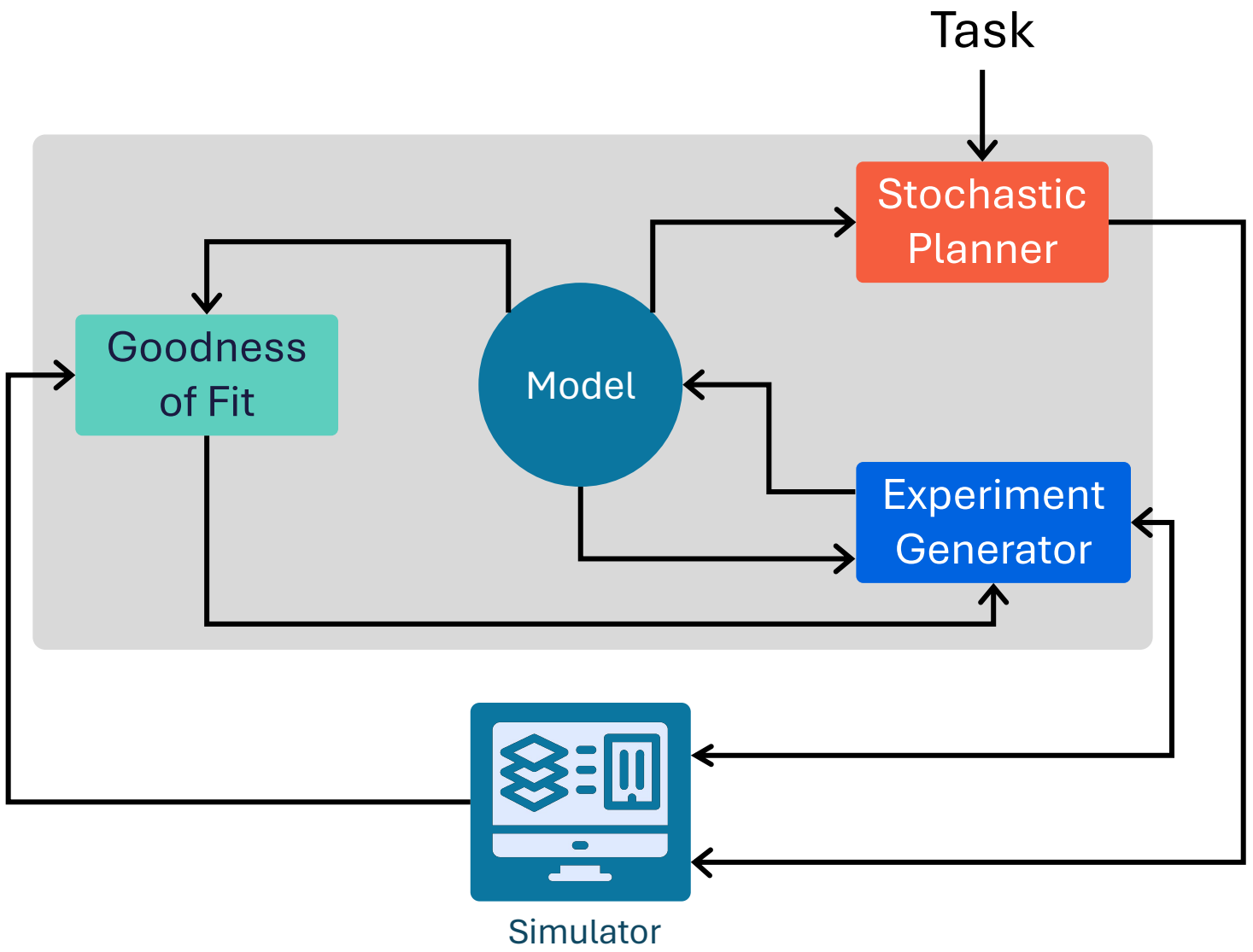


Policy: Generated Autonomously by Reduction to Non-Deterministic Planning

This process monotonically improves the learned PPDDL model

```
(:action open-door
  :parameters (?l1)
  :precondition (and
    (+/-/∅)(has_key)
    (+/-/∅)(door_open)
    (+/-/∅)(door_adjacent ?l1)
    (+/-/∅)(player_at ?l1))
  :effect (probabilistic
    0.xx (and
      (+/-/∅)(has_key)
      (+/-/∅)(door_open)
      (+/-/∅)(door_adjacent ?l1)
      (+/-/∅)(player_at ?l1))
    0.yy (and
      (+/-/∅)(has_key)
      (+/-/∅)(door_open)
      (+/-/∅)(door_adjacent ?l1)
      (+/-/∅)(player_at ?l1)))
```

Continual Learning and Planning (CLaP)



Empirical Evaluation

Setup

- 4 benchmark domains with 5 tasks each
  - Tireworld
  - ExplodingBlocks
  - FirstResponders
  - Elevators

- Changing action transitions, initial state, and goal for each task

- Limited simulator budget per task

Baselines

Oracle

Knows precisely when each novelty is introduced.

Q-Learning

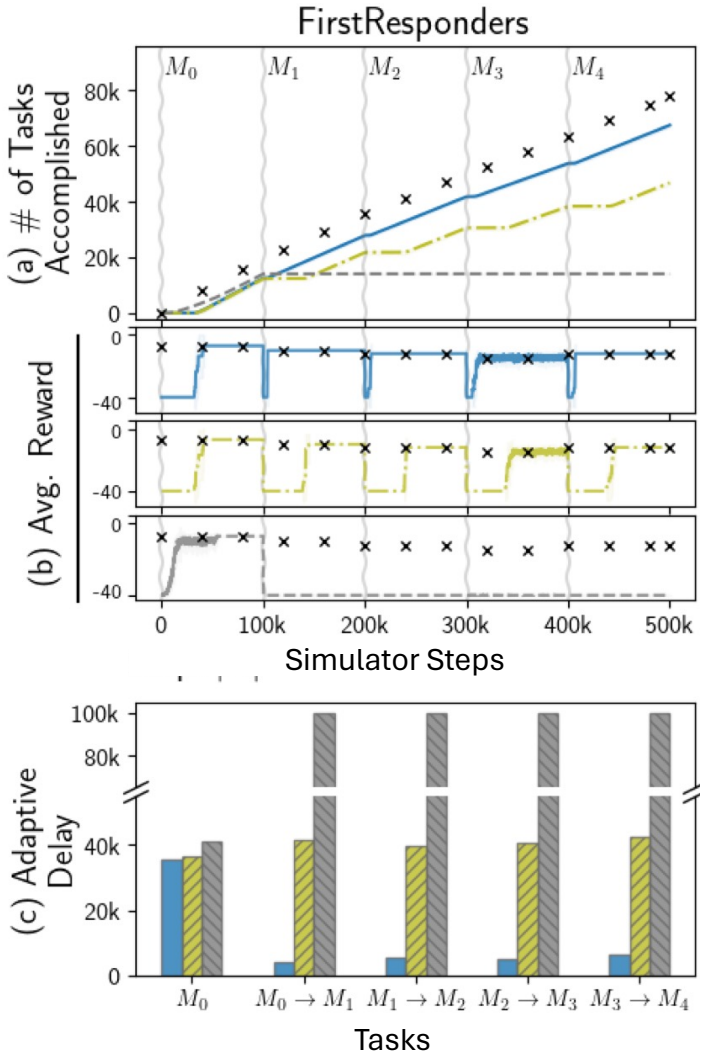
Vanilla Q-Learning that has no information about novelty.

Adaptive-QACE

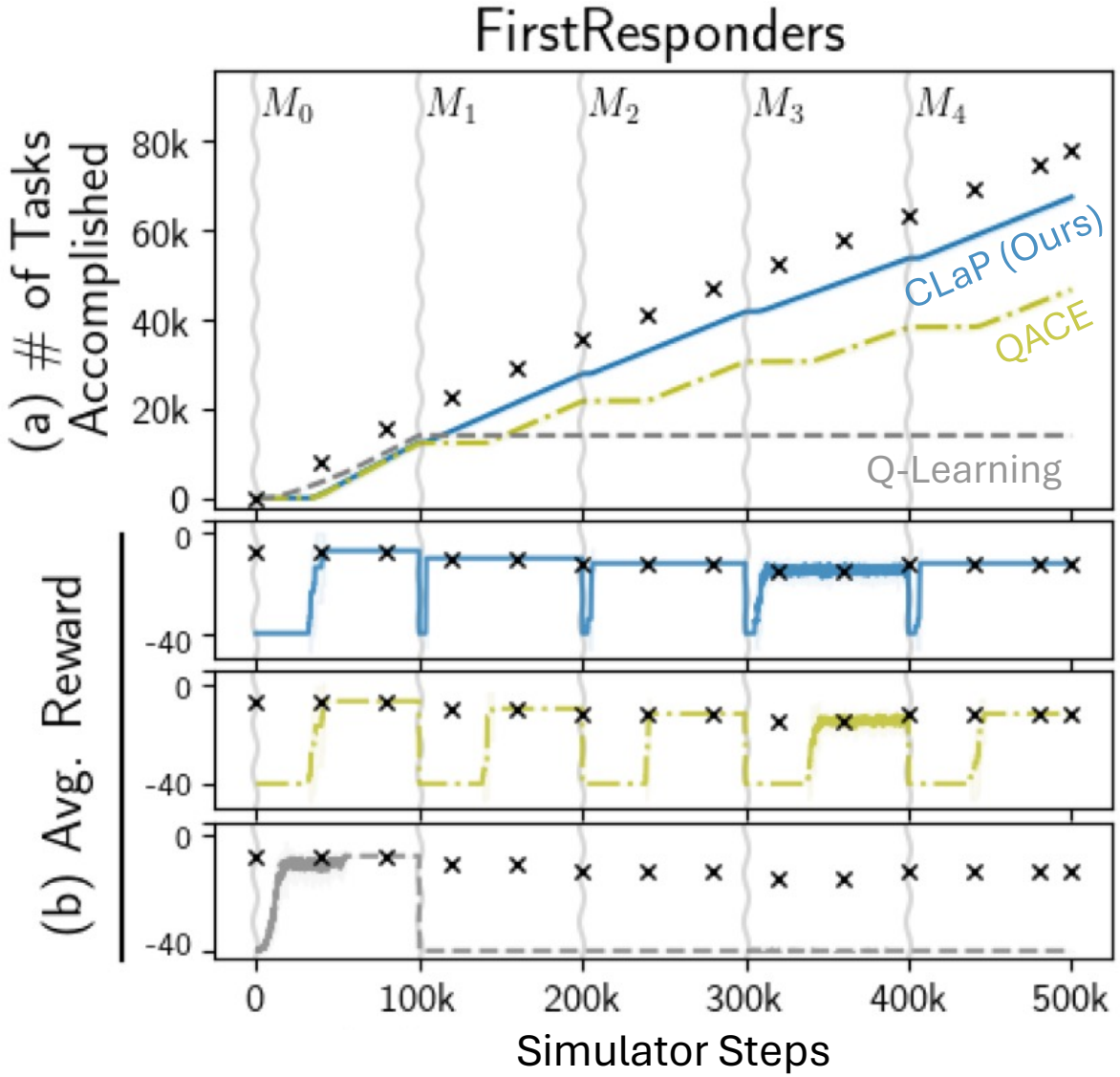
Modification of QACE to incorporate differential learning.

[**Verma**, Karia, Srivastava; NeurIPS '23]

CLaP's Performance Comparative to Oracle



Higher Values Better



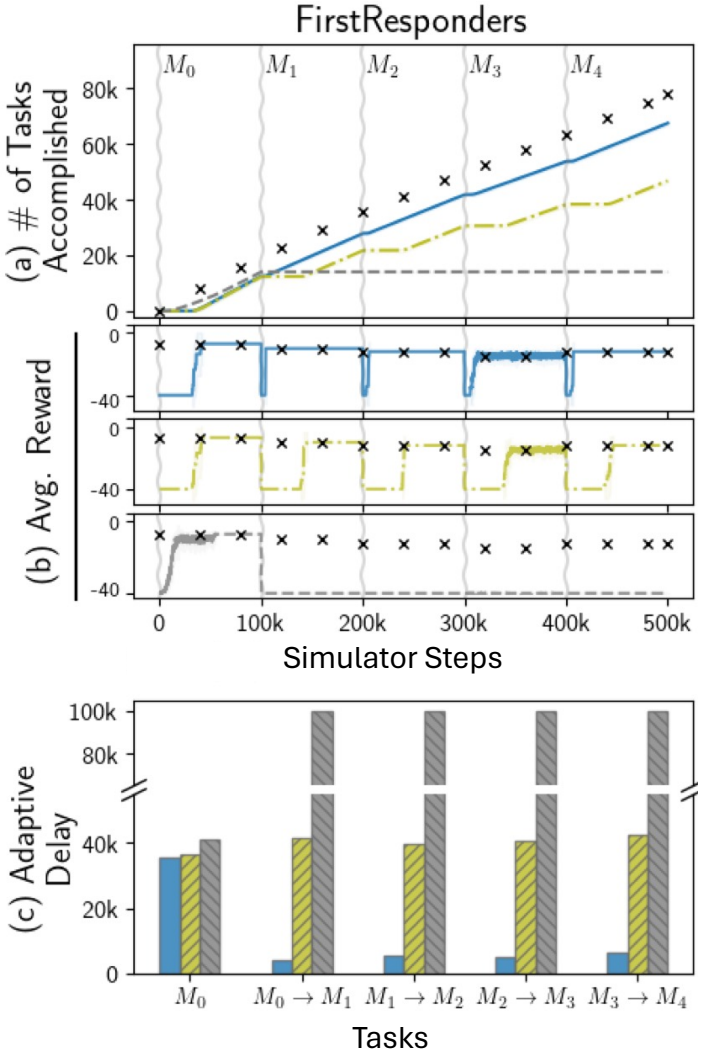
x Oracle (Upper bound on performance)

Differential Learning (CLaP) : Ours

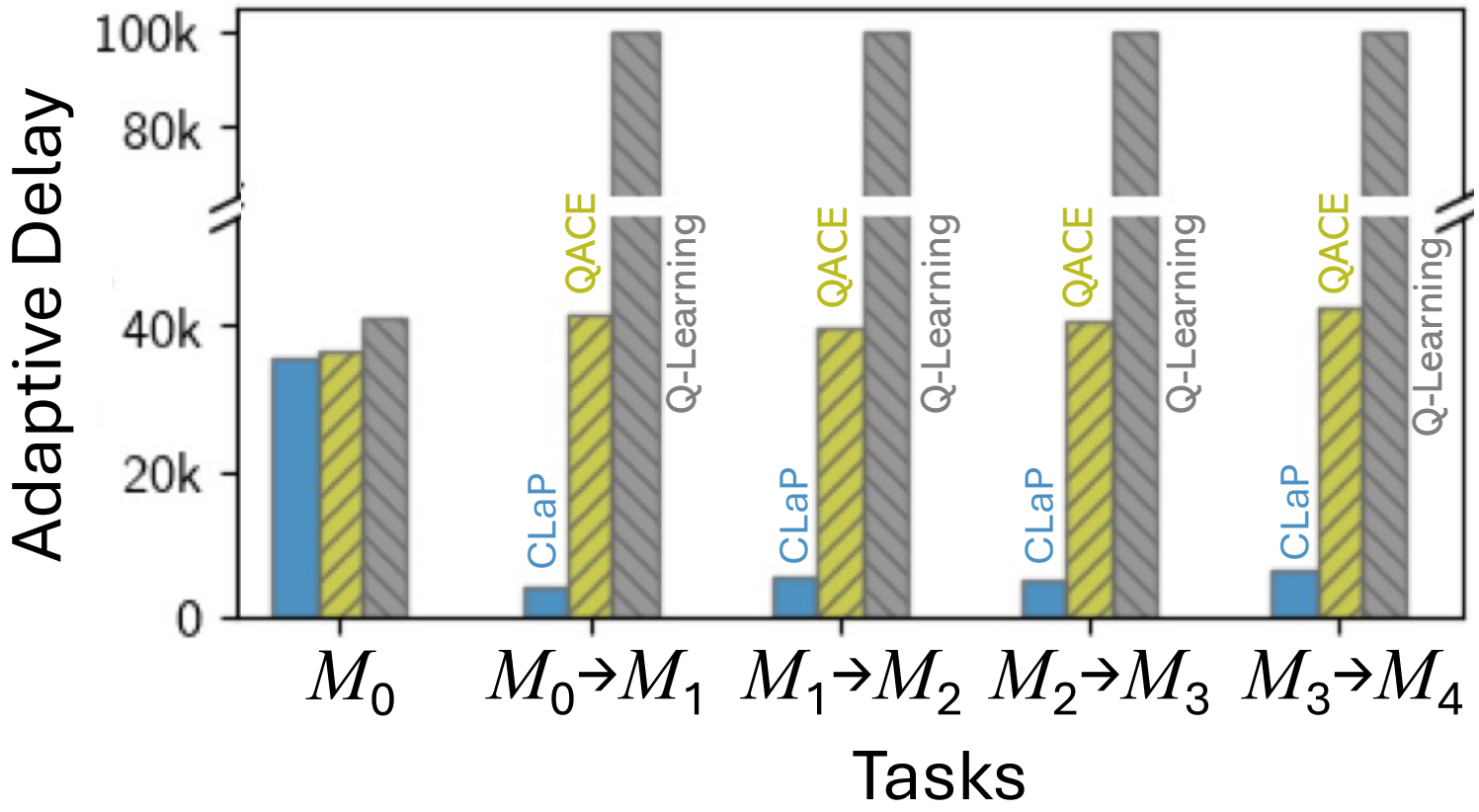
Relearning: Adaptive QACE

Q-Learning

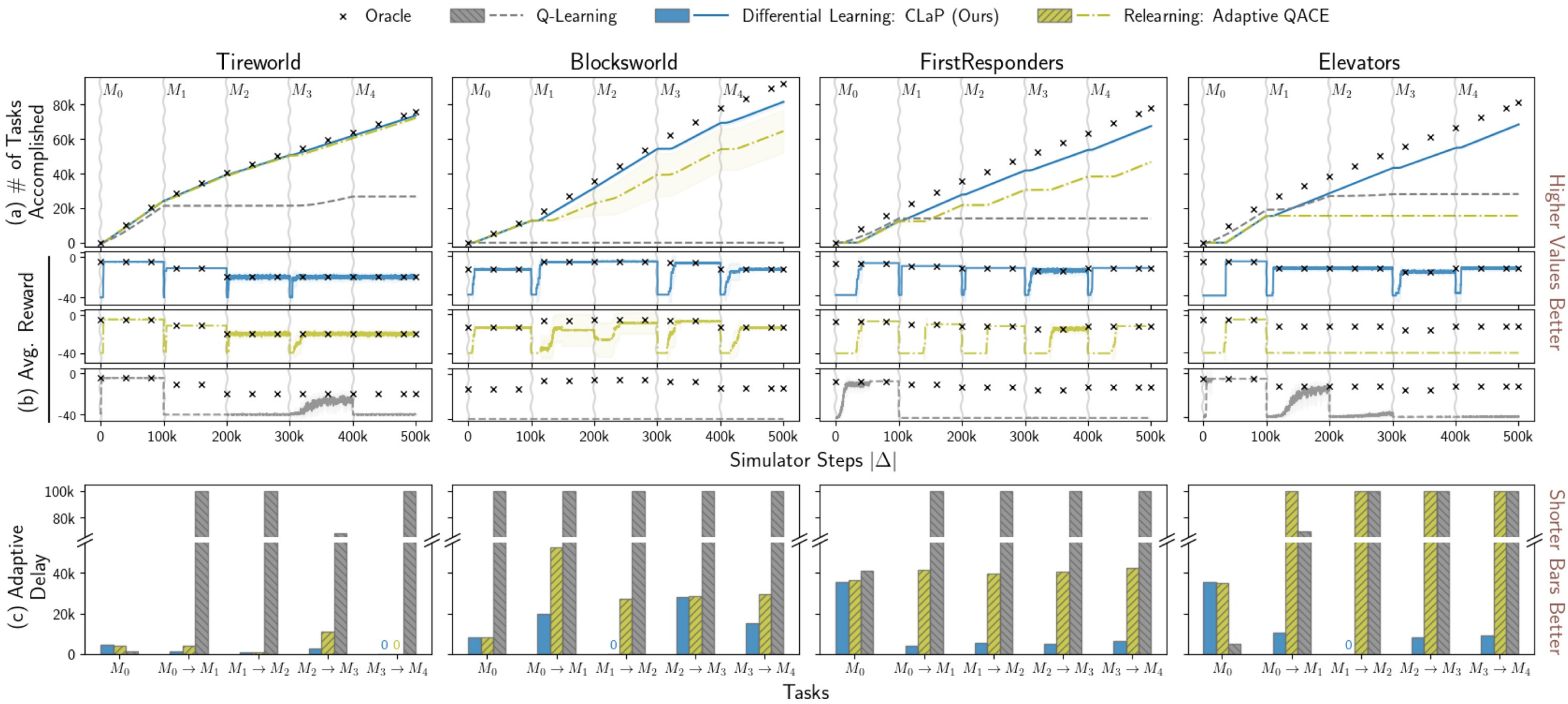
CLaP Few Shot Transfers in Non-Stationary Settings



Lower Values Better



Same Trends Across All Domains



Theoretical Guarantee: Locally Convergent Learning

Theorem 1. *Let M be an RMDP with a series of transition system changes $\delta_1, \ldots, \delta_n$ at timesteps $t_1, \ldots, t_n$ implemented using a simulator Δ, then during each stationary epoch between t_i and t_{i+1} CLaP performs locally convergent model learning.*

- Between each consecutive stage, ClaP's model learning is *locally convergent*.
- *Locally Convergent:* The model keeps getting better.

Conclusions and Future Work

- CLaP is a sample-efficient method for solving tasks under non-stationarity
- If we use some resources for experiment design, that makes up for the costs by increasing efficiency

Future Directions

- Include information about the future goal(s)
- Include priors on the transition function change when available

verma.pulkit@asu.edu