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

Rushang Karia*, Pulkit Verma*, Gaurav Vipat, Siddharth Srivastava

Autonomous Agents and Intelligent Robots
Arizona State University

Motivation

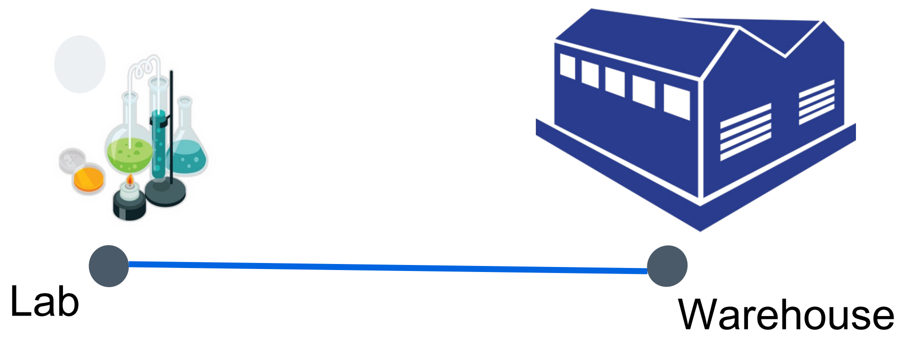
- Stream of tasks not known in advance
- Unknown, non-stationary environment dynamics
- Relational state representation
- Limited simulator budget per task

Driver Information	
Driver License	Name
D0000000	John
D0000001	Amy
...	...

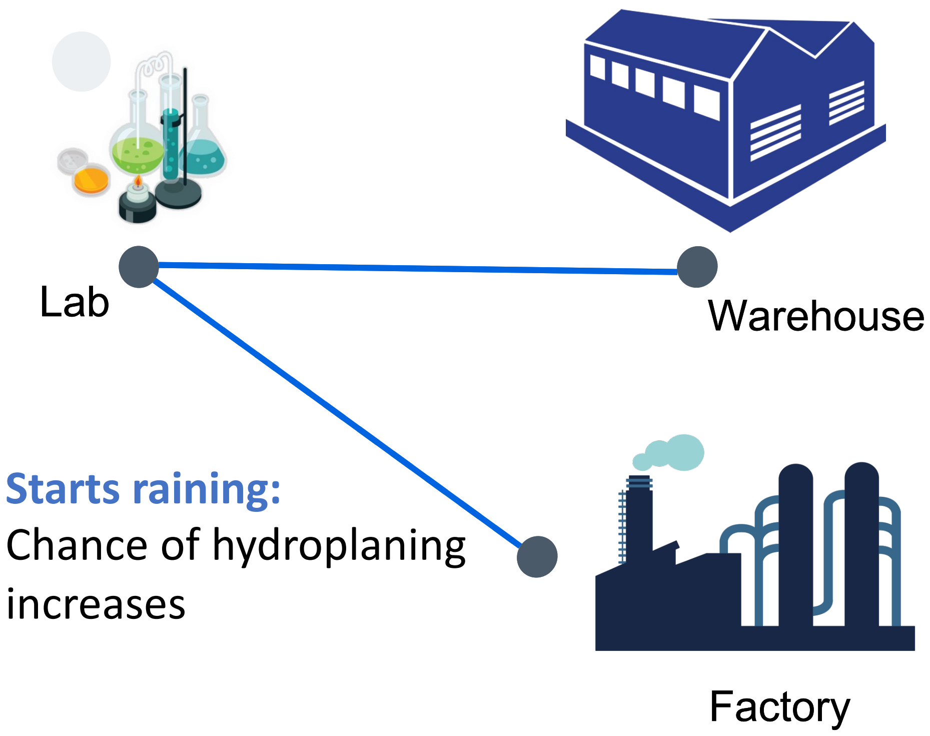
Truck Log		
Truck #	Driver #	Destination

Package Info				
ID	Name	Source	Destination	Carried By
ABC1	AMZN	Tempe	San Jose	NONE
...	...	...		

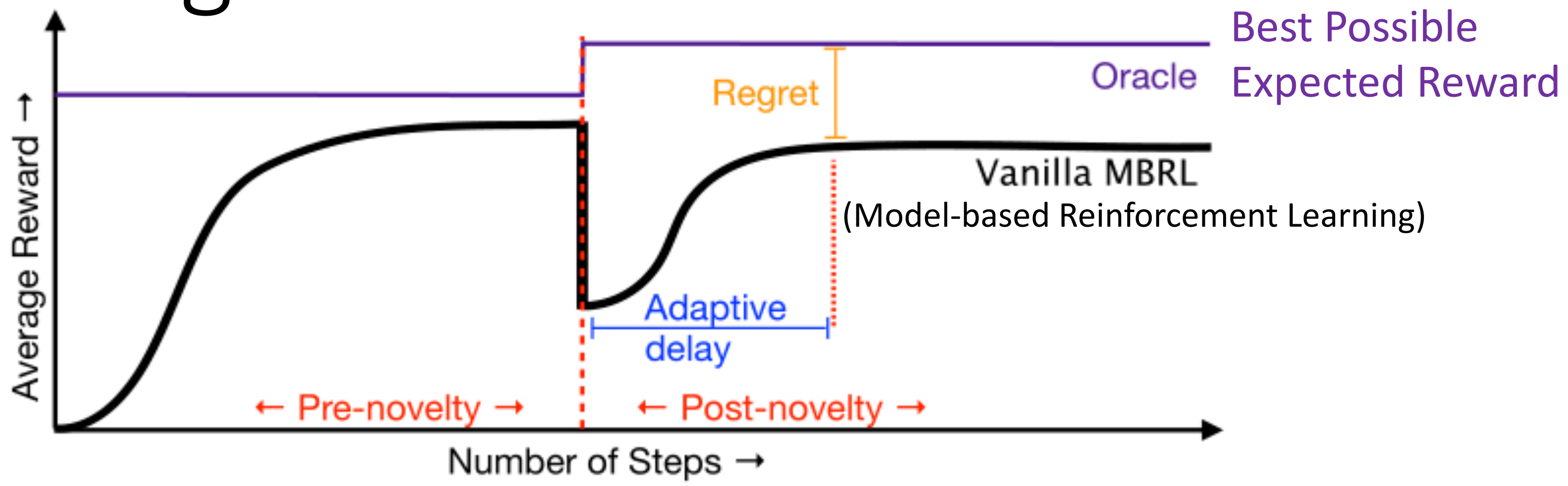
Task 0: Deliver goods from warehouse to lab



Task 1: Deliver chemicals to the factory



Problem Setting



Problem

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

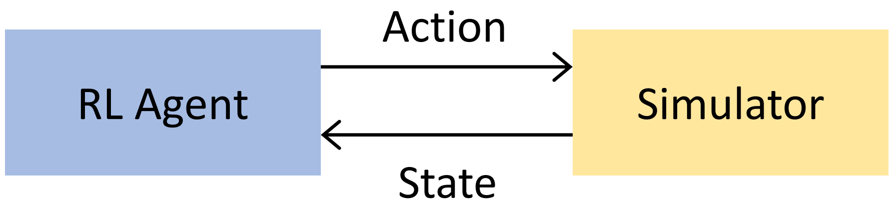
Objective

- Maximize the tasks accomplished (goals reached) within the simulator budget
  - Need to adapt fast (minimize adaptive delay), and compute good solutions (minimize regret)

Figure Source: J. Balloch et al., NovGrid: A Flexible Grid World for Evaluating Agent Response to Novelty, AAAI Spring Symposium 2022 on Designing AI for Open Worlds

Reinforcement Learning (RL)

- Collect experience from the simulator and use it to solve tasks



$$Q(s,a) = (1-\alpha)Q(s,a) + \gamma\left[R(s,a) + \max_{a'} Q(s',a')\right]$$

Advantages

✓Low input requirements

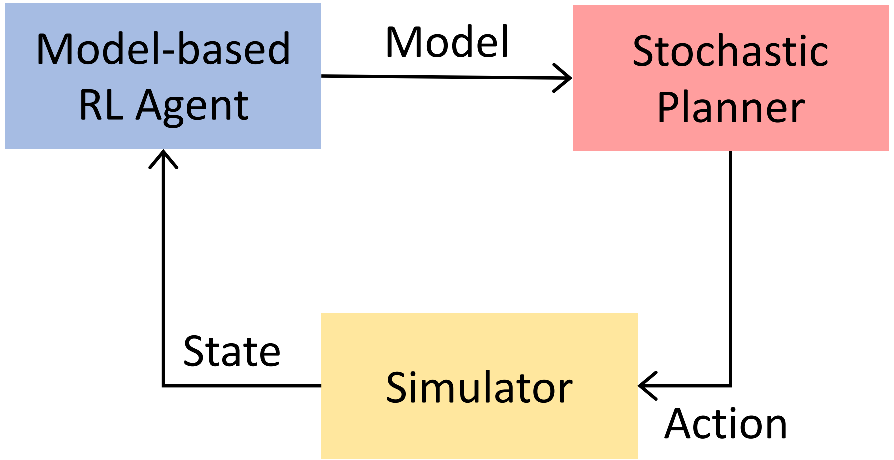
✓Can handle non-stationarity

Disadvantages

× Sample inefficient

× Not suitable for transfer

Learning and Planning



- Learn a **model** using the simulator

- Use the **model** to compute a policy and execute it on the simulator

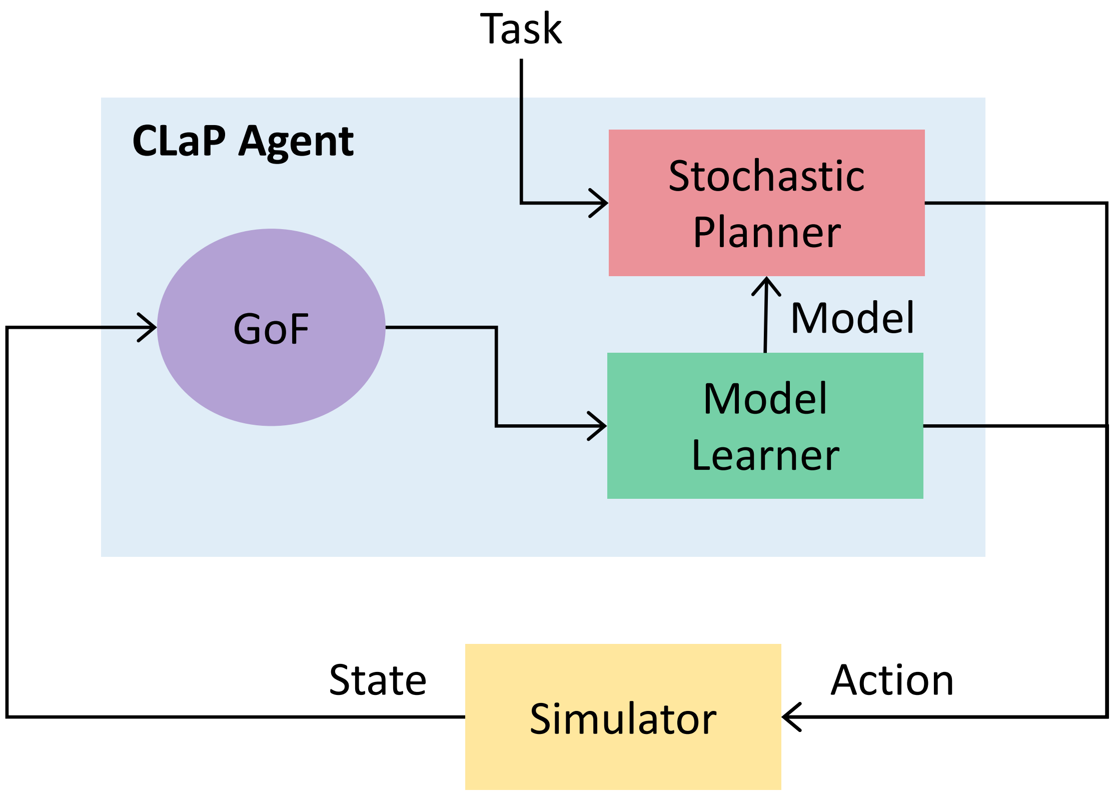
$$V^*(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} \delta(s,a,s') V^*(s') \right]$$

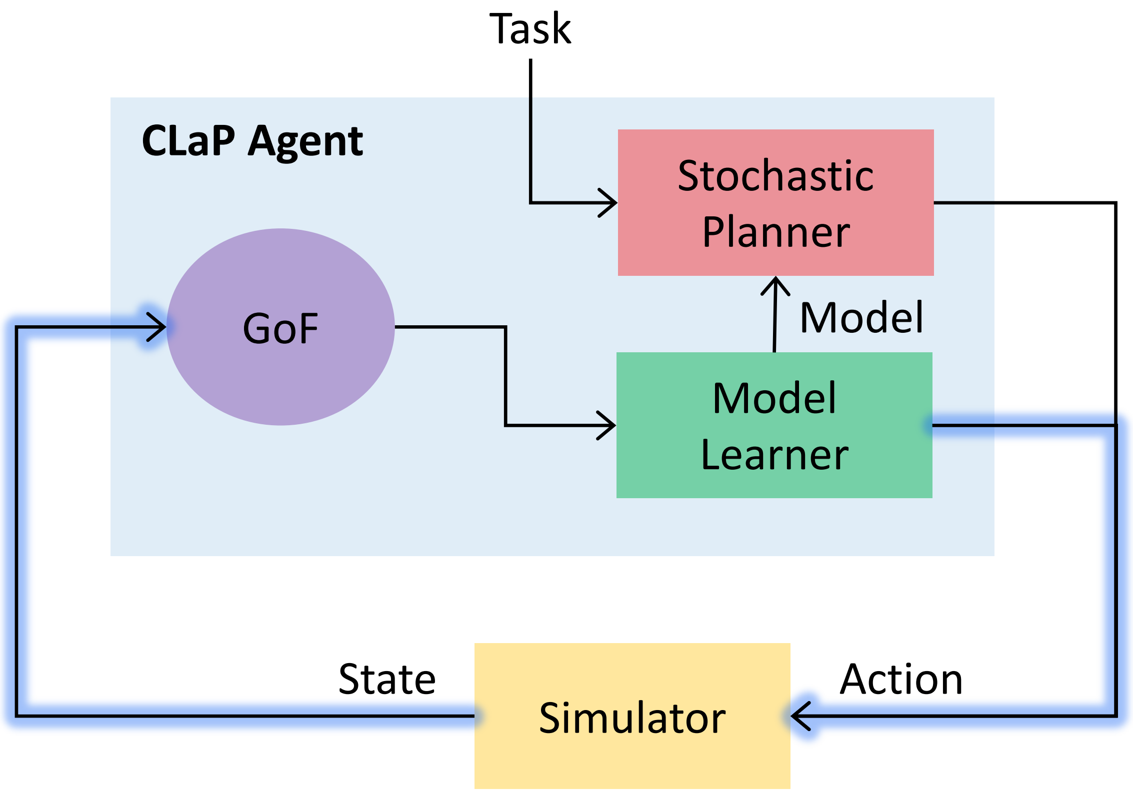
Challenges in the Learning and Planning Paradigm

1. **How do we generate useful experience for learning models while ensuring sample efficiency?**
   - Need to explore the state space to generate experience for learning good models
   - If not systematic, the model-learning process might be very sample inefficient

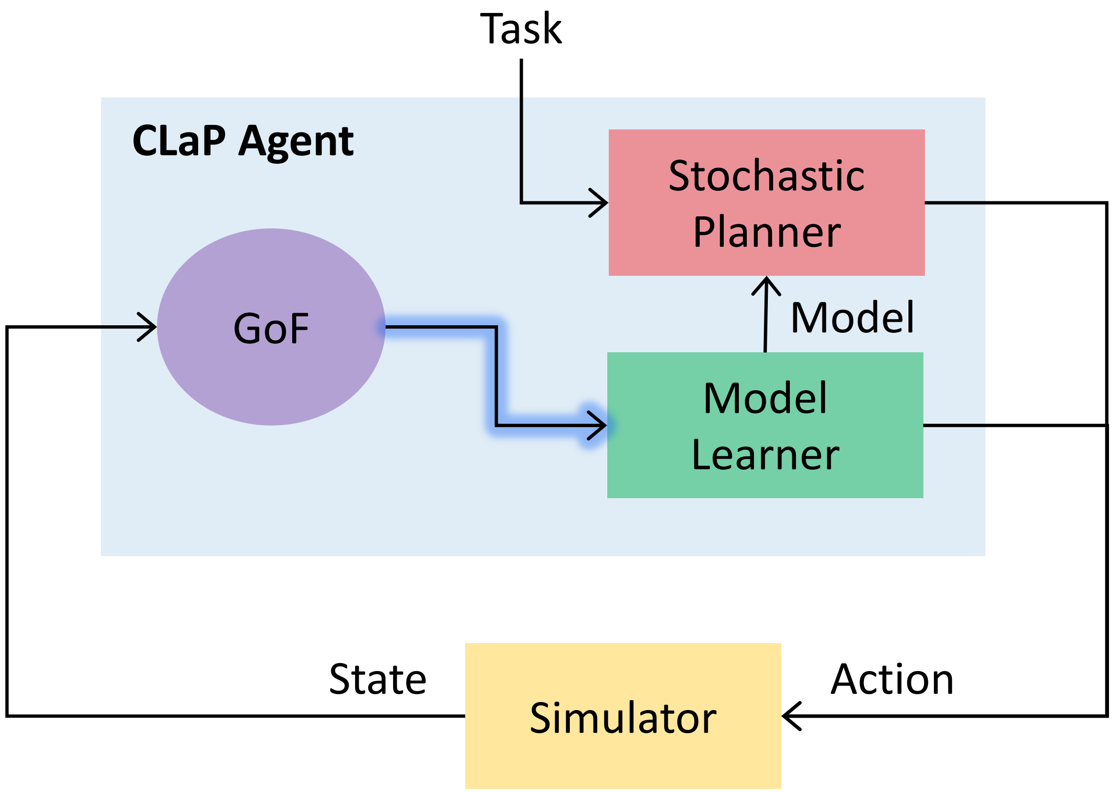
2. **Is learning a model worth it rather than learning a policy using RL?**
   - Learning full models will learn irrelevant actions not useful for solving the current task
   - Non-stationarity might render a lot of the computational effort expended wasted

Our Approach: Continual Learning and Planning (CLaP)

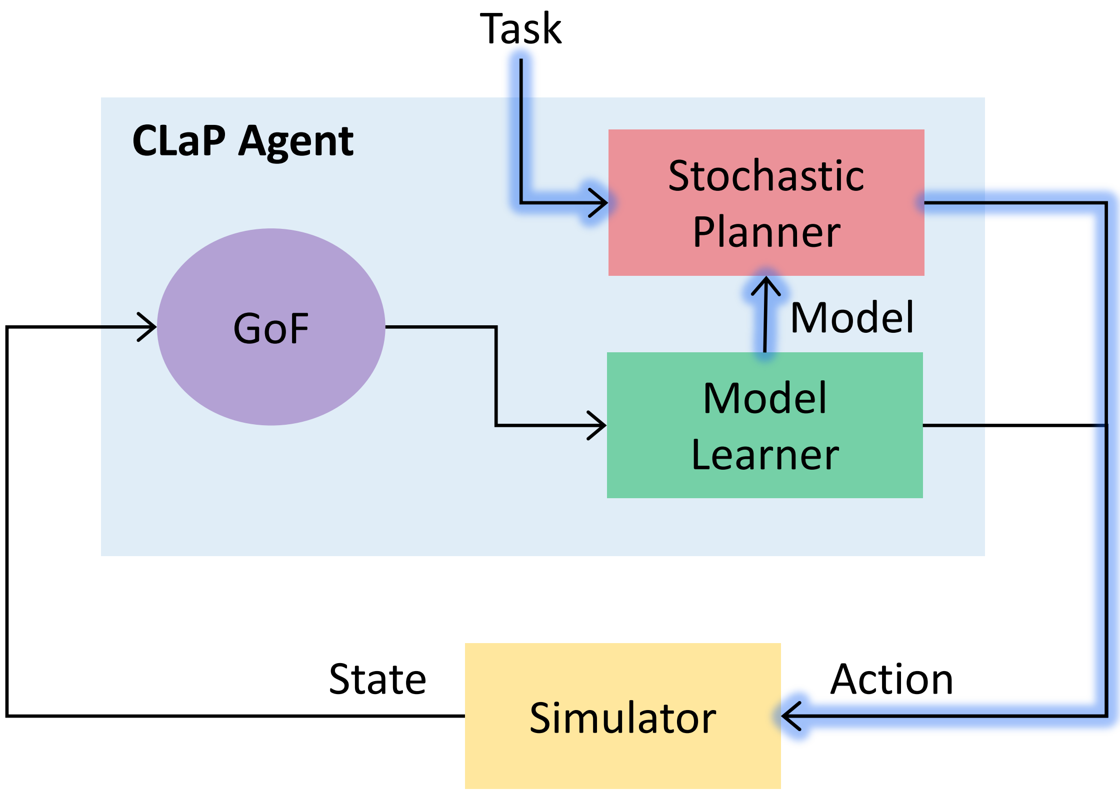




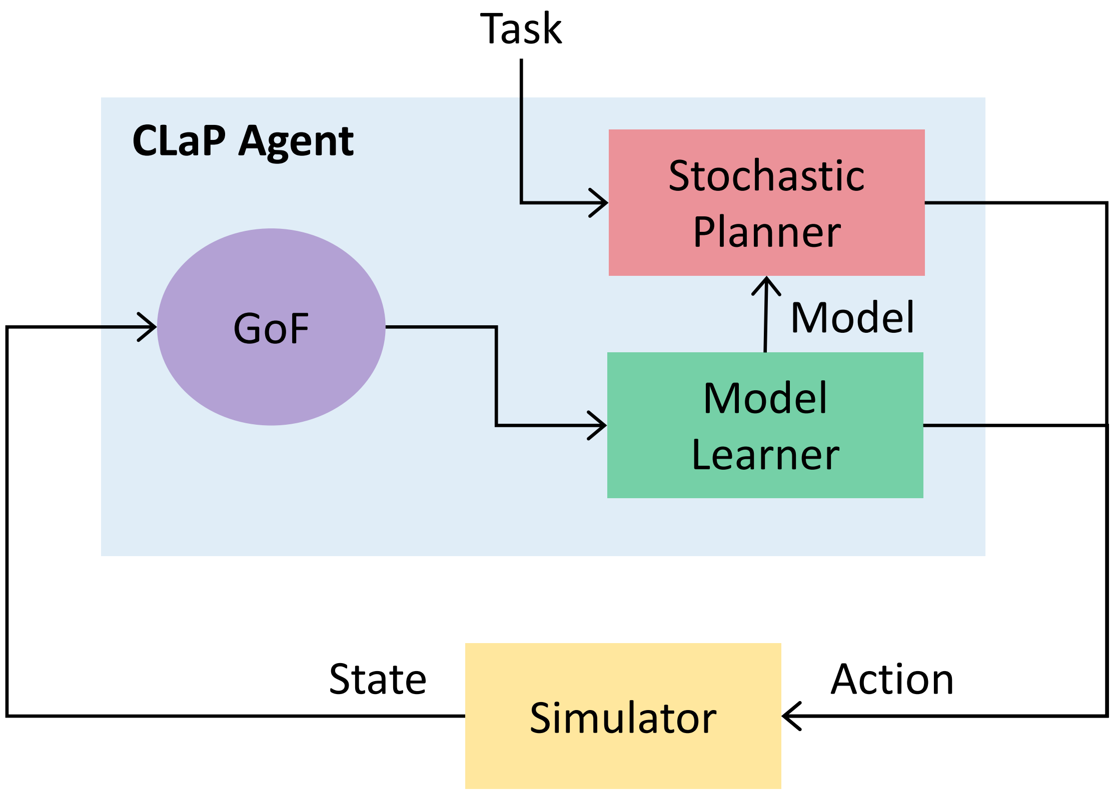
- Use an active query-based learning approach for learning lifted PPDDL models
  - Simulator’s implementation does not need to be PPDDL!
- Keeps track of uncertainty in models/discrepancies with experience
- Automatically generates investigative behavior for resolving model uncertainty



- We employ goodness-of-fit (GoF) tests to quickly detect whether effects are being sampled from the same distribution



- Finally, we utilize a stochastic, model-based planner to compute policies and use these computed policies to accomplish the task



Theoretical Results

- We guarantee that our approach is sound (always learns correct models)
- We also show monotonic improvement as more data is collected

Taxonomy of Model-based Learning

	Known Drift	**Unknown Drift**
Comprehensive (full) learning	QACE-S (Verma et al; 2023)	QACE (Verma et al; 2023)
Need-based Learning	-	**CLaP**

- **Known/Unknown Drift:** Whether changes about the transition system are advertised to the agent

- **Comprehensive/Need-based:** Whether relearning must be done from scratch or whether need-based updates can be made

Empirical Evaluation

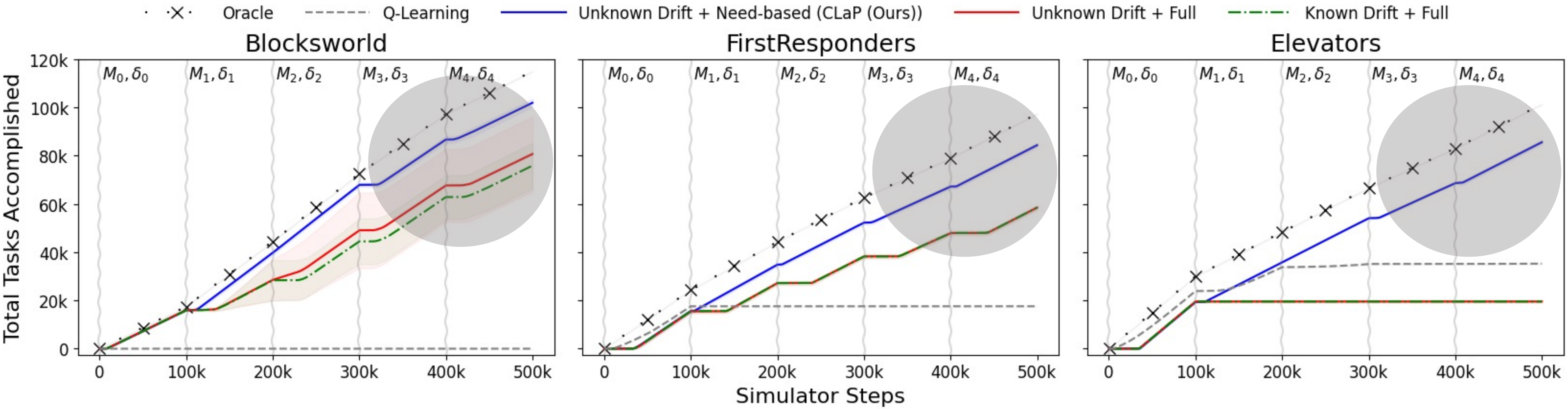
- 4 benchmark domains from the Intl. Probabilistic Planning Competition (IPPC)

- 5 tasks per domain (100k step budget per task, horizon=40)
  - Different init states, goals, and transition functions between each task

- **Non-stationarity:** perform [1-6] changes in a random action from prev. task
  - Modifications can either add/delete/modify preconditions or effects

Also used two additional baselines: Oracle and Q-Learning

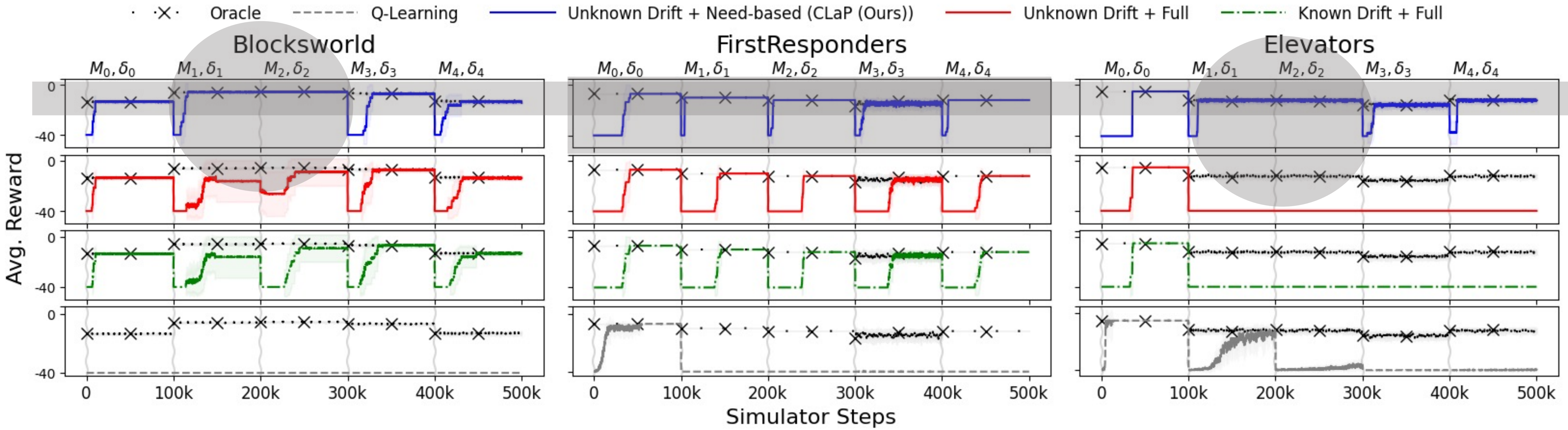
Sample Efficiency (CLaP = solid blue line, **higher better**)



Sample Efficiency: We measure the total tasks accomplished within the simulator budget

- CLaP completes more tasks
- CLaP also closely matches the performance of the oracle

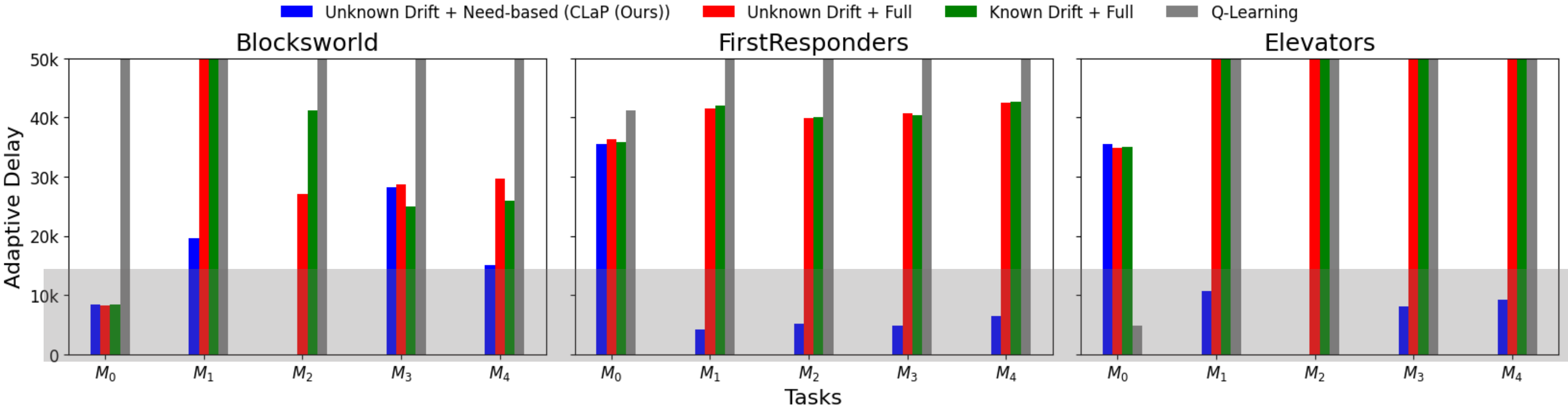
Avg. Reward (CLaP = solid blue line, **higher better**)



Avg. Reward: We measure avg. reward obtained while executing tasks greedily

- CLaPs avg. reward converges to that of the Oracle
  - Our approach can learn models that closely approximate the ground truth
- CLaP zero-shot transfers in some cases and few-shot transfers policies

Adaptive Delay (CLaP = solid blue line, **lower better**)



Adaptive Delay: We measure the total # of steps required for a steady-state performance of a task within a fraction of the Oracle's (10%)

- CLaP has much better generalizability
  - It transfers knowledge such that very little learning is required

Conclusions and Future Work

- CLaP is a sample-efficient method for solving tasks under non-stationarity
- Epistemic guidance helps learning models efficiently

Future Directions

- Include information about the goal in the model learning process
- What if priors on the transition function change and/or goals was available?

	Known Drift	**Unknown Drift**
Comprehensive (full) learning	QACE-S (Verma et al; 2023)	QACE (Verma et al; 2023)
Need-based Learning	**?**	**CLaP**

Conclusions and Future Work

- CLaP is a sample-efficient method for solving tasks under non-stationarity
- Epistemic guidance helps learning models efficiently

Future Directions

- Include information about the goal in the model learning process
- What if priors on the transition function change and/or goals was available?

Thank you!
Please stop by the poster!