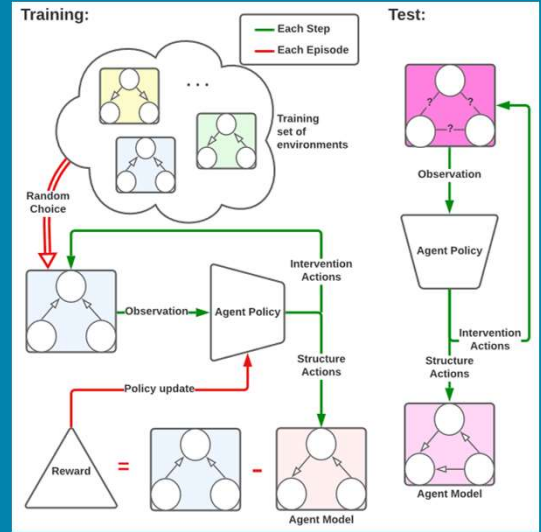


A Meta-Reinforcement Learning Algorithm for Causal Discovery

Andreas Sauter, Erman Acar, Vincent François-Lavet

published in Causal Learning and Reasoning (CleaR) 2023

We learn a **Causal Discovery** algorithm through **reinforcement learning (MCD)** that **efficiently** estimates the causal structure of environments with few nodes.



The Problem

- We want to find a DAG that represents the causal structure of our environment. But, ...
- The search space of DAGs is big and, therefore, hard to navigate.
- Interventions can help but are costly to perform.

Our Approach

- Provided a history of (interventional) samples and a current estimate of their causal relations, we want to learn to:
 - refine the current structure estimate, and
 - pick which intervention to perform

Meta-Learning Procedure

1. Randomly pick a training environment.
2. Perform a fixed amount of interventions and structure updates with the agent policy.
3. Compare the estimated DAG to the ground truth.
4. Update policy and go to 1.

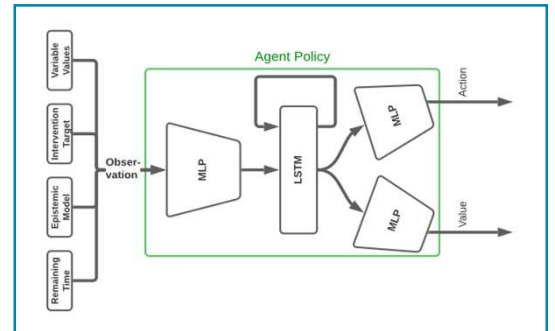
Results on Test Set

	3 Variables	4 Variables	Matched Sample	No Intervention
Random	4.43 ± 0.9	4.80 ± 1.7	4.43 ± 0.9	4.43 ± 0.9
DCDI	2.94 ± 0.7	4.44 ± 1.8	3.46 ± 1.0	-
ENCO	3.18 ± 1.1	3.74 ± 1.7	2.40 ± 0.8	-
NOTEARS	2.50 ± 0.9	3.72 ± 1.8	2.84 ± 1.02	2.50 ± 0.9
MCD (ours)	1.28 ± 0.7	3.60 ± 1.6	1.28 ± 0.7	2.6 ± 1.44

Mean structural hamming distance over 50 training environments

Conclusion

- We can meta-learn a causal discovery algorithm through reinforcement learning that performs budgeted interventions.
- This proof of concept needs further investigations to scale beyond toy problems.



The network architecture of our agent's policy

Step	1	2	3	4	5	6	7-9	Env
Epistemic Model								
Action	del(x2->x1)	del(x1->x0)	do(x1=5.0)	add(x0->x2)	add(x0->x1)	rev(x1->x0)	rev(x1->x0)	G1
Epistemic Model								
Action	del(x2->x1)	do(x1=5.0)	add(x1->x2)	add(x0->x1)	del(x2->x1)	del(x2->x1)	del(x2->x1)	G2

An example episode of the learned algorithm on two observationally equivalent environments

