Individual Project

Causal Discovery in Explainable Reinforcement Learning

Siran Shen

Imperial College London

Introduction

- *Reinforcement learning (RL)* is widely applied in society.

- To trust *RL agents*, humans need to understand how they make decisions.

- It is crucial for RL agents to become *explainable*.

Problem Statement and Goal

- This project is about explain RL models with *causality*.
- Project aim: automatise the process of inferring action influence models (AIM) at training time.
- Goal: to generate a *directed Acyclic graph(DAG)* with its corresponding action matrix (AM) which is needed for the AIM generation automatically.

Proposed Solution

- Main idea: a new method built upon the existing causal discovery algorithm - Greedy Equivalence Search (GES).

- The proposed algorithm can be spilt into two parts:

1. *Complete partial DAG (CPDAG)* and Action Matrix Generator
2. Causal DAG Maker

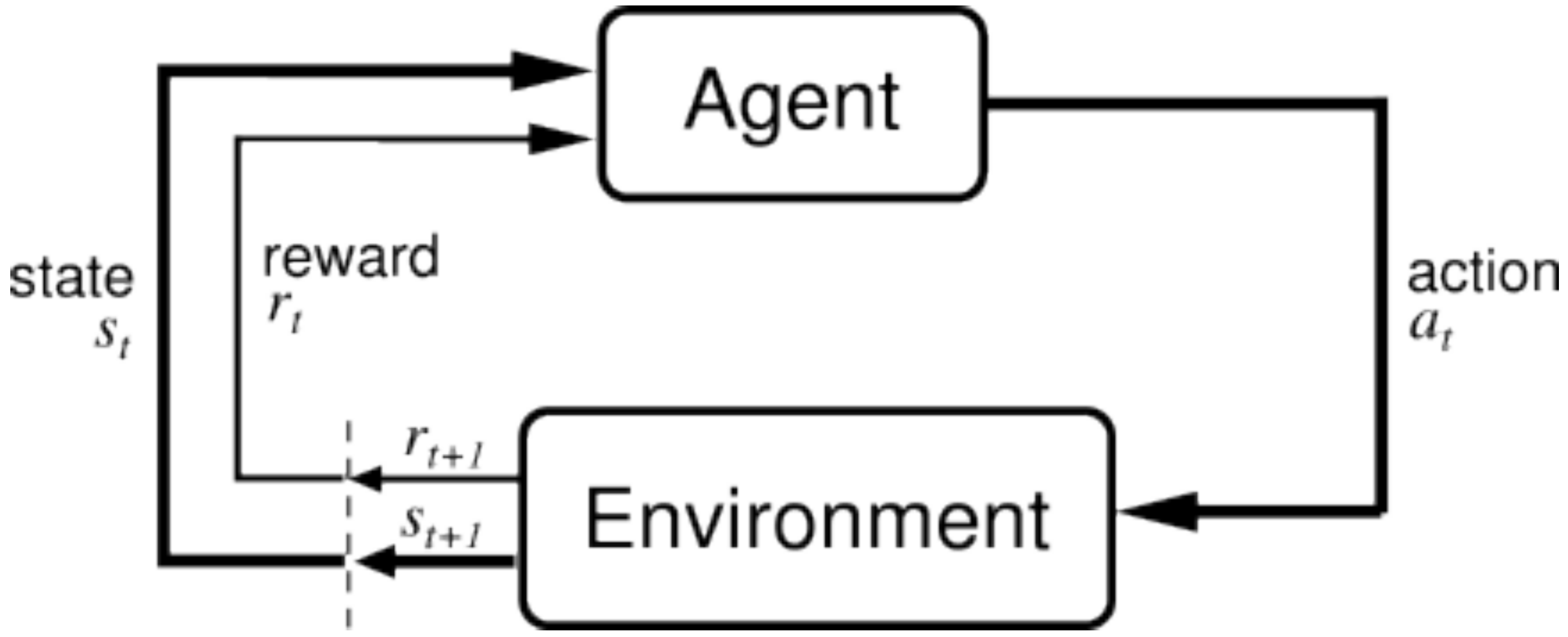
- Three RL environments:
  1. Taxi problem
  2. Cart pole
  3. Lunar landing

Background

Reinforcement Learning

- Reinforcement learning (RL): a process where agents try to find a best solution from environment.

- In RL, agent interacts with the environment continually: illustrated on the right.

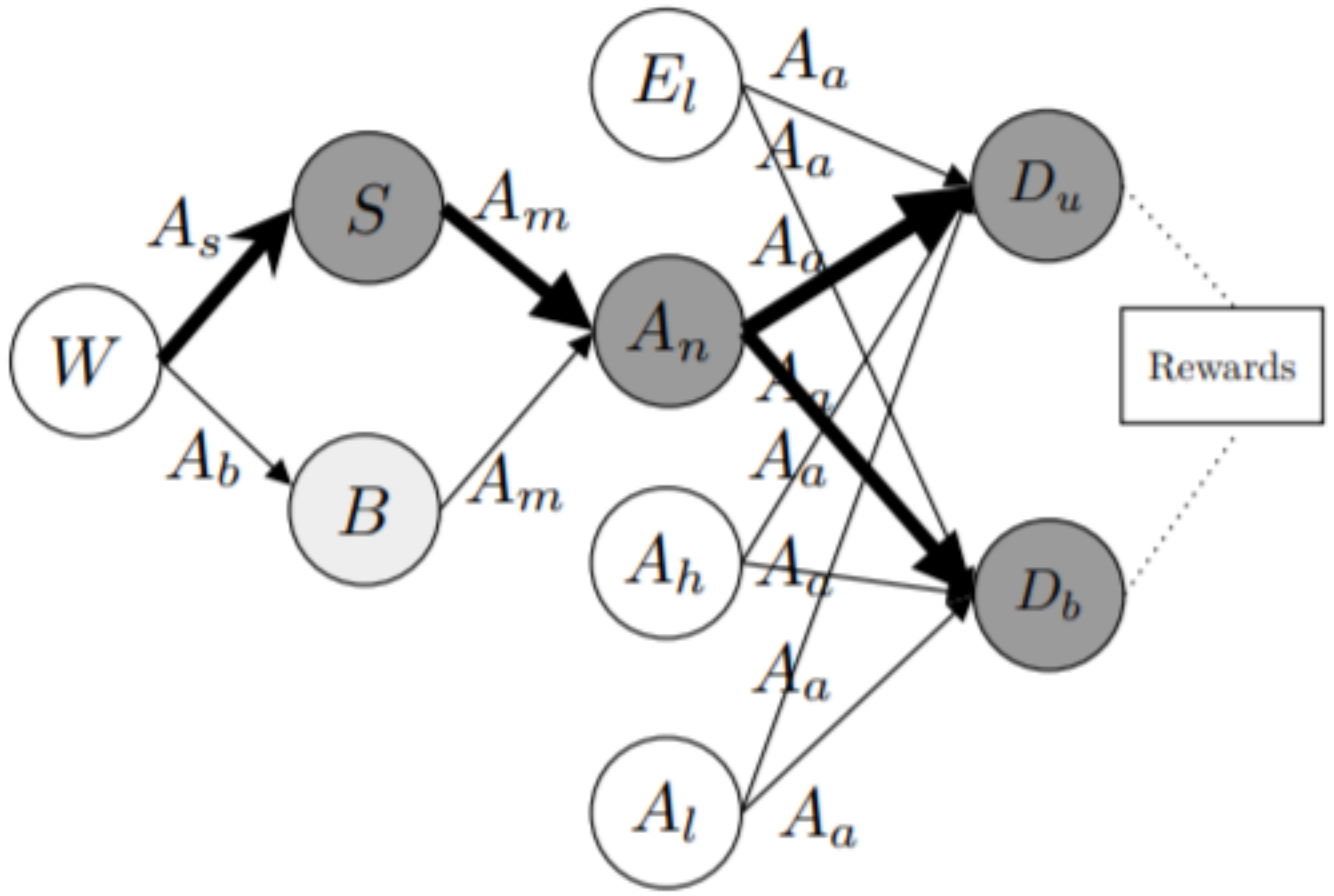


Agent-Environment Interactions in RL

Background

Notions

- Action Influence Model (AIM):



State variables:
W - Worker number
S - Supply depot number
B - barracks number
E - enemay location
A_n - Ally unit number
A_h - Ally unit health
A_l - Ally unit location
D_u - Destoryed units
D_b - Destroyed buildings
Actions:
A_s - build supply depot
A_b - build barracks
A_m - train offensive unit
A_a - attack

StarCraft Action Influence Graph

Background

Notions

- Directed Acyclic Graph (DAG): a graph without cycles and only having directed edges.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A Causal DAG in matrix form of StarCraft Action Influence Graph

- Action Matrix (AM): each non-zero element represents the index of the action performed on that edge.

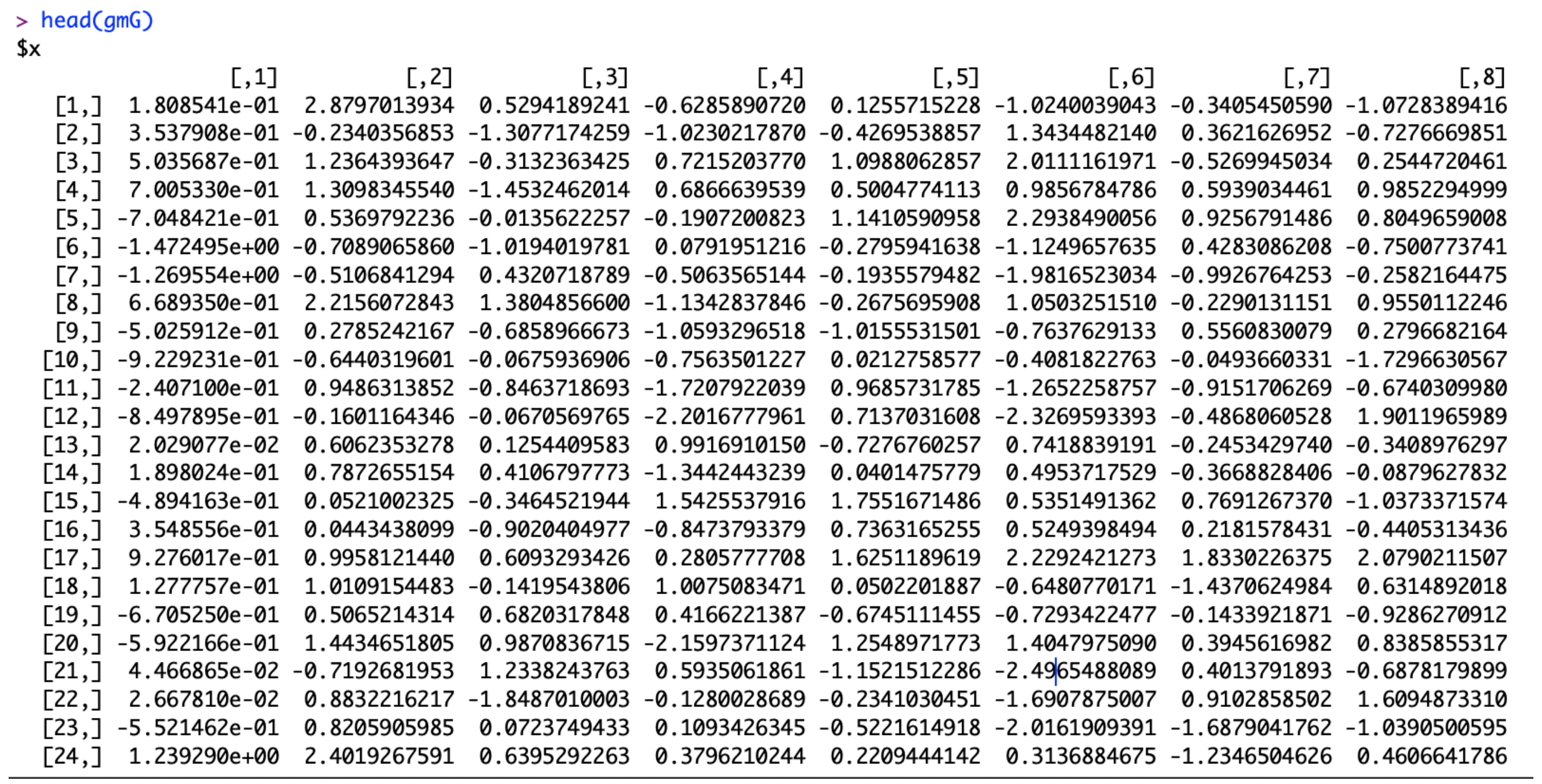
$$\begin{bmatrix} 0 & 91 & 42 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 477 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 477 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13 & 13 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13 & 13 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13 & 13 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13 & 13 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The associated action matrix

Background

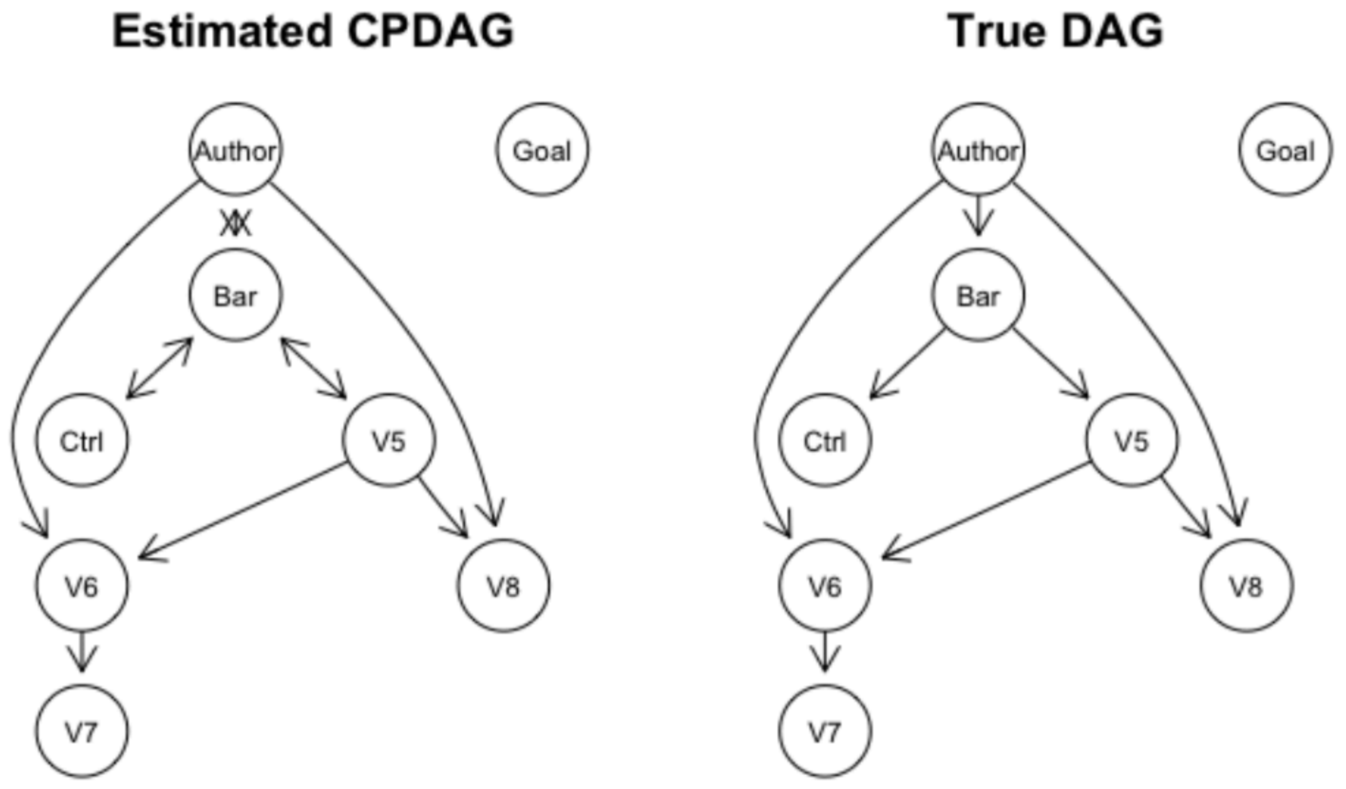
Greedy Equivalence Search(GES) Algorithm

- Input: a dataset along with an empty graph
- Output: a CPDAG with a score.
- Three stage: forward, backward and turning.



```
> head(gmG)
$x
               [,1]          [,2]          [,3]          [,4]          [,5]          [,6]          [,7]          [,8]
 [1,]  1.808541e-01  2.8797013934  0.5294189241 -0.6285890720  0.1255715228 -1.0240039043 -0.3405450590 -1.0728389416
 [2,]  3.537908e-01 -0.2340356853 -1.3077174259 -1.0230217870 -0.4269538857  1.3434482140  0.3621626952 -0.7276669851
 [3,]  5.035687e-01  1.2364393647 -0.3132363425  0.7215203770  1.0988062857  2.0111161971 -0.5269945034  0.2544720461
 [4,]  7.005330e-01  1.3098345540 -1.4532462014  0.6866639539  0.5004774113  0.9856784786  0.5939034461  0.9852294999
 [5,] -7.048421e-01  0.5369792236 -0.0135622257 -0.1907200823  1.1410590958  2.2938490056  0.9256791486  0.8049659008
 [6,] -1.472495e+00 -0.7089065860 -1.0194019781  0.0791951216 -0.2795941638 -1.1249657635  0.4283086208 -0.7500773741
 [7,] -1.269554e+00 -0.5106841294  0.4320718789 -0.5063565144 -0.1935579482 -1.9816523034 -0.9926764253 -0.2582164475
 [8,]  6.689350e-01  2.2156072843  1.3804856600 -1.1342837846 -0.2675695908  1.0503251510 -0.2290131151  0.9550112246
 [9,] -5.025912e-01  0.2785242167 -0.6858966673 -1.0593296518 -1.0155531501 -0.7637629133  0.5560830079  0.2796682164
[10,] -9.229231e-01 -0.6440319601 -0.0675936906 -0.7563501227  0.0212758577 -0.4081822763 -0.0493660331 -1.7296630567
[11,] -2.407100e-01  0.9486313852 -0.8463718693 -1.7207922039  0.9685731785 -1.2652258757 -0.9151706269 -0.6740309980
[12,] -8.497895e-01 -0.1601164346 -0.0670569765 -2.2016777961  0.7137031608 -2.3269593393 -0.4868060528  1.9011965989
[13,]  2.029077e-02  0.6062353278  0.1254409583  0.9916910150 -0.7276760257  0.7418839191 -0.2453429740 -0.3408976297
[14,]  1.898024e-01  0.7872655154  0.4106797773 -1.3442443239  0.0401475779  0.4953717529 -0.3668828406 -0.0879627832
[15,] -4.894163e-01  0.0521002325 -0.3464521944  1.5425537916  1.7551671486  0.5351491362  0.7691267370 -1.0373371574
[16,]  3.548556e-01  0.0443438099 -0.9020404977 -0.8473793379  0.7363165255  0.5249398494  0.2181578431 -0.4405313436
[17,]  9.276017e-01  0.9958121440  0.6093293426  0.2805777708  1.6251189619  2.2292421273  1.8330226375  2.0790211507
[18,]  1.277757e-01  1.0109154483 -0.1419543806  1.0075083471  0.0502201887 -0.6480770171 -1.4370624984  0.6314892018
[19,] -6.705250e-01  0.5065214314  0.6820317848  0.4166221387 -0.6745111455 -0.7293422477 -0.1433921871 -0.9286270912
[20,] -5.922166e-01  1.4434651805  0.9870836715 -2.1597371124  1.2548971773  1.4047975090  0.3945616982  0.8385855317
[21,]  4.466865e-02 -0.7192681953  1.2338243763  0.5935061861 -1.1521512286 -2.4965488089  0.4013791893 -0.6878179899
[22,]  2.667810e-02  0.8832216217 -1.8487010003 -0.1280028689 -0.2341030451 -1.6907875007  0.9102858502  1.6094873310
[23,] -5.521462e-01  0.8205905985  0.0723749433  0.1093426345 -0.5221614918 -2.0161909391 -1.6879041762 -1.0390500595
[24,]  1.239290e+00  2.4019267591  0.6395292263  0.3796210244  0.2209444142  0.3136884675 -1.2346504626  0.4606641786
```

gmG Data



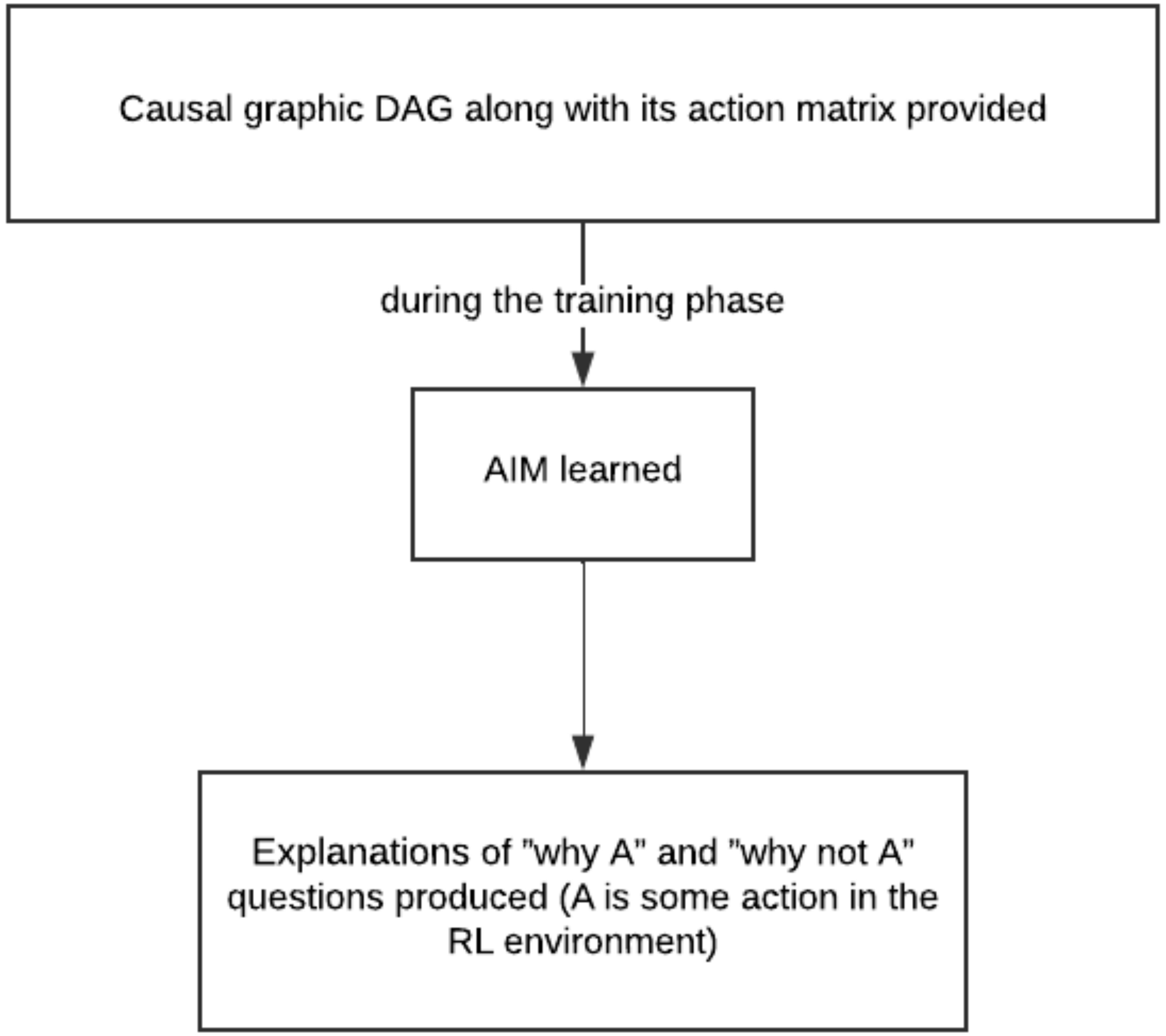
Estimated CPDAG

True DAG

Estimated and True DAG from gmG Data

Background

Learning Explanations from Casual Graphic Model



Causal graphic DAG along with its action matrix provided

during the training phase

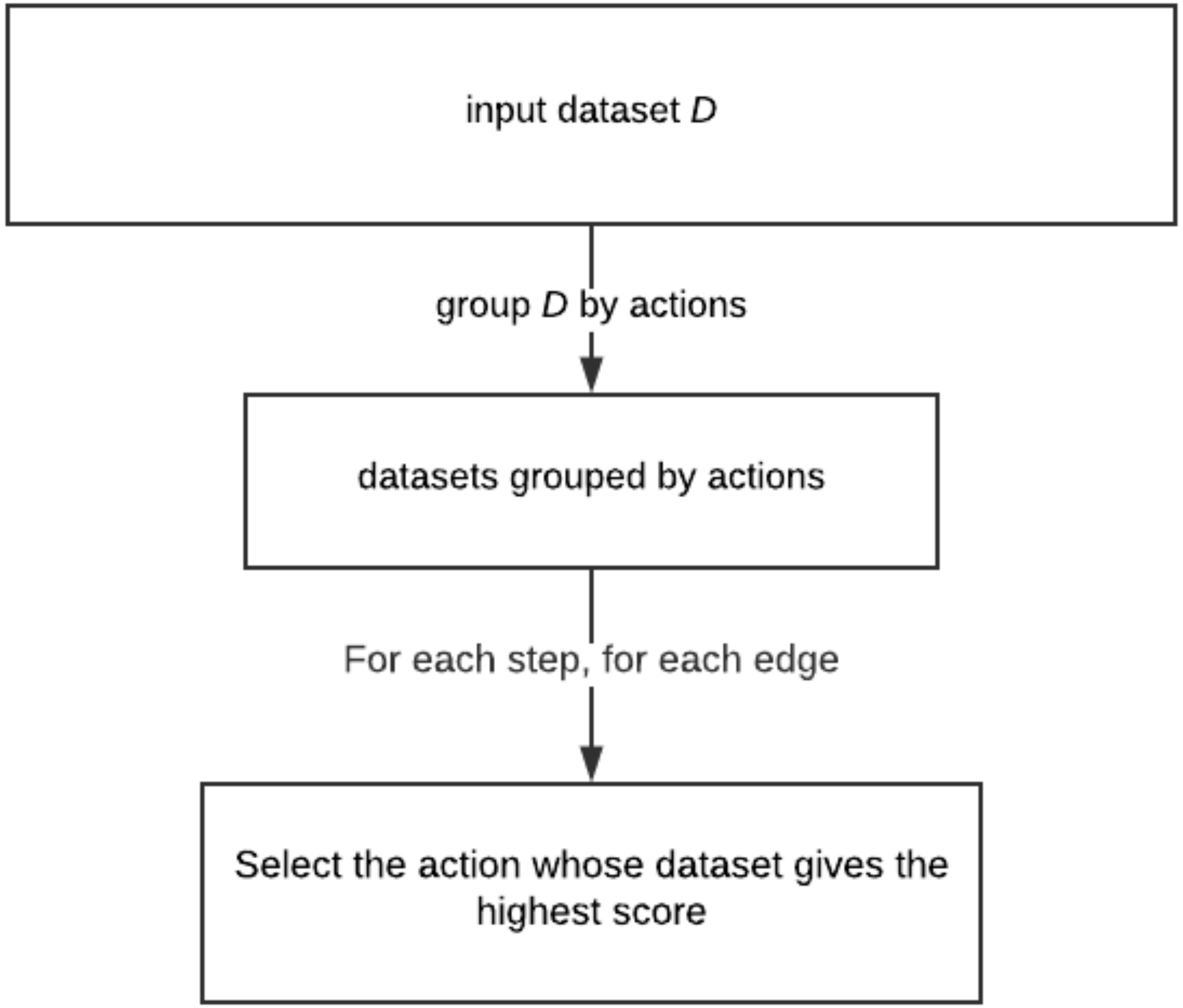
AIM learned

Explanations of "why A" and "why not A" questions produced (A is some action in the RL environment)

Methodology

CPDAG and Action Matrix Generator

- Input: a dataset *D*
- Output: a CPDAG with a score
- Based on GES algorithm
- Group dataset by RL agent's actions first



input dataset *D*

group *D* by actions

datasets grouped by actions

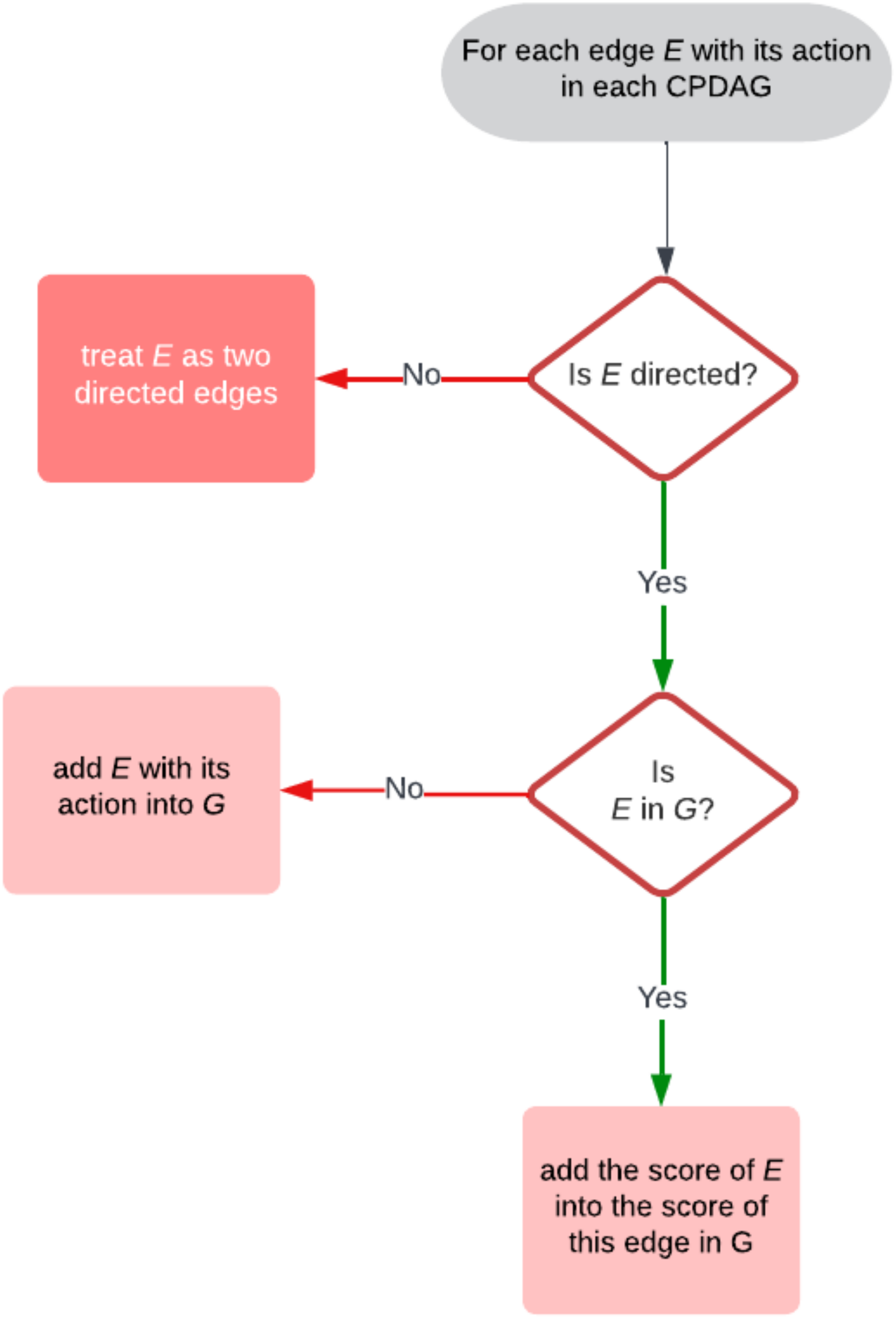
For each step, for each edge

Select the action whose dataset gives the highest score

Methodology

Casual DAG Maker - first stage

- Input: a set of CPDAG-Score pairs *S*
- Output: A graph *G* where there are multiple edges between two nodes
- Add the set of CPDAGs to G one by one.



For each edge *E* with its action in each CPDAG
Is *E* directed?
No
treat *E* as two directed edges
Yes
Is *E* in *G*?
No
add *E* with its action into *G*
Yes
add the score of *E* into the score of this edge in G

Methodology

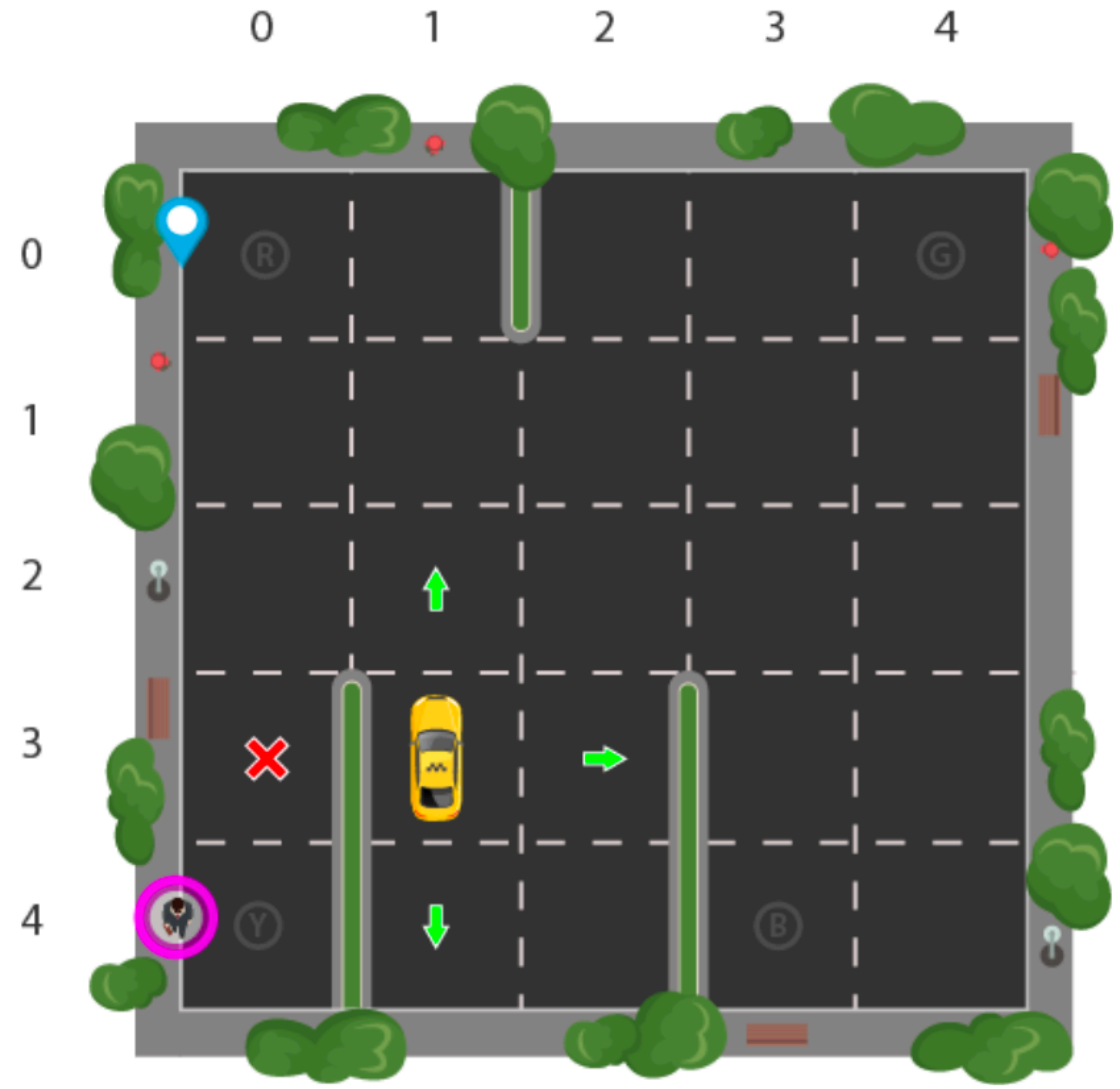
Casual DAG Maker
- second stage

- Input: the graph *G* gotten from previous stage
- Output: A DAG with its AM
- Threshold parameter *H:* 0.1
- Get the final causal DAG by merging all the edges between two variables into one unique directed edge.

Environments and Implementation

Taxi Problem - Environment

- 5 × 5 grid world.
- Taxi cannot move across obstacles.
- 4 special locations: R, G, Y, B.
- 6 Actions: move north/ east/ west, pick up/ drop off passenger.
- Goal: drive the passenger to destination successfully



Environments and Implementation

Taxi Problem - Implementation

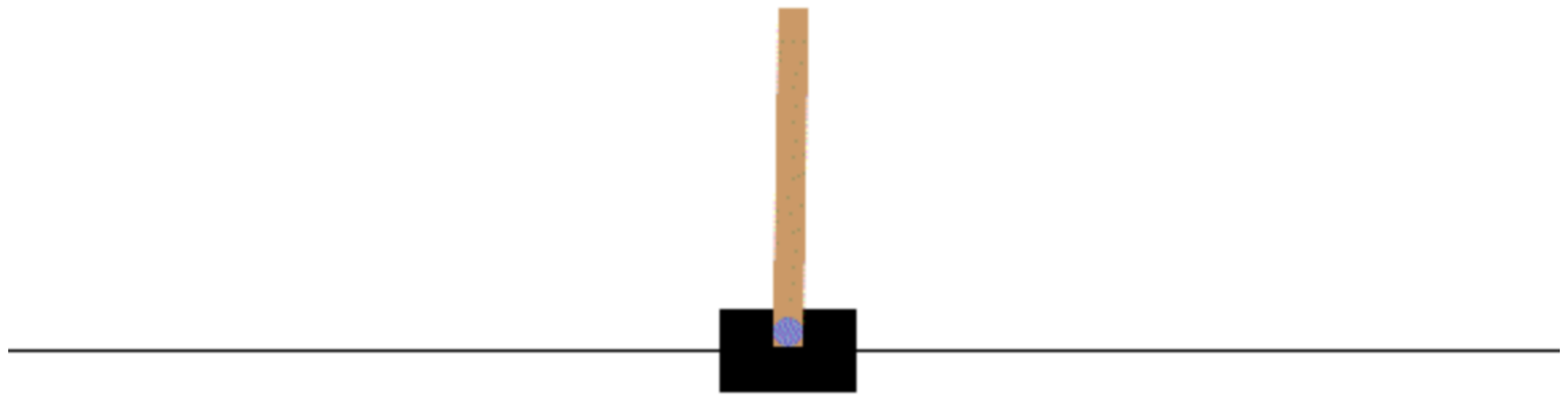
- Train the agent by Q-learning algorithm.
- Run the game for 100 episodes with the trained agent.
- Get a data set with 4 columns: taxi location, passenger location, destination location, action.
- Final DAG produced: 3 × 3 matrix in form of:

$$\begin{bmatrix} taxi\_position & passenger\_position & destination\_location \\ passenger\_position & \cdots & \\ destination\_location & \cdots & \end{bmatrix}$$

Environments and Implementation

CartPole Problem - Environment

- A pole attached to a cart moving along a frictionless track.
- 2 actions:
  1. push cart to the left,
  2. push cart to the right.
- Task: to keep the pole balanced for as long as possible.



Environments and Implementation

CartPole Problem - Implementation

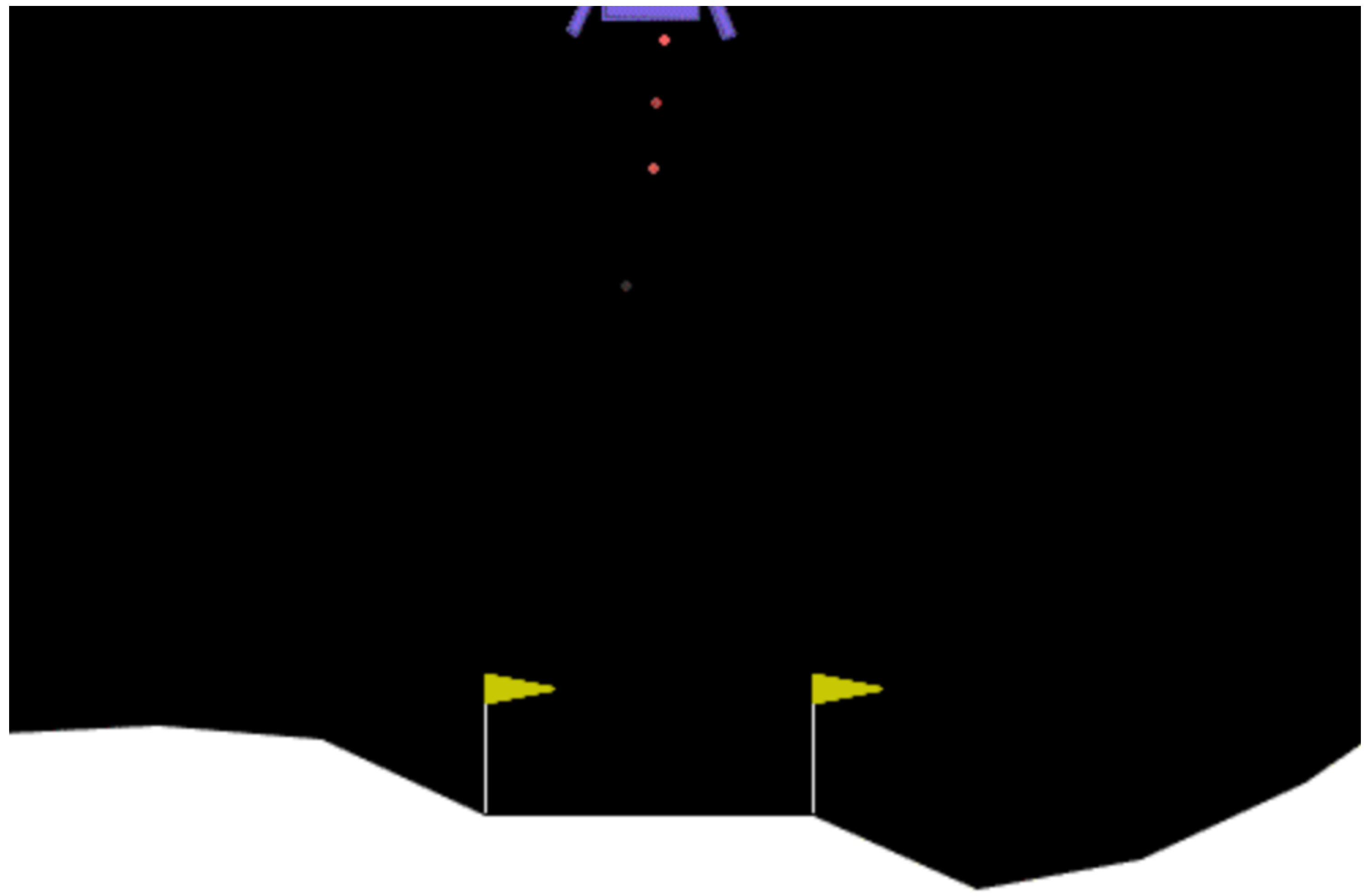
- Train the agent by REINFORCE algorithm.
- Run the game for 100 episodes with the trained agent.
- Get a dataset with 5 columns: cart velocity, cart position, pole angle, pole angular velocity, action.
- The final DAG produced: 4 × 4 matrix in form of:

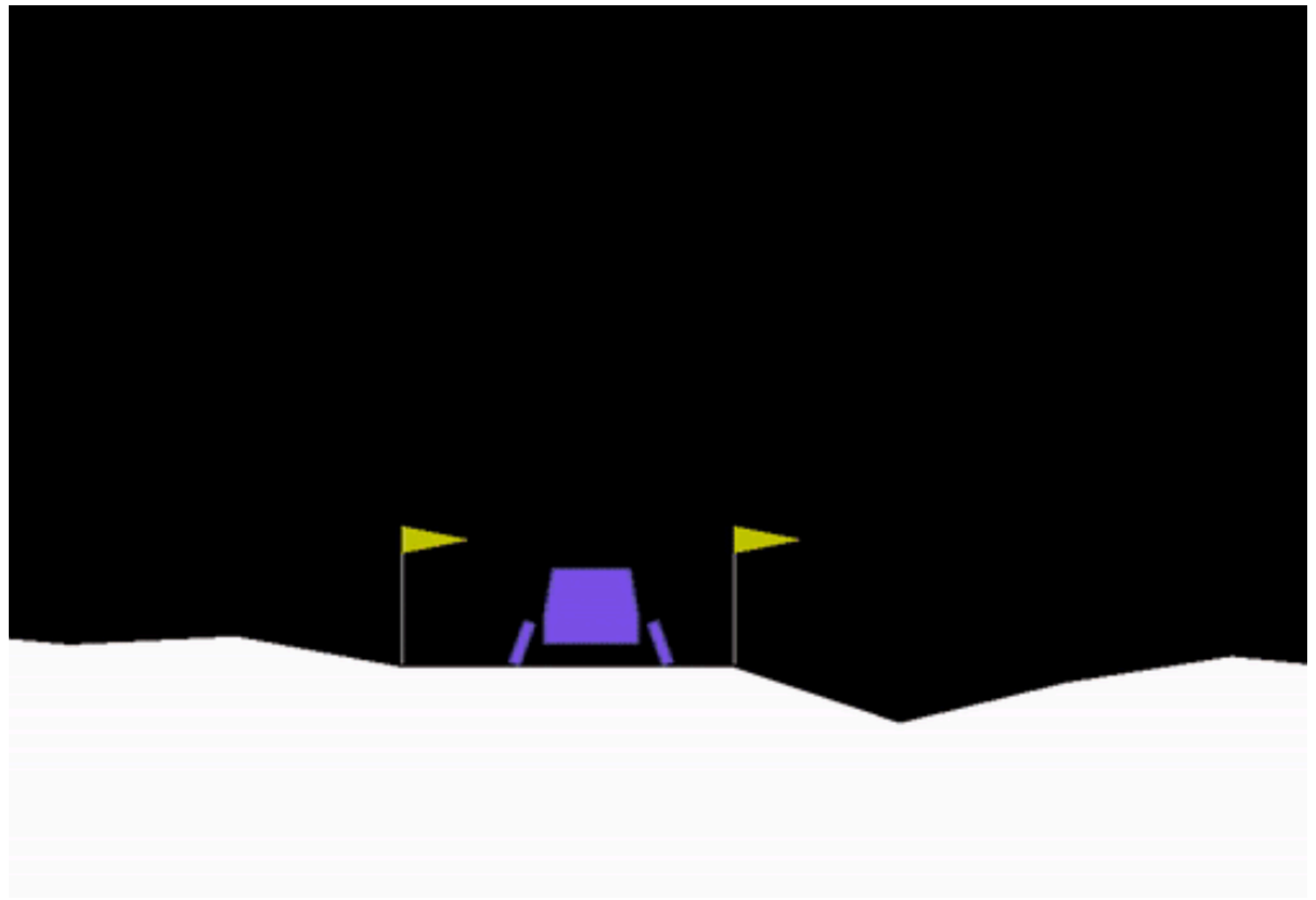
$$\begin{bmatrix} cart\_velocity & cart\_position & pole\_angle & pole\_angular\_velocity \\ cart\_position & \cdots & & \\ pole\_angle & \cdots & & \\ pole\_angular\_velocity & \cdots & & \end{bmatrix}$$

Environments and Implementation

LunarLand Problem - Environment

- Task: Land the LunarLander safely in the landing pad.
- 4 Actions:
  1. do nothing
  2. fire left orientation engine
  3. fire main engine
  4. fire right orientation engine





Environments and Implementation

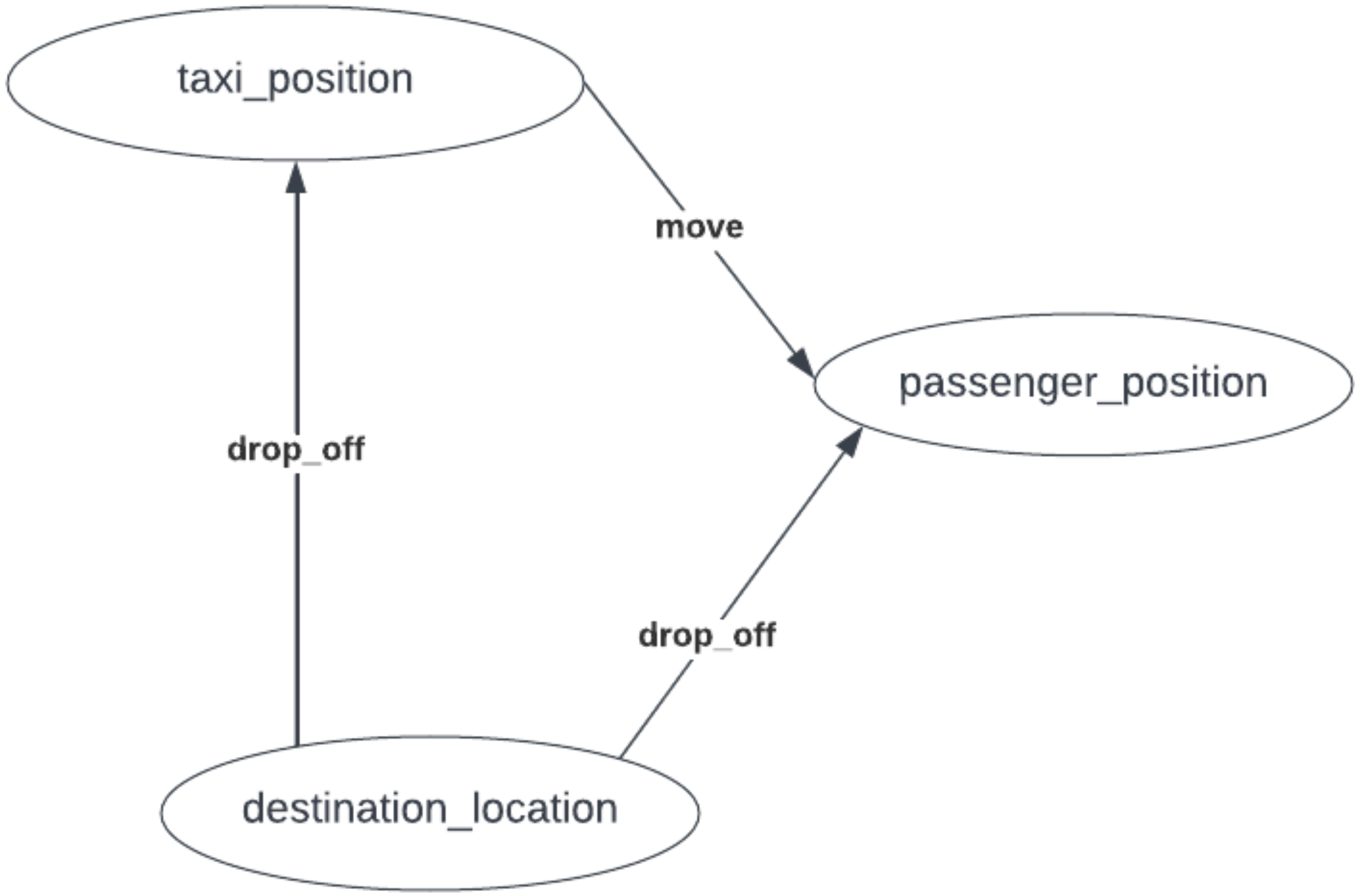
LunarLand Problem - Implementation

- Train the agent by Deep Q-Network algorithm.
- Run the game for 100 episodes with the trained agent.
- Get a dataset with 7 columns: lander xCoor, lander yCoor, lander xV, lander yV, lander angle, lander angularV, total reward, action.
- The final DAG produced: 6 × 6 matrix in form of:

$$\begin{bmatrix} xCoor & yCoor & xVelocity & yVelocity & angle & angularVelocity \\ yCoor & \cdots & & & & \\ xVelocity & \cdots & & & & \\ yVelocity & \cdots & & & & \\ angle & \cdots & & & & \\ angularVelocity & \cdots & & & & \end{bmatrix}$$

Experimental Results

Taxi Problem - Causal DAG with AM



Experimental Results

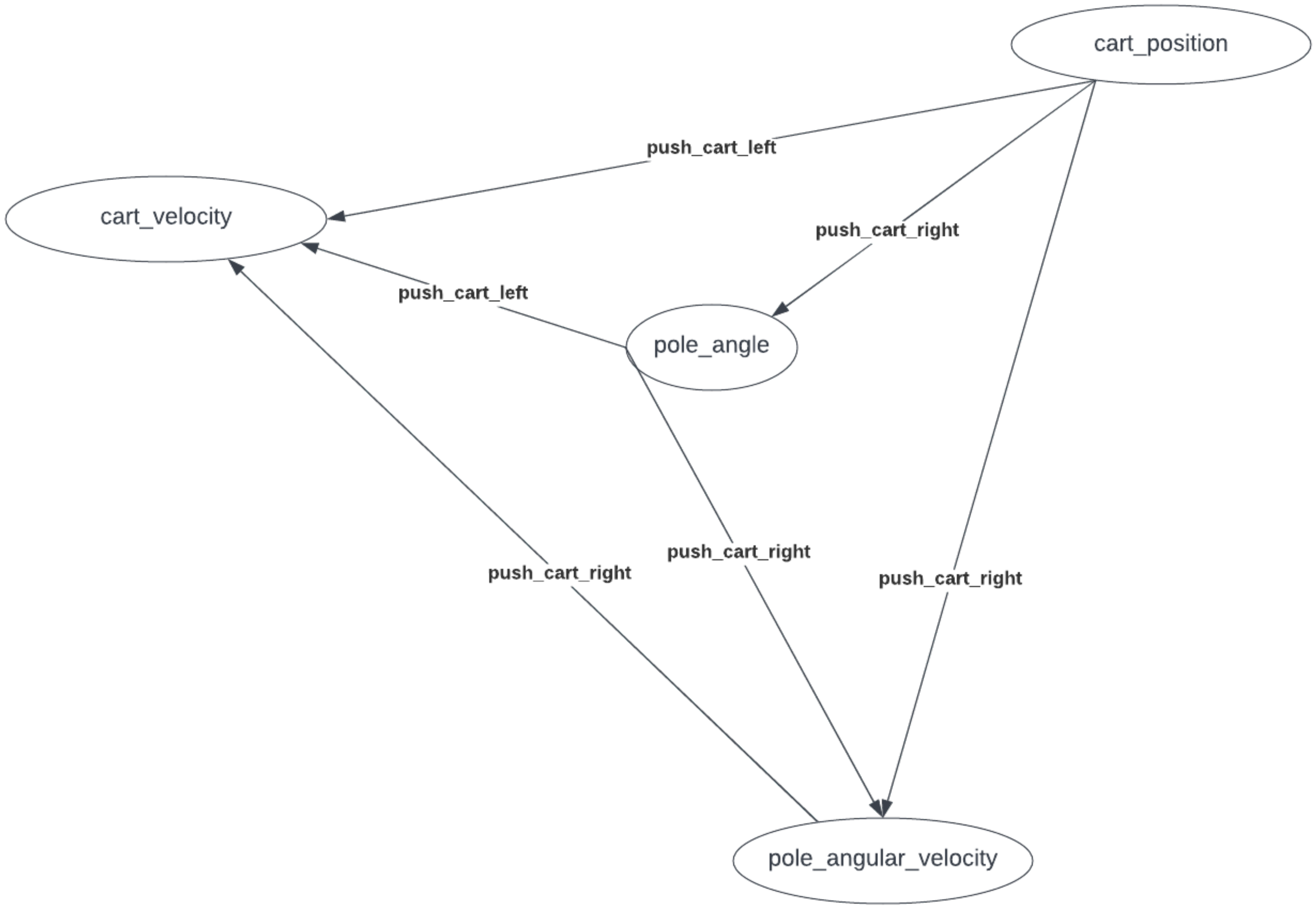
Taxi Problem - Explanations

- ”Why?” Explanation: Before taxi pick up the passenger, taxi does “move” action because the goal is to get passenger location.

- ”Why not?” Explanation: Before taxi pick up the passenger, taxi does “move” action, not action “drop off” because it is more desirable to do action ”move” to change taxi position, as the goal is to get passenger location.

Experimental Results

CartPole Problem - Causal DAG with AM



Experimental Results

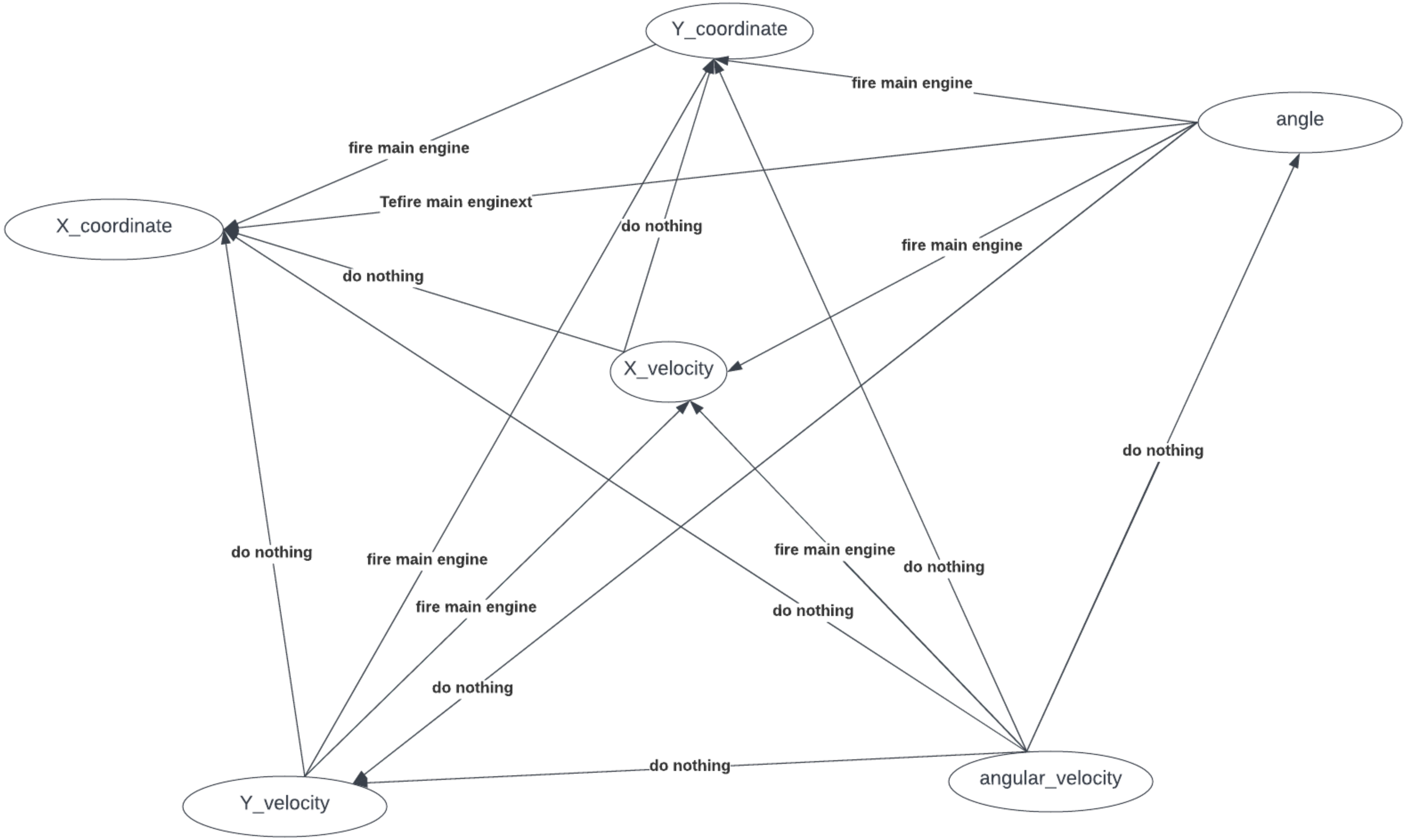
CartPole Problem - Explanations

- “Why?” explanation: The force pushes cart to the right because the goal is to change cart position that depends on pole angular velocity/ pole angle.

- “Why not?” explanation: The force pushes cart to the left, but not right because it is more desirable to do action ”push cart to the left”, to have more “pole angular velocity”/ “pole angle”, as the goal is to have ”cart position”.

Experimental Results

LunarLand Problem - Causal DAG with AM



Experimental Results

LunarLand Problem - Explanations

- why perform ”do nothing” action:

1. Because the goal is to change the coordinate of the lander in x.
2. Because the goal is to change the coordinate of the lander in x, that depends on the coordinates of the lander in y.
3. Because the goal is to change the coordinate of the lander in x, which is influenced by its linear velocities in x, the coordinate of the lander in y, that depends on its linear velocities in y.

Experimental Results

LunarLand Problem - Explanations

- why perform “fire main engine” action:

  1. Because the goal is to increase the coordinate of the lander in x-axis.
  2. Because the goal is to increase the coordinate of the lander in x-axis that depends on the coordinate of the lander in y-axis.
  3. Because the goal is to increase the coordinate of the lander in x-axis that depends on its linear velocities in x-axis.

Experimental Results

LunarLand Problem - Explanations

- why perform “do nothing” action but not “fire main engine” action:

  1. Because it is more desirable to do action do nothing to have more the coordinate of the lander in y-axis, in order to have more its linear velocity in x-axis, as the goal is to have the coordinate of the lander in x-axis.

  2. Because it is more desirable to do action do nothing to have more its linear velocity in x-axis, as the goal is to have the coordinate of the lander in x-axis.

Experimental Results

LunarLand Problem - Explanations

- why perform “fire main engine” action but not “do nothing” action:

1. Because it is more desirable to do action fire main engine to have less its linear velocity in y-axis in order to have more its angle, as the goal is to have the coordinate of the lander in x-axis.
2. Because it is more desirable to do action fire main engine, to have more the coordinate of the lander in y-axis, as the goal is to have the coordinate of the lander in x-axis.
3. Because it is more desirable to do action fire main engine, to have more its linear velocity in x-axis, to have less its linear velocity in y-axis, as the goal is to have the coordinate of the lander in x-axis.

Evaluation

Achievements

- Causal DAG along with its action matrix can be generated successfully
- meaningful explanations can be produced as well.

Evaluation

Future Work

- DAG generated from the environment with fewer variables could be more reliable.
- Our designed algorithm might treat correlation between two variables as causal relation.
- a computational evaluation is still needed.

Any questions?