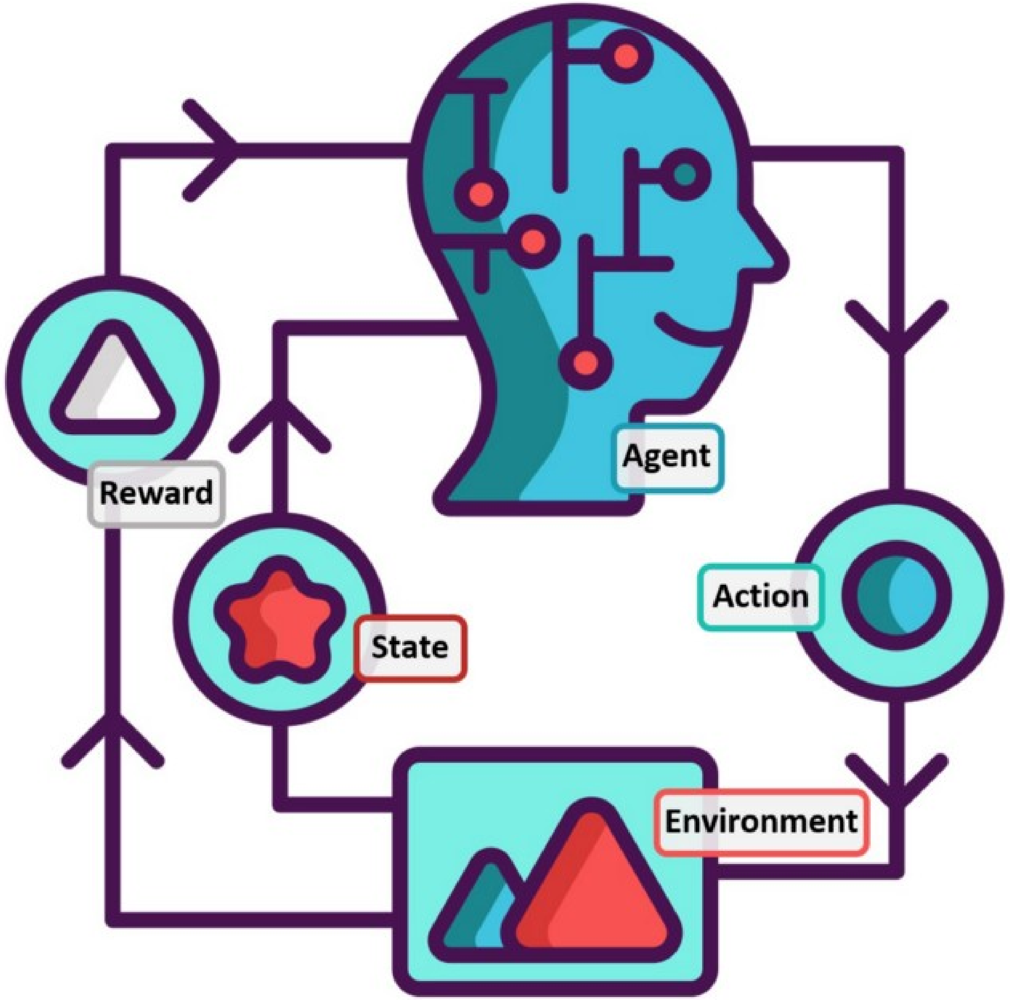
AIMLAI Workshop
2023

Predicate-based explanation of an RL agent via action importance

Léo Saulières

Martin C. Cooper – Florence Bannay


Agent
Reward
State
Action
Environment

History-Explanation based on Predicates (HXP)

Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate *d*

History-Explanation based on Predicates (HXP)

Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate d

Question: Which actions were important to ensure that d was achieved, given the agent's policy π?

History-Explanation based on Predicates (HXP)

Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate d

Question: Which actions were important to ensure that d was achieved, given the agent's policy π?

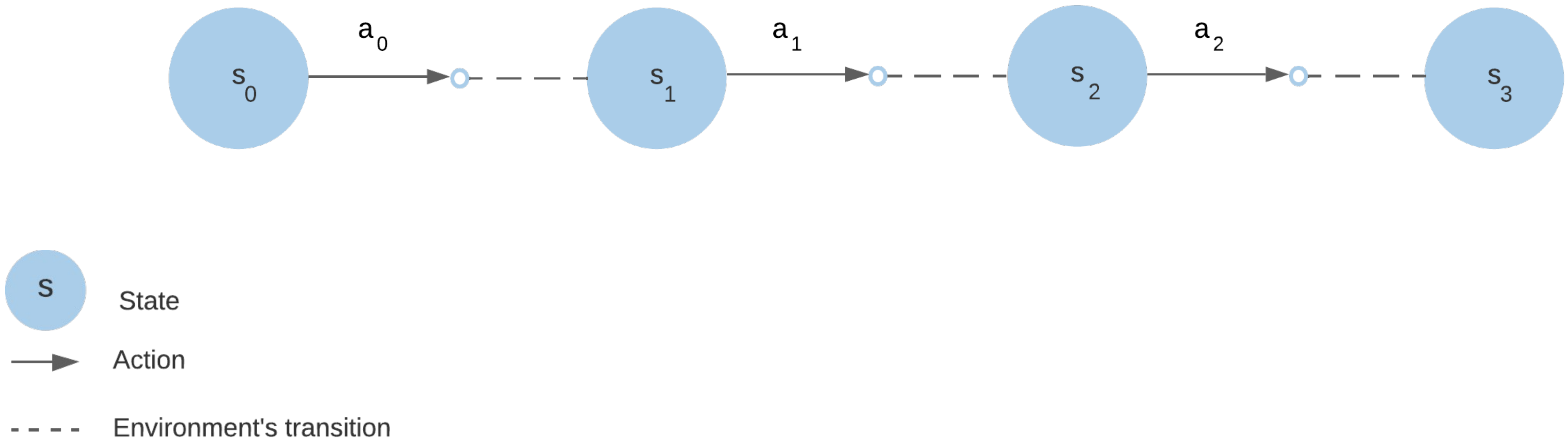
Idea: Compute the *action importance score* for each state-action (s,a) in the length-k history h

History-Explanation based on Predicates (HXP)

Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate *d*

Question: Which actions were important to ensure that *d* was achieved, given the agent's policy π?

Idea: Compute the *action importance score* for each state-action *(s,a)* in the length-*k* history *h*



History-Explanation based on Predicates (HXP)

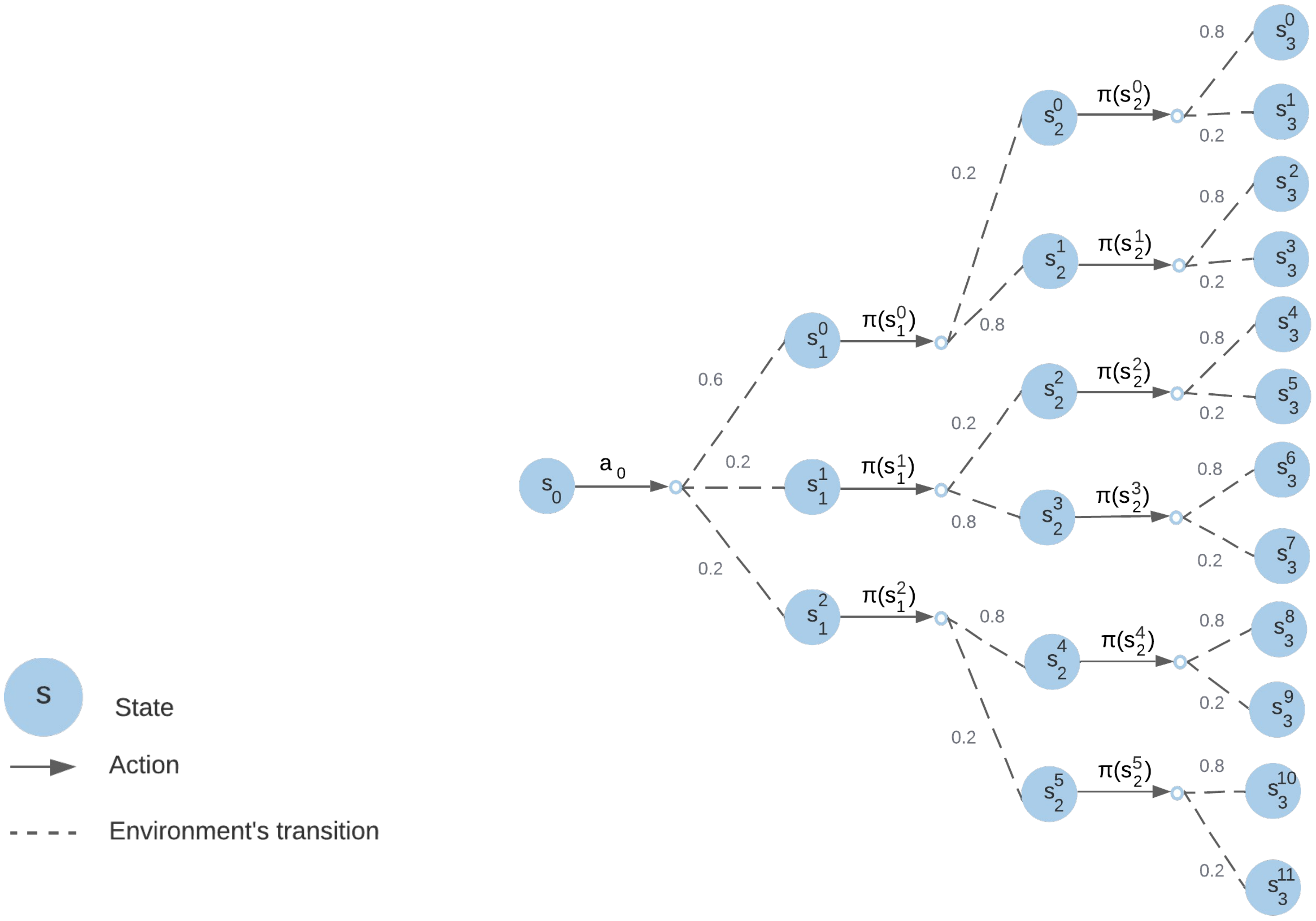
Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate d

Question: Which actions were important to ensure that d was achieved, given the agent's policy π?

Idea: Compute the *action importance score* for each state-action (s,a) in the length-k history h

- Generate the set of length-k scenarios starting by doing a from s
  Use of π and the transition function

History-Explanation based on Predicates (HXP)



History-Explanation based on Predicates (HXP)

Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate d

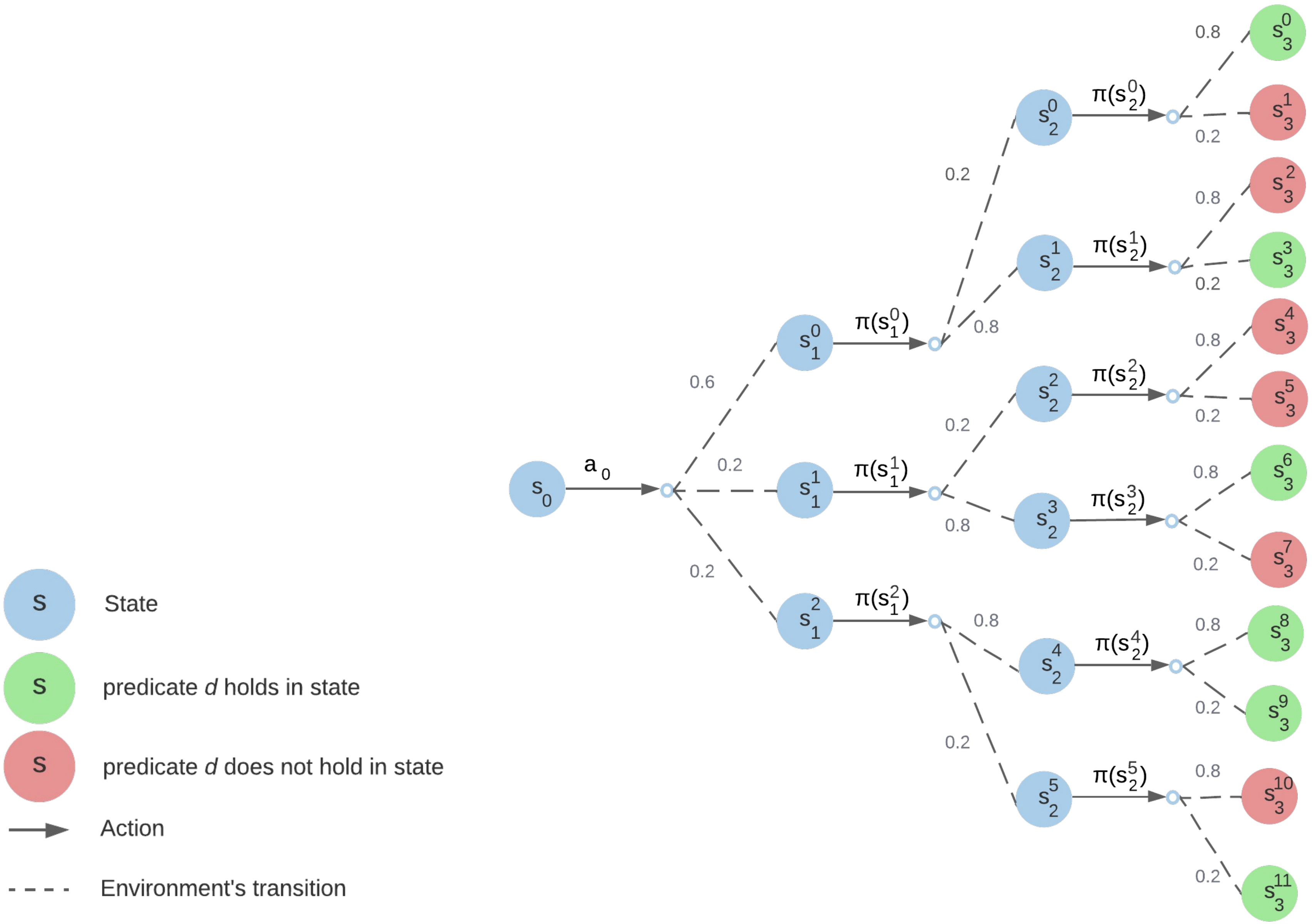
Question: Which actions were important to ensure that d was achieved, given the agent's policy π?

Idea: Compute the *action importance score* for each state-action (s,a) in the length-k history h

- Generate the set of length-k scenarios starting by doing a from s
  Use of π and the transition function

- Compute the probability to reach a final state at horizon k which respects d (utility)

History-Explanation based on Predicates (HXP)



History-Explanation based on Predicates (HXP)

Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate *d*

Question: Which actions were important to ensure that *d* was achieved, given the agent's policy π?

Idea: Compute the *action importance score* for each state-action *(s,a)* in the length-*k* history *h*

- Generate the set of length-*k* scenarios starting by doing *a* from *s*
  Use of π and the transition function
- Compute the probability to reach a final state at horizon *k* which respects *d* (utility)
  Utility lies in range [0, 1]
- Repeat the process for each action $a' \in A(s) \setminus \{a\}$

History-Explanation based on Predicates (HXP)

Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate d

Question: Which actions were important to ensure that d was achieved, given the agent's policy π?

Idea: Compute the *action importance score* for each state-action (s,a) in the length-k history h

- Generate the set of length-k scenarios starting by doing a from s
  Use of π and the transition function

- Compute the probability to reach a final state at horizon k which respects d (utility)
  Utility lies in range [0, 1]

- Repeat the process for each action $a' \in A(s) \setminus \{a\}$

The action importance score of an action a, from a state s in the history is the difference between the utility of a and the average utility of any other action $a' \in A(s) \setminus \{a\}$

History-Explanation based on Predicates (HXP)

Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate *d*

Question: Which actions were important to ensure that *d* was achieved, given the agent's policy π?

Idea: Compute the *action importance score* for each state-action *(s,a)* in the length-*k* history *h*

> *The action importance score of an action a, from a state s in the history is the difference between the utility of a and the average utility of any other action $a' \in A(s) \setminus \{a\}$*

Action importance score lies in range [-1;1]

History-Explanation based on Predicates (HXP)

Goal: Explain past agent’s interactions with the environment (history) through the prism of a predicate d

Question: Which actions were important to ensure that d was achieved, given the agent’s policy π?

Idea: Compute the *action importance score* for each state-action (s,a) in the length-k history h

The action importance score of an action a, from a state s in the history is the difference between the utility of a and the average utility of any other action $a' \in A(s) \setminus \{a\}$

Action importance score lies in range [-1;1]

Problem: Computationnaly expensive method (#W[1]-hard)

History-Explanation based on Predicates (HXP)

Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate *d*

Question: Which actions were important to ensure that *d* was achieved, given the agent's policy π?

Idea: Compute the *action importance score* for each state-action *(s,a)* in the length-*k* history *h*

> *The action importance score of an action a, from a state s in the history is the difference between the utility of a and the average utility of any other action* $a' \in A(s) \setminus \{a\}$

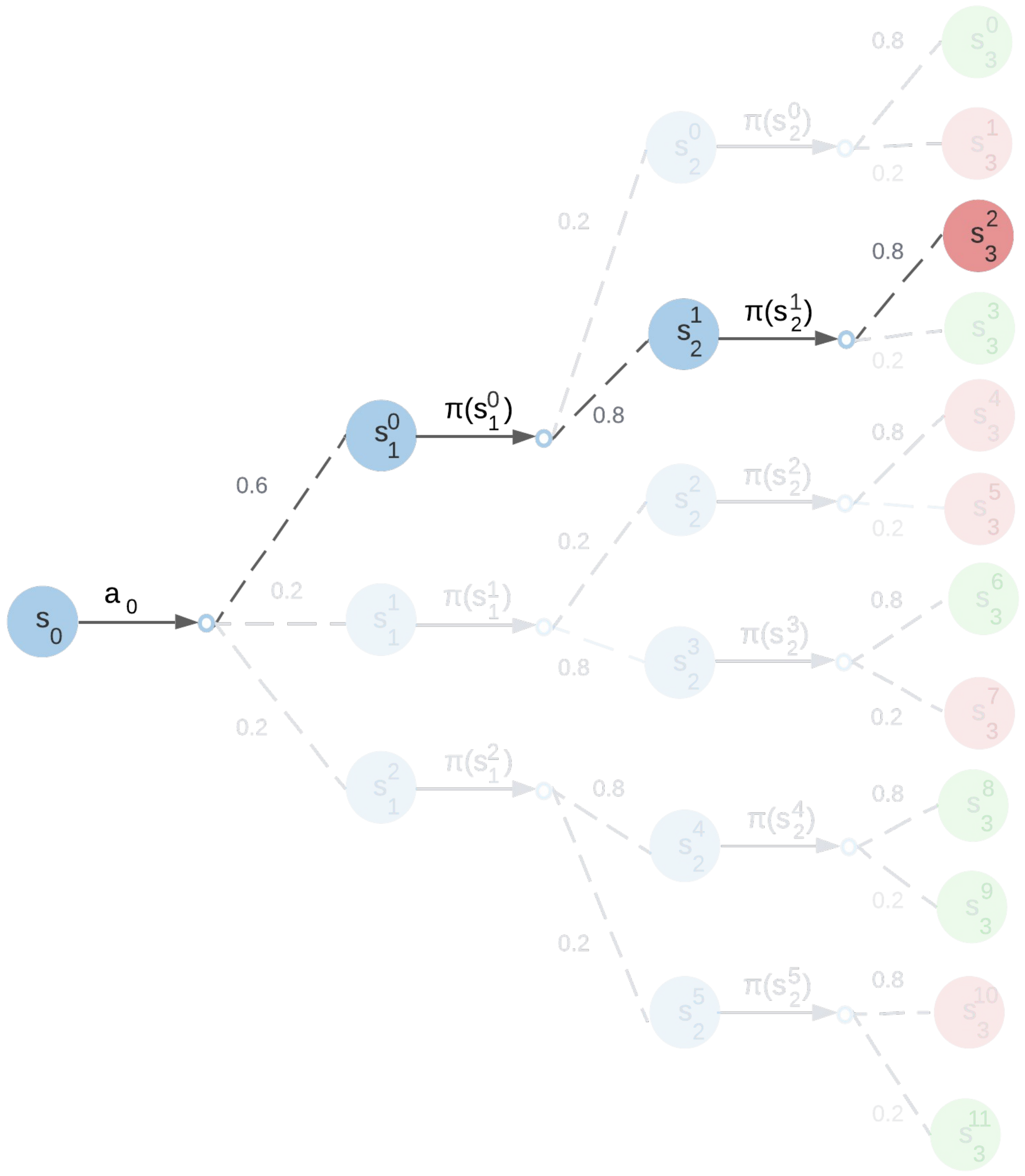
Action importance score lies in range [-1;1]

Problem: Computationnaly expensive method (#W[1]-hard)

Solution: Generate a large range of scenarios, but not the unlikely ones

Most probable transition at each time-step

History-Explanation based on Predicates (HXP)



- s — State
- s — predicate *d* holds in state
- s — predicate *d* does not hold in state
- Action
- Environment's transition

History-Explanation based on Predicates (HXP)

Goal: Explain past agent's interactions with the environment (history) through the prism of a predicate *d*

Question: Which actions were important to ensure that *d* was achieved, given the agent's policy π?

Idea: Compute the *action importance score* for each state-action *(s,a)* in the length-*k* history *h*

> *The action importance score of an action a, from a state s in the history is the difference between the utility of a and the average utility of any other action* $a' \in A(s) \setminus \{a\}$

Action importance score lies in range [-1;1]

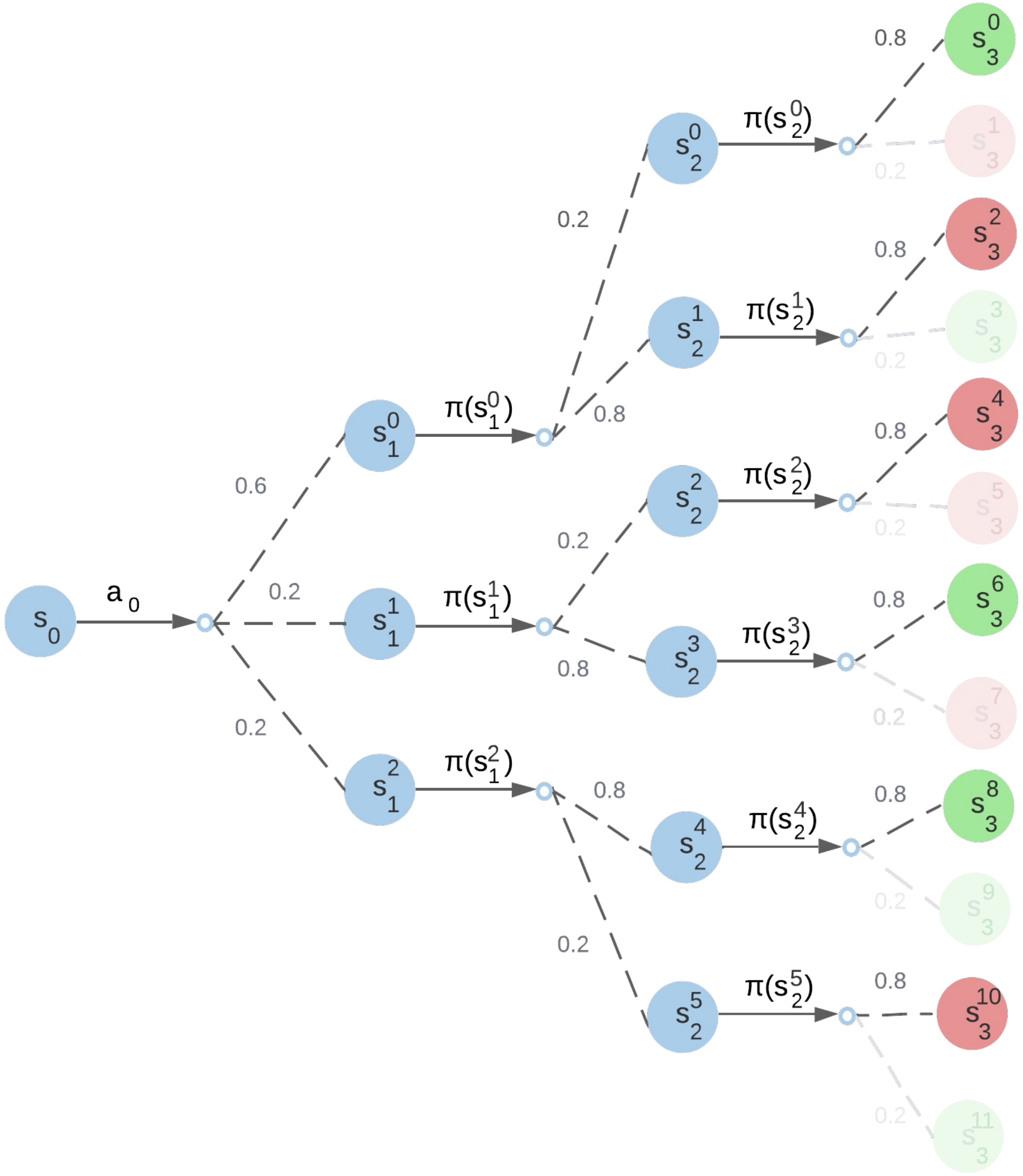
Problem: Computationnaly expensive method (#W[1]-hard)

Solution: Generate a large range of scenarios, but not the unlikely ones

Most probable transition at the *n* last time-step(s)

History-Explanation based on Predicates (HXP)

$n = 1$



S State

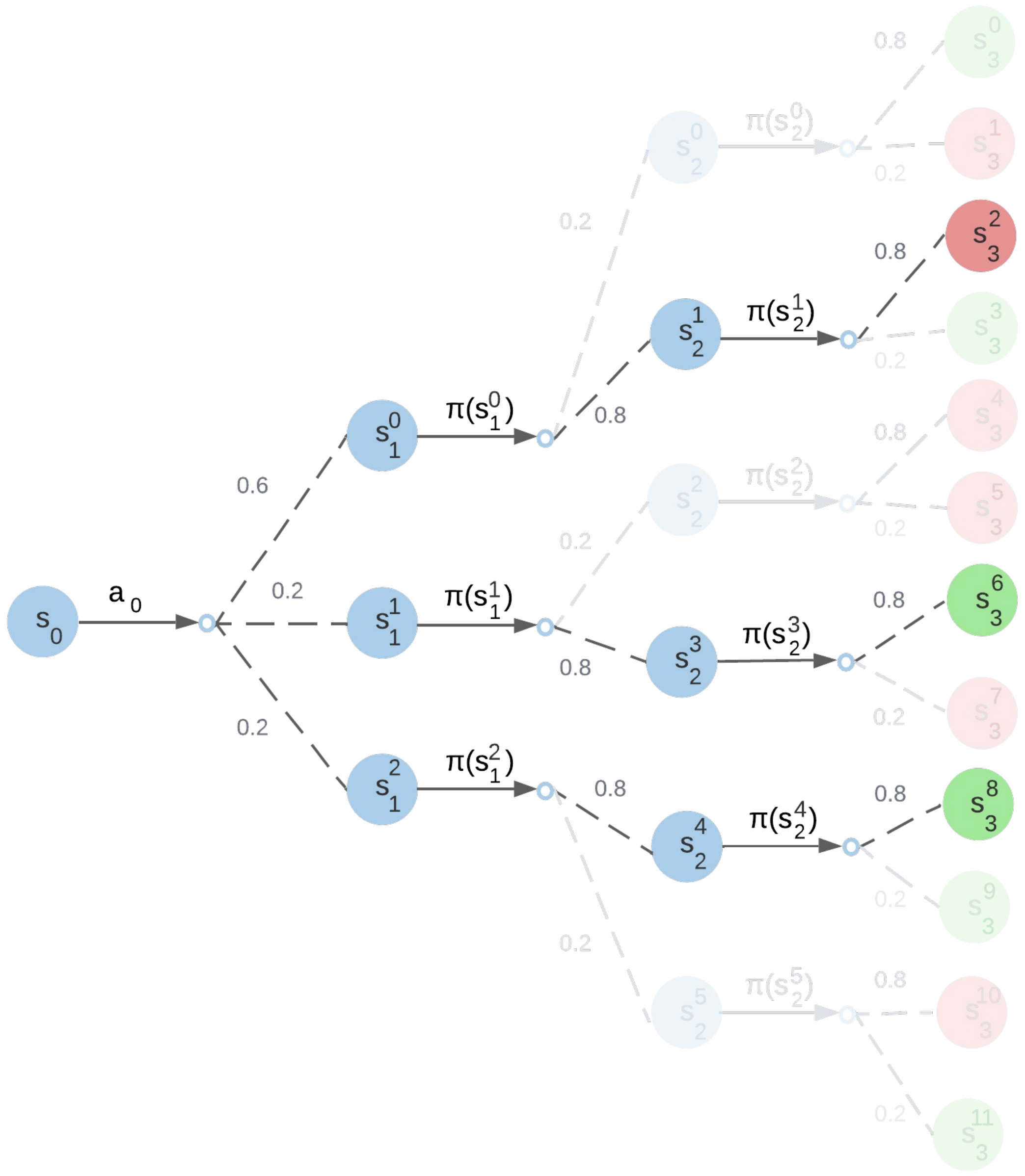
S predicate *d* holds in state

S predicate *d* does not hold in state

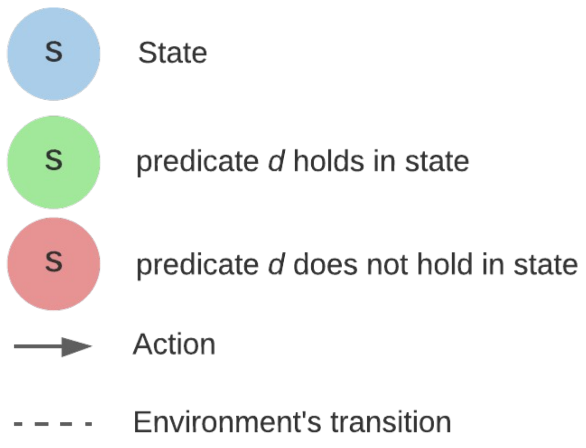
Action

Environment's transition

$n = 2$



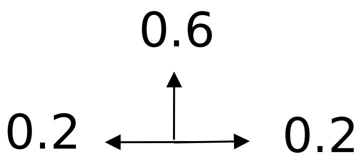
s0
a0
0.6
0.2
0.2
s1 0
s1 1
s1 2
π(s1 0)
π(s1 1)
π(s1 2)
0.2
0.8
0.2
0.8
0.8
0.2
s2 0
s2 1
s2 2
s2 3
s2 4
s2 5
π(s2 0)
π(s2 1)
π(s2 2)
π(s2 3)
π(s2 4)
π(s2 5)
0.8
0.2
s3 0
s3 1
s3 2
s3 3
s3 4
s3 5
s3 6
s3 7
s3 8
s3 9
s3 10
s3 11



s
State
s
predicate d holds in state
s
predicate d does not hold in state
Action
Environment's transition

Frozen Lake

Transition function (⬆)



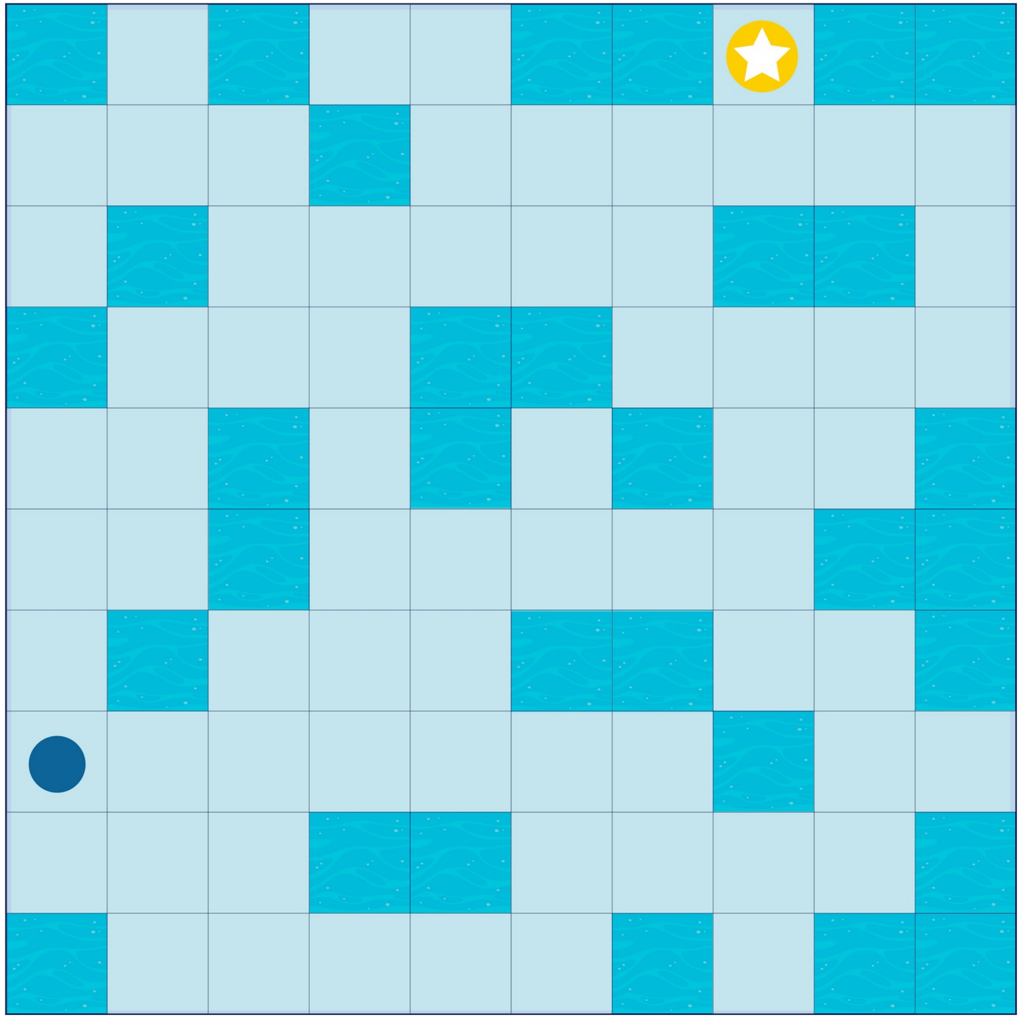
Actions ⬆⬇⬅➡

Reward function

- +1 in Goal position
- +0 otherwise

Algorithm Tabular Q-learning

Predicates *goal, holes, region*



Connect4

Transition function

Player 2's policy

Actions

Column number

Reward function

- +1 if win
- -1 if lose
- +0.5 if draw
- +0 otherwise

Algorithm

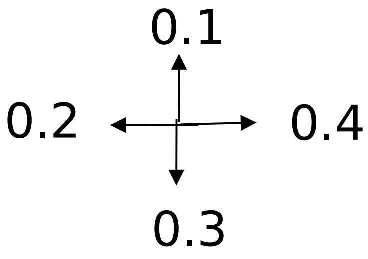
Deep Q Network (DQN)

Predicates

win, lose, 3 in a row, avoid 3 in a row, control mid-column

Drone Coverage

Transition function



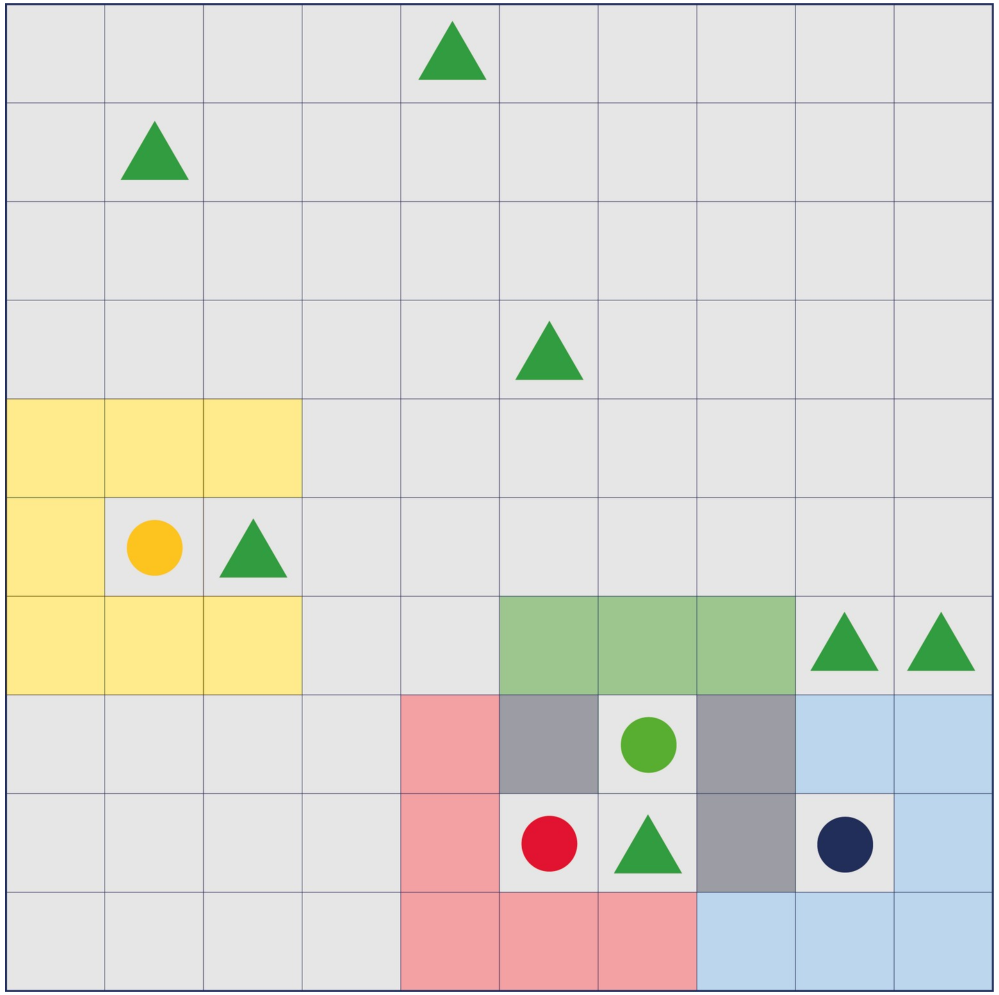
Actions

↑↓⟵⟶■

Reward function

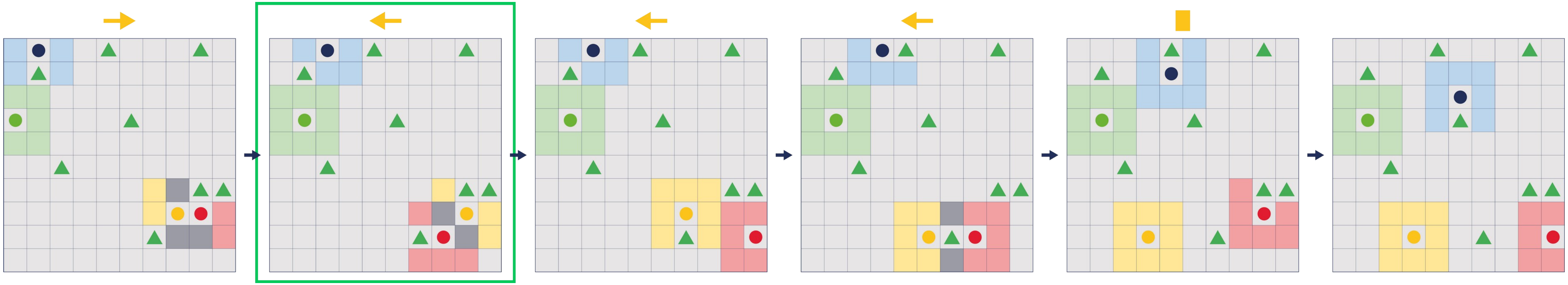
- +3 or +0.25 * |fc|
- -1 per drone in view range
- -3 in crash case

Algorithm Deep Q Network (DQN)



Predicates *(Local / Global) maximum reward, perfect cover, no drones, crash, region*

Local maximum reward



Time-step	0	1	2	3	4	Run-time (s)
Exh	-0.339	**0.475**	0.108	0.108	0.002	15.19
1L	-0.351	**0.488**	0.114	0.113	0.002	9.78
2L	-0.36	**0.506**	0.11	0.107	0.0	3.95
3L	-0.34	**0.498**	0.12	0.115	0.0	1.38
4L	-0.3	**0.45**	0.175	0.175	0.0	0.44

Similarity score

Goal: Compare two length-k vectors v_1, v_2 of action importance scores

Similarity score

Goal: Compare two length-k vectors v_1, v_2 of action importance scores

How ? L2 norm

Similarity score

Goal: Compare two length-k vectors v_1, v_2 of action importance scores

How ? L2 norm

Similarity score: inverse normalised L2 norm

$$\text{similarity}(v_1, v_2) = 1 - \frac{L2(v_1, v_2)}{2\sqrt{k}}$$

Overall Results

Average similarity scores of HXP

Problem		Exh-1L	Exh-2L	Exh-3L	Exh-4L
Frozen Lake		0.992	0.983	0.971	0.954
Drone Coverage	Local	0.991	0.981	0.974	0.961
	Global	0.992	0.983	0.977	0.967
Connect4		0.995	0.979	0.955	0.918

Overall Results

Average running time (in seconds) of HXP

Problem		Exh	1L	2L	3L	4L
Frozen Lake		0.006	0.005	0.003	0.002	0.001
Drone Coverage	Local	28.19	19.08	7.74	2.65	0.81
	Global	27.69	18.82	7.63	2.61	0.8
Connect4		21.51	20.49	6.51	1.58	0.33

Conclusion

HXP:

- Analyse past agent’s interactions with the environment:
  - Predicate-based approach
  - Action importance evaluation
- Approximate HXP to reduce computation time

Given a history, display to the user the most imporant action(s) and corresponding state(s) according to the achievement of a certain predicate

Action importance scores are computed with the use of the agent’s policy and transition function

Conclusion

HXP:

- Analyse past agent's interactions with the environment:
  - Predicate-based approach
  - Action importance evaluation
- Approximate HXP to reduce computation time

Limits:

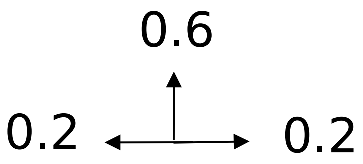
- Transition function must be known
- Trade-off between time saving and correctness of the scores generated
- Explain short histories

Future works:

- Explain long histories
- Additional information: most important *transition(s)*

Frozen Lake

Transition function (↑)



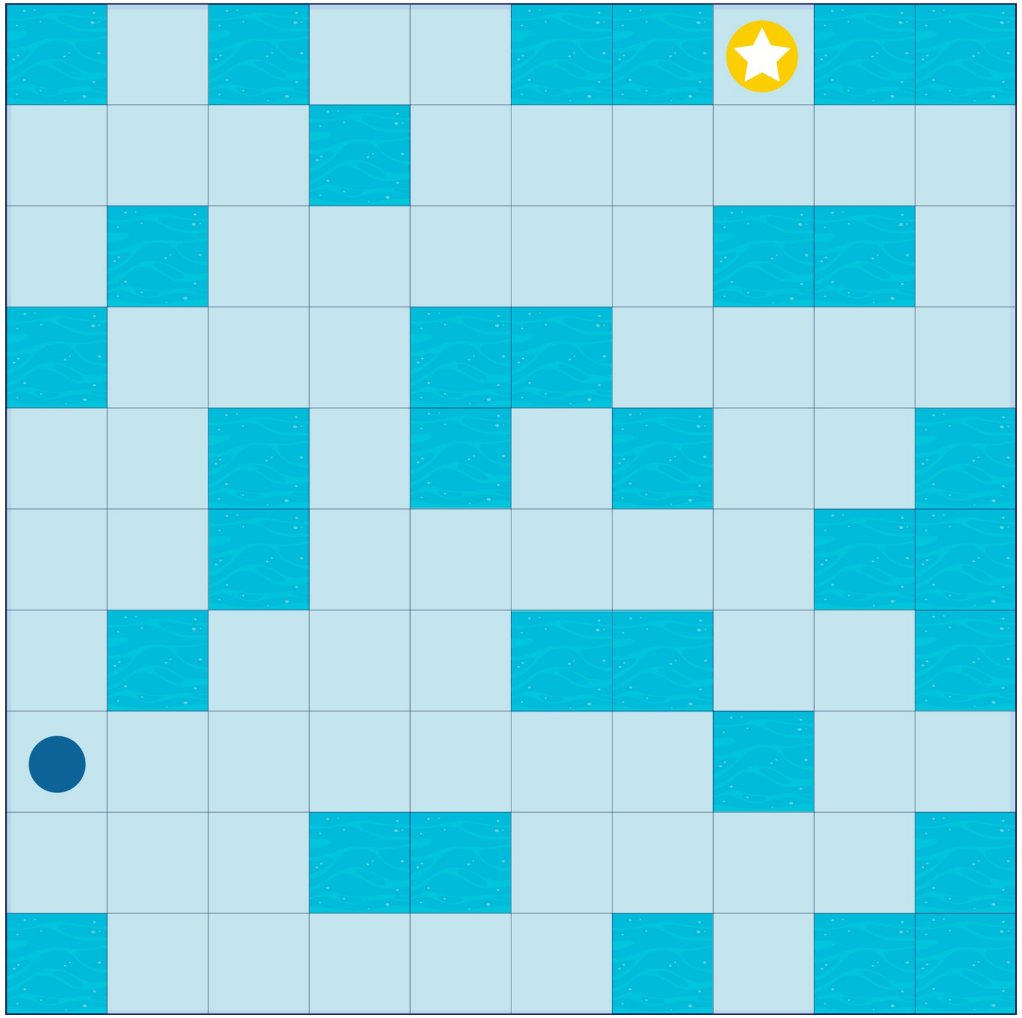
Actions ↑ ↓ ← →

Reward function

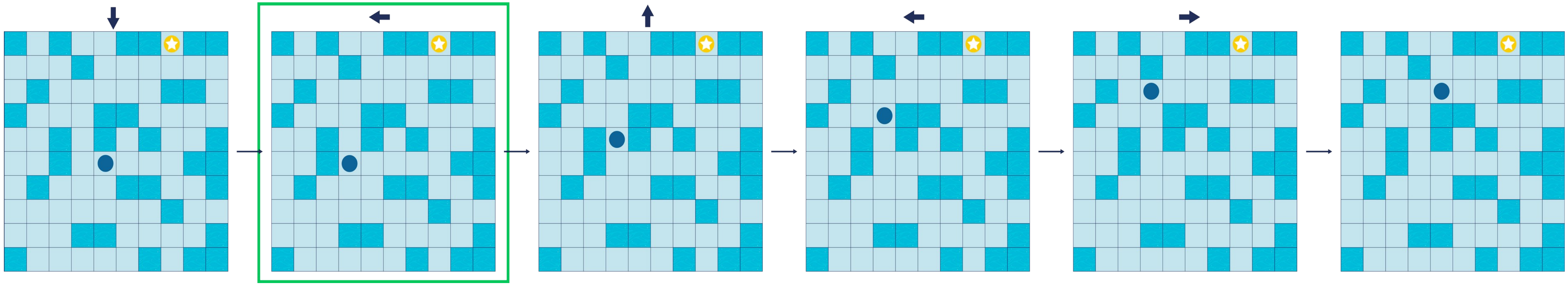
- +1 in Goal position
- +0 otherwise

Algorithm Tabular Q-learning

Predicates *goal, holes, region*



Holes



Time-step	0	1	2	3	4	Run-time (s)
Exh	-0.323	**0.315**	-0.262	-0.294	-0.119	0.025
1L	-0.34	**0.301**	-0.301	-0.303	-0.105	0.017
2L	-0.315	**0.379**	-0.317	-0.355	-0.109	0.014
3L	-0.387	**0.36**	-0.333	-0.373	-0.067	0.009
4L	-0.4	**0.467**	-0.467	-0.333	-0.067	0.008

Connect4

Transition function

Player 2's policy

Actions

Column number

Reward function

- +1 if win
- -1 if lose
- +0.5 if draw
- +0 otherwise

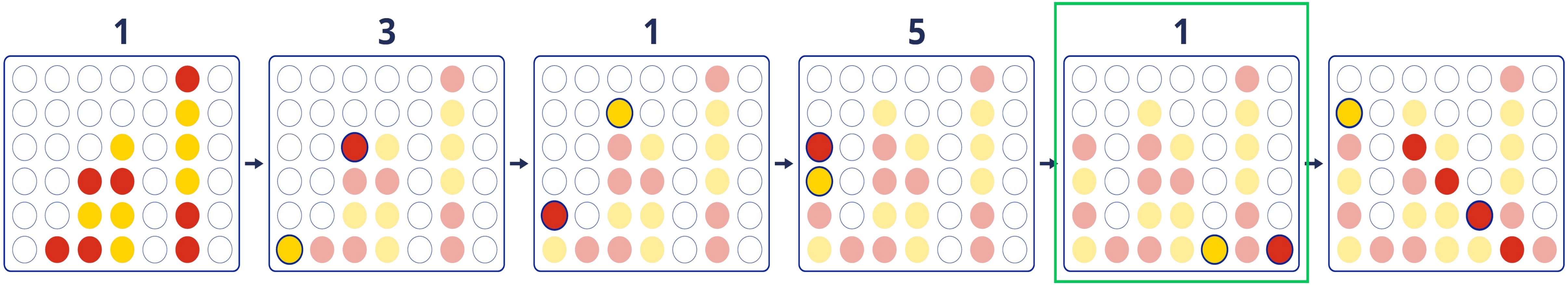
Algorithm

Deep Q Network (DQN)

Predicates

win, lose, 3 in a row, avoid 3 in a row, control mid-column

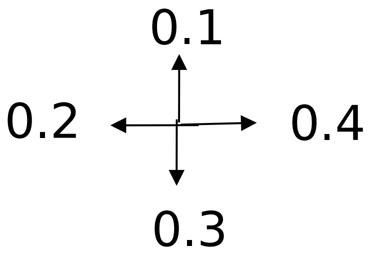
Lose



Time-step	0	1	2	3	4	Run-time (s)
Exh	-0.053	-0.082	-0.046	0.234	**0.256**	6.74
1L	-0.077	-0.074	-0.023	0.279	**0.32**	7.5
2L	-0.066	-0.061	-0.016	0.276	**0.349**	3.15
3L	-0.067	-0.046	0.04	0.286	**0.421**	0.96
4L	-0.167	-0.067	0.1	**0.5**	0.393	0.36

Drone Coverage

Transition function



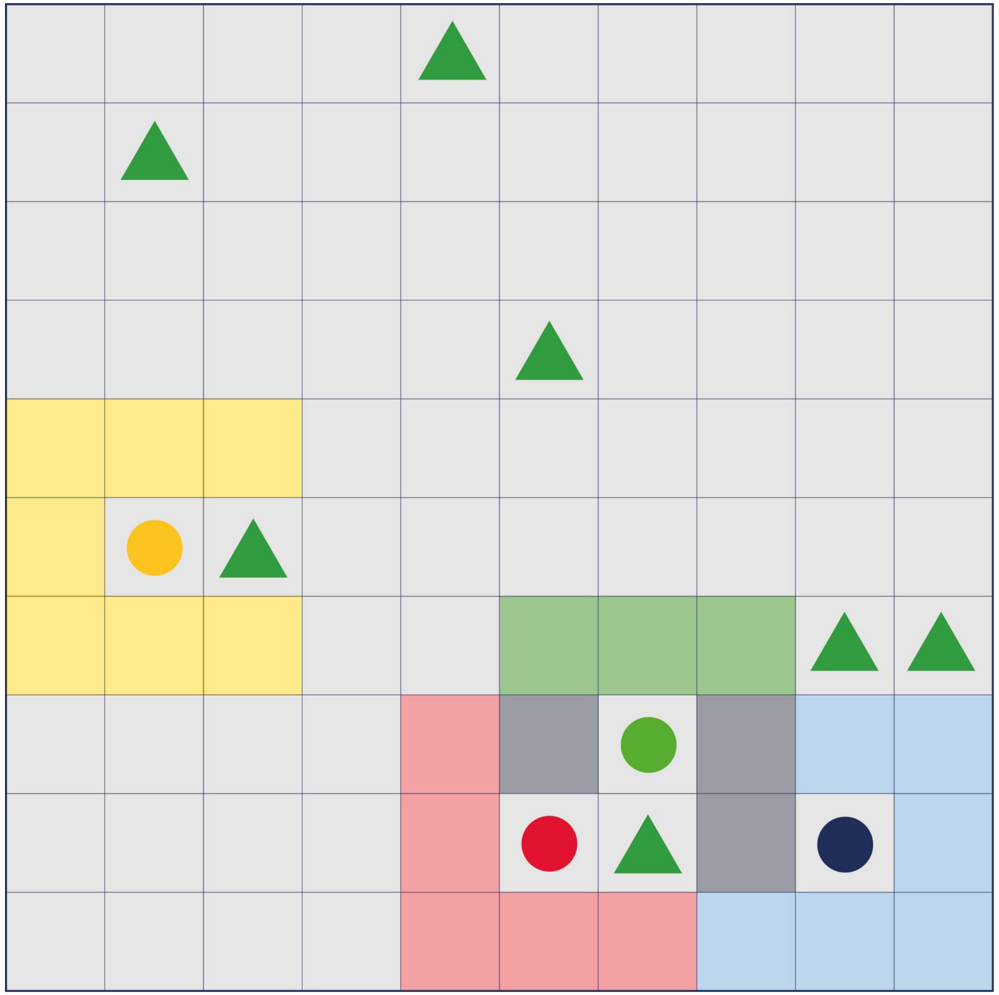
Actions

⬆⬇⬅➡■

Reward function

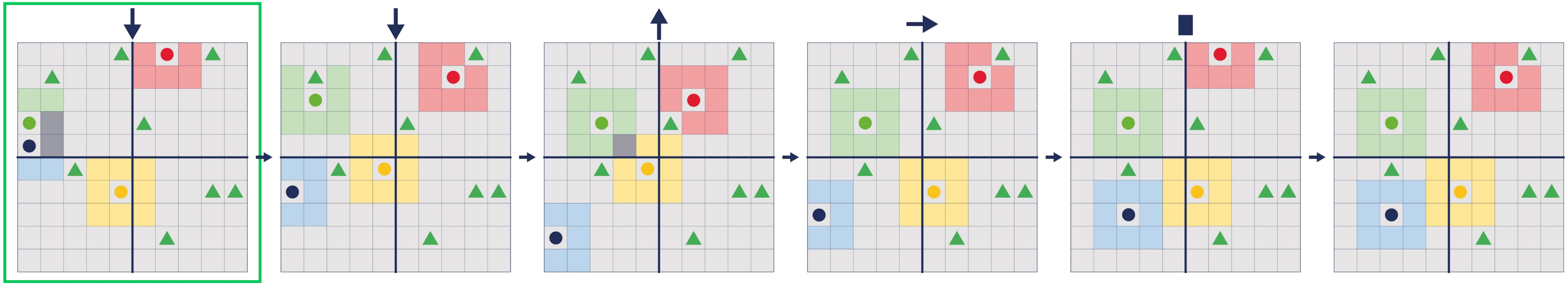
- +3 or +0.25 * |fc|
- -1 per drone in view range
- -3 in crash case

Algorithm Deep Q Network (DQN)



Predicates *(Local / Global) maximum reward, perfect cover, no drones, crash, region*

Global region



Time-step	0	1	2	3	4	Run-time (s)
Exh	**0.819**	0.025	0.0	0.0	0.005	21.22
1L	**0.826**	0.025	0.0	0.0	0.011	11.42
2L	**0.837**	0.025	0.0	0.0	0.0	4.62
3L	**0.86**	0.025	0.0	0.0	0.0	1.66
4L	**0.8**	0.025	0.0	0.0	0.0	0.53