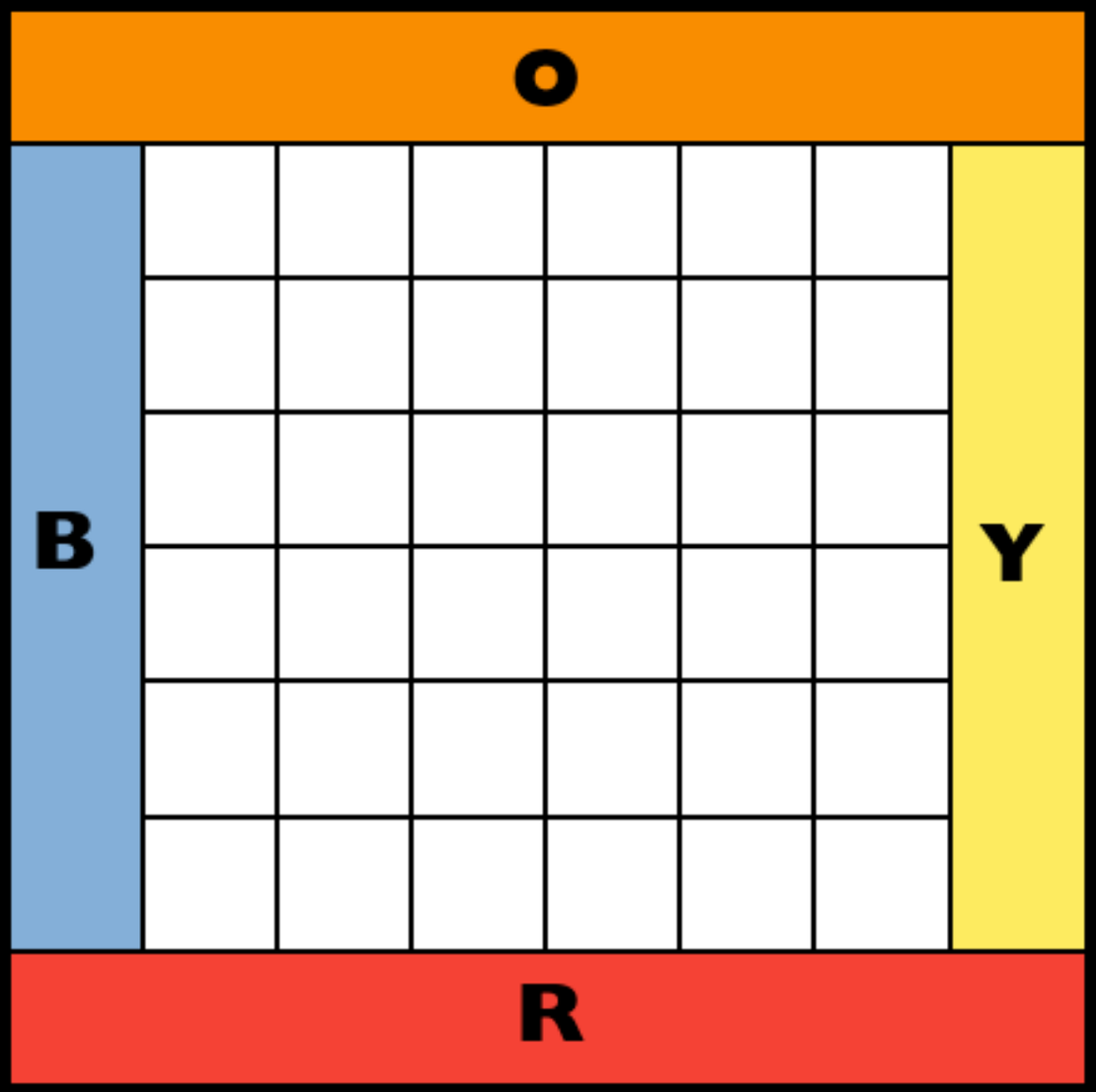
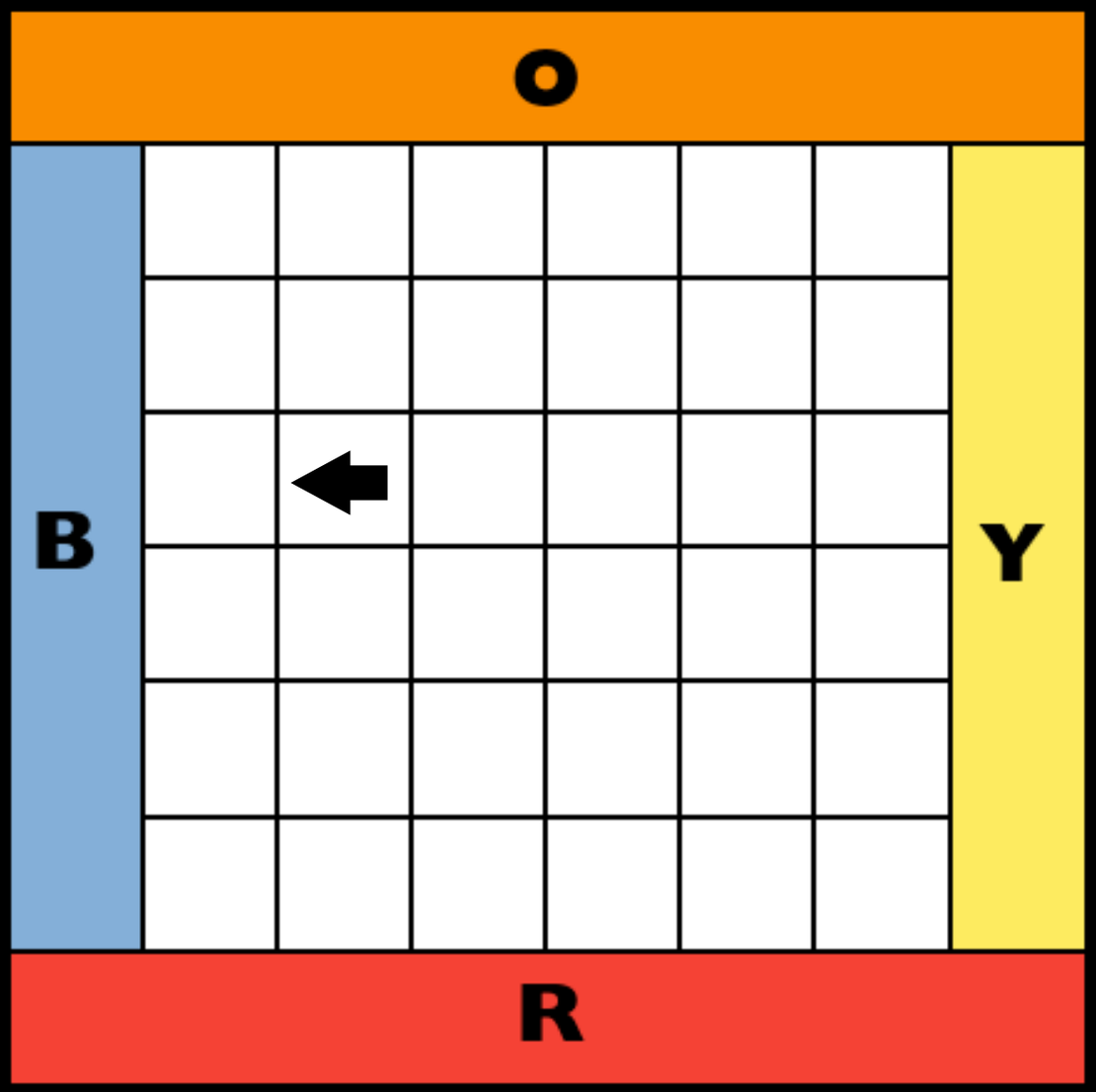
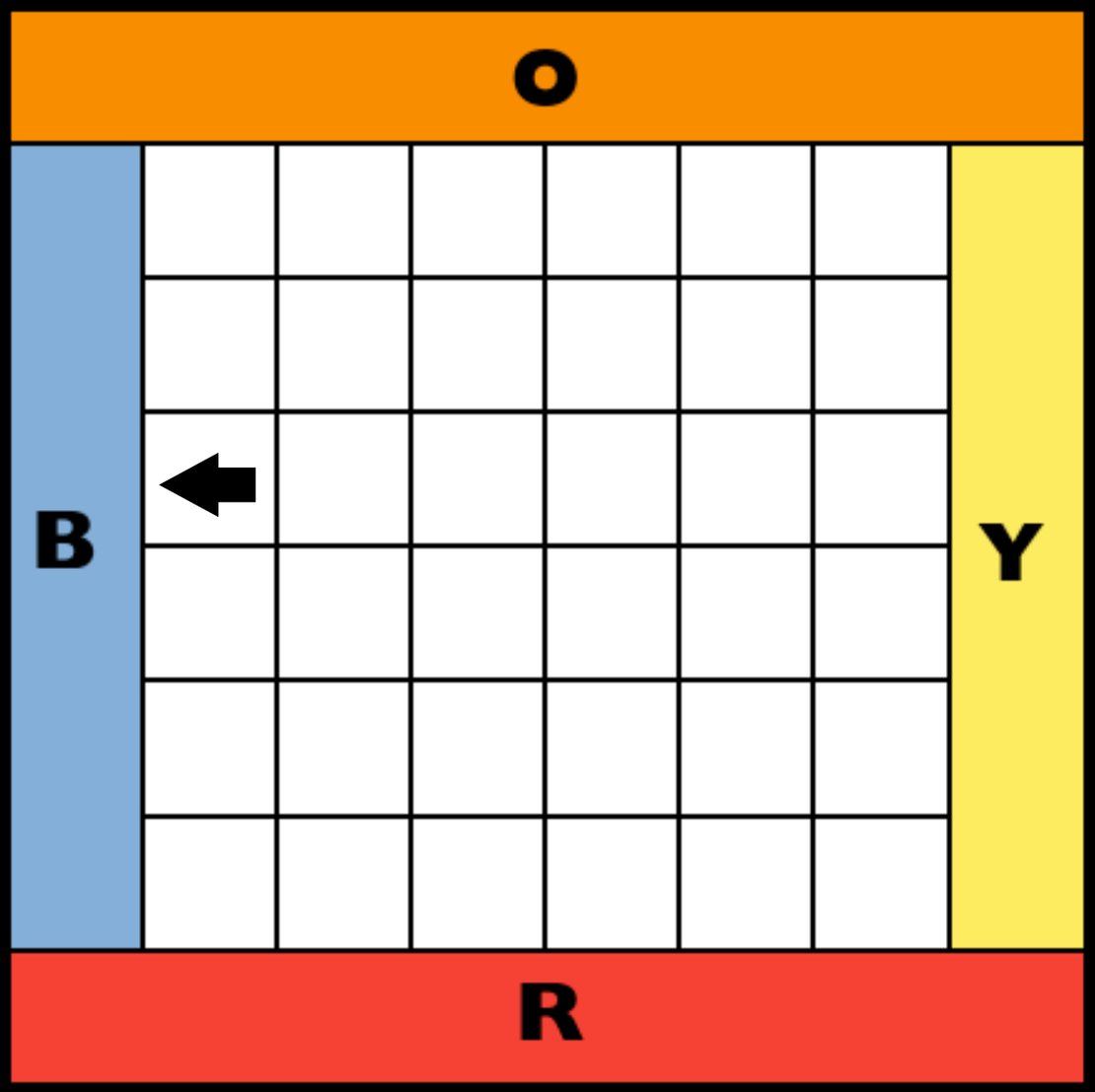
Learning approximate predictive models

Monica Dinculescu

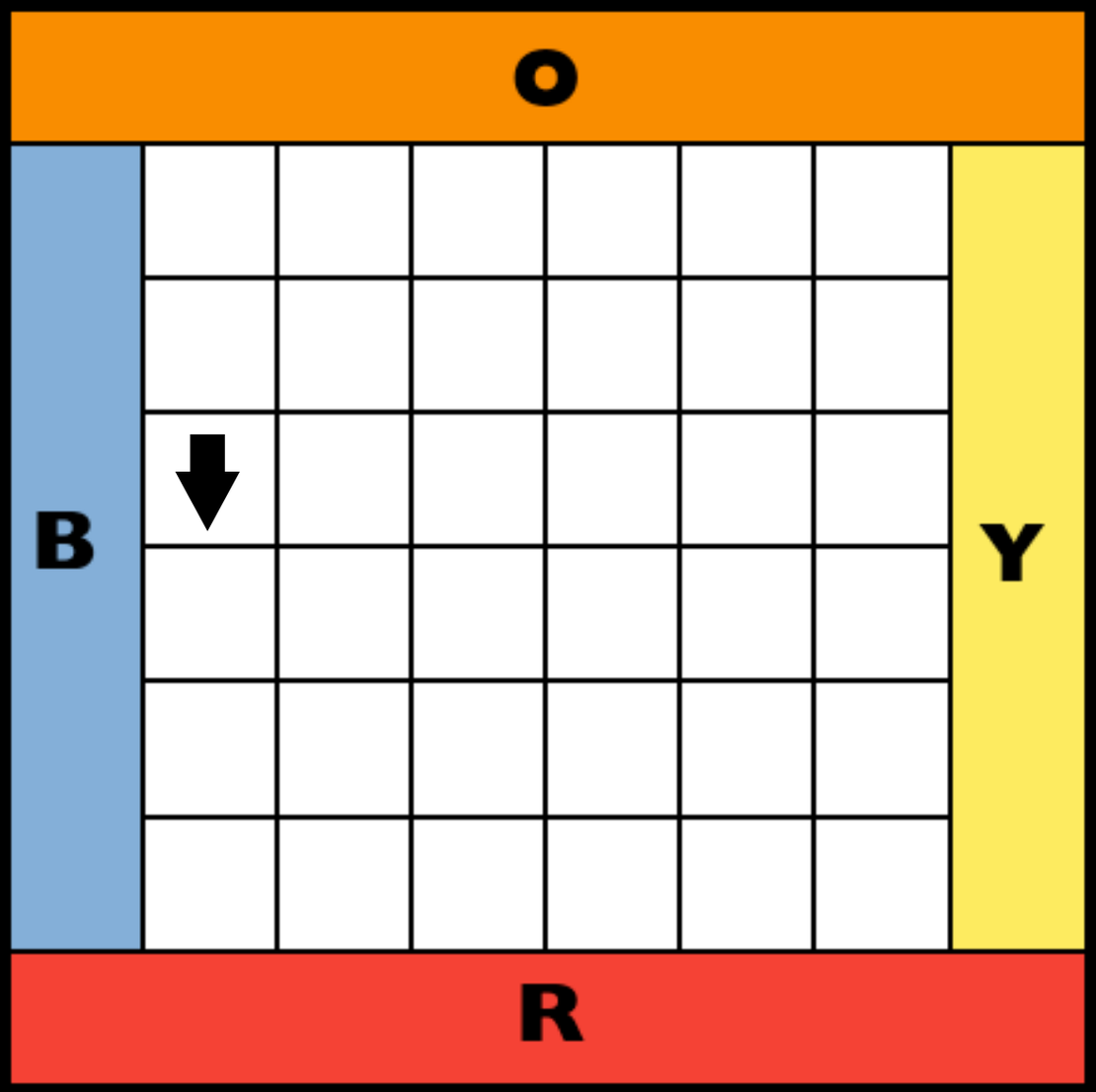


6x6x4 states
2 actions: Forward, Turn Left
Noise: action has no effect 5% of the time

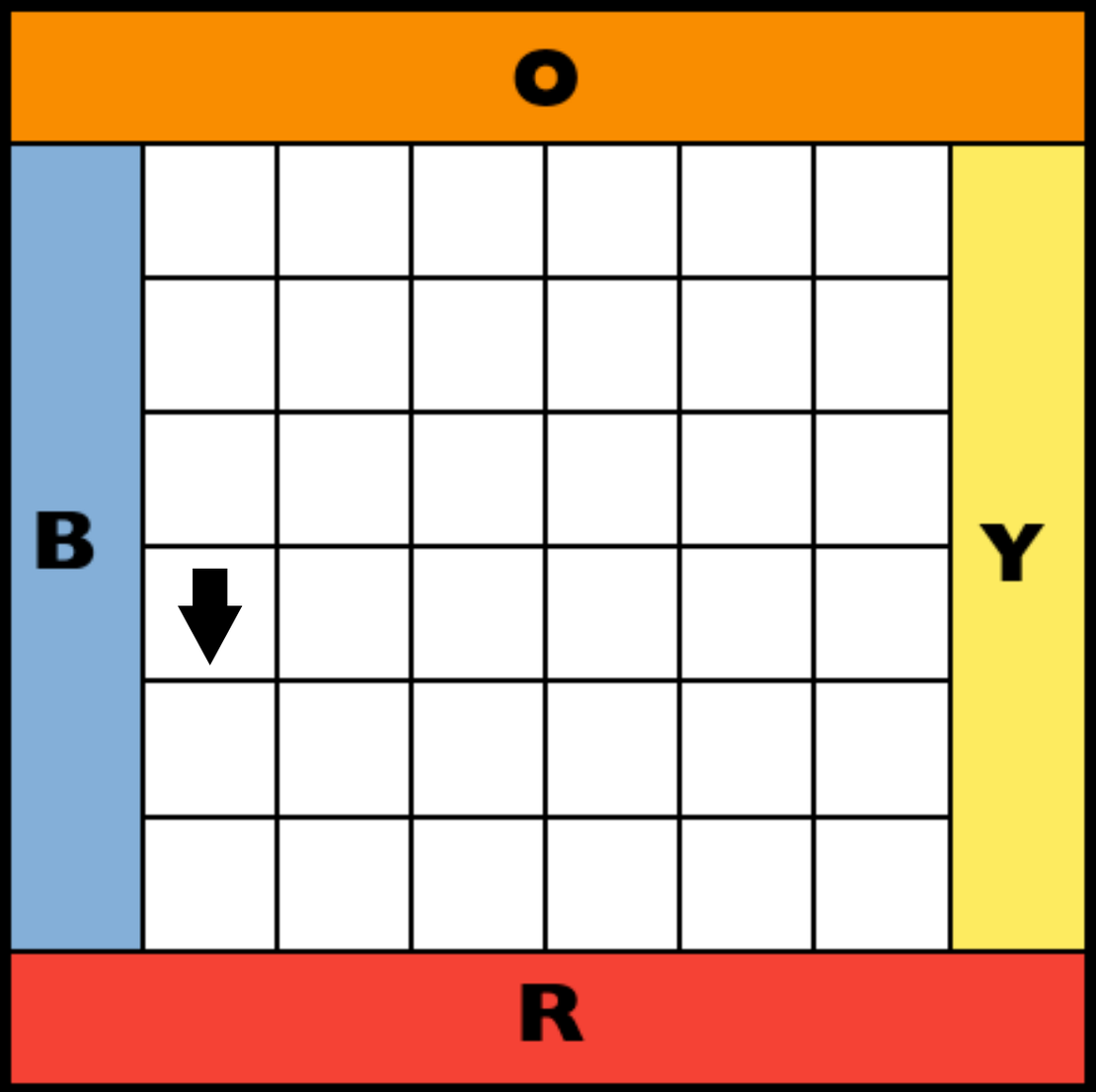

O
B
Y
R



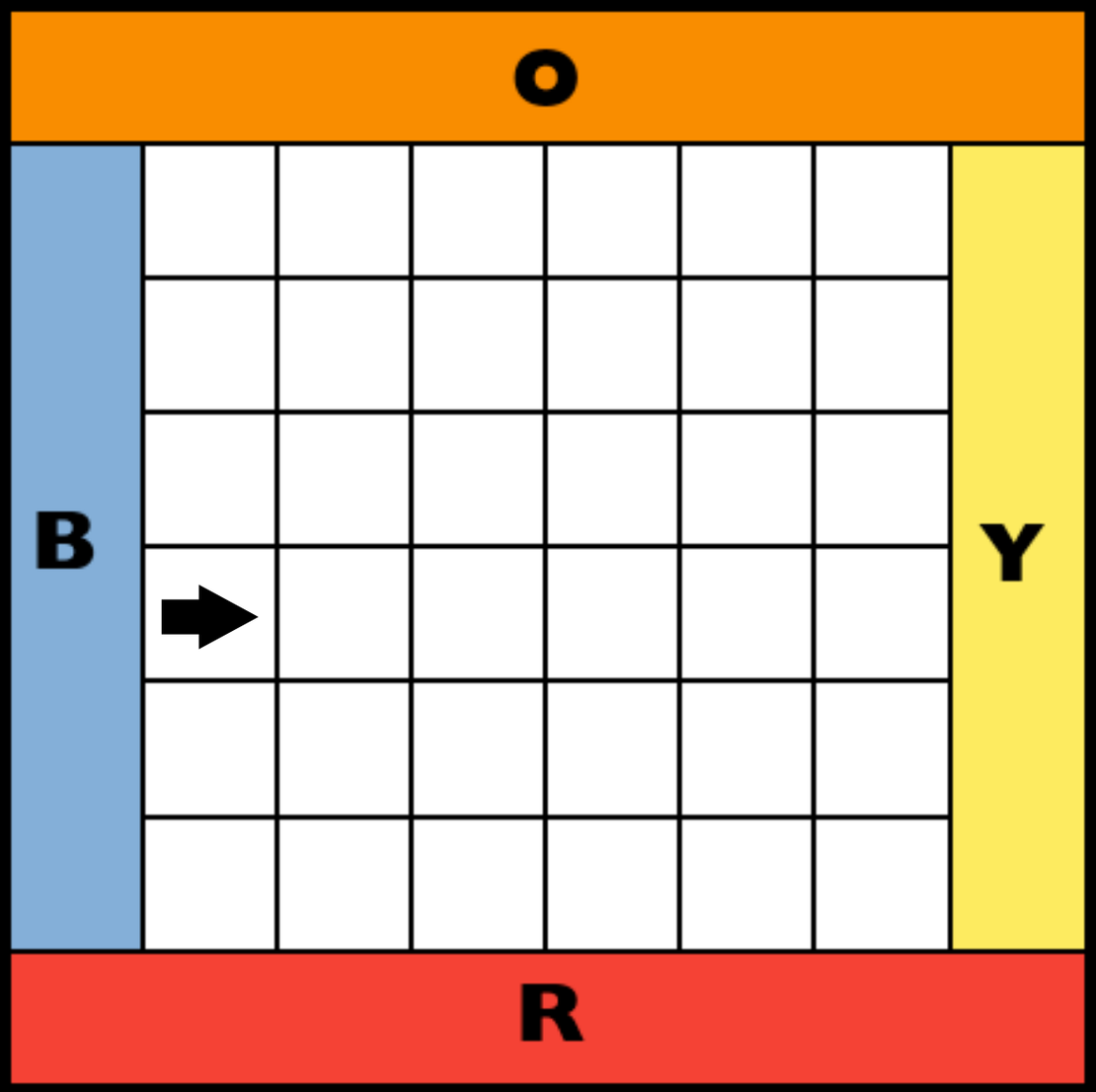
History: Fb.



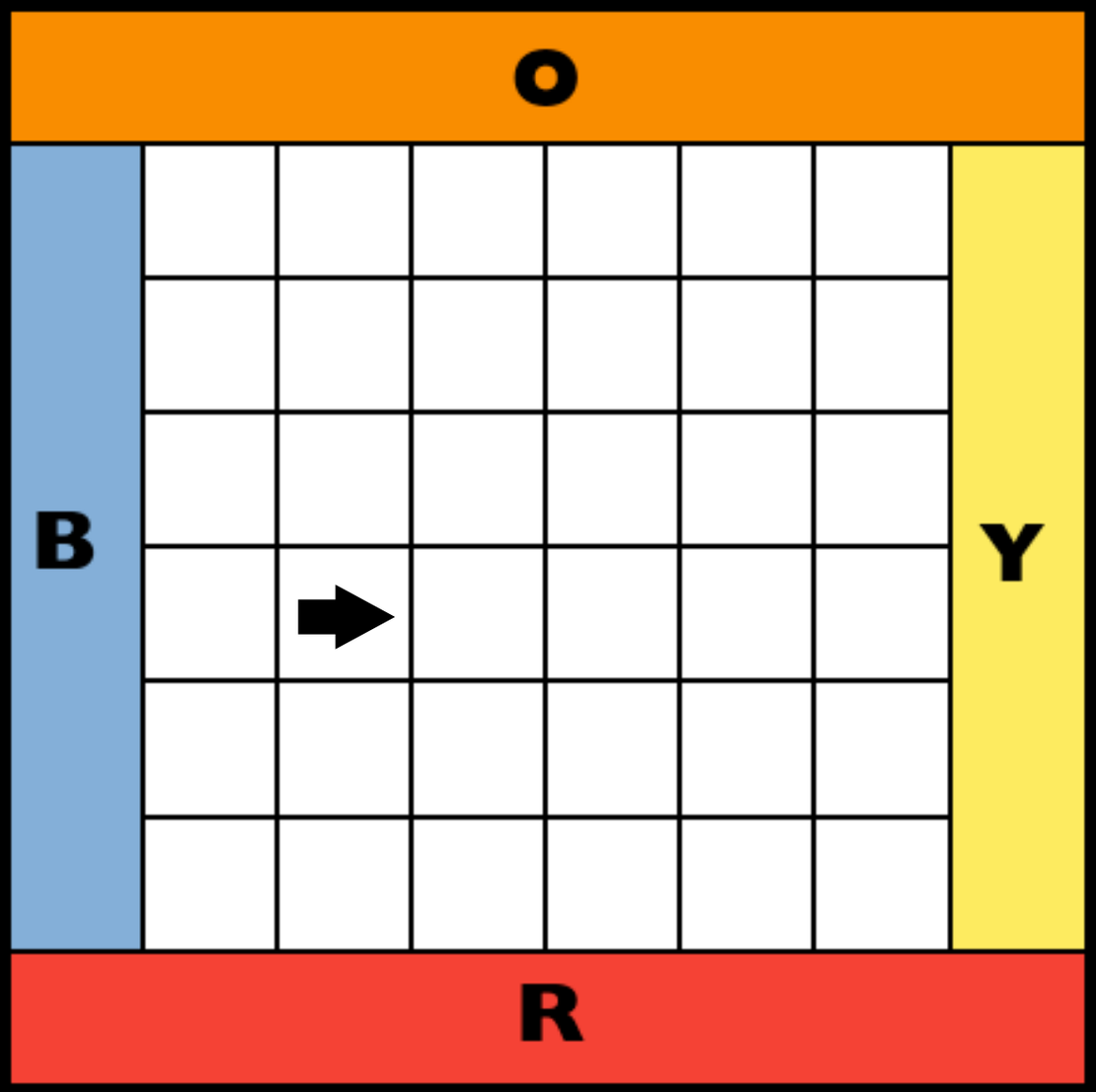
History: Fb. Lw.



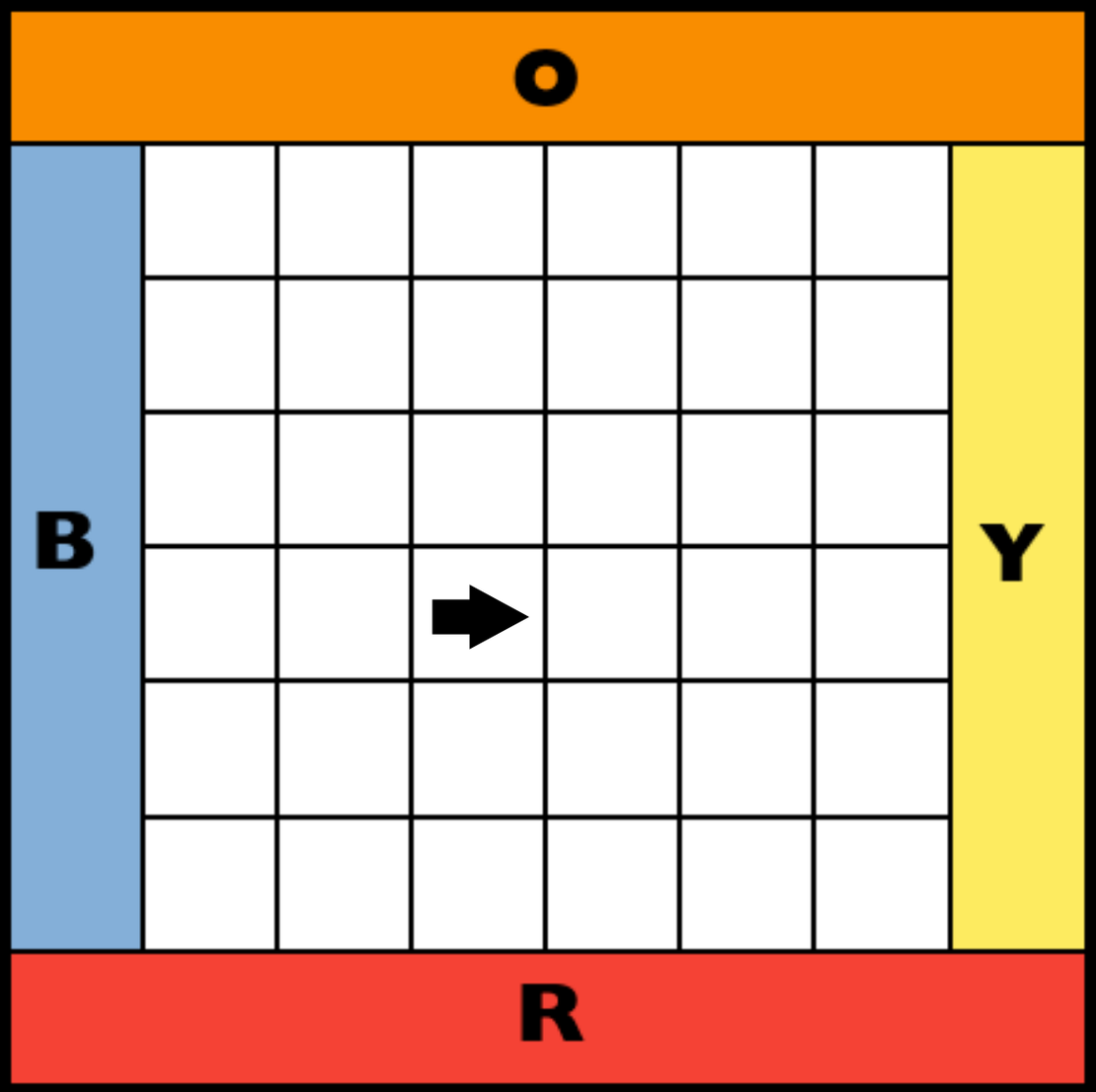
History: Fb. Lw. Fw.



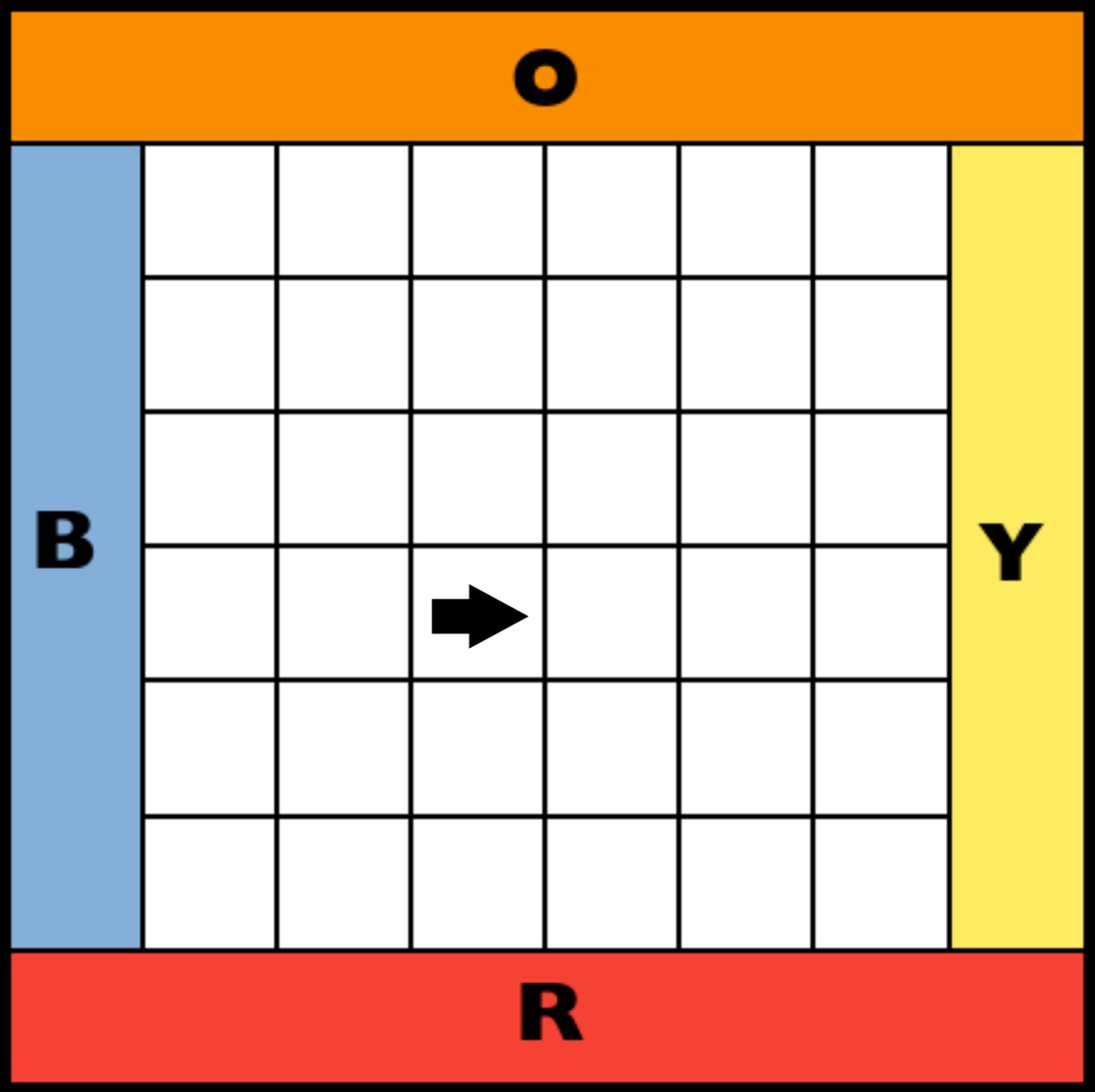
History: Fb. Lw. Fw. Lw.



History: Fb. Lw. Fw. Lw. Fw.

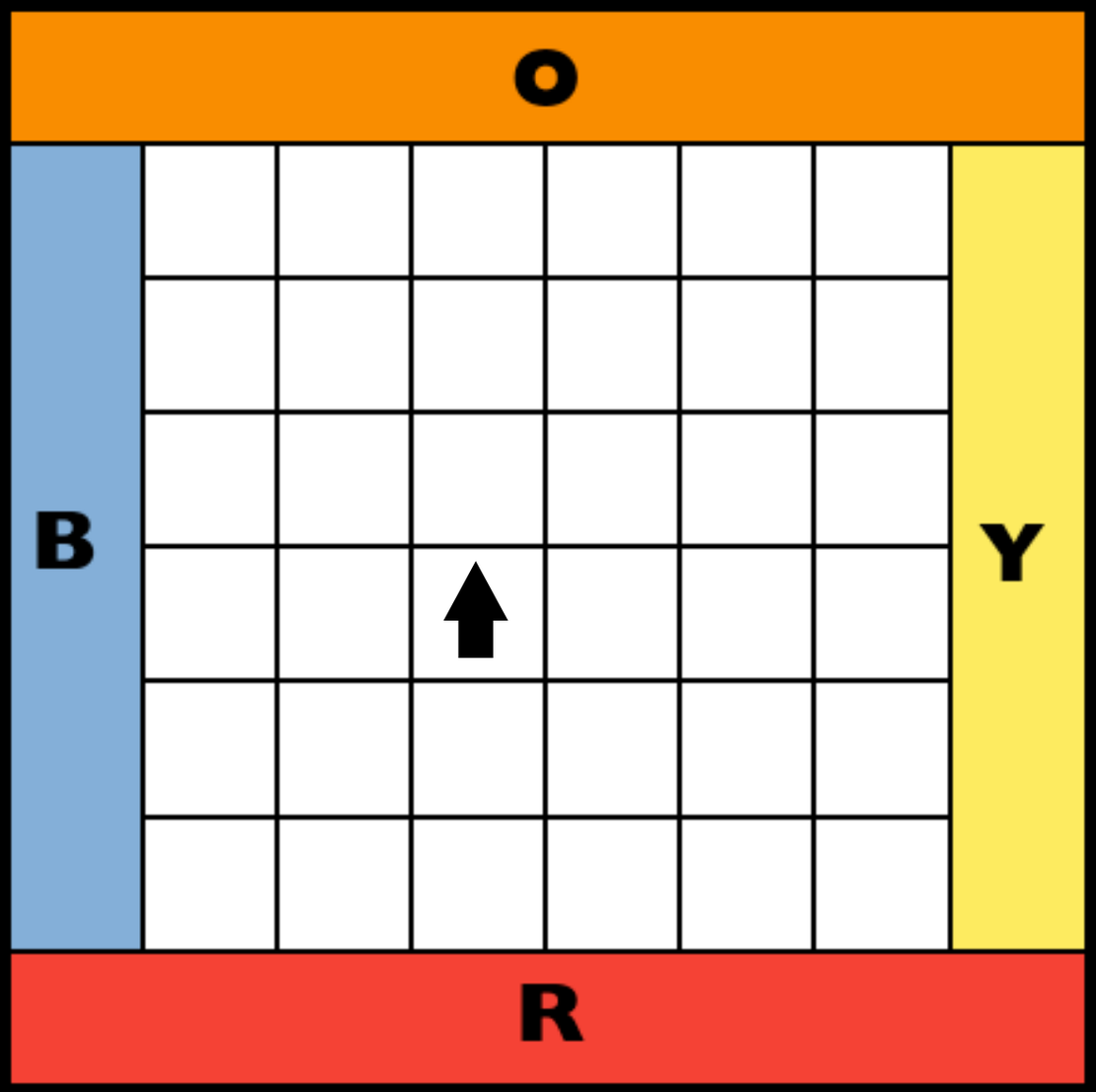


History: Fb. Lw. Fw. Lw. Fw. Fw.

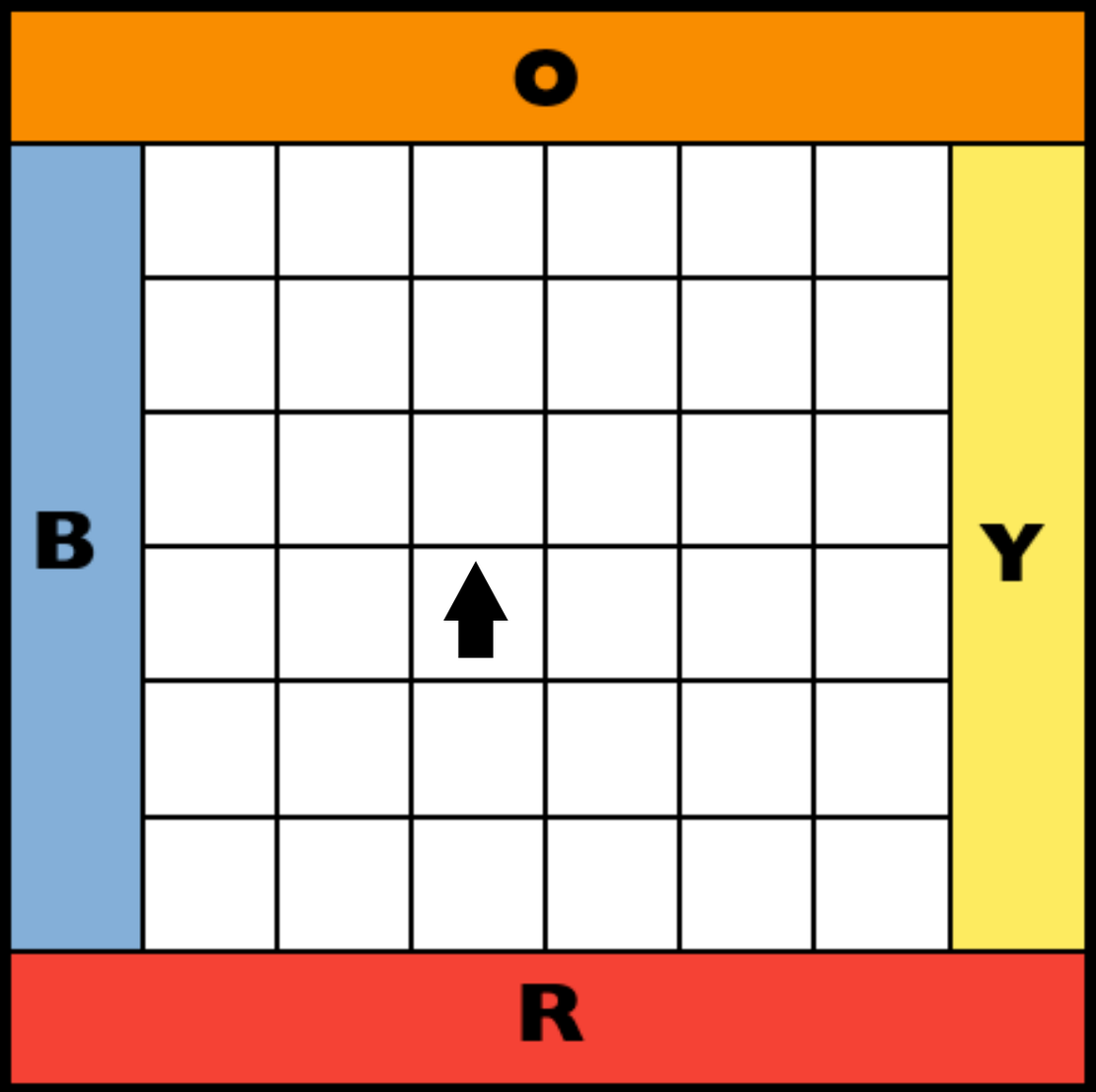


History: Fb. Lw. Fw. Lw. Fw. Fw. Lw.

5% slip

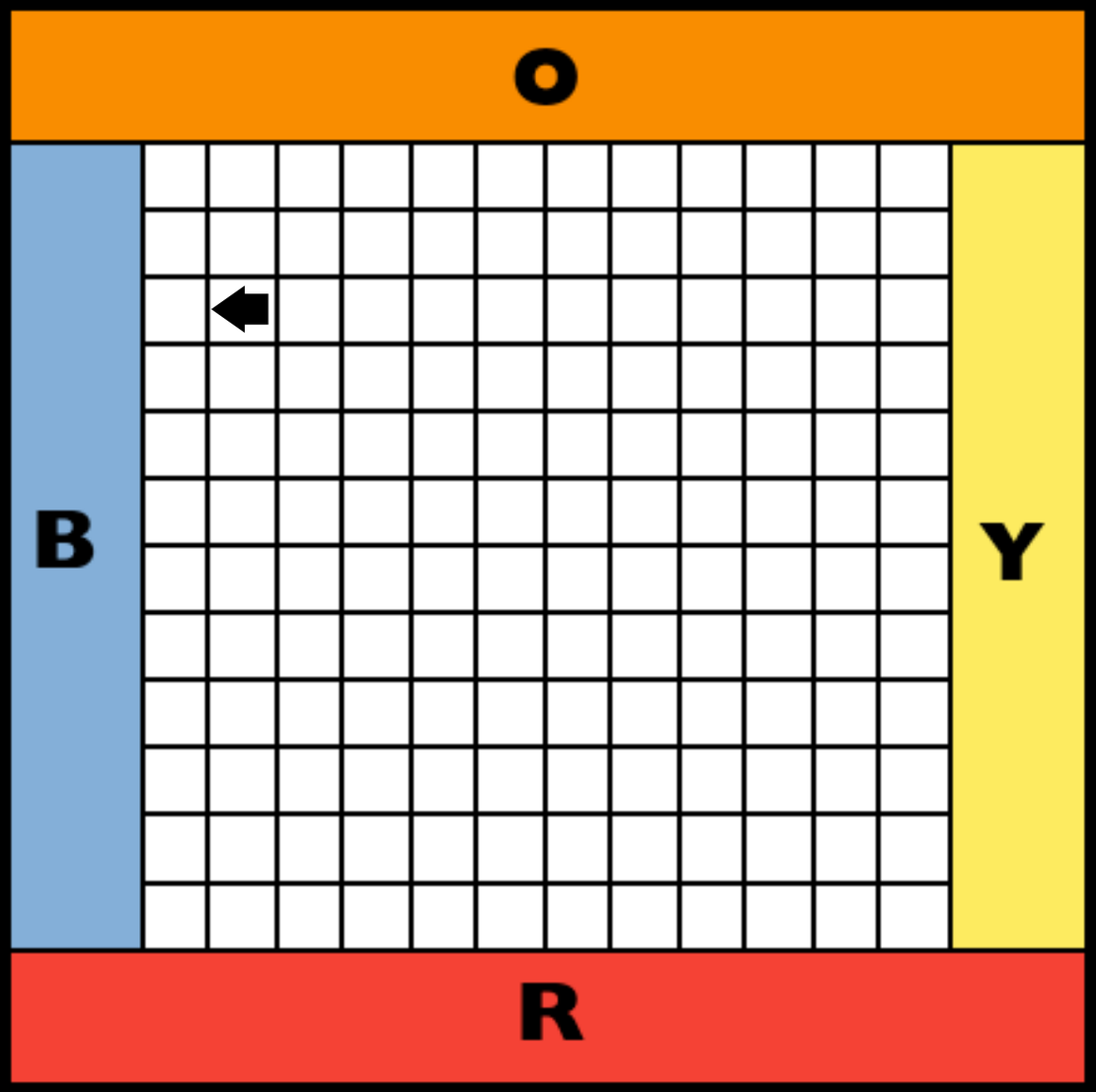


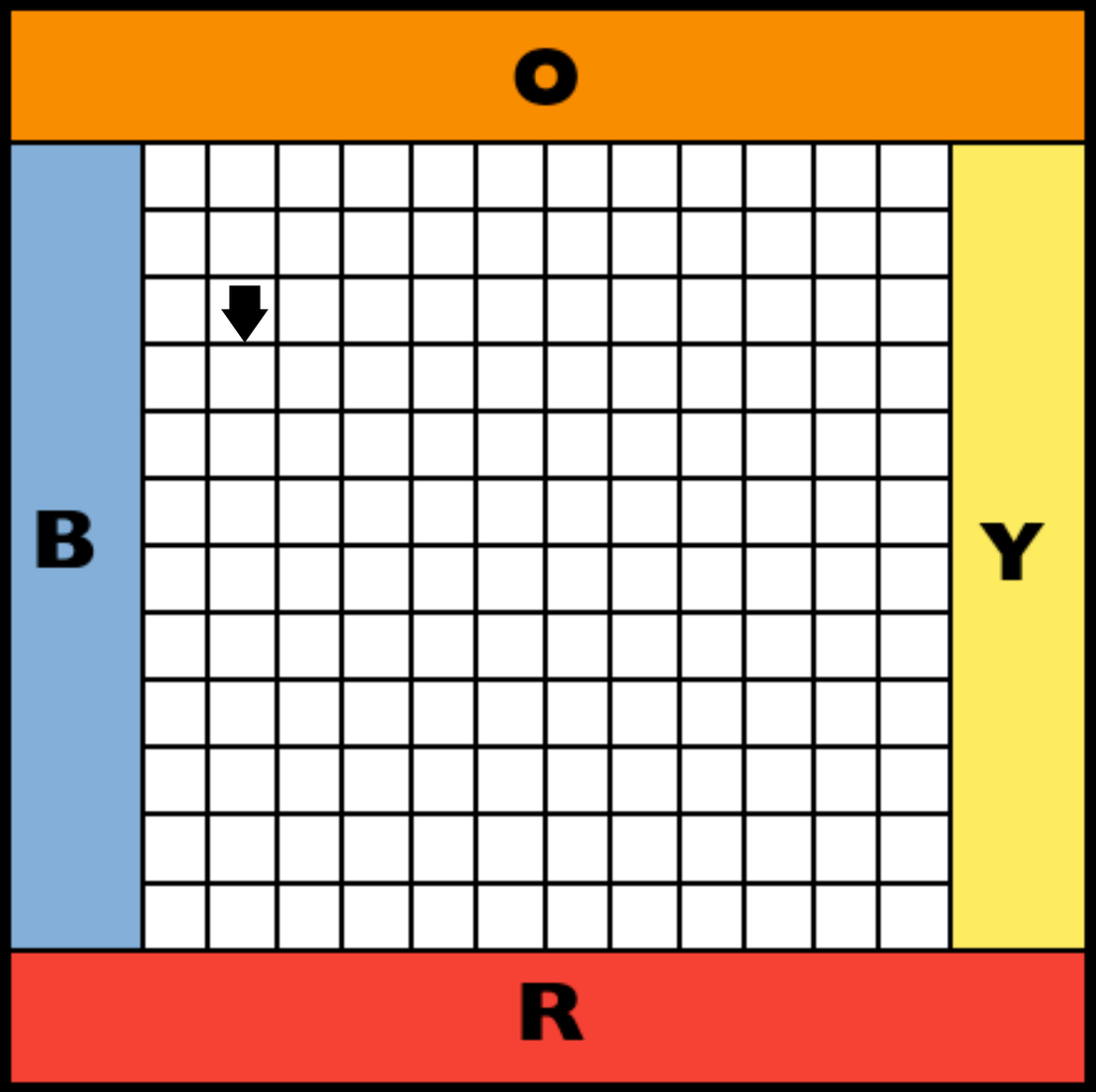
History: Fb. Lw. Fw. Lw. Fw. Fw. Lw. Lw.



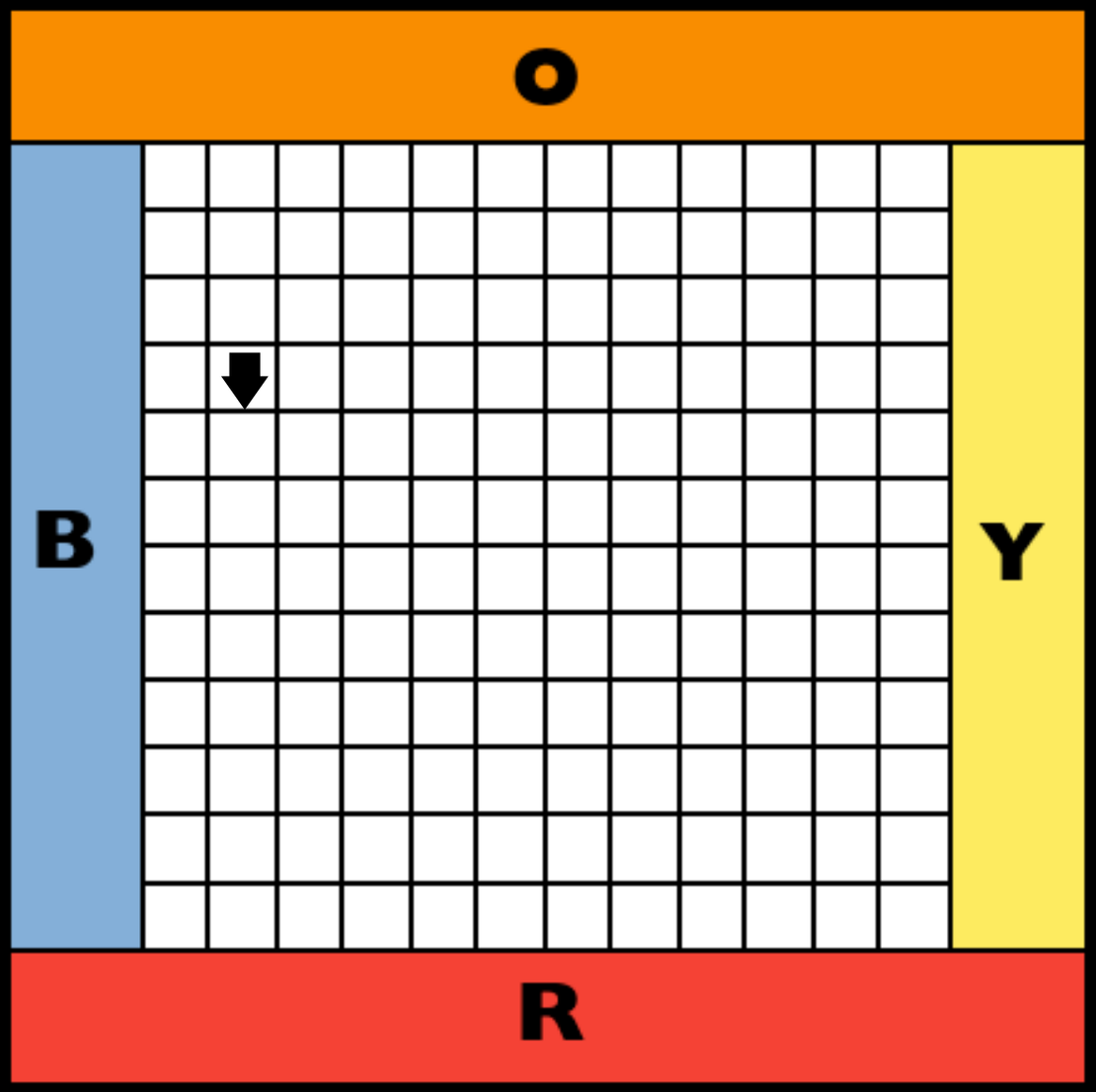
History: Fb. Lw. Fw. Lw. Fw. Fw. Lw. Lw.

Predict: Will I see Orange “in the near future”?

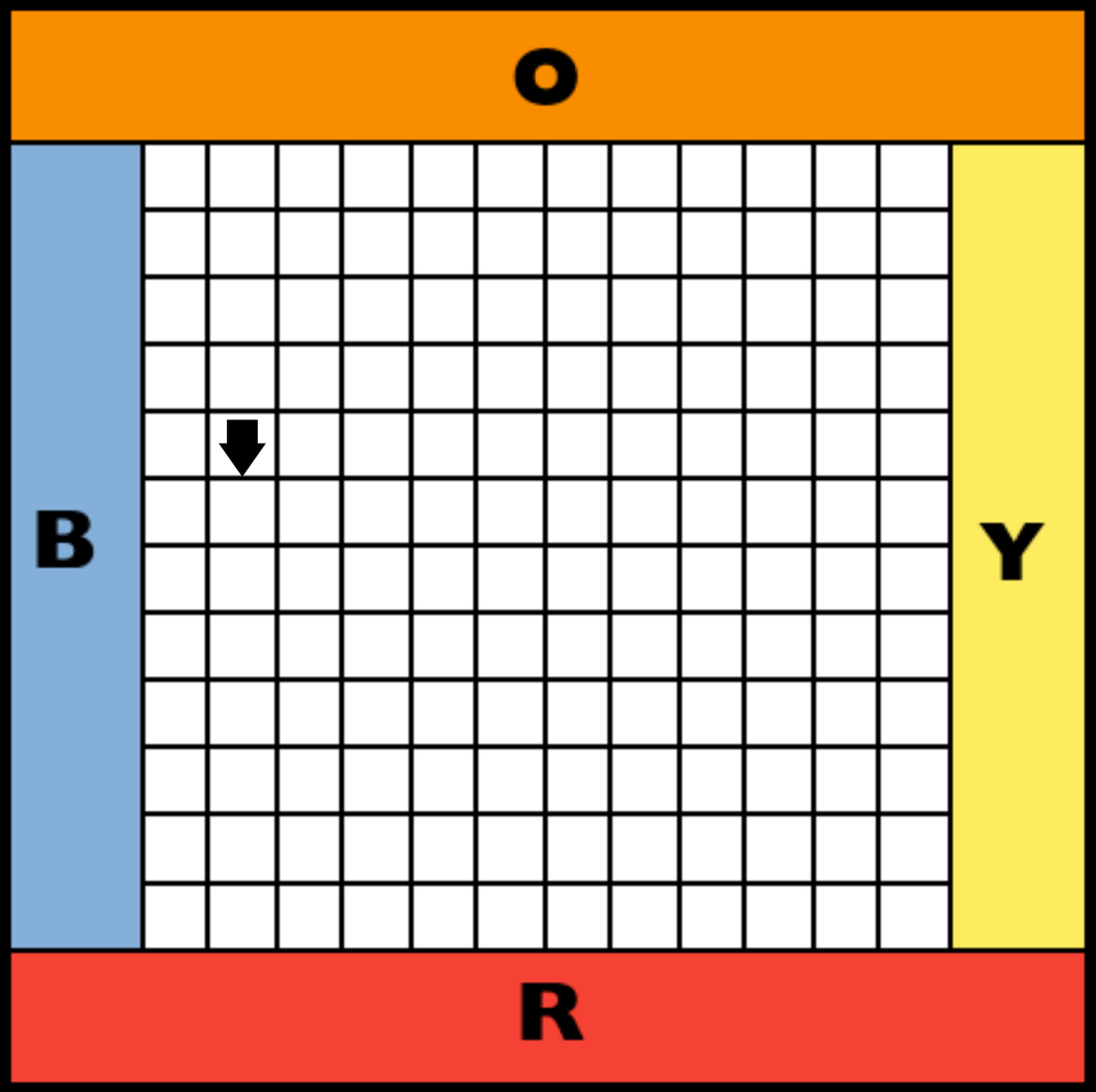

O
B
Y
R



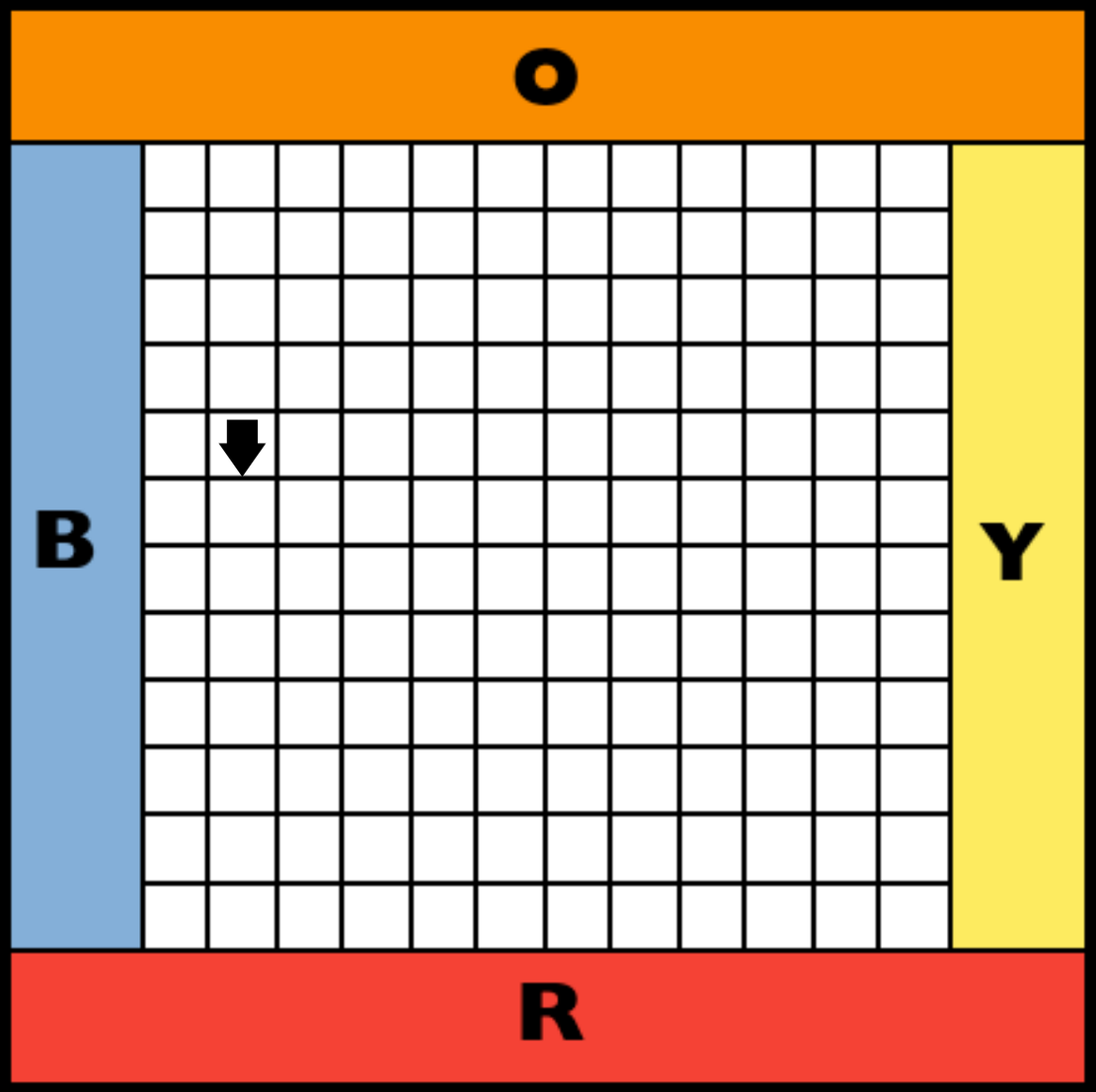
History: Lw.



History: Lw. Fw.

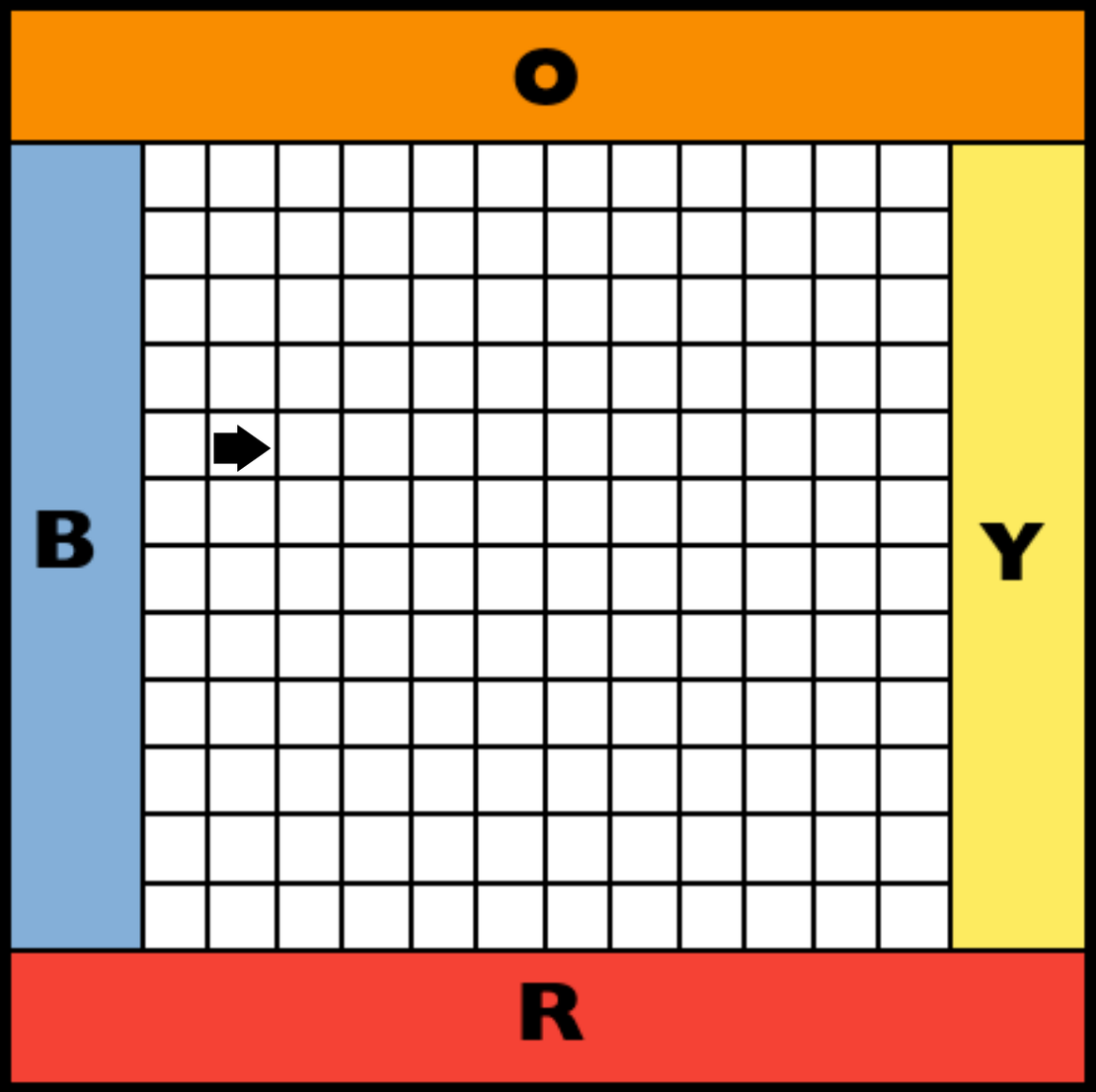


History: Lw. Fw. Fw.

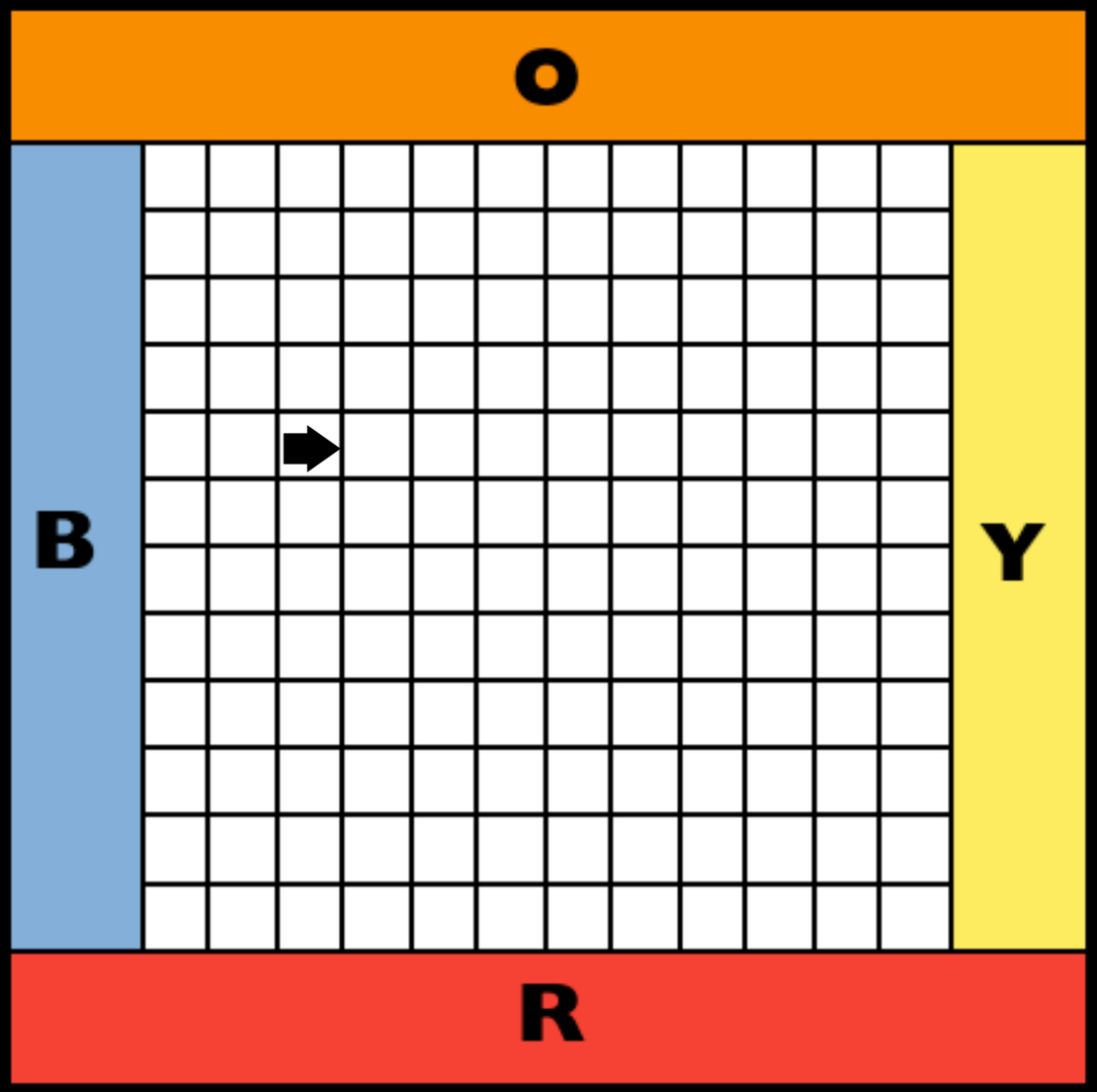


History: Lw. Fw. Fw. Fw.

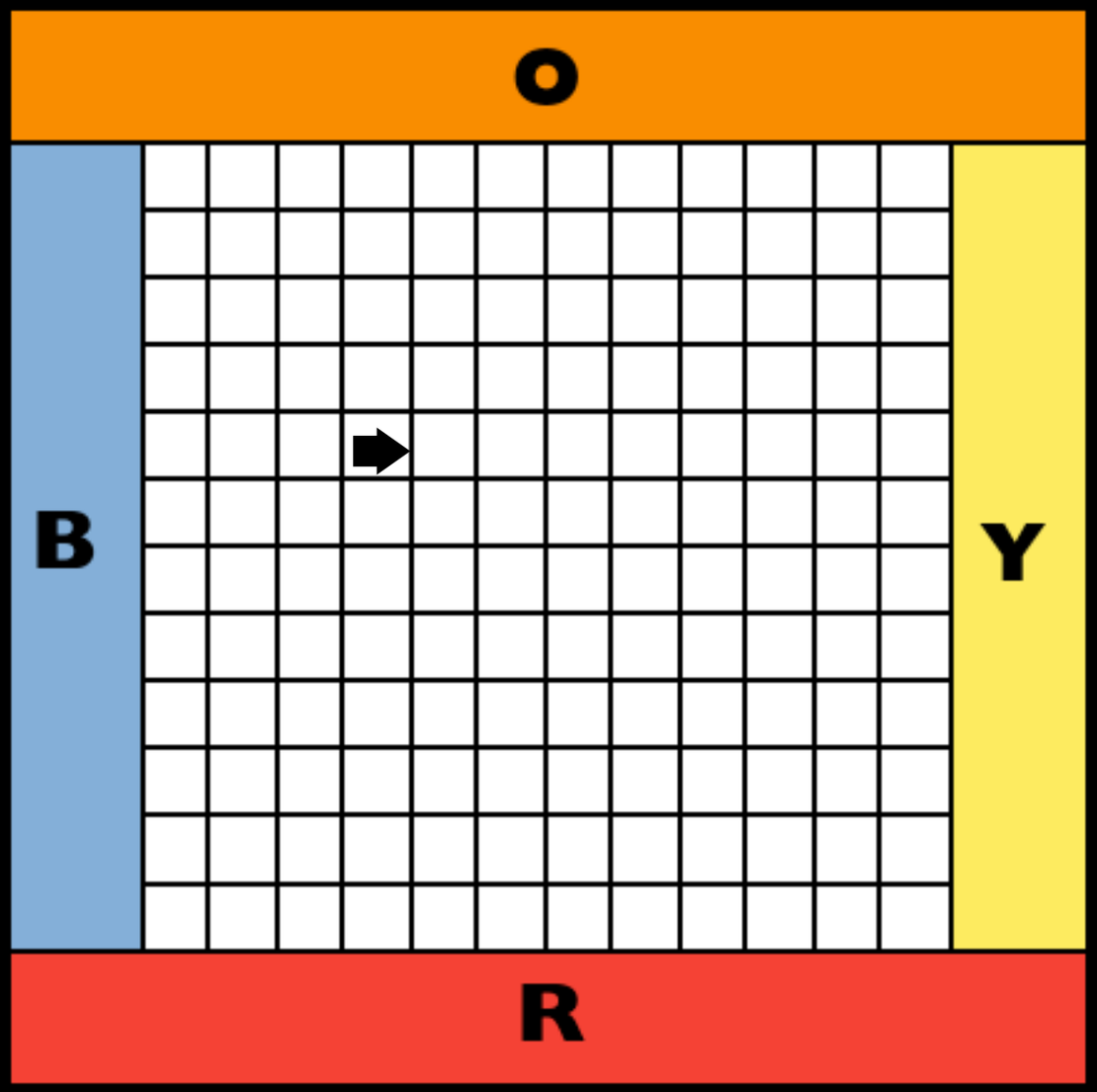
5% slip



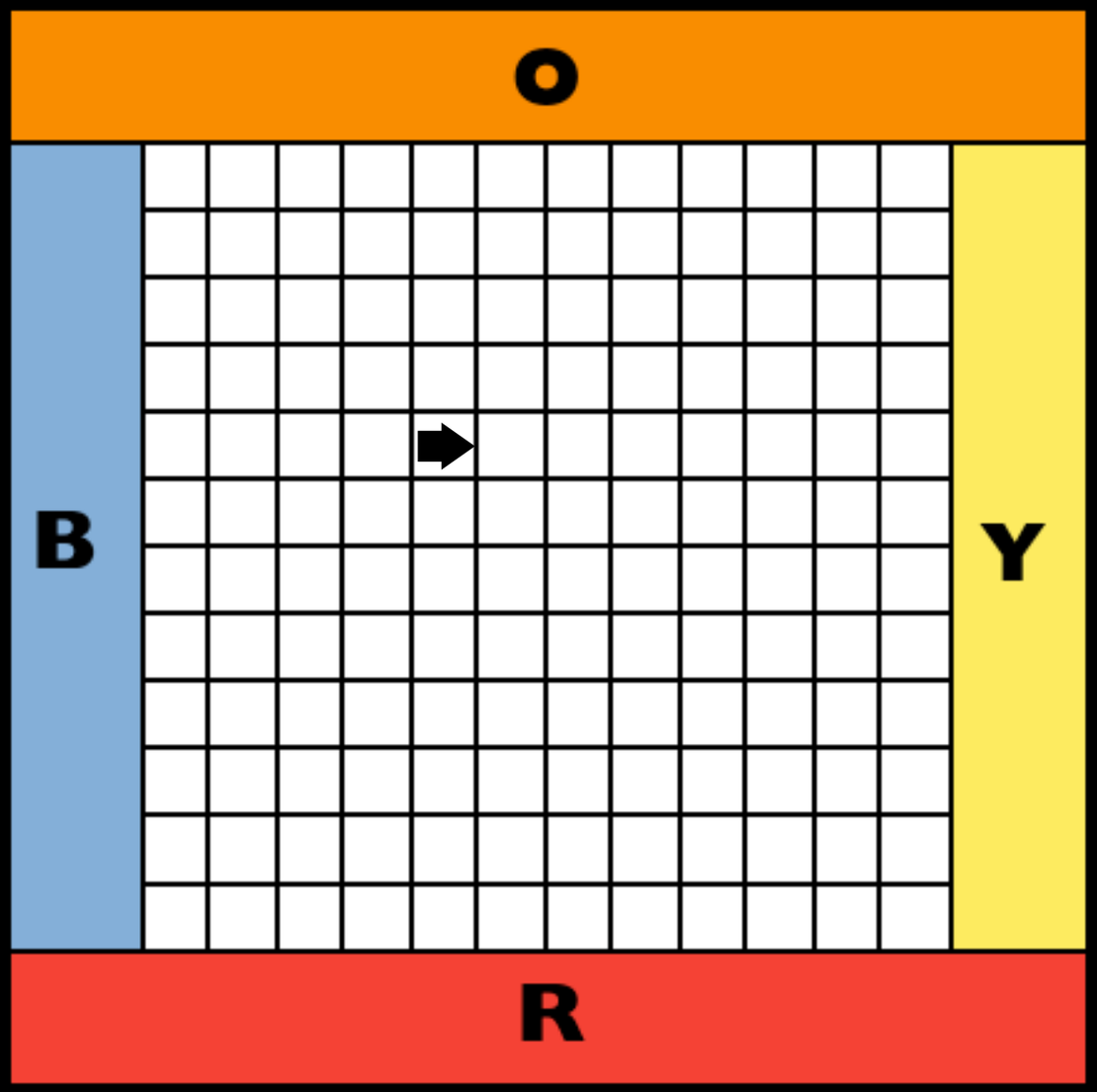
History: Lw. Fw. Fw. Fw. Lw.



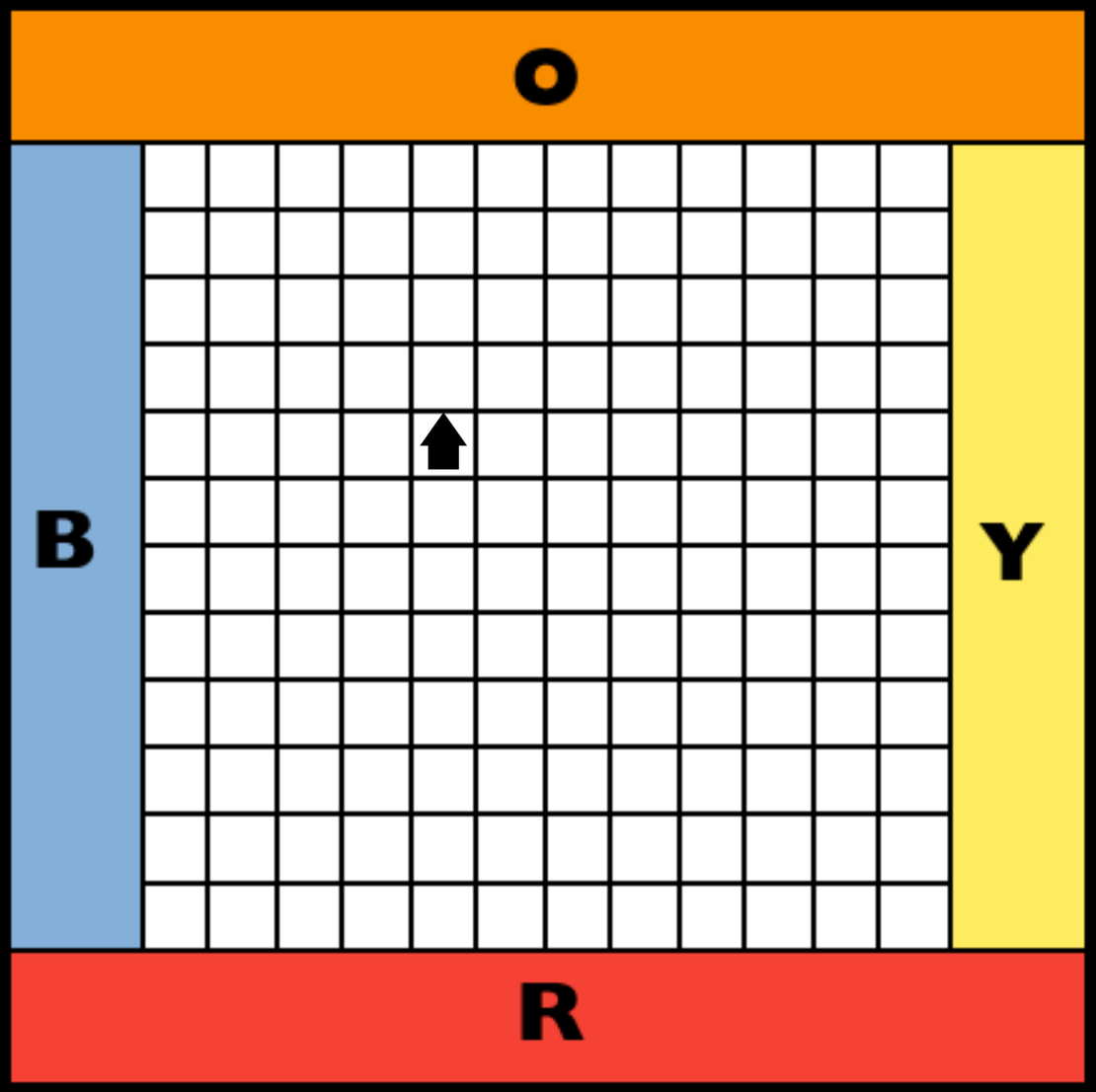
History: Lw. Fw. Fw. Fw. Lw. Fw.



History: Lw. Fw. Fw. Fw. Lw. Fw. Fw.

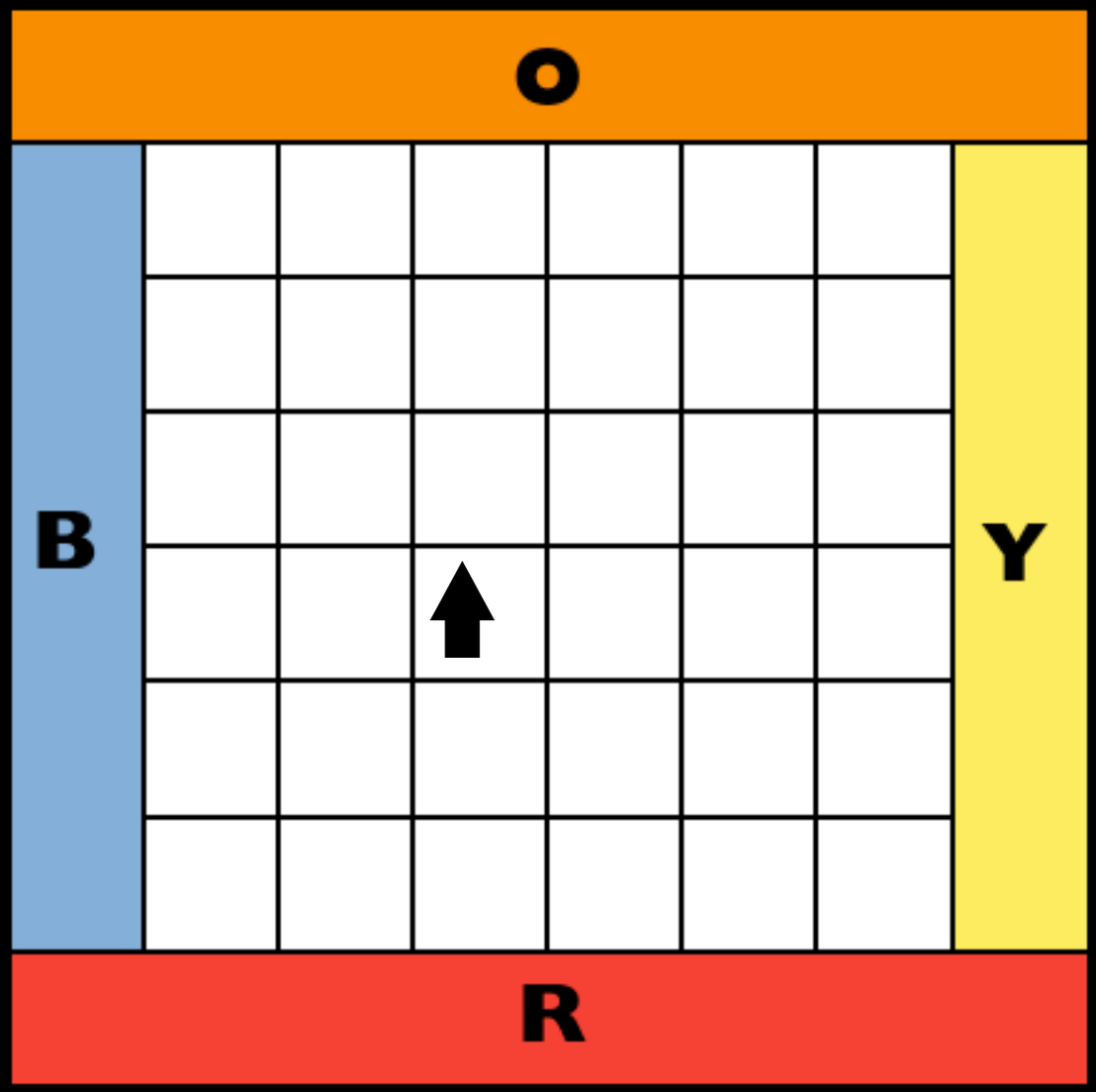


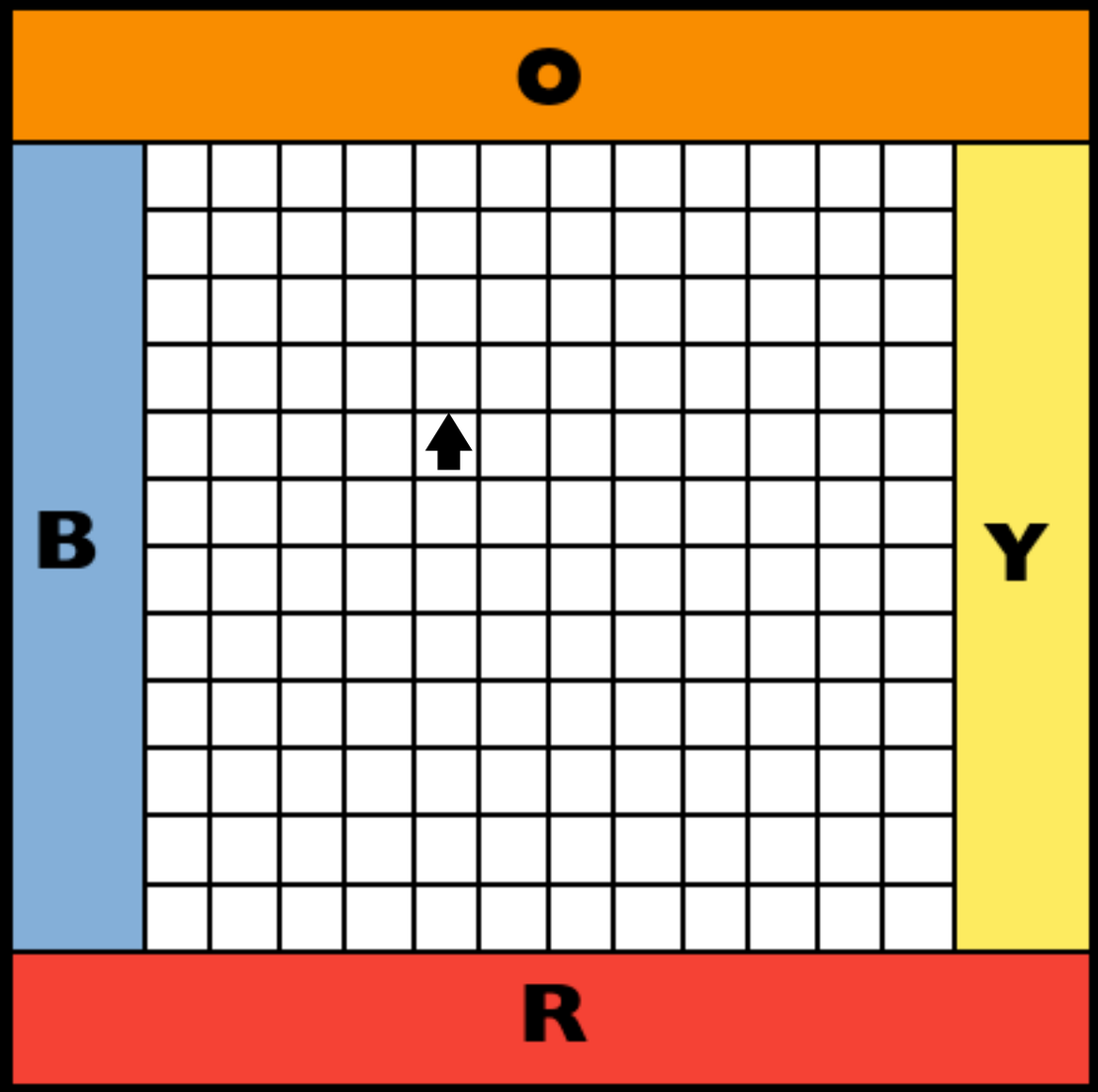
History: Lw. Fw. Fw. Fw. Lw. Fw. Fw. Fw.



History: Lw. Fw. Fw. Fw. Lw. Fw. Fw. Fw.

What is the probability now?





How do I predict if I will see
Orange "in the near future"?

History based methods

- In order to predict any test, you need to be able to predict all tests
- Size of table increases with length of histories/tests
- Need to summarize data!

Summarizing the data

- Approximate behaviour by limiting the tests of interest

Summarizing the data

- Approximate behaviour by limiting the tests of interest
- Filter tests : collapse the space of predictions the agent is interested in

Summarizing the data

- Approximate behaviour by limiting the tests of interest
- Filter tests : collapse the space of predictions the agent is interested in
- Filter histories: extract predictive information from the time series

Probes

Probes

1. On tests: $f : O^* \to \mathbb{R}$

   - restriction: compositionality

     $f(\omega_1) = f(\omega_2) \Rightarrow f(\omega_1 o) = f(\omega_2 o)$

Probes

1. On tests: $f : O^* \to \mathbb{R}$

   - restriction: compositionality

     $f(\omega_1) = f(\omega_2) \Rightarrow f(\omega_1 o) = f(\omega_2 o)$

2. On histories: $g : (\Sigma \times O)^* \to \mathbb{R}$

History probes

- Should maintain enough information to predict well the outcomes of test probes

History probes

- Should maintain enough information to predict well the outcomes of test probes
- Good heuristic : eligibility traces
- Temporarily assign credit for the occurrence of an event (i.e. a colour)

What we want

- A machine that holds a summary of all the histories and tests in the world

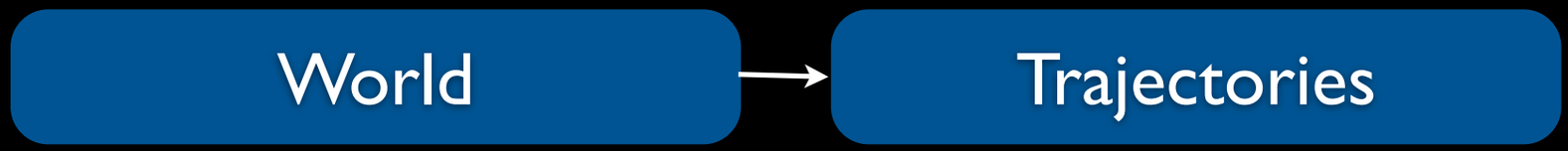
What we want

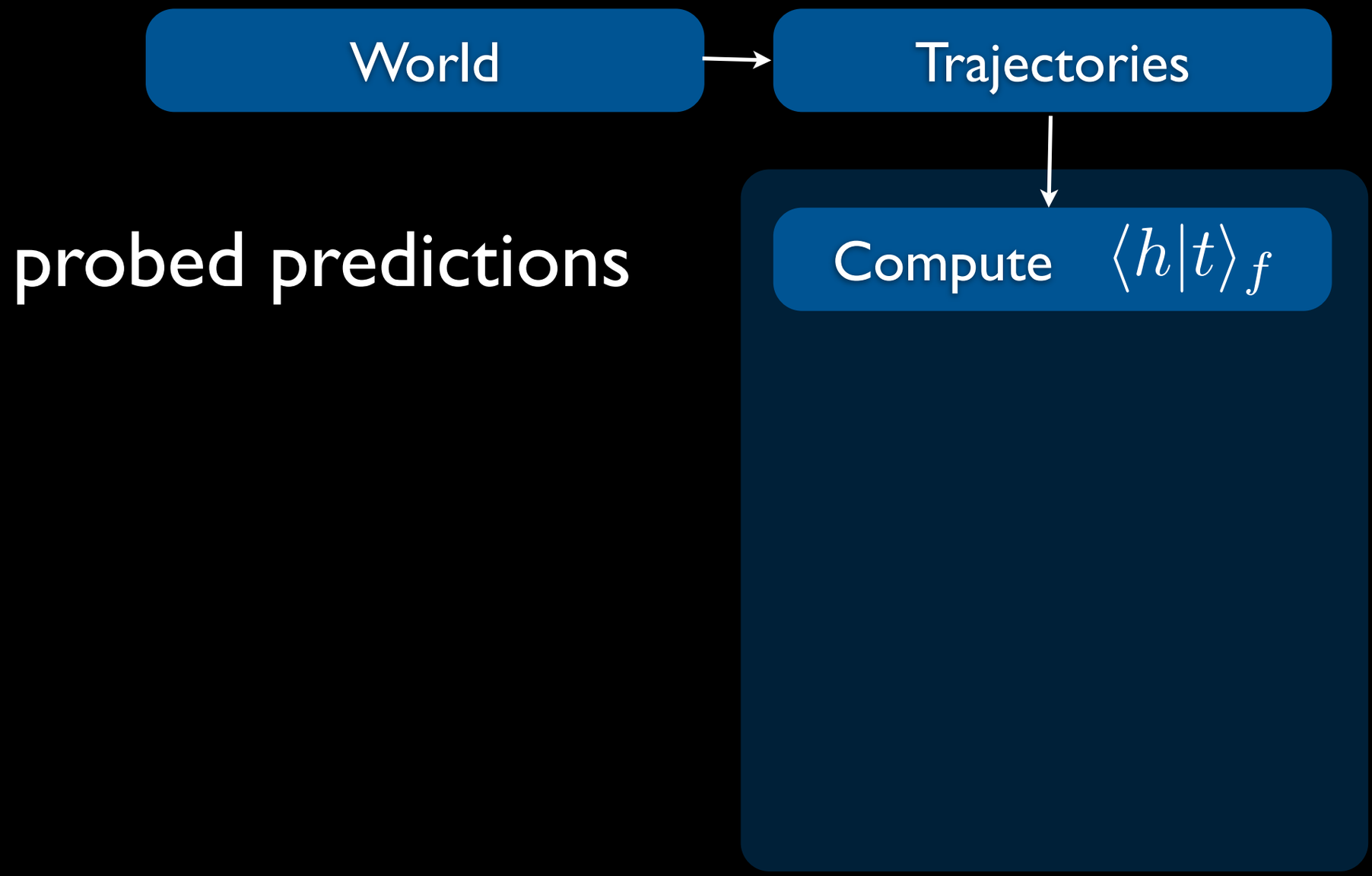
- A machine that holds a summary of all the histories and tests in the world
  - States = summaries of histories
  - Observations = predictions of probes

What we want

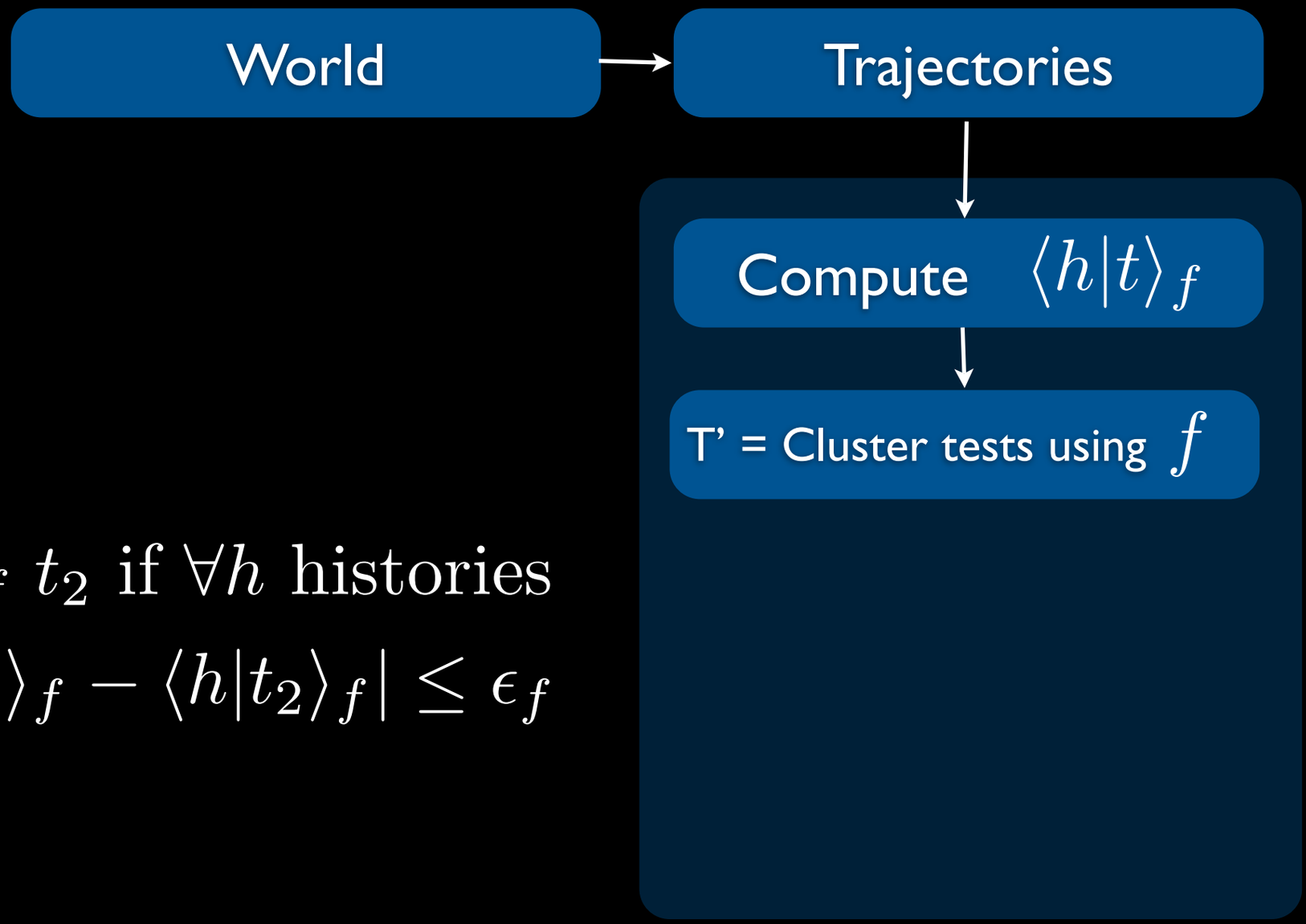
- A machine that holds a summary of all the histories and tests in the world
  - States = summaries of histories
  - Observations = predictions of probes
- Only predicts tests of interest

(Current) Algorithm


World
Trajectories

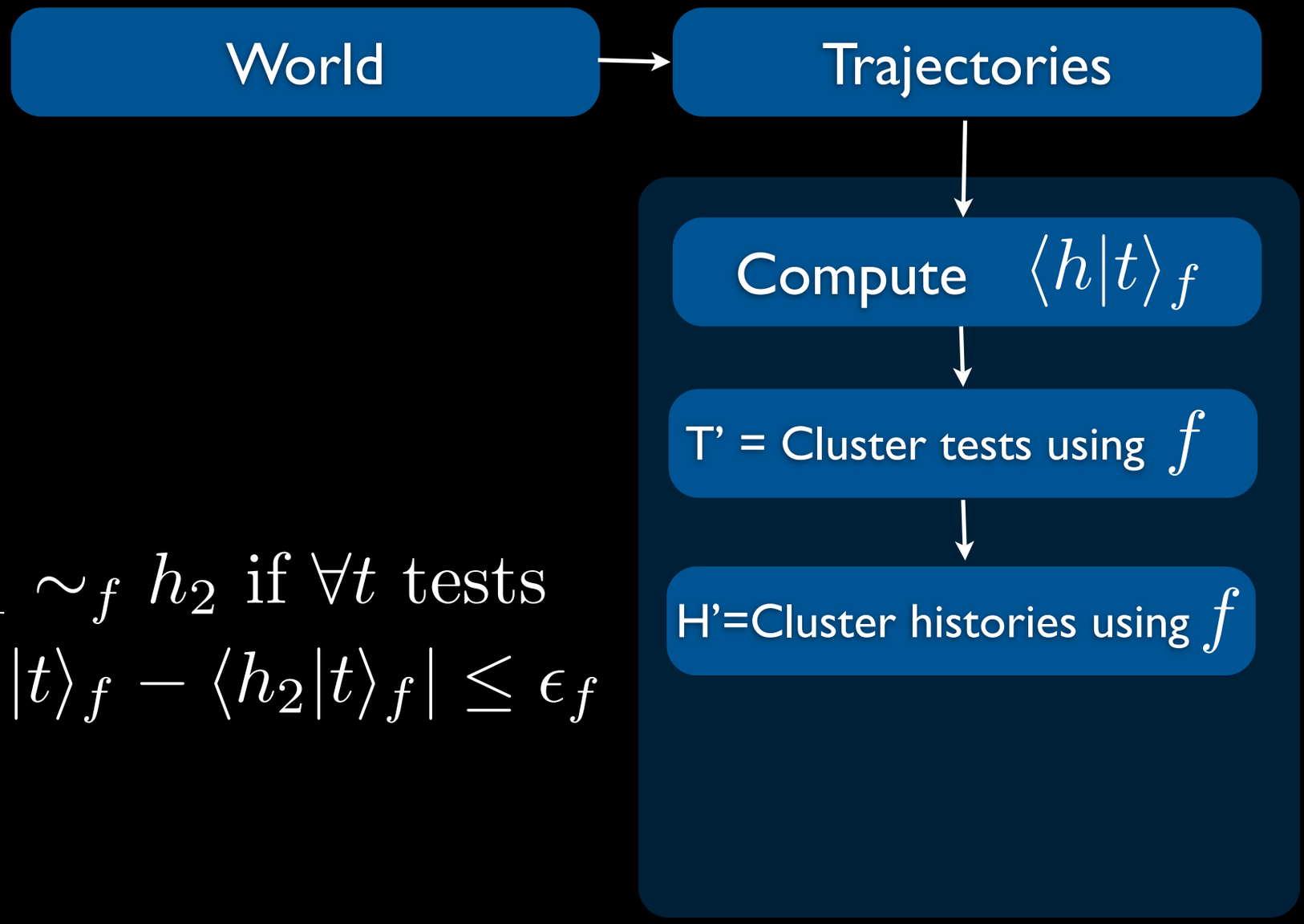


probed predictions



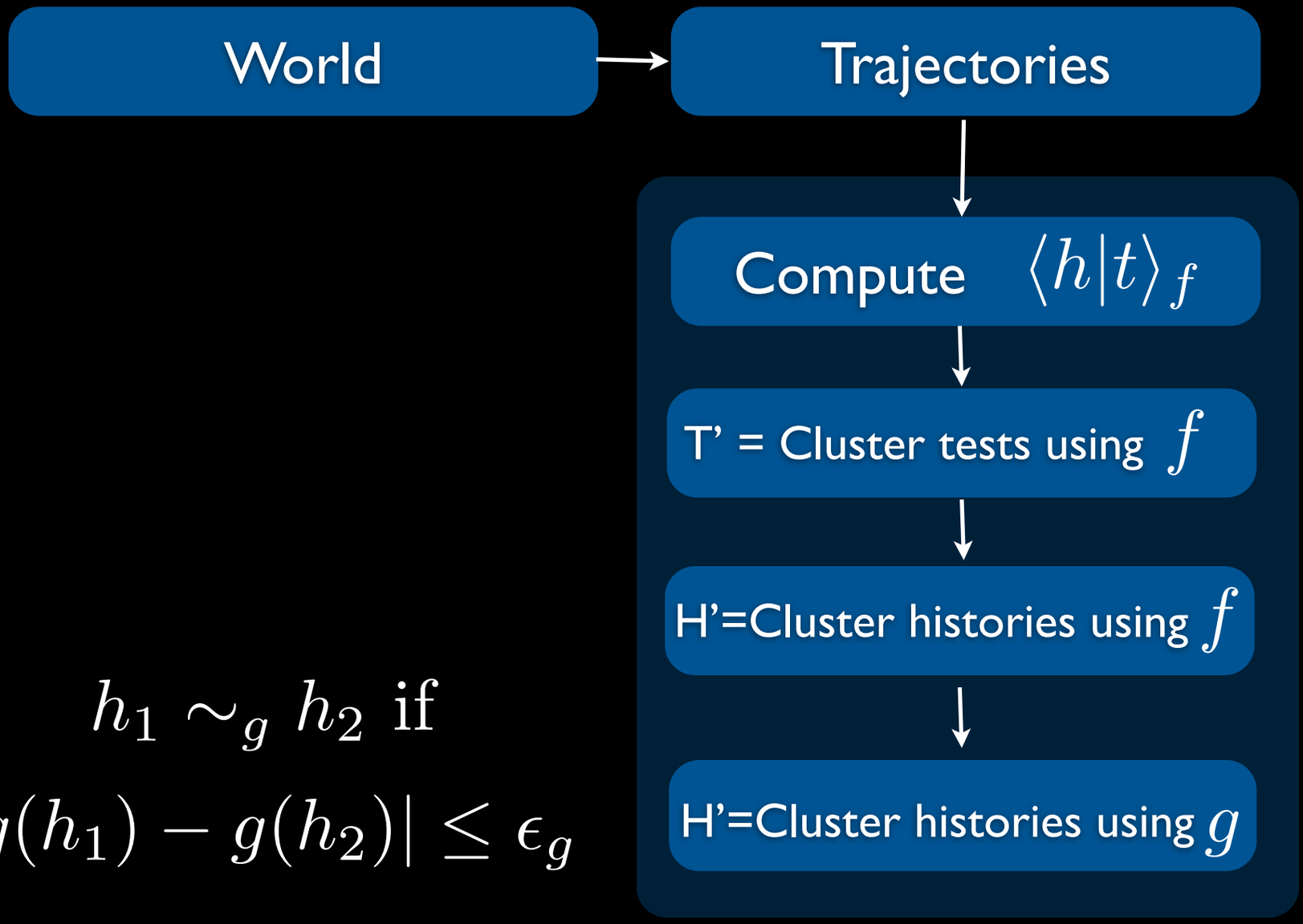
$t_1 \sim_f t_2$ if $\forall h$ histories

$|\langle h|t_1\rangle_f - \langle h|t_2\rangle_f| \leq \epsilon_f$



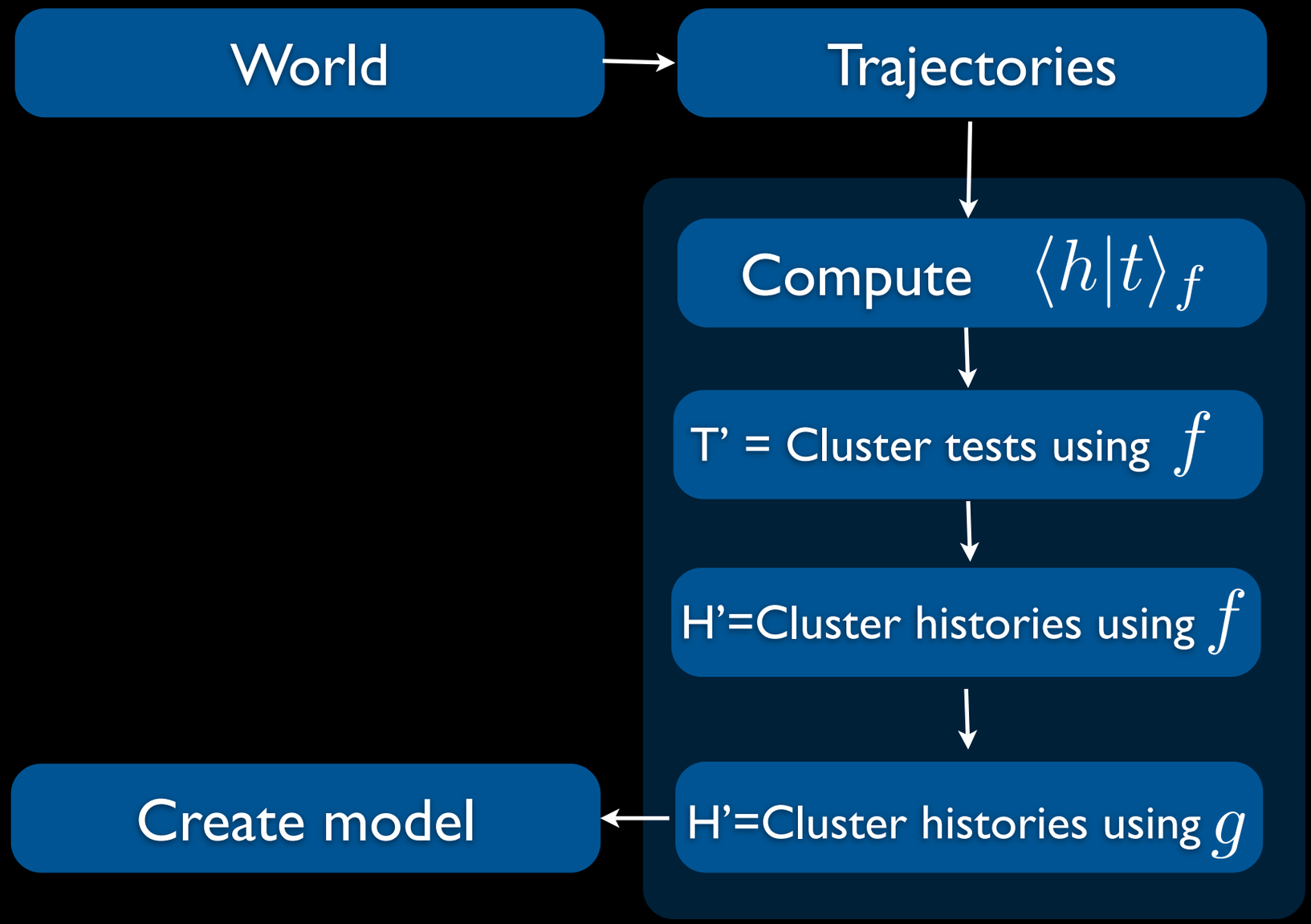
$h_1 \sim_f h_2$ if $\forall t$ tests

$|\langle h_1|t\rangle_f - \langle h_2|t\rangle_f| \leq \epsilon_f$



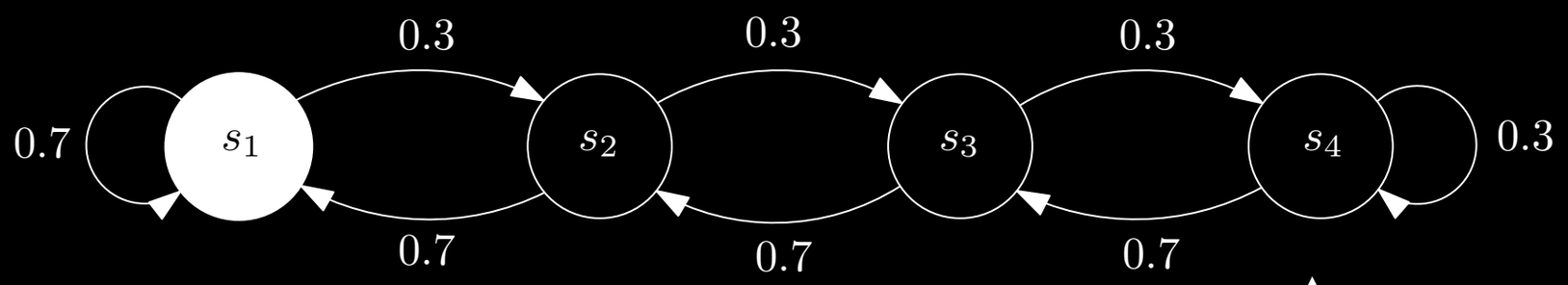
$$h_1 \sim_g h_2 \text{ if}$$

$$|g(h_1) - g(h_2)| \leq \epsilon_g$$


World
Trajectories
Compute $\langle h|t\rangle_f$
T' = Cluster tests using f
H'=Cluster histories using f
H'=Cluster histories using g
Create model

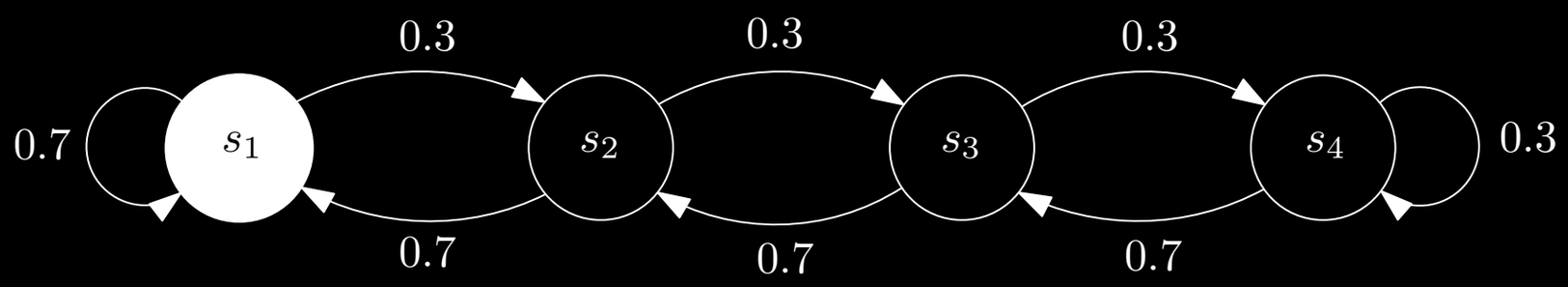
Results

Tunnel world



start state

Tunnel world



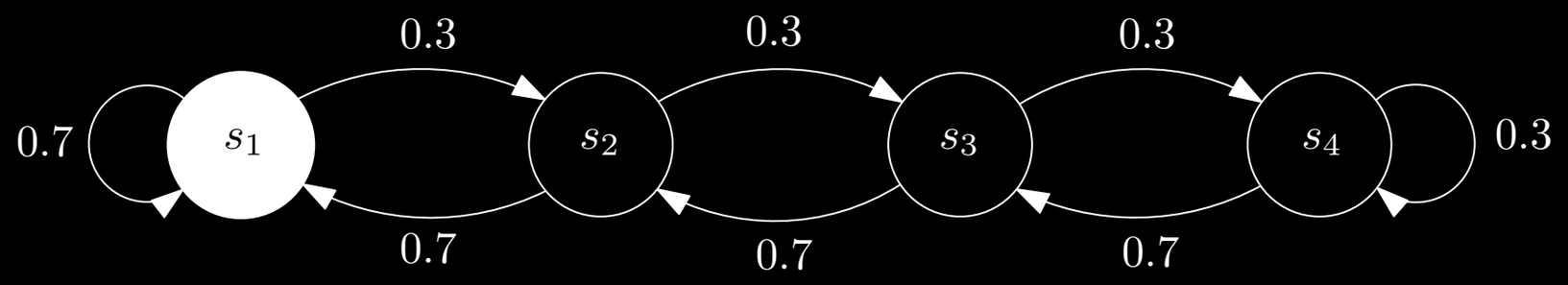
Algorithm parameters:

$\epsilon_f = 0.04$

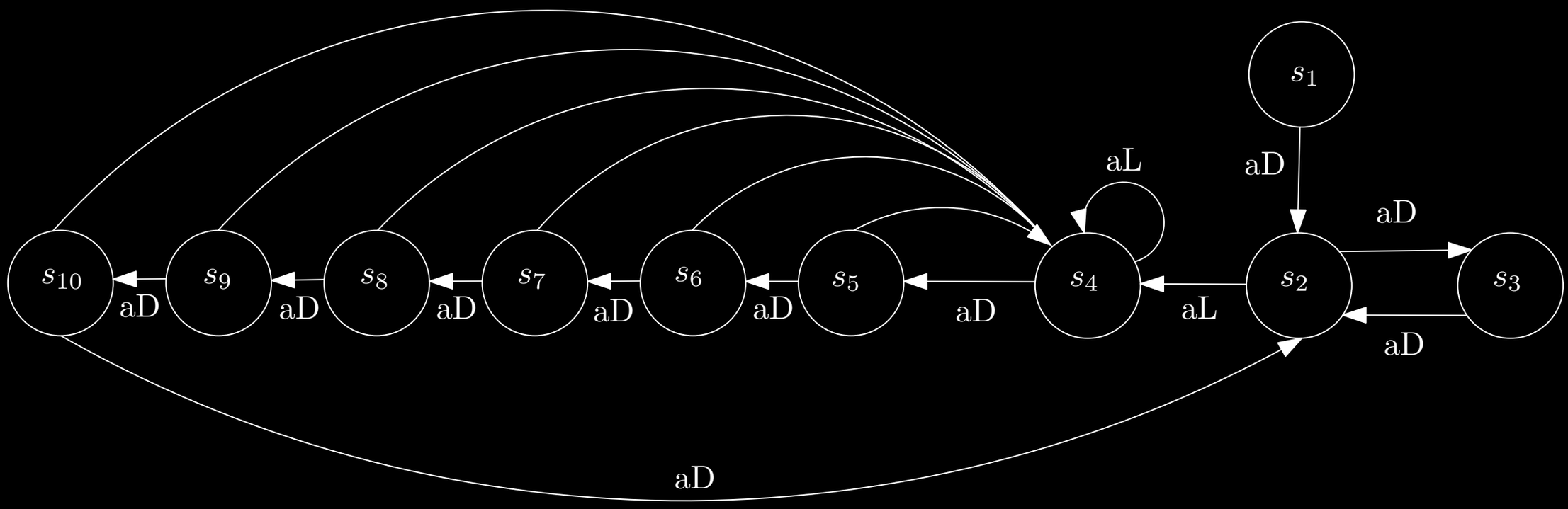
$\epsilon_g = 0.02$

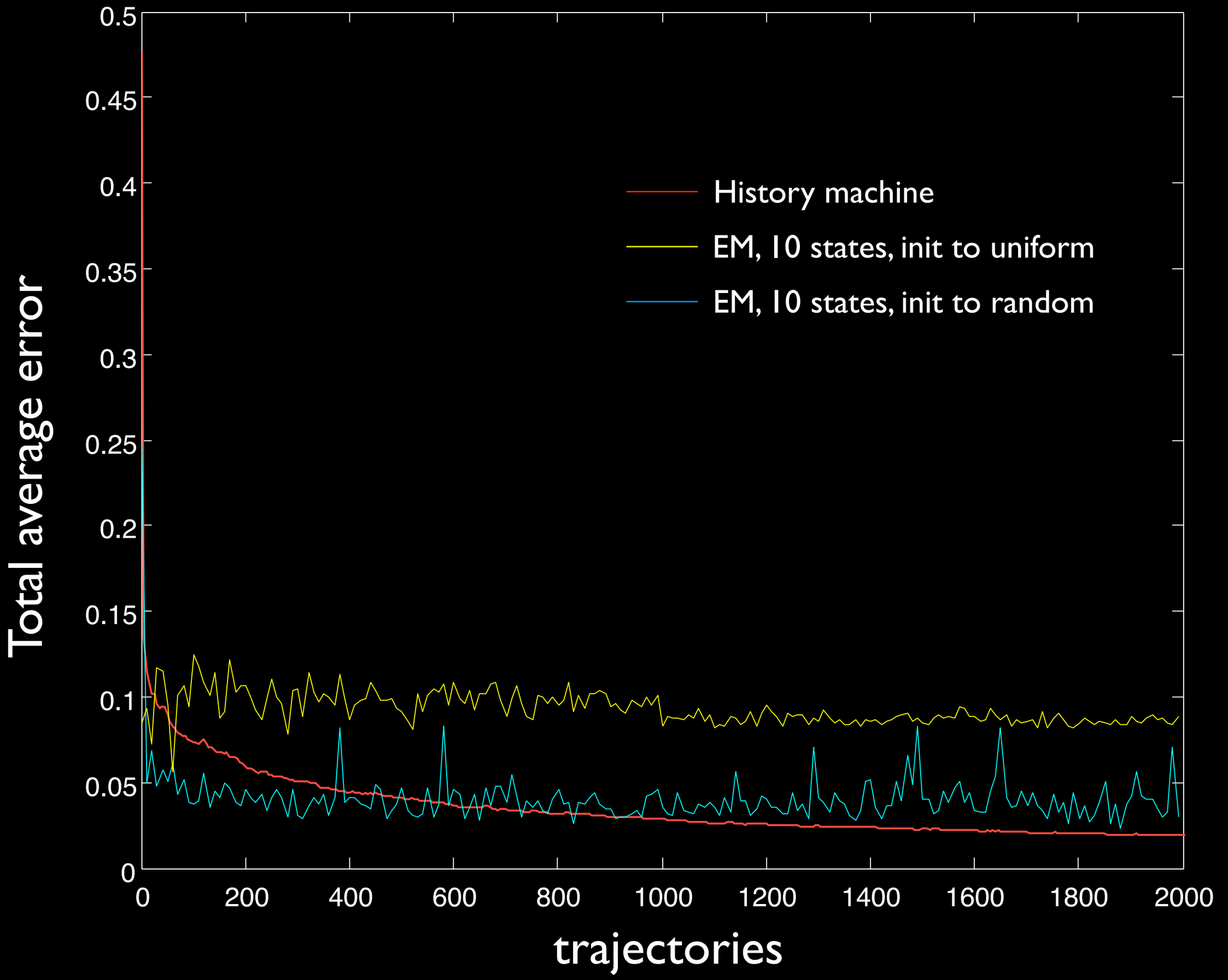
trajectories of length 14

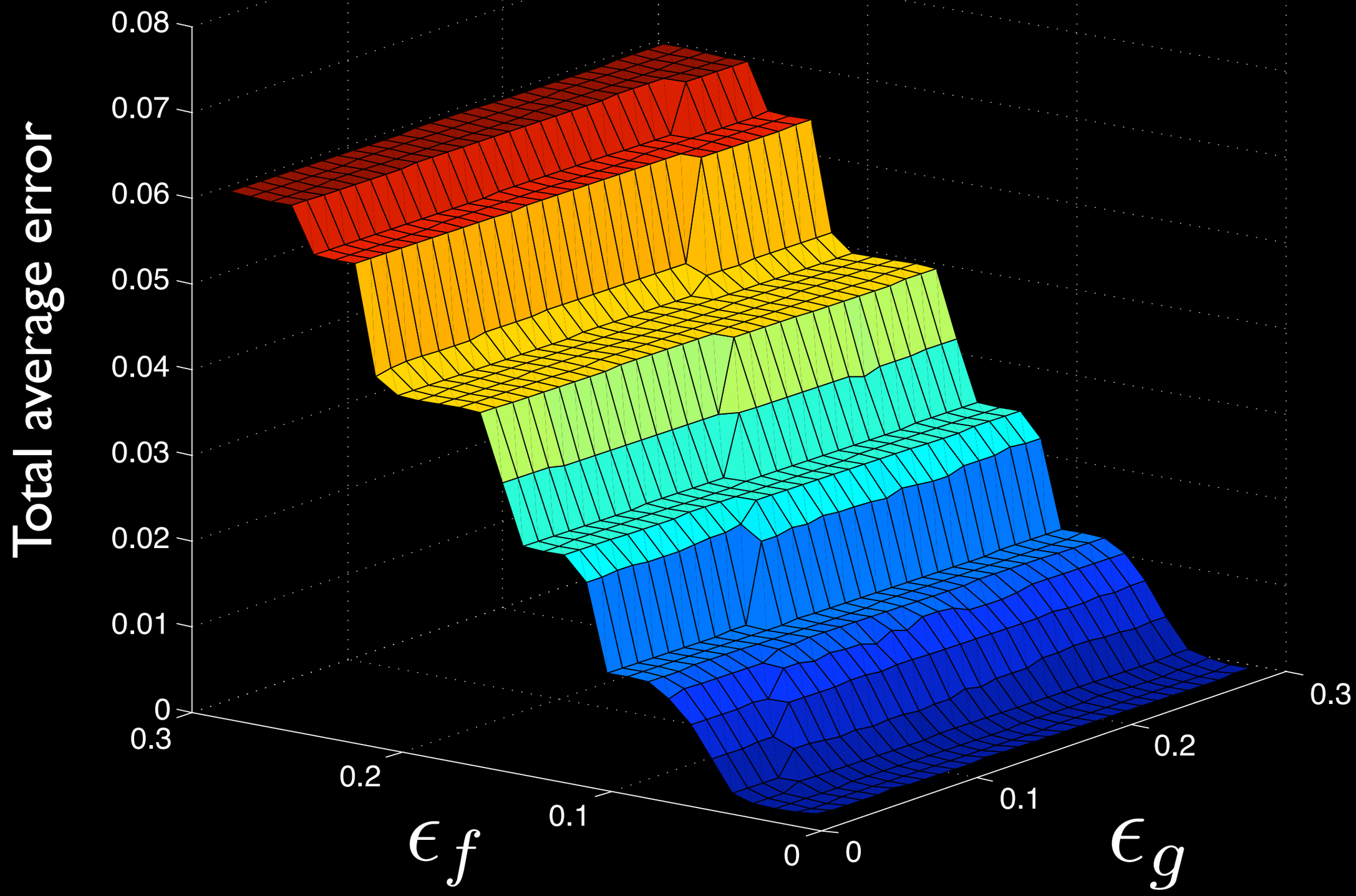
Tunnel world

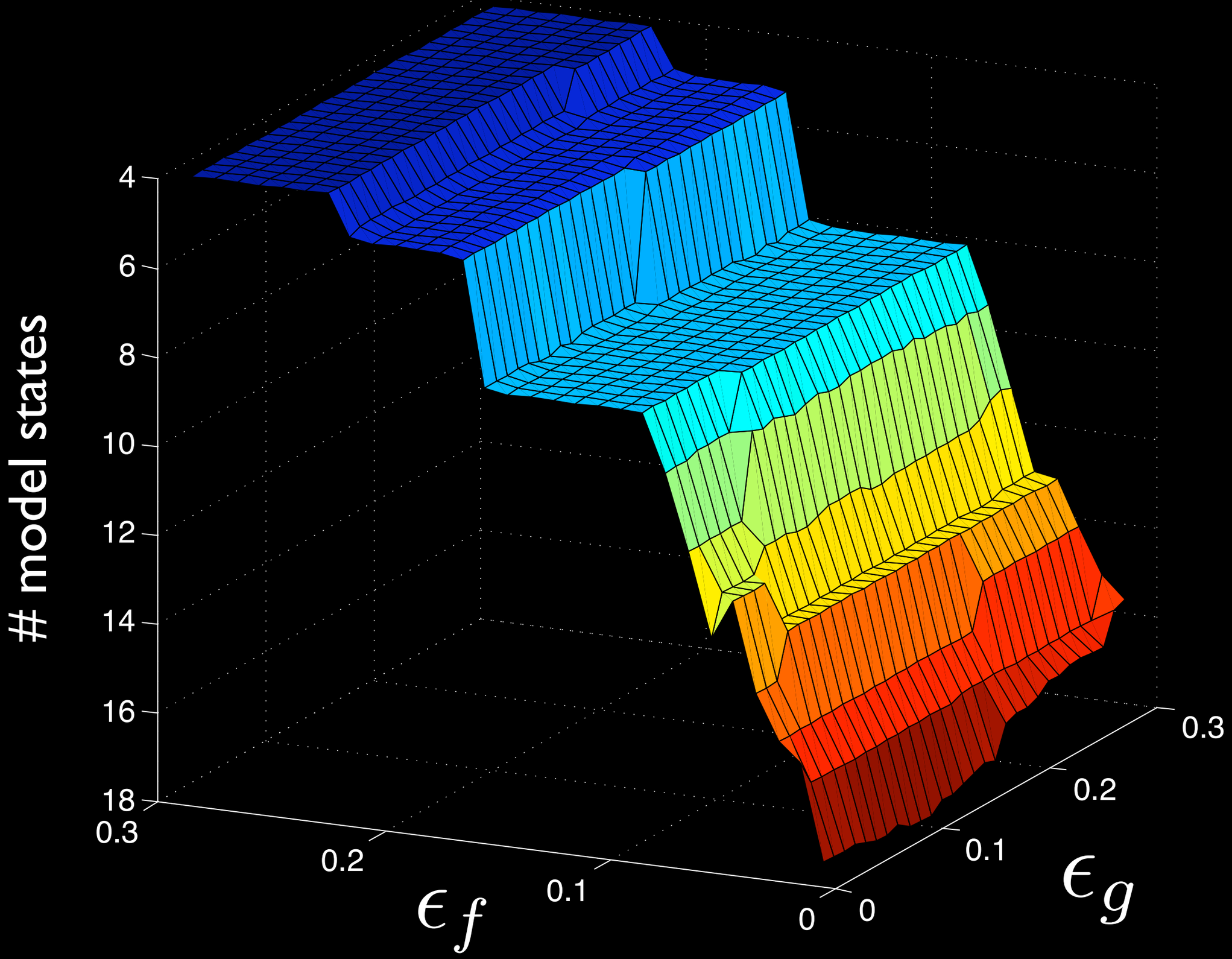


Learned representation:

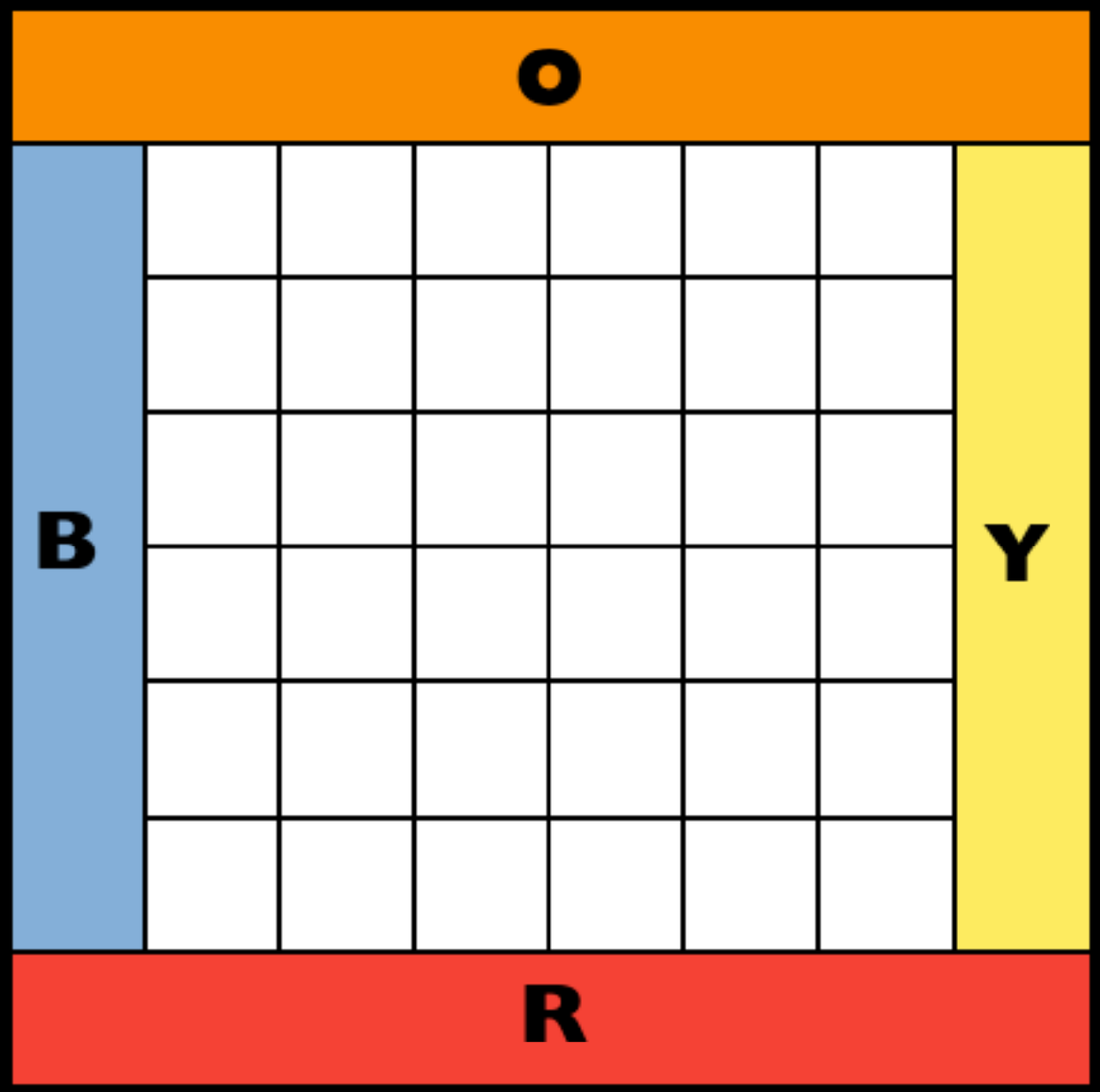



History machine
EM, 10 states, init to uniform
EM, 10 states, init to random
Total average error
trajectories
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Total average error
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0.3
0.2
0.1
0
ϵ_f
0
0.1
0.2
0.3
ϵ_g


model states
4
6
8
10
12
14
16
18
0.3
0.2
0.1
0
ϵ_f
0
0.1
0.2
0.3
ϵ_g

Grid world

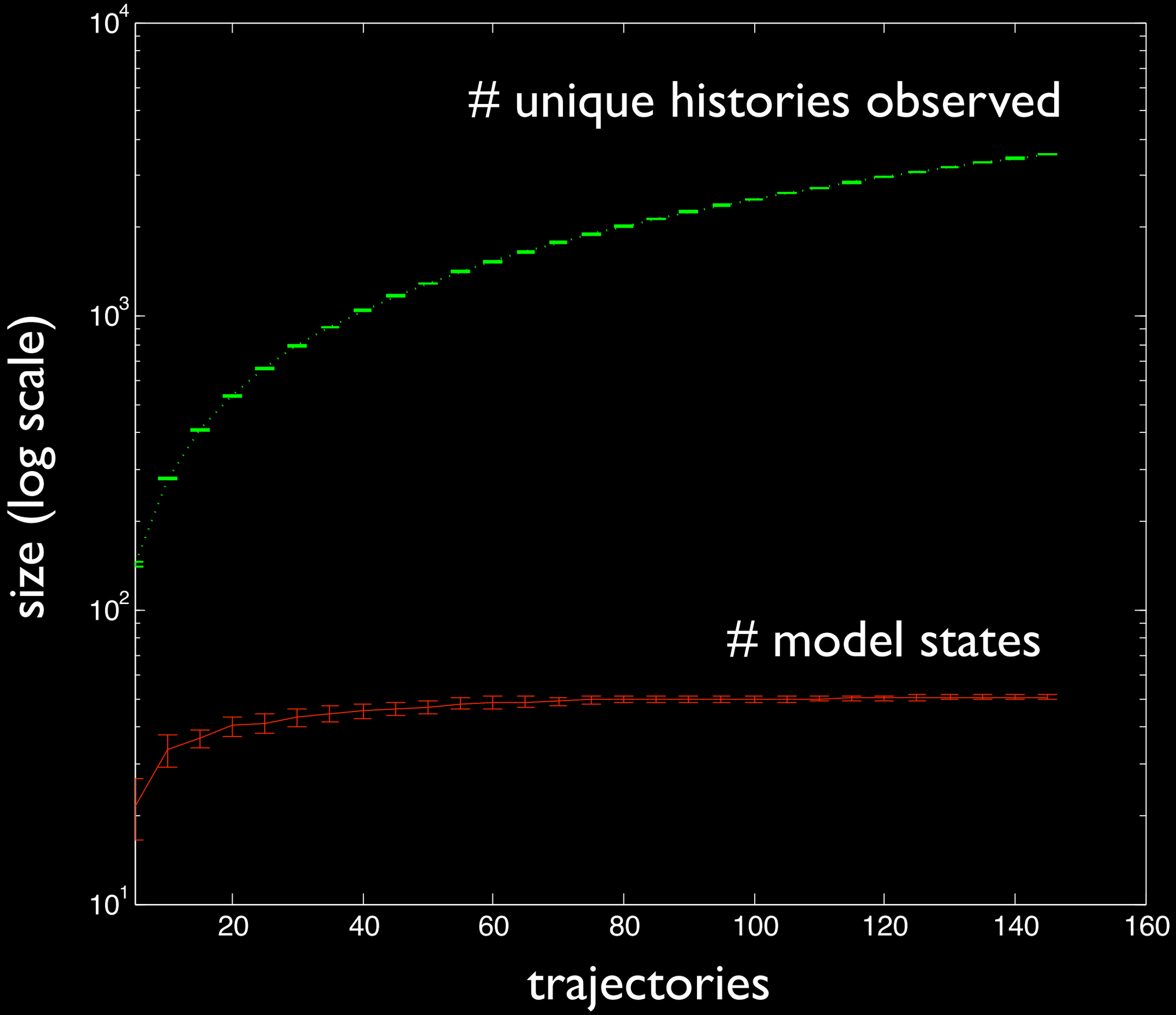


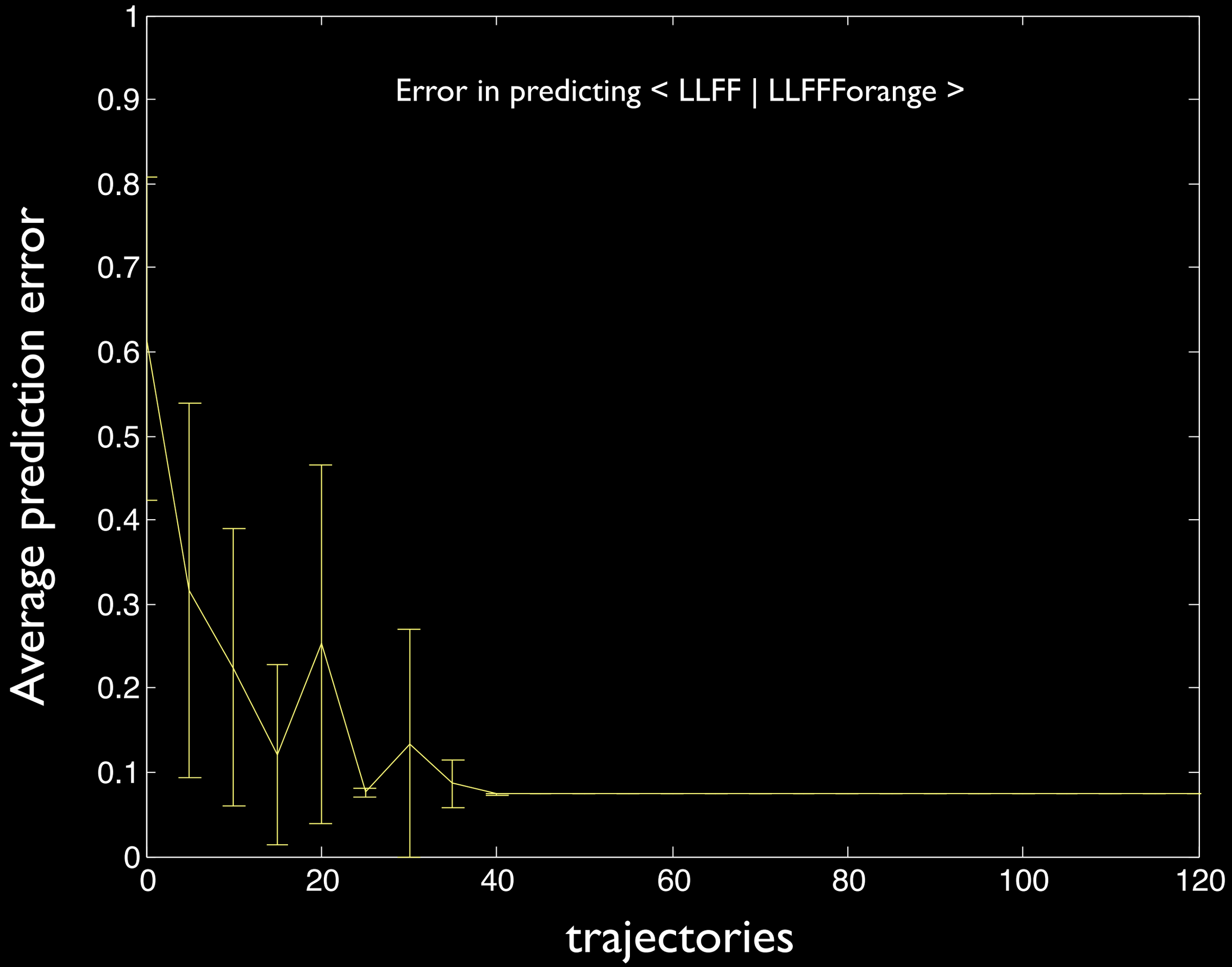
Algorithm Parameters:

$\epsilon_f = 0.04$

$\epsilon_g = 0.02$

trajectories of length 40


10^4
10^3
10^2
10^1
size (log scale)
unique histories observed
model states
20
40
60
80
100
120
140
160
trajectories


Error in predicting < LLFF | LLFFForange >
Average prediction error
trajectories
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
20
40
60
80
100
120

Error in predicting < [] | Forange >

Average prediction error

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

0
5
10
15
20
25
30
35
40

trajectories

Conclusions

- Algorithm learns an approximate model that makes accurate predictions wrt to a set of tests of interest
- Needs a small amount of data
- Future work: scaling up to larger environments
- More efficient estimation of $\langle h|t\rangle_f$

Questions?