reinforcement learning

pavlov's dog

nervous system ←  → digestion

robot flipping pancakes

http://cs.stanford.edu/groups/littledog/

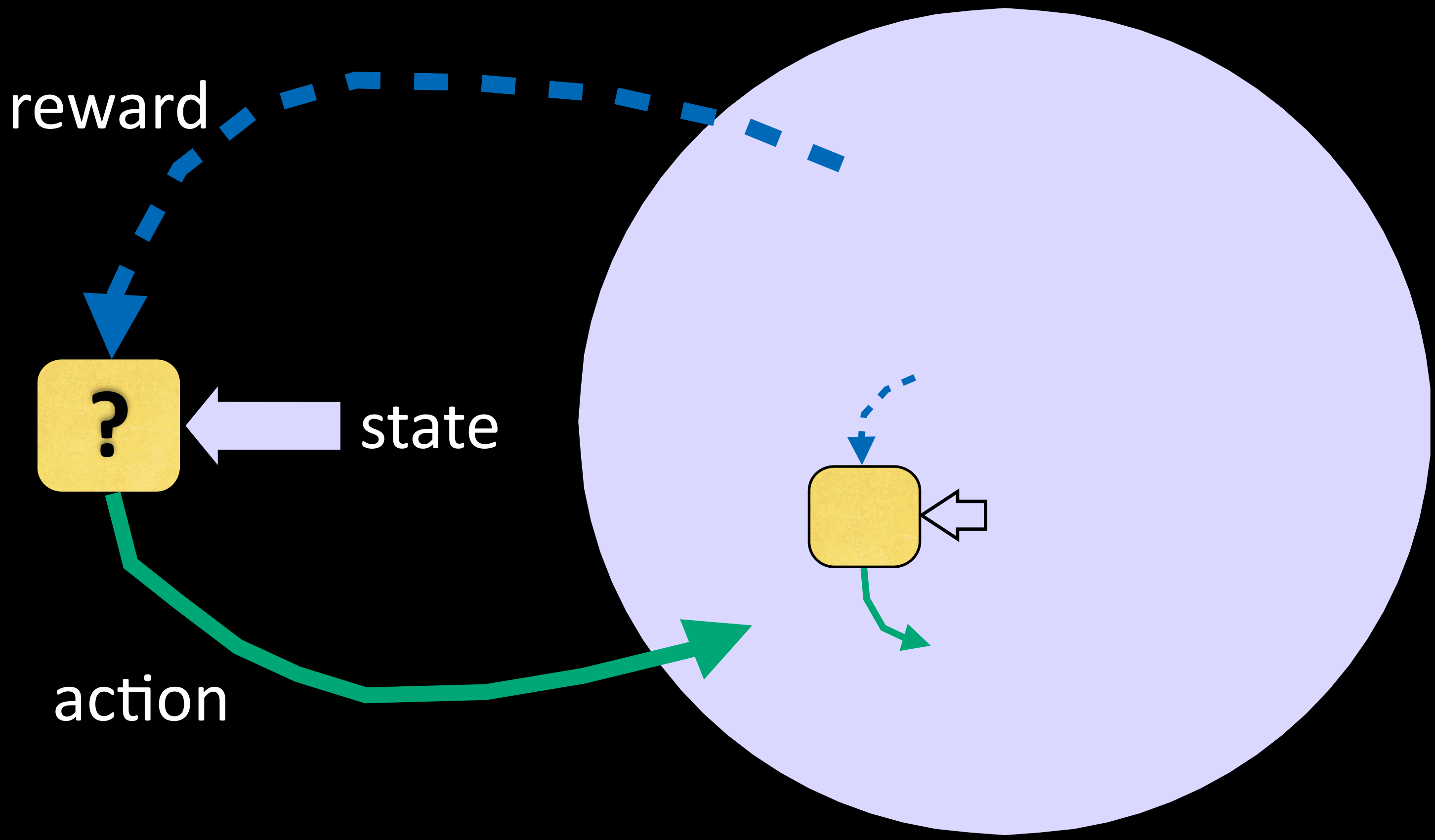
another example

- a child learning to walk:
  - tries out many different strategies
  - some do not work (**falling**), some seem to work (**staying up longer and longer**)
  - the **ones that do not work are discarded**
  - the **ones that work are tried again and again** until perfected or replaced by better strategies

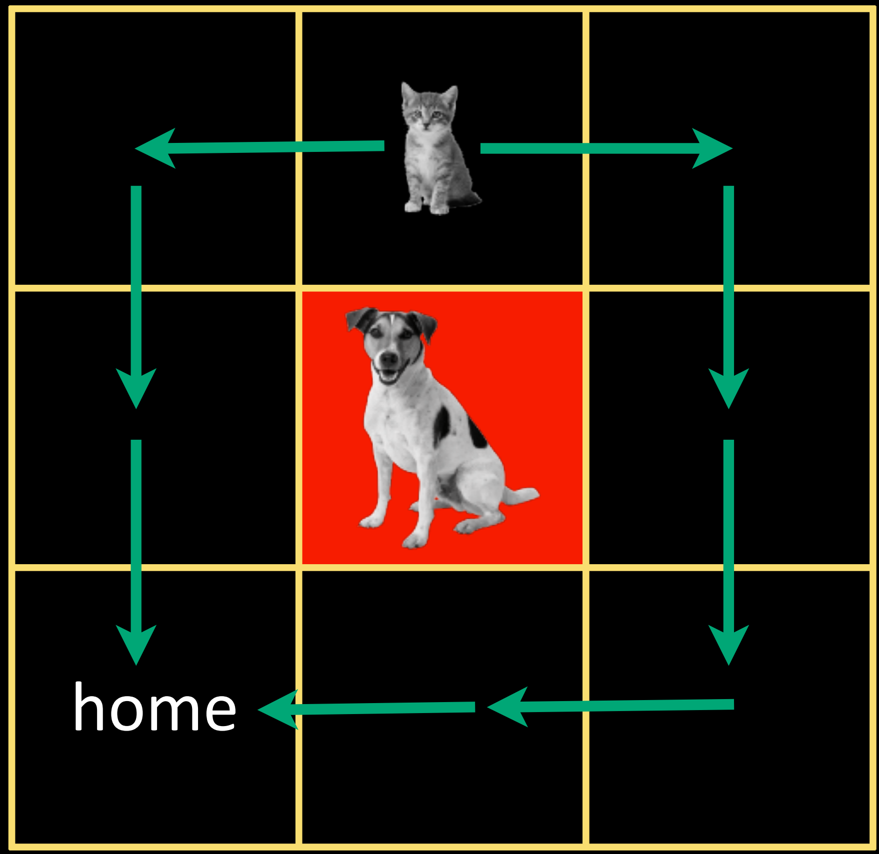
hovering... inverted!

Inverted autonomous helicopter flight via reinforcement learning, Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger and Eric Liang. In *International Symposium on Experimental Robotics*, 2004.
URL: http://heli.stanford.edu/

the problem

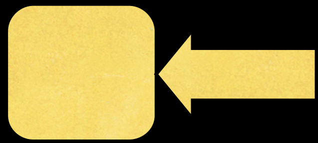


toy problem



state and action spaces

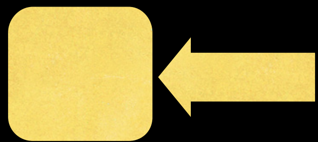
- size of these spaces can be quite large
- specifying the spaces is crucial in designing a good learning agent



5 integer values between
1 and 100: {22,44,12,67,9}

size of state space = 100 x 100 x 100 x 100 x 100

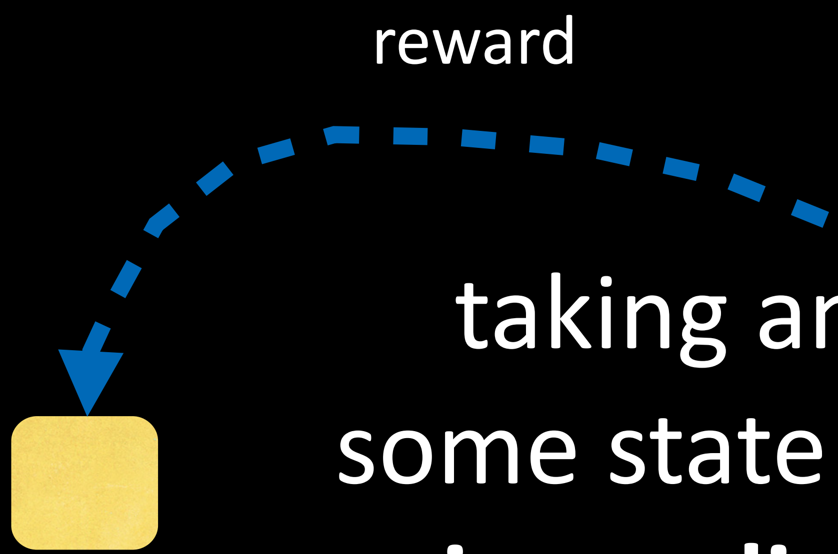
can quantise state space differently



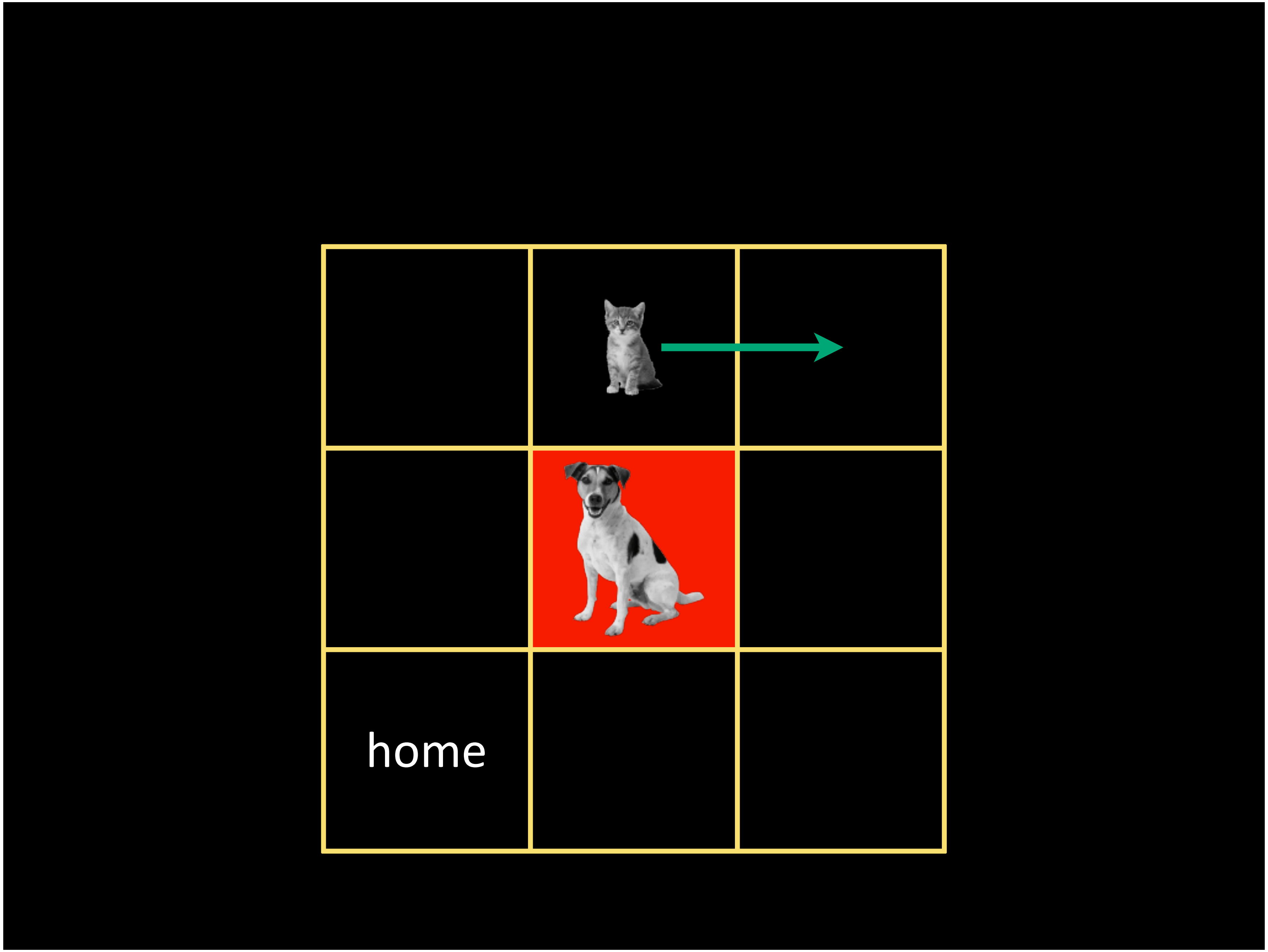
5 values belonging
to 2 classes: {1, 2, 1, 2, 1}

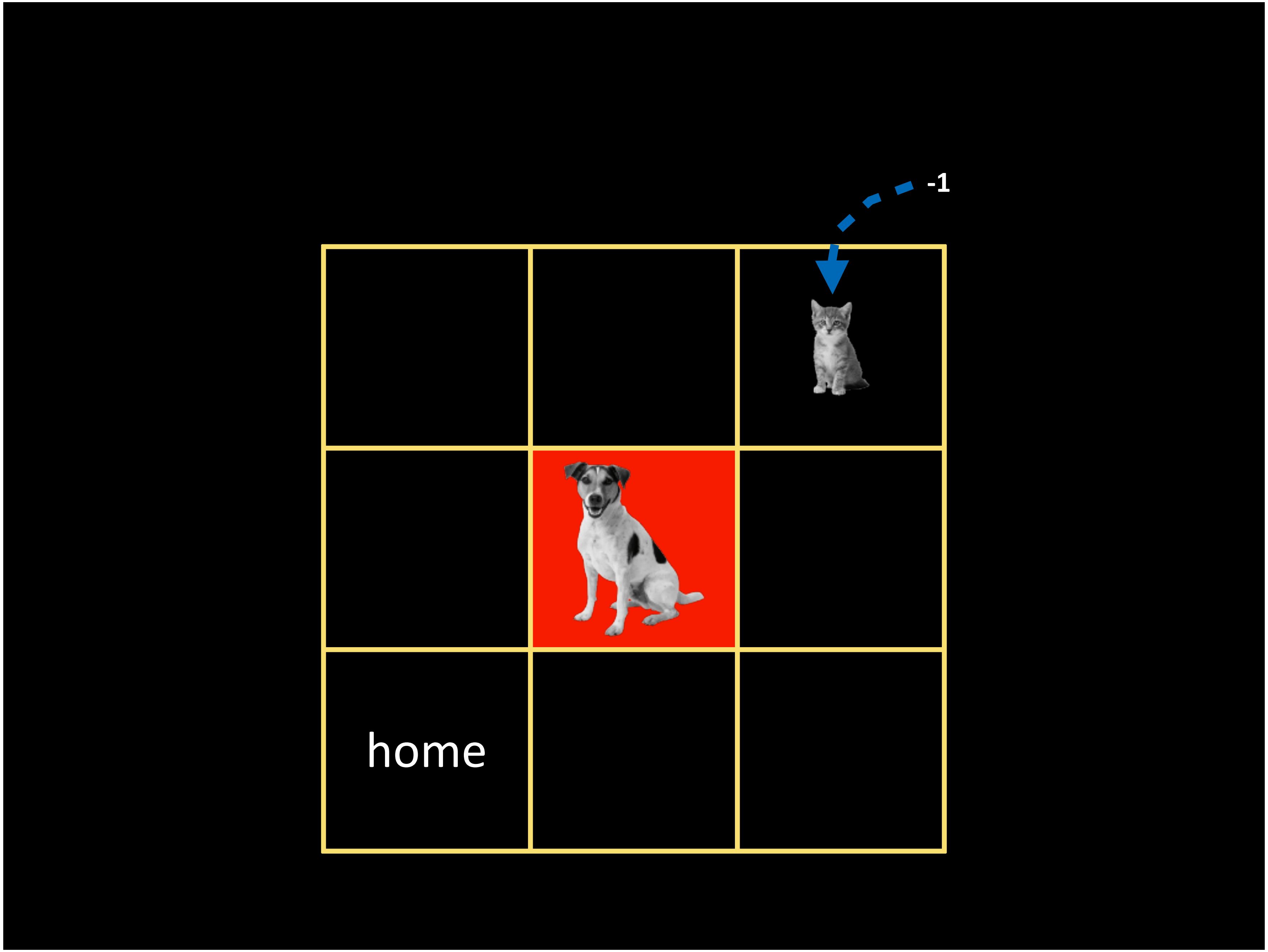
size of state space = 2 x 2 x 2 x 2 x 2

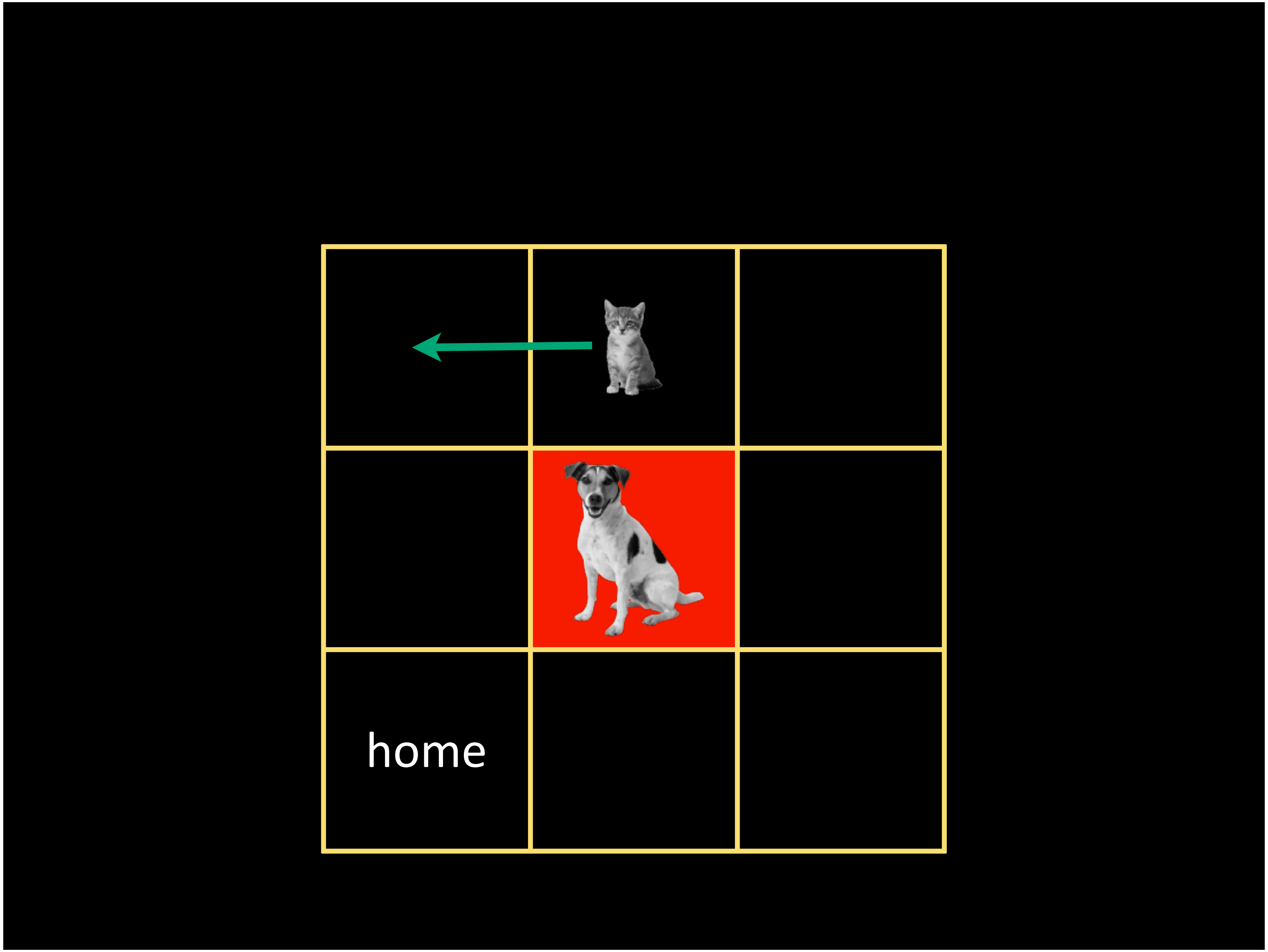
in the toy problem? 9

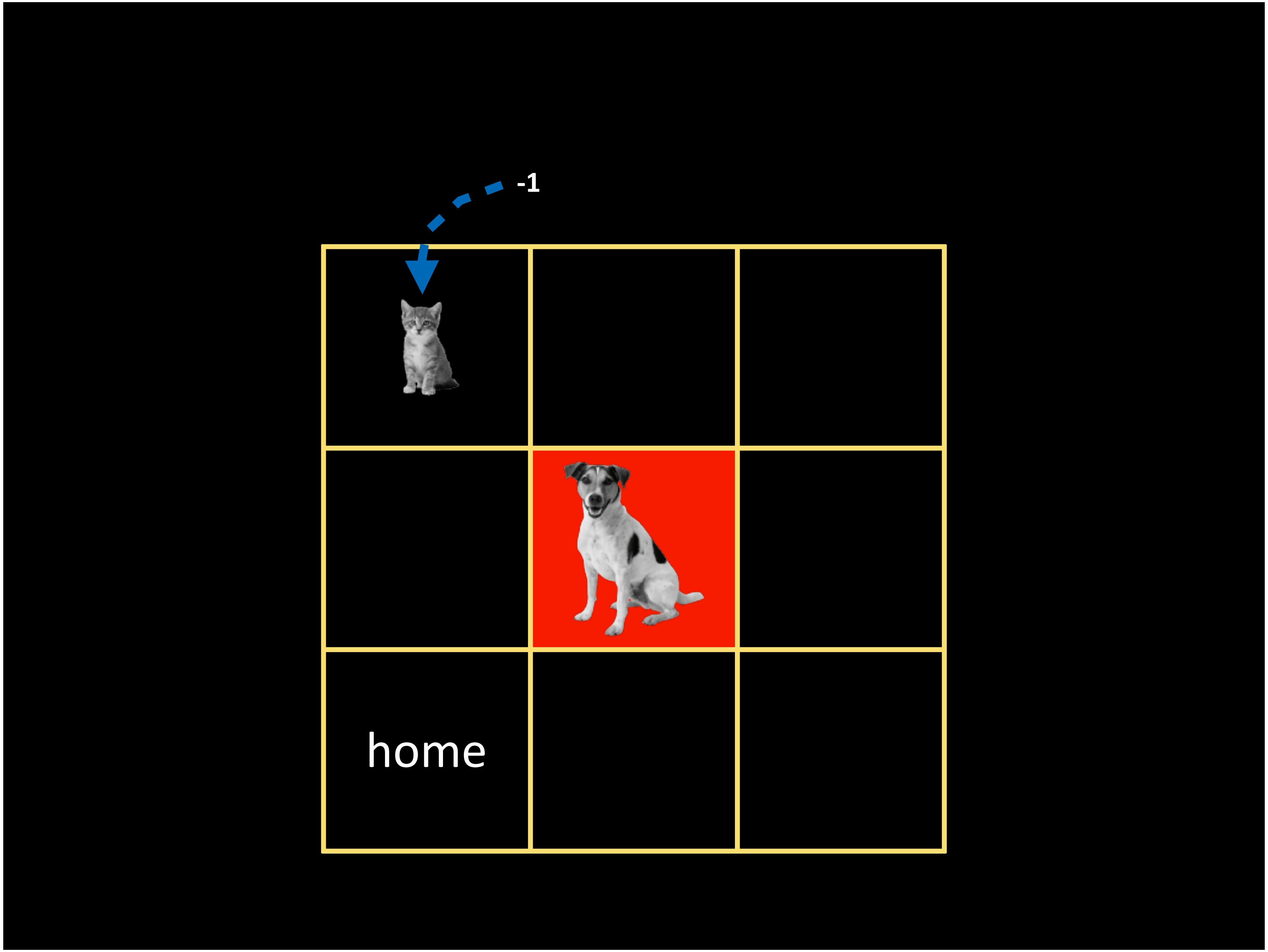


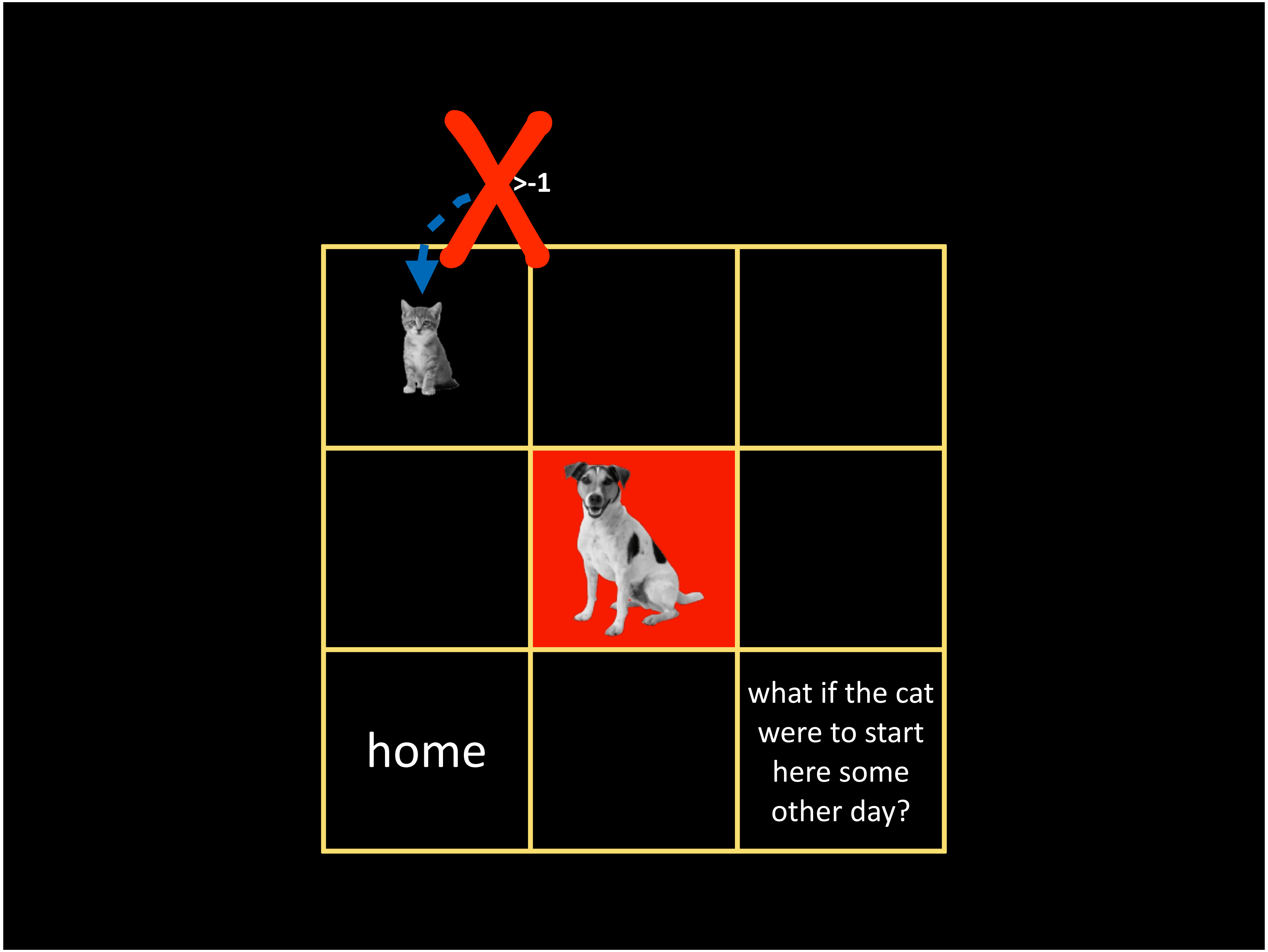
taking an action in
some state results in an
immediate reward
(can be negative)


home


-1
home


home


-1
home


>-1
home
what if the cat
were to start
here some
other day?

reward system should tell
the agent:

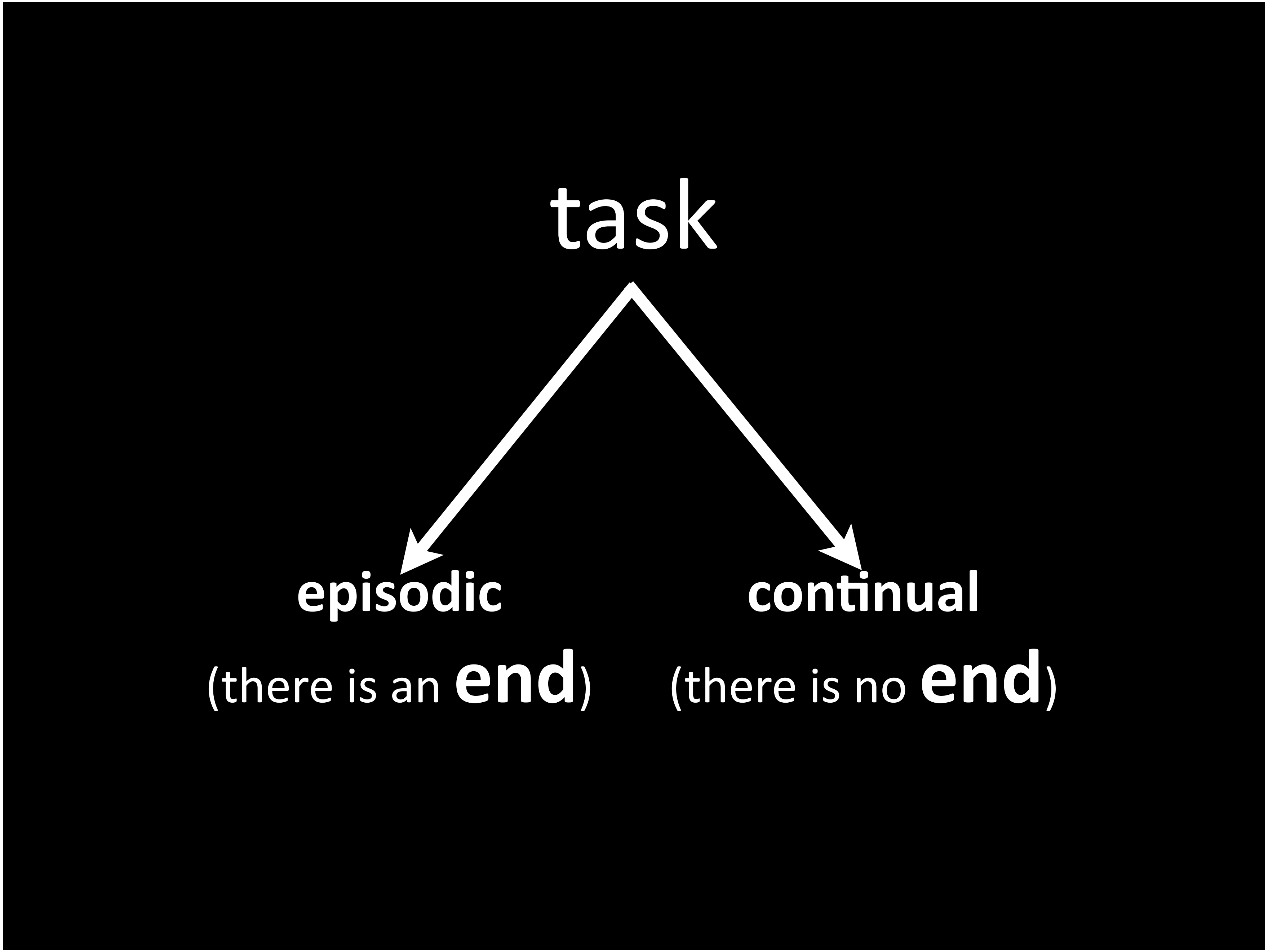
what to achieve

rather than how to achieve

but agent has to choose an action based on **expected "long term" reward** (cumulative reward in the long run)

expected "long term" reward (cumulative reward in the long run)

?


task
episodic
(there is an end)
continual
(there is no end)

episodic

(there is an **end**)

agent taking **finite (say 5) steps** till the end...

should act based on the

average of the following

$$R_0 = r_1 + r_2 + r_3 + r_4 + r_5$$

continual

(there is no **end**)

agent can continue acting for **infinite steps in time...**

should **discount** future rewards and act based on the **average of the following**

$$R_0 = r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 + \gamma^4 r_5 + \ldots$$

discount

future reward is probably **more uncertain** than **immediate reward**

shortsighted?
ϒ=0

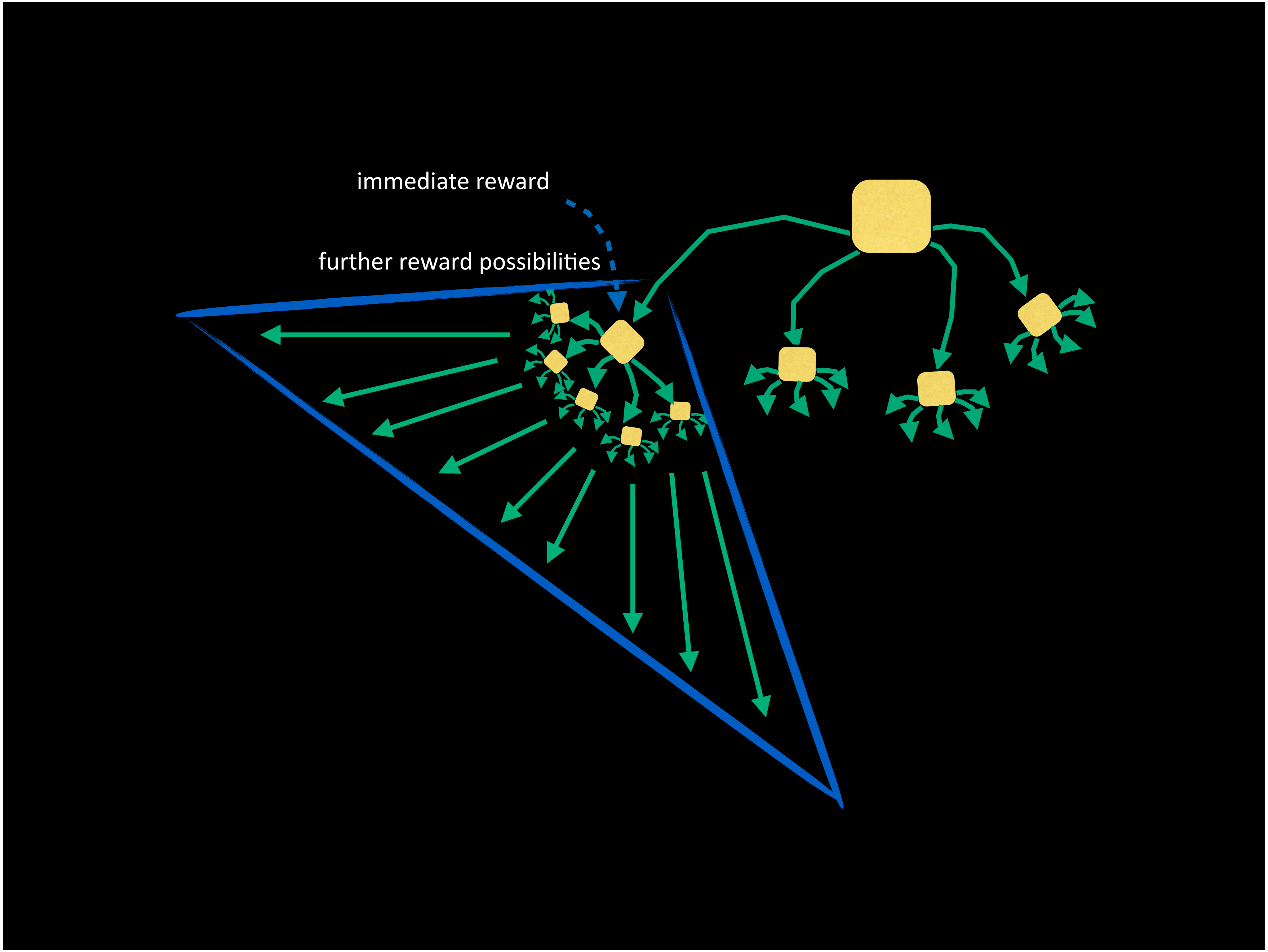
$$0 \le \gamma \le 1$$

farsighted?
ϒ=1

$$R_0 = r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 + \gamma^4 r_5 + \ldots$$

$$R_0 = \sum_{k=0}^{T} \gamma^k r_{k+1}$$

$$\mathrm{E}\left\{ R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \right\}$$


immediate reward
further reward possibilities

expected
“long term” reward

$$\mathbf{E}\left\{ \boxed{R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1}} \right\}$$

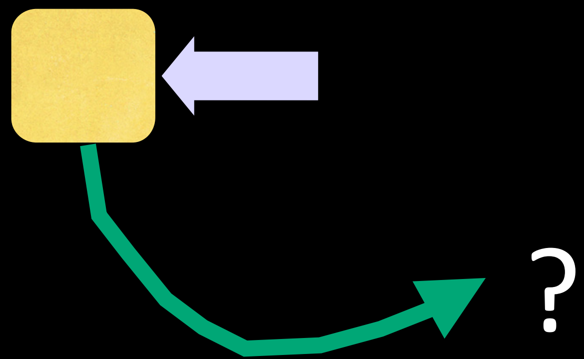
but these expected
rewards are
not known to agent
beforehand!

whether they are known or not, the
agent has to act somehow!

how to act/action selection?

how to get to know/estimate?

action selection?



values of each possible action

in the current state?

expected reward for

carrying out the action is its **value**

but **what are these values?**

<<expected rewards are not known>>

<<actions based on expected rewards>>

these expected rewards **$E\{R_t\}$** are to be **estimated by agent whilst acting!**

Q

	a	b	c
1	2	0	1
2	3	0	-1
3	-5	6	2
4	2	3	1
.	.	.	.
.	.	.	.
.	.	.	.
n	7	8	7

agent maintains **values**
for actions within each state

selects actions using these values based on a
“policy”

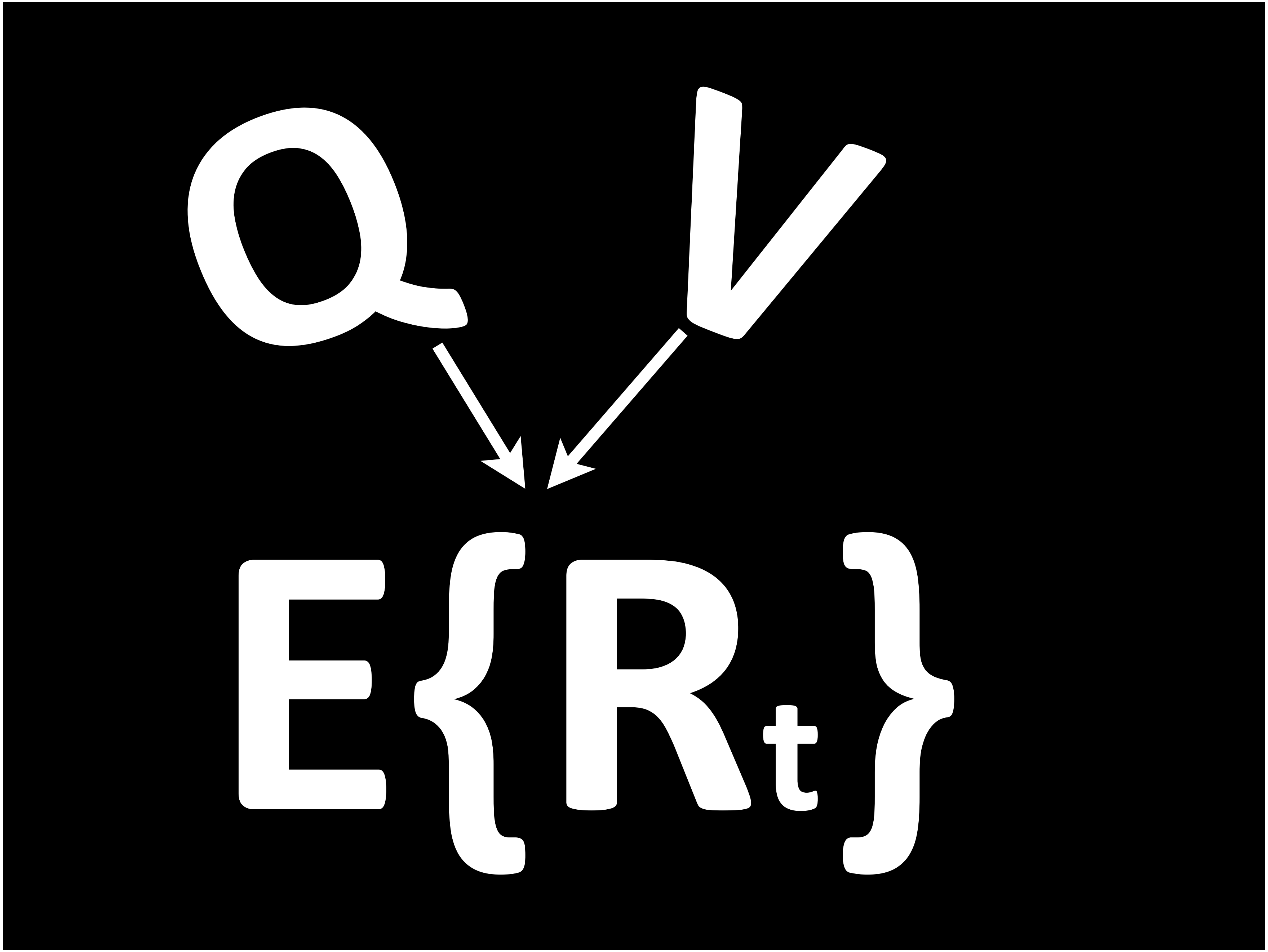
V

1	2
2	3
3	-5
4	2
.	.
.	.
.	.
n	7

agent **maintains state values**

selects actions using these values based on a

“policy”


Q
V
$E\{R_t\}$

policy?

probability of choosing an action, given a state

π

$Q^{\pi}(s,a)$

$V^{\pi}(s)$

usual policies

greedy

choose
best action

ε-greedy

choose
best action with probability 1-ε

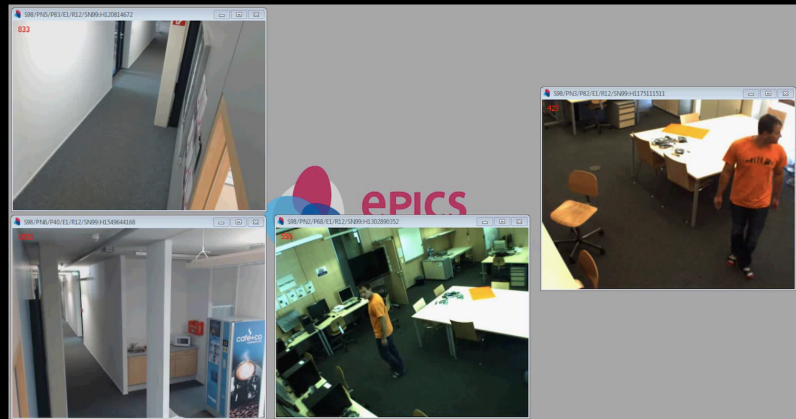
soft-max

choose
action with probability given by its value

exploration vs. exploitation

policy = multi-armed bandit strategy

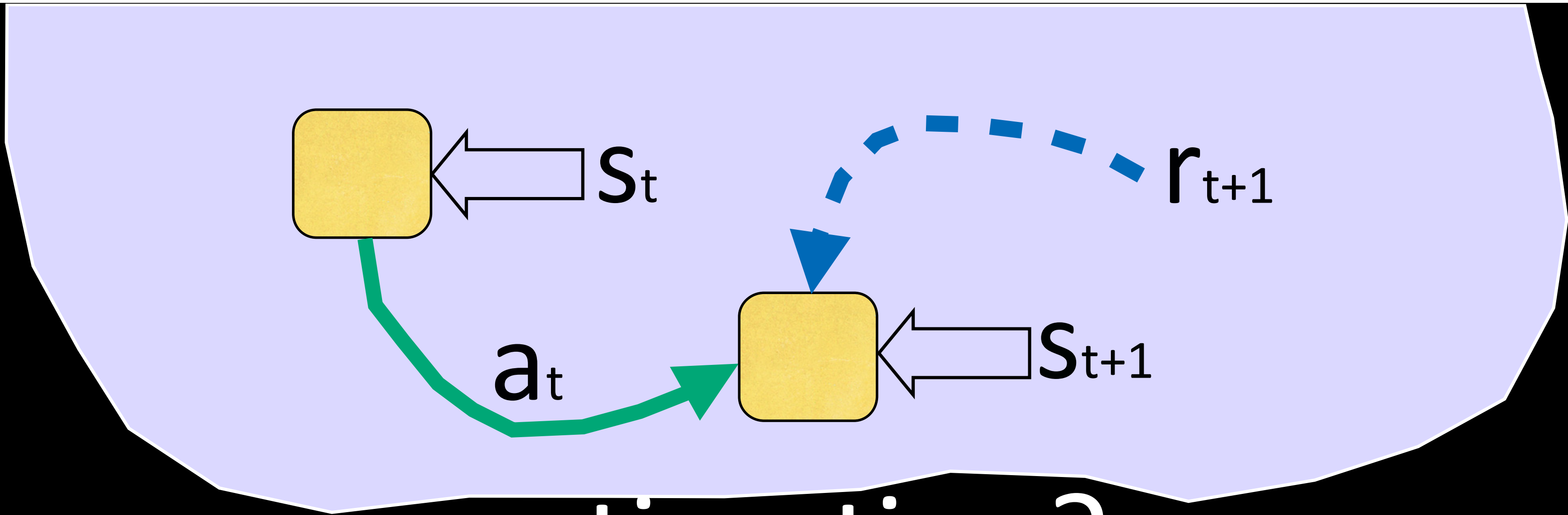
Yamaguchi先生, http://en.wikipedia.org/wiki/File:Las_Vegas_slot_machines.jpg



Learning to be different: Heterogeneity and efficiency in distributed smart camera networks, P. R. Lewis, L. Esterle, A. Chandra, B. Rinner, and X. Yao, In Proceedings of the IEEE Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2013.

(forthcoming) Static, dynamic and adaptive heterogeneity in socio-economic distributed smart camera networks, P. R. Lewis, L. Esterle, A. Chandra, B. Rinner, J. Torresen, and X. Yao, ACM Transactions on Autonomous and Adaptive Systems (TAAS), ACM, 2014.

A/B Testing



estimation?

<<use currently visible rewards to update values of where you are coming from>>

the current state (or state-action pair) has an estimated value (say zero/random initially),

which can be **used together with r_{t+1} to update value of previous state (or state-action pair)**

i.e.

fraction of (currently visible rewards - old value)

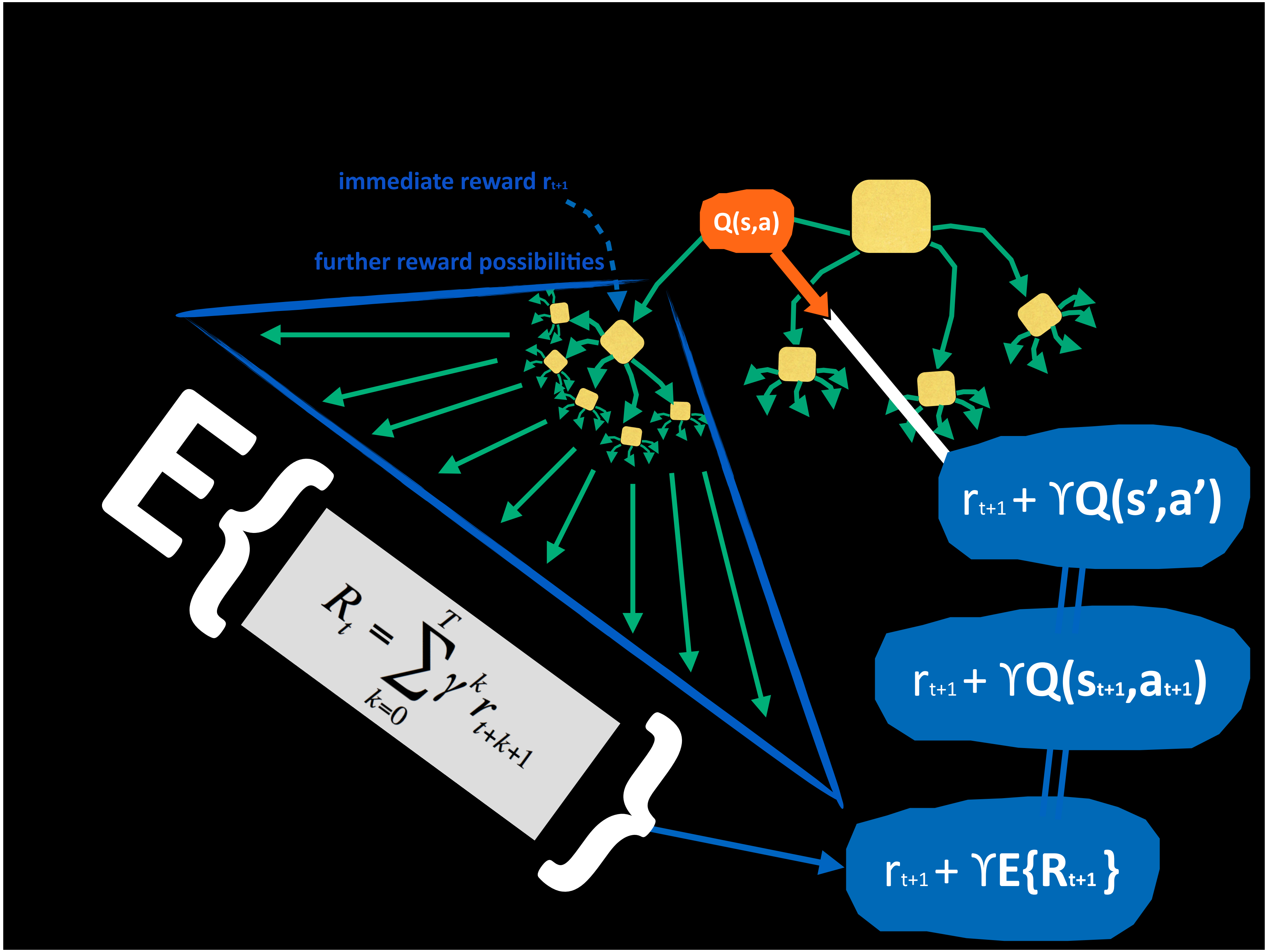
+

old value

↓

new value

(1-fraction) old value + fraction curr. vis. rewards



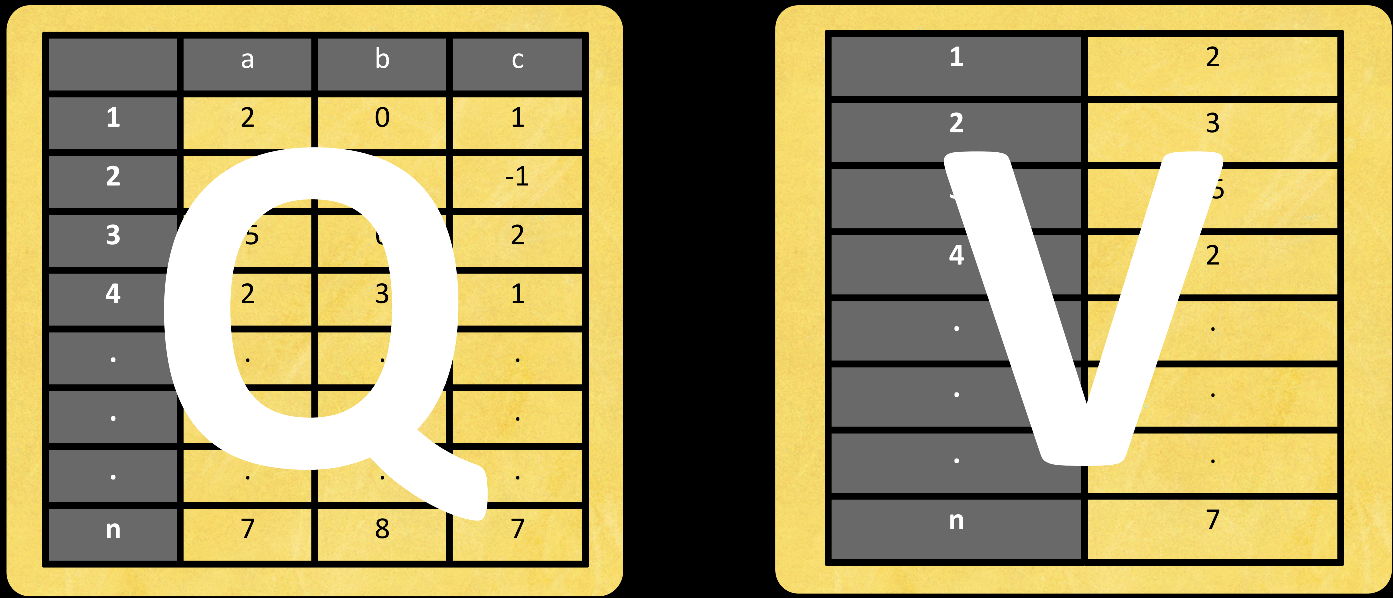
immediate reward r_{t+1}
further reward possibilities
Q(s,a)
E{ $R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1}$ }
$r_{t+1} + \gamma Q(s',a')$
$r_{t+1} + \gamma Q(s_{t+1},a_{t+1})$
$r_{t+1} + \gamma E\{R_{t+1}\}$

$$V(s) \leftarrow V(s) + \mu(r + \gamma V(s') - V(s))$$

e.g.

$$Q(s,a) \leftarrow Q(s,a) + \mu(r + \gamma Q(s',a') - Q(s,a))$$

e.g. **update**
a lookup table maintaing expected rewards



$$Q(s,a) \leftarrow Q(s,a) + \mu(r + \gamma Q(s',a') - Q(s,a))$$

let's play with a version of the above update rule:

$$Q(s,a) \leftarrow Q(s,a) + \mu(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

indicates a’ to be the action with maximum value in next state s’

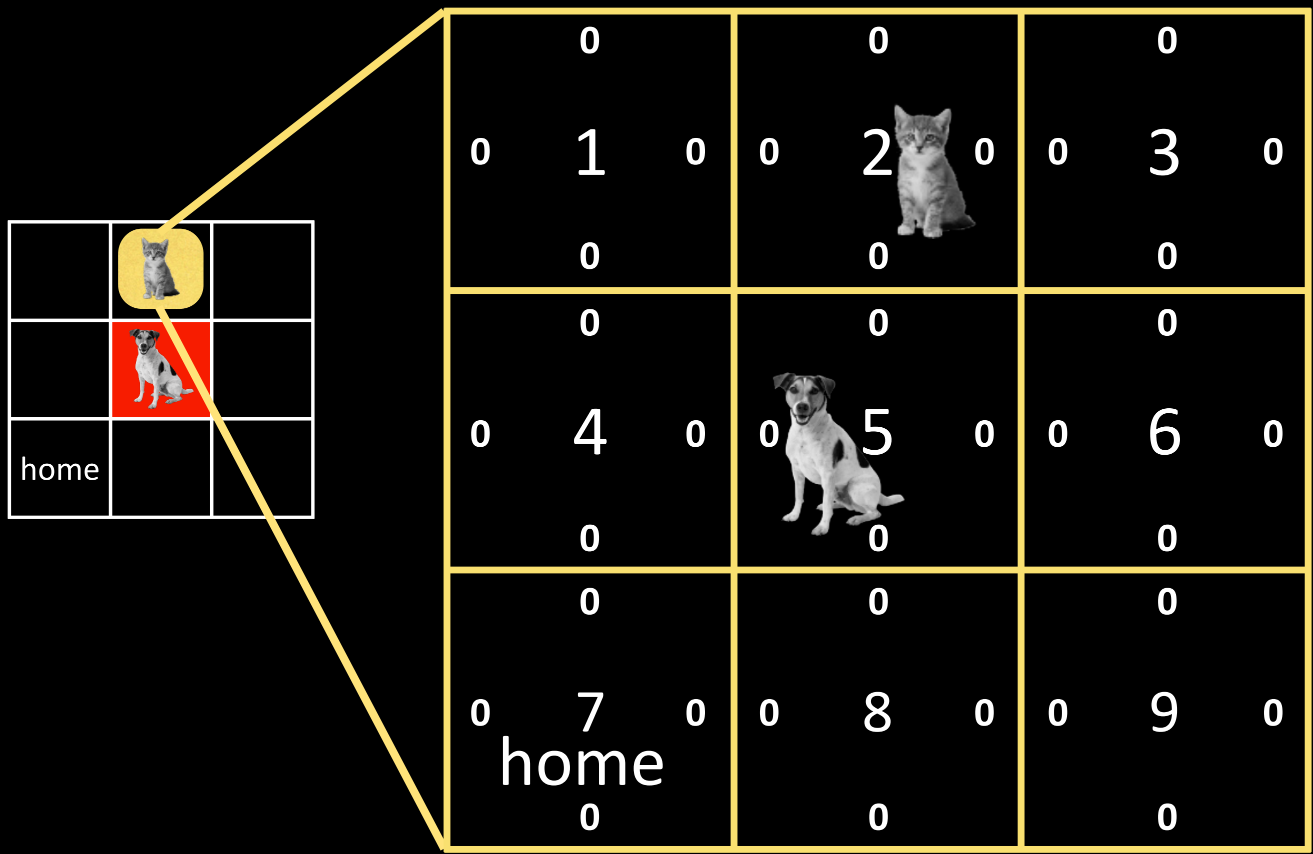
let’s play with a version of the above update rule:

$$Q(s,a) \leftarrow Q(s,a) + \mu(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

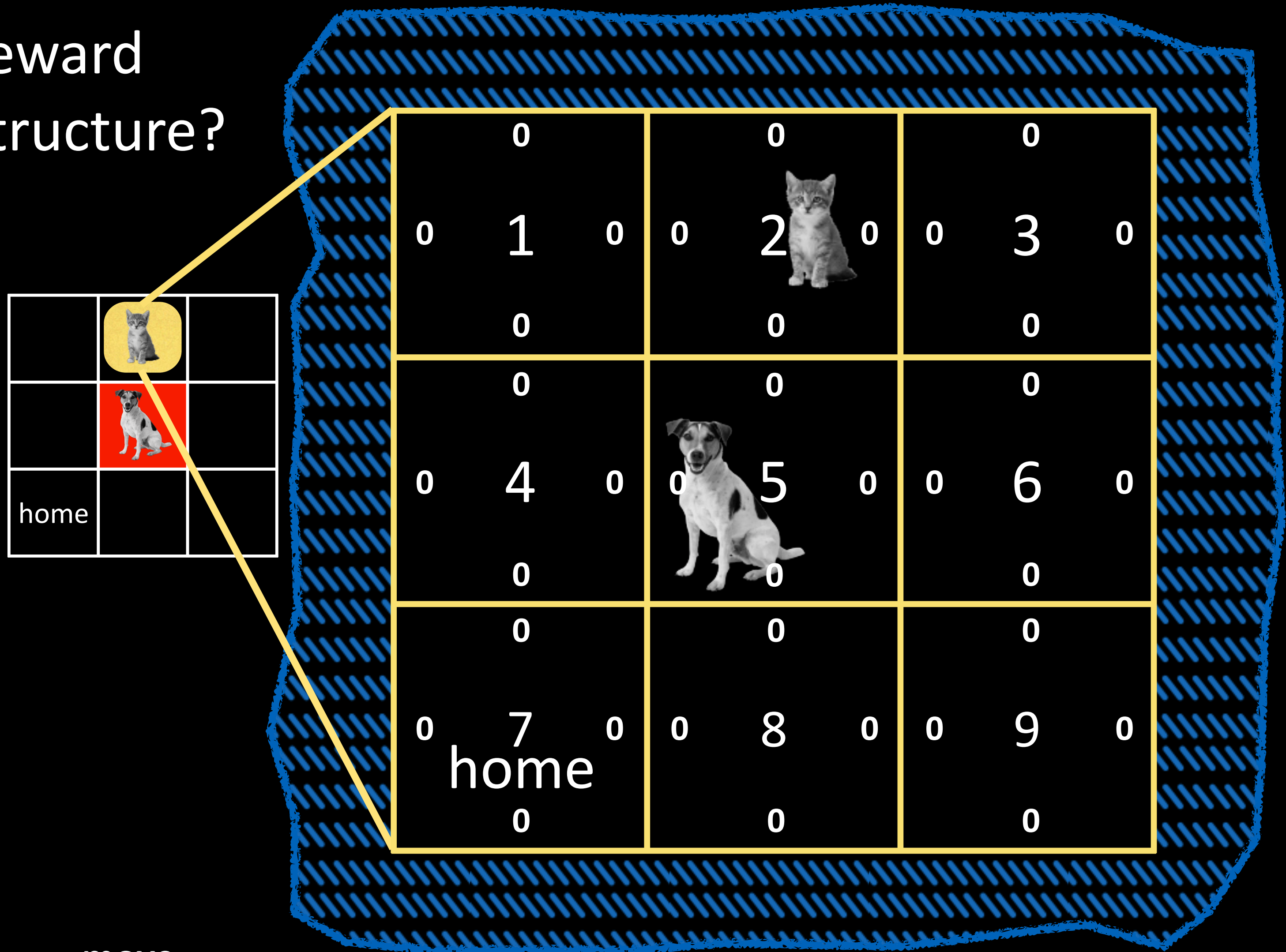
our toy problem
lookup table

	N	S	E	W
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0

our toy problem
lookup table



reward structure?



move…

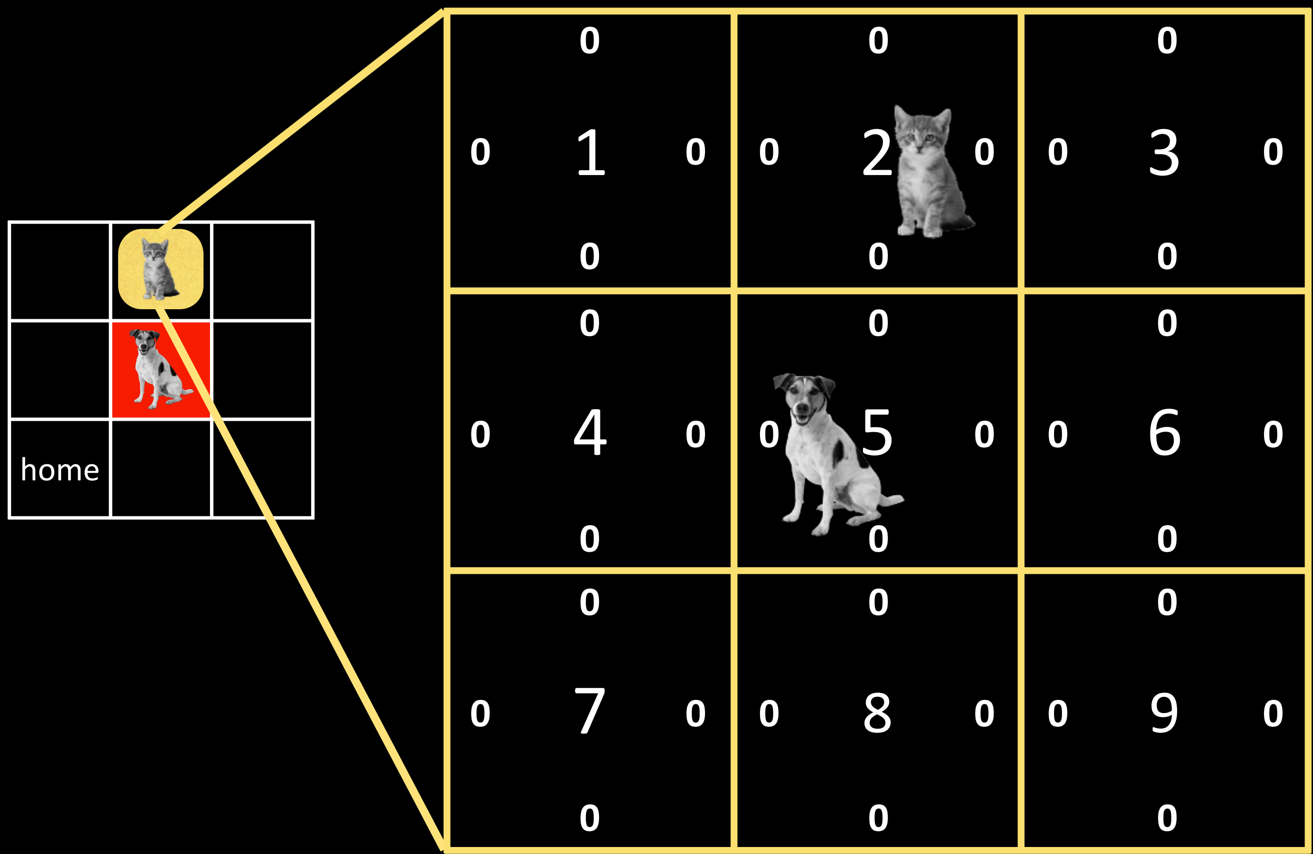
to any cell except 5 and 7: **-1**

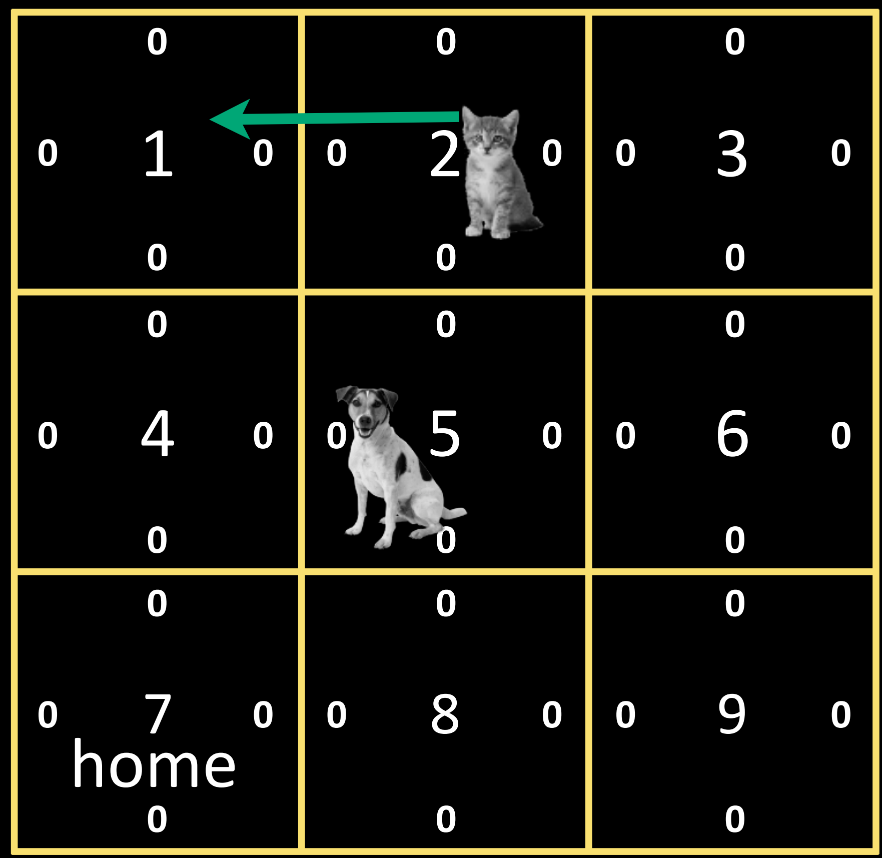
out of bounds: **-5**

to 5: **-10**

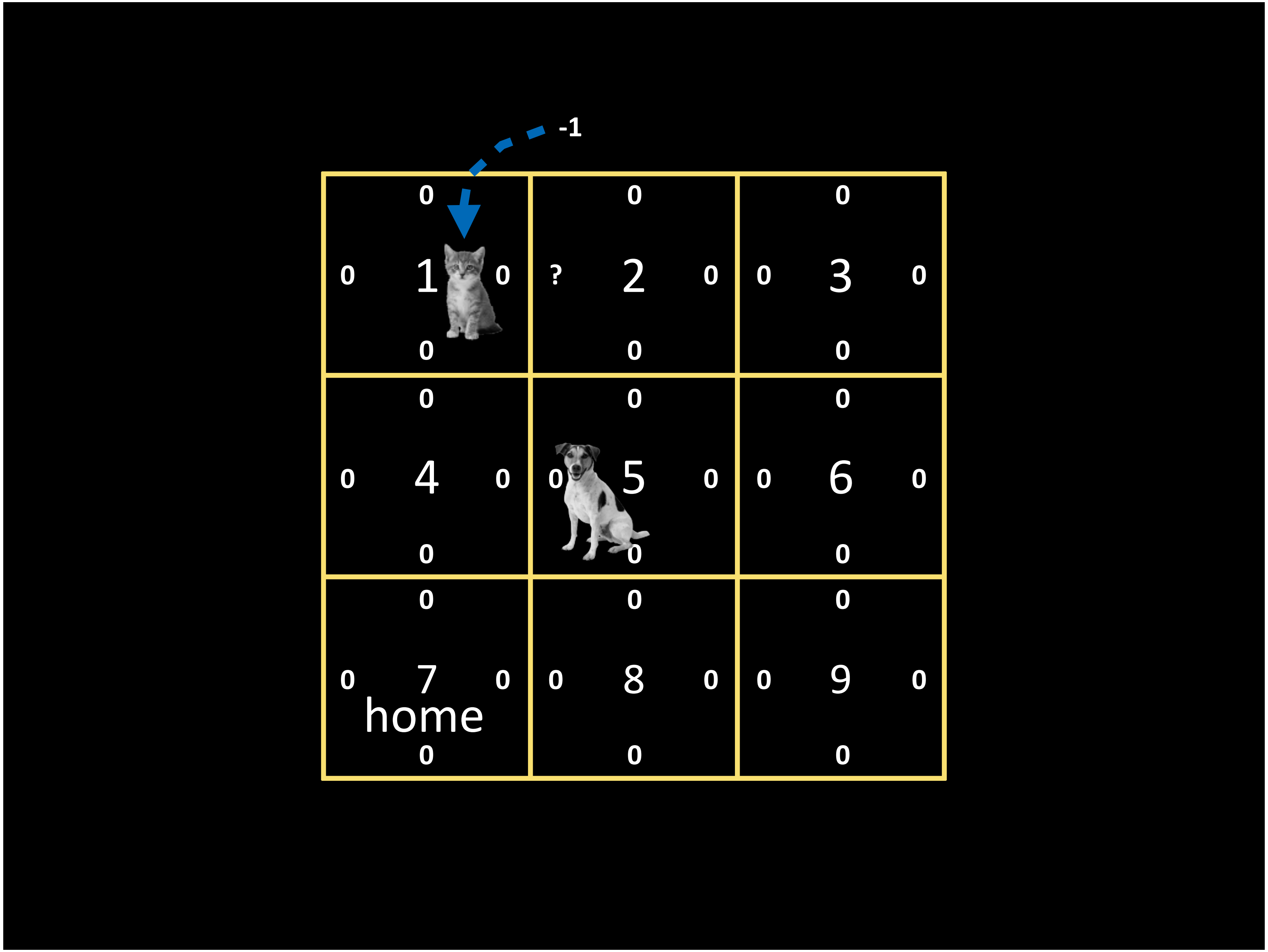
to 7/home: **10**

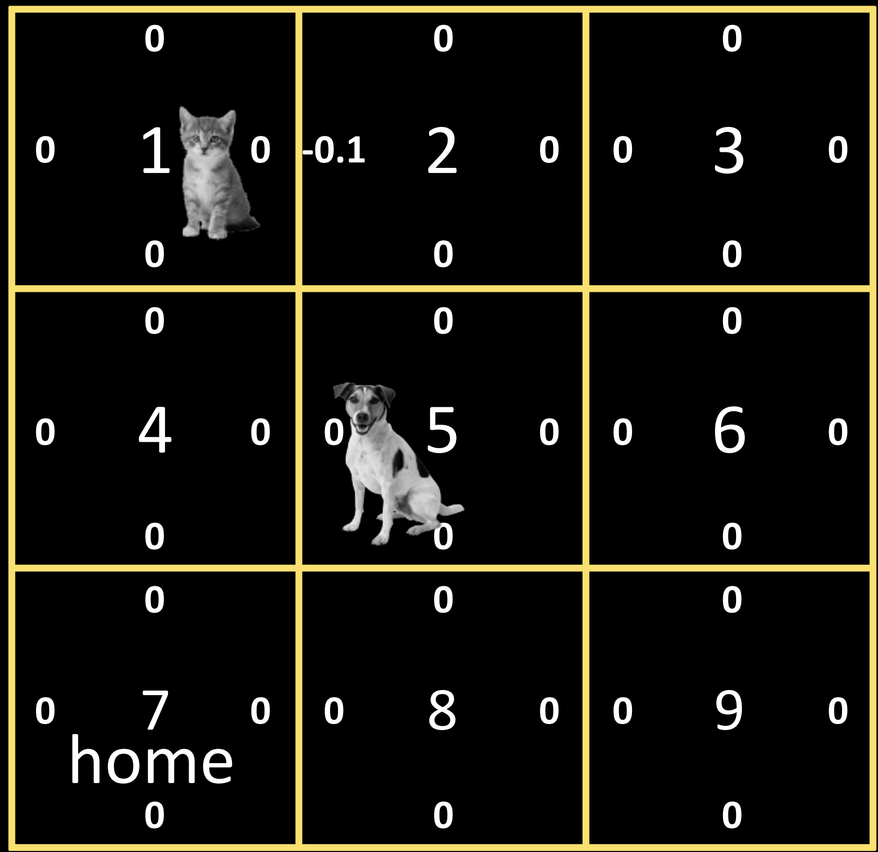
let's fix $\mu = 0.1$, $\gamma = 0.5$



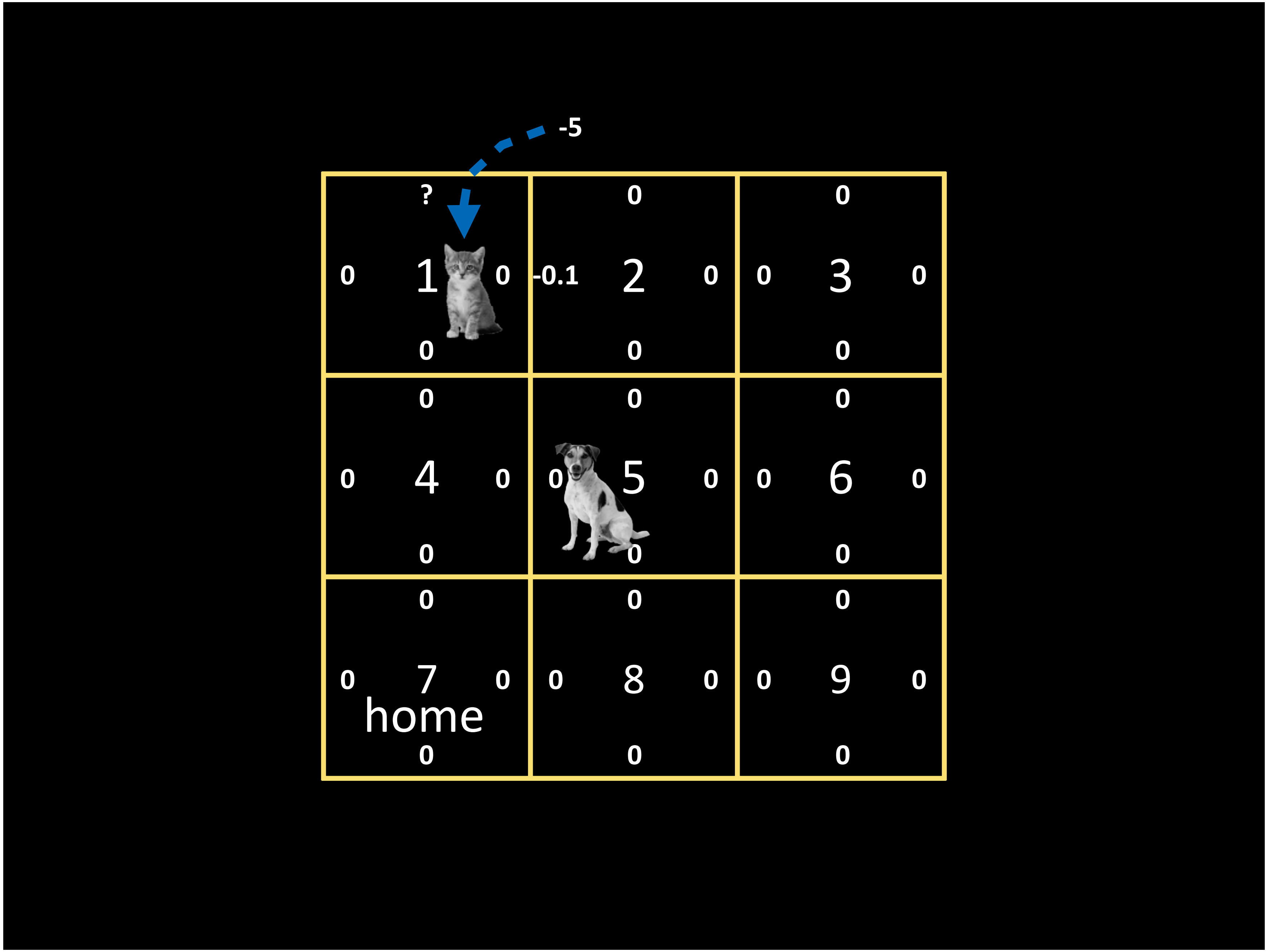


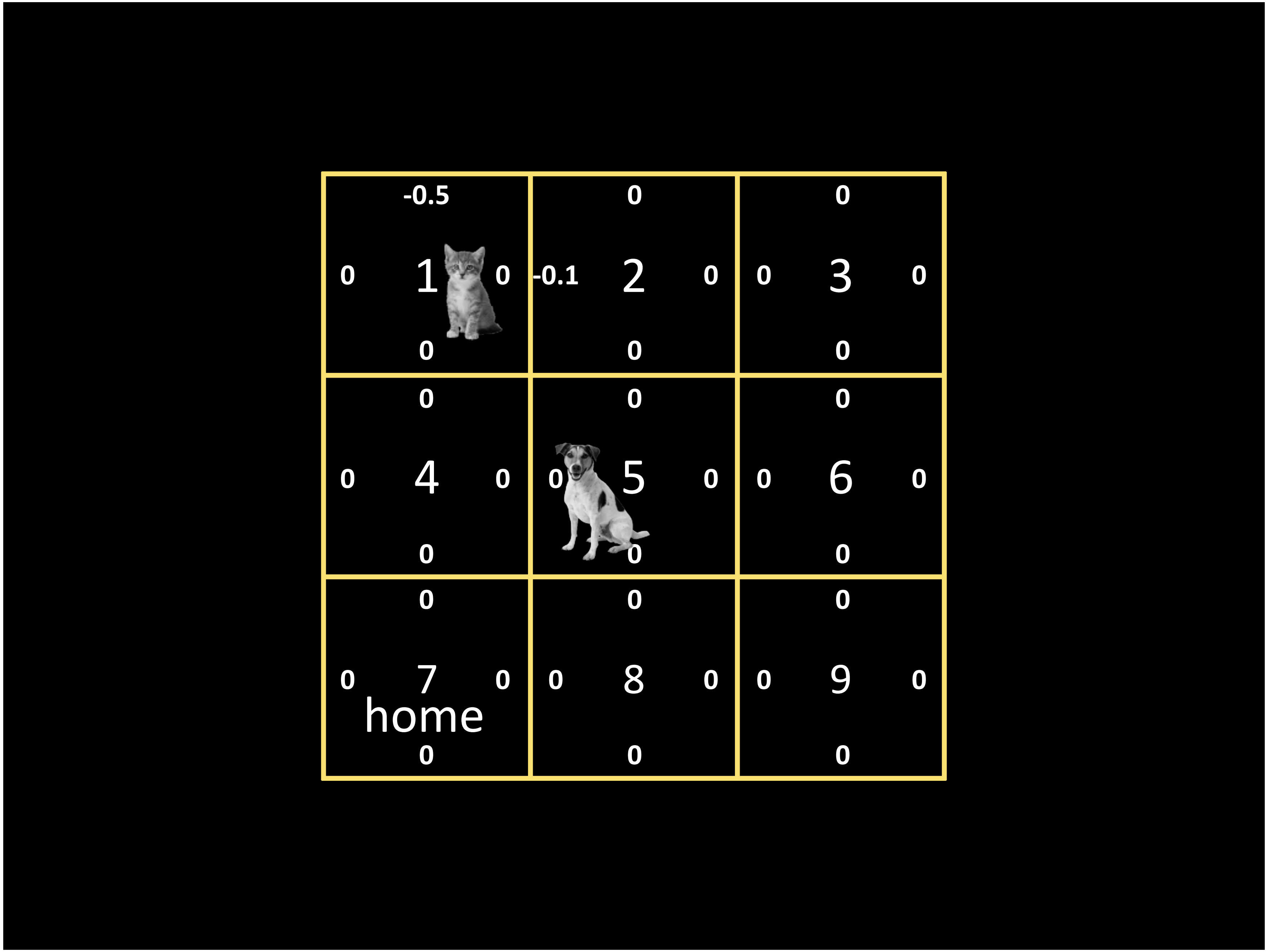
episode 1 begins...

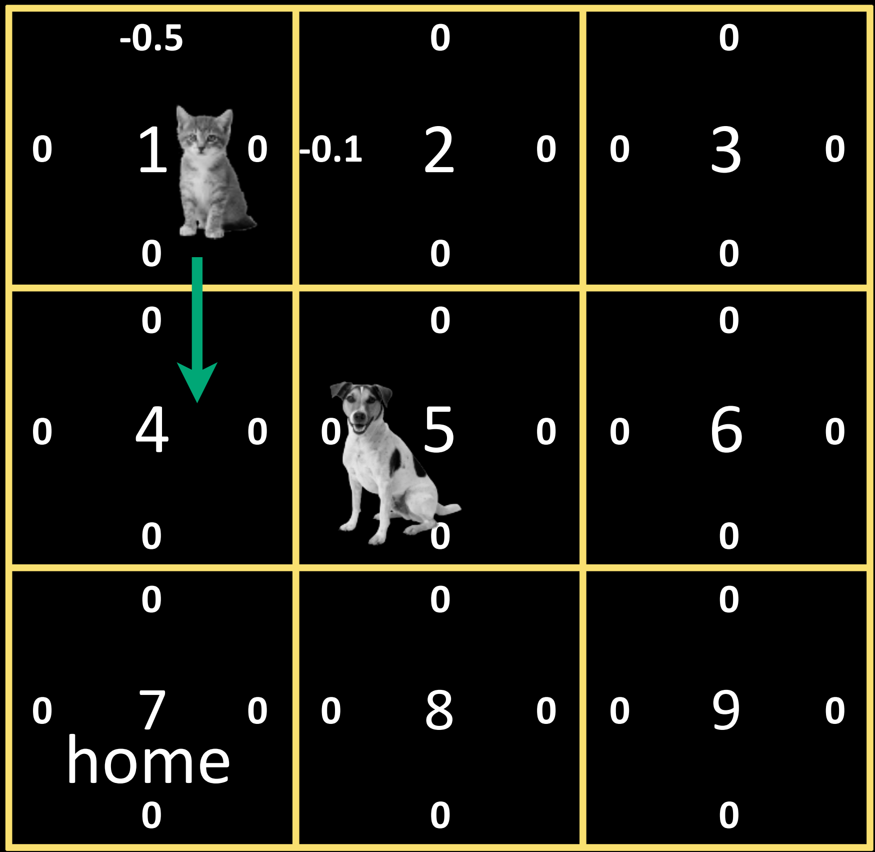

-1
0
0 1 0
0
0
? 2 0
0
0
0 3 0
0
0
0 4 0
0
0
0 5 0
0
0
0 6 0
0
0
0 7 0
home
0
0
0 8 0
0
0
0 9 0
0


0 0 0
0 1 0 -0.1 2 0 0 3 0
0 0 0
0 0 0
0 4 0 0 5 0 0 6 0
0 0 0
0 0 0
0 7 0 0 8 0 0 9 0
home
0 0 0

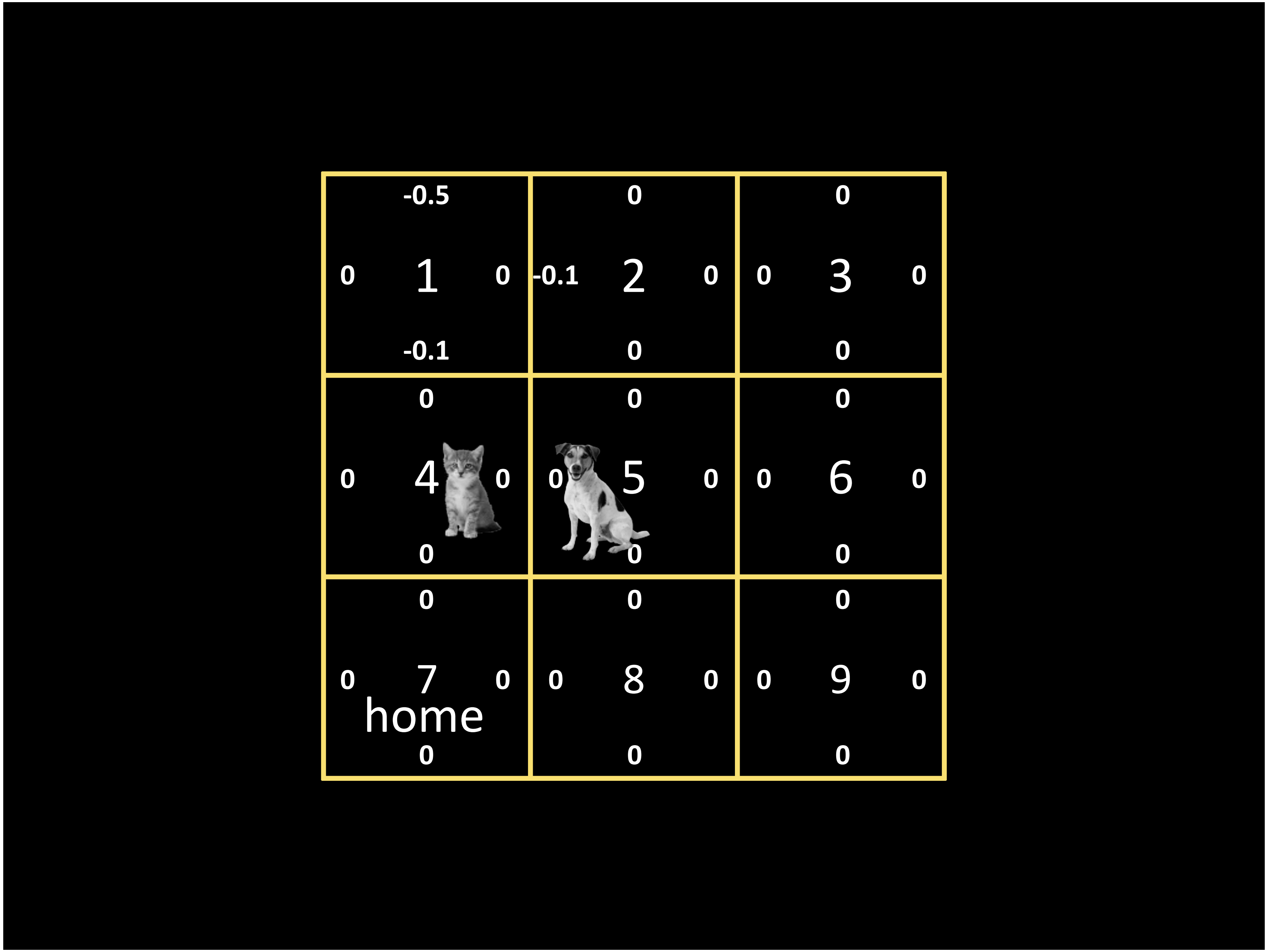

0
0 1 0
0
0
-0.1 2 0
0
0
0 3 0
0
0
0 4 0
0
0
0 5 0
0
0
0 6 0
0
0
0 7 0
home
0
0
0 8 0
0
0
0 9 0
0

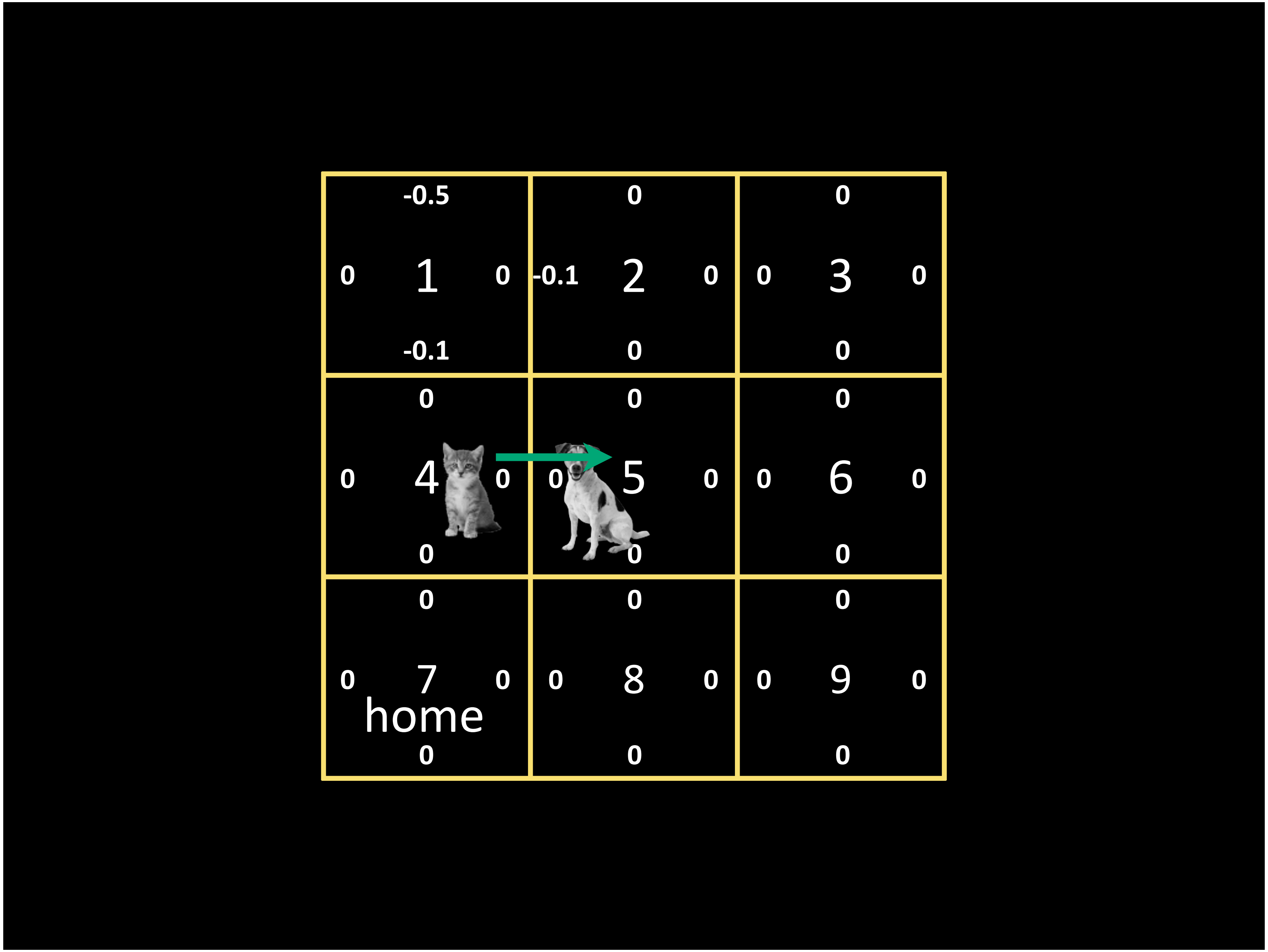

-5
?
0 1 0
0
-0.1 2 0
0
0 3 0
0
0
0 4 0
0
0
0 5 0
0
0
0 6 0
0
0
0 7 0
home
0
0
0 8 0
0
0
0 9 0
0

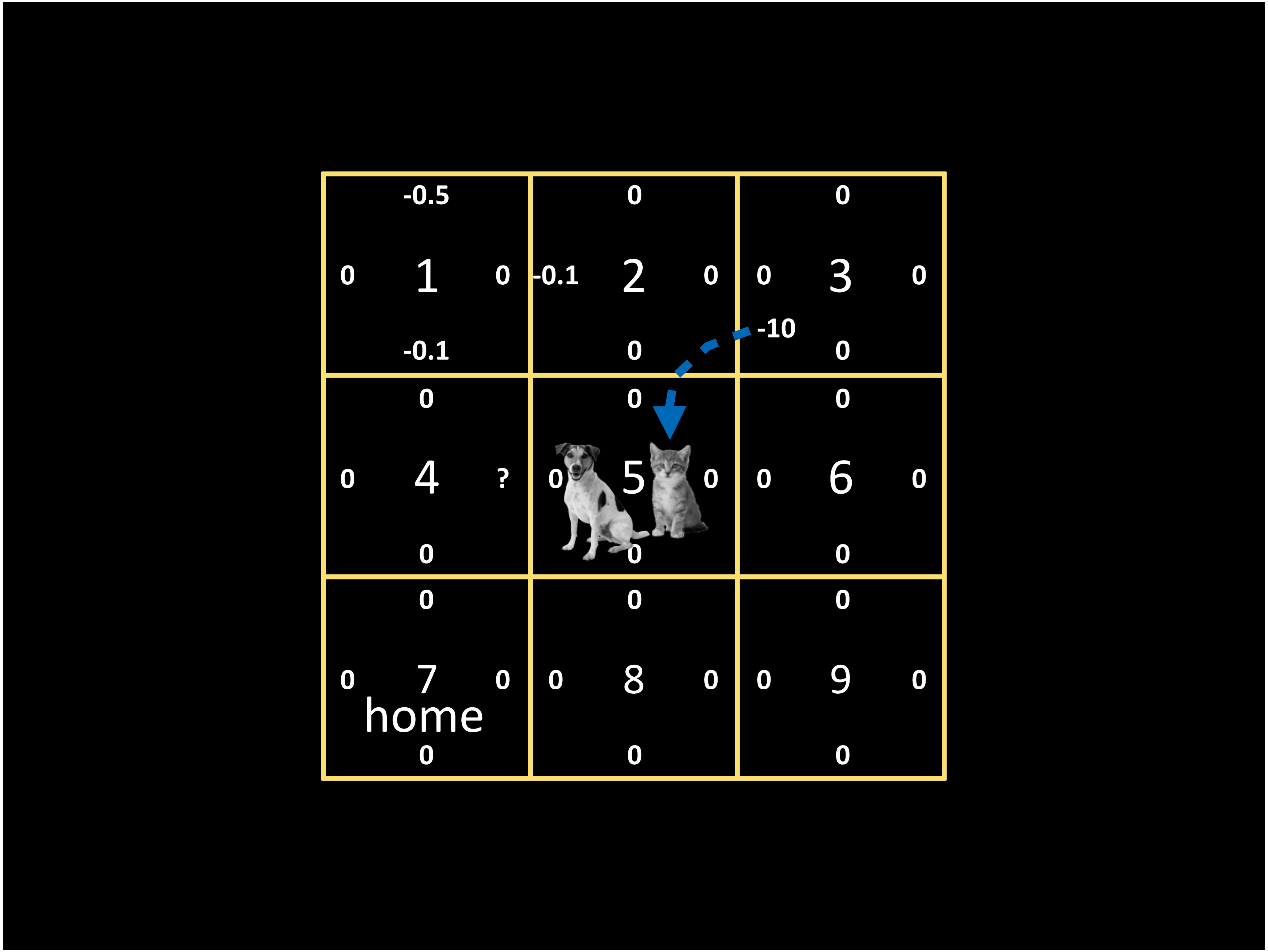

-0.5
0 1 0
0
0
-0.1 2 0
0
0
0 3 0
0
0
0 4 0
0
0
0 5 0
0
0
0 6 0
0
0
0 7 0
home
0
0
0 8 0
0
0
0 9 0
0

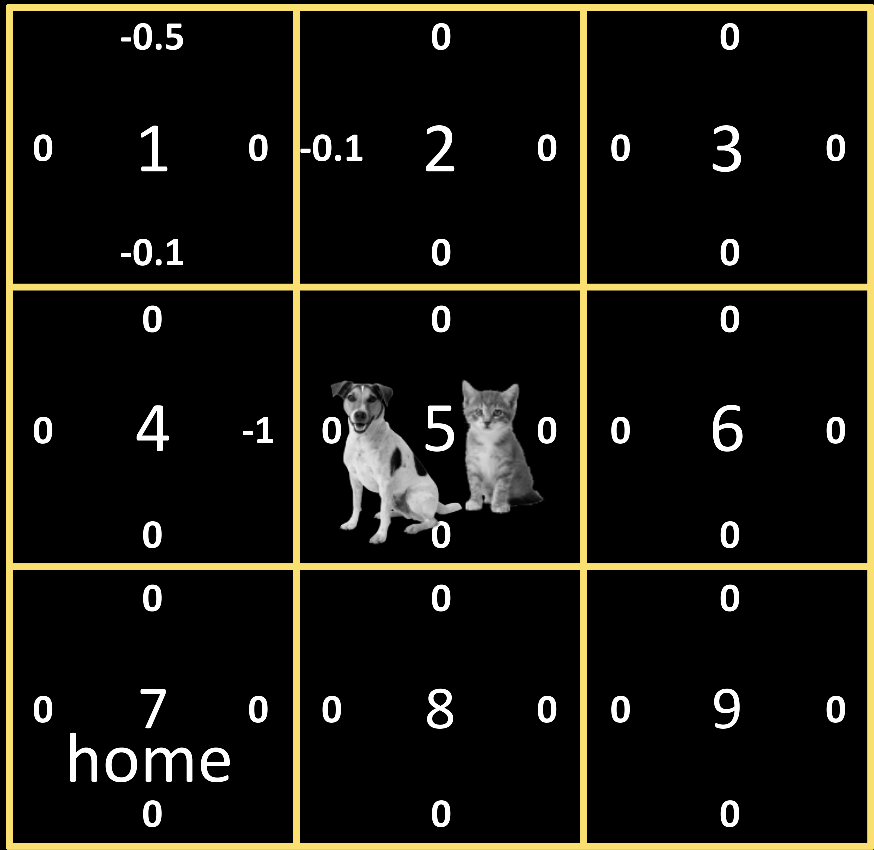

-0.5
0
0
0
1
0
-0.1
2
0
0
3
0
0
0
0
0
0
0
0
4
0
0
5
0
0
6
0
0
0
0
0
0
0
0
7
0
0
8
0
0
9
0
home
0
0
0

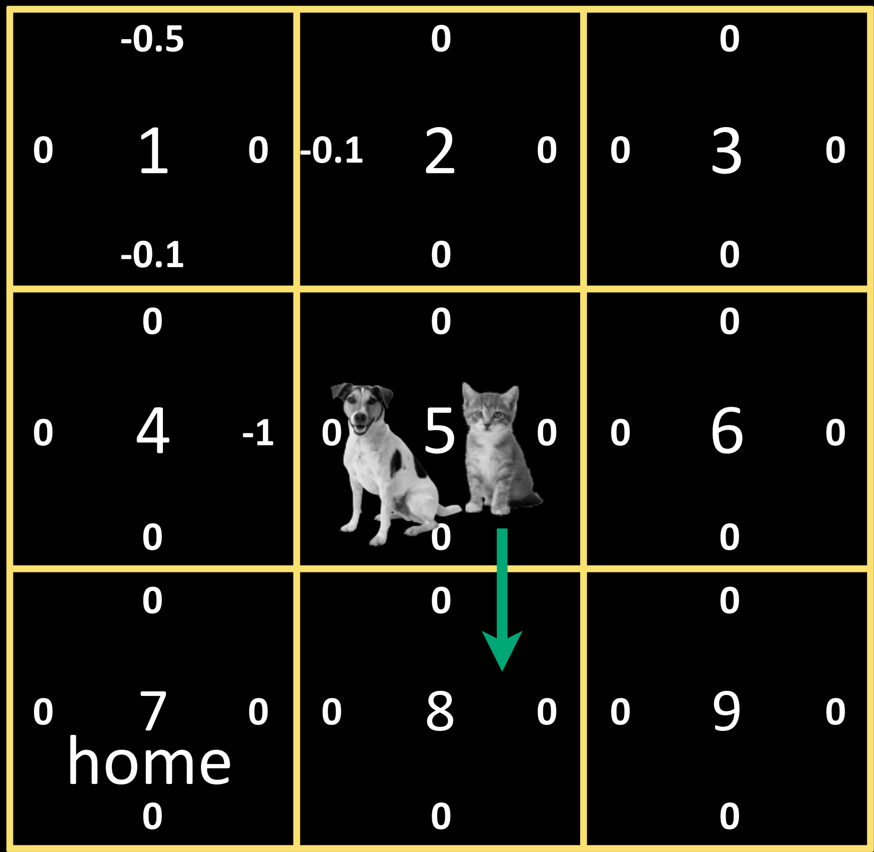

-0.5
0
1
0
?
-0.1
2
0
-1
0
0
3
0
0
0
0
4
0
0
0
0
5
0
0
0
0
6
0
0
0
0
7
0
home
0
0
0
8
0
0
0
0
9
0
0

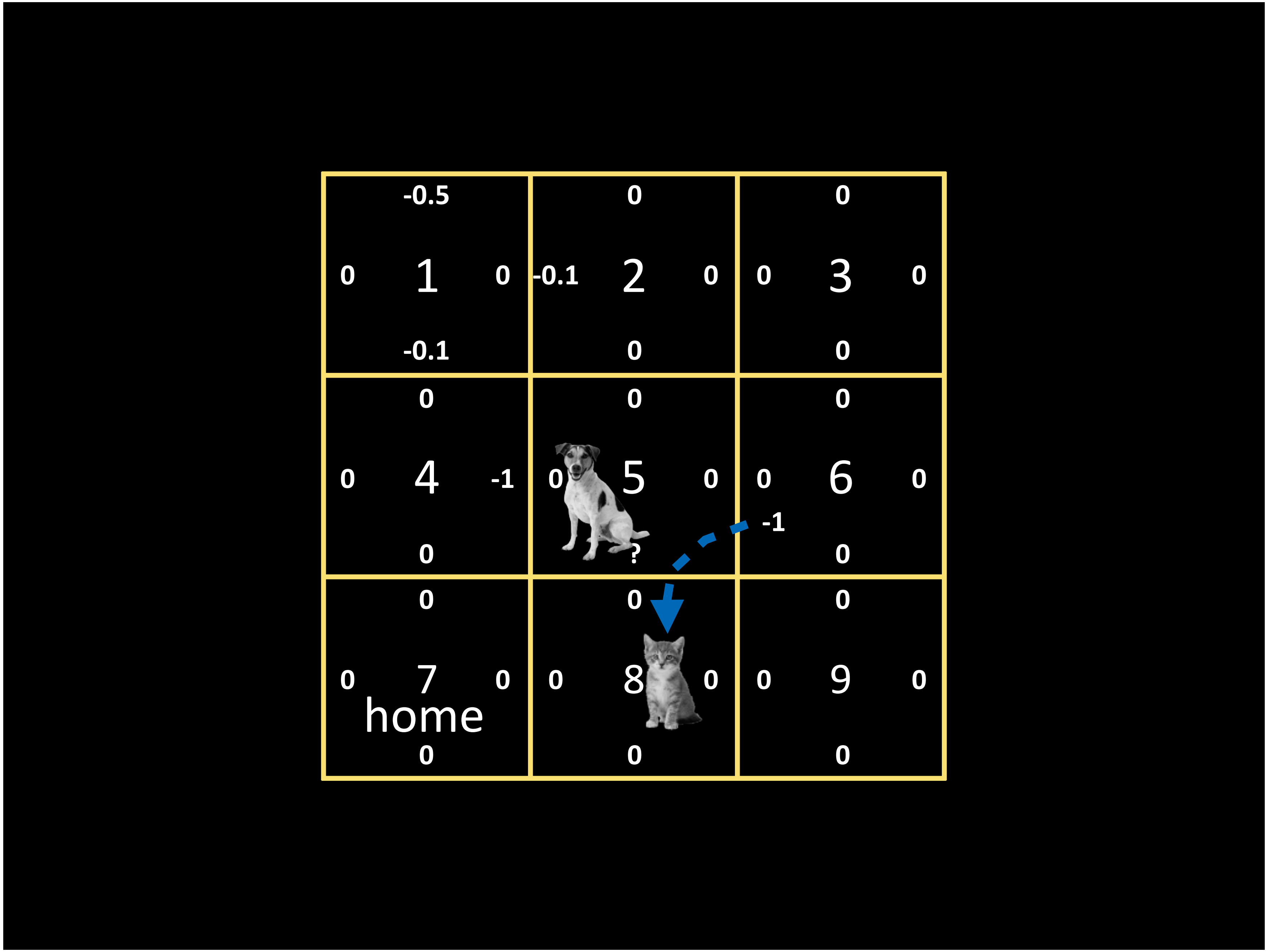

-0.5
0 1 0
-0.1
0
-0.1 2 0
0
0
0 3 0
0
0
0 4 0
0
0
0 5 0
0
0
0 6 0
0
0
0 7 0
home
0
0
0 8 0
0
0
0 9 0
0


-0.5
0
1
0
-0.1
-0.1
2
0
0
0
0
3
0
0
0
4
0
0
0
5
0
0
0
0
6
0
0
0
0
7
0
home
0
0
0
8
0
0
0
0
9
0
0

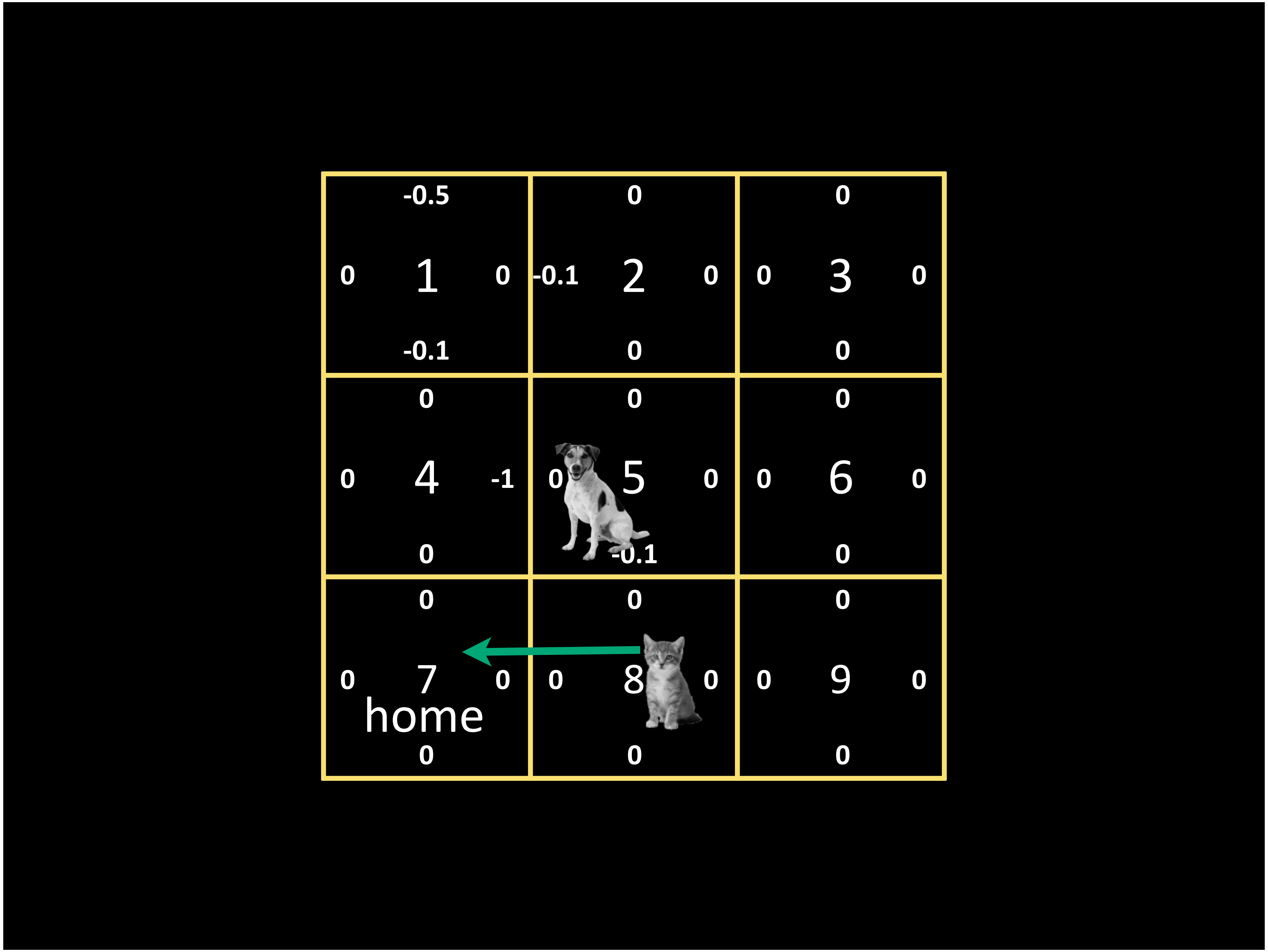

-0.5
0 1 0
-0.1
0
-0.1 2 0
0
0
0 3 0
0
-10
0
0 4 ?
0
0
0 5 0
0
0
0 6 0
0
0
0 7 0
home
0
0
0 8 0
0
0
0 9 0
0


-0.5
0 1 0
-0.1
0
-0.1 2 0
0
0
0 3 0
0
0
0 4 -1
0
0
0 5 0
0
0
0 6 0
0
0
0 7 0
home
0
0
0 8 0
0
0
0 9 0
0


-0.5
0 1 0
-0.1
0
-0.1 2 0
0
0
0 3 0
0
0
0 4 -1
0
0
0 5 0
0
0
0 6 0
0
0
0 7 0
home
0
0
0 8 0
0
0
0 9 0
0


-0.5
0
1
0
-0.1
-0.1
2
0
0
0
0
3
0
0
0
4
-1
0
0
0
5
0
?
0
6
0
-1
0
0
0
7
0
home
0
0
0
8
0
0
0
0
9
0
0


-0.5
0
0
0 1 0
-0.1 2 0
0 3 0
-0.1
0
0
0
0
0
0 4 -1
0 5 0
0 6 0
0
-0.1
0
0
0
0
0 7 0
home
0 8 0
0 9 0
0
0
0


-0.5
0 1 0
-0.1
0
-0.1 2 0
0
0
0 3 0
0
0
0 4 -1
0
0
0 5 0
-0.1
0
0 6 0
0
0
0 7 0
home
0
0
0 8 0
0
0
0 9 0
0

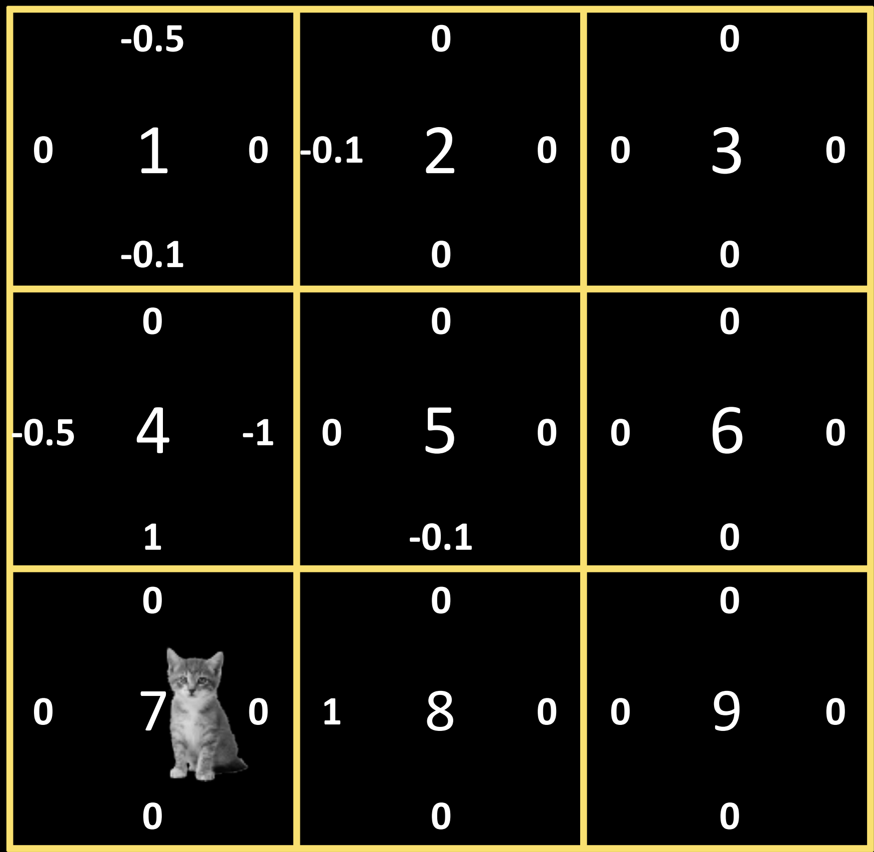

-0.5
0
0
0
1
0
-0.1
2
0
0
3
0
-0.1
0
0
0
0
0
0
4
-1
0
5
0
0
6
0
0
-0.1
0
0
0
0
-10
0
7
0
?
8
0
0
9
0
home
0
0
0

episode 1 ends.

let's work out the next episode, starting at state 4

go WEST and then SOUTH

how does the table change?

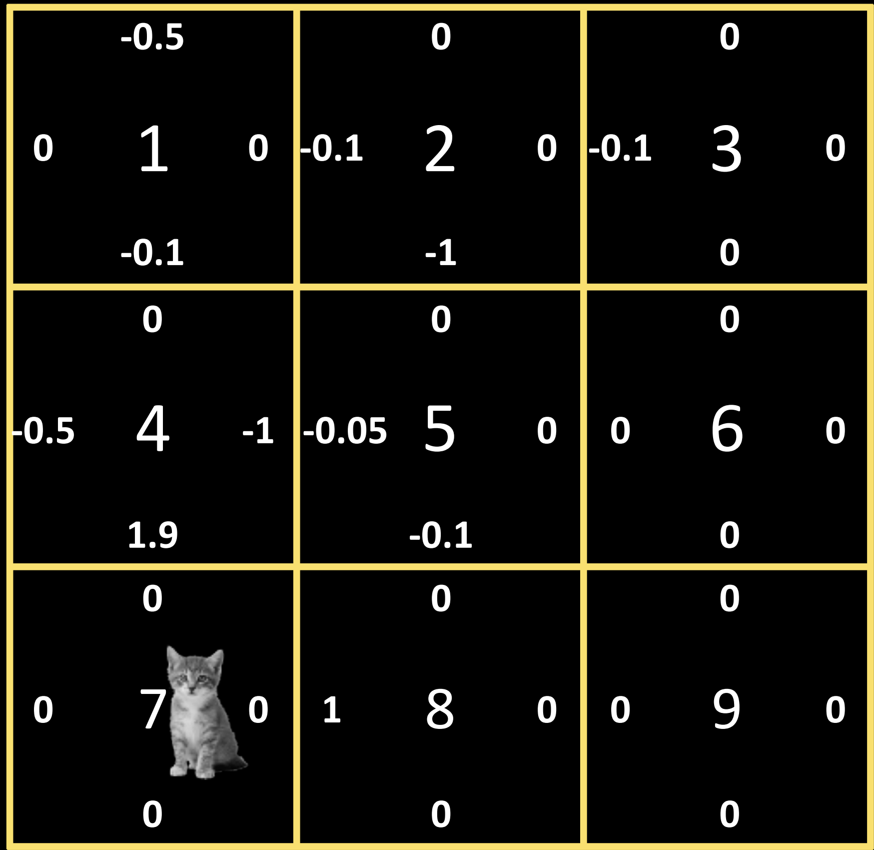


-0.5
0 1 0
-0.1
0
-0.1 2 0
0
0
0 3 0
0
0
-0.5 4 -1
1
0
0 5 0
-0.1
0
0 6 0
0
0
0 7 0
0
0
1 8 0
0
0
0 9 0
0

and the next episode, starting at state 3

go WEST -> SOUTH -> WEST -> SOUTH

how does the table change?


-0.5
0 1 0
-0.1
0
-0.1 2 0
-1
0
-0.1 3 0
0
0
-0.5 4 -1
1.9
0
-0.05 5 0
-0.1
0
0 6 0
0
0
0 7 0
0
0
1 8 0
0
0
0 9 0
0

what we just saw was some episodes of

Q-learning

value updates based on **optimal policy**:
value of **best next action**

off-policy learning

SARSA-learning?

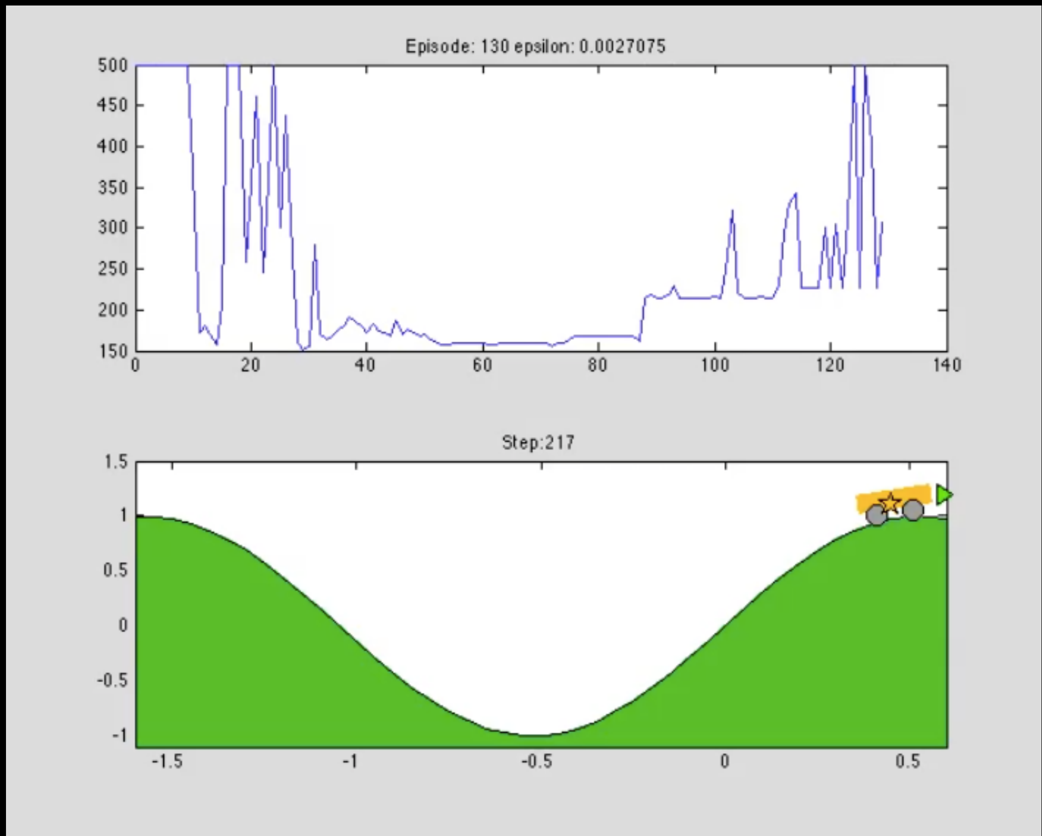
value updates based on **used policy**:

value of **the actual next action**

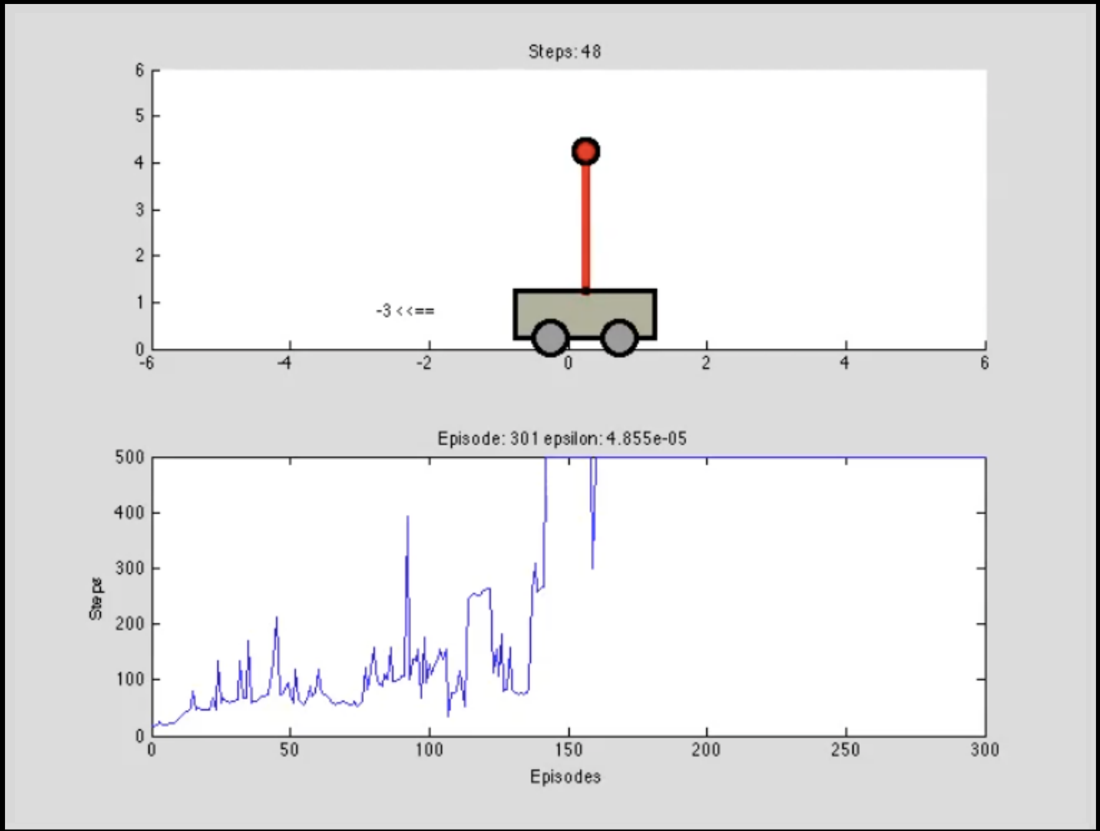
on-policy learning


Q
SARSA

mountain car...



pole balancing...



Pole balancing in reality: http://www.youtube.com/watch?v=Lt-KLtkDlh8

matlab code for you to play with...

available online for the curious (extremely easy to run):
http://jamh-web.appspot.com/download.htm#Reinforcement_Learning:

please do e-mail for questions, and if you want to work on reinforcement learning research projects:
arjun@studix.com / chandra@ifi.uio.no

coyote learning what not to do...