Reinforcement Learning

Chris Watkins

How animals learn?

In repeated trials, animal faces two alternatives.

Animal tries each choice repeatedly.

Animal increases probability of choosing the course of action that is better rewarded.

The problem of delayed rewards

Previous theory incoherent.

How does the animal know when the ‘trial’ begins and ends?

When the rat is half-way to the cup, it has not yet been rewarded.

Why doesn’t it learn at this point that there is no reward for going half-way to the cup? Then the rat would not start.

Typically, getting a good reward may require some unpleasant effort first. Why doesn’t the animal learn **not** to make the effort?

States, actions, rewards

At each time step:

- agent senses world-state s
- agent chooses action a
- world changes to state s’
- agent receives reward r a number

Markov property: the reward, and the effects of actions, depend *only* on current state, and not on previous history

Agent’s experience consists of sequence of <s,a,s’,r>

Simplify! Assume a finite number of states, finite number of actions.

At each time-step *t,* agent seeks to choose actions to maximise discounted sum of rewards:

$$r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$$

Bellman equations in Q

Optimal actions implicitly defined by
optimal value function V

$$V(s) = \max_a \mathrm{E}[r + \gamma V(s') \mid s, a]$$

Introduce Q :

$$Q(s,a) = \mathrm{E}[r + \gamma V(s') \mid s, a]$$

Note that: $$V(s) = \max_a Q(s,a)$$

Substitute Q for V:

$$Q(s,a) = \mathrm{E}[r + \gamma \max_b Q(s',b) \mid s, a]$$

Optimal Q : Bellman’s equations, expressed with Q

If the Markov property holds – that is, if the effect of action *a* depends *only* on the current state *x* – then if for **all** *x* and *a,* it is true that:

$$Q(x,a) = E[r(x,a) + \gamma \max_b Q(x',b)]$$

then *Q* is optimal, and the policy of taking that action that maximizes Q in each state yields the highest possible expected discounted payoff.

The optimal Q function is unique, and is the best possible in all states.

How can we find this optimal Q?

Q learning

Agent has a table of Q values, one for each state-action pair

Set Q(s,a) to arbitrary initial value for each state *s* and action *a*

For each episode of experience $\langle s, a, s', r \rangle$

update the value of *Q(s,a)*

$$Q(s,a) \leftarrow r + \max_b Q(s',b)$$

Key points

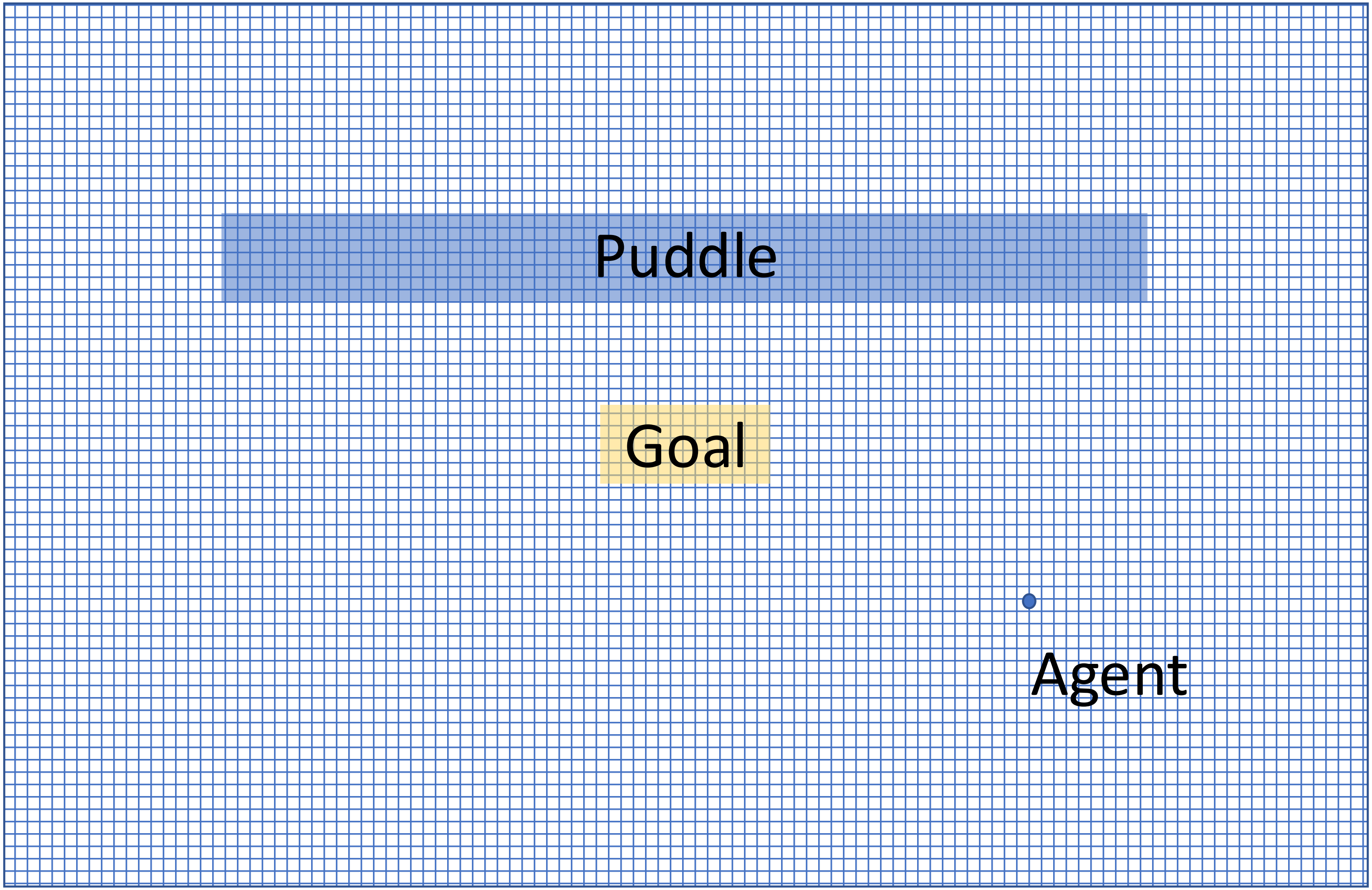
A Q-learning agent:

- maintains and updates an estimate of *Q(x,a)* for all states *x* and actions *a*
- does not need to be able to predict the effects of its actions
- remembers only the previous state and action for one time step
- can learn a policy that maximises rewards over a time-scale much longer than its memory
- an internal predictive world-model can speed up learning by providing additional simulated experience

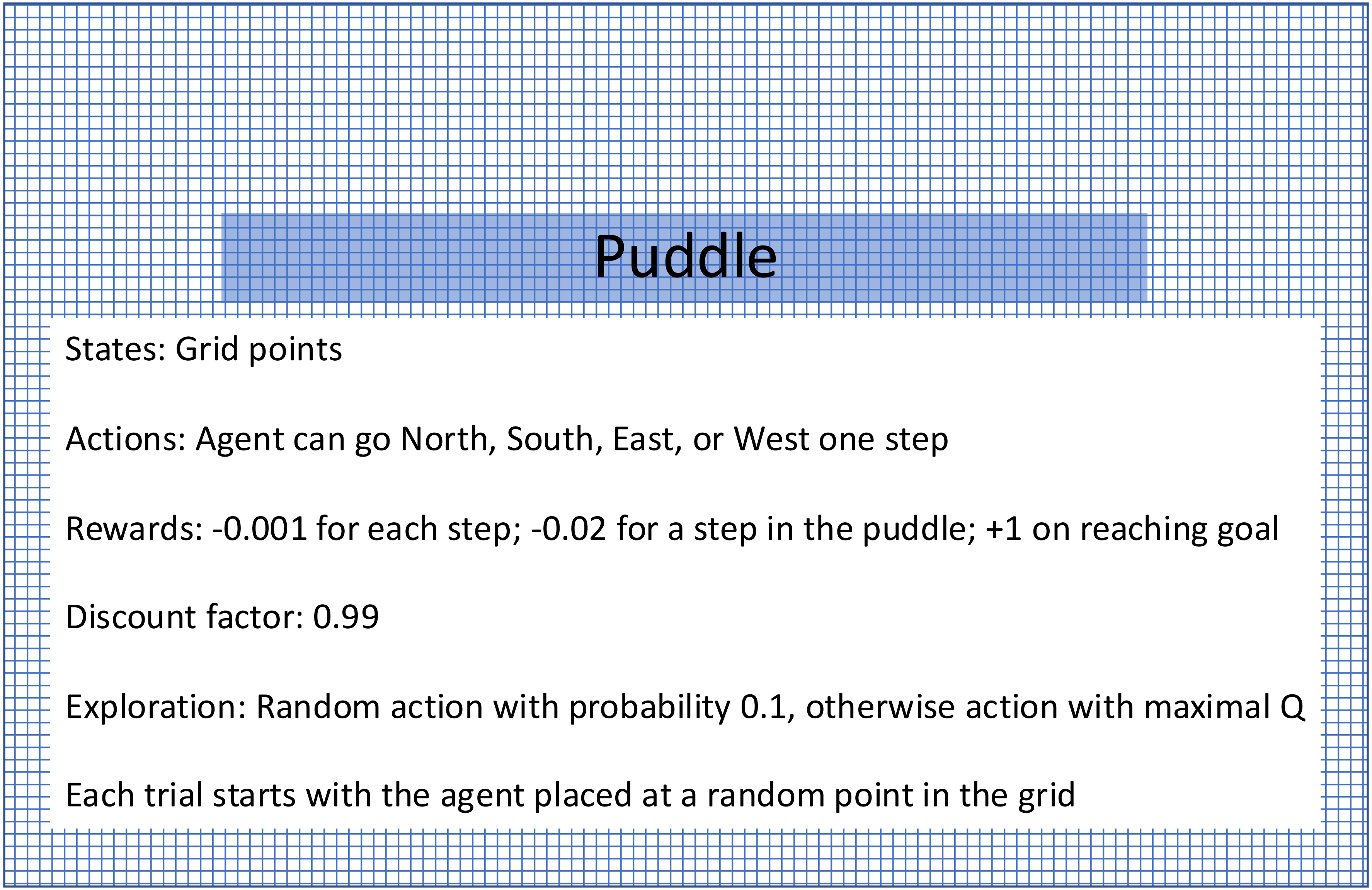
Q-learning seems to be ‘entry level learning’ for simple organisms ?

A natural ‘upgrade path’ by adding world-model, planning ... ?

Q learner in the ‘Puddle world’



Q learner in the ‘Puddle world’



States: Grid points

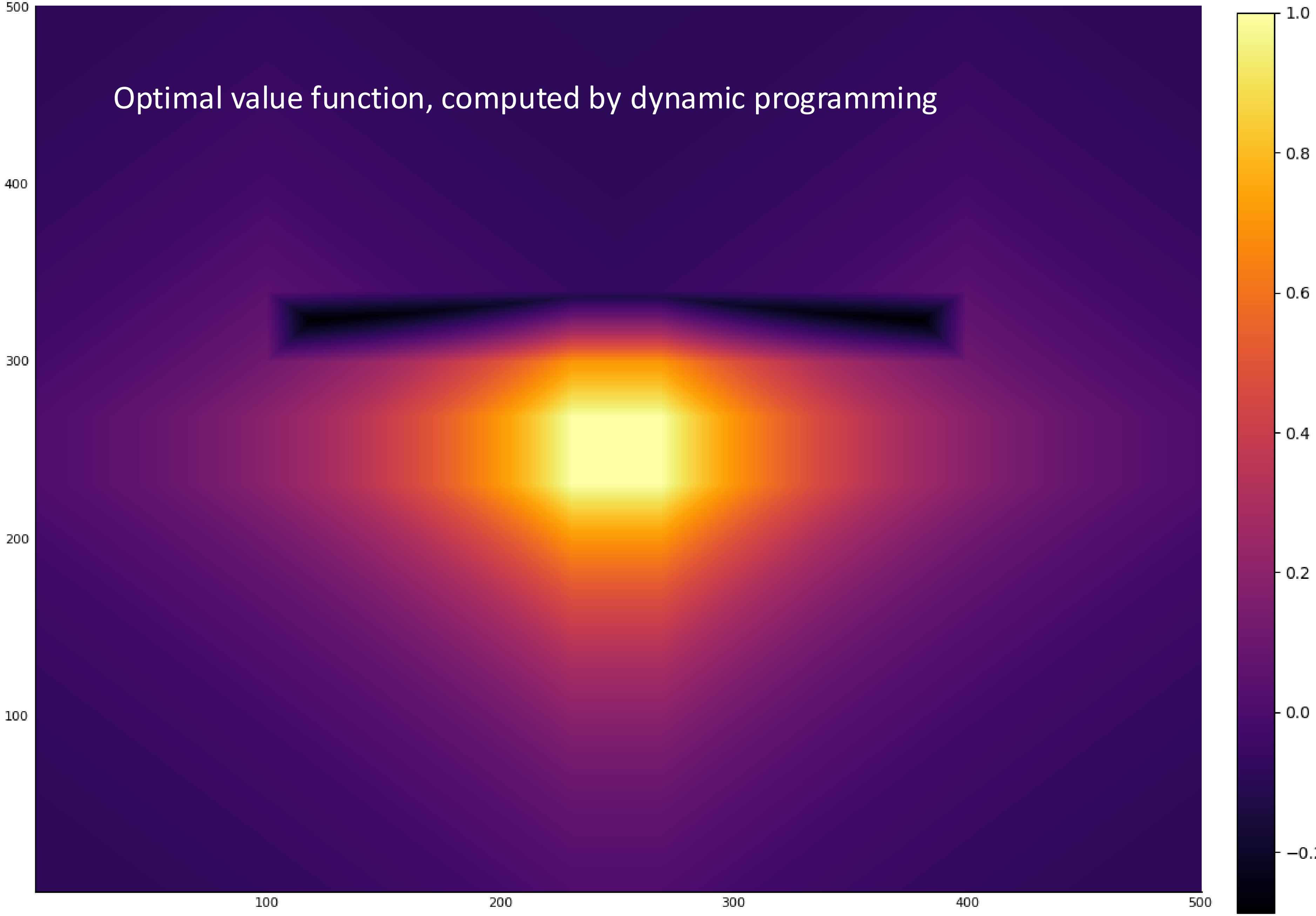
Actions: Agent can go North, South, East, or West one step

Rewards: -0.001 for each step; -0.02 for a step in the puddle; +1 on reaching goal

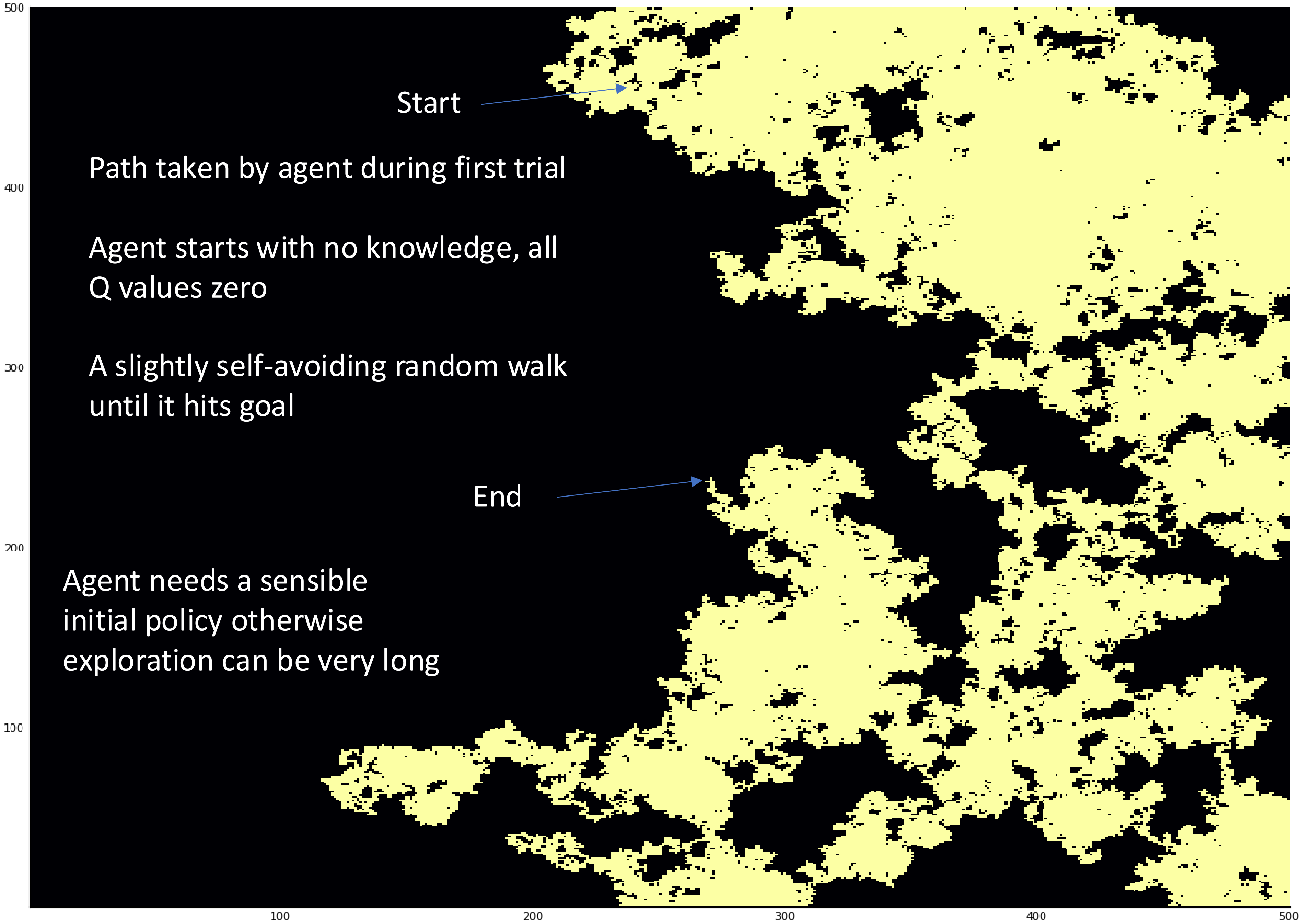
Discount factor: 0.99

Exploration: Random action with probability 0.1, otherwise action with maximal Q

Each trial starts with the agent placed at a random point in the grid



Optimal value function, computed by dynamic programming



Start

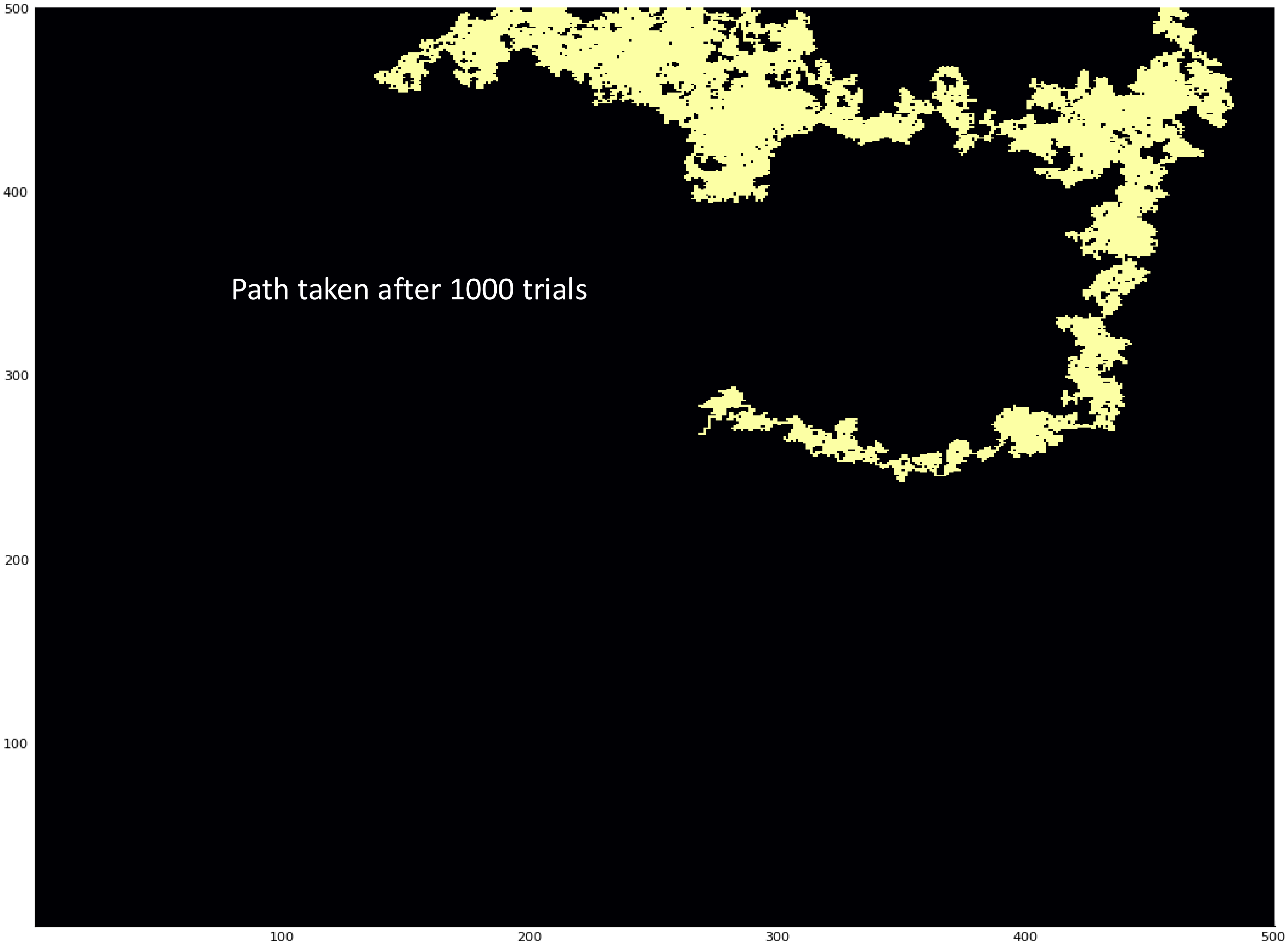
Path taken by agent during first trial

Agent starts with no knowledge, all Q values zero

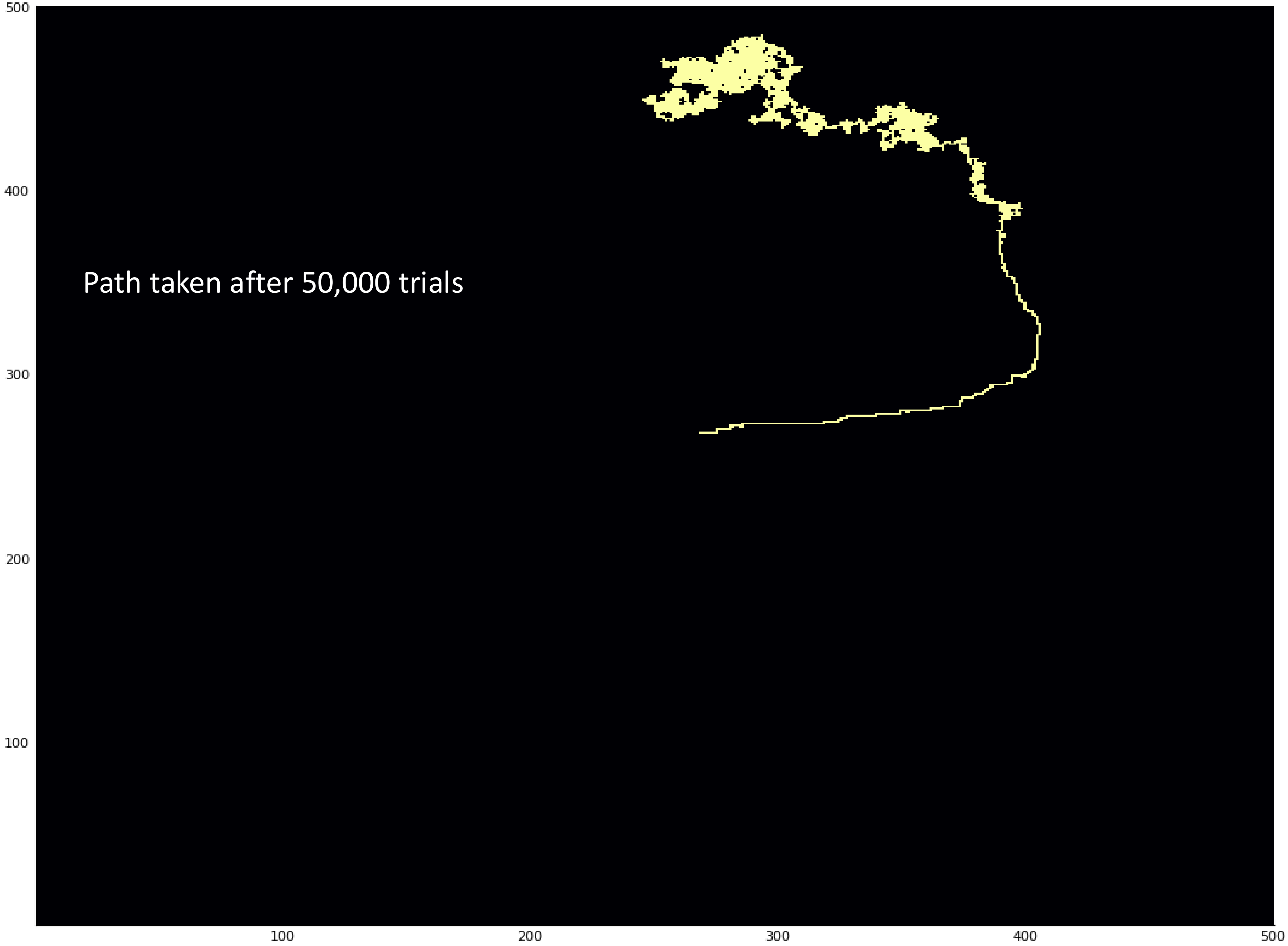
A slightly self-avoiding random walk until it hits goal

End

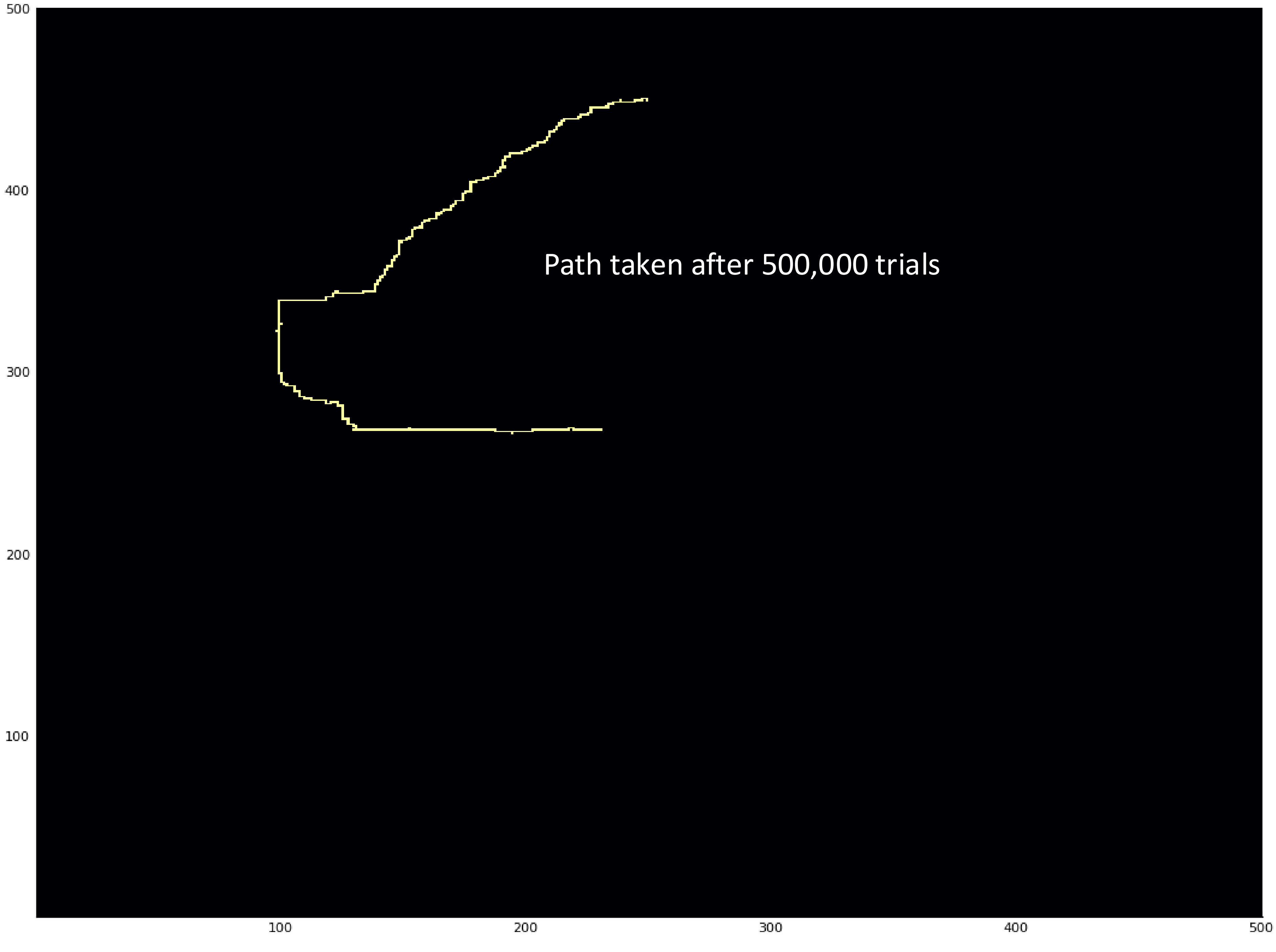
Agent needs a sensible initial policy otherwise exploration can be very long



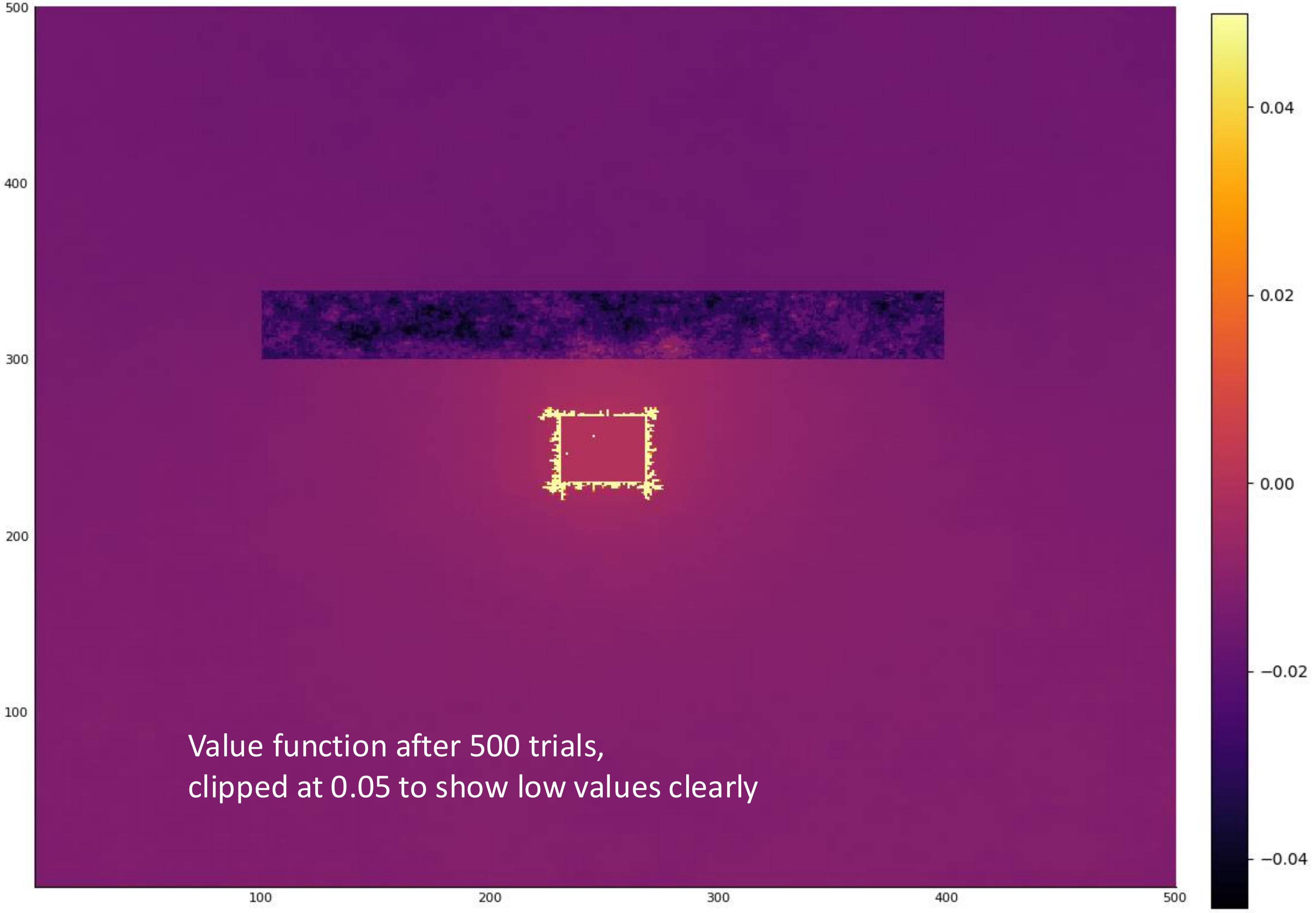
Path taken after 1000 trials



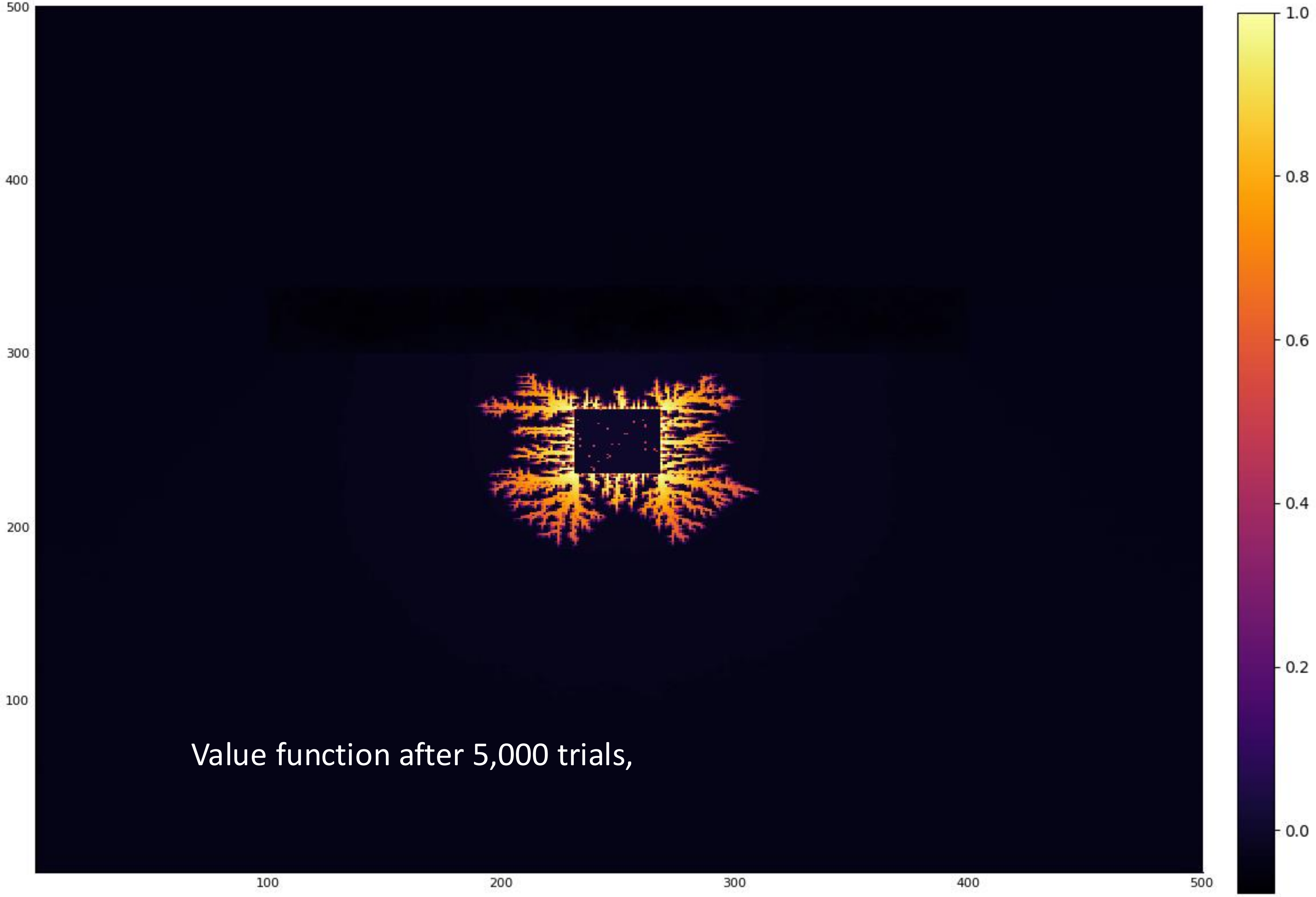
Path taken after 50,000 trials



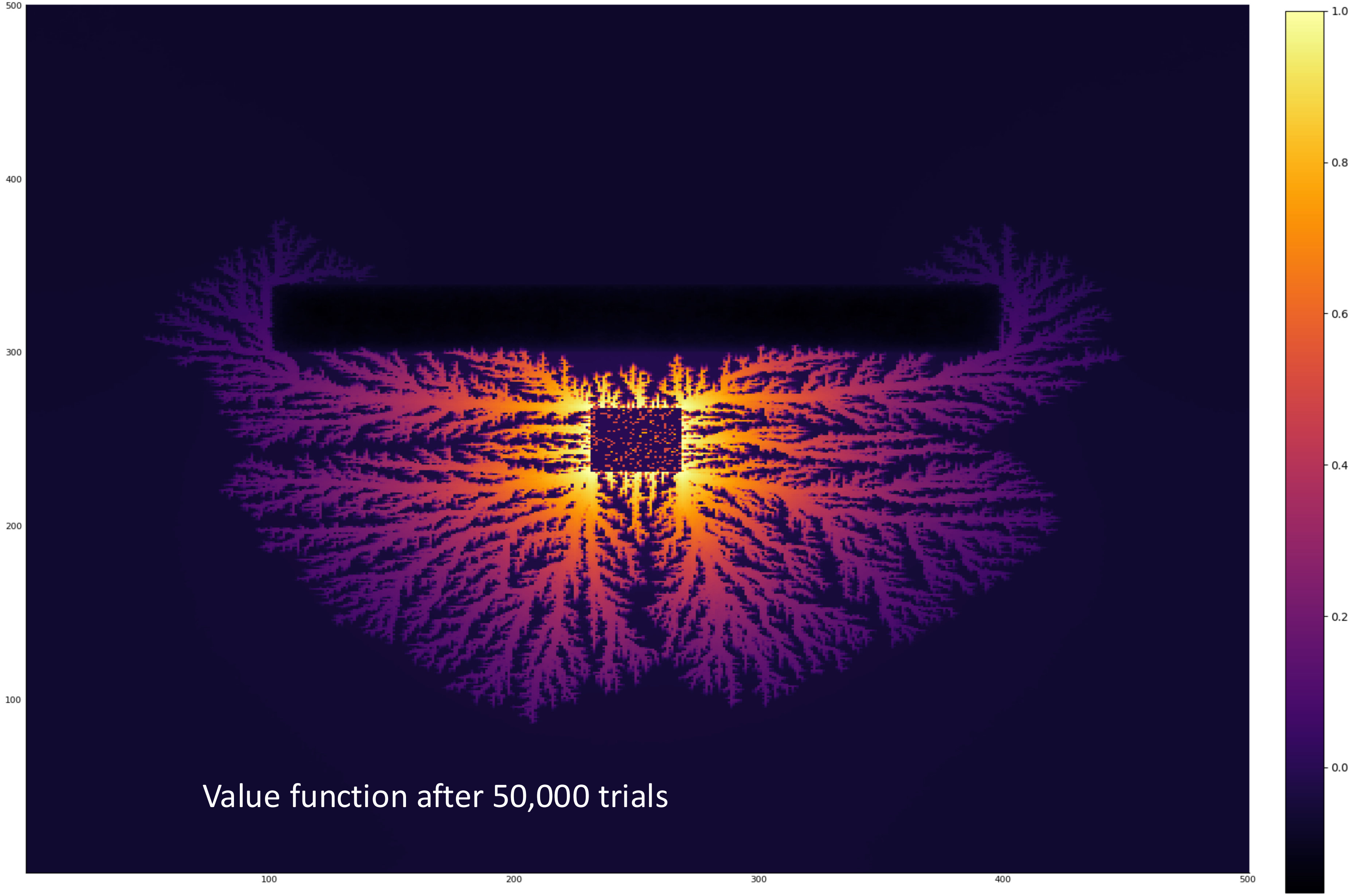
Path taken after 500,000 trials



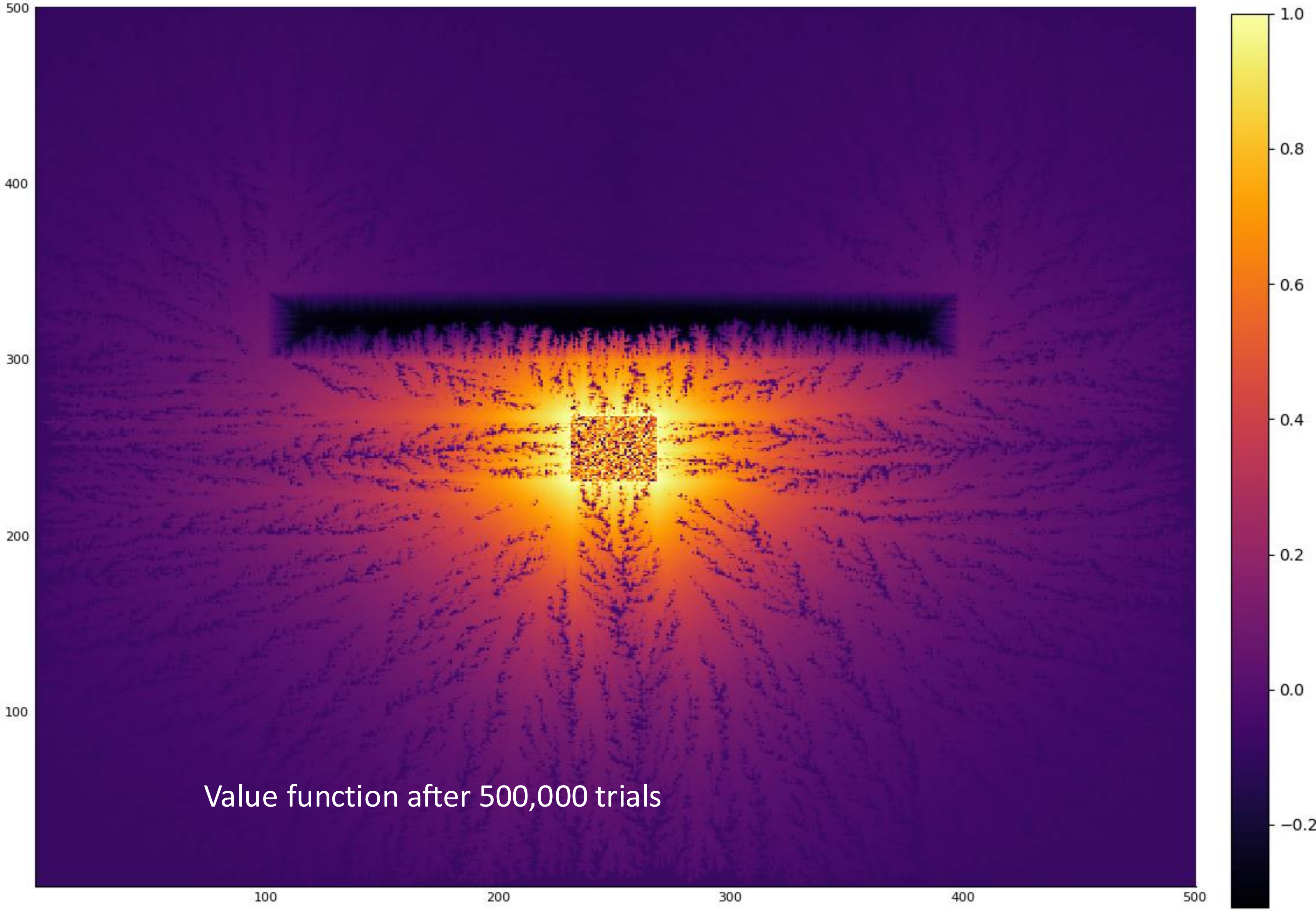
Value function after 500 trials,
clipped at 0.05 to show low values clearly



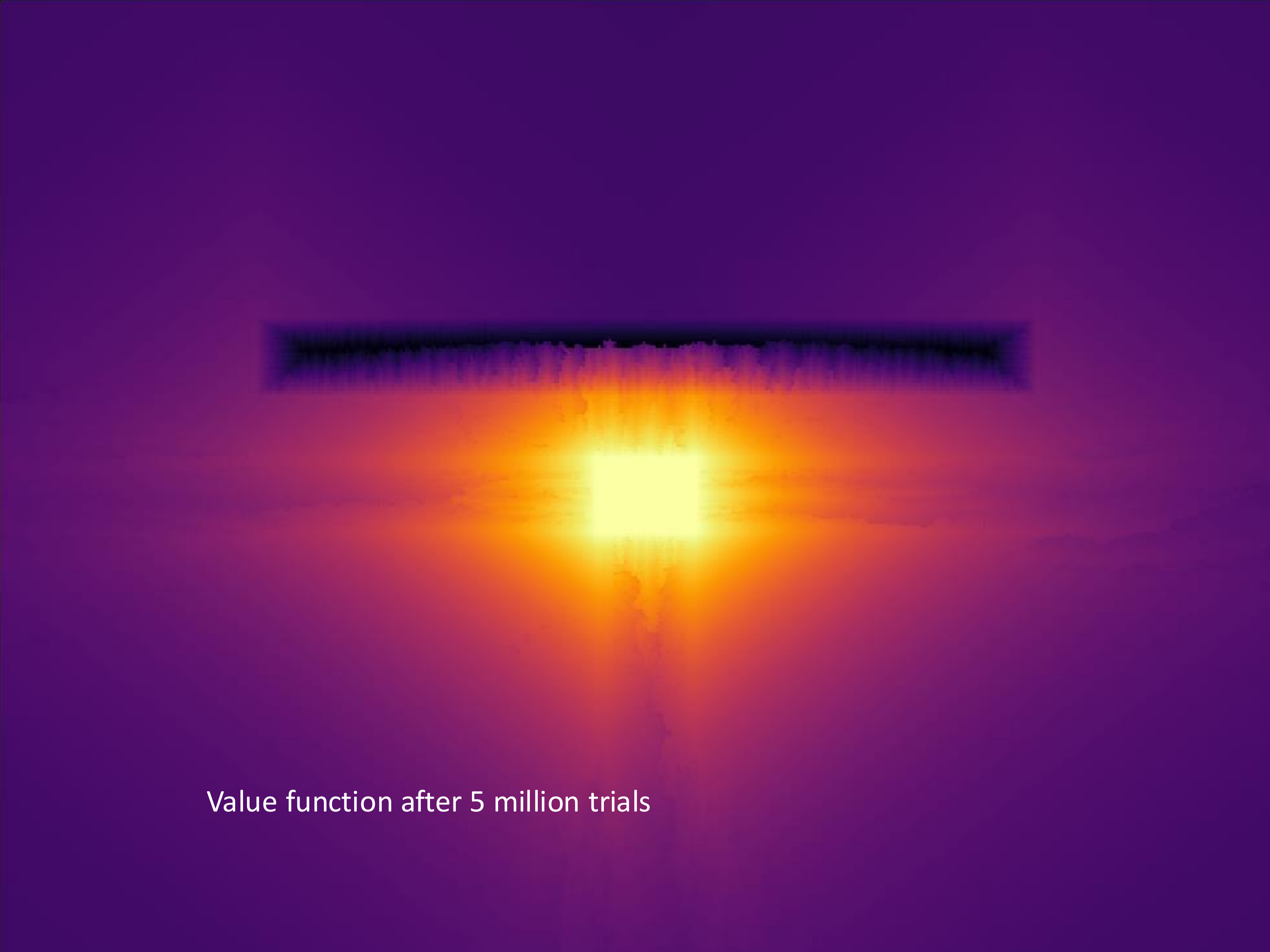
Value function after 5,000 trials,



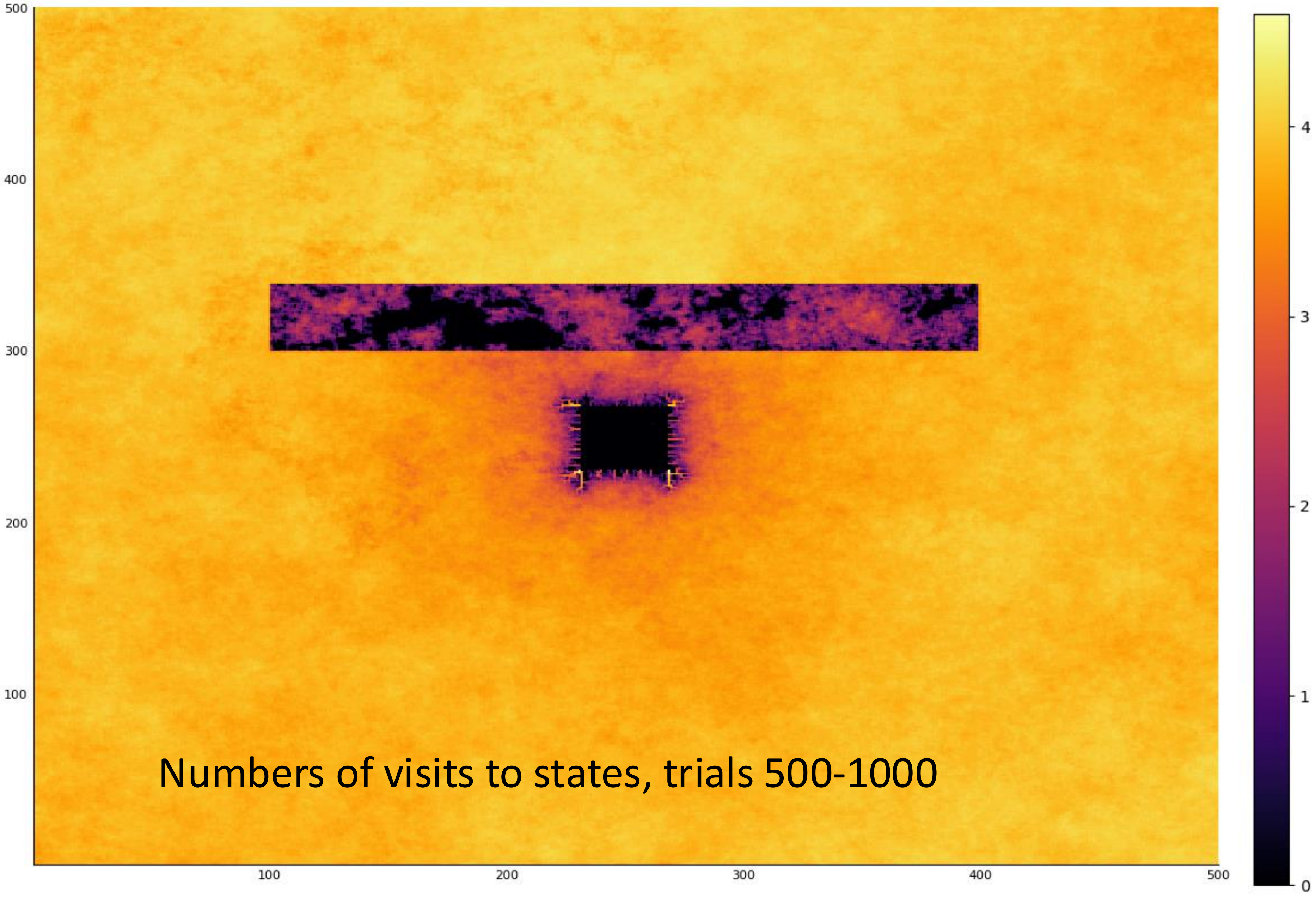
Value function after 50,000 trials



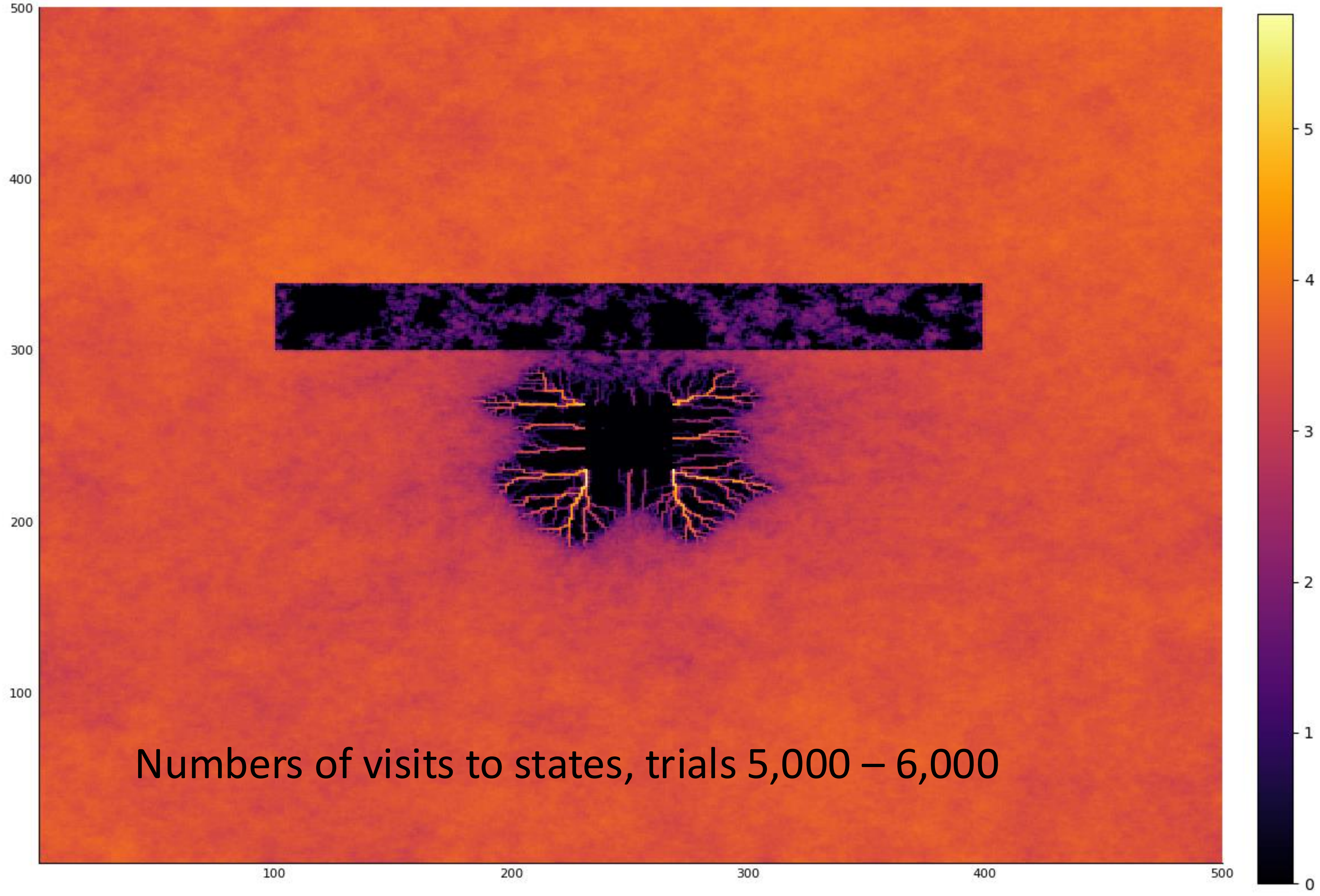
Value function after 500,000 trials



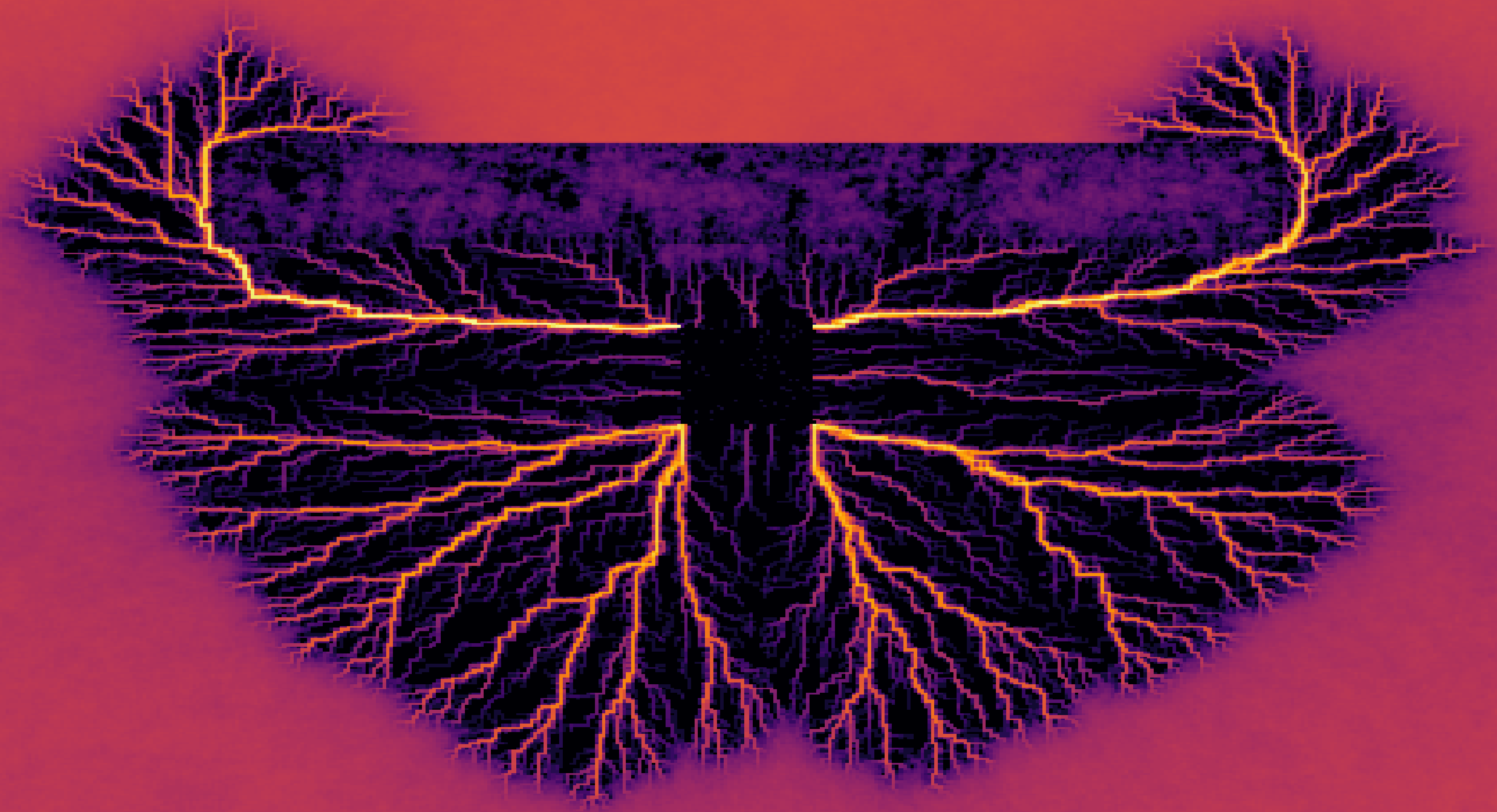
Value function after 5 million trials



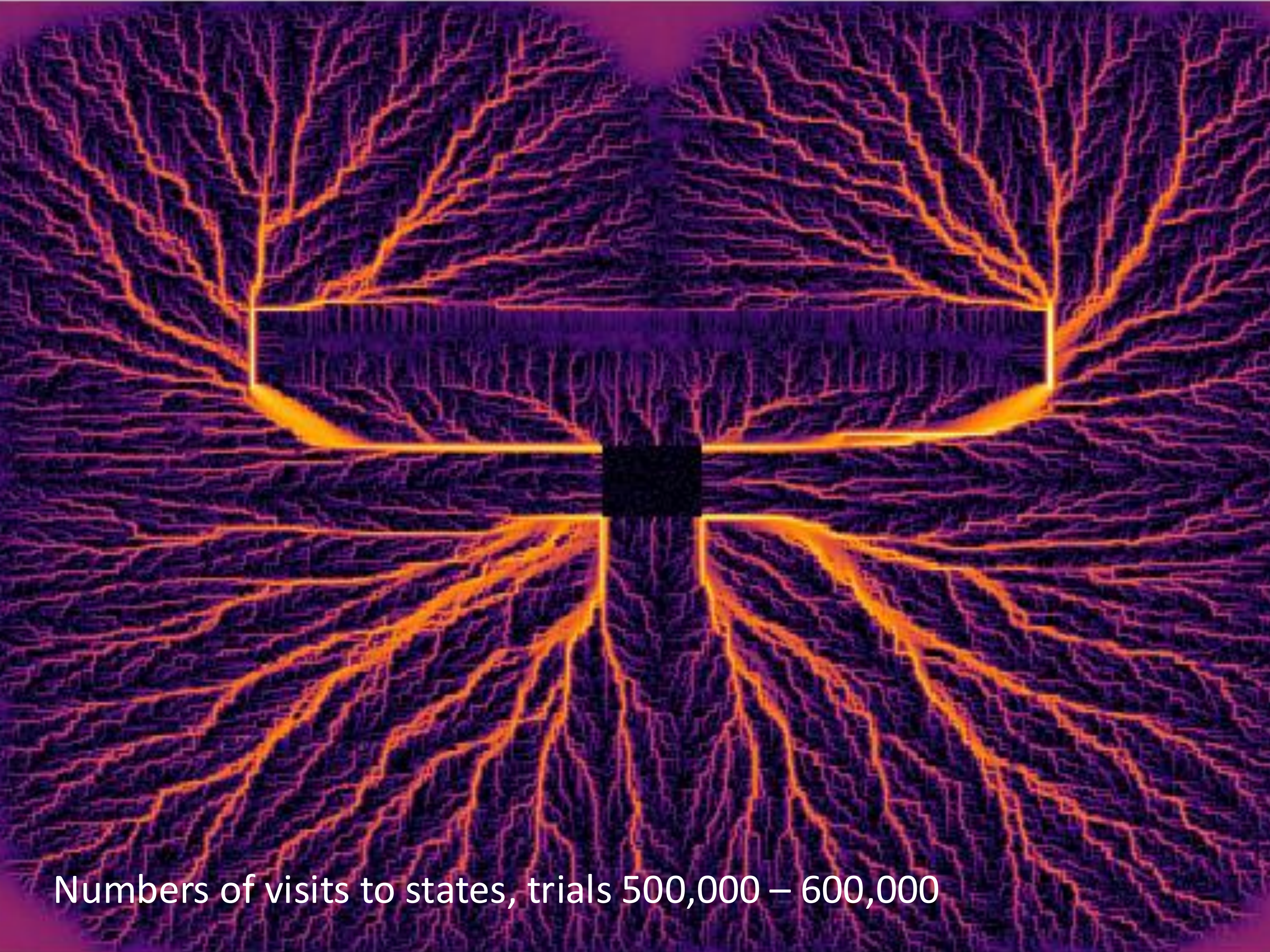
Numbers of visits to states, trials 500-1000



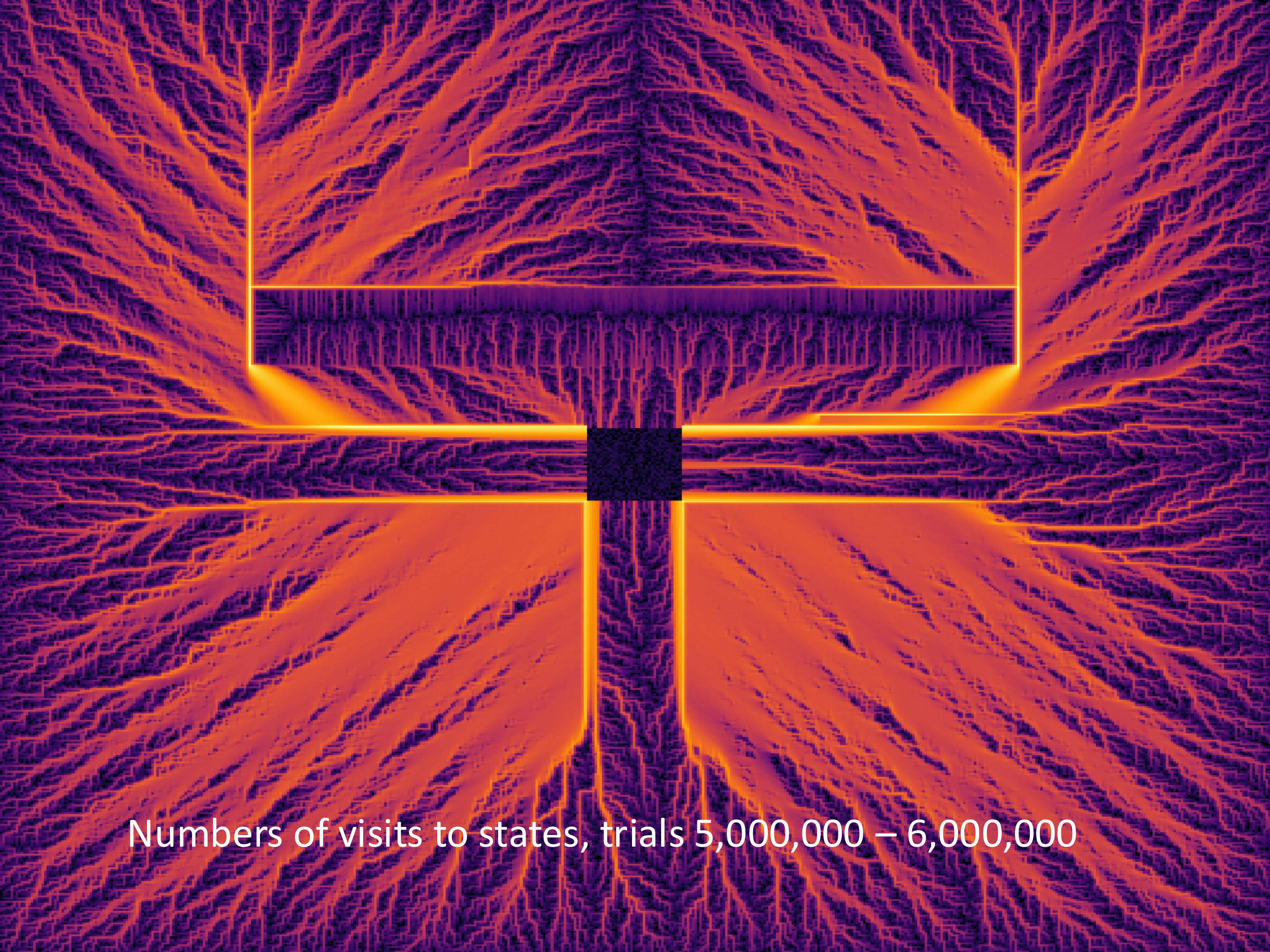
Numbers of visits to states, trials 5,000 – 6,000



Numbers of visits to states, trials 50,000 – 60,000



Numbers of visits to states, trials 500,000 – 600,000



Numbers of visits to states, trials 5,000,000 – 6,000,000

Remark 1

The grid-world or ‘puddle-world’ is not a realistic model of navigation in space.

Think of it instead as an abstract state space in which the agent can identify which state it is in, and the agent has several possible ‘moves’ to other states, any of which it can choose.

The gridworld is convenient because we can visualize what is going on: the values, the visit counts, and the paths.

The agent has a goal that it can reach after many preparatory steps, each of which has a small cost.

How can the agent learn to obtain the delayed reward?

Remarks 2

A Q-learning agent develops ‘path habits’ in state space

Early chance explorations and successes may have long-term influence on paths taken

Not all of the state space may be explored
- but this is part of the point of RL!

Exploration needs to be structured for the problem.

Initial Q values can be important:
- if too low, agent is pessimistic and not exploratory
- if too high, agent is pathologically optimistic