# The explore exploit dilemma

Computational Cognitive Science 2014

Dan Navarro

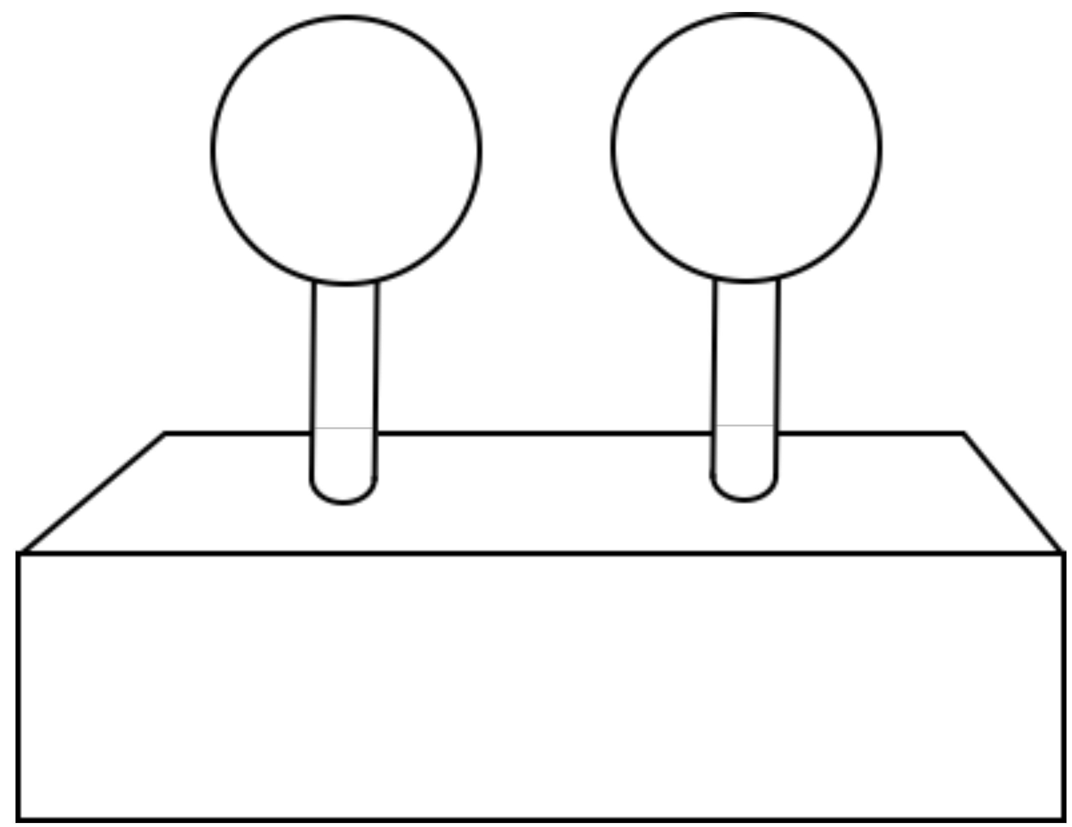
# The Turker's dilemma
# (a.k.a. observe or bet)



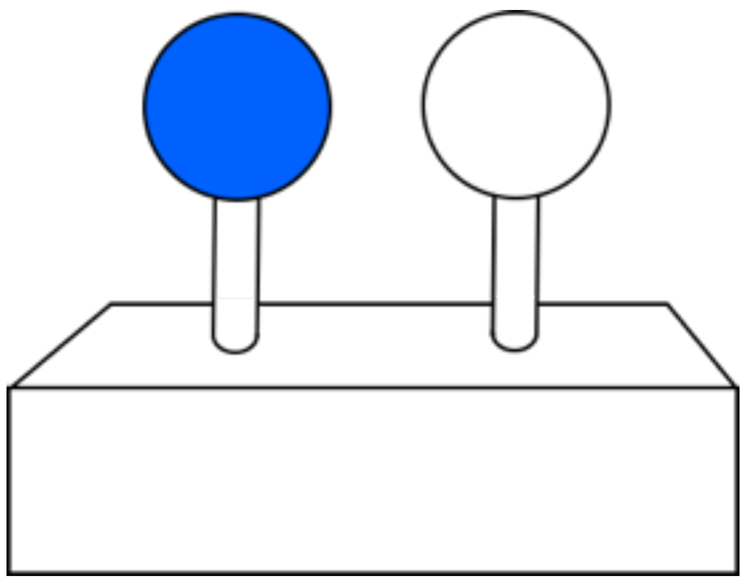
**Do the HIT:** Tag some images, and eventually get a reward when the requester pays. <u>If</u> they pay.

**Do your research:** Check out Turkopticon (etc.), read the reviews for the requester. Maybe check out Turk and see if there are any better jobs on offer?

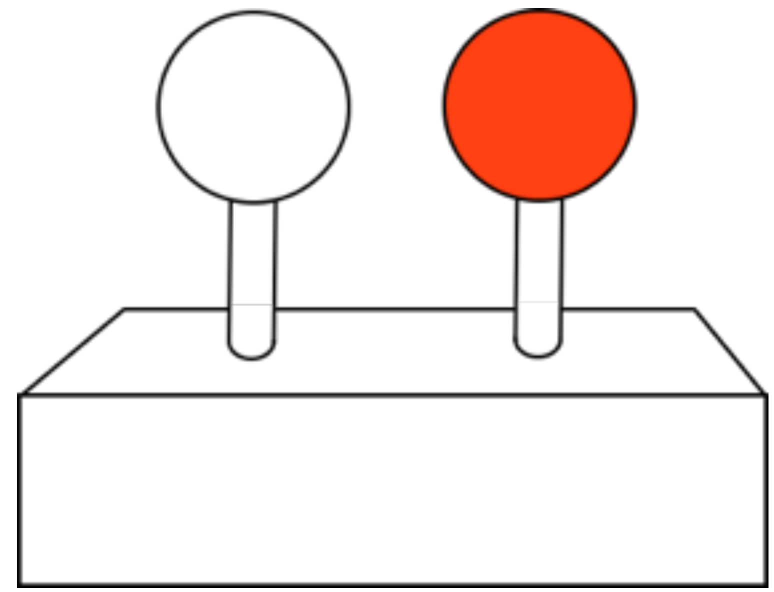
# The observe or bet task



This is a “blox” machine

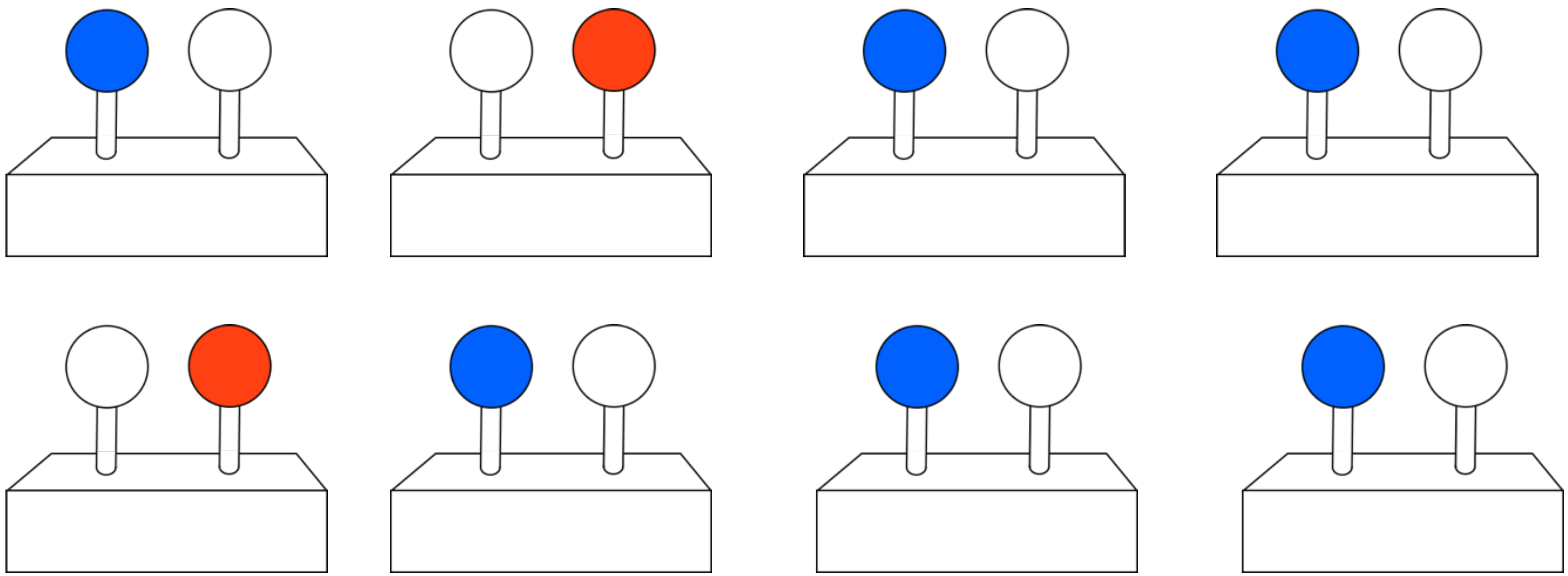


It has a blue light



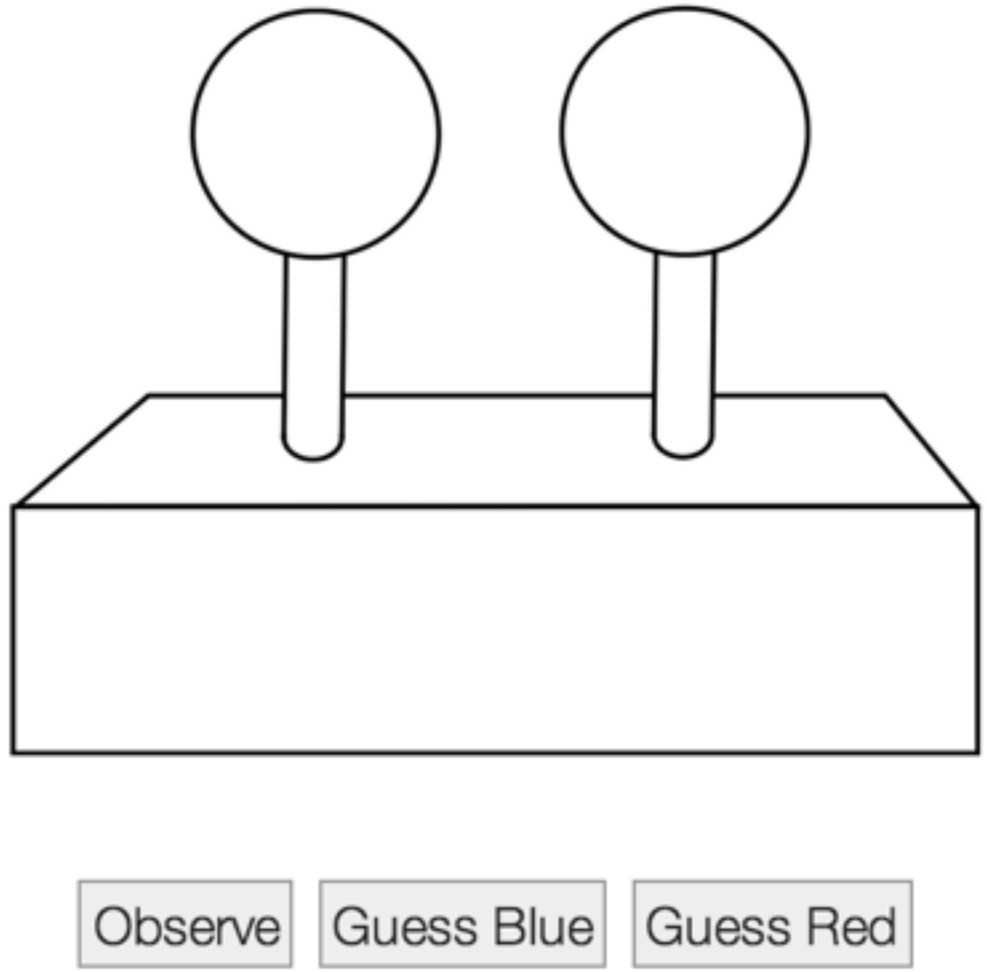
and a red light

These lights flash intermittently.



One light tends to come on
more often than the other.

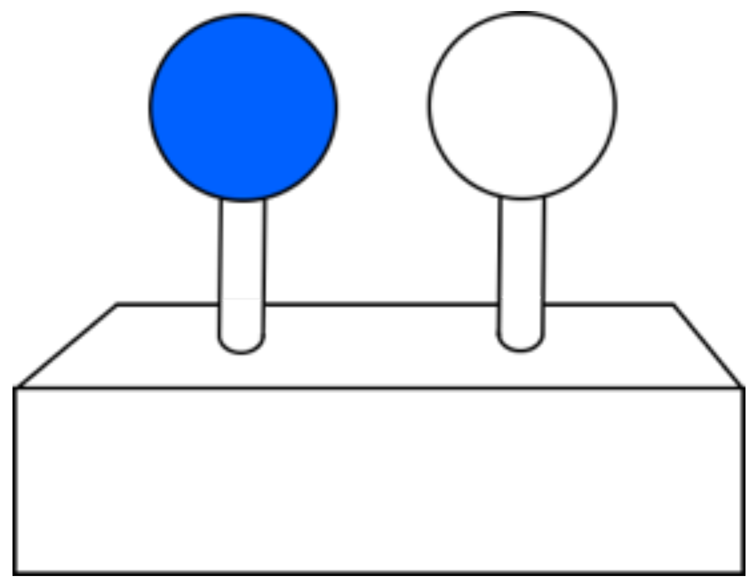
You don't know which one





At every point in time, you can make an observation or bet on which outcome will occur…

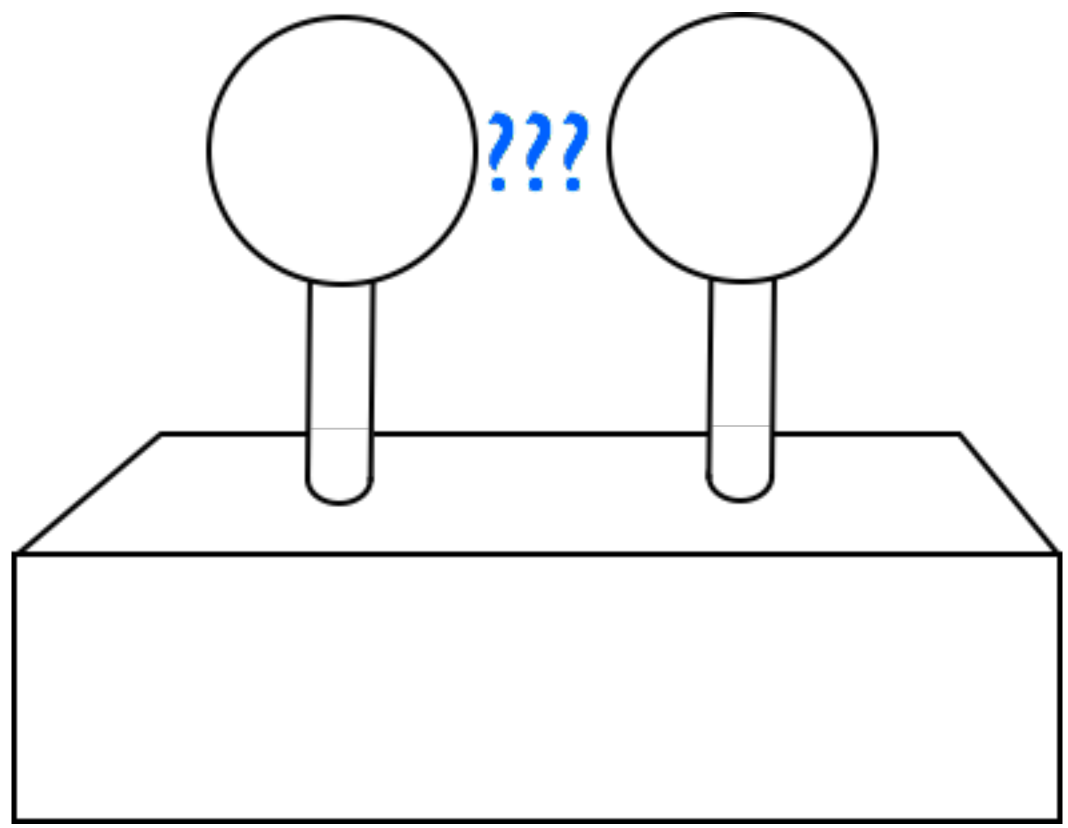
If you **OBSERVE**, you get to see what colour light turns on



“Observing” is like doing your research. You learn something about the state of the world, but receive no rewards.

If you **GUESS BLUE**, you will get a reward (+1) if you're correct, a loss if you're wrong (-1).

You don't find out whether you were right or wrong until later.




"Betting" is like the Turker committing to a HIT. You spend the time on it, and you should get a reward if you've chosen well. But you don't find out whether you've done well until later.

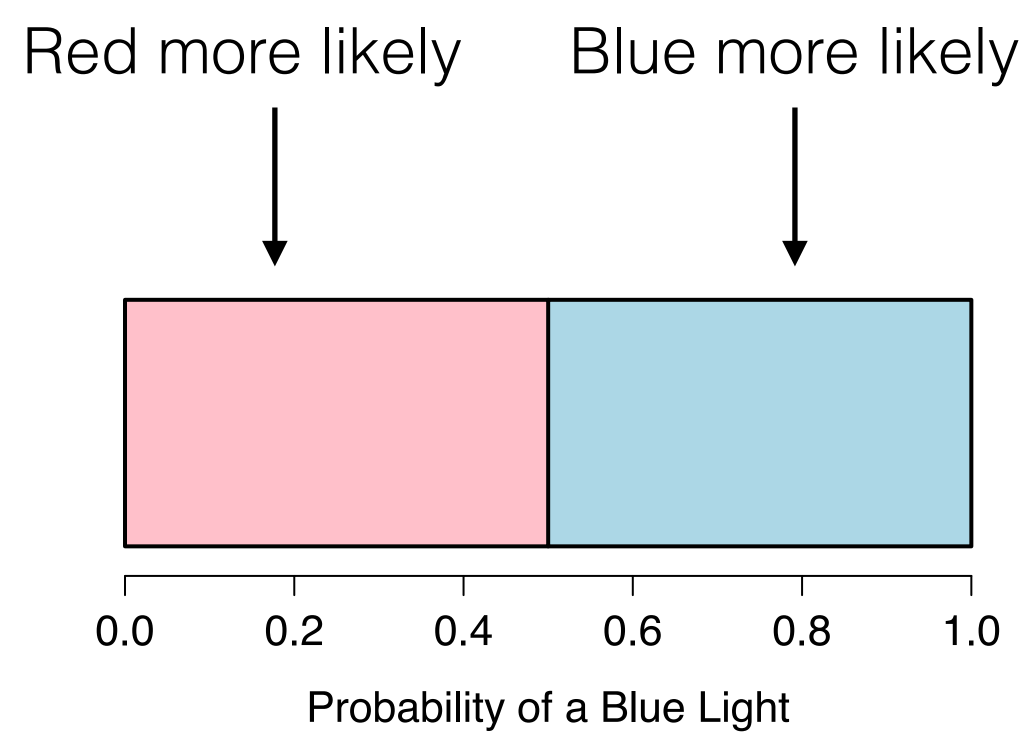
- Objective: get as many points as possible
  - Correct predictions (“winning bets”) win 1 point
  - Incorrect predictions (“losing bets”) lose 1 point

- Task structure:
  - If you **observe** the outcome, you can’t bet so you give up any chance on getting a reward
  - If you **bet** on the outcome, you don’t find out if you were right or wrong (until the end of the task)

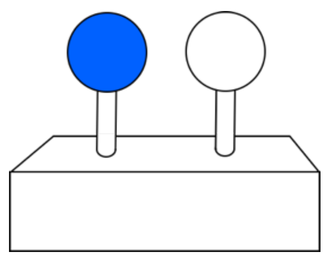
# Bayesian inference

Prior beliefs about the probability that the light will be blue

$$P(\theta)$$

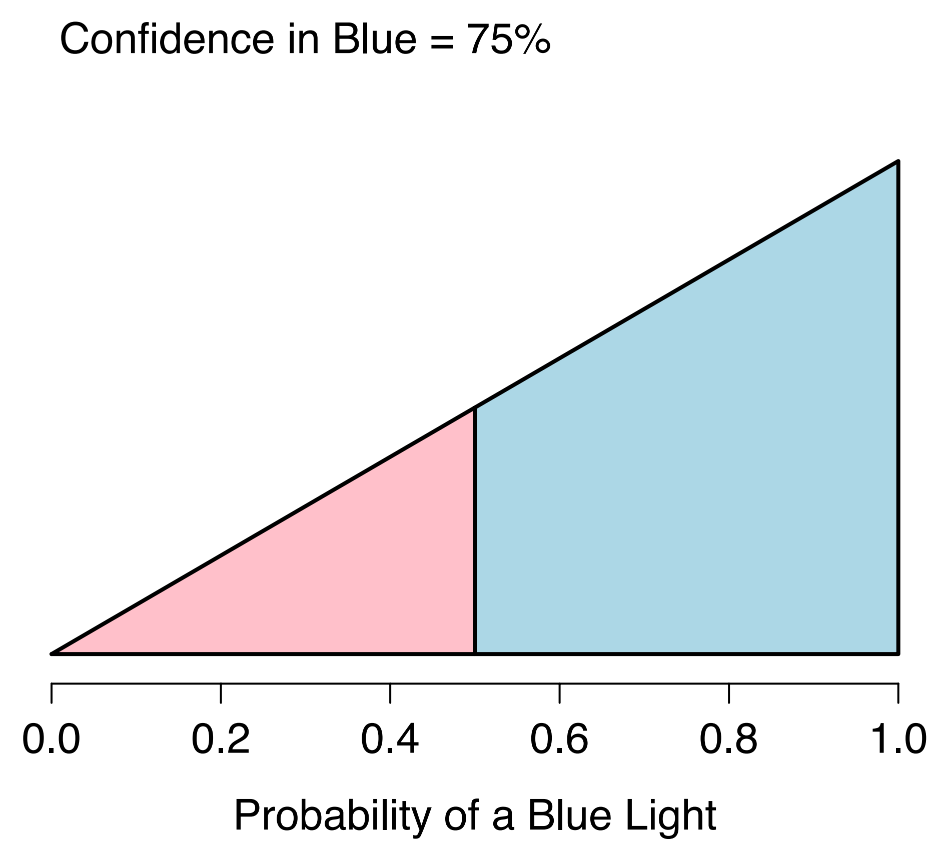


# Bayesian inference

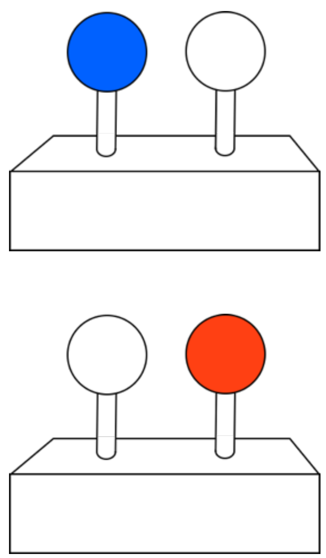


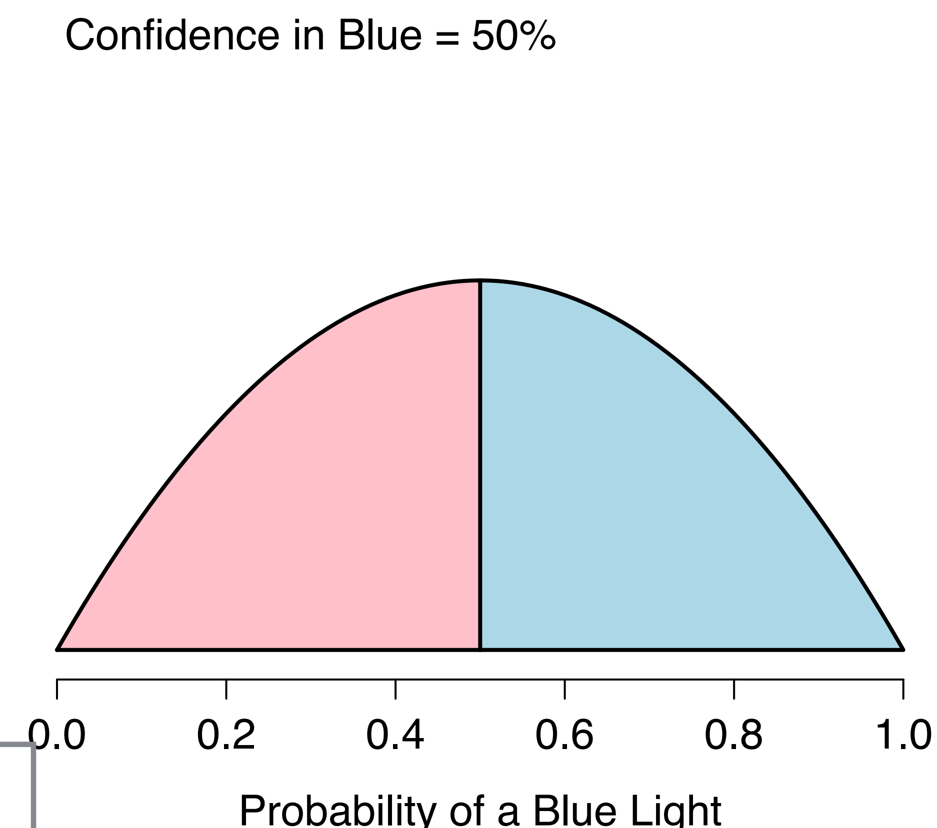
Posterior beliefs given a single OBSERVE action on trial 1

$$P(\theta|x) \propto P(x|\theta)P(\theta)$$



# Bayesian inference

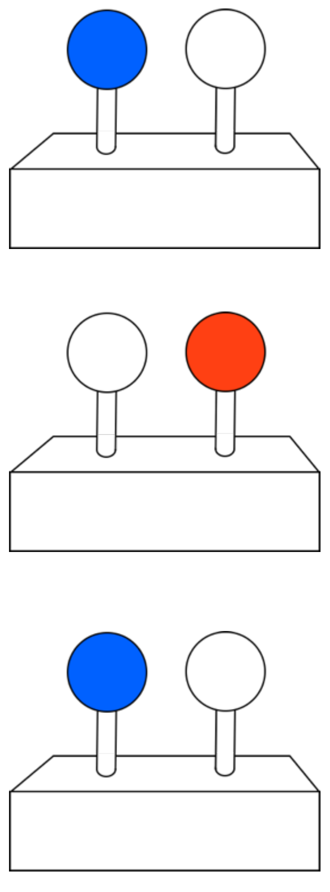


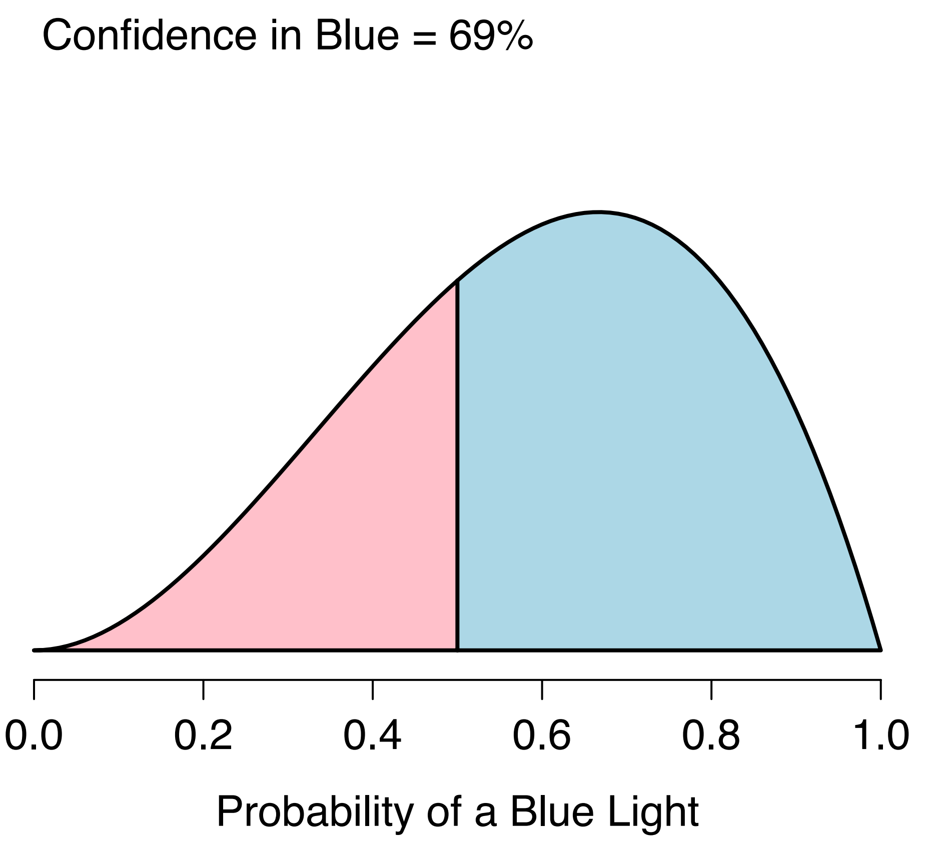


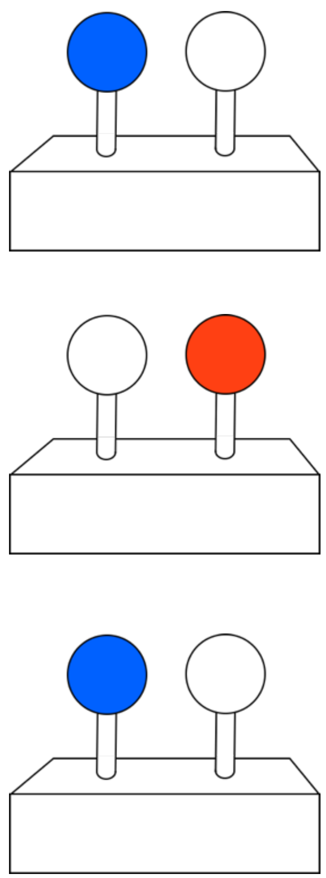

Beliefs updated sequentially: today's posterior is tomorrow's prior

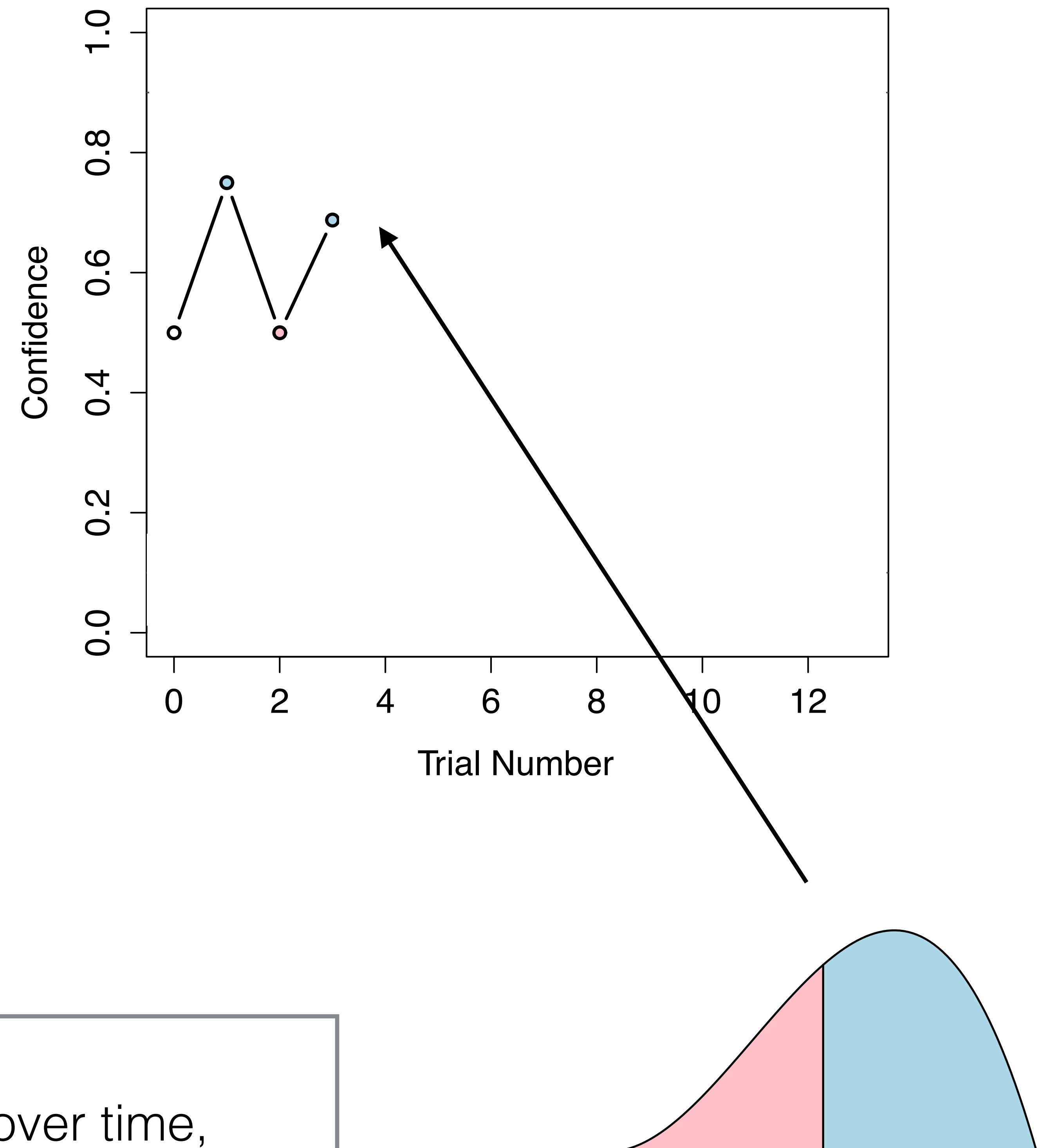
$$P(\theta|\mathbf{x}_t) \propto P(x_t|\theta)P(\theta|\mathbf{x}_{t-1})$$

# Bayesian inference

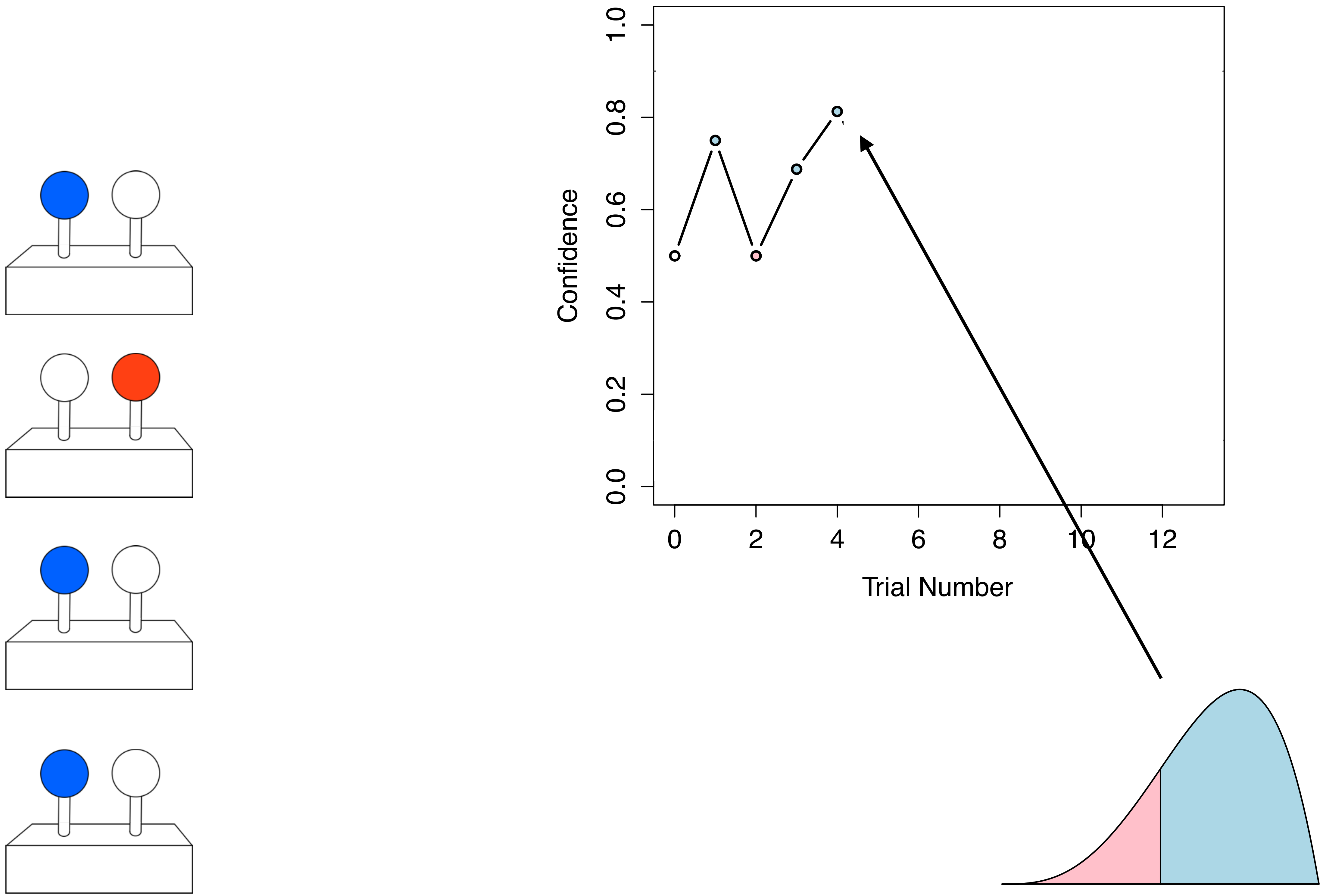




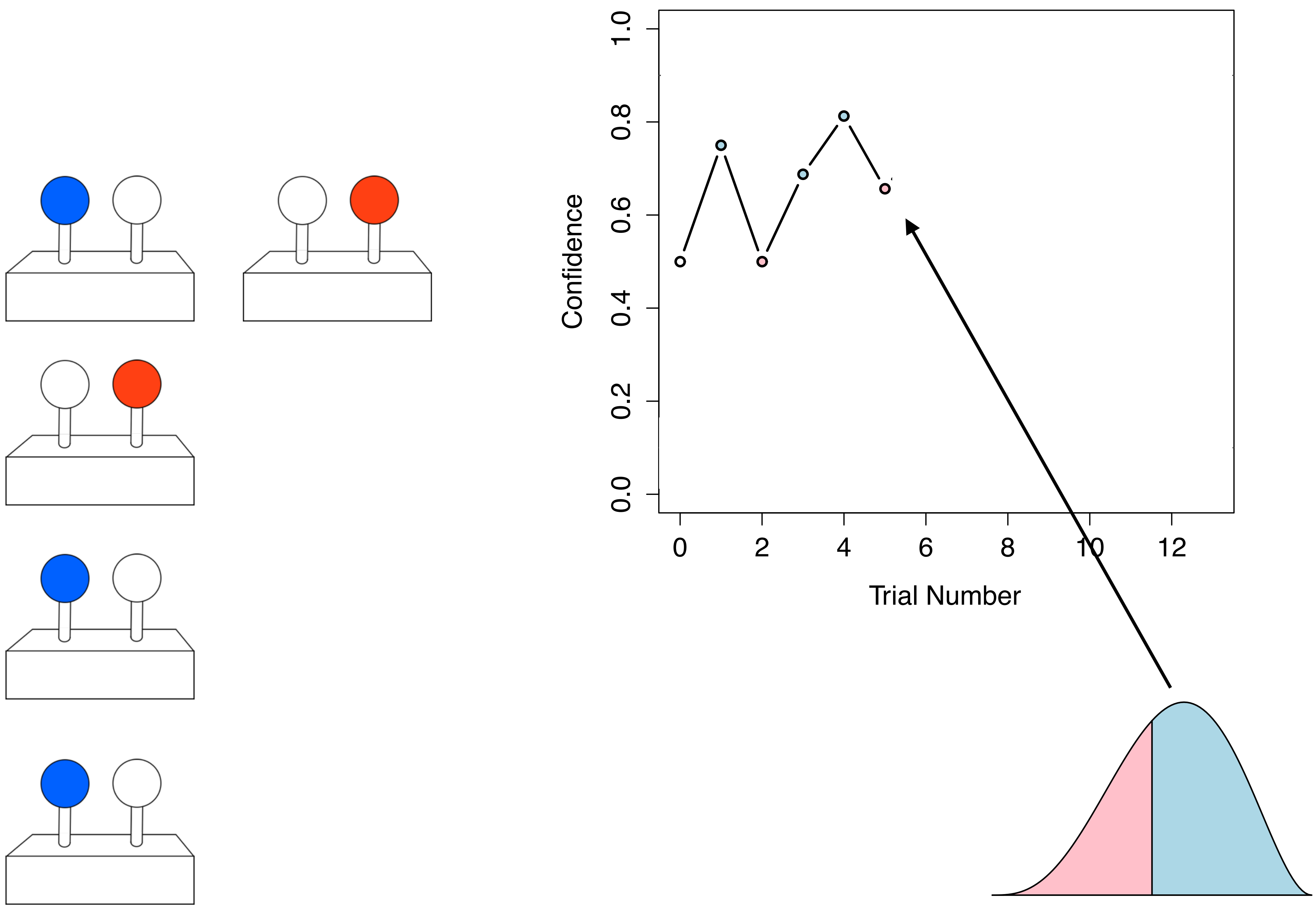


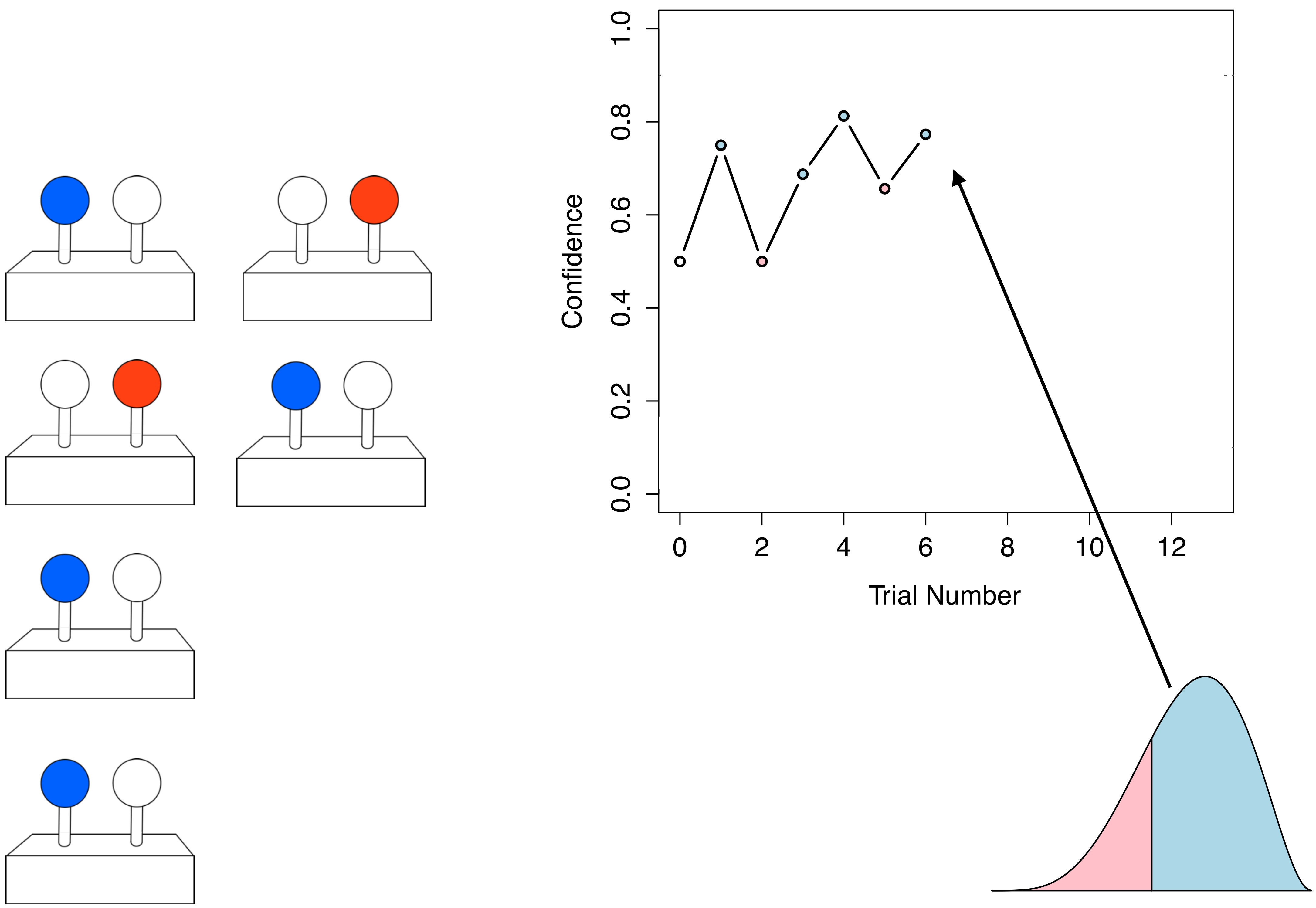



Plot the learner’s confidence over time, as more observations are requested

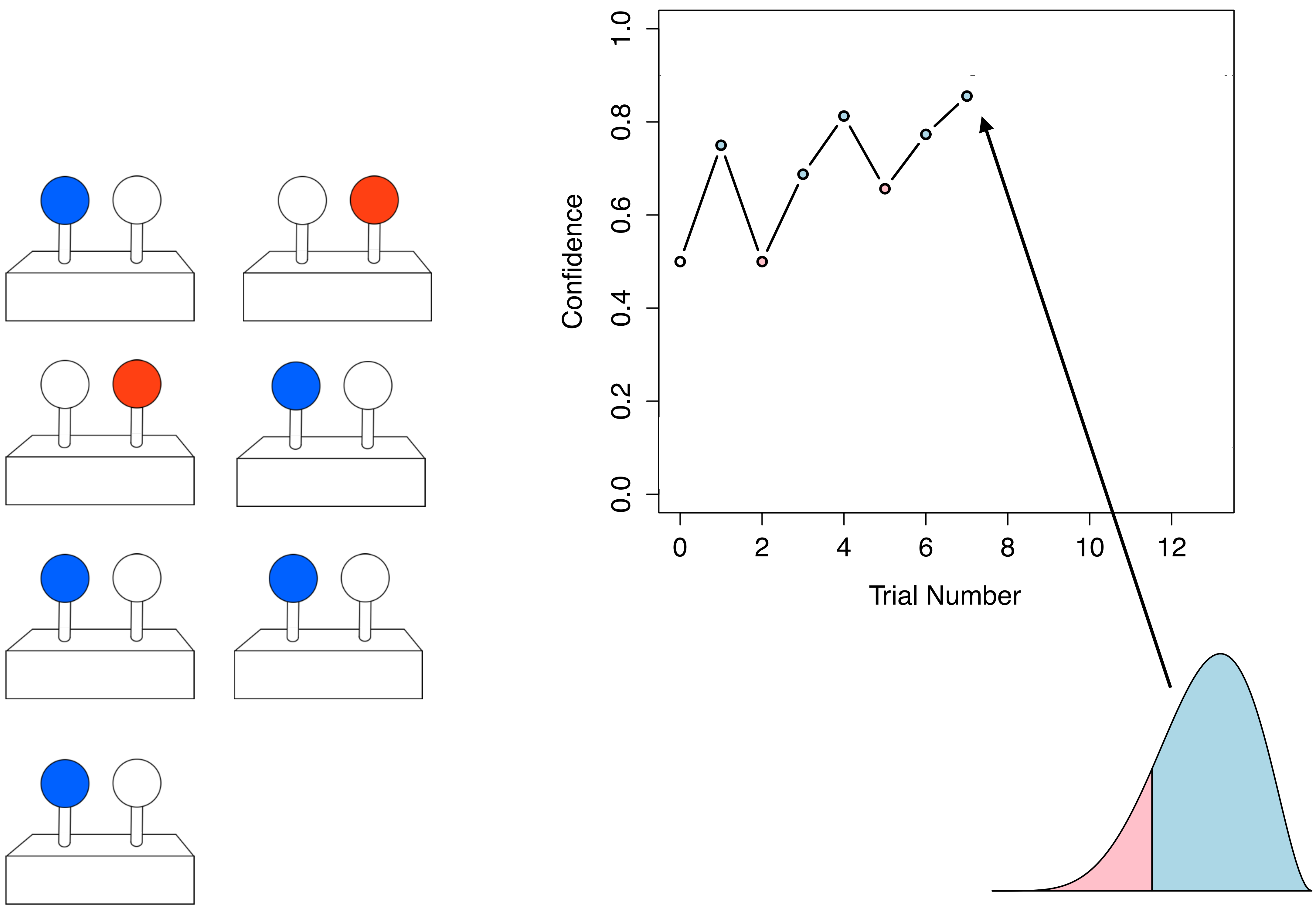


Confidence
Trial Number
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12

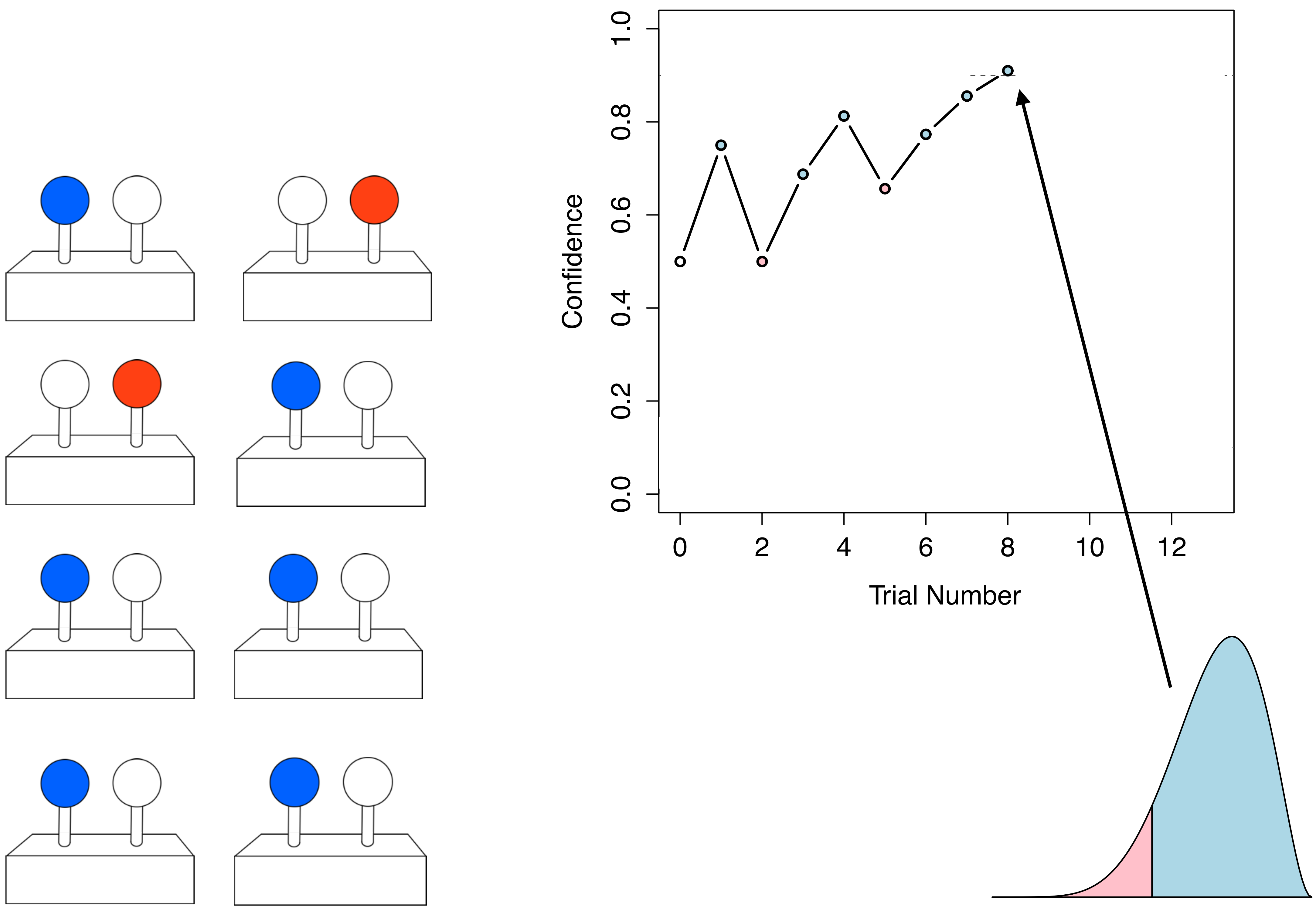

Confidence
Trial Number
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12



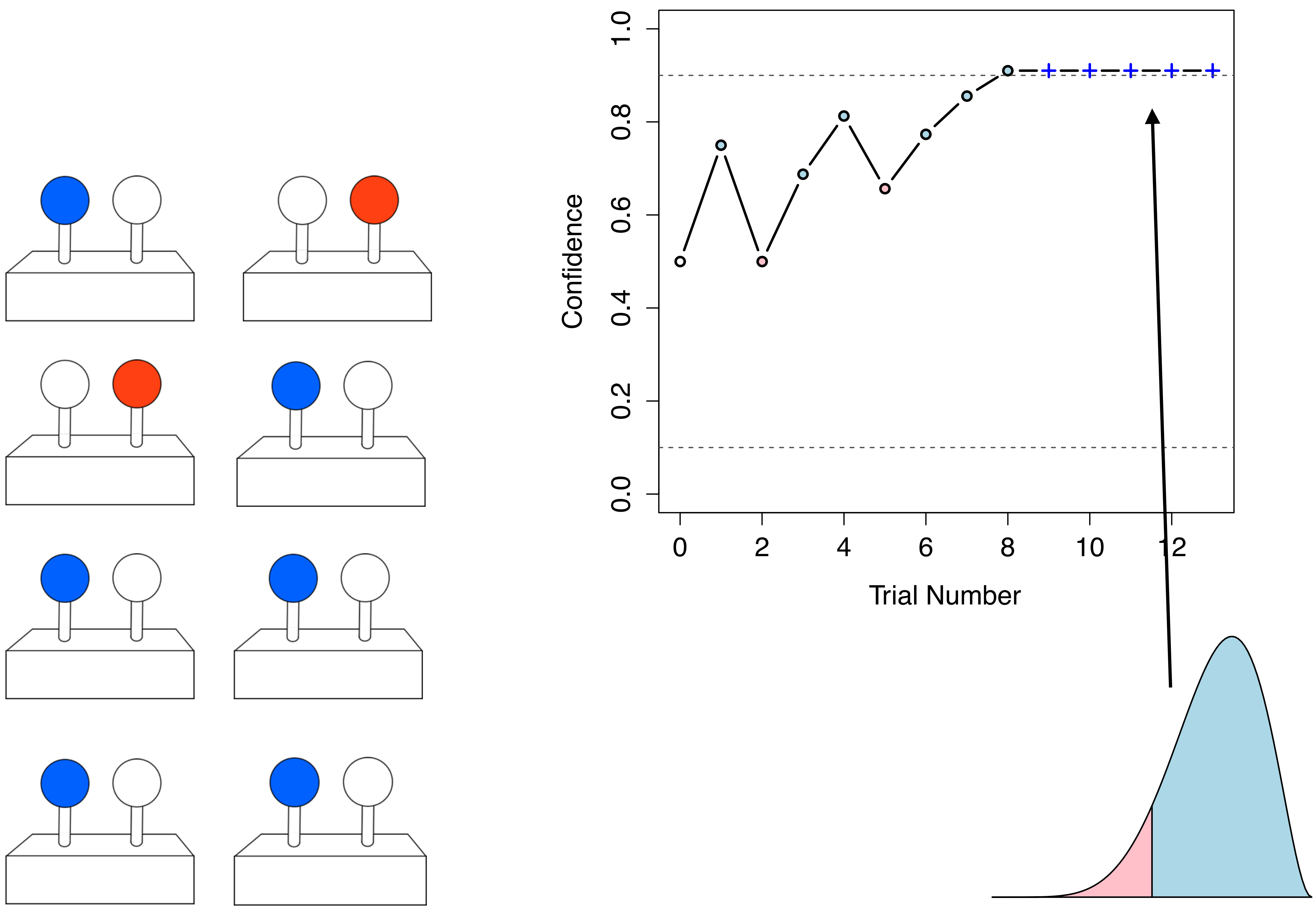
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
Trial Number



Confidence
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12
Trial Number

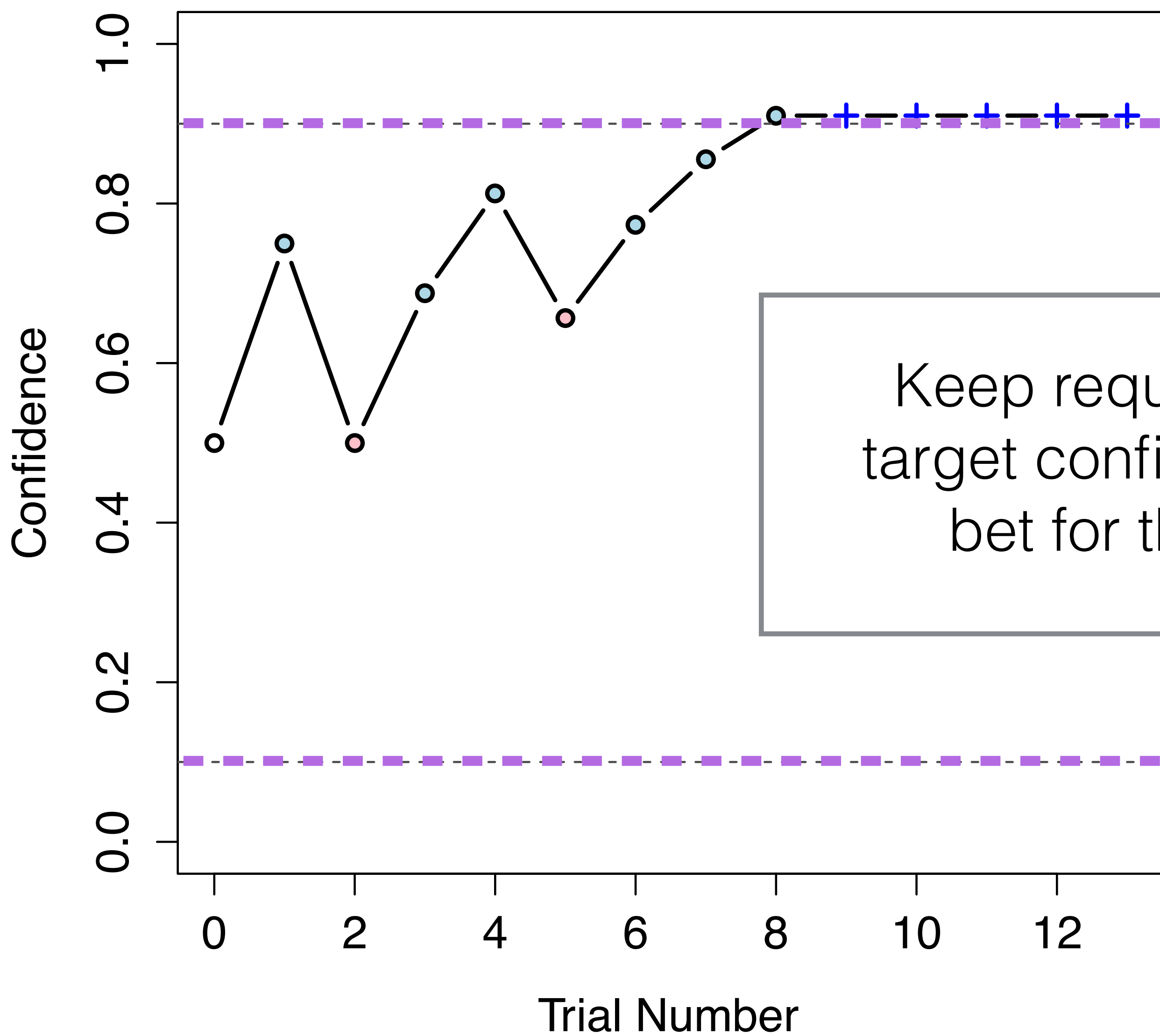


Confidence
Trial Number
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12



Confidence
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12
Trial Number

# Optimal* policy is the random walk model for 2-AFC tasks…




Keep requesting observations until a target confidence level is reached then bet for the rest of the experiment

“Optimal” policy:

**O O O O O O O B B B B B B B B B B B B B B B B B B**

Keep making observations until you figure out the right betting strategy

Then trust blindly in your strategy forever more

"Optimal" policy:

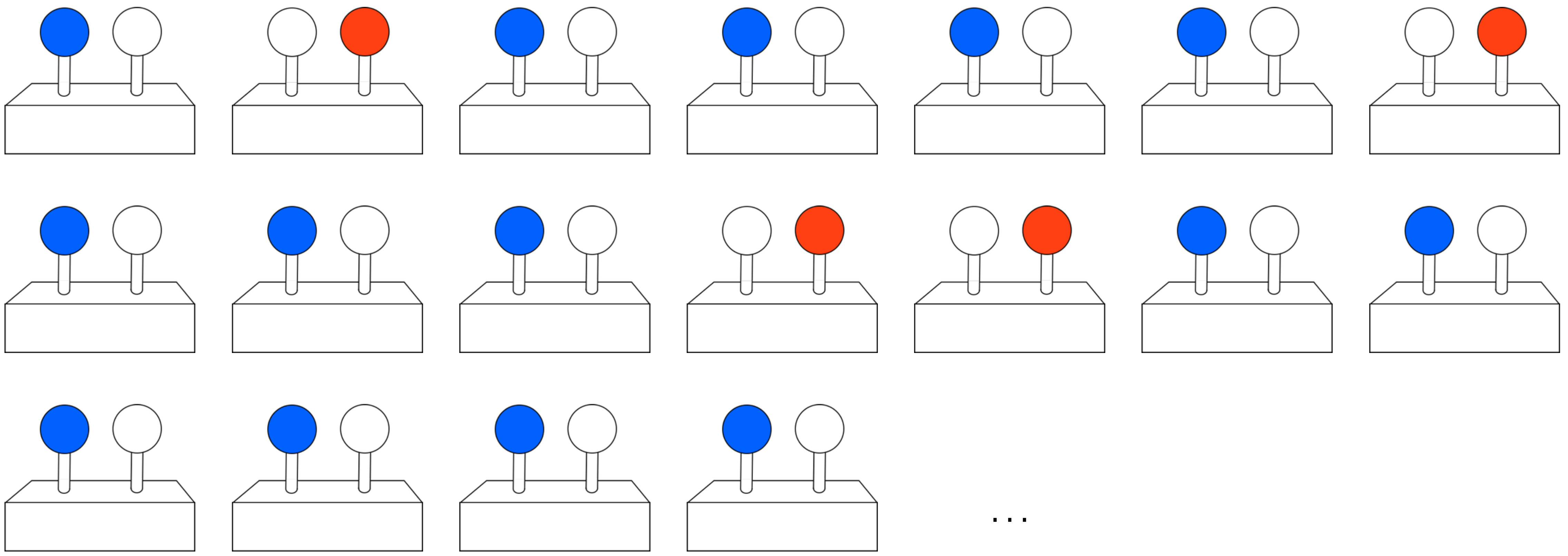
O O O O O O O B B B B B B B B B B B B B B B B B B

Grossly typical pattern of human performance:

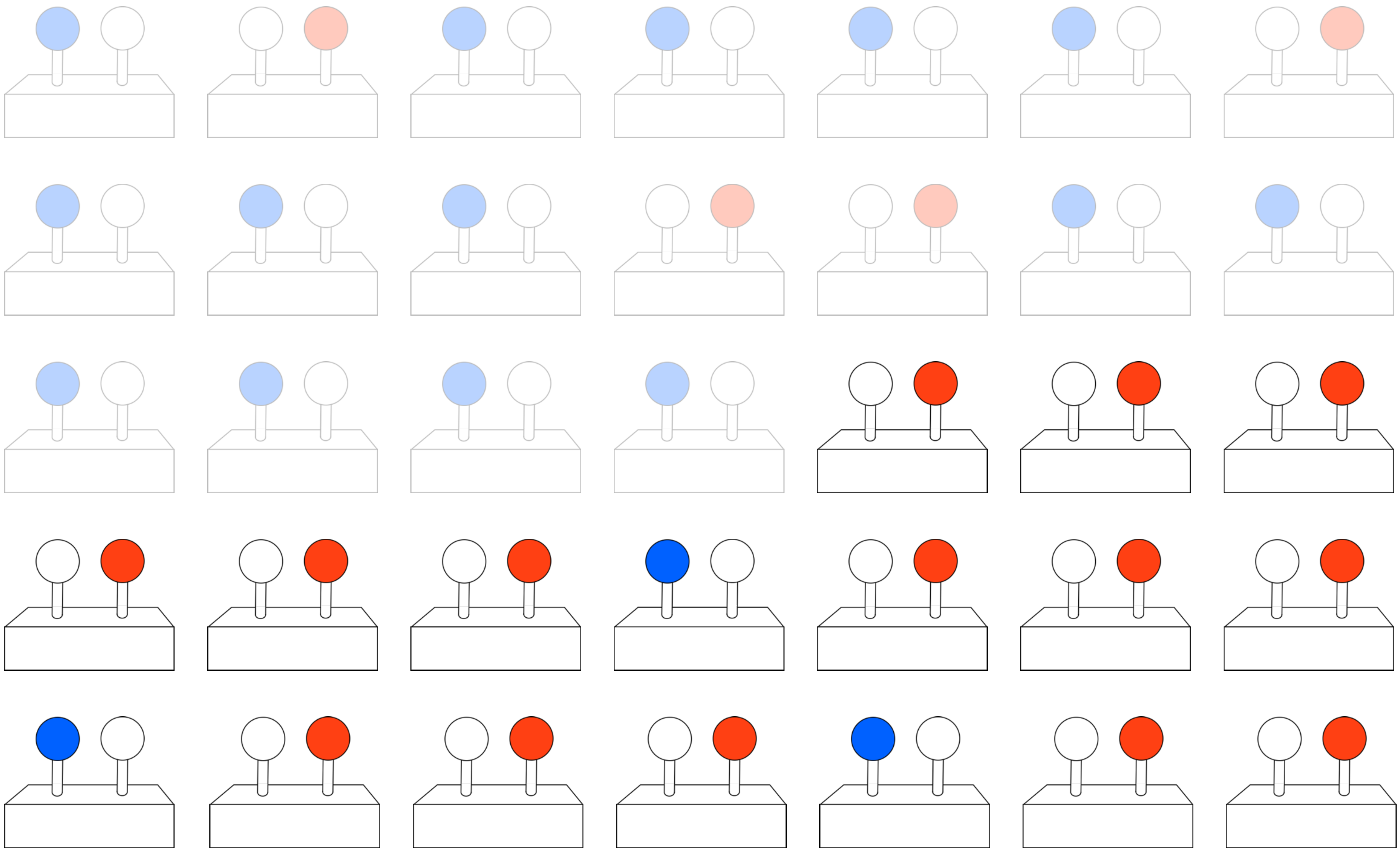
O O O O O O O B B B B B B O O B B B B B B O B B B

Humans don't trust their strategy:
they constantly check to see if it is
working

# Why check?



# Because things change.



# What's the difference?

**Static world**: today's posterior is tomorrow's prior

$$P(\theta|\mathbf{x}_t) \propto P(x_t|\theta)P(\theta|\mathbf{x}_{t-1})$$

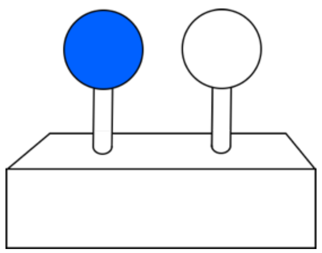
# What's the difference?

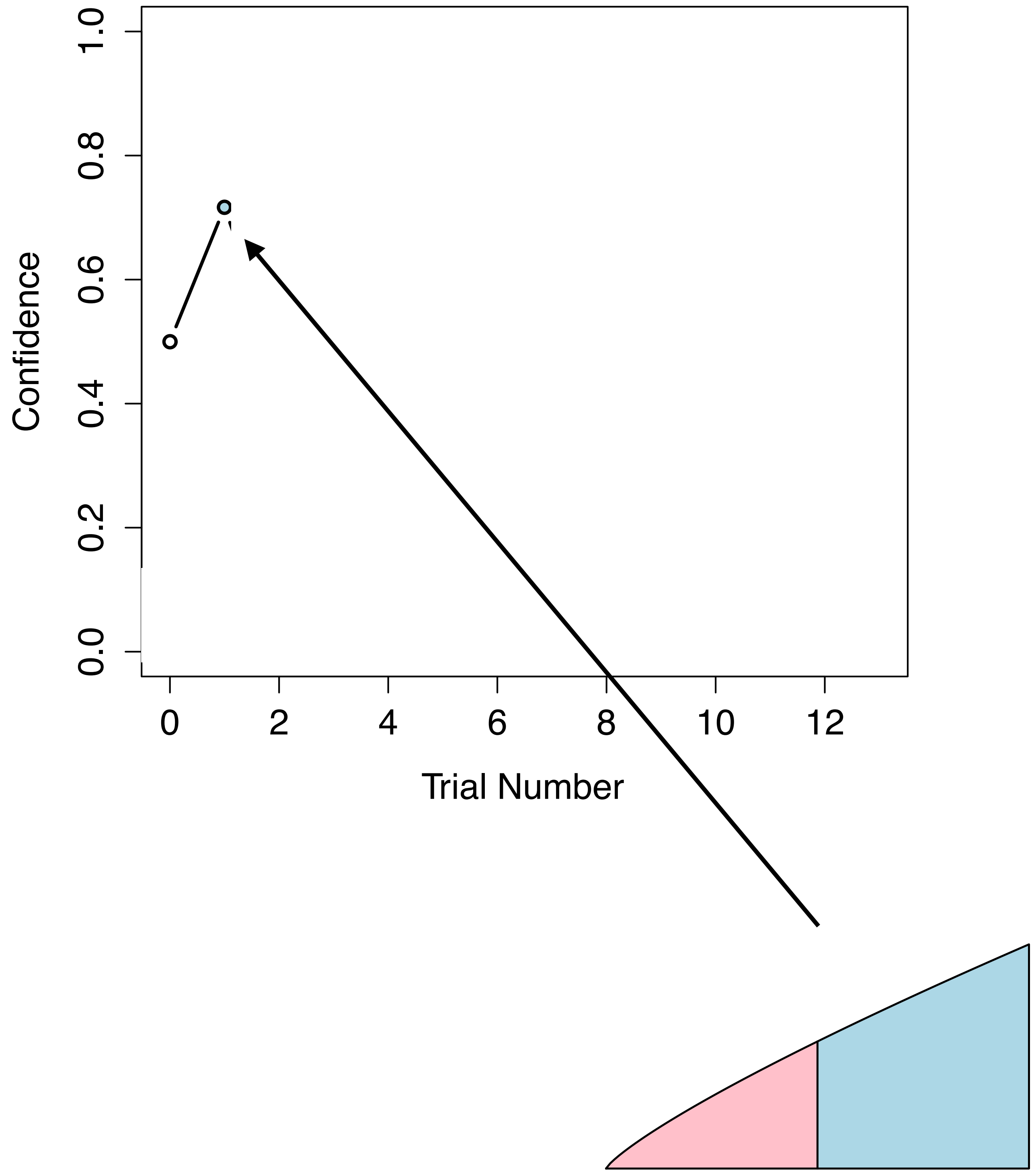
**Static world**: today's posterior is tomorrow's prior

$$P(\theta|\mathbf{x}_t) \propto P(x_t|\theta)P(\theta|\mathbf{x}_{t-1})$$

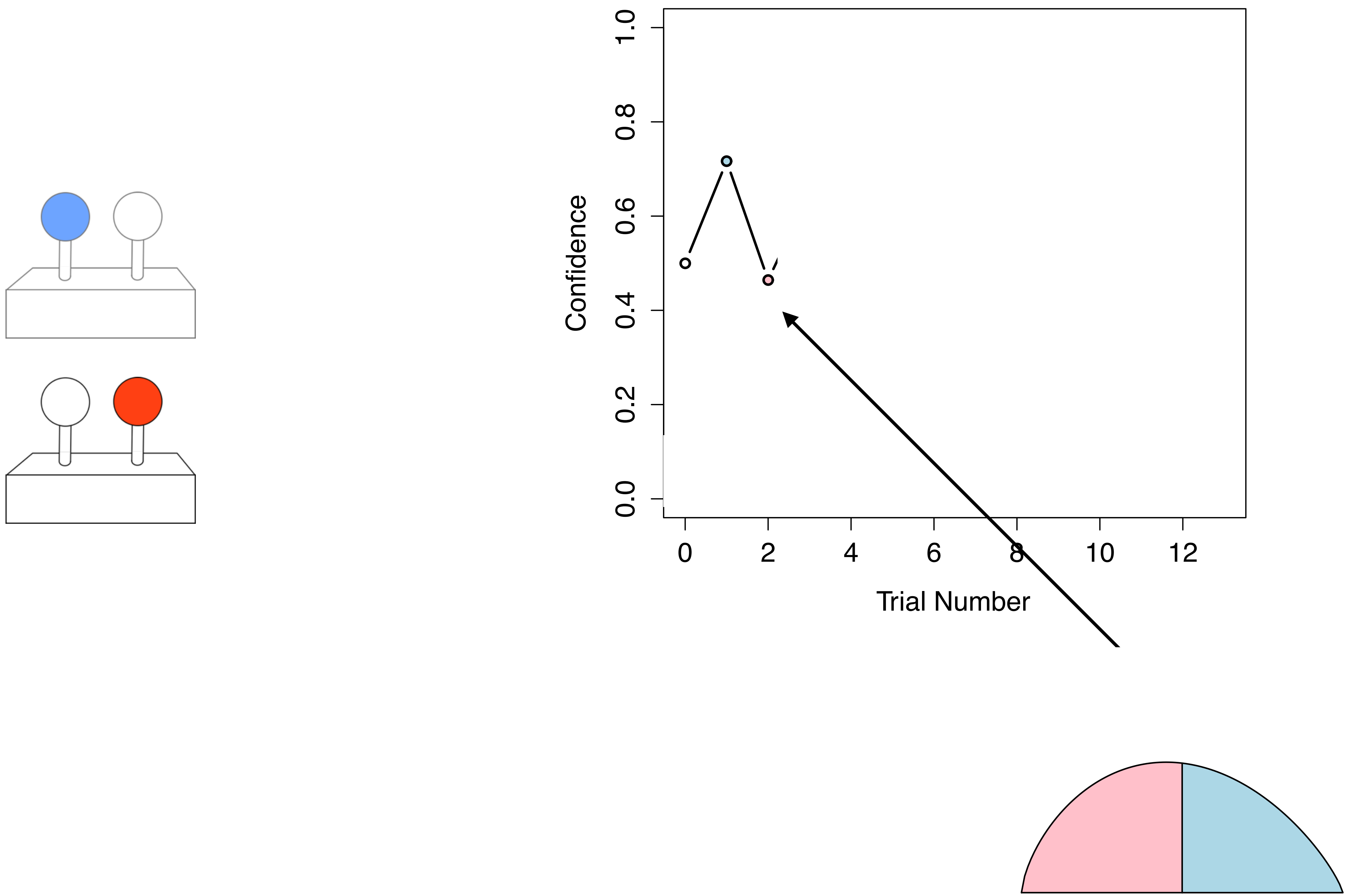
**Dynamic world**: today's posterior shapes tomorrow's prior, but the world changes a bit overnight…

$$P(\theta_t|\mathbf{x}_t) \propto P(x_t|\theta_t) \int_0^1 P(\theta_t|\theta_{t-1})P(\theta_{t-1}|\mathbf{x}_{t-1})\ d\theta_{t-1}$$

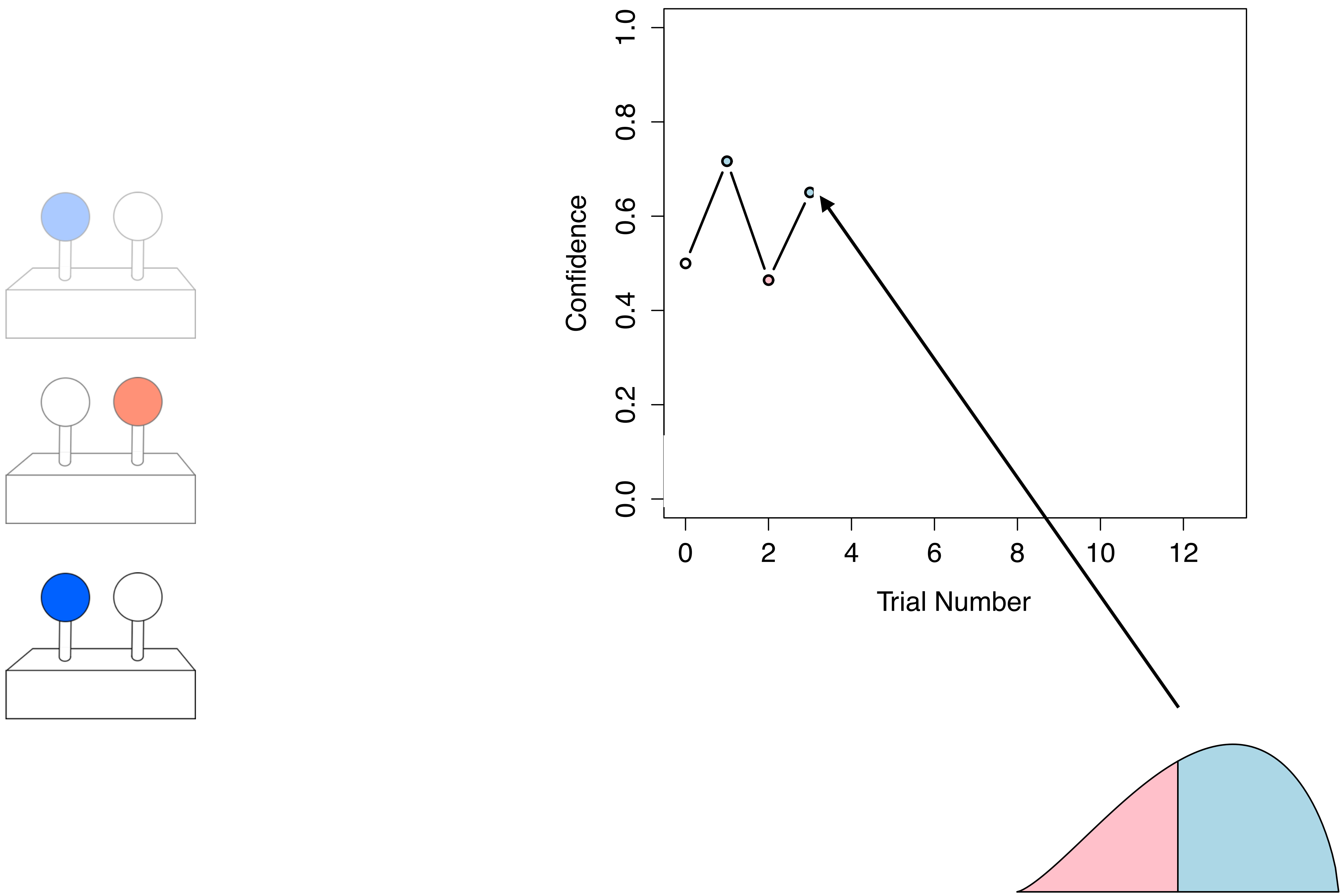




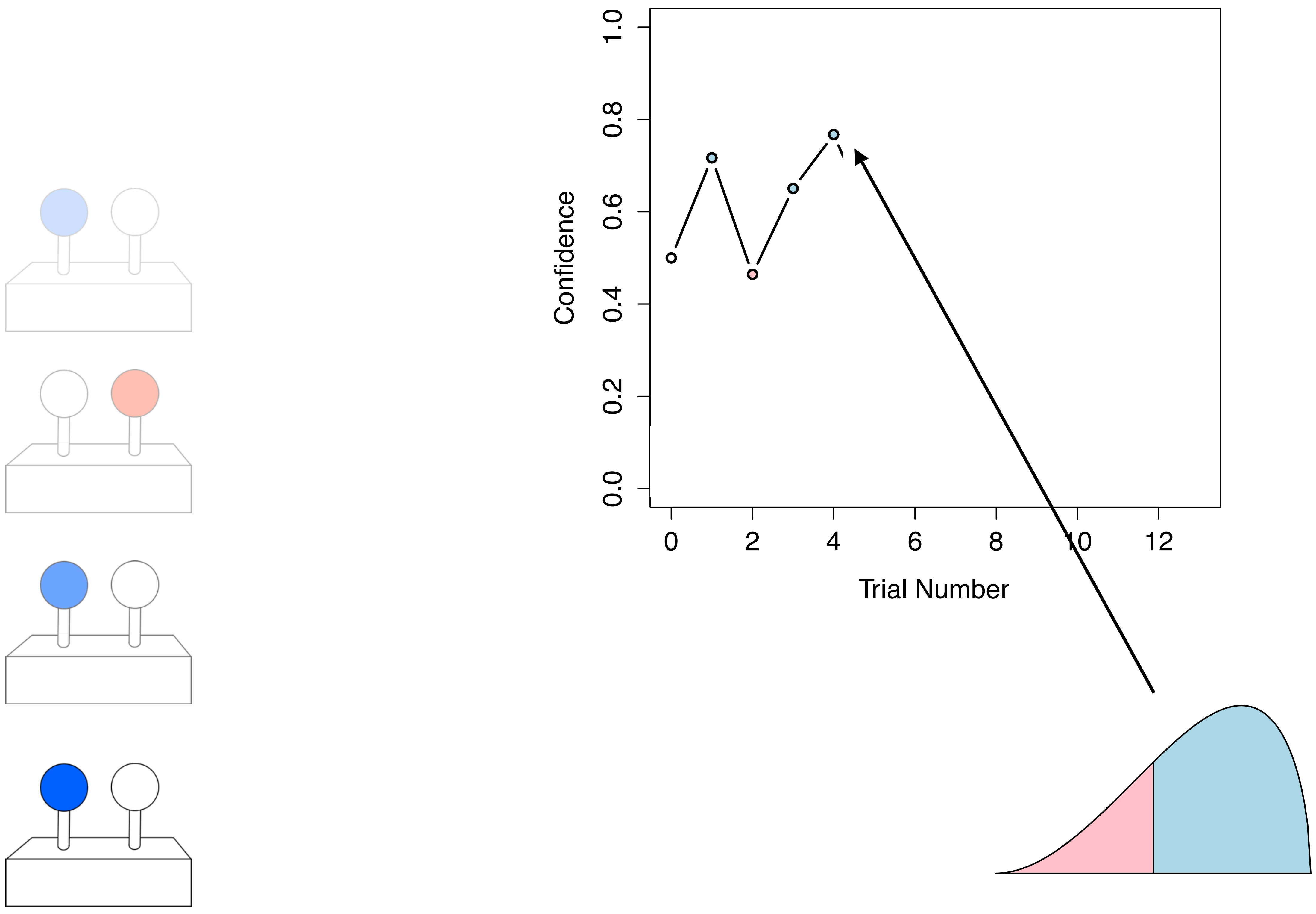
1.0
0.8
0.6
0.4
0.2
0.0
Confidence
0
2
4
6
8
10
12
Trial Number

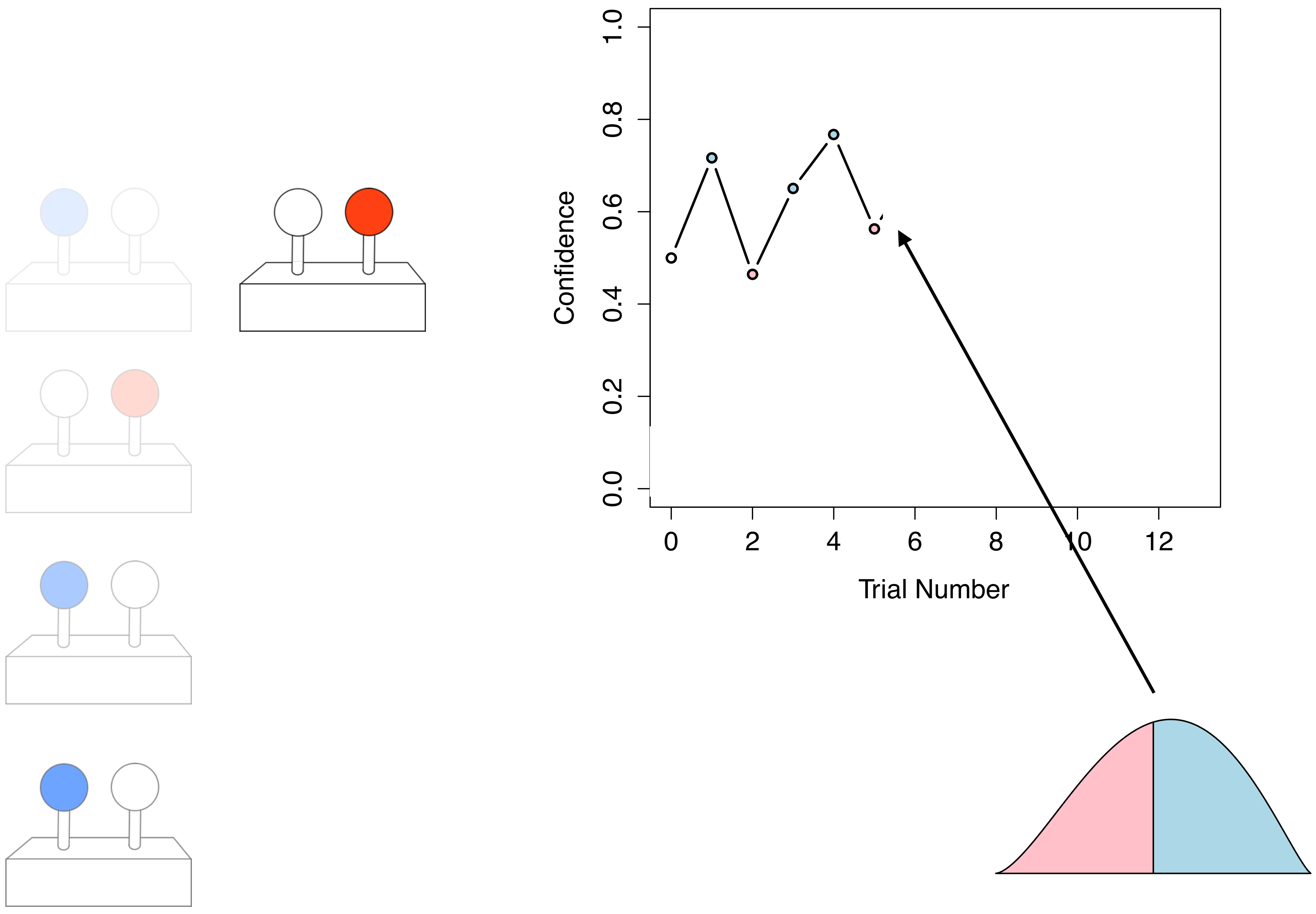


Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
Trial Number

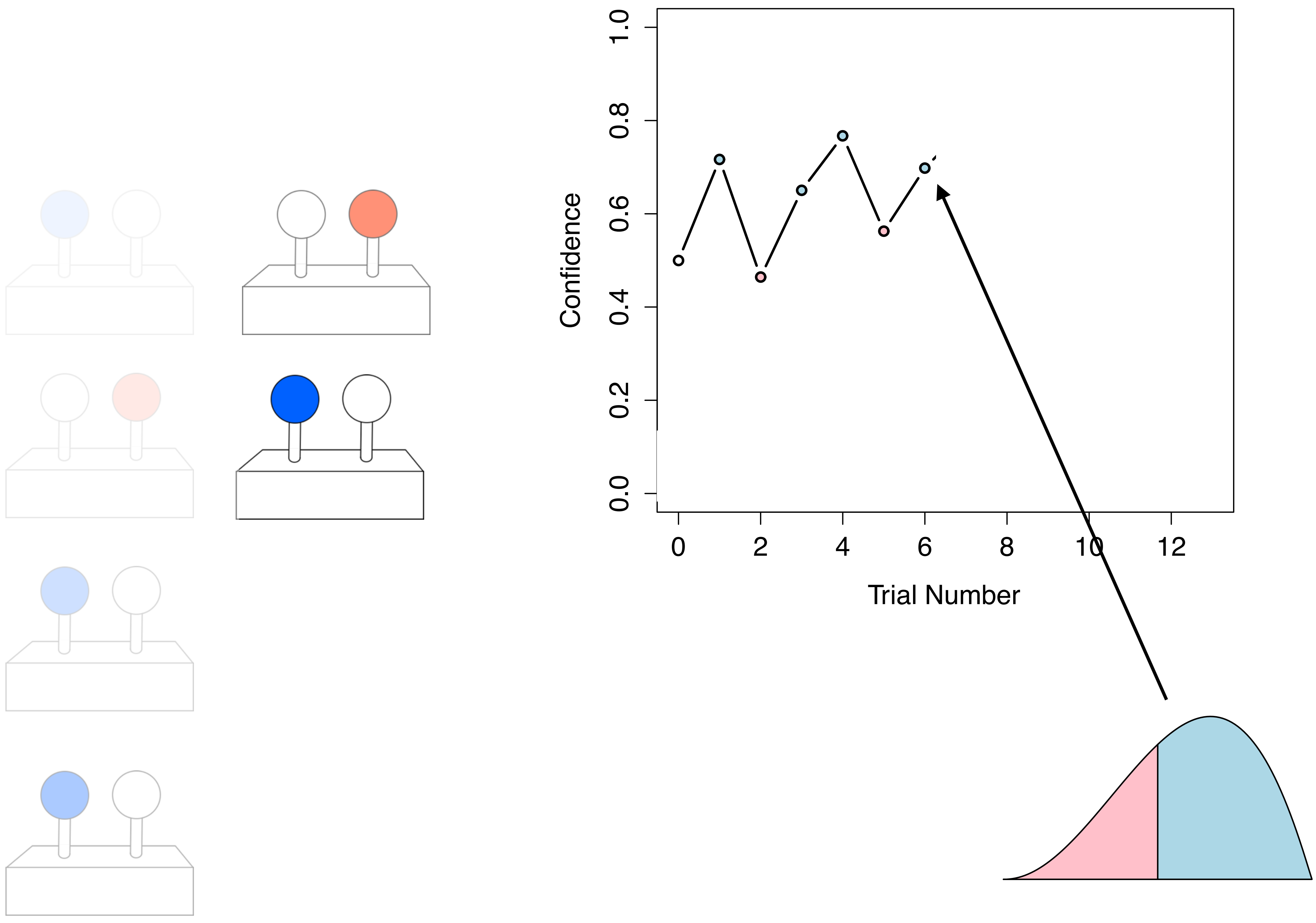


Confidence
Trial Number
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12

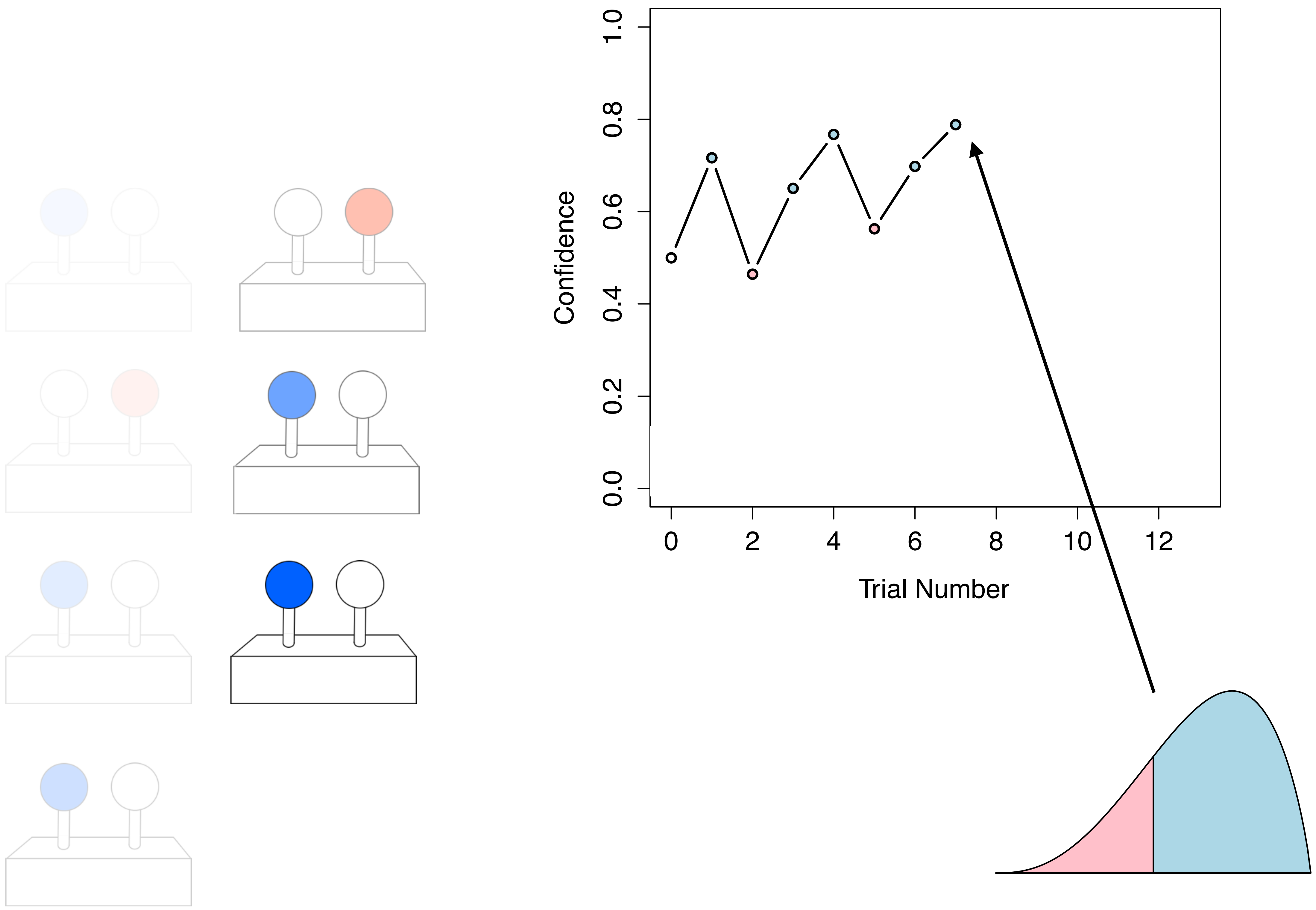

Confidence
Trial Number
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12



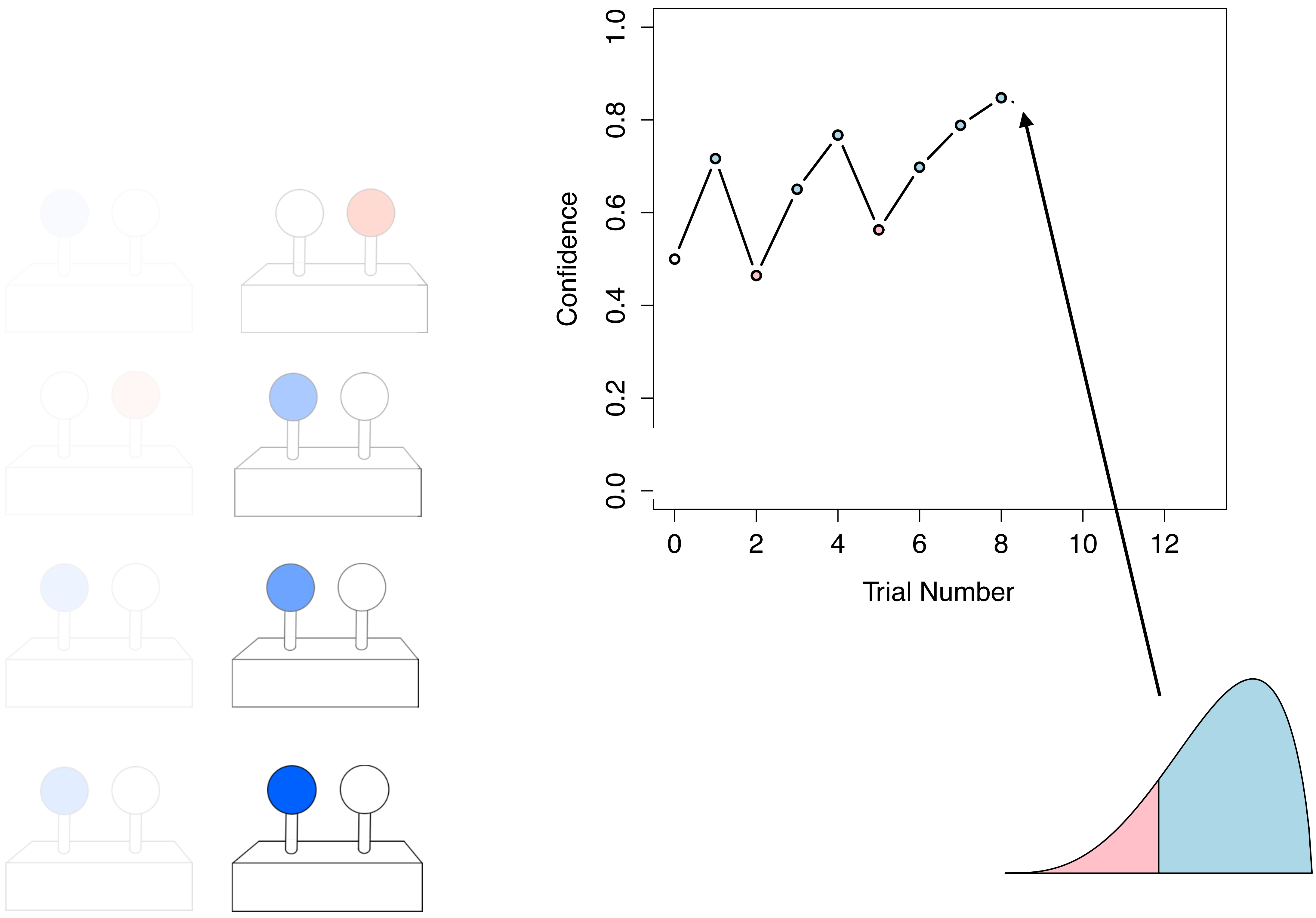
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
Trial Number



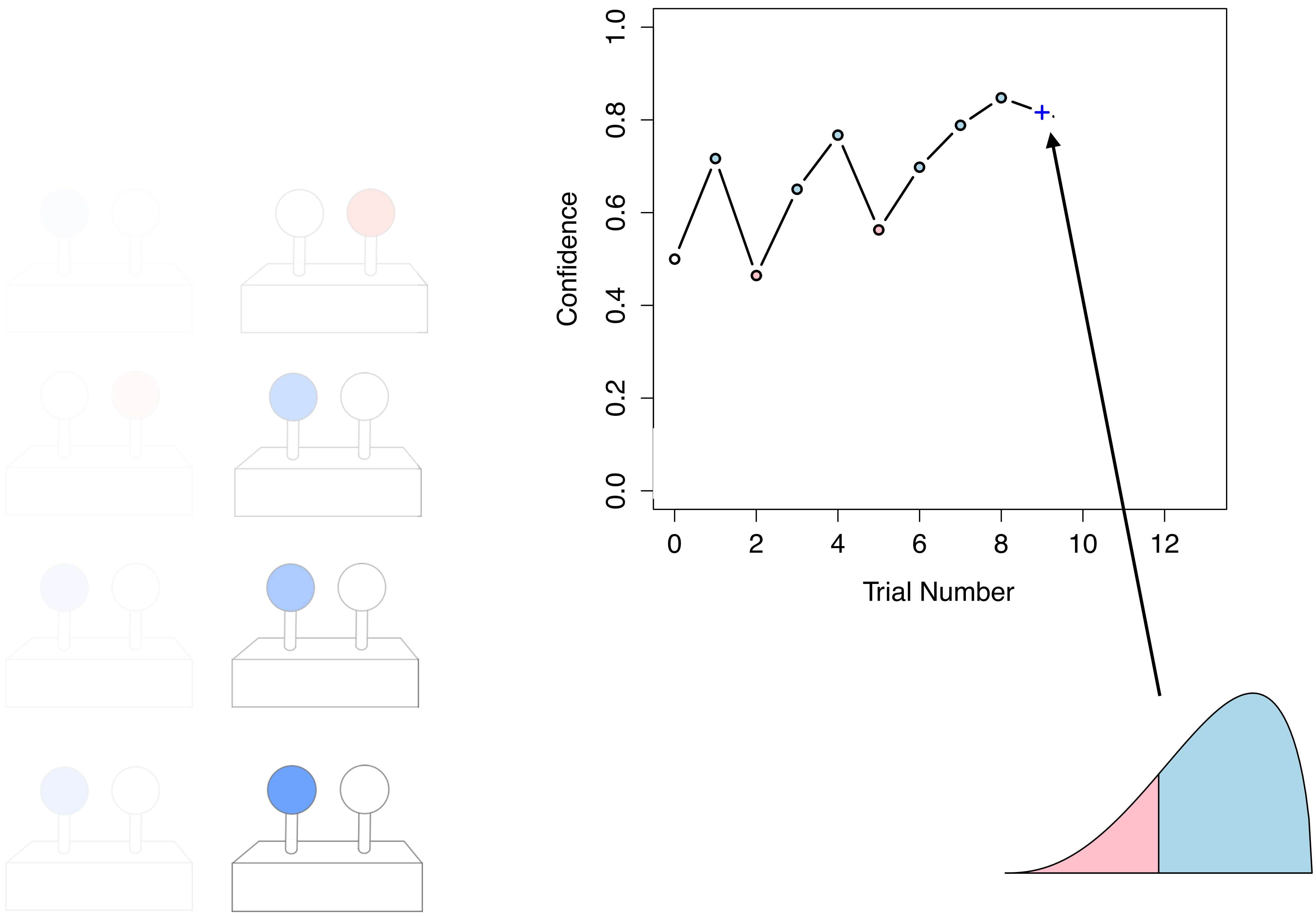
Confidence
Trial Number
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12



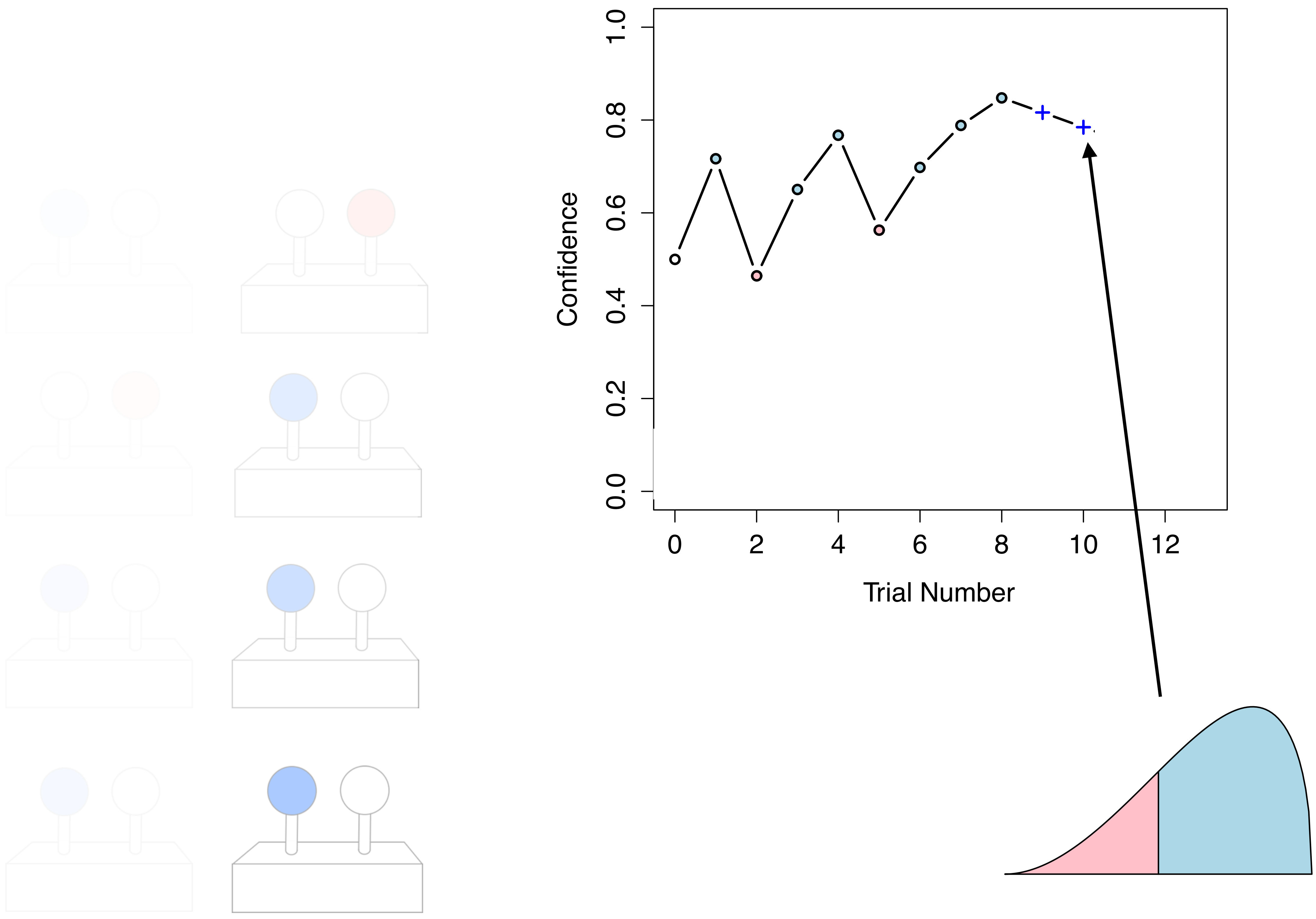
Confidence
Trial Number
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12



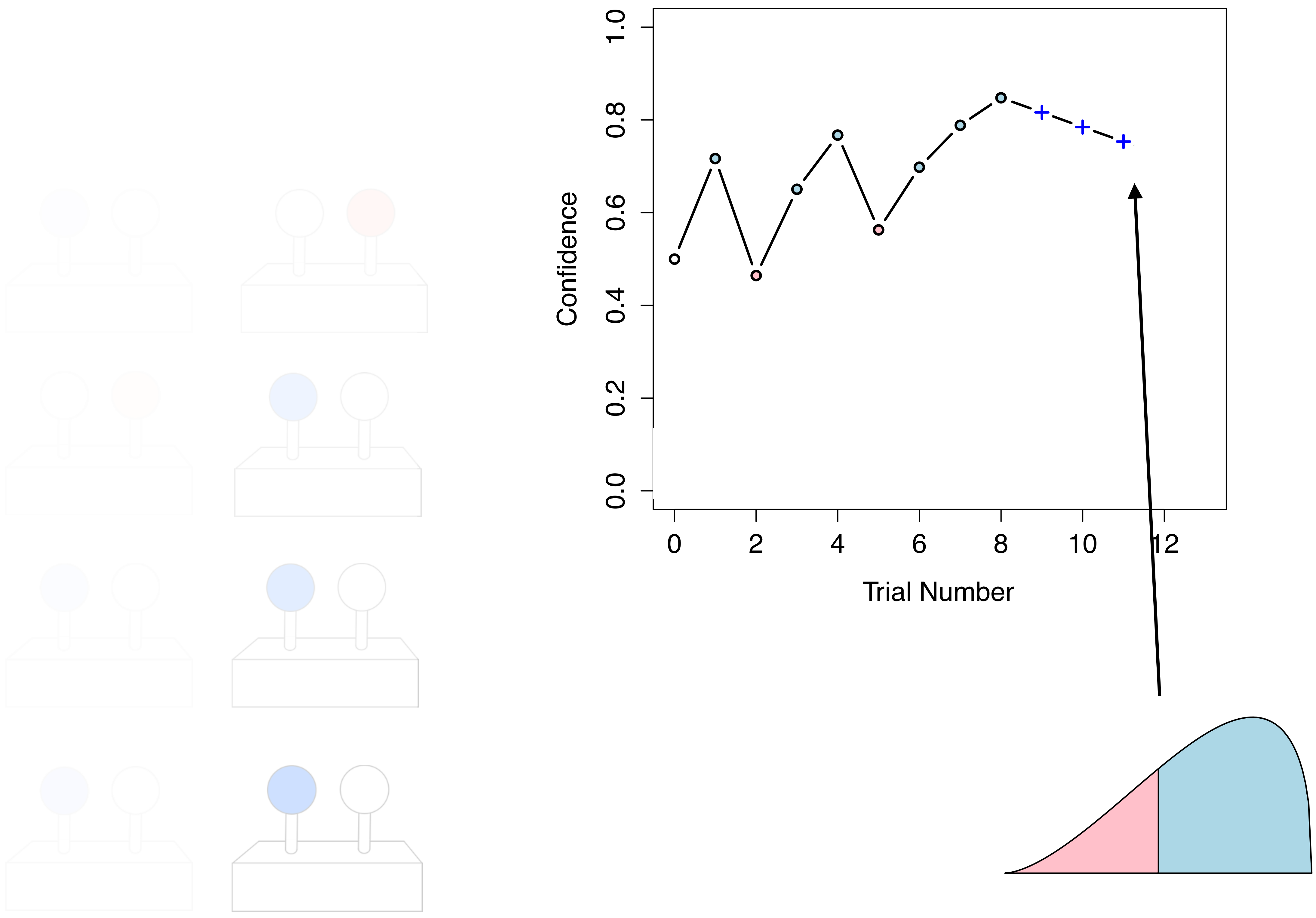
1.0
0.8
0.6
0.4
0.2
0.0
Confidence
0
2
4
6
8
10
12
Trial Number



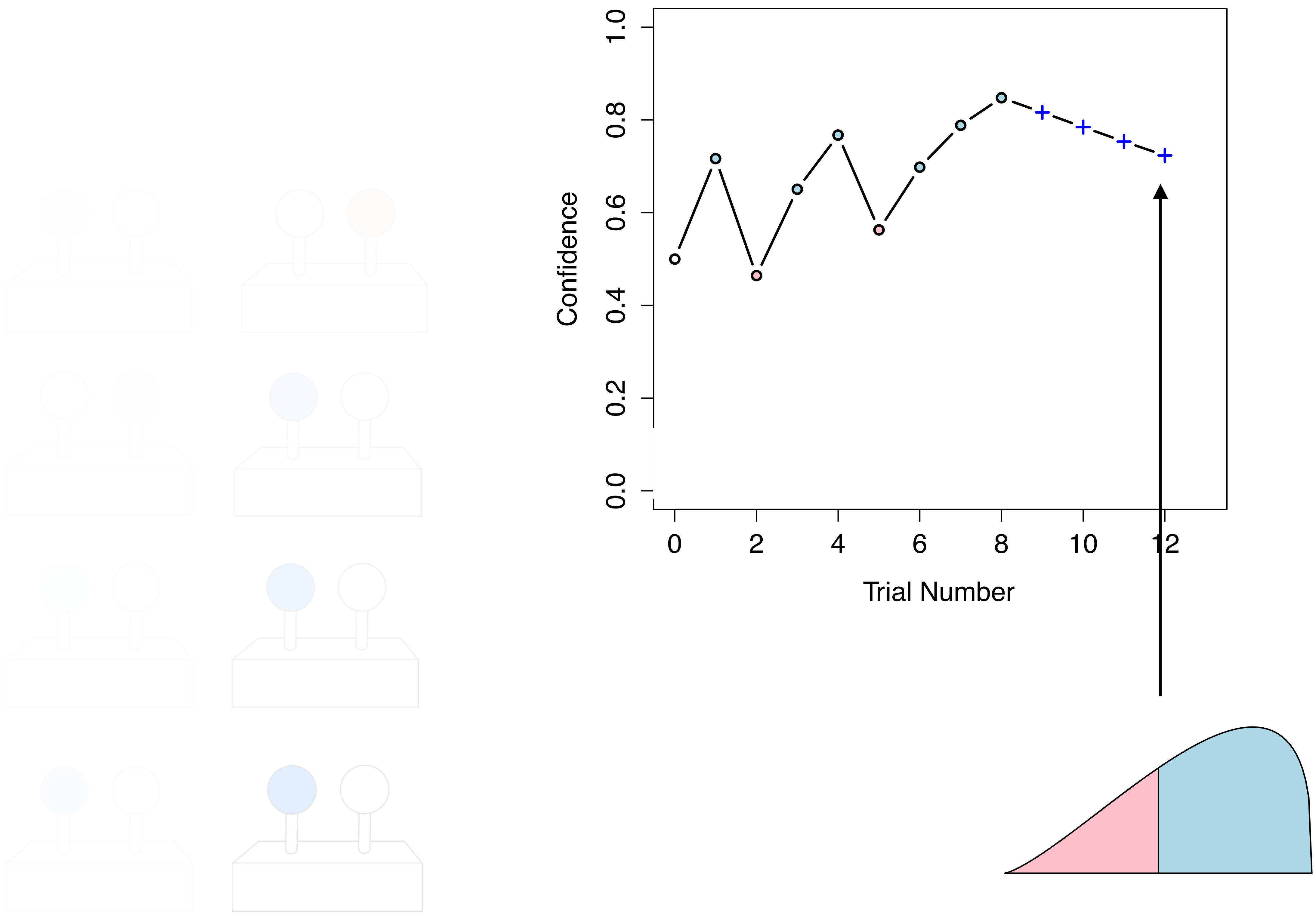
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12
Trial Number



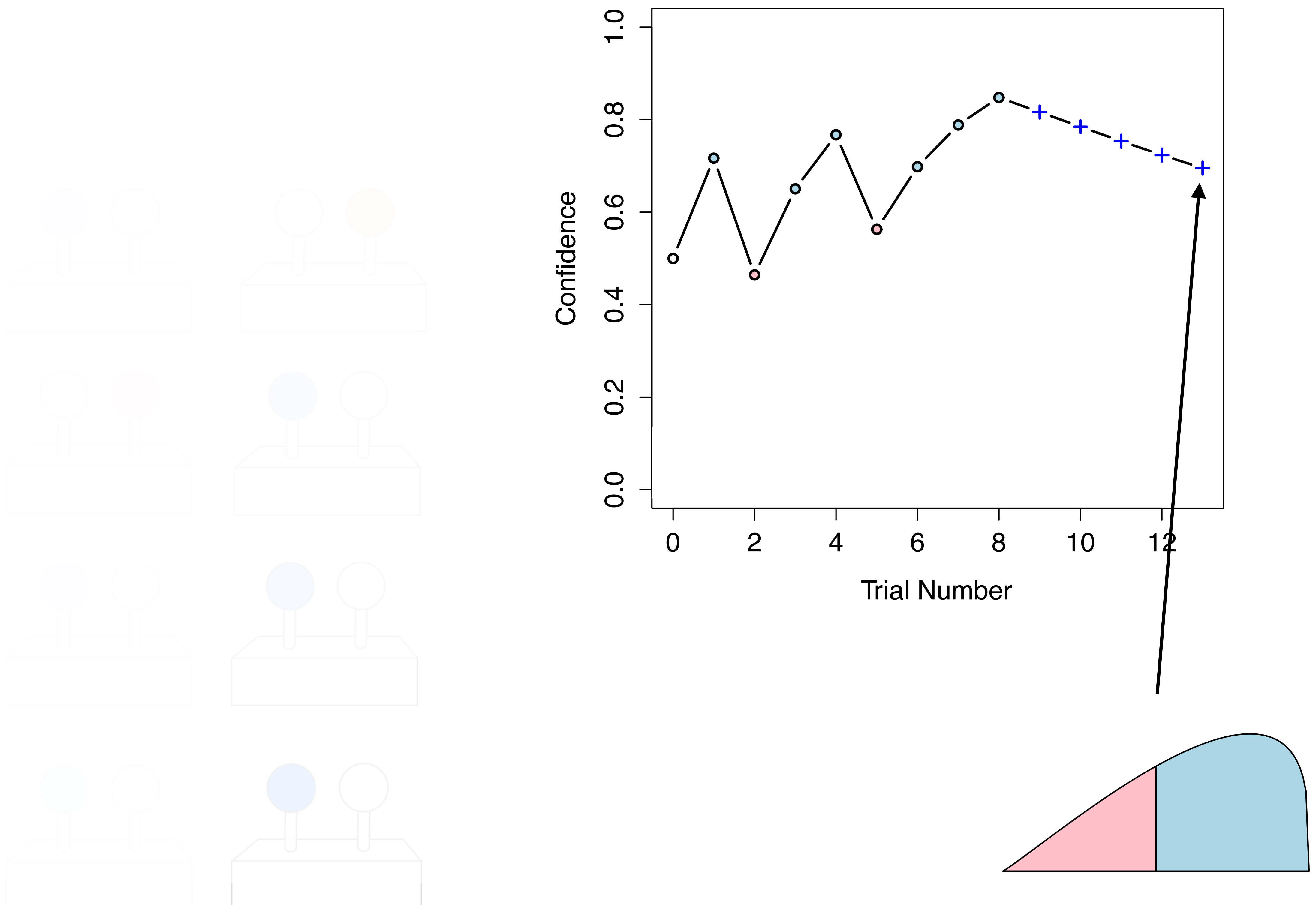
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
Trial Number



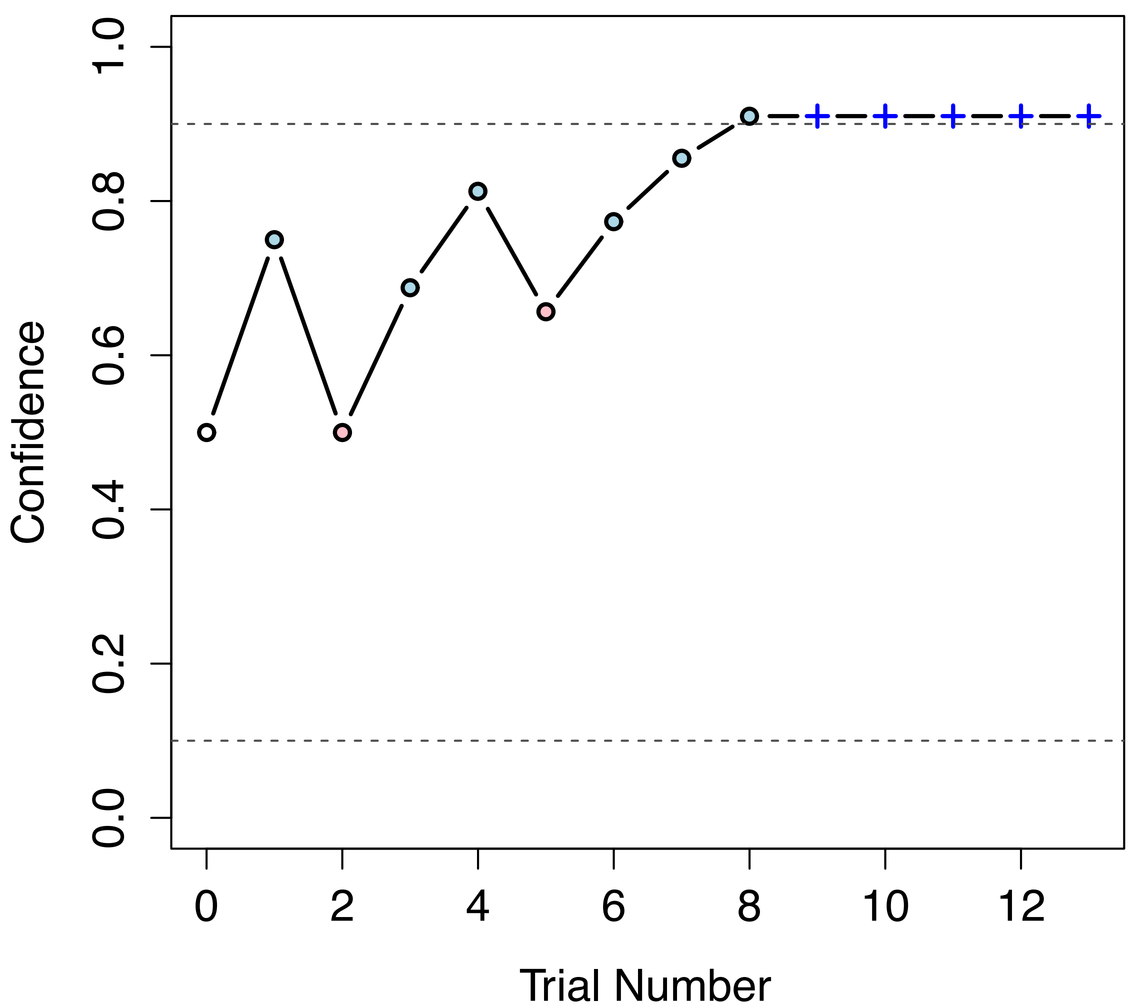
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12
Trial Number



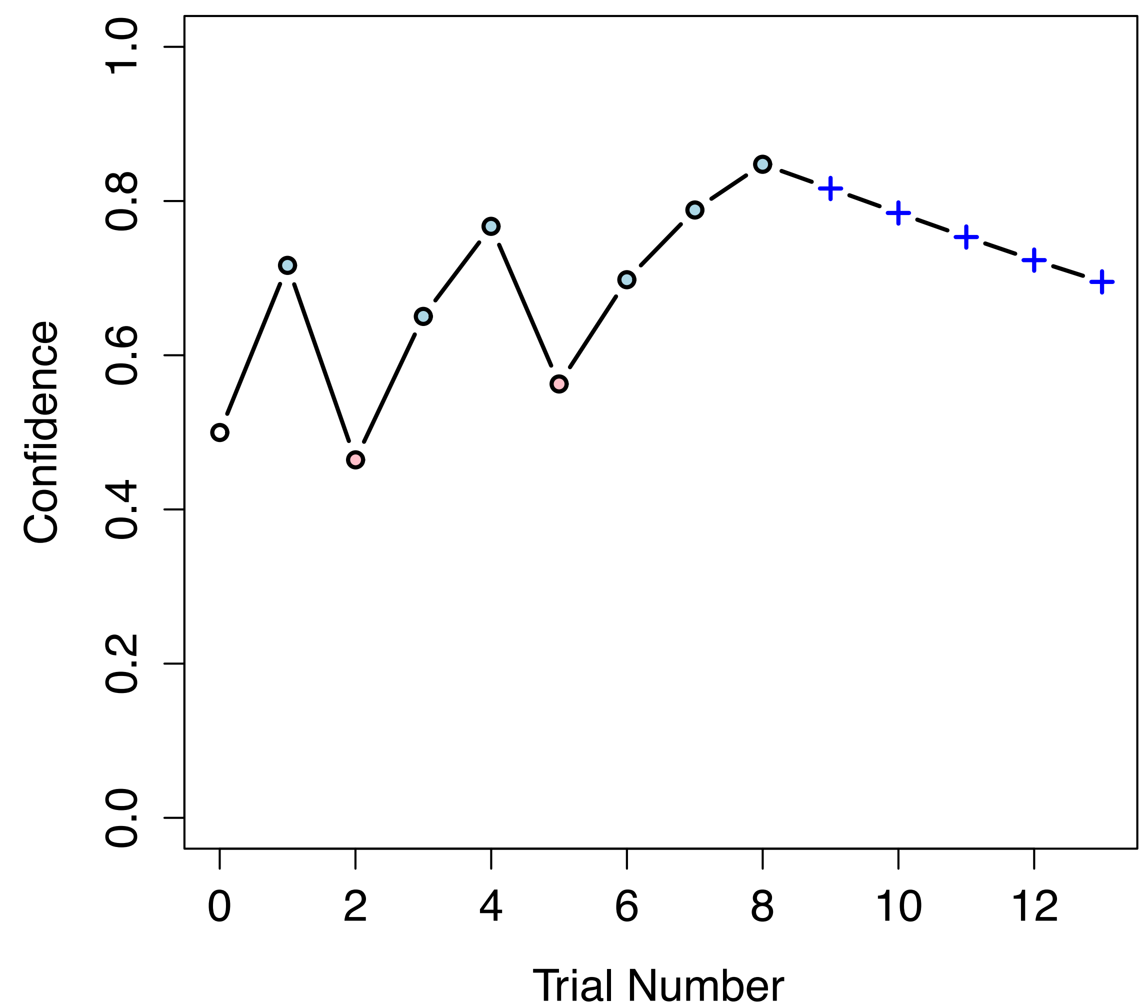
Confidence
Trial Number
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12


Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
Trial Number

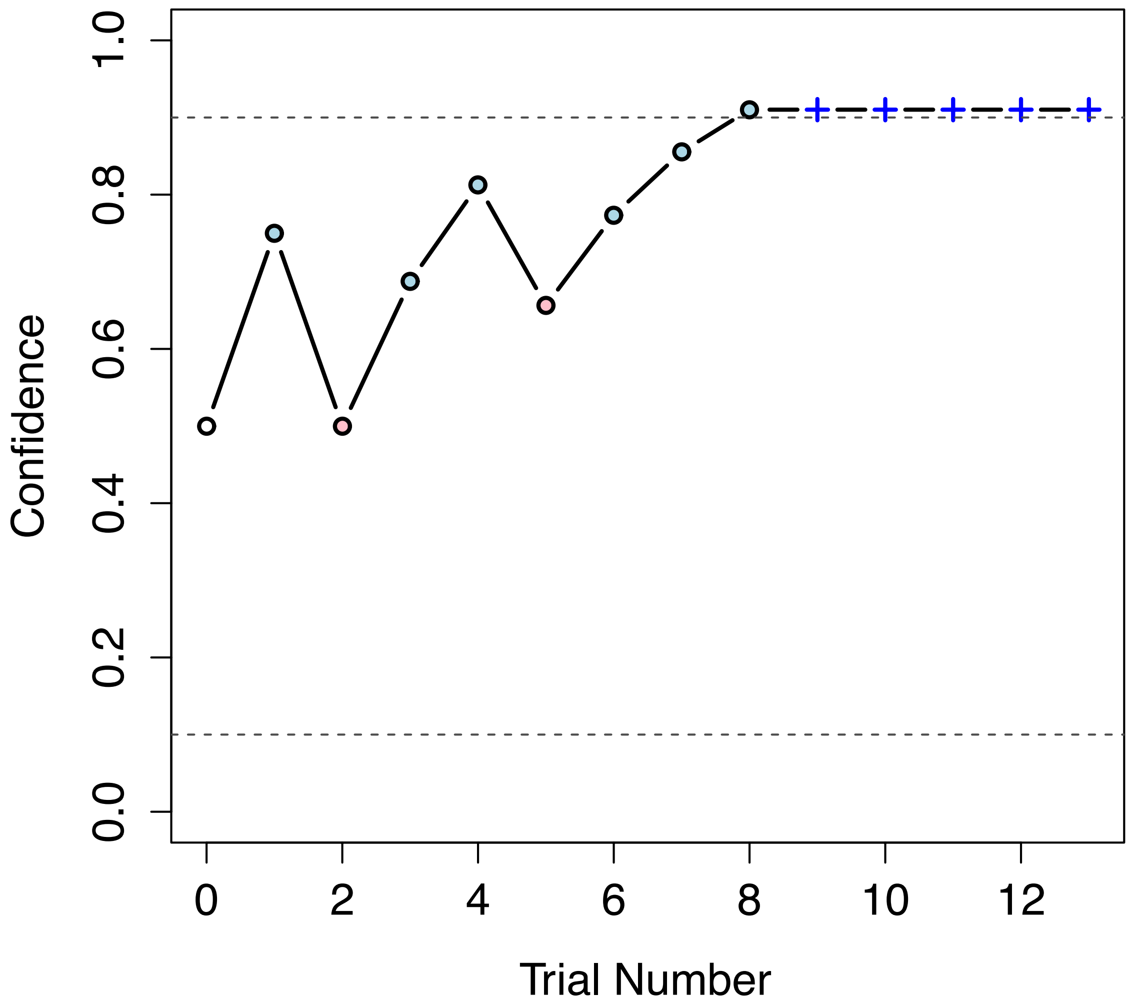
## Static world…



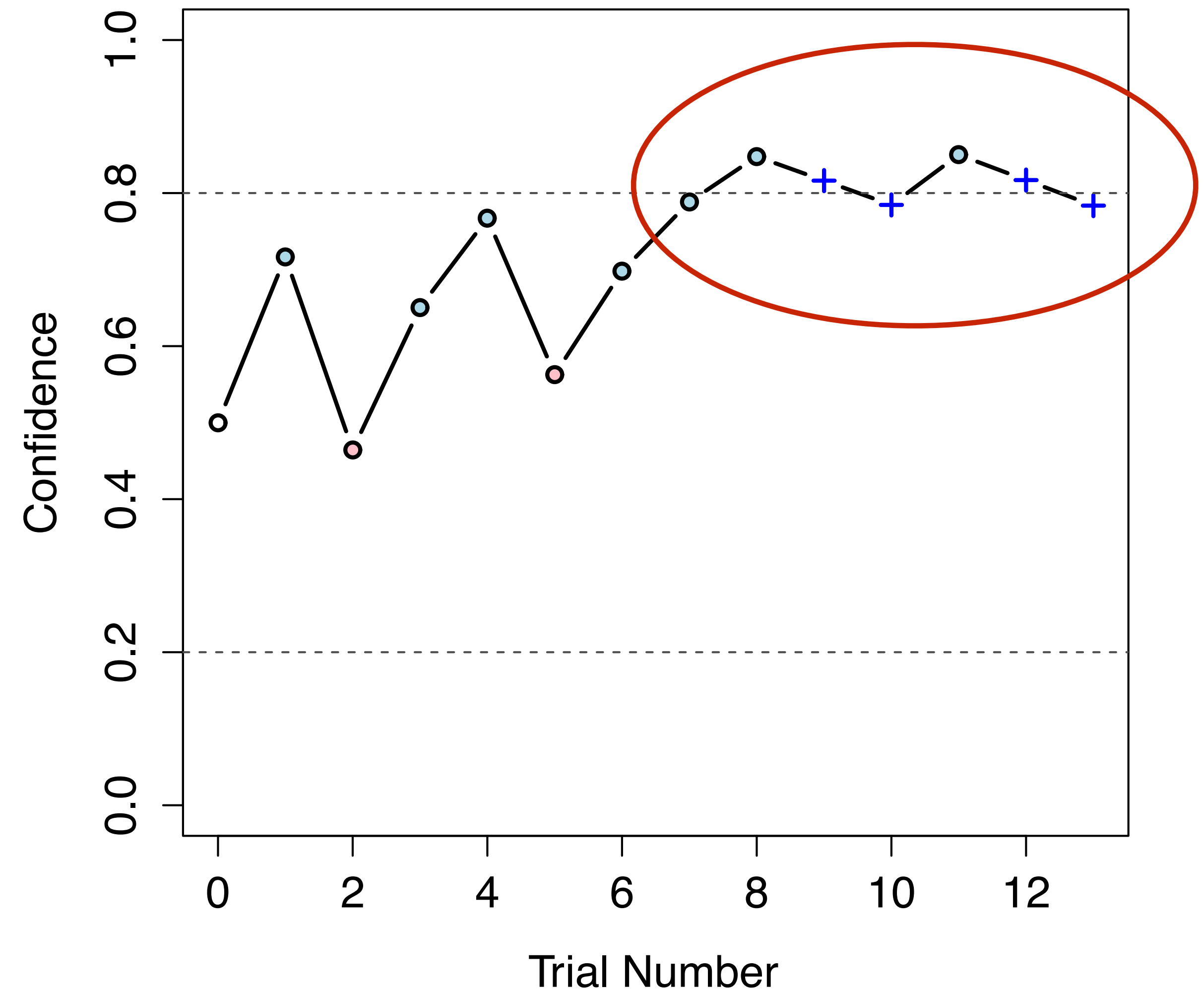

## Dynamic world…



Static world…




Dynamic world…



Human-like strategies start to seem terribly reasonable now…

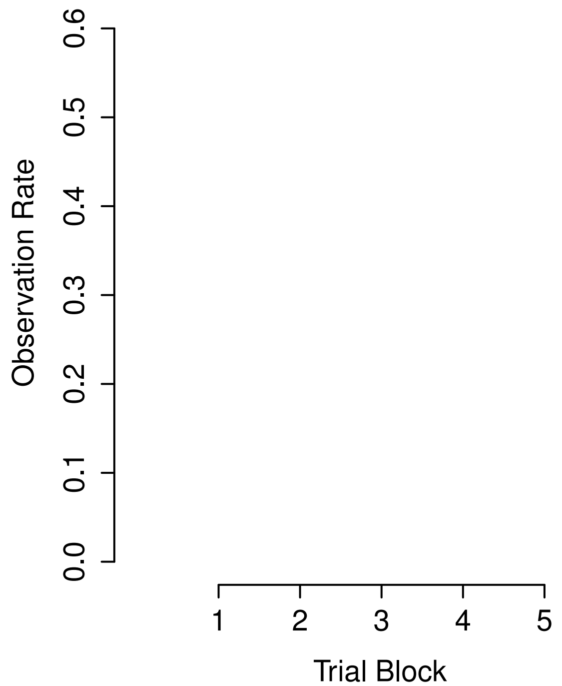
**O O O O O O O B B B B B B O O B B B B B B O B B B**

This is what a rational learner does when making choices in a changeable world

# Observe or bet in a changing world

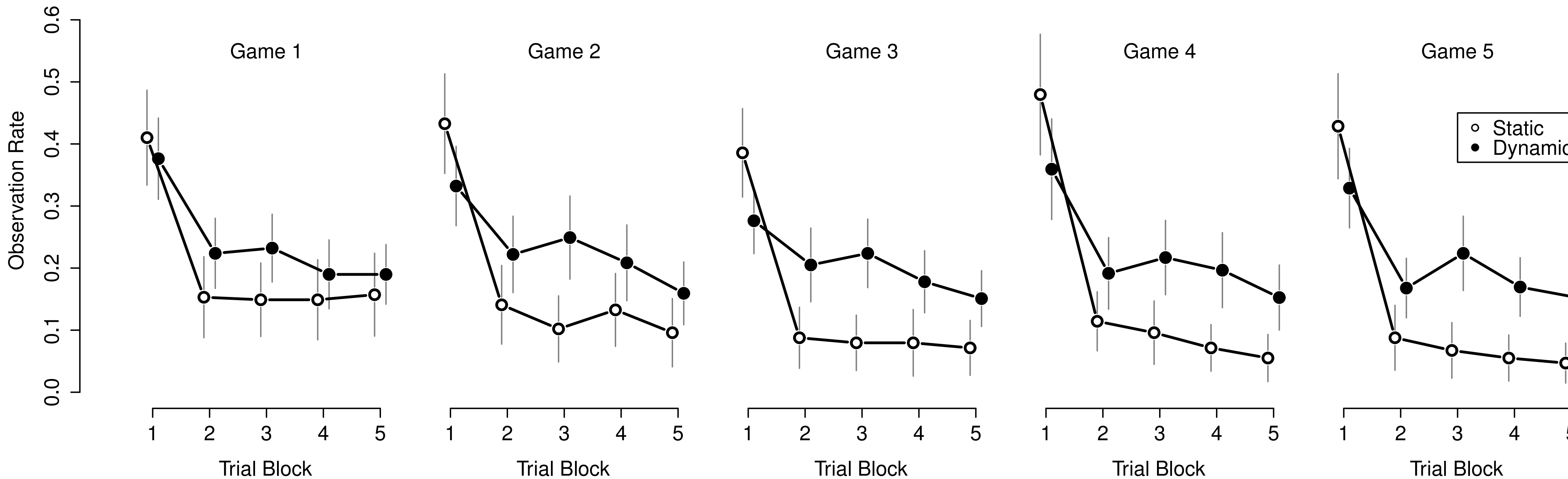
- Each person plays 5 OB tasks, 50 trials long
- Static condition: bias is always 75% towards the one option (e.g. blue)
- Dynamic condition: bias starts 75% towards one option (e.g. blue) but flips (to red) part way through the task
- Dynamic condition: participants were told that changes could happen, and to expect it to happen a few times
- Participants: 108 workers on Amazon Mechanical Turk

# Results

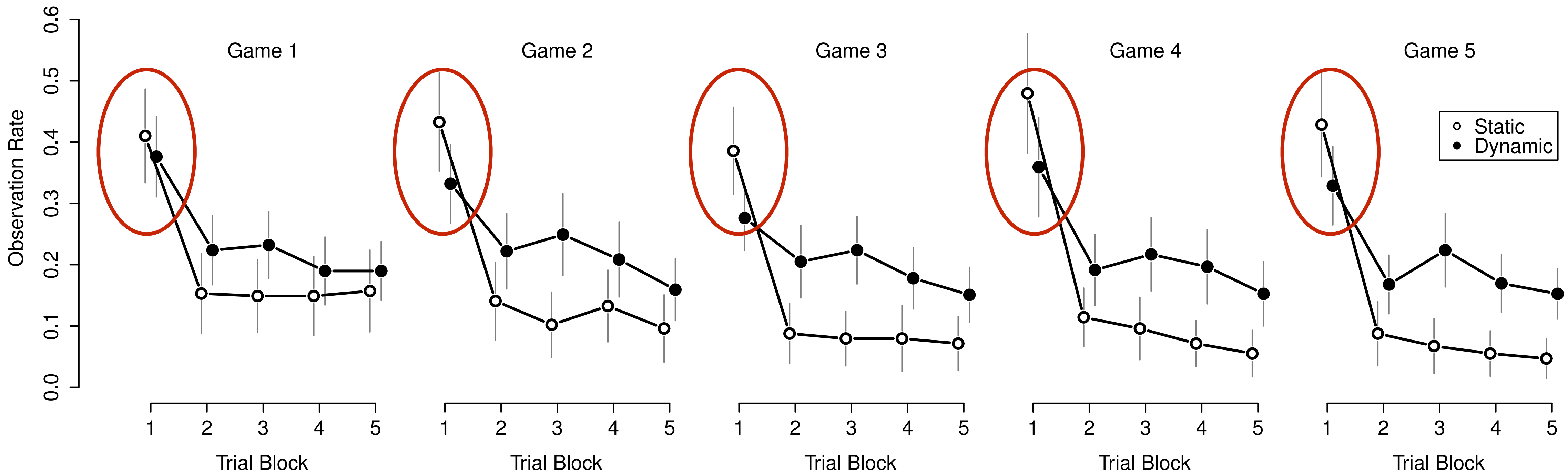



Group the trials into 5 blocks of 10 trials
For each block, plot the proportion of trials
spent on OBSERVE actions

# Results

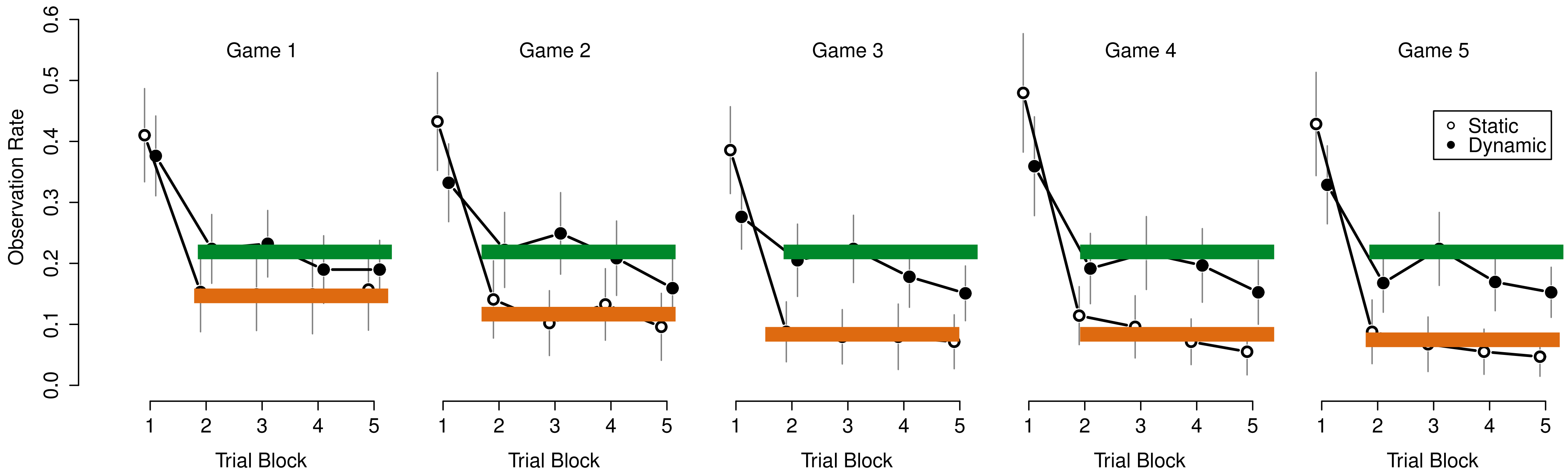

Game 1
Game 2
Game 3
Game 4
Game 5
Observation Rate
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1
2
3
4
5
Trial Block
Static
Dynamic

# Results

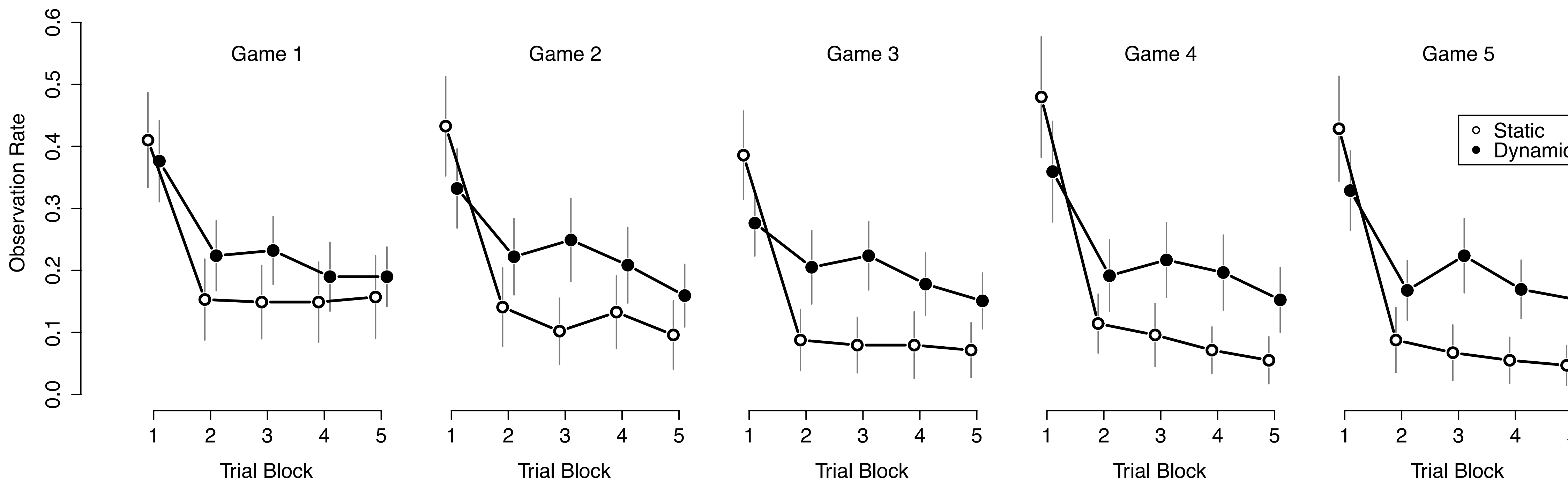


Game 1
Game 2
Game 3
Game 4
Game 5
Observation Rate
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Trial Block
1
2
3
4
5
Static
Dynamic

# Results

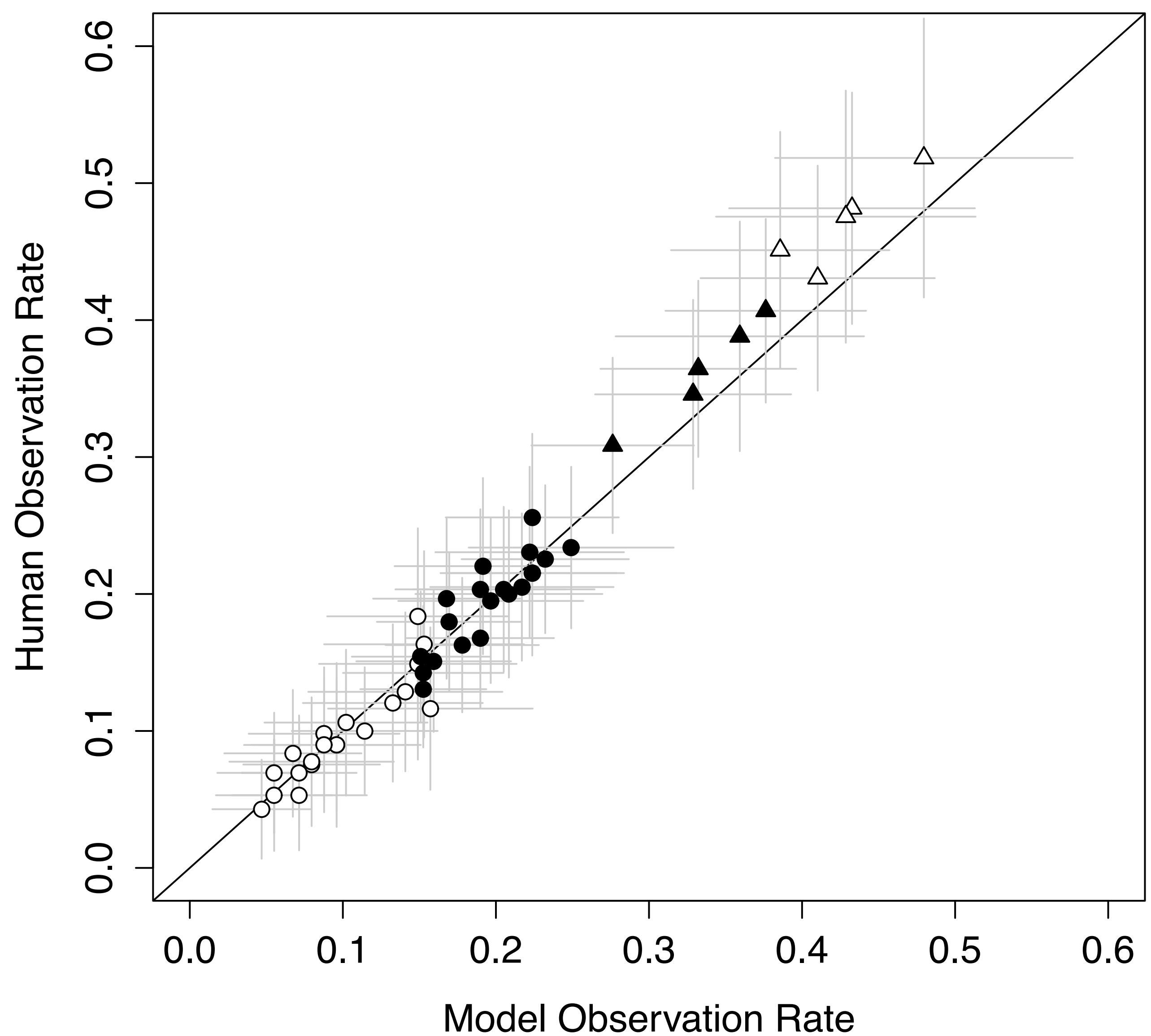

Observation Rate
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Game 1
Game 2
Game 3
Game 4
Game 5
Static
Dynamic
1
2
3
4
5
Trial Block

# Model produces human like behaviour



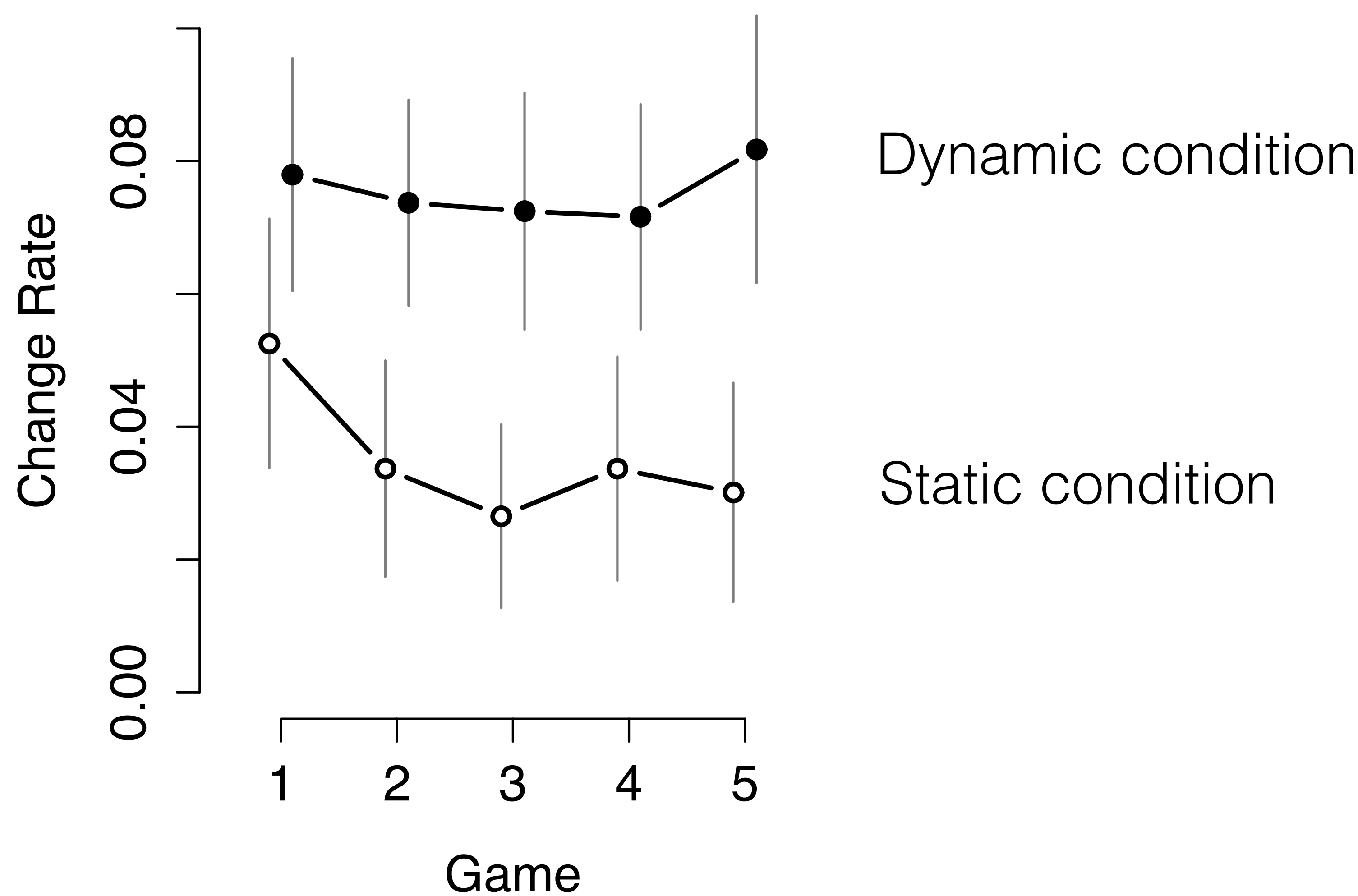

* Every game is fit separately for each person (2 parameters to describe a single OB task: a change rate and a confidence threshold)

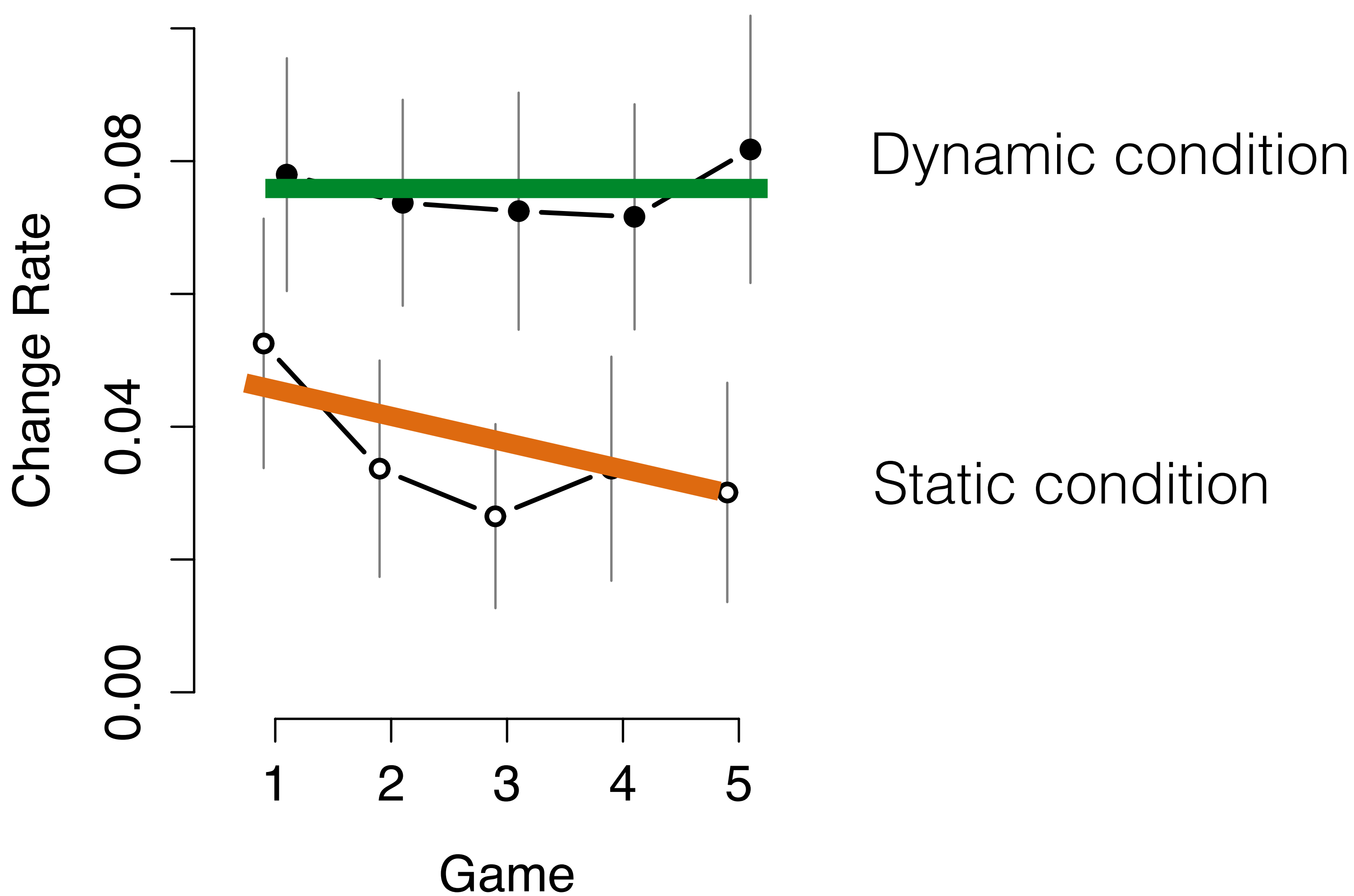
# The fits are really good*




* Meh. I'm pretty sure this model has too many free parameters. It's a useful descriptive model, but I wouldn't read too much into this just yet

# Parameter estimates are interesting



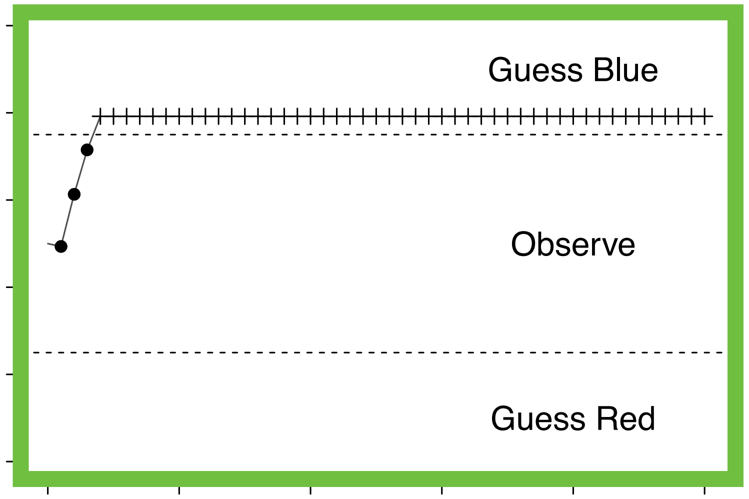

Change Rate
0.00
0.04
0.08
1
2
3
4
5
Game
Dynamic condition
Static condition

World is…

static

dynamic

static




Learner assumes…

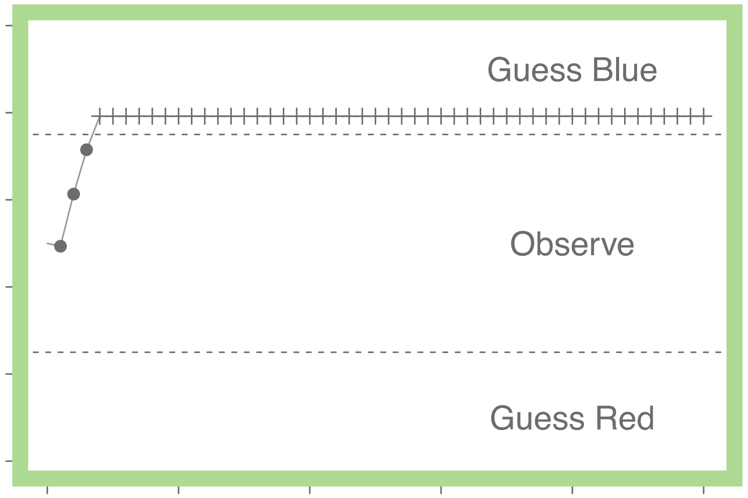
dynamic

World is…

static

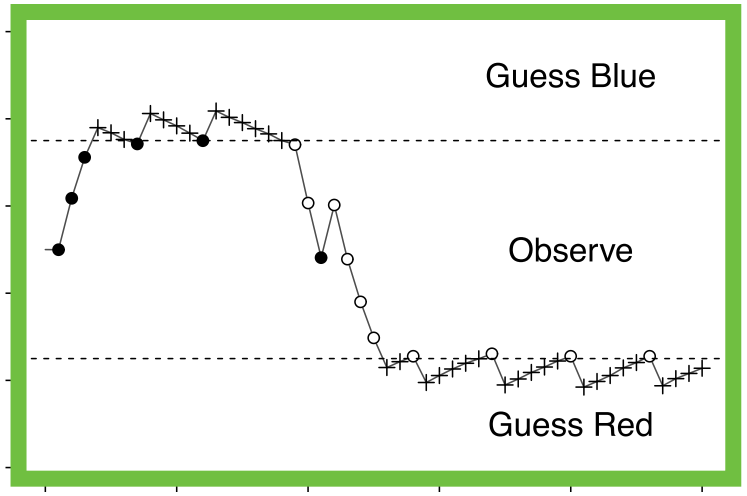
dynamic

static




Learner assumes…

dynamic



World is…

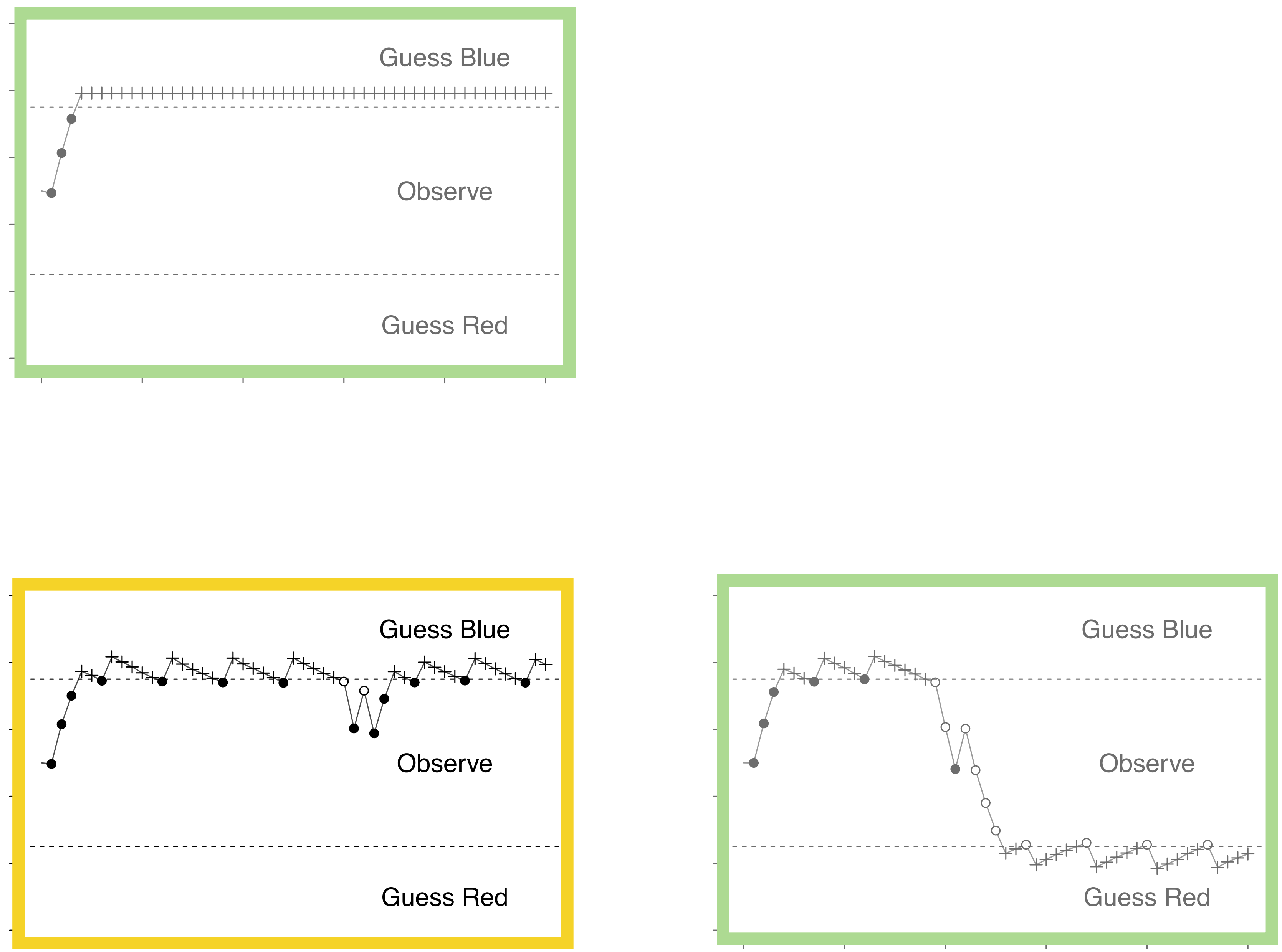
static

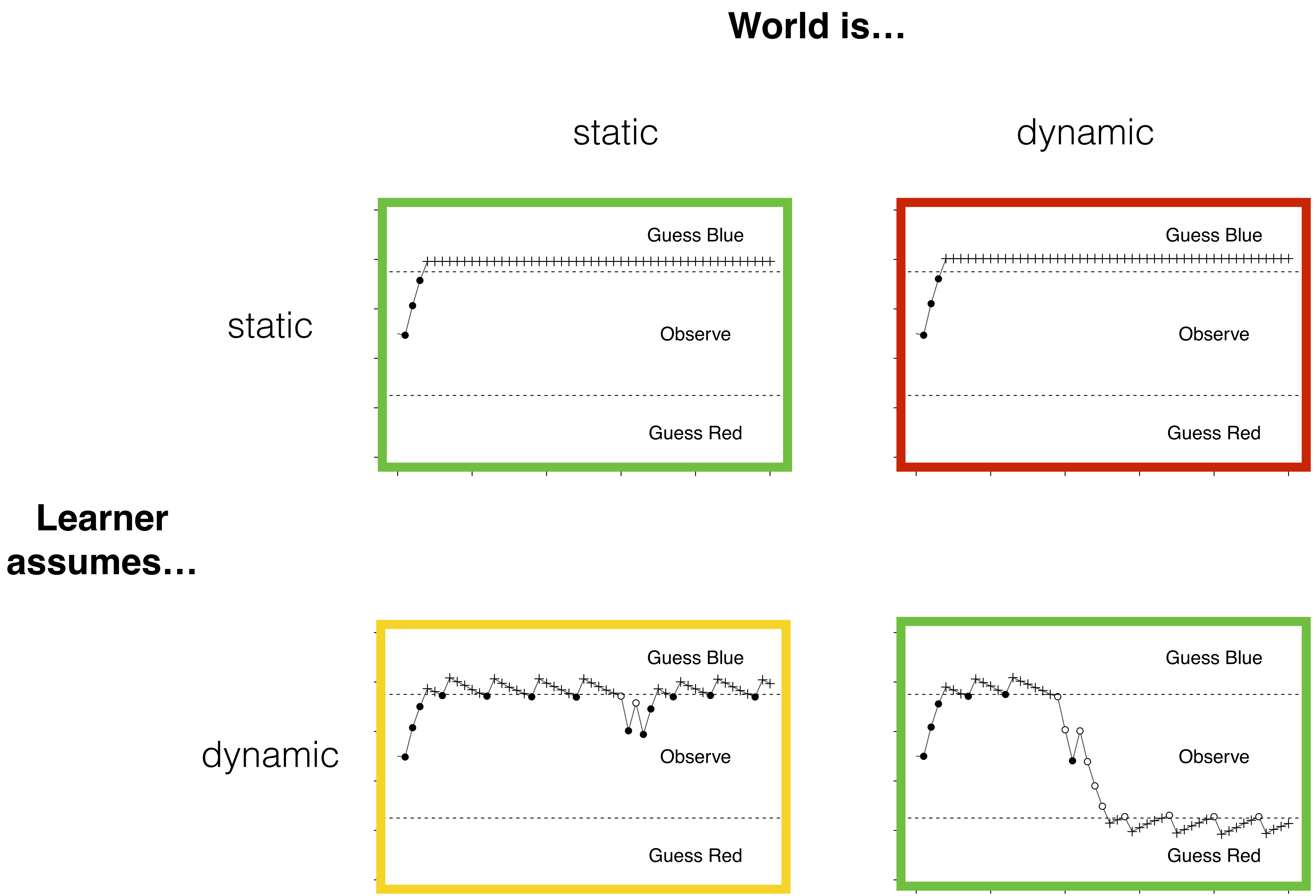
dynamic

Learner assumes…

static

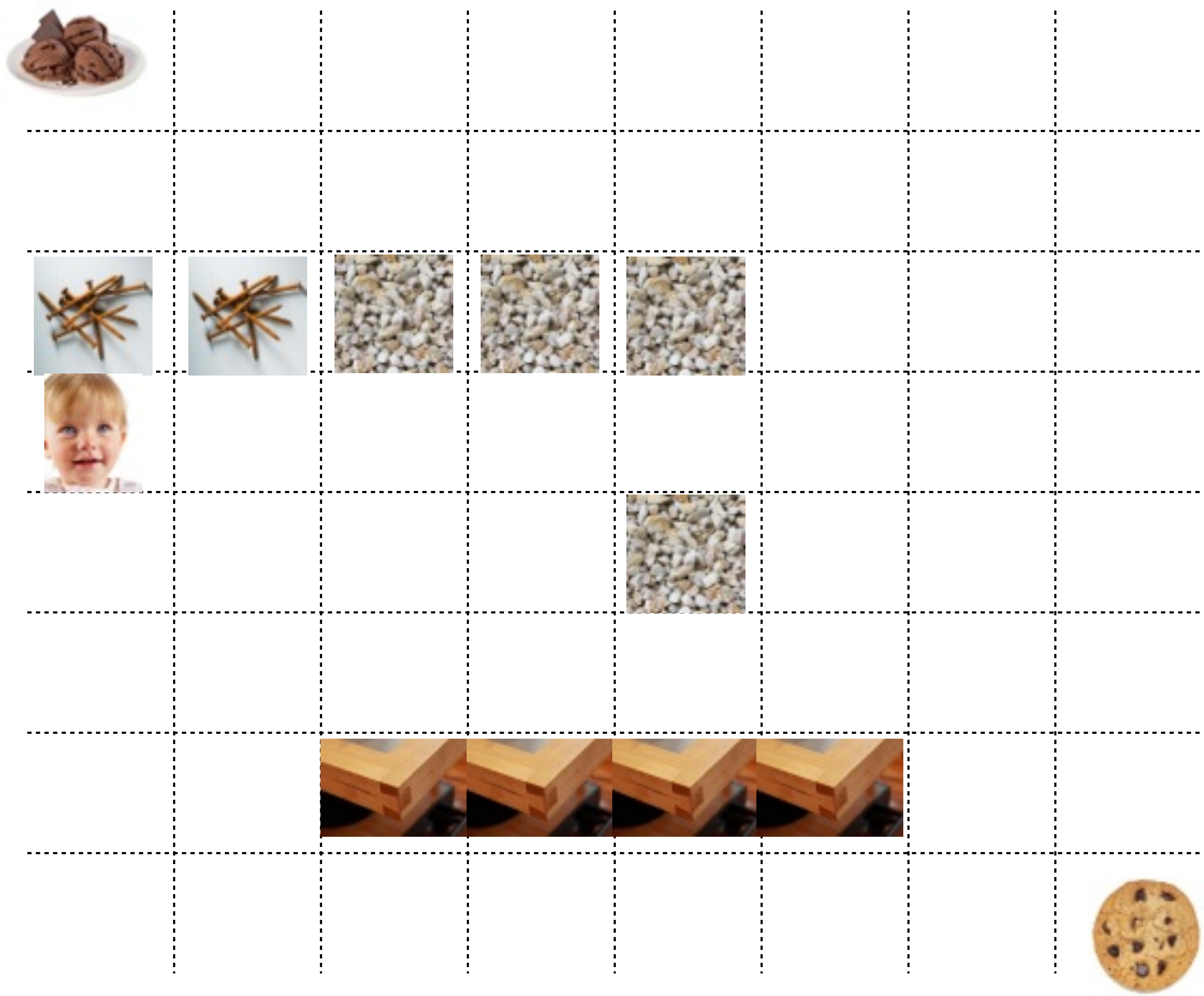
dynamic



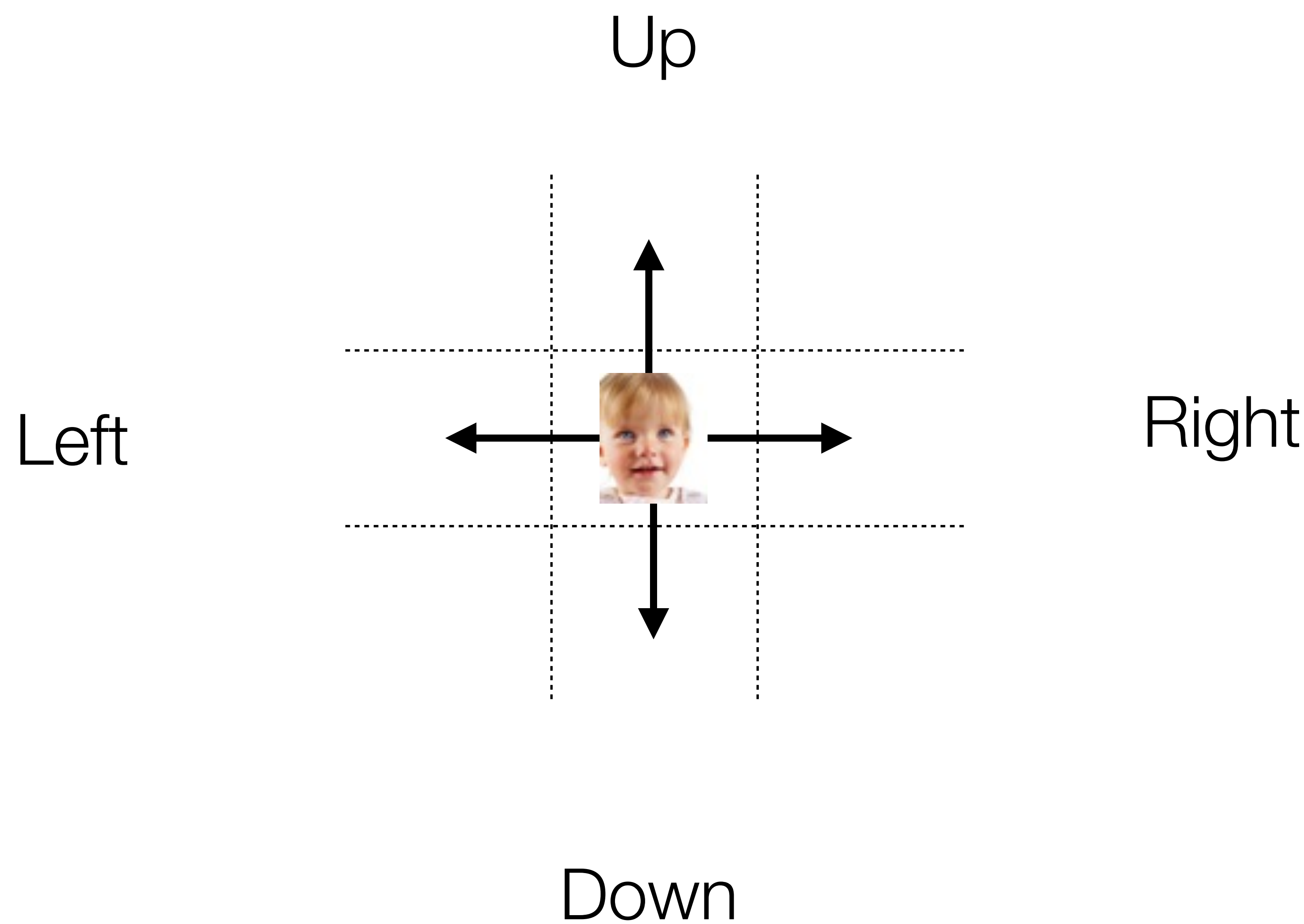

World is…
static
dynamic
Learner assumes…
static
dynamic
Guess Blue
Observe
Guess Red
Guess Blue
Observe
Guess Red
Guess Blue
Observe
Guess Red
Guess Blue
Observe
Guess Red

Back to machine learning:
Optimal decision making across a
<u>sequence</u> of decisions…

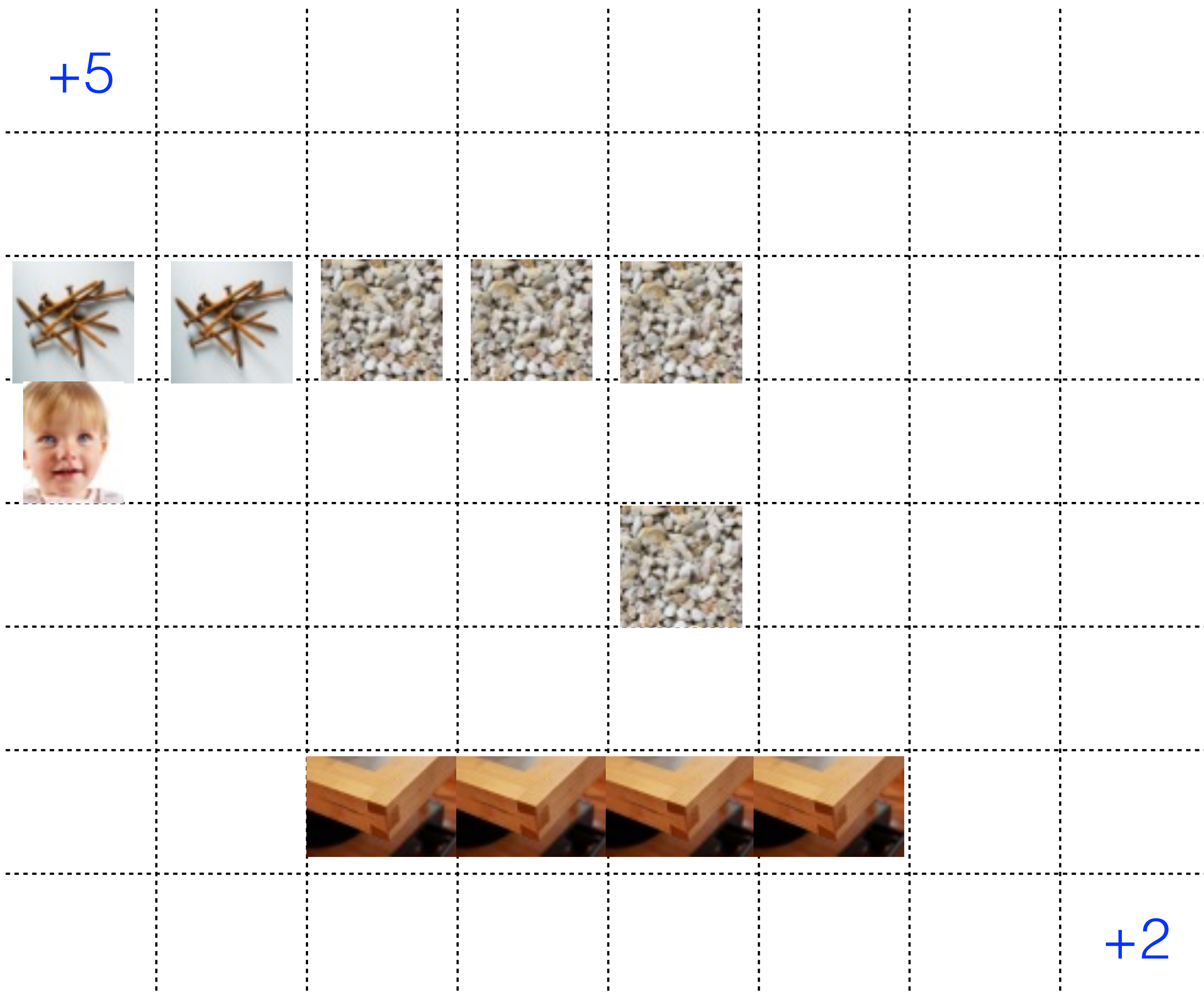
# The toddler's dilemma



# Four possible moves at each time step



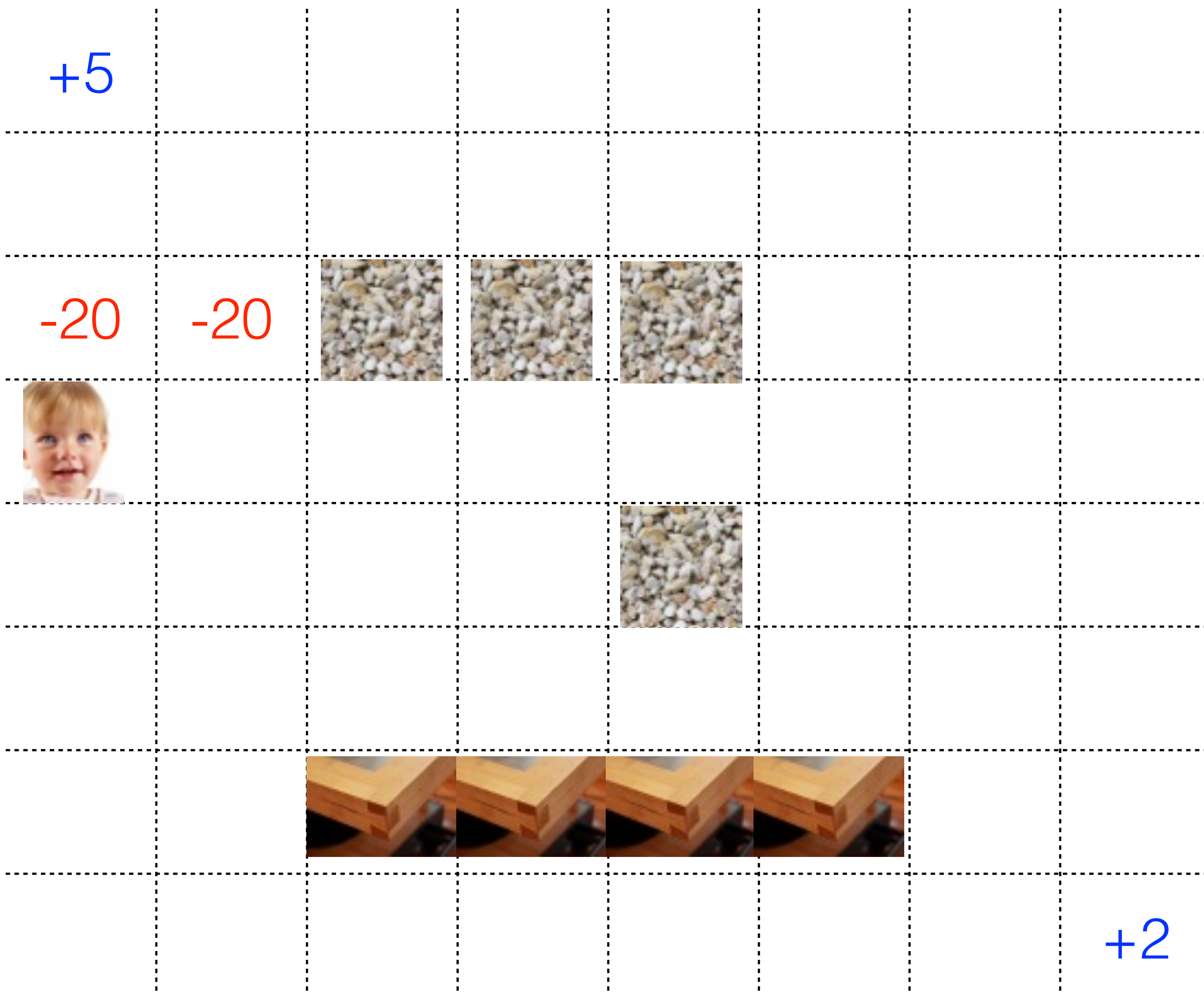
# Two possible rewards…

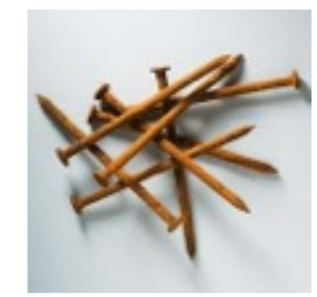



A one-off reward of +5

A one-off reward of +2

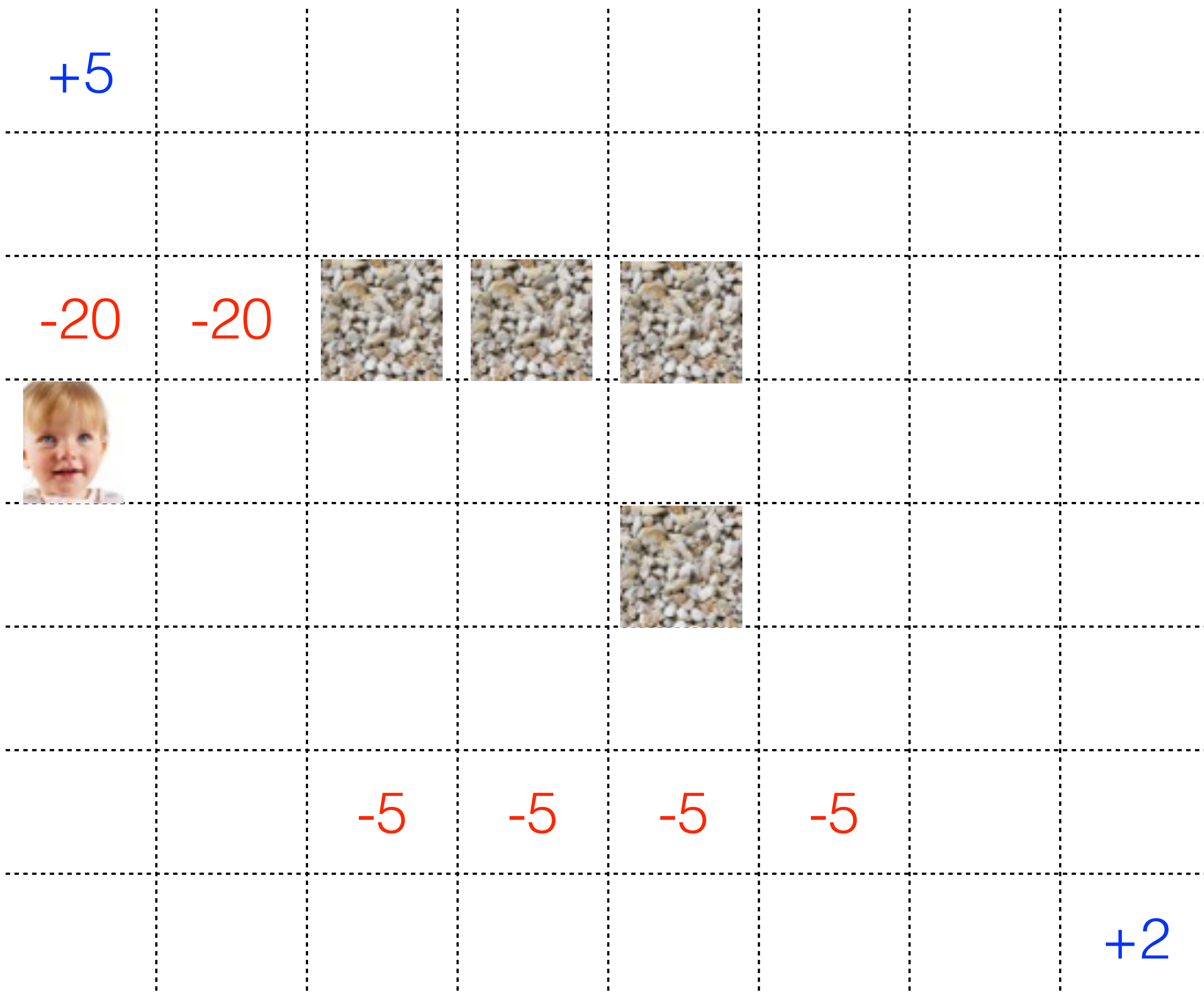
# Environmental hazards



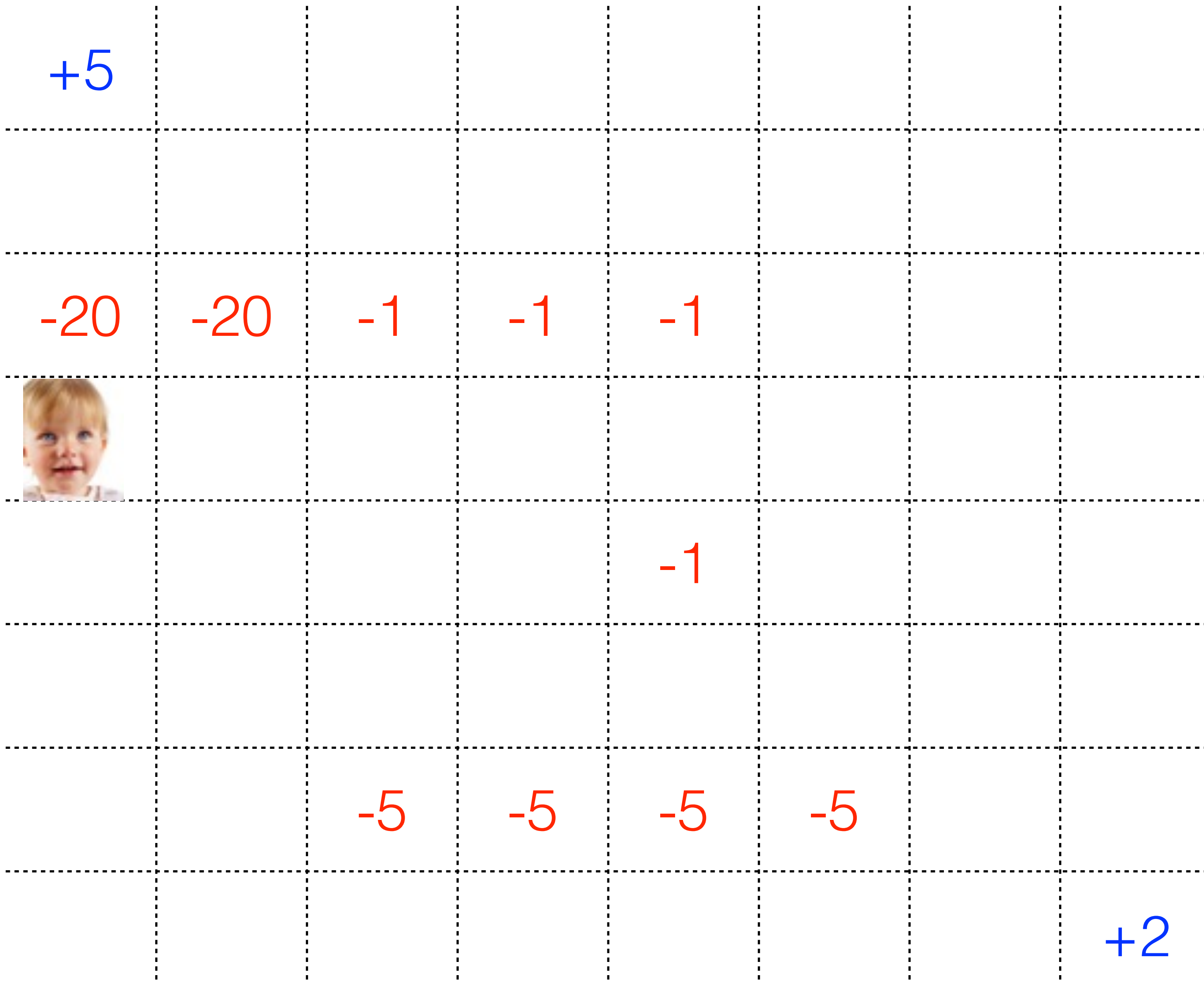
-20 penalty every time

# Environmental hazards



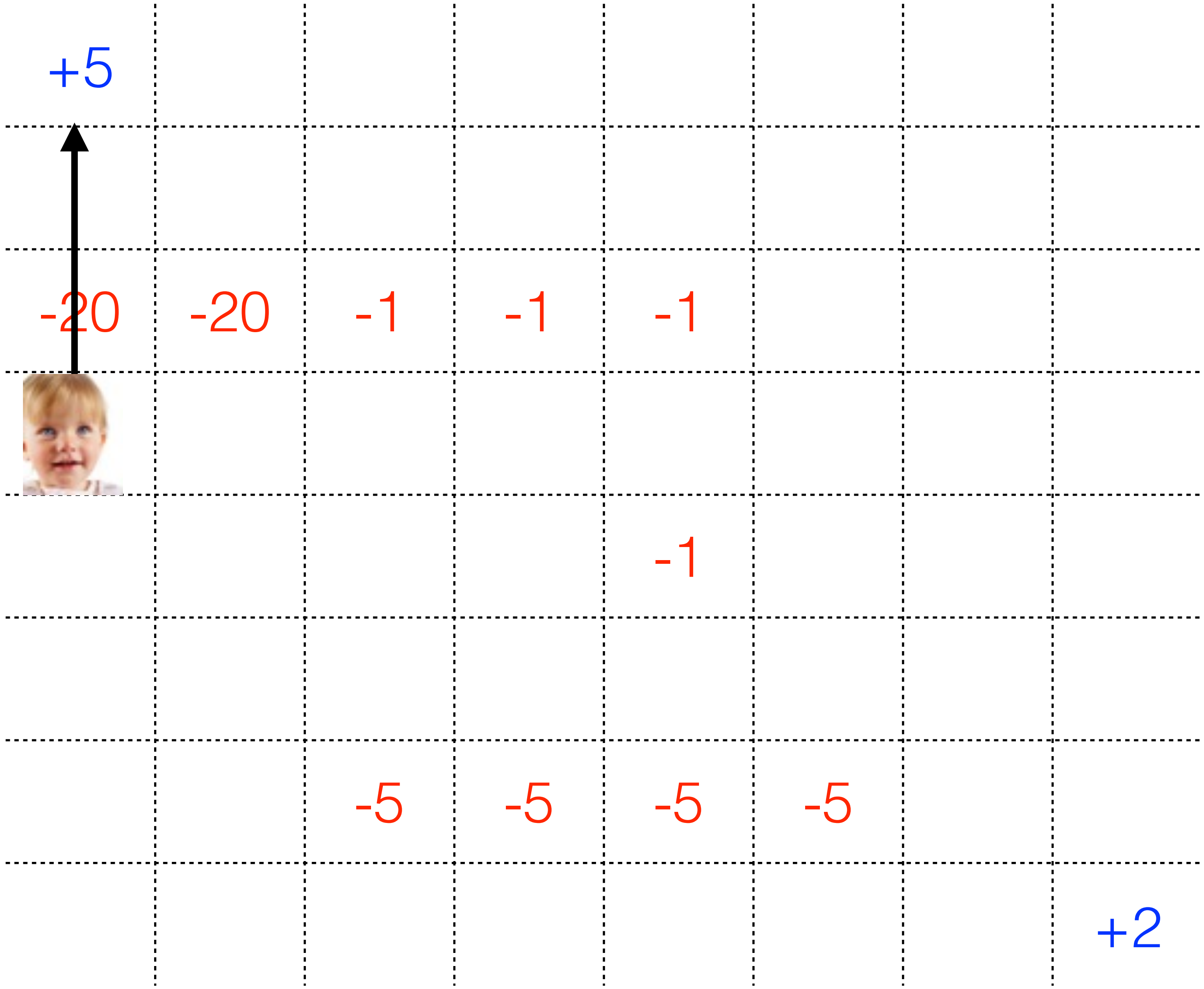

-5 penalty
every time

# Environmental hazards

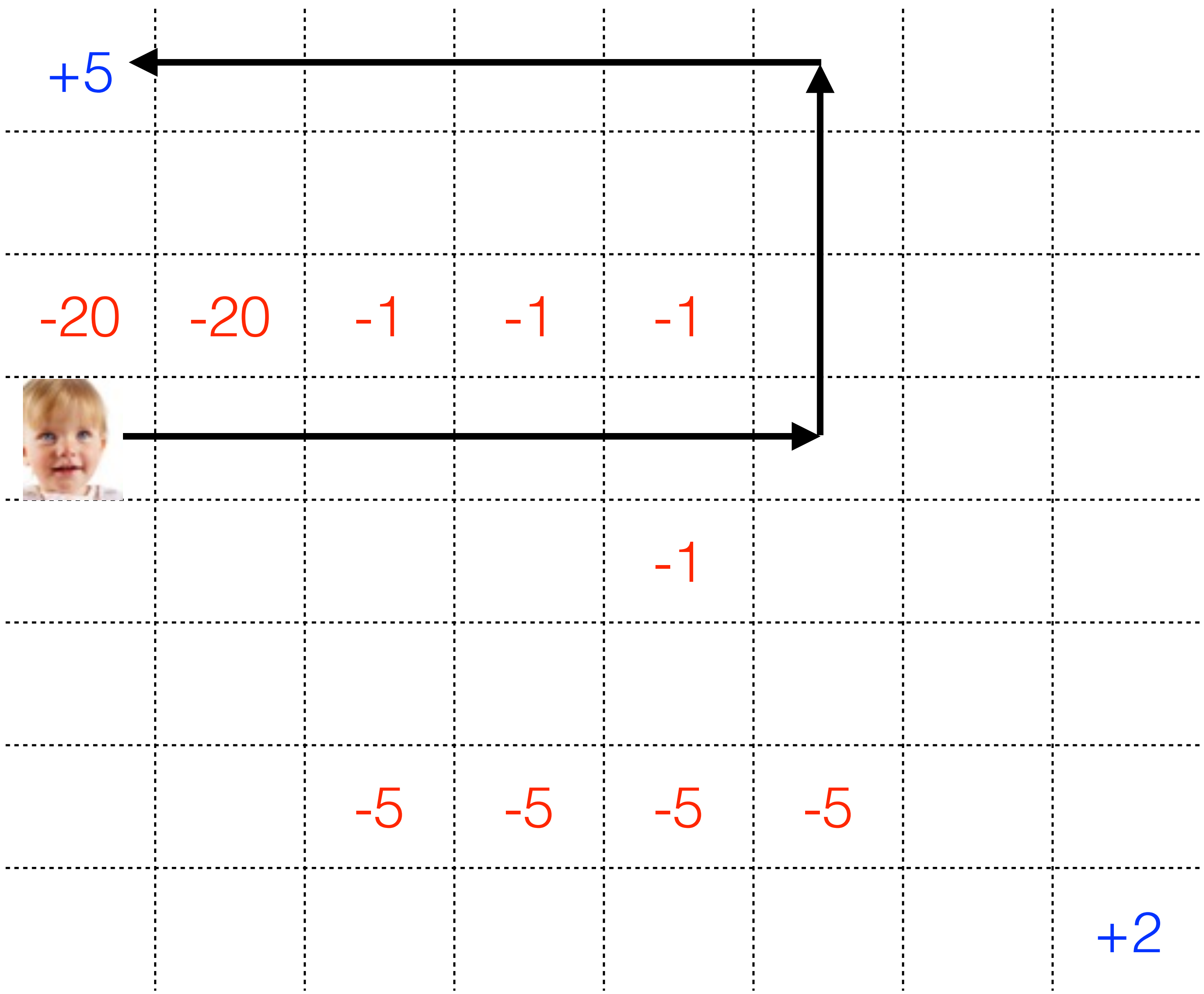



-1 penalty every time

# Shortest path is painful

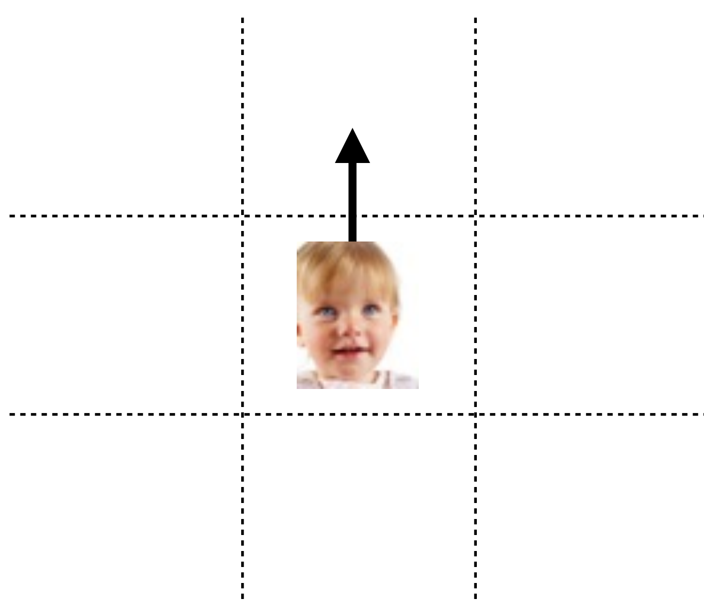


# Safest path is long

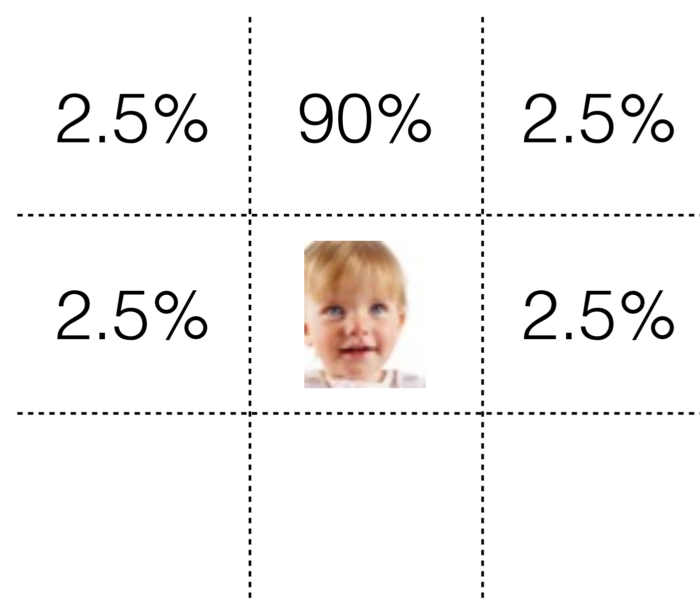


# But toddler motor control is imperfect

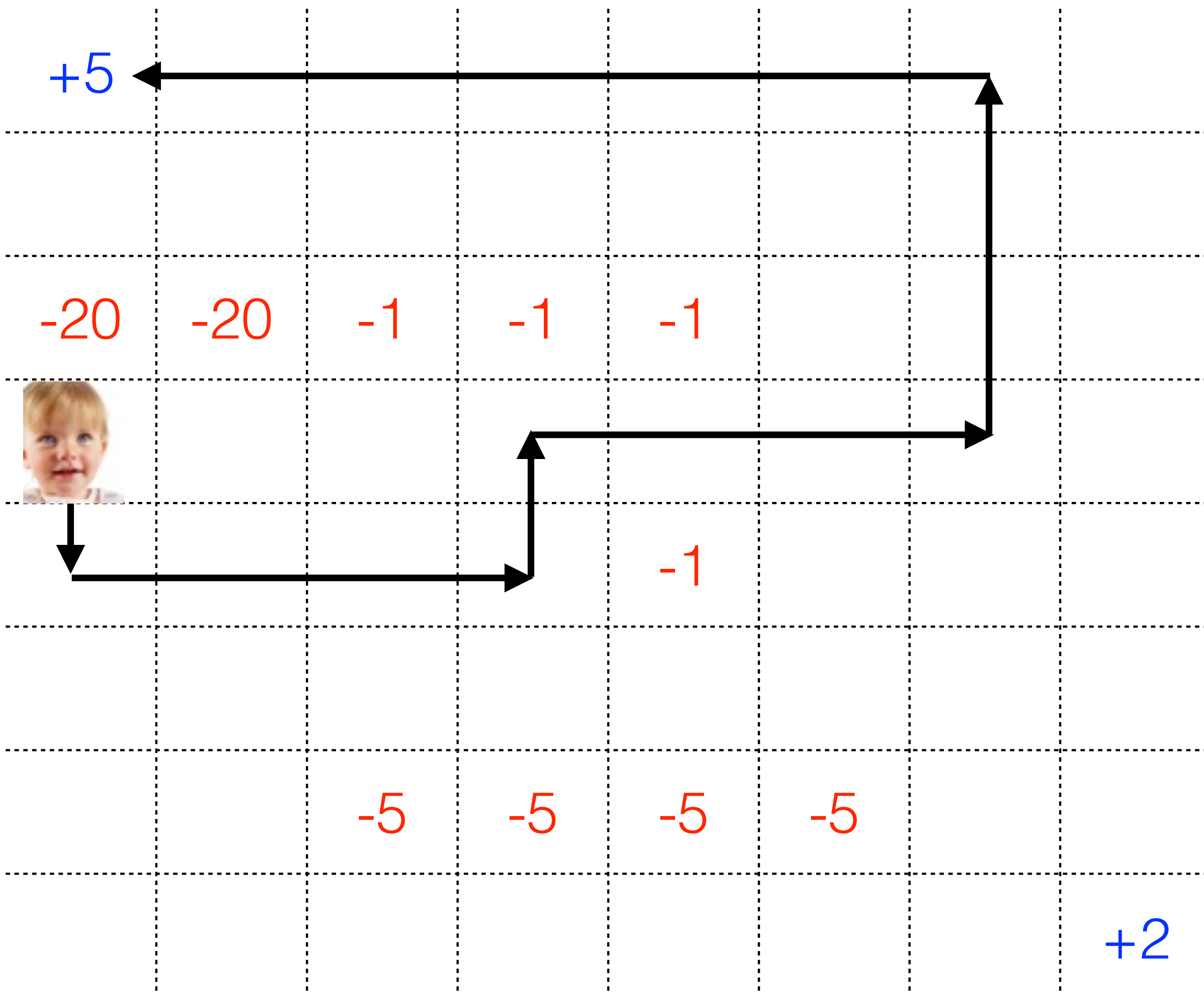
Intended action



Outcome

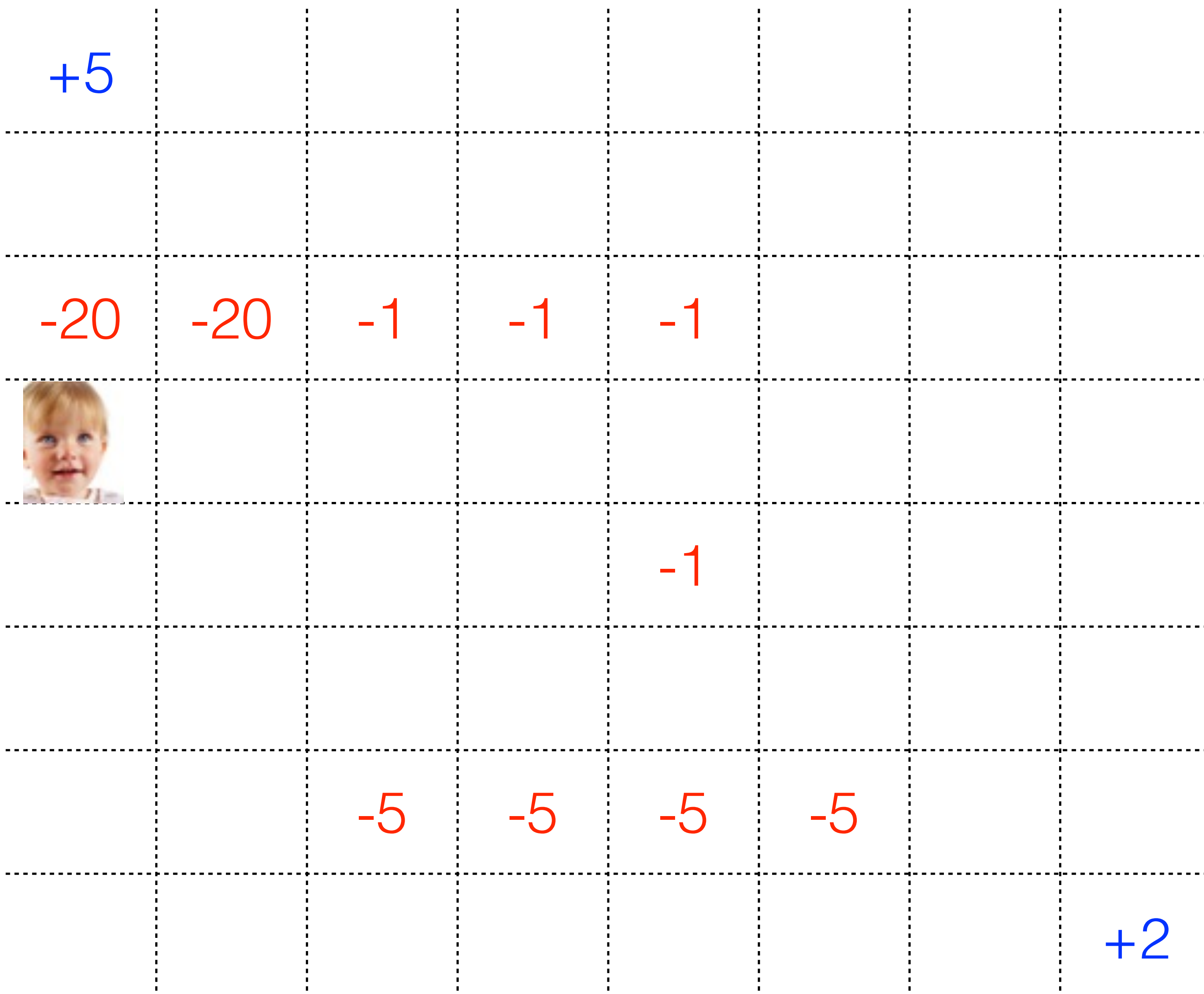


# So the safest path is <u>really</u> long




And toddlers are busy people who can't afford to wait that long!

# What should she do?

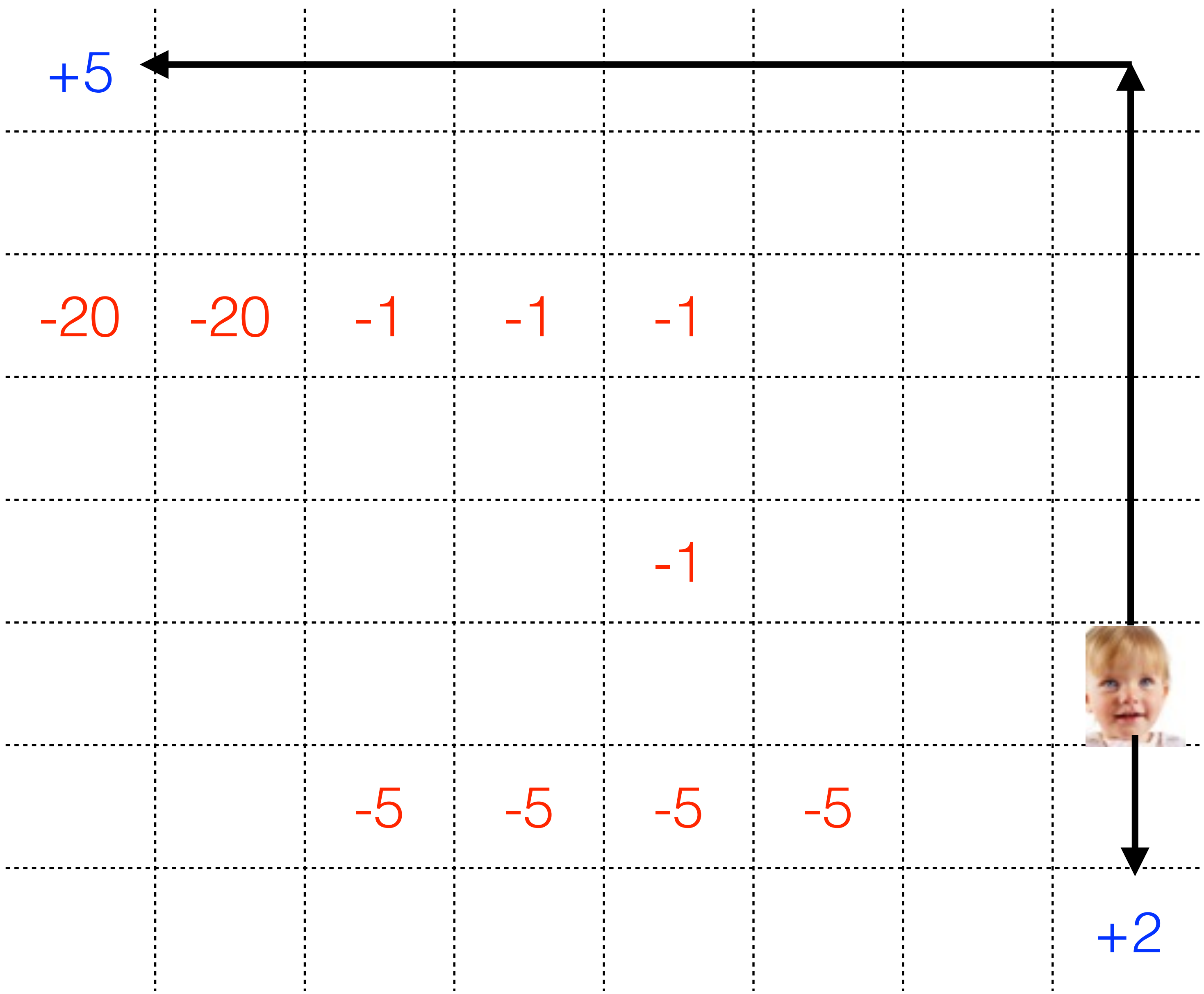


# What should she do?

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| +5 | | | | | | | |
| | | | | | | | |
| -20 | -20 | -1 | -1 | -1 | | | |
| | | | | | | | |
| | | | | -1 | | | |
| | | | | | | |  |
| | | -5 | -5 | -5 | -5 | | |
| | | | | | | | +2 |

What if she started here?

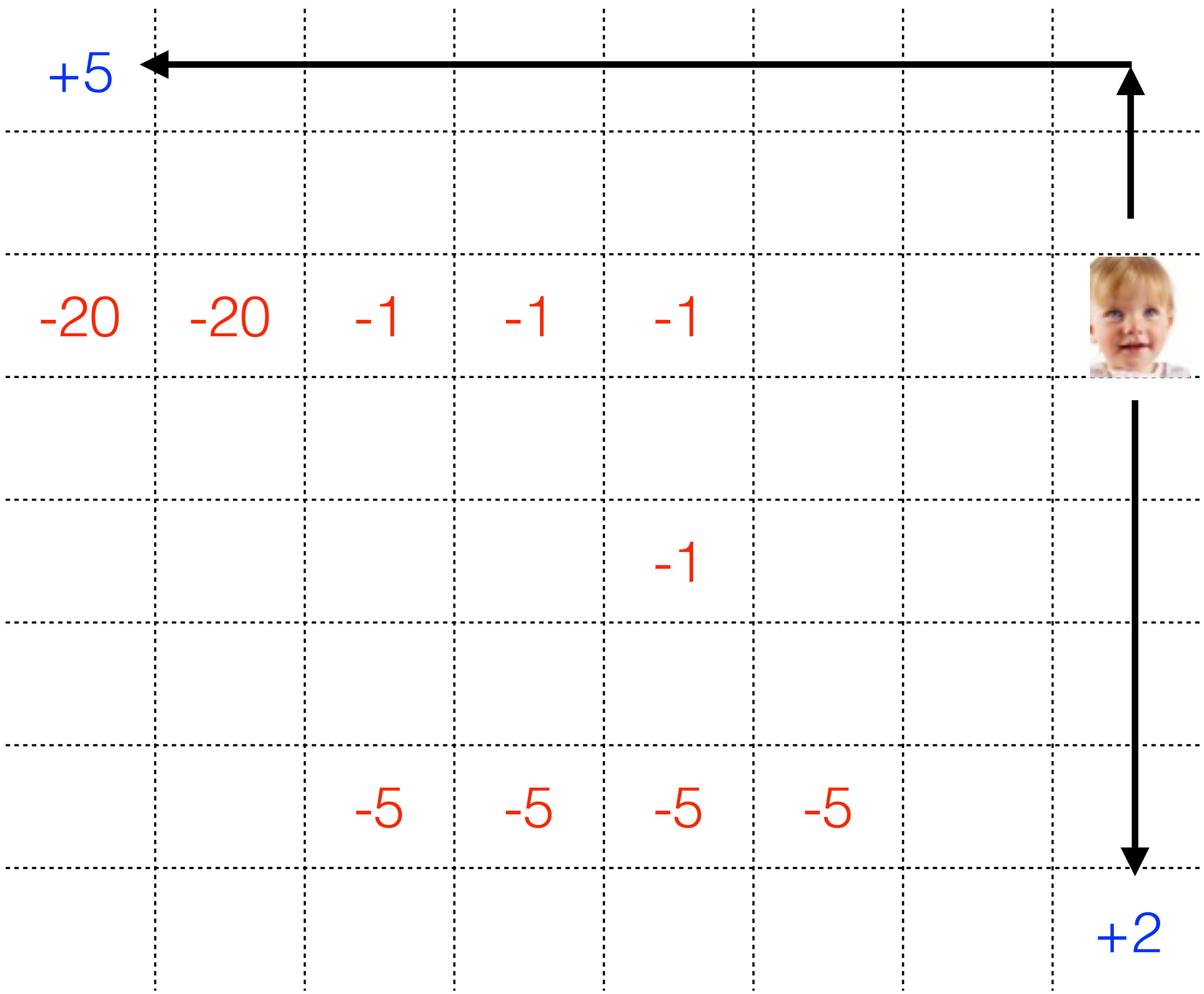
# What should she do?




Ice cream eventually?

Or cookie quickly?

# What should she do?




What about now?

# What if she were super-clumsy?

## Normal toddler

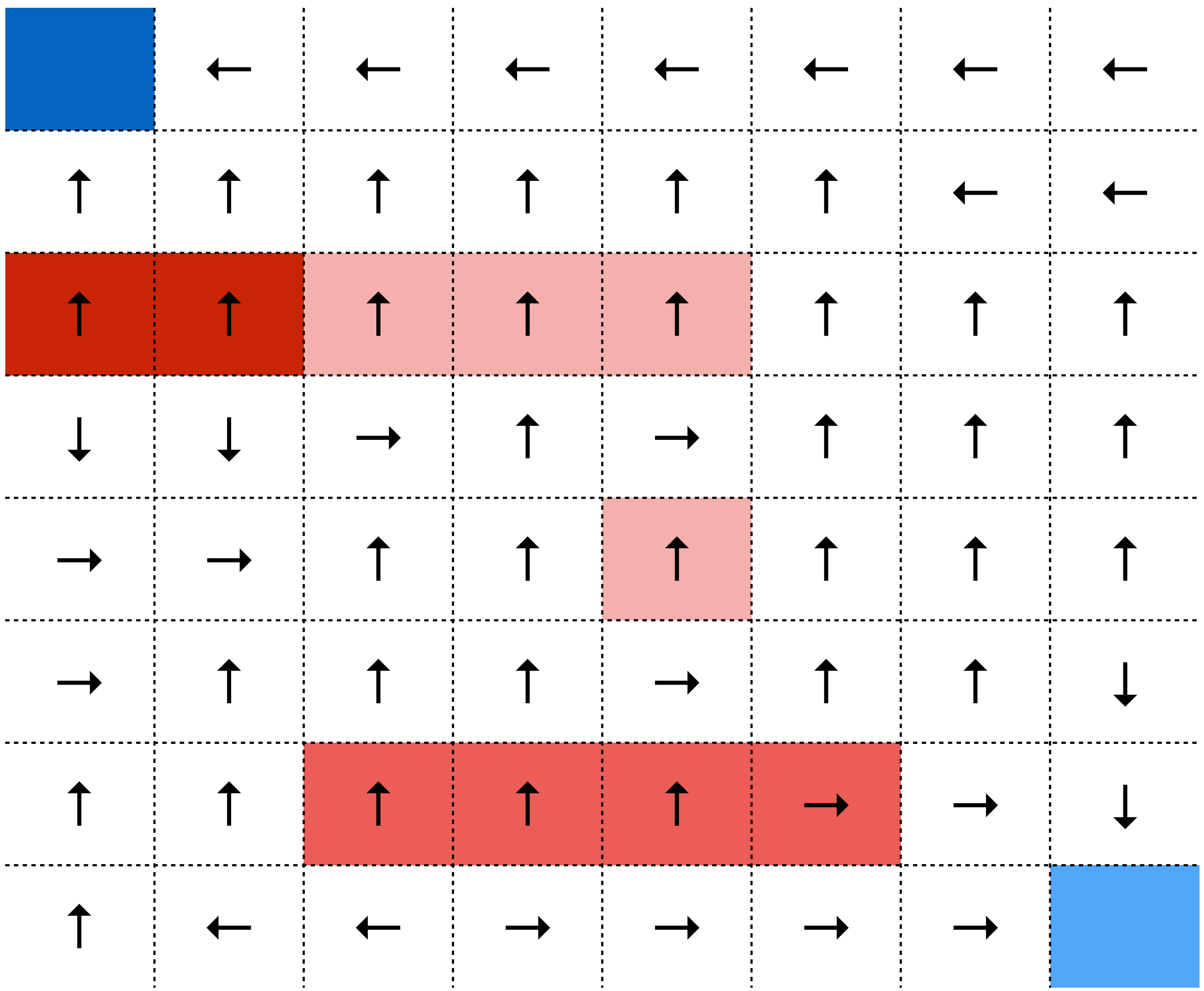
| | | |
|---|---|---|
| 2.5% | 90% | 2.5% |
| 2.5% |  | 2.5% |
| | | |

## Clumsy toddler

| | | |
|---|---|---|
| 12.5% | 50% | 12.5% |
| 12.5% |  | 12.5% |
| | | |

# Markov decision policies

# Markov decision policies



The agent needs to have a decision policy that selects actions.

Each state is associated with an action. Because the action depends only on the state that you're in, it's a Markov decision policy (MDP)

# Learning an MDP

# Bellman equations

$$u(s)$$

Utility of
state s

# Bellman equations

$$u(s) = r(s)$$

Utility of state s

Reward obtained from being in state s

Expected utility of future rewards

# Bellman equations

$$u(s) = r(s) +$$ Expected utility of future rewards

Utility of state s

Reward obtained from being in state s

# Bellman equations

$$u(s) = r(s) + \gamma$$

Expected utility of
future rewards

Utility of
state s

Reward
obtained from
being in state s

Rewards in the future
aren't as valuable as
rewards now, so we apply
a "discount" factor (i.e.,
number less than 1) to
devalue the future slightly

# Bellman equations

$$u(s) = r(s) + \gamma \max_a \sum_{s'} u(s')P(s'|a,s)$$

This is the equation for the expected utility of action a

# Bellman equations

$$u(s) = r(s) + \gamma \max_a \sum_{s'} u(s')P(s'|a,s)$$

Utility of
state s’

# Bellman equations

$$u(s) = r(s) + \gamma \max_a \sum_{s'} u(s')P(s'|a,s)$$

Utility of state s'

Probability that you'll be in state s' next, given that you're currently in state s and you make action a

# Bellman equations

$$u(s) = r(s) + \gamma \max_a \sum_{s'} u(s')P(s'|a,s)$$

Summed over all possible states that you might end up in

Utility of state s'

Probability that you'll be in state s' next, given that you're currently in state s and you make action a

# Bellman equations

$$u(s) = r(s) + \gamma \max_a \sum_{s'} u(s') P(s'|a, s)$$

Agent is assumed to select the best possible action next time

Summed over all possible states that you might end up in

Utility of state s’

Probability that you’ll be in state s’ next, given that you’re currently in state s and you make action a

# Update the utilities…

$$u_{i+1}(s) \leftarrow r(s) + \gamma \max_a \sum_{s'} u_i(s')P(s'|a,s)$$

Updated utility on iteration i+1 of the algorithm

The utility assigned to state s at iteration i of the algorithm

```
Initialise all utilities u(s) = r(s)
Loop until utilities do not change:
   Perform a “Bellman update” to the utilities

Decision policy is:
   Choose the action with highest expected utility!
```

Demonstration code: mdp.R

# Treating the observe or bet task as an MDP problem

Optimal policy must satisfy Bellman's equation over **belief states**:

$$U(\mathbf{b}) = R(\mathbf{b}) + \gamma \max_a \sum_{\mathbf{b}'} P(\mathbf{b}'|a, \mathbf{b}) U(\mathbf{b}')$$

Each belief state **b** corresponds to a distribution over possible world states. The posterior $P(\theta|\mathbf{x})$ is the belief.

Reward expected now given current beliefs about the world

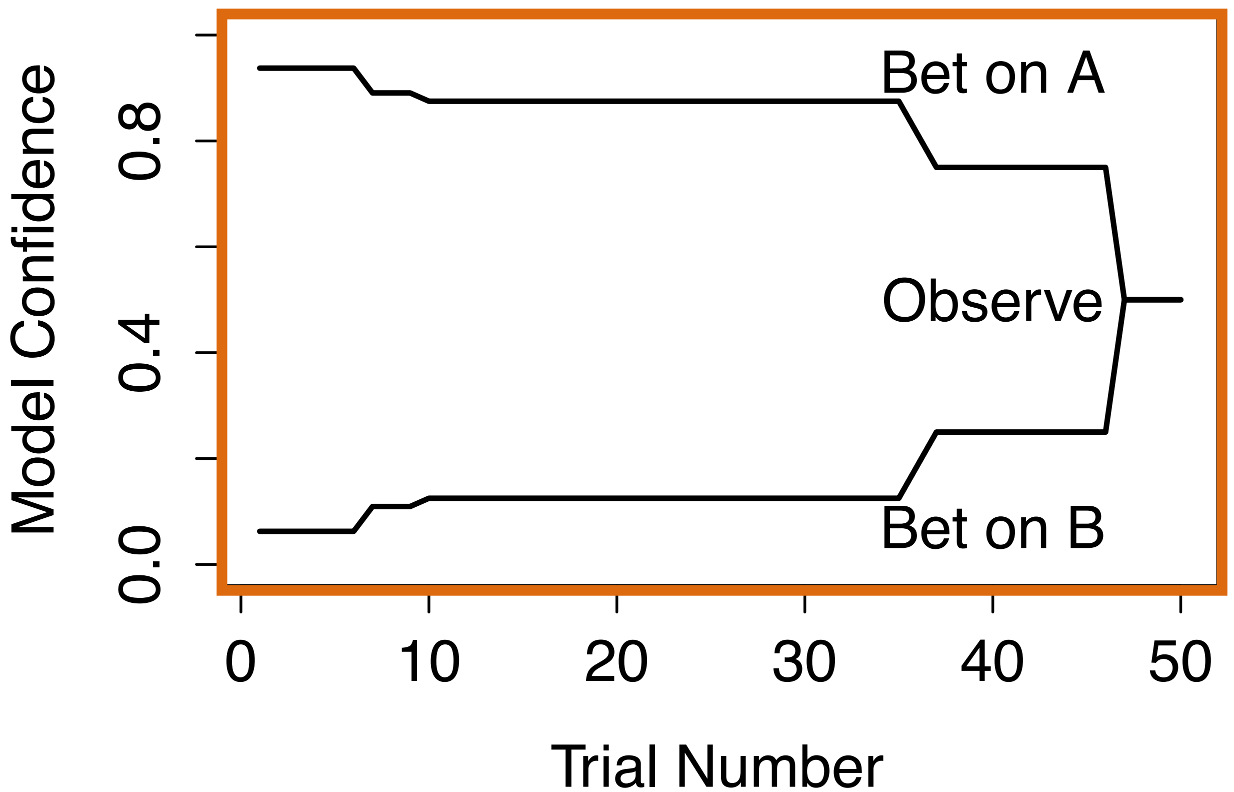
$$U(\mathbf{b}) = R(\mathbf{b}) + \gamma \max_a \sum_{\mathbf{b}'} P(\mathbf{b}'|a, \mathbf{b}) U(\mathbf{b}')$$

Utility assigned to future rewards, temporally discounted and dependent on continuing to use the optimal policy

Space of beliefs is high dimensional, but the observe or bet task is simple enough that value iteration works:

$$U(\mathbf{b}) \leftarrow R(\mathbf{b}) + \gamma \max_a \sum_{\mathbf{b}'} P(\mathbf{b}'|a, \mathbf{b}) U(\mathbf{b}')$$

# The answer.


Model Confidence
0.0
0.4
0.8
Bet on A
Observe
Bet on B
0
10
20
30
40
50
Trial Number