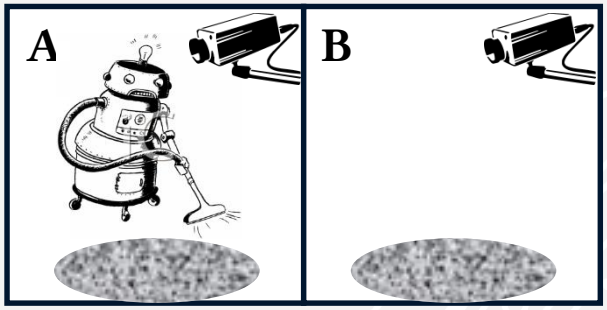
Markov Models and Reinforcement Learning

Stephen G. Ware

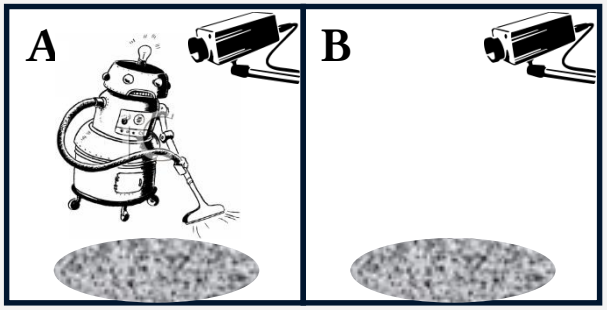
CSCI 4525 / 5525

Camera Vacuum World (CVW)

- 2 discrete rooms with cameras that detect dirt.
- A mobile robot with a vacuum.
- The goal is to ensure both rooms are clean.

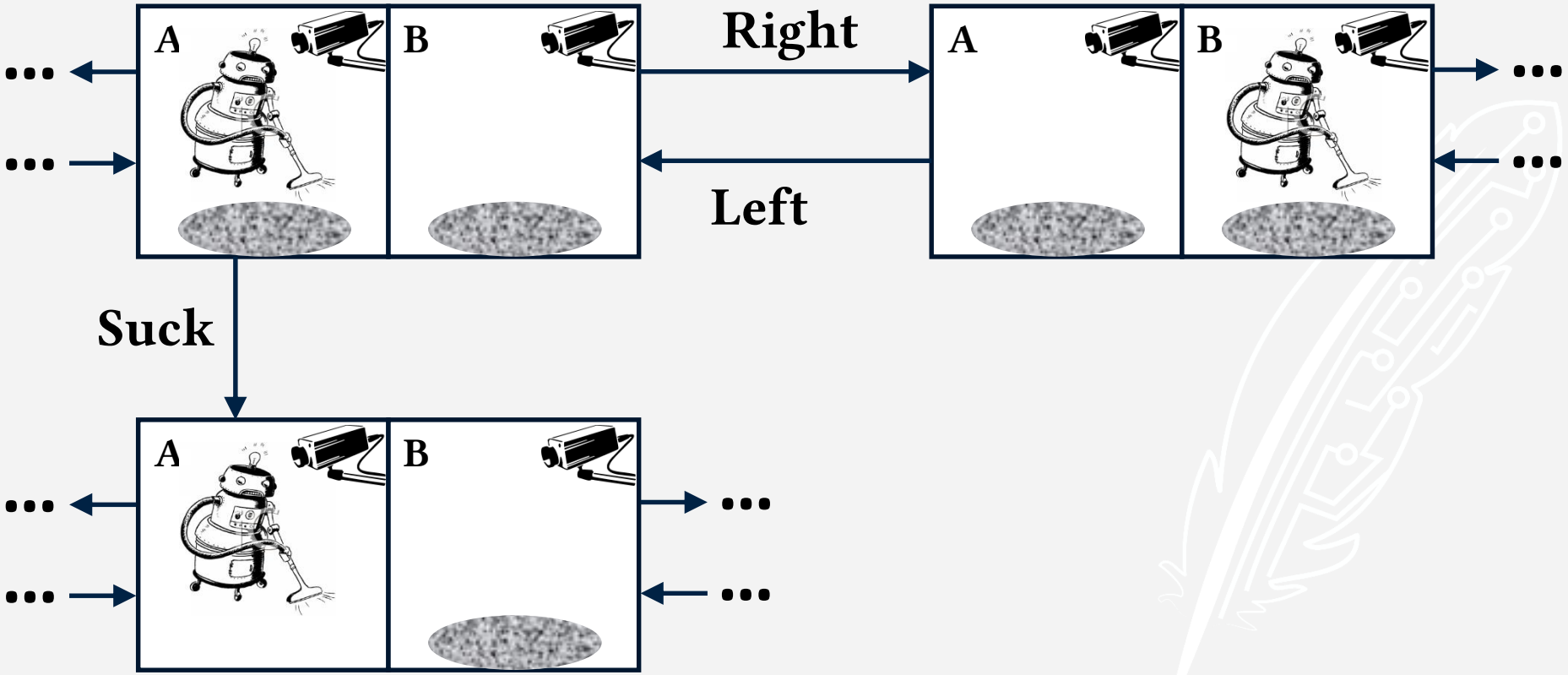


Camera Vacuum World



Observable:	Fully
Agents:	Single
Deterministic:	Deterministic
Episodic:	Episodic
Static:	Static
Discrete:	Discrete

CVW State Space



Deterministic Process

A **deterministic process** describes how the world transitions from one state to another by taking actions. It can be described as a graph whose nodes are states and whose edges are actions.

Most state spaces that we have considered so far (e.g. CVW) are deterministic processes because there is no uncertainty about the outcomes of an action.

Making Decisions with a Deterministic Process

A **decision process** is a process in which some state has been labeled the start state, some states have been labeled as goal states, and a rational agent is expected to find a way to reach the goal from the start.

We call a solution to a process a policy. A **policy** is a function which, for any given current state, specifies which action to take next.

CVW Policy

$\pi(Robot\ at\ A, A\ dirty, B\ dirty) = Suck$
$\pi(Robot\ at\ A, A\ clean, B\ dirty) = Right$
$\pi(Robot\ at\ B, A\ dirty, B\ clean) = Left$

… for every possible state

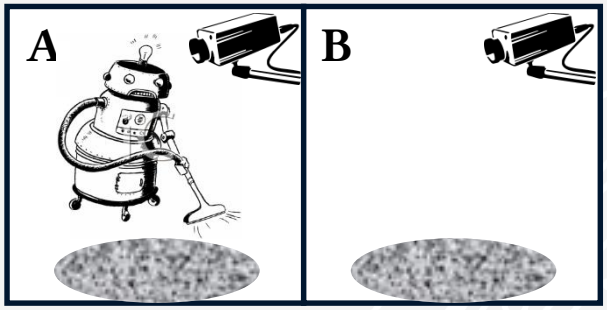
Finding Deterministic Policies

Just as many problems considered so far can be modeled using a deterministic processes, many search-based solutions we have considered are ways of finding a policy.

In general, the problem of solving a deterministic decision process is equivalent to classical planning.

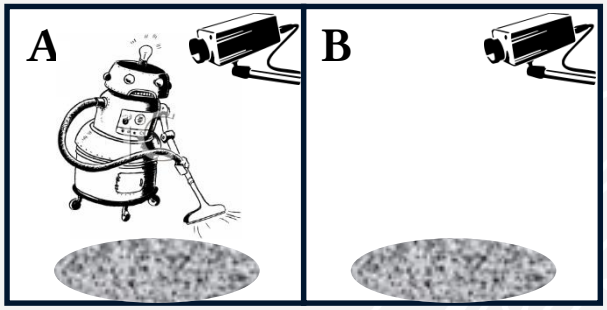
CVW

- 2 discrete rooms with cameras that detect dirt.
- A mobile robot with a vacuum.
- The goal is to ensure both rooms are clean.

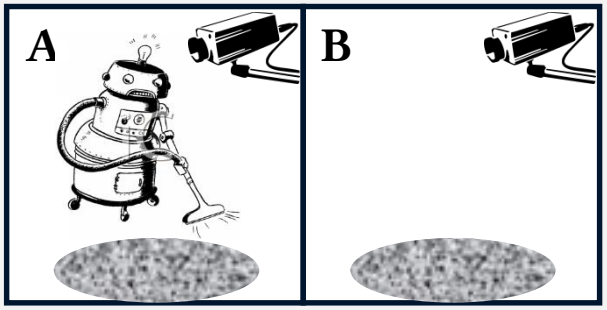


Stochastic CVW (SCVW)

- 2 discrete rooms with cameras that detect dirt.
- A mobile robot with a vacuum that fails to clean 10% of the time.
- The goal is to ensure both rooms are clean.

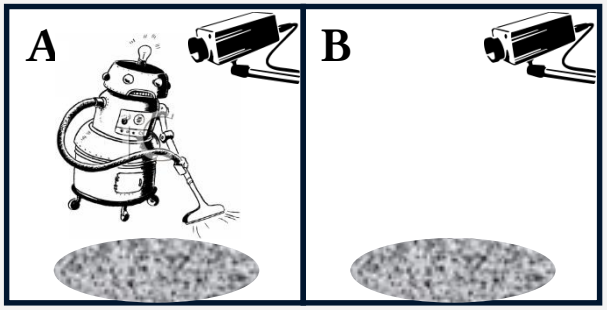


CVW



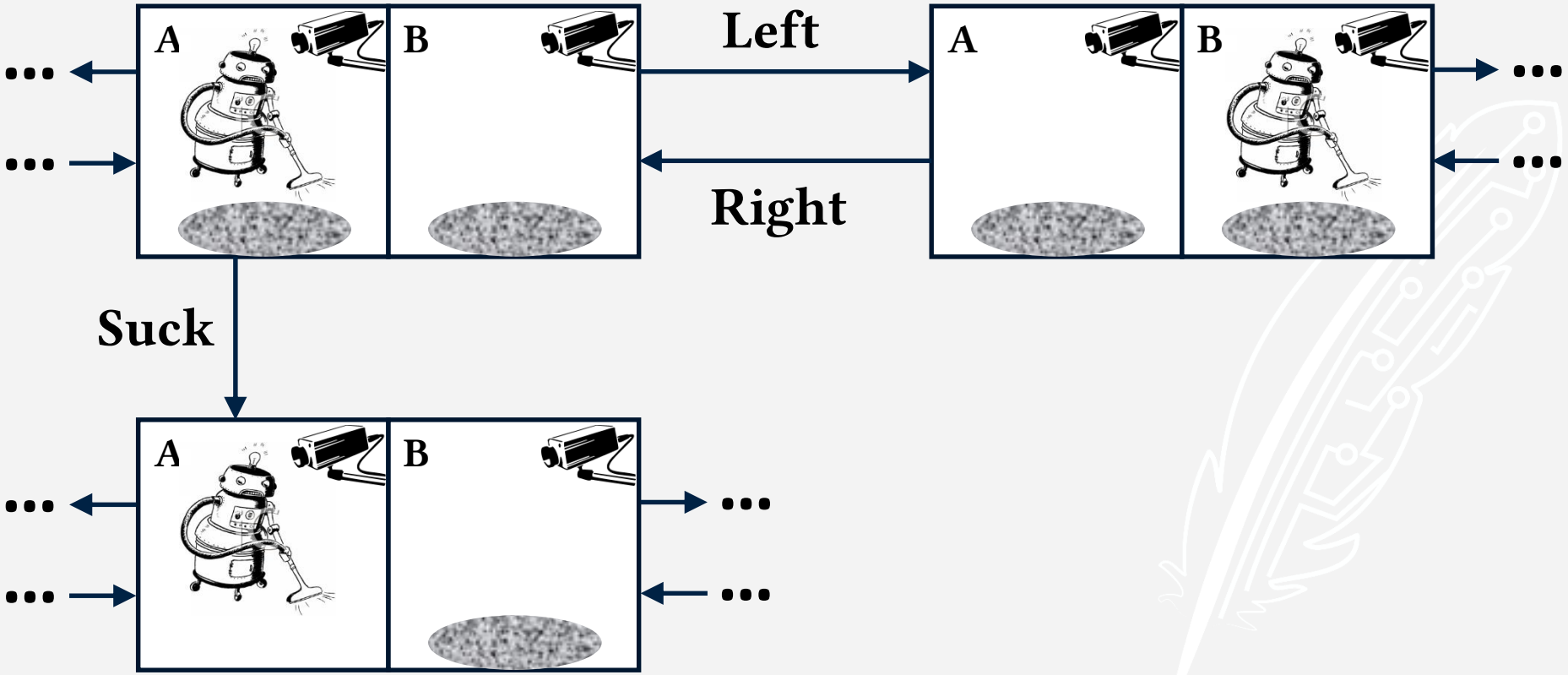
Observable:	Fully
Agents:	Single
Deterministic:	Deterministic
Episodic:	Episodic
Static:	Static
Discrete:	Discrete

SCVW

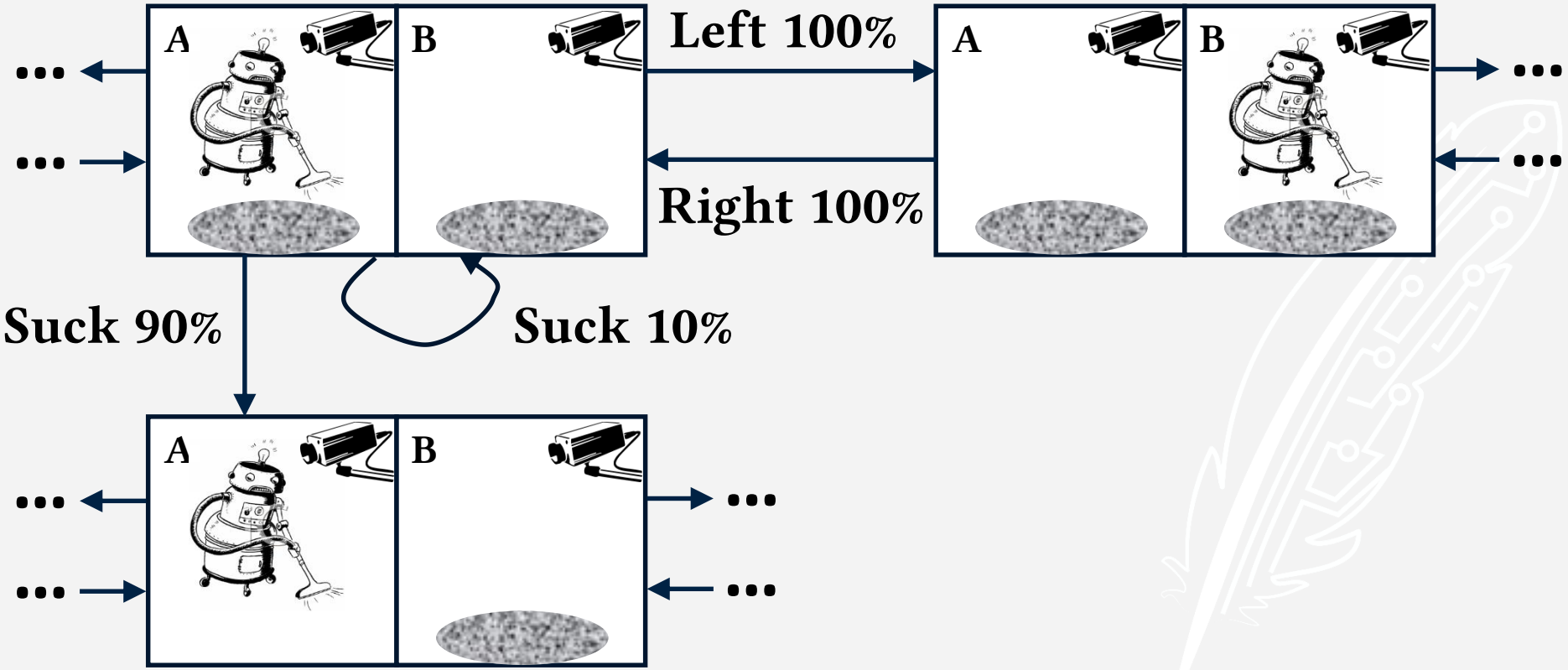


Observable:	Fully
Agents:	Single
Deterministic:	Stochastic
Episodic:	Episodic
Static:	Static
Discrete:	Discrete

CVW State Space



SCVW State Space



Stochastic Process

A **stochastic process** describes how the world transitions from one state to another by taking actions with uncertain effects. It can be described as a graph whose nodes are states and whose edges are actions (annotated with the probability the action will occur in that way).

SCVW is a stochastic processes because the outcome of an action cannot be known in advance (but once taken, the outcome is known).

Markov Process (Markov Chain)

When the probability of transitioning to a next state depends only on the previous state and the action taken, we say a stochastic process has the **Markov Property**.

This property is named for Andrey Markov, a famous mathematician who studied stochastic processes.

Markov Process (Markov Chain)

In other words, we don't need to know all the past states you have been in or all the past actions you have taken.

We only need to know your current state and the action you intend to take. From that, we can predict (stochastically) which next state you will be in after taking the action.

Markov Chains

Markov chains can be used to model simple real world processes.

One common application is text mining. Each node in the graph represents a word (w) that appears in the text. An edge leads from $w1$ to $w2$ if, somewhere in the corpus, we see $w1$ followed by $w2$. The probability of an edge represents the percentage of times $w2$ is seen to come after $w1$.

King James Programming

A bot trained on two corpuses, *The King James Bible* and *Structure and Interpretation of Computer Programs*, that generates random saying such as:

"Jesus saith unto them, Ye know that the relationship between Fahrenheit and Celsius temperatures is Such a constraint."

"By running the test with more and more in knowledge and in all things approving ourselves as the ministers of the LORD, and they provoked him to jealousy with that which is good. He that doeth good is of God: but the calf of the sin offering, and the other registers that need to be immediately sworn and notarized."

Markov Decision Process (MDP)

A **Markov Decision Process** is a decision process based on a Markov chain.

An agent cannot always predict the result of an action. Thus, any policy for solving an MDP must account for all states that an agent might accidentally end up in.

This can be thought of a classical planning but where things sometimes go wrong.

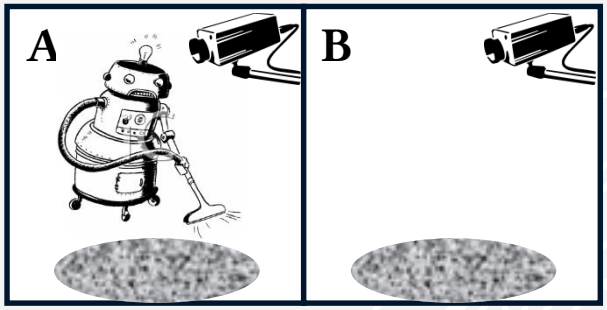
SCVW Policy

$\pi(Robot\ at\ A, A\ dirty, B\ dirty) = Suck$
$\pi(Robot\ at\ A, A\ clean, B\ dirty) = Right$
$\pi(Robot\ at\ B, A\ dirty, B\ clean) = Left$

… for every possible state

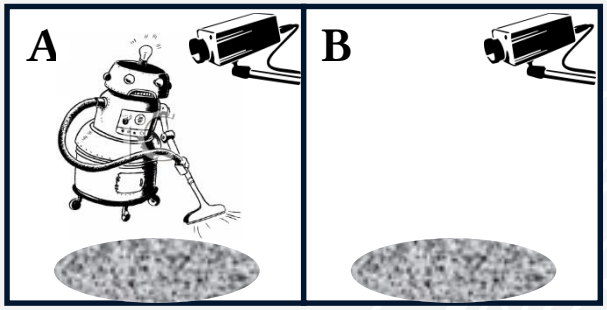
SCVW

- 2 discrete rooms with cameras that detect dirt.
- A mobile robot with a vacuum that fails to clean 10% of the time.
- The goal is to ensure both rooms are clean.

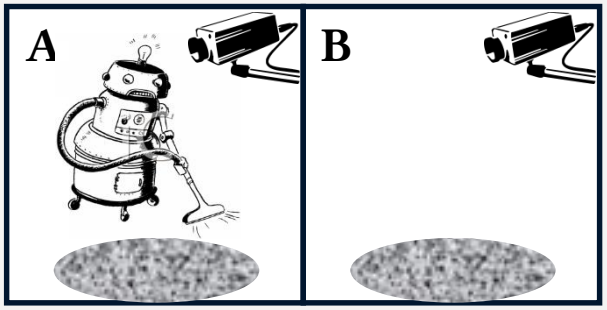


Hidden SCVW (HSCVW)

- 2 discrete rooms with cameras that detect dirt 95% of the time dirt is present.
- A mobile robot with a vacuum that fails to clean 10% of the time.
- The goal is to ensure both rooms are clean.

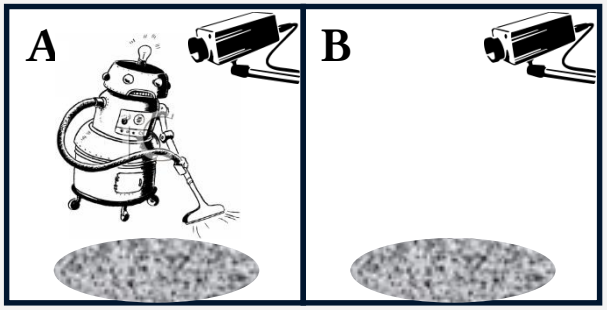


SCVW



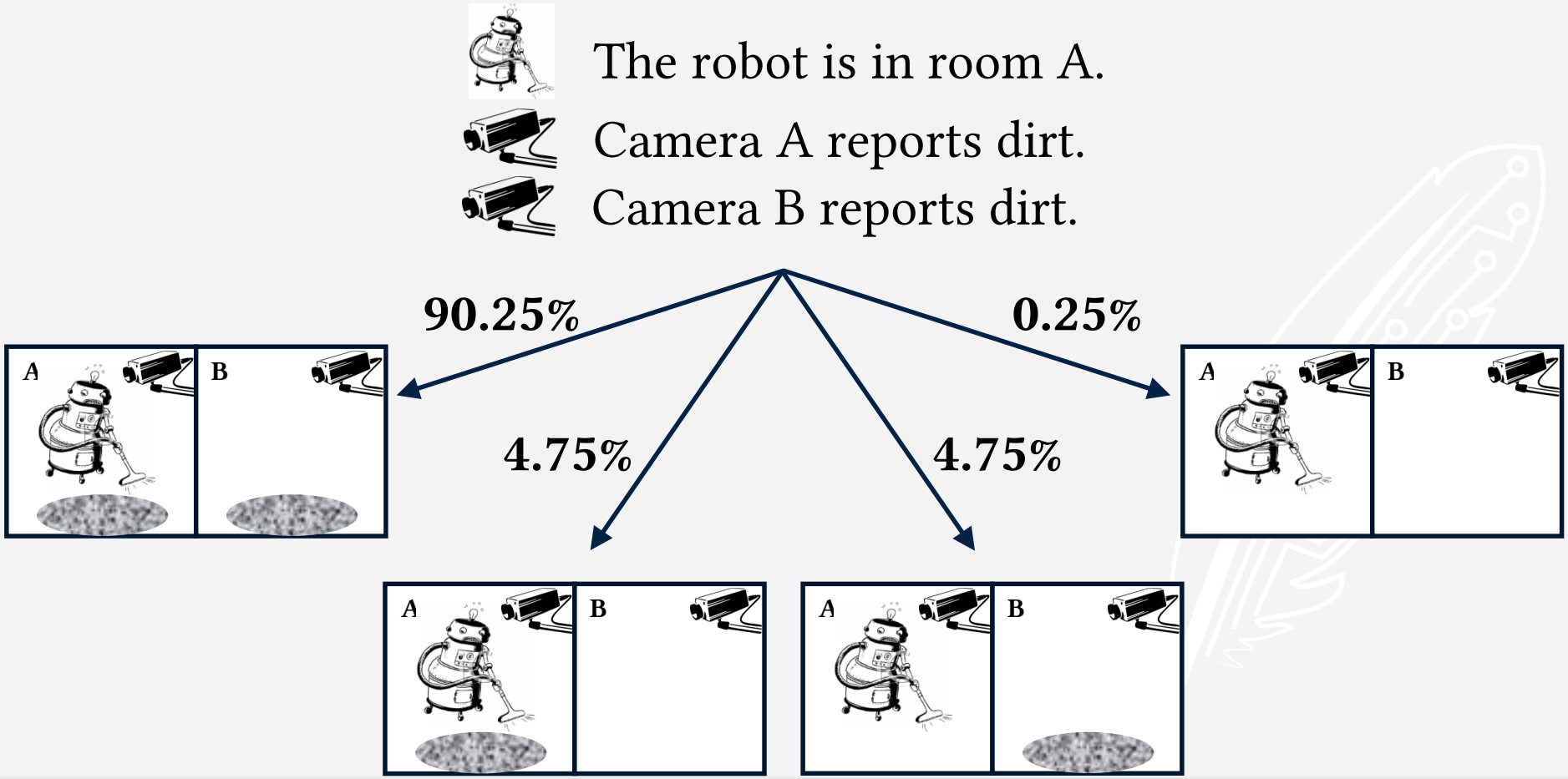
Observable:	Fully
Agents:	Single
Deterministic:	Stochastic
Episodic:	Episodic
Static:	Static
Discrete:	Discrete

HSCVW



Observable:	Partially
Agents:	Single
Deterministic:	Stochastic
Episodic:	Episodic
Static:	Static
Discrete:	Discrete

HSCVW State Space



Hidden Markov Model (HMM)

A **Hidden Markov Model** is a process which is assumed to operate according to a Markov process, but whose state is not directly observable.

Based on whatever observations are available, an agent must maintain a probability distribution of possible current states and their likelihoods.

Partially Observable Markov Decision Process (POMDP)

A **Partially Observable Markov Decision Process** is a decision process based on a hidden Markov model.

An agent does not know the actual state of the world, but can guess it based on observations. It must choose a policy which is expected to maximize the chance of reaching a solution.

Markov Processes

How the World Works	**How an Agent Makes Decisions in that World**
Deterministic Process →	Planning
Markov Chain →	Markov Decision Process
Hidden Markov Model →	Partially Observable Markov Decision Process

Reinforcement Learning

Reinforcement Learning is a kind of machine learning in which labeled data is not available, but for which periodic feedback (in the form of rewards and punishments) is available.

An agent takes actions, observes its reward or punishment, and eventually learns which actions lead to success and which lead to failure.

For example, considering training a dog.

RL as Supervised Learning

Reinforcement learning can be thought of as a kind of supervised learning in which the agent must generate its own labeled data.

RL and POMDP's

Most reinforcement learning algorithms are designed to learn optimal policies for MDP's and POMDP's.

- The agent has observations which tell it which states it might be in (and how likely). It assumes the process is a Markov process.
- The agent takes actions and observes rewards or punishments.
- The agent eventually learns how to behave in the environment so as to maximize its reward.

Exploration vs. Exploitation

When an agent takes actions whose outcomes are relatively unknown, it is called **exploration**.

When an agent takes actions whose outcomes are known to produce high rewards, it is called **exploitation**.

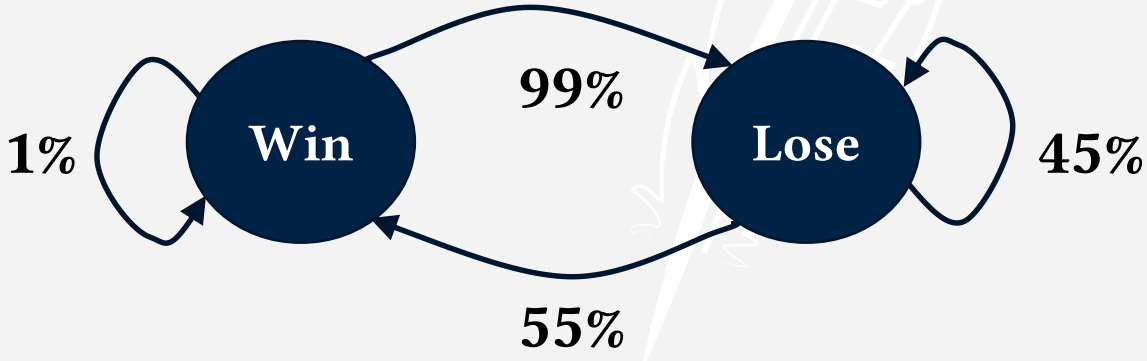
One of the central problems in RL is striking a balance between exploration (learning new information) and exploitation (using known information).

n-Armed Bandits Problem

Imagine you have $1000 that you intend to spend on a row of slot machines (one-armed bandits) in Las Vegas. How can you spend that money so as to maximize the amount of money you have left at the end of the day?

n-Armed Bandits Problem

We assume each machine is controlled by a Markov process.



n-Armed Bandits Problem

But we can't see it; we can only observe the results (wins or losses), so it is a hidden Markov model.

n-Armed Bandits Problem

Our task is to find an optimal policy—that is, to solve a partially observable Markov decision process.

n-Armed Bandits Problem

Say you have played Machine A 20 times. You have won 10 times and lost 10 times. You have played Machine B 1 time and lost. Is it logical to conclude that you should never play Machine B because, so far, you have lost 100% of the time?

n-Armed Bandits Problem

No. You are exploiting Machine A too early without exploring Machine B enough. Machine B might have a better win/loss ratio than A. You need to gather more information.