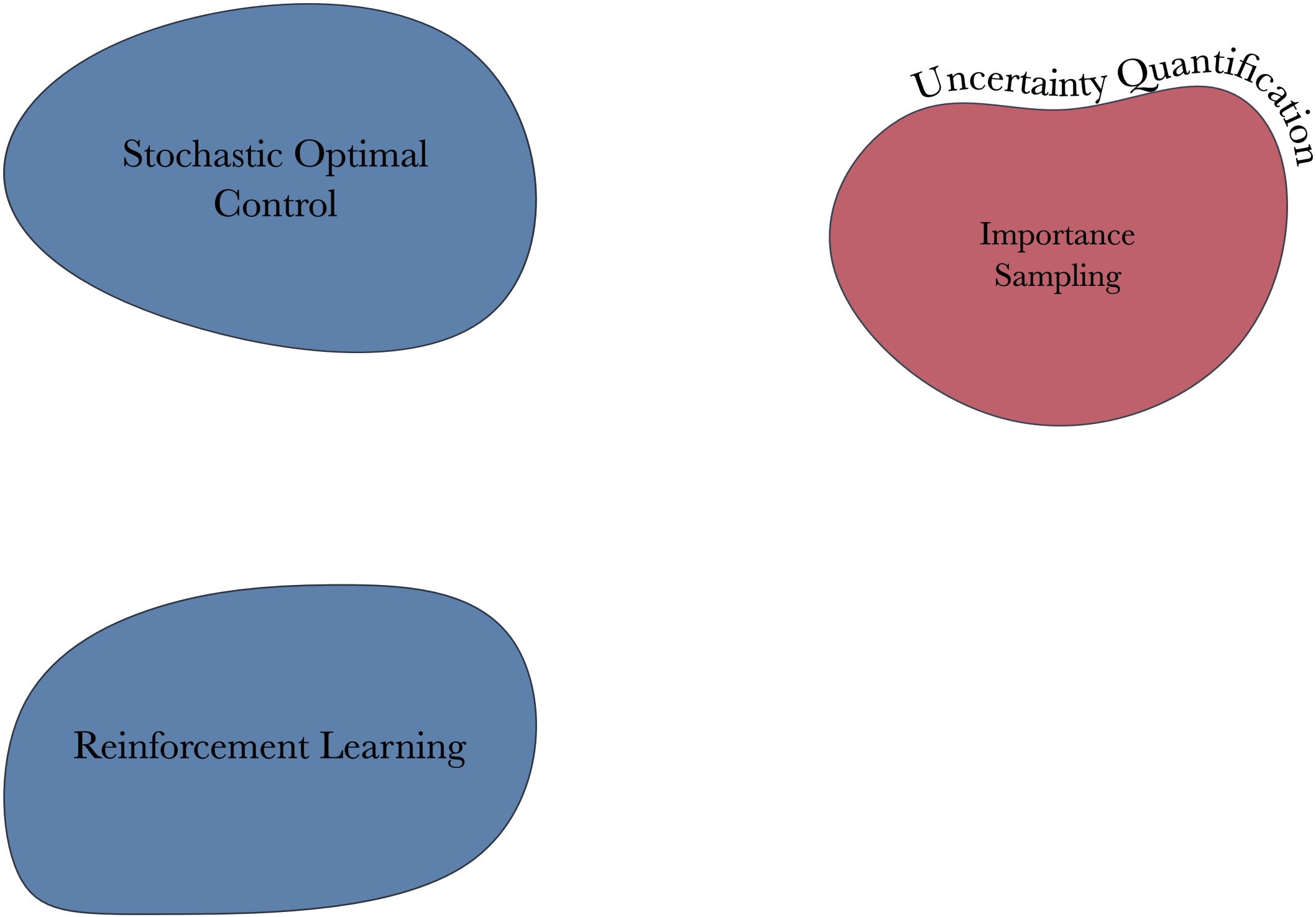
Connecting SOC with RL – Importance sampling

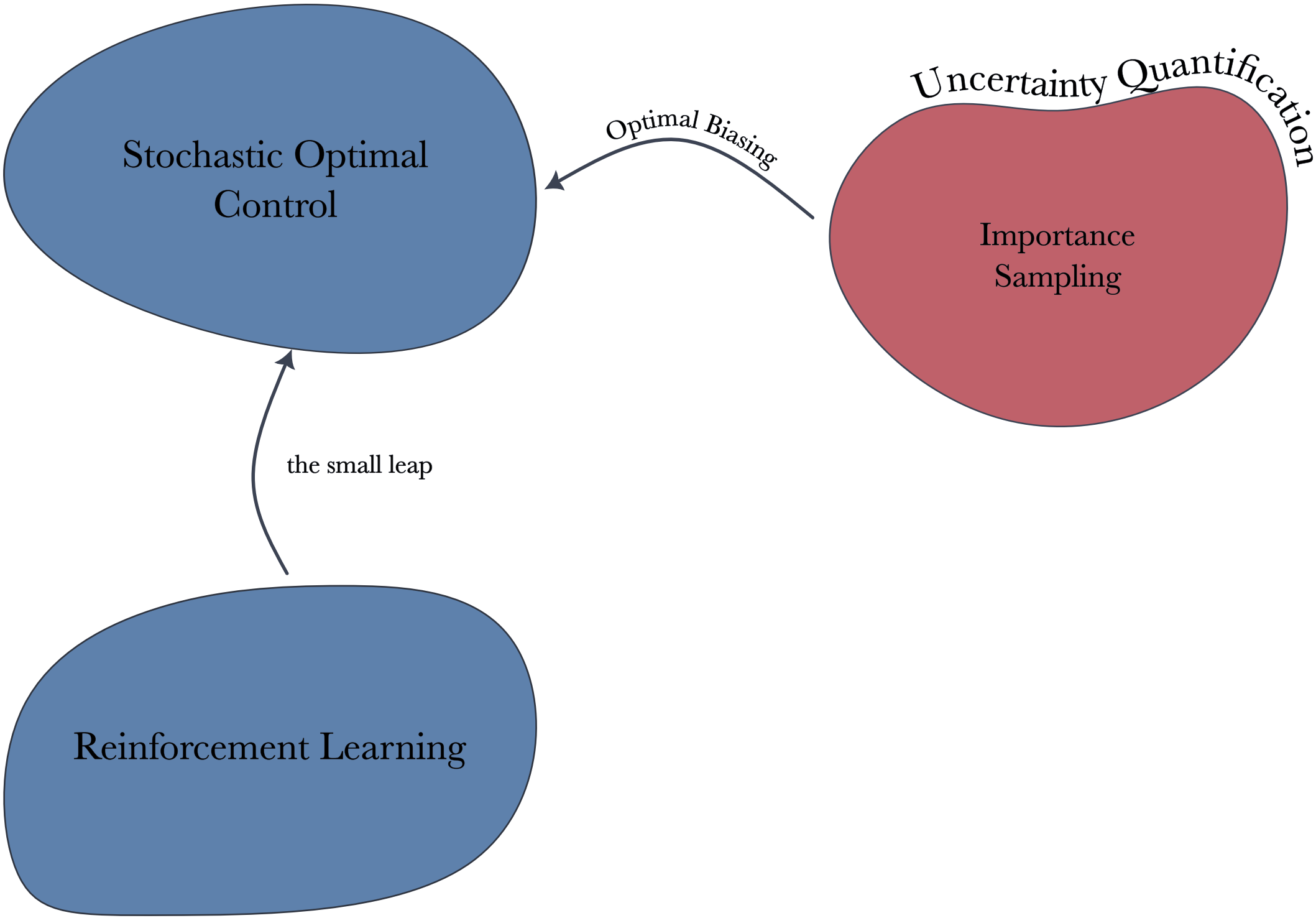
Alonso Cisneros

Zuse Institute Berlin


Stochastic Optimal Control
Uncertainty Quantification
Importance Sampling
Reinforcement Learning

Speaker notes

By the end of the talk these will all be connected. What I want to do is build the connection to yet another blob: ABMs


Stochastic Optimal Control
Optimal Biasing
Uncertainty Quantification
Importance Sampling
the small leap
Reinforcement Learning

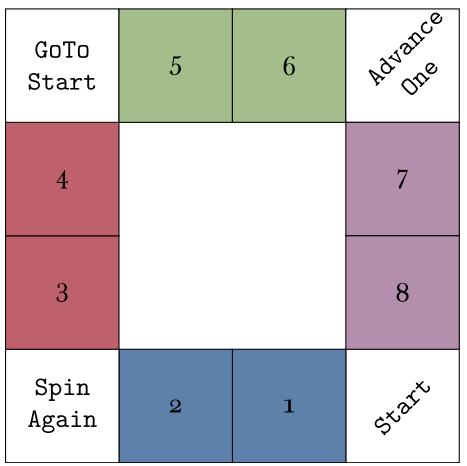
Outline

1. *Crash* course on RL
2. What is importance sampling

- The connection to optimization
- Optimal biasing

4. Optimal biasing as an RL problem
5. The things I'd like to connect

Crash course on Reinforcement Learning



A miniopoly board

Speaker notes

- I’m training a robot to become the best Miniopoly player
- The rules:
  - Players play in turns
  - They move a number of squares determined by a 4-sided dice roll
  - After completing a lap, it gets a reward of x dollars
  - The trap squares do what it says on the square
  - They can buy property and hotels in the squares.
    - If they land of a square someone owns, they pay
    - If someone lands on their square, they charge rent
  - The game ends when someone runs out of money

- The game has a state at turn t denoted s_t
- At a turn t players roll the dice
- The change in money after buying/paying rent/charging rent is recorded as a reward r_t

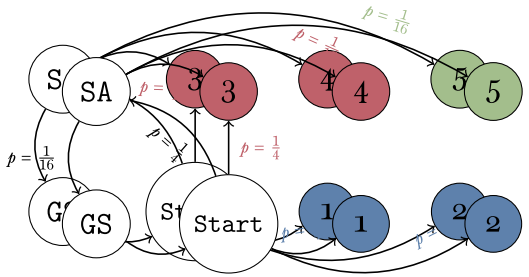
Important

We train our robot to maximize the rewards as it takes actions exploring the space of states

Speaker notes

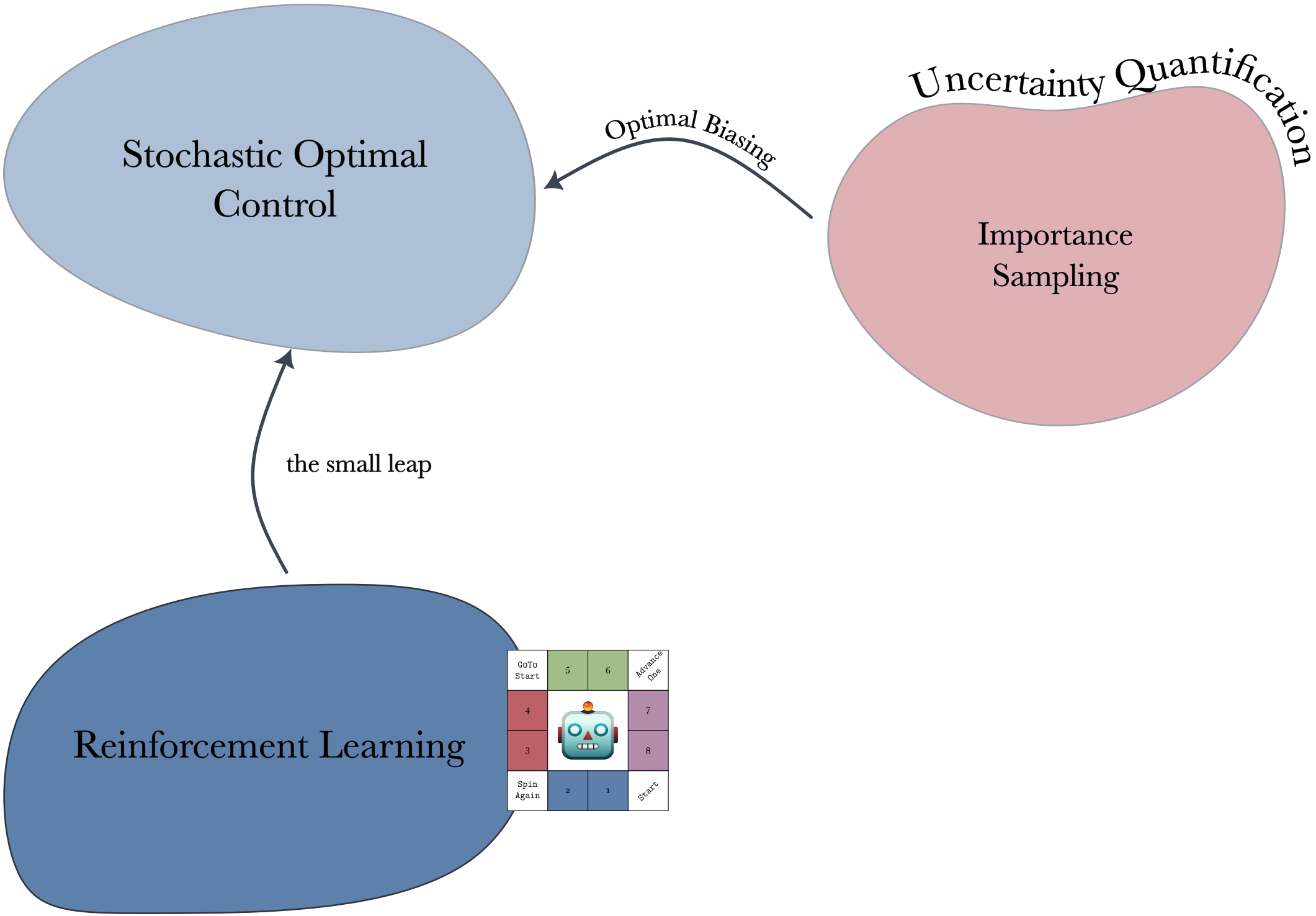
- The state of the game an any given time is information like, who owns what squares, how much money they have, in what positions each player is, and so on.
- Once a player lands in another square, they can choose to buy it if available. If it’s not, we carry out the accounting of how much rent is, and let the player know how much it won/lost and to what square this is connected.

Dynamics of the space



Speaker notes

In this example we have full access to the dynamics of the problem. This is not always the case


Stochastic Optimal Control
Optimal Biasing
Uncertainty Quantification
Importance Sampling
the small leap
Reinforcement Learning
GoTo Start
5
6
Advance One
4
7
3
8
Spin Again
2
1
Start

Speaker notes

- We just reviewed very quickly the first bubble

What if we don’t know how square transitions work?

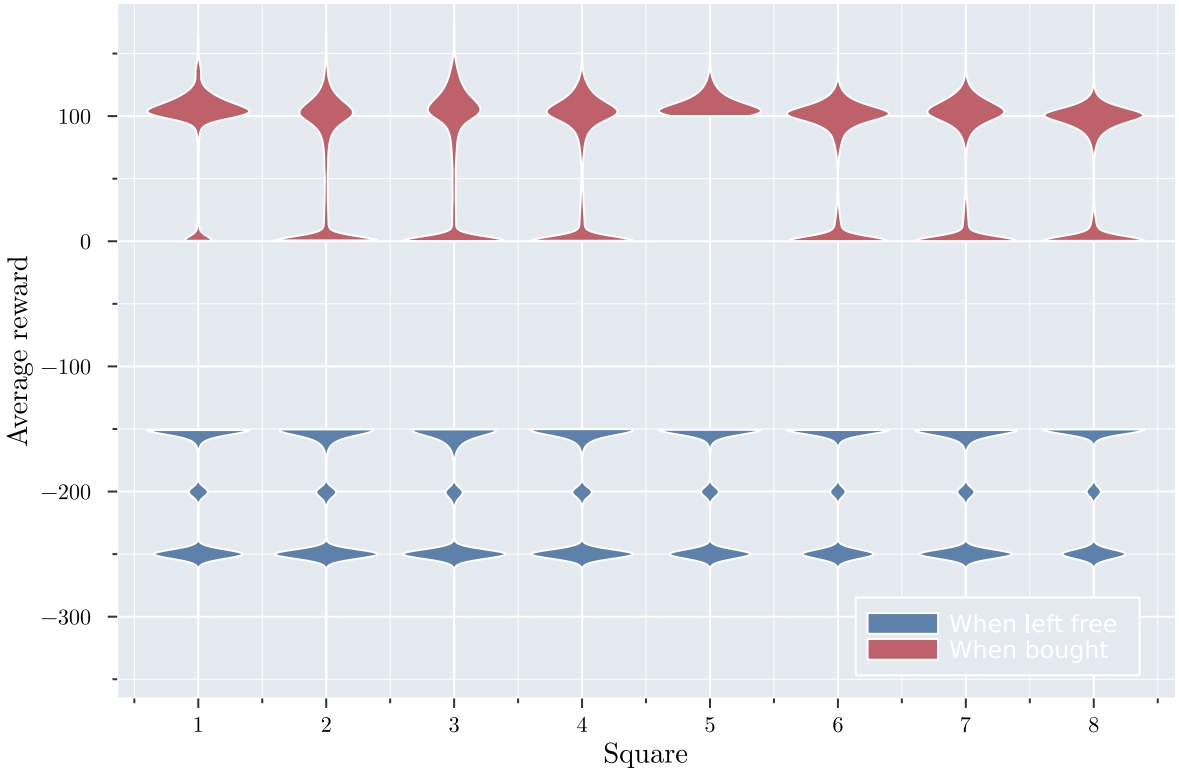
We calculated transition probability *with* the knowledge of the dice

Speaker notes

We’ll now move on to MCMC. These concepts are not necessarily related, but I will take advantage of this example so that we can have a reasonable inutition from the start.

Markov Chain Monte Carlo

- We let the robot roam around and buy squares as it pleases
  - For any square, it can either buy it or not
- We register what it gained or lost by buying or not buying a square by the end of the game.


100
0
−100
−200
−300
Average reward
1
2
3
4
5
6
7
8
Square
When left free
When bought

Speaker notes

- Graph:
  - This is a violin plot. It shows the estimated probability densities of observing…
  - The x axis shows the different squares
  - The y axis shows the estimated rewards. i.e. money
  - The red distributions correspond to the expected reward when buying a square, while the blue the expected loss when not buying them
  - i.e. When we buy squares we expect to profit from them, but clearly not all squares are as profitable, look at the different shapes of the distributions. On the other hand, it looks like losing any given square leads to the same expected loss

Stochastic Optimal Control

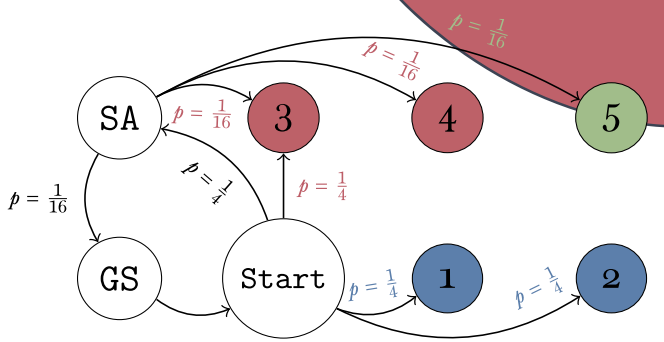
Optimal Biasing

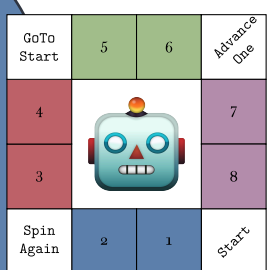
Uncertainty Quantification

Importance Sampling

the small leap

Reinforcement Learning





What we’ve covered so far

Speaker notes

Now we have introduced MCMC as a way for our robot to explore it’s environment and estimate which moves are beneficial to his goal. This is what the diagram is representing

Moving on…

Importance Sampling

- We wanted to compute the expected reward of the robot after the entire game
- MCMC often fails in metastable systems
- Importance sampling aims to remedy this

> **Important**
>
> The general idea of importance sampling is to draw random variables from another probability measure and subsequently weight them back in order to still have an unbiased estimator of the desired quantity of interest

Speaker notes

- We **estimated** this quantity by observing and measuring an empirical average. But our approximation for extremely unlikely states will always be bad by virtue of how little samples we have.
- Metastability makes MCMC extremely hard to apply. The variance of our estimations is always going to be enormous under these conditions
- We can aim to make sampling faster by reducing the inherent variance
- **After Callout** In the case of stochastic processes this change of measure corresponds to adding a control to the original process

More formally…

Speaker notes

- Where:
  - X_s is the position of our particle at time s
  - V is a “potential”
  - We assume there exists a unique strong solution that is ergodic
- Note that τ is a.s. finite
- Where $\mathcal{W}$ serving as a measure of “work” over a trajectory

Our main goal is to compute

$$\Psi(X) := \mathbb{E}^x[I(X)] := \mathbb{E}[I(X) \mid X_0 = x]$$

But…

- MCMC has terrible properties because of metastability
- Closed forms of $\Psi(X)$ maybe don't exist

> **Tip**
>
> - We can "push" the particle adding force, as long as we account for it and correct for that bias
> - That "push" is achieved by adding a control u.

The new, controlled dynamics are now described as

$$\mathrm{d}X_s^u = (-\nabla V(X_s^u) + \sigma(X_s^u)\, u(X_s^u))\mathrm{d}s + \sigma(X_s^u)\mathrm{d}W_s,$$

Via Girsanov, we can relate our QoI to the original as such:

$$\mathbb{E}^x\left[I(X)\right] = \mathbb{E}^x\left[I(X^u)M^u\right],$$

where the exponential martingale

$$M^u := \exp\left(-\int_0^{\tau^u} u(X_s^u)\cdot \mathrm{d}W_s - \frac{1}{2}\int_0^{\tau^u} |u(X_s^u)|^2 \mathrm{d}s\right)$$

corrects for the bias the pushing introduces.

Important

The previous relationship always holds. But the variance of the estimator depends *heavily* on the choice of u.

Clearly, we aim to achieve the smallest possible variance through on *optimal control* u^*

$$\mathrm{Var}\left(I(X^{u^*})M^{u^*}\right) = \inf_{u\in\mathcal{U}}\{\mathrm{Var}(I(X^u)M^u)\}$$

Speaker notes

- Where:
  - X_s^u is the position of our particle at time s under control u
  - The potential u is an Itô integrable function satisfying a linear growth condition
- Note that τ is a.s. finite
- Where $\mathcal{W}$ serving as a measure of “work” over a trajectory

Connection to optimization

It turns out [1] that the problem of minimizing variance corresponds to a problem in optimal control

The cost functional J to find the variance minimizing control is

$$J(u;x) := \mathbb{E}^x \left[\mathcal{W}(X^u) + \frac{1}{2} \int_0^{\tau^u} |u(X_s^u)|^2 \mathrm{d}s \right],$$

With this formulation,

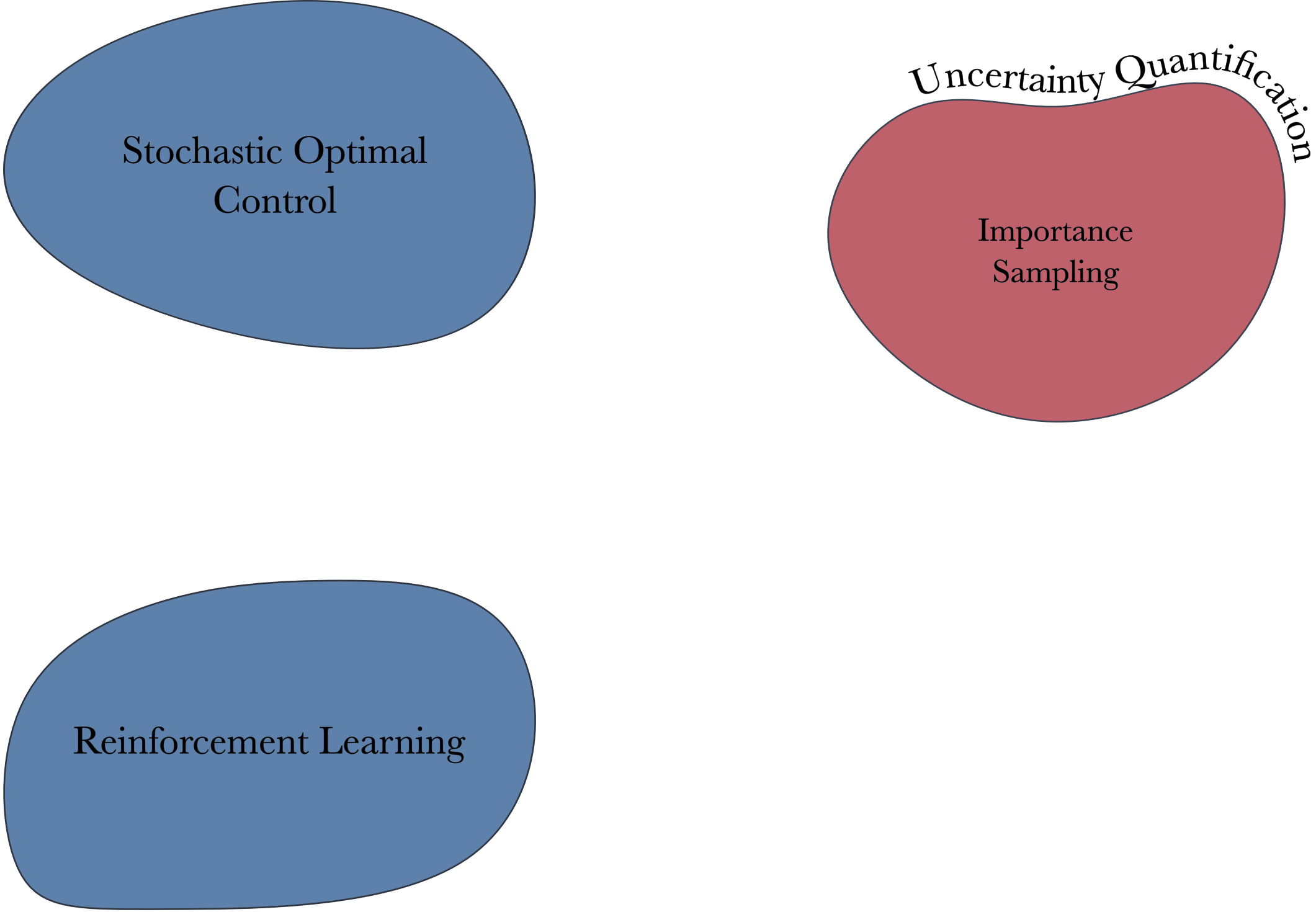
$$\Phi(x) = \inf_{u \in \mathcal{U}} J(u; x).$$

Important

The optimal bias achieves zero variance:

$$\operatorname{Var}\left(I(X^{u^*}) M^{u^*}\right) = 0.$$

Optimal biasing through RL


Stochastic Optimal Control
Uncertainty Quantification
Importance Sampling
Reinforcement Learning

- Let's reconsider the SOC problem (excuse the change in notation)
- We discretize the dynamics equation

$$
\begin{aligned}
\boldsymbol{s}_{t+1} &= \boldsymbol{s}_t + (-\nabla V(\boldsymbol{s}_t) + \sigma u(\boldsymbol{s}_t))\Delta t + \sigma\sqrt{\Delta t}\,\eta_{t+1} \\
\boldsymbol{s}_0 &= \boldsymbol{x}
\end{aligned}
$$

Speaker notes

- Sorry for the slightly different notation
- Where
  - Our state is now represented by s
  - We have the same potential V
  - The diffusion term is σ again
  - Δt is the length of the time step
  - The term $\sqrt{\Delta t}\eta_{t+1}$ is a Brownian increment, $\eta_t \sim N(0, 1)$

The time-discretized objective function is given by

$$J(u;x) := \mathbb{E}^x \left[g(s_{T_u}) + \sum_{t=0}^{T_{u-1}} f(s_t)\Delta t + \frac{1}{2} \sum_{t=0}^{T_{u-1}} |u(s_t)|^2 \Delta t \right]$$

- Our stopping time τ is now denoted T_u

Some formalities

- The state space $\mathcal{S}$ is all possible $s \in \mathbb{R}^d$
- The action space $\mathcal{A}$ is the codomain of all possible controls $\mathbb{R}^d$
- The stopping time T_u for the controlled process is a.s. finite

- We'll approximate the control with Galerkin projections u_θ
- We still need to derive probability transition and reward functions

- The return we want to optimize depends on a rewards function

$$r_t = r(\boldsymbol{s}_t, a_t) := \begin{cases} -f(\boldsymbol{s}_t)\Delta t - \frac{1}{2}|a_t|^2 \Delta t & \text{if } \boldsymbol{s}_t \notin \mathcal{T} \\ -g(\boldsymbol{s}_t) & \text{if } \quad \boldsymbol{s}_t \in \mathcal{T} \end{cases}$$

Speaker notes

- The reward function is defined such that the corresponding return along a trajectory equals the negative term inside the expectation of the time-discretized cost functional
- Notice that the reward signal is in general not sparse since the agent receives feedback at each time step but the choice of the running cost f and the final cost g can influence this statement.

Stochastic Optimal Control

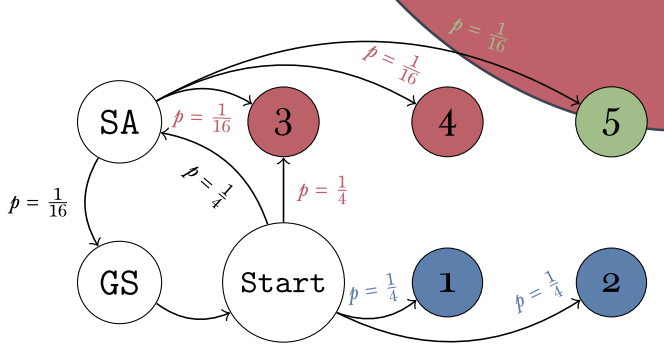
Optimal Biasing

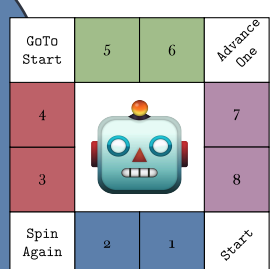
Uncertainty Quantification

Importance Sampling

the small leap

Reinforcement Learning





What I want to do

- The connection (Quer and Borrell 2024) works because of the properties of J
- Two posibilities
  1. Take the SOC already published in (Helfmann et al. 2023) and pose the *right* cost functional (the “easy one”)
  2. Go back to the MFE and then
     - Break the assumption of fully connected Agent-Agent network
     - Find the MFE without that assumption
     - Pose a SOC
     - Find the right cost functional

Speaker notes

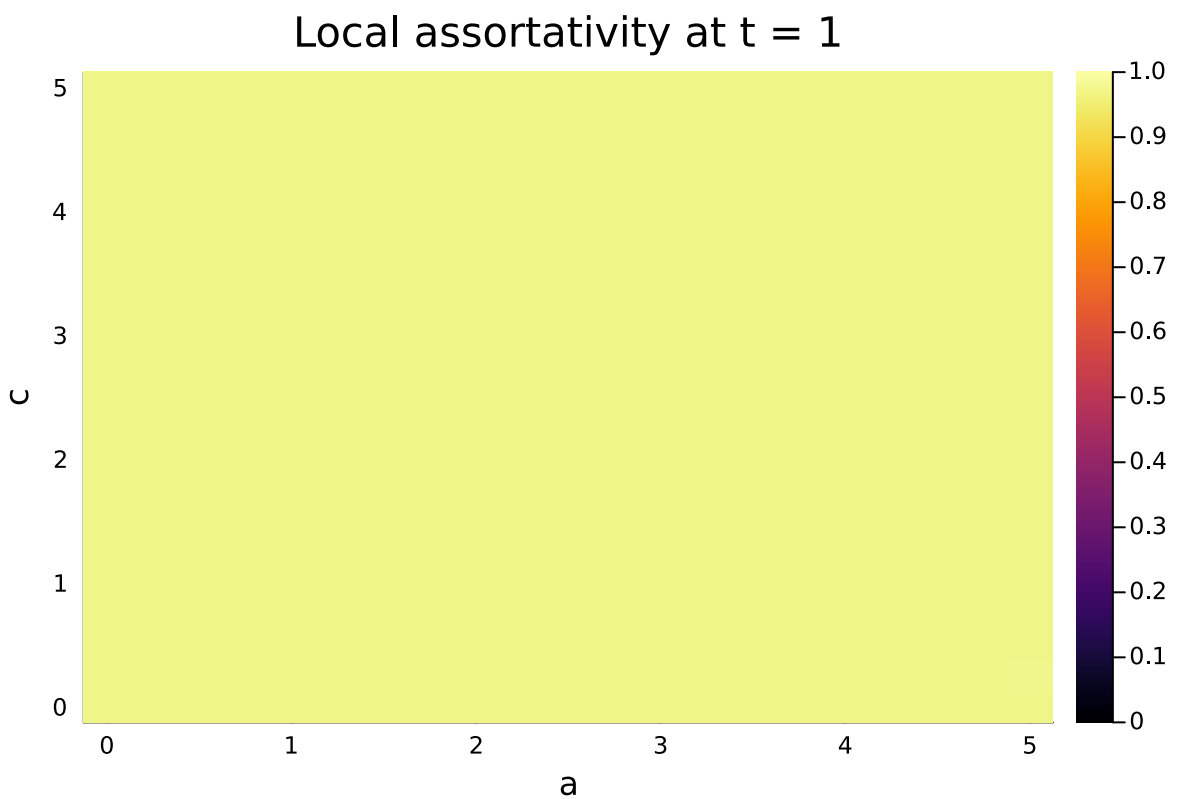
- Half of the magic is the fact that the functional to be optimized to solve the optimal biasing SOC problem leads to HJB. It’s an open question whether I can make our SOC compatible

What I’m trying at the moment

- Approach it from the MFE side
  - Looking for generalizations of McKean-Vlasov PDEs with time-dependent networks
  - Thinking what graphon I can get from the network
- Finding the right cost functional on the “easy” case

Bonus: Assortativity & network topology

Start | End | End (echo chamber)



References

Helfmann, Luzie, Nataša Djurdjevac Conrad, Philipp Lorenz-Spreen, and Christof Schütte. 2023. “Modelling Opinion Dynamics Under the Impact of Influencer and Media Strategies.” *Scientific Reports* 13 (1): 19375. https://doi.org/10.1038/s41598-023-46187-9.

Quer, J., and Enric Ribera Borrell. 2024. “Connecting Stochastic Optimal Control and Reinforcement Learning.” *Journal of Mathematical Physics* 65. https://doi.org/10.1063/5.0140665.

Ribera Borrell, Enric, Jannes Quer, Lorenz Richter, and Christof Schütte. 2024. “Improving Control Based Importance Sampling Strategies for Metastable Diffusions via Adapted Metadynamics.” *SIAM Journal on Scientific Computing* 46 (2): S298–323. https://doi.org/10.1137/22M1503464.