Some general considerations on „applied“ RL

Closing lecture for the RL bootcamp 2024 in Salzburg
Simon Hirlaender

Strongly influenced by Sergey Levines Lectures

Outline

- Practical questions in RL
- Some remarks on RL
- What if you know how the world works?
- The basis of the decisions
- Challenges with the basic algorithms
- Some philosophical perspectives
- Final remarks

Outline

- **Practical questions in RL**
- Some remarks on RL
- What if you know how the world works?
- The basic of the decisions
- Challenges with the basic algorithms
- Some philosophical perspectives
- Final remarks

Practically important questions before you start

- **Understand what you want to solve!**
  - ➡ Clearly define the problem and its scope
- Try to formulate your problem mathematically as **MPD** (if possible)
  - ➡ Define variables, constraints, and objectives clearly
- Find the **correct solution approach**. Is RL the right tool?
  - ➡ If yes what kind of **algorithm** is appropriate?
  - ➡ If no what are the **alternatives?**
- How can you measure success and generalisation?
  - ➡ Define Metrics: Establish clear metrics to evaluate the model's performance (e.g., cumulative reward, convergence rate).
  - ➡ Benchmarking: Compare the model's performance against baseline methods or existing solutions.

Outline

- Practical questions in RL
- **Some remarks on RL**
- What if you know how the world works?
- The basis of the decisions
- Challenges with the basic algorithms
- Some philosophical perspectives
- Final remarks

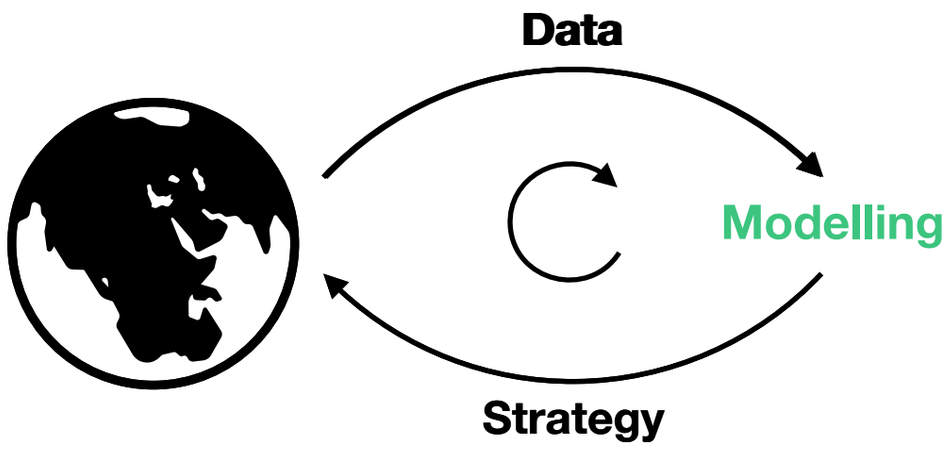
RL - what is it today?

- What is addressed by RL: position in AI, community of researchers applying tools, data-driven dynamic programming
- Little knowledge of probabilistic mechanism how data and rewards change over time
- Probe and learn dynamics to find control
- General field of sequential decision making (SDM)

RL is powerful

Learning to make good decisions under uncertainty

Information → Decision → Information → Decision → Information → …

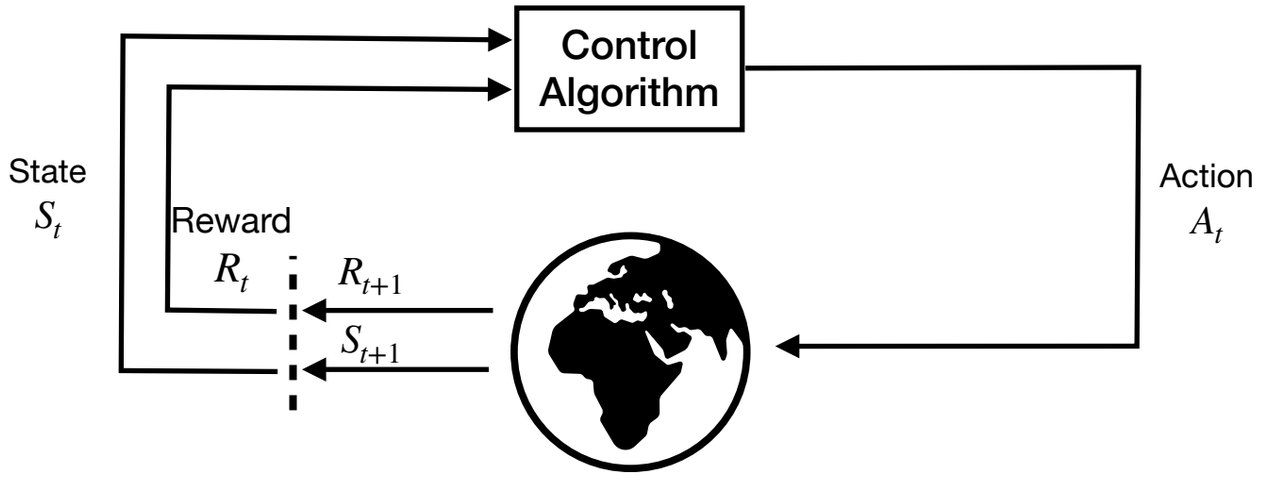


- What do we do with data? ⇒ Generally we try to derive a decision!
- Decisions always part of an evolving environments and change them (policy may invalidate empirical basis of model) - consequences
- RL is an **active paradigm** - prediction is static

RL involves

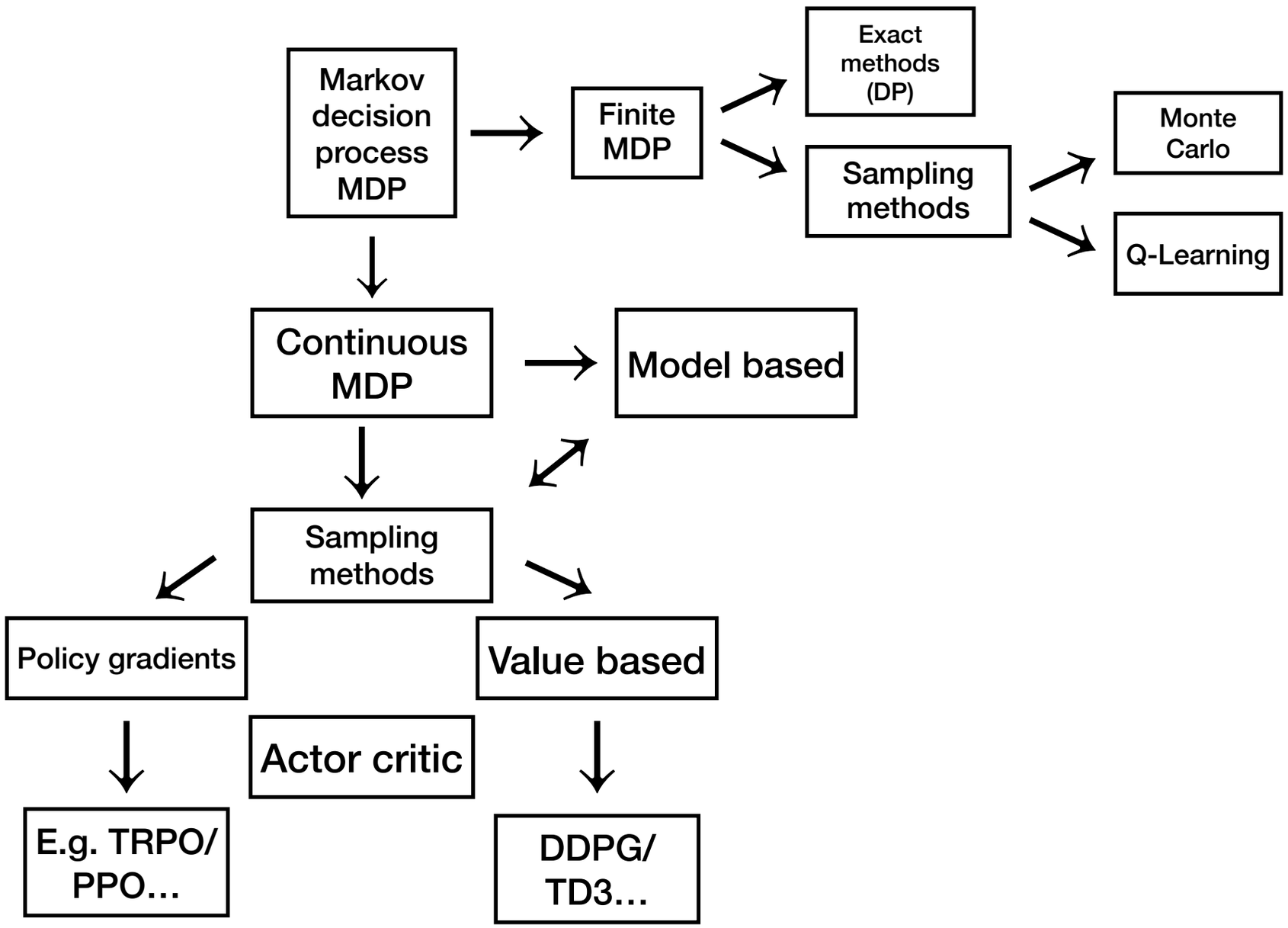
	AI Planning	SL	UL	RL
Optimize	x	x	x	x
Learning from experience		x	x	x
Generalize	x	x	x	x
Delayed consequences	x			x
Exploration				x

What is the objective in RL?: find π^*



- H horizon (H can be infinite)
- A policy is defining a distribution over actions conditioned on states $\pi(a_t \mid s_t)$
- The trajectory $\tau = (s_0, a_0, \ldots s_H, a_H)$
- The trajectory distribution $p_\pi(\tau) = \rho \prod_{t=0}^{H} \pi(a_t \mid s_t) P(s_{t+1} \mid s_t, a_t)$
- The objective is $J(\pi) = \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t) \right]$
- The optimal policy $\pi^* = \arg\max_\pi J(\pi) = \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t) \right]$

Landscape


Markov decision process MDP
Finite MDP
Exact methods (DP)
Sampling methods
Monte Carlo
Q-Learning
Continuous MDP
Model based
Sampling methods
Policy gradients
Value based
Actor critic
E.g. TRPO/ PPO…
DDPG/ TD3…

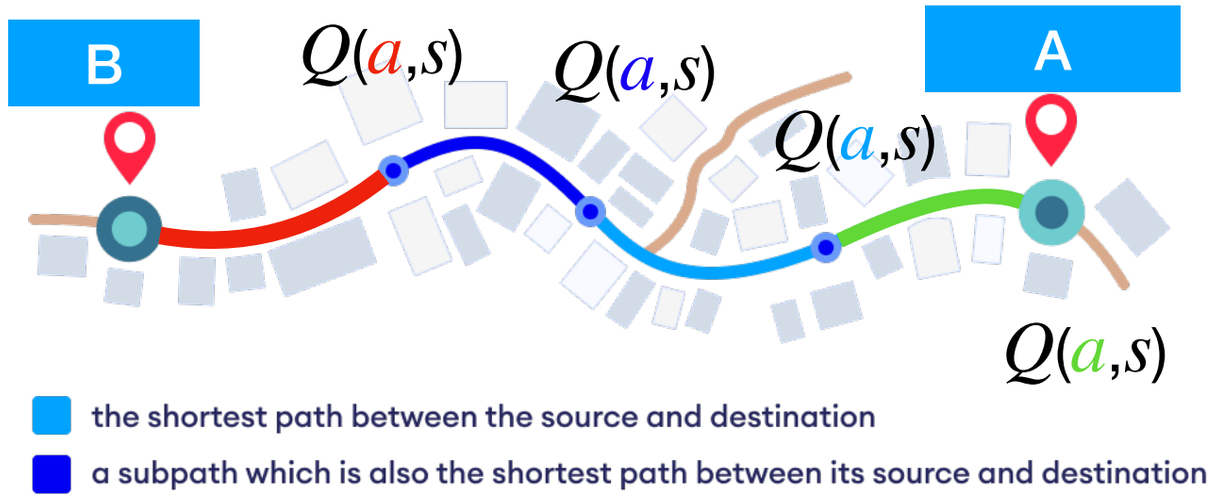
Outline

- Practical questions in RL
- Some remarks on RL
- **What if you know how the world works?**
- The basis of the decisions
- Challenges with the basic algorithms
- Some philosophical perspectives
- Final remarks

Excursion: What if we know how the world works?

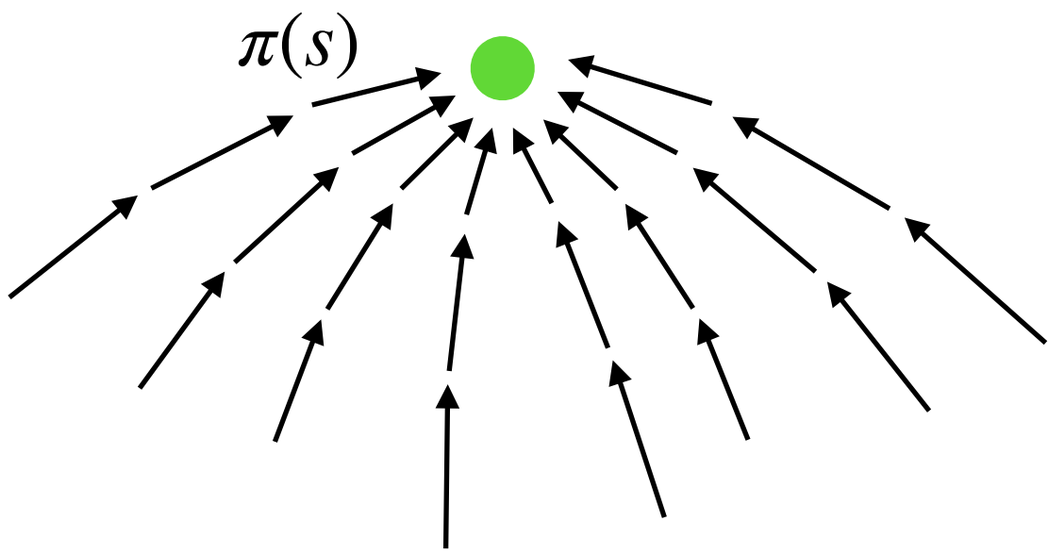
What if we have a model $P(s_{t+1} \mid s_t, a_t)$ (and $r(s_t, a_t)$)?

Bellman ~1957: dynamic programming



- Bellman idea:
  - ➡ Exact backwards recursion (if all transition probabilities are perfectly known) → unique solution for optimal policy
  - ➡ Basis of all **value-based** methods in RL - estimating the future reward of each state and construct a policy from there

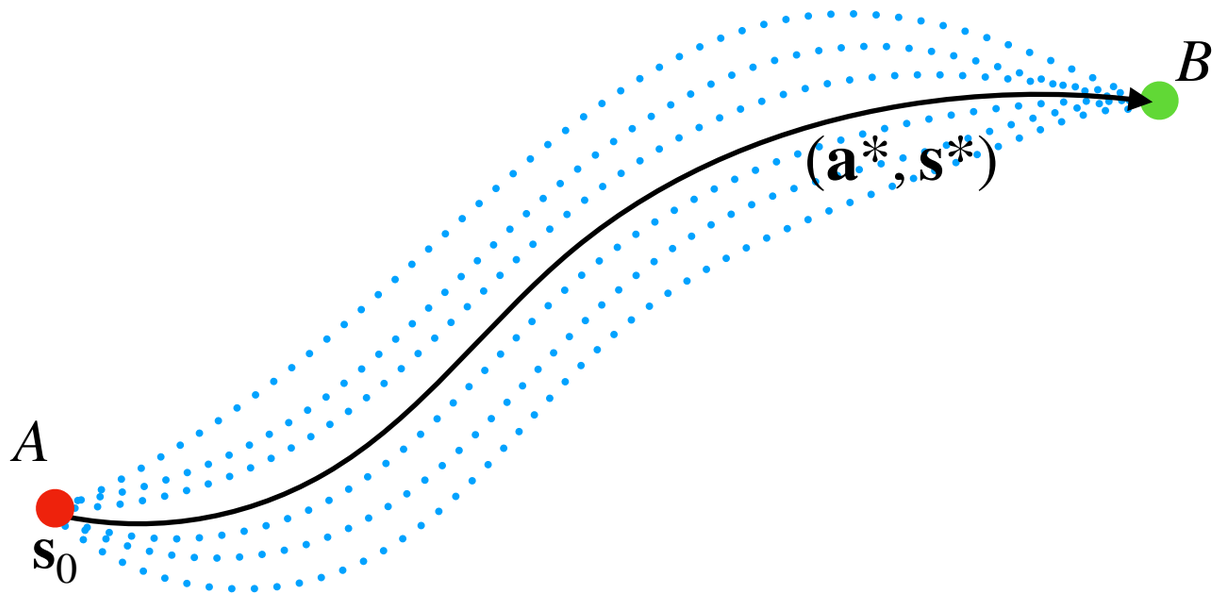
Solution 1: Dynamic programming



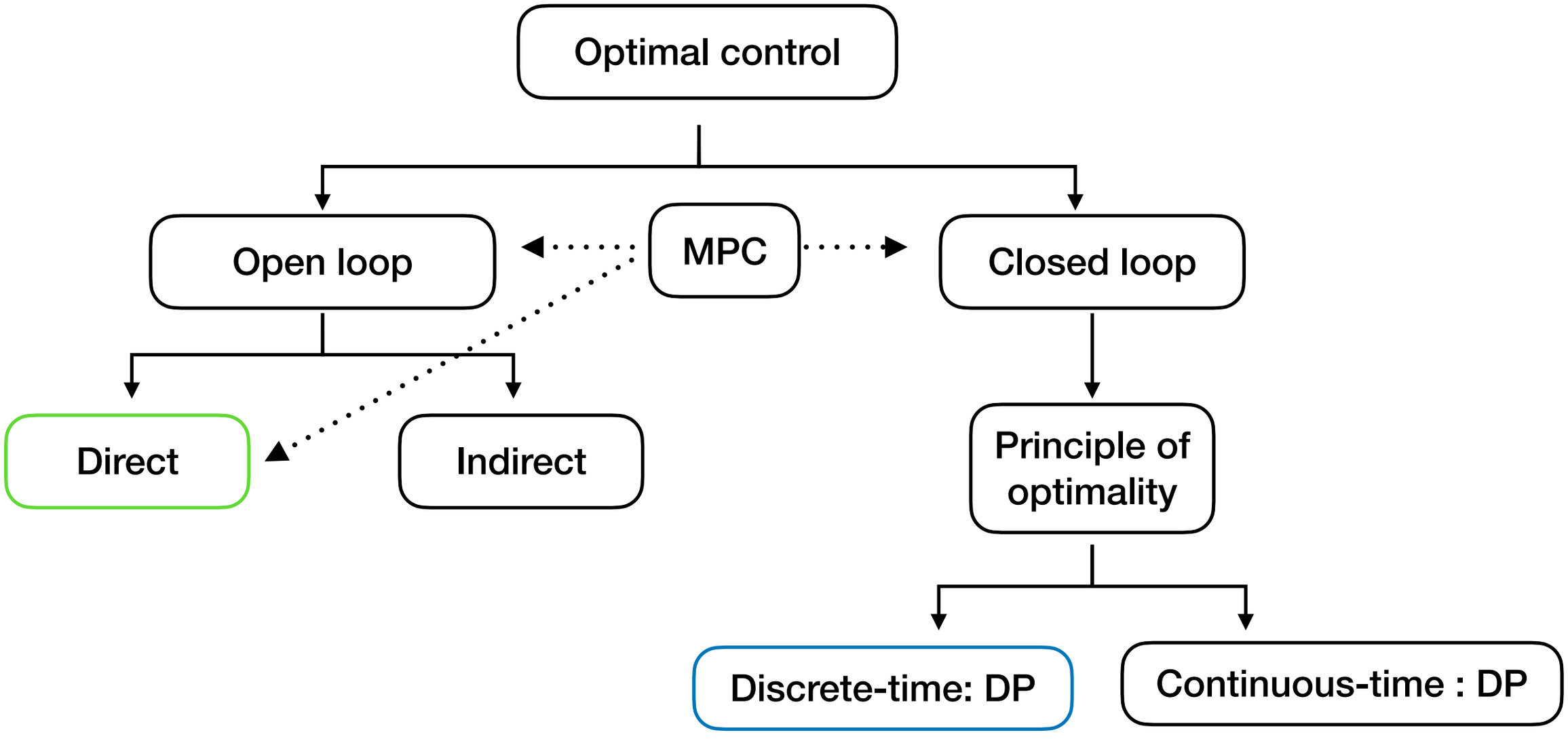
- Dynamic Programming (Principle of Optimality - sufficient condition)
  - ➡ Compositionally of optimal paths
  - ➡ Closed-loop solutions: find a solution for all states at all times
- Solvable via Bellman equation in a backward recursive fashion
- Algorithms as e.g. Value iteration, Policy iteration (see Sutton and Barto)
- No direct notion of constraints for states or actions!

Solution 2: Non-feedback control

- Calculus of Variations - Pontryagin Maximum Principle PMP (necessary condition)
- PMP turns functional minimisation in a function minimisation at each point in time
- Find a solution-sequence $(\mathbf{a}^*, \mathbf{s}^*)$ for a **given** initial state $\mathbf{s}_0$
- Can handle constraints e.g. $\mathbf{s}_t \in S$, $\mathbf{a}_t \in A$
- But: open loop cannot stabilise the system!

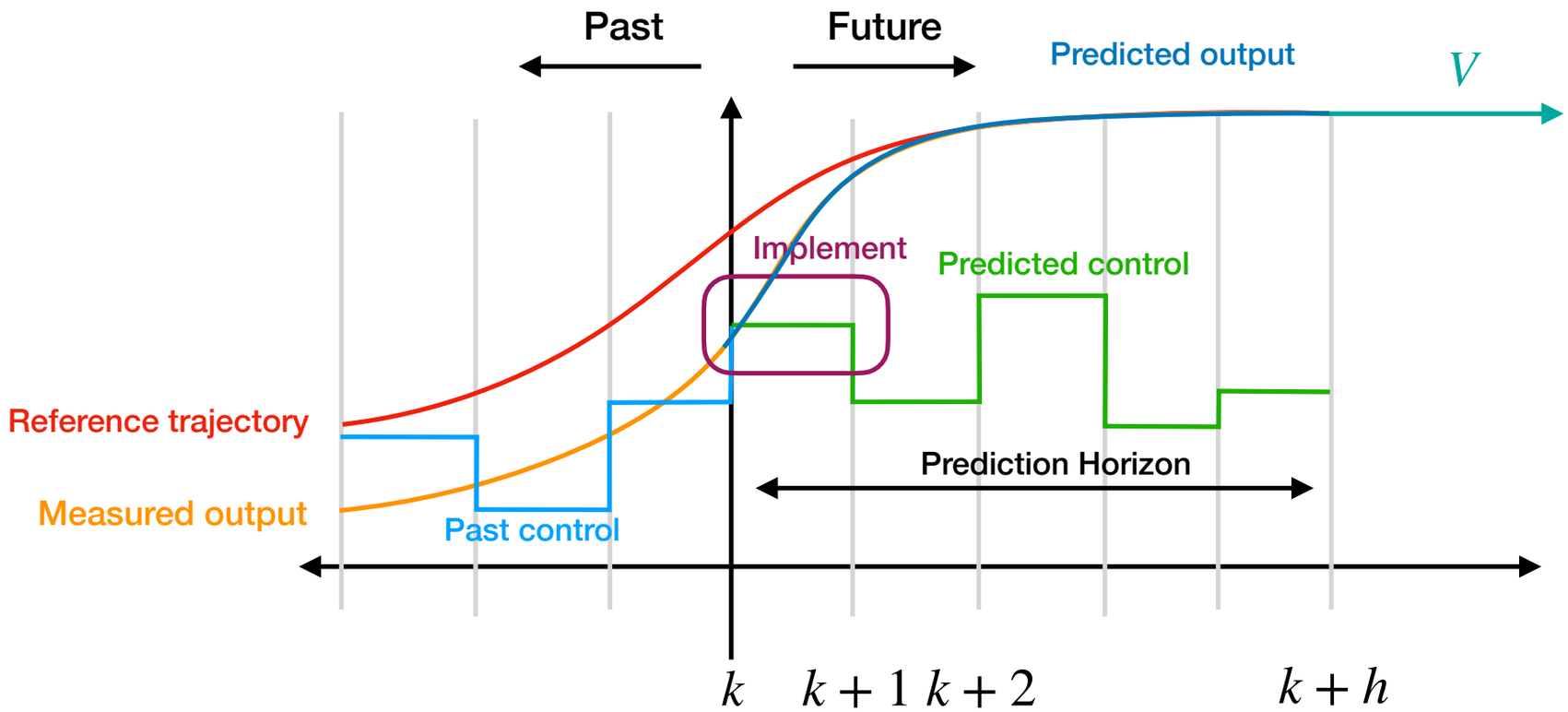


Best of both worlds - model predictive control (MPC)



Adapted from AA 203: Optimal and Learning-Based Control

Model predictive control (MPC)



Want to solve infinite (RL) optimisation problem:

$$\text{maximise}_{\pi_t} \lim_{T\to\infty} \mathbb{E}_{W_t}[\frac{1}{T}\sum_{t=0}^{T} R_t(S_t, A_t, W_t)]$$

$$\text{subject to: } S_{t+1} = f_t(S_t, A_t, W_t)$$

$$A_t = \pi(S_t)$$

$$S_0 = s$$

Instead MPC computes an open loop control on finite horizon:

Optimise for finite horizon

$$\text{maximise}_{\{a_t\}} \mathbb{E}_{W_t}[\sum_{t=0}^{H-1} R_t(S_t, A_t, W_t) + V(S_H)]$$

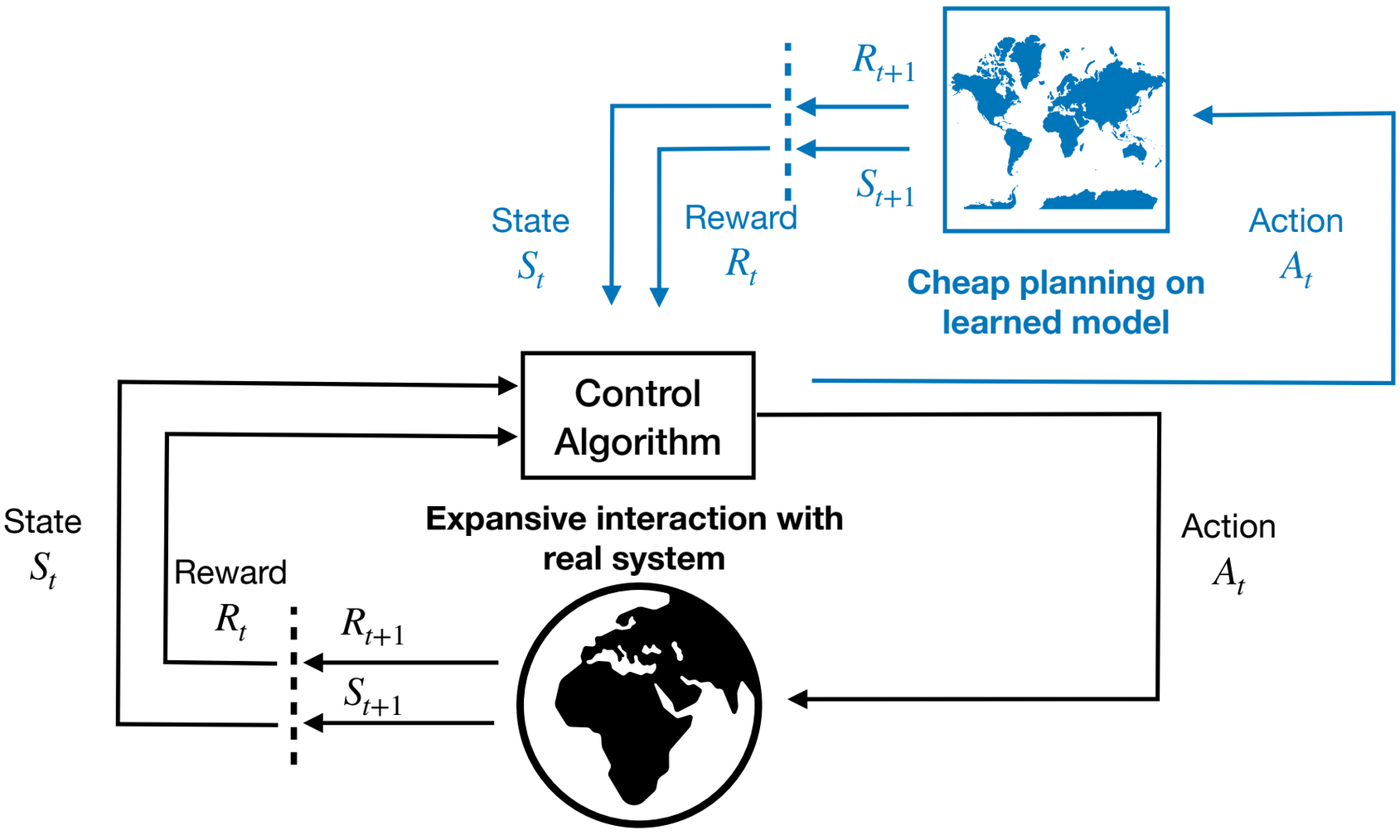
$$\text{subject to: } S_{t+1} = f_t(S_t, A_t, W_t)$$

$$S_0 = s$$

Final cost performance for robustness

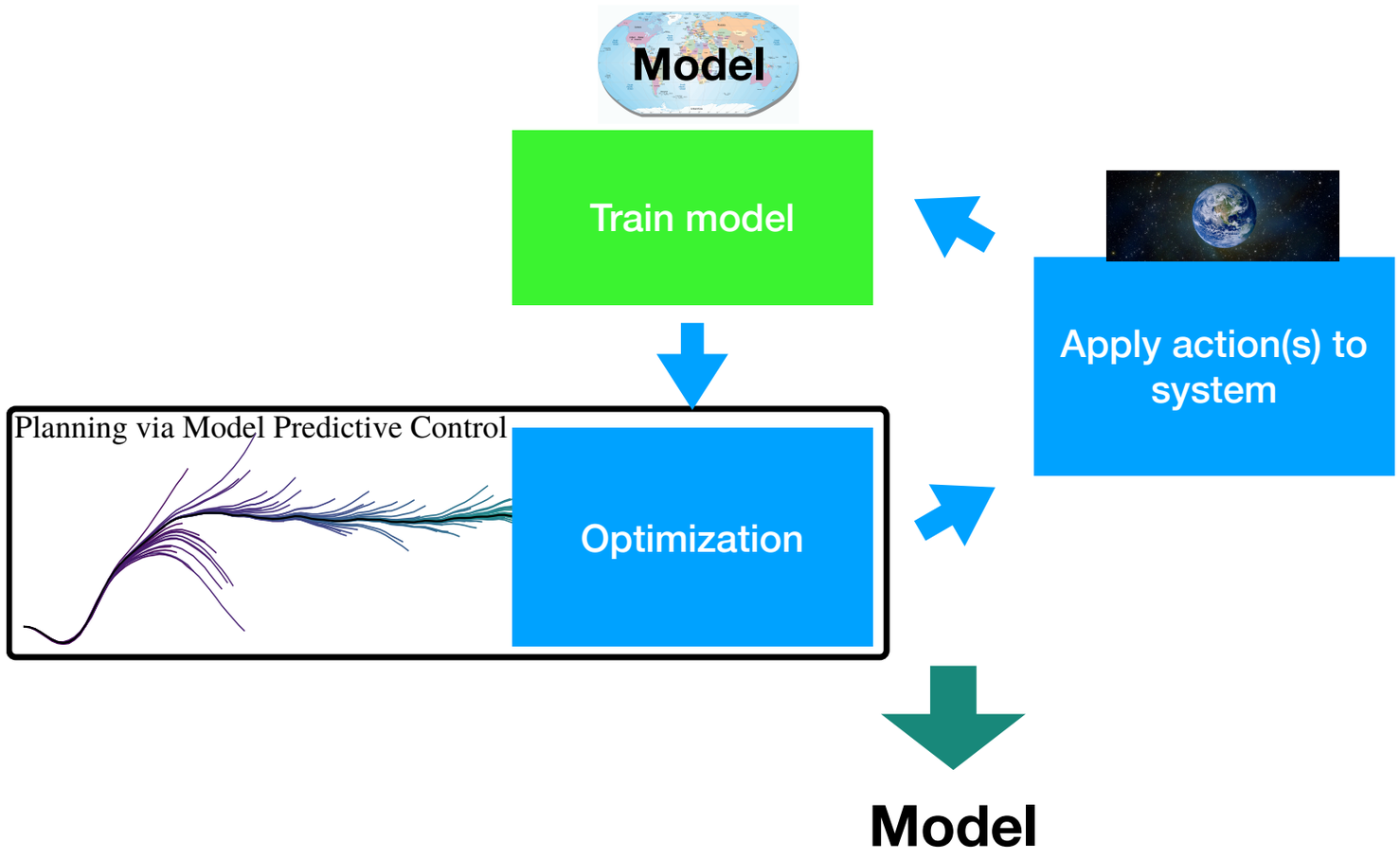
W_t **is a random variable**

If we don't have a model? - we learn it!



Model-based RL

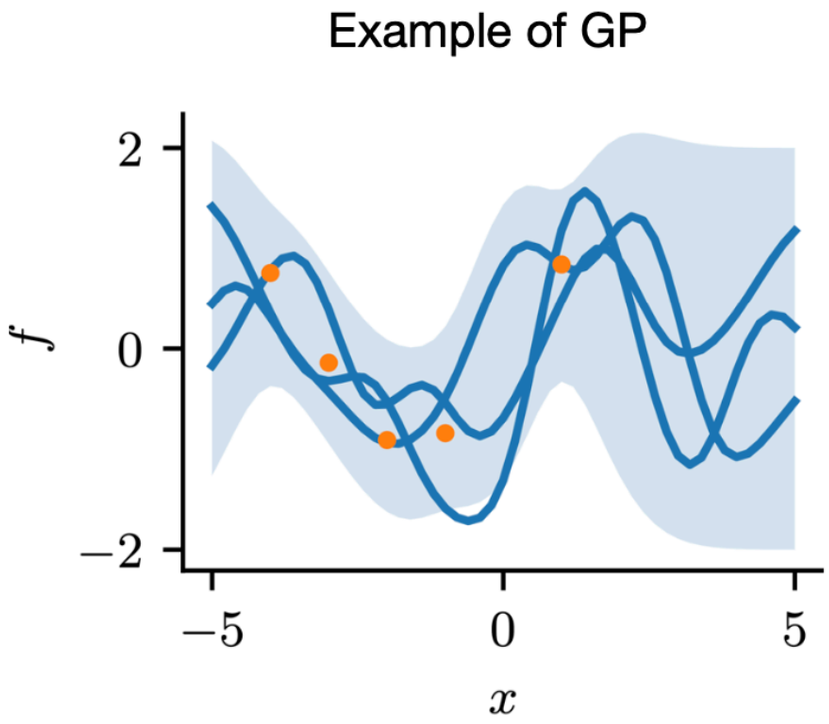
Example: Data-driven MPC



- Setup the dynamics-reward model
- Use PMP to obtain sparse optimization with gradient information
- Choose optimization algorithm
- Consider safety (constraints)
- Set up training

Using non-parametric function approximators

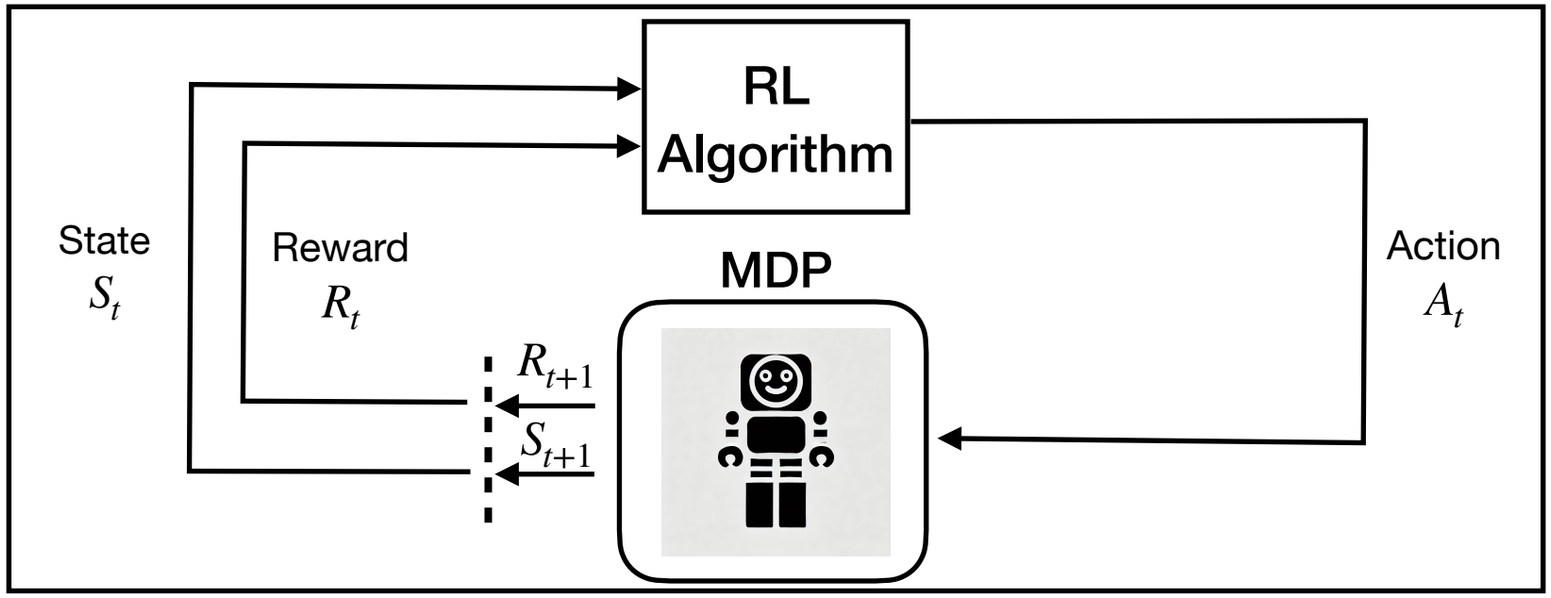
- Learn the model from data:
  - ➡ Aleatoric uncertainties - unavoidable
  - ➡ Epistemic uncertainties - due to lack of data -minimise model bias
- Gaussian processes (GPs) are used assuming $\mathbf{s}_{t+1} = \mathbf{f}(\mathbf{s}_t, \mathbf{a}_t, W_t)$ and $W_t \sim \mathcal{N}(0,\sigma)$
- Use Radial-basis-function Kernel - allow for analytical propagation of uncertainties (deterministic propagation of the mean $\mu(s_t)$ and the covariance $\Sigma(s_t)$)
- $\mathbb{E}[r(s_t, a_t)] = \int r(s_t, a_t)\mathcal{N}(s_t \mid \mu_t, \Sigma_t)ds_t$



Outline

- Practical questions in RL
- Some remarks on RL
- What if you know how the world works?
- **The basis of the decisions**
- Challenges with the basic algorithms
- Some philosophical perspectives
- Final remarks

The problem

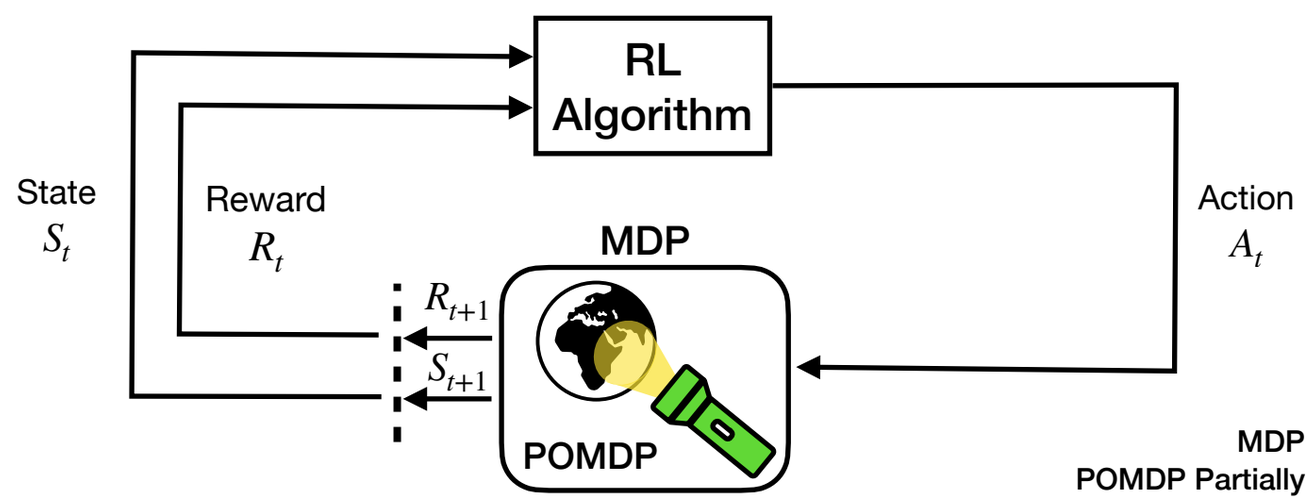


MDP Markov decision process
POMDP Partially observable Markov decision process

But...

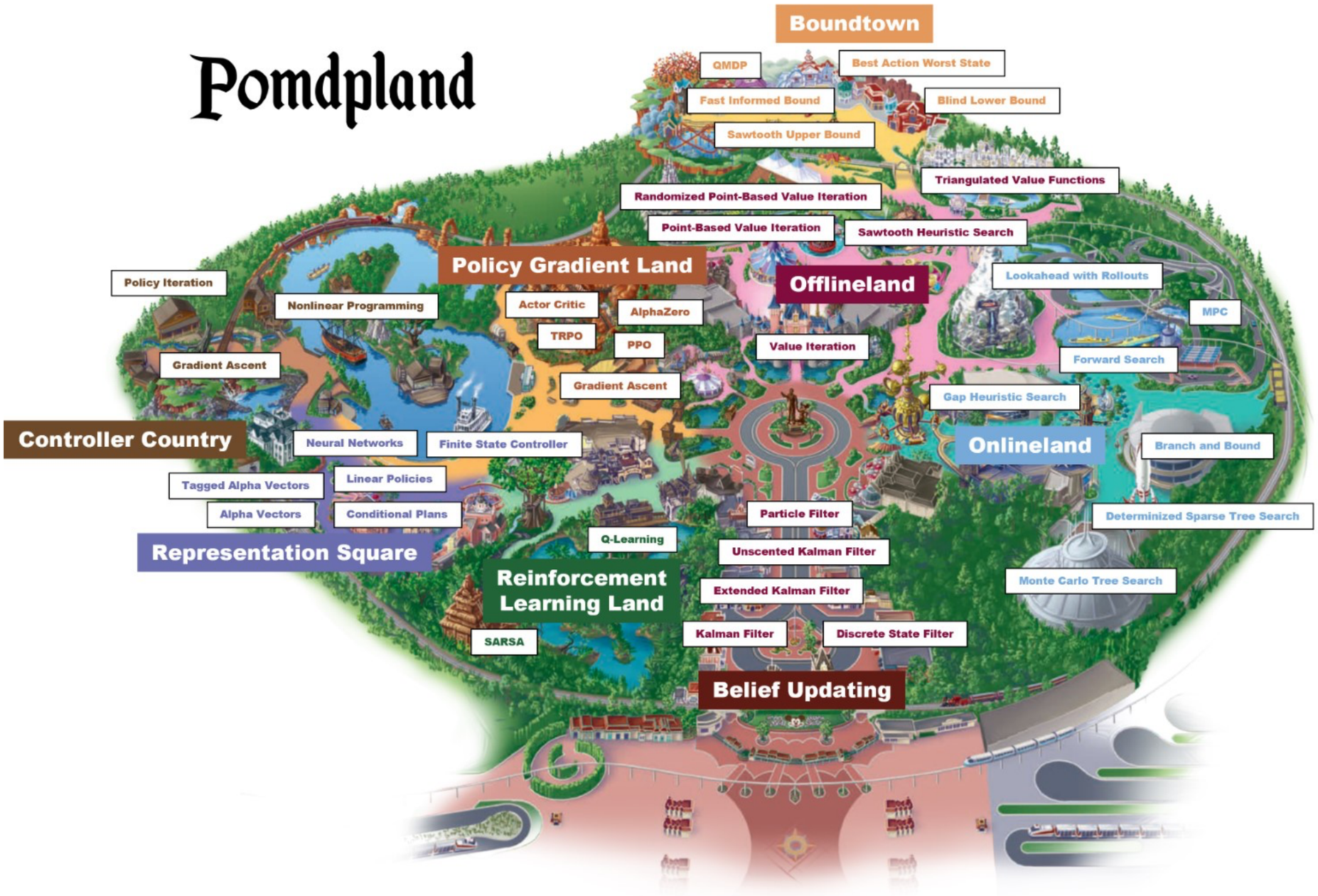
- Definition an MDP is a tuple $(\mathbb{S}, \mathbb{A}, P, R, \gamma, \rho)$:
  - $\mathbb{S}$ is a set of states ($s \in \mathbb{S}$)
  - $\mathbb{A}$ is a set of actions ($a \in \mathbb{A}$)
  - P is a dynamic model (transition kernel), defined for every action $P(s_{t+1} = s' \,|\, s_t = s, a_t = a)$ - in our case it is linear in state and action
  - R is a reward function $R(s_t = s, a_t = a) = \mathbb{E}[r_t \,|\, s_t = s, a_t = a]$
  - Discount factor $\gamma \in [0,1]$

- Definition partially observable MDP (POMDP) is a tuple $\mathcal{M} = (\mathbb{S}, \mathbb{A}, \mathbb{O}, P, R, E, \gamma, \rho)$:
  - $\mathbb{O}$ is the set of observations ($o \in \mathbb{O}$)
  - E is an emission function defining the distribution $E(o_t \,|\, s_t)$



MDP Markov decision process
POMDP Partially observable Markov decision process

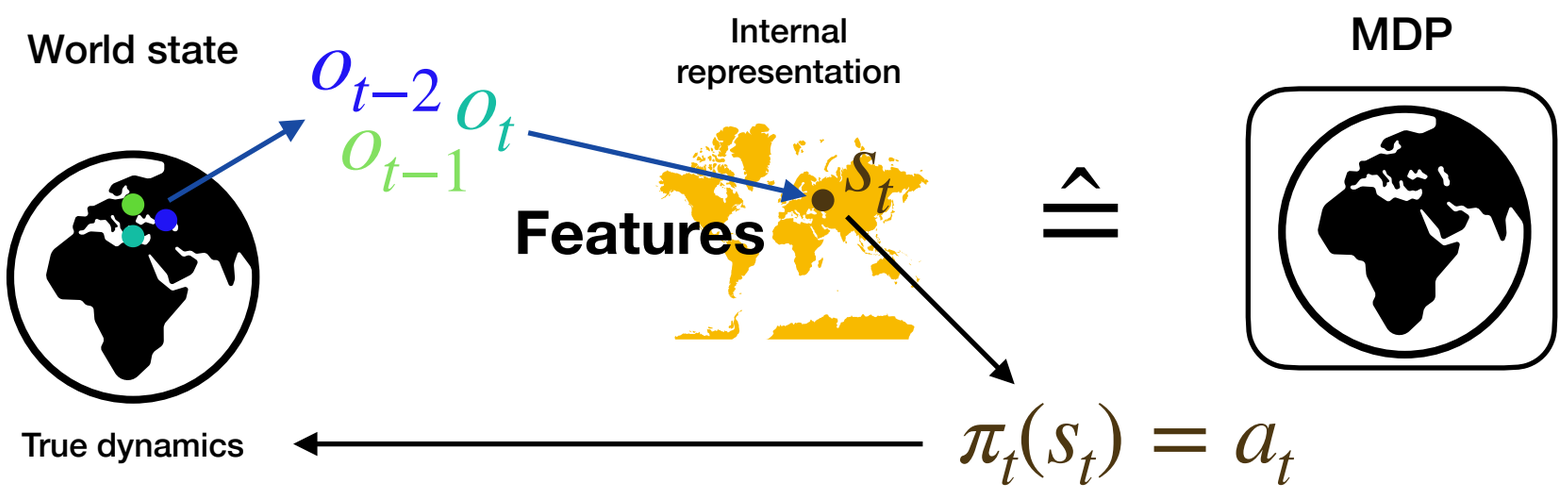
Wellcome to POMDPs



From Mykel Kochenderfer

Problem design - capture the right thing

- Solve an SDM problem: Information→Decision→Information→Decision→…
- Generally stochastic!
- Consequently we build a feedback system not planing too far in the future:
  - Define a **<u>state</u>** $s_t = h_t(o_t, a_{t-1}, o_{t-1}, a_{t-2}, o_{t-2} \dots)$, as a function holding **<u>sufficient statistics</u>** until time step t for a decision - (example pong)
  - Decision based on s_t via: $a_t = \pi_t(s_t)$ - the policy - optimise an expected aggregate of future rewards

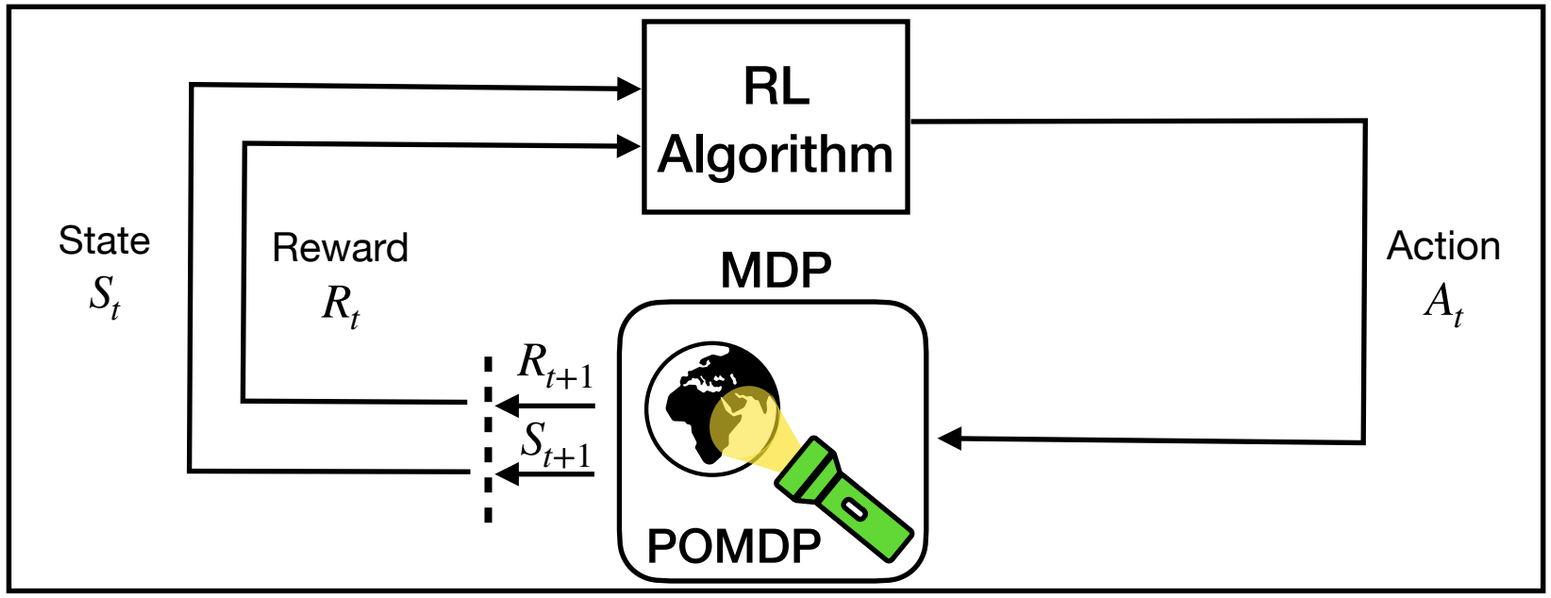


- Rarely the observation o is the state s, the world state is, but often we assume it is certainty equivalence!
- POMDP ⇒ MDPs!

How bad is it?

- Linear POMDP: believe state - $O_t = h_t(S_t, A_t, W_t)$
  - ➡ Static output feedback is NP hard (linear in O_t and dynamics)
  - ➡ General POMDPs are PSPACE hard
- There are ways out - separation principle:
  - ➡ Filtering $\hat{s}_t = f(\{o_t\})$ - prediction problem
  - ➡ Action based on certainty equivalence
  - ➡ Optimal filtering - if dynamics are linear and noise is Gaussian - Kalman filtering - general belief propagation - LQG
  - ➡ Kalman filtered state - optimal in estimation and control
  - ➡ Estimate state with prediction $S_t = h(\tau_t)$, τ_t are time lags

And that's not all…

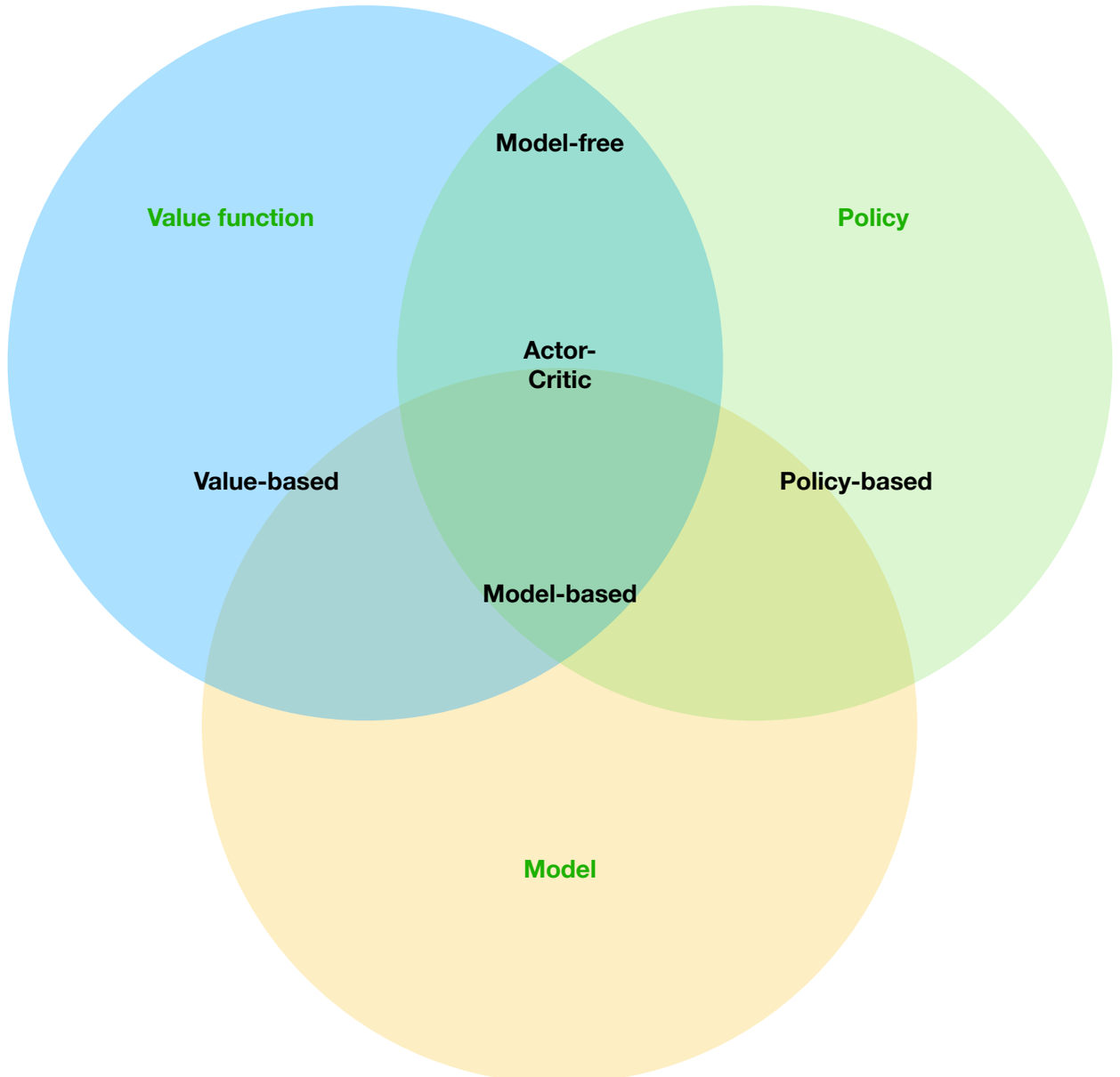


MDP Markov decision process
POMDP Partially observable Markov decision process

Outline

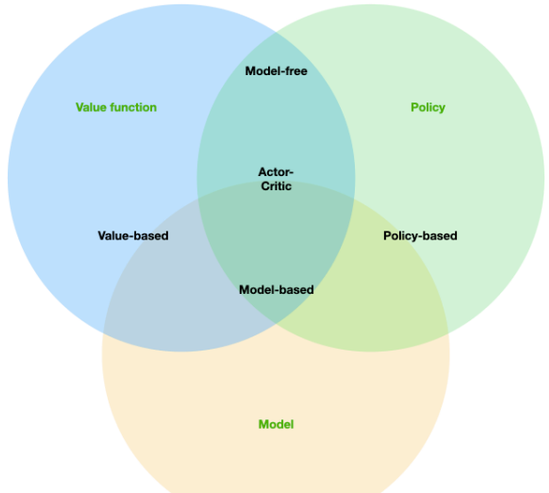
- Practical questions in RL
- Some remarks on RL
- What if you know how the world works?
- The basis of the decisions
- **Challenges with the basic algorithms**
- Some philosophical perspectives
- Final remarks

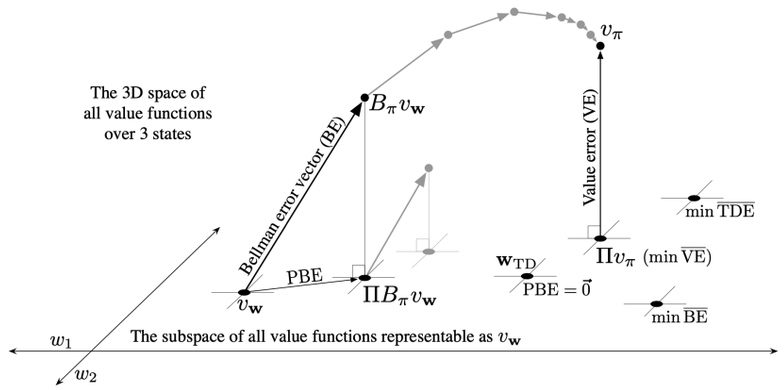
Different approaches



Stability and hyperparameter tuning

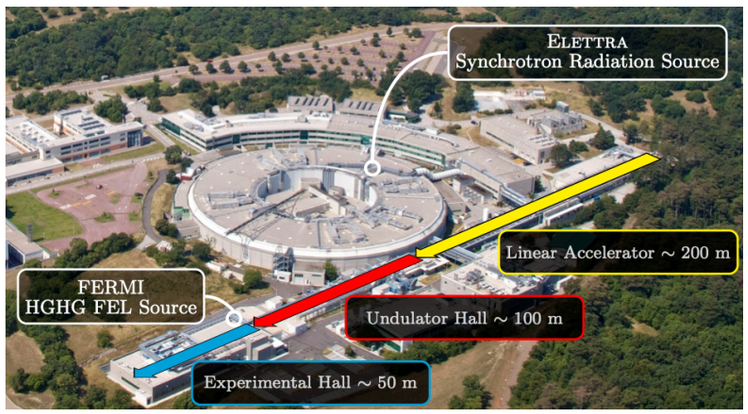
- Designing stable RL algorithms is very difficult
- Q-learning/state-value function estimation - deadly triad:
  - ➡ Fitted Q-learning/state-value functions with deep neural networks are typically not contractors → no convergence guarantees!
  - ➡ Many parameters for stability: target network update, replay buffer size, clipping, learn rate,…
  - ➡ What holds from SL in deep learning?
- Gradient-based methods/reinforce/likelihood ratio:
  - ➡ Very high variance gradient estimator
  - ➡ Many samples (exponential?)/complex baselines
  - ➡ Parameters: Batch-size, learning rate, design of baselines
- Model-based RL Algorithms:
  - ➡ Model class and fitting method
  - ➡ Optimising the policy with respect to the model is not trivial (back-propagation through time), dyna-style inherits problems of model-free approaches
  - ➡ More subtle challenges: policy shows tendency to exploit the model (exploitation)

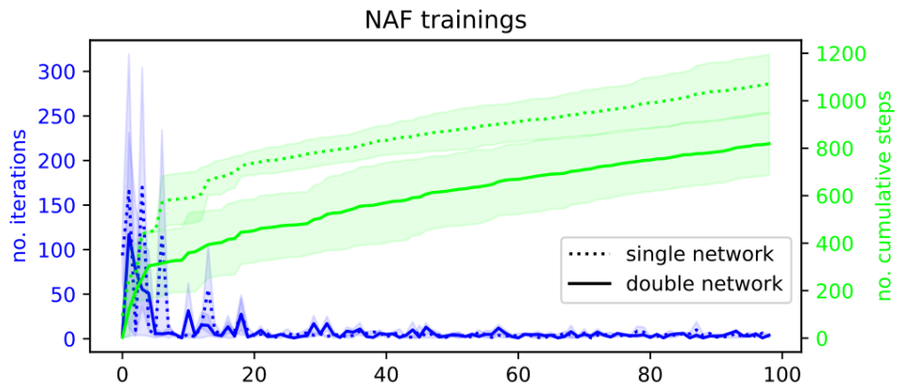




Hyper-parameters

- You can't do hyper-parameter sweeps in real applications
  - ➡ How representative is a simulator? (Usually not very)
- Sample complexity = time to execute algorithm times number of iterations for sweeps
  - ➡ Stochastic search and gradient-based optimisation
- Can we develop more stable algorithms that are not so sensitive to the hyper-parameters?





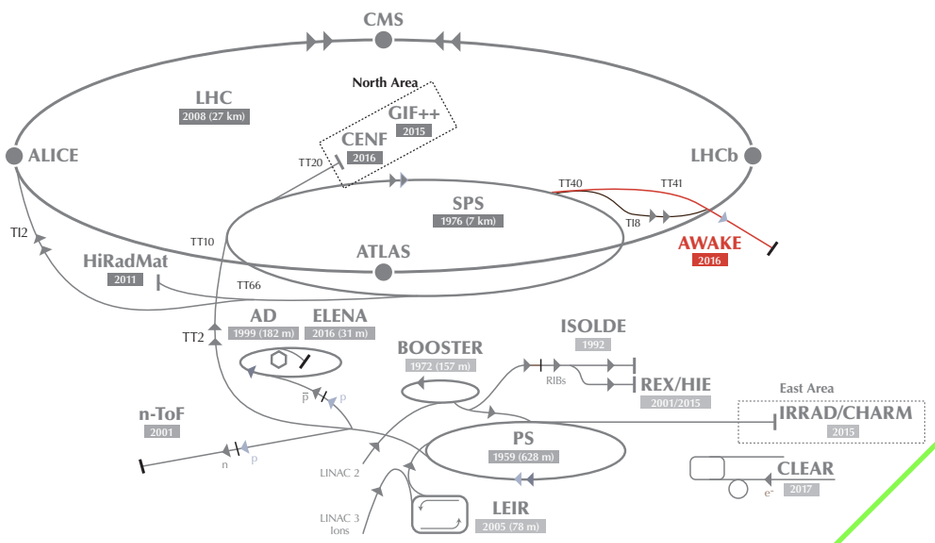
What can research do?

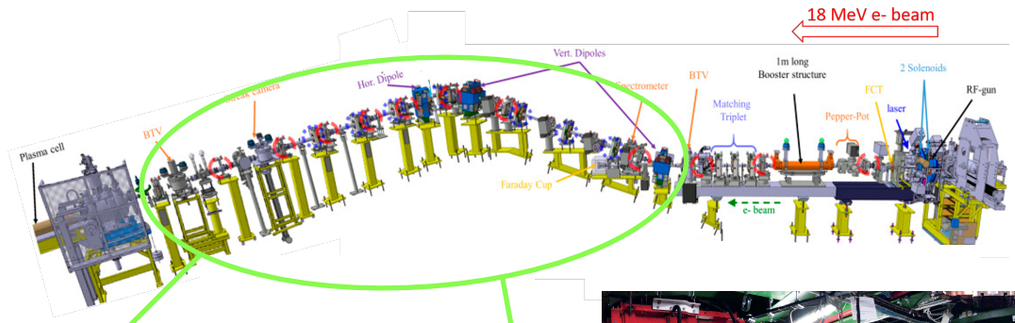
- Algorithms with favourable improvement and convergence properties
  - ➡ TRPO
  - ➡ Safe RL and High-confidence Policy improvement http://proceedings.mlr.press/v37/thomas15.html (batch guarantees)
- Algorithms that adaptively adjust the parameters
  - ➡ E.g.: Q-prop https://arxiv.org/pdf/1611.02247.pdf (stability of PG and efficiency of Q-learnig)

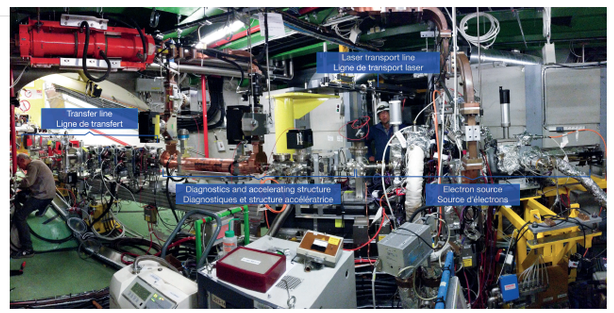
Not important to break benchmarks, but essential to make RL a reliable tool in the real world!

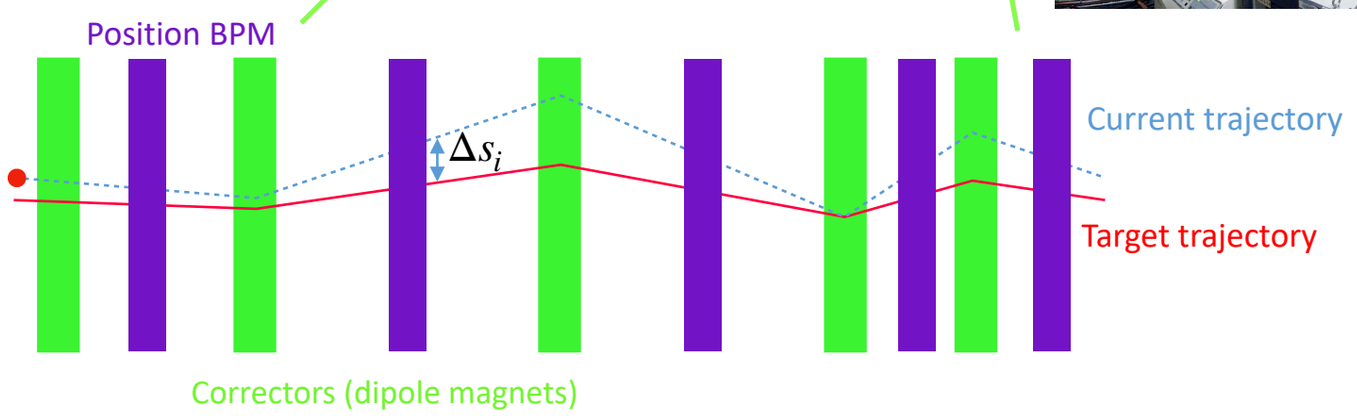
- More research needed!

Example: CERN Advanced WAKe field Experiment









- AWAKE electrons - start 5 MV (RF gun), accelerated to 18 MeV transported through beam line of 12 m to the AWAKE plasma cell.
- Vertical 1 m step and a 60° bend bring electron beam parallel SPS proton beam shortly plasma cell.
- The trajectory is controlled with 10 horizontal and 10 vertical steering dipoles according to the measurements of 10 beam position monitors (BPMs).
- We can simulate the physics easily.

Sample efficiency on AWAKE

Evolutionary/ gradient free

- Derivative free methods: (NES, CMA,..)

On-policy Policy gradient

- 10 x Online methods (A3C)
- 10 x Policy-gradient methods (TRPO)

Actor-critic

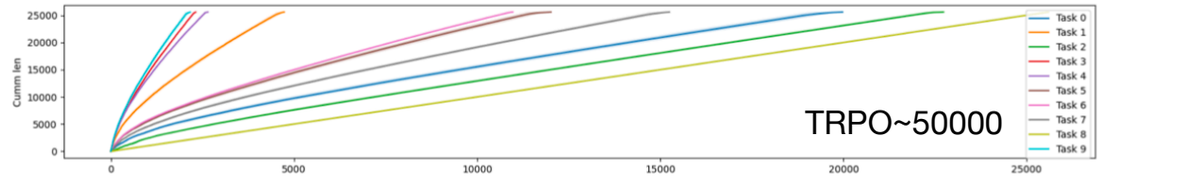
- 10 x Replay-Buffer + Value function estimation (Q-Learning, DDPG, TD3, NAF, SAC,…)

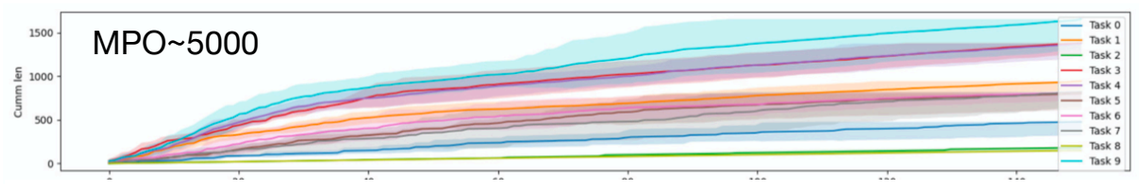
Off-policy Q-learning

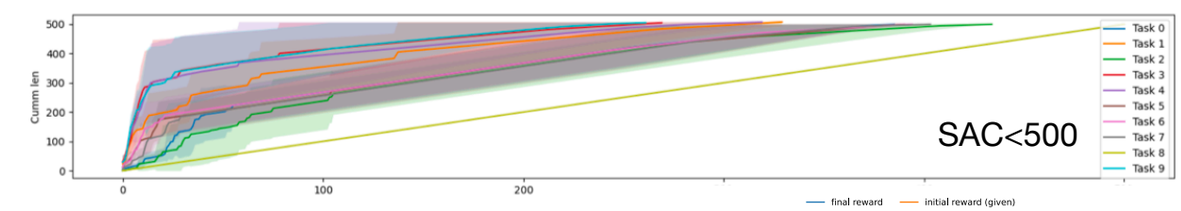
- 10 x Model-based RL methods (MPO, Guided Policy Search, Dyna)

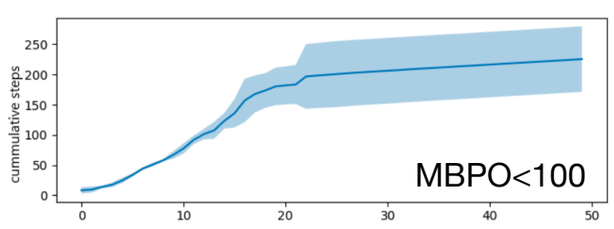
Model-based

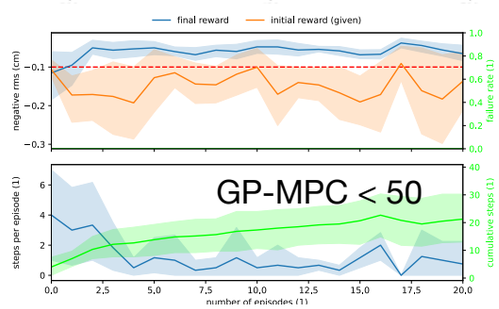
- 10 x Model-based shallow methods (no ANNs) Few shot GPs...

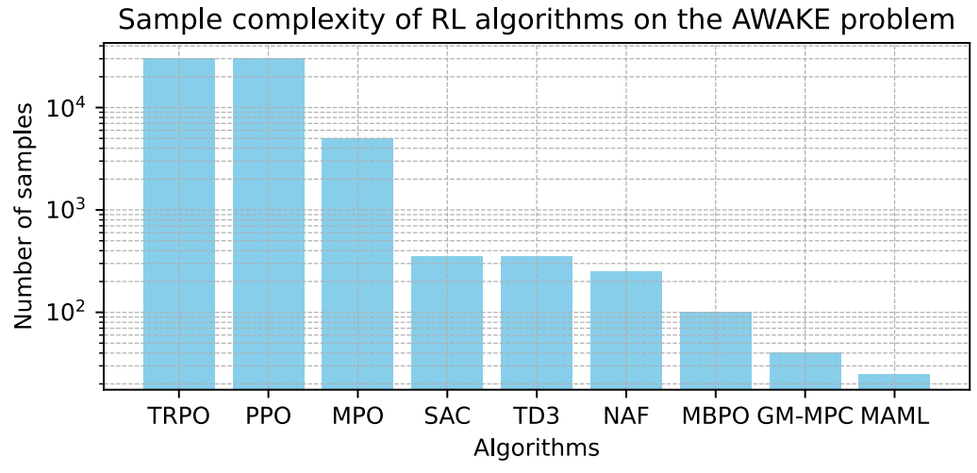








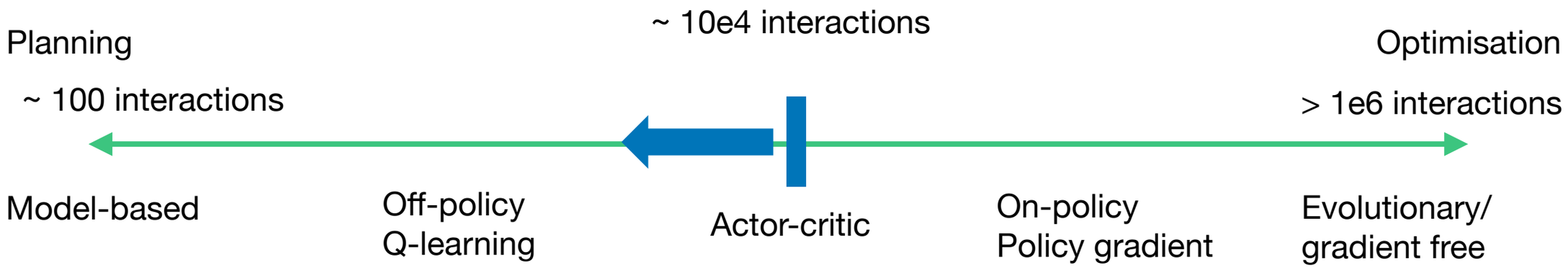


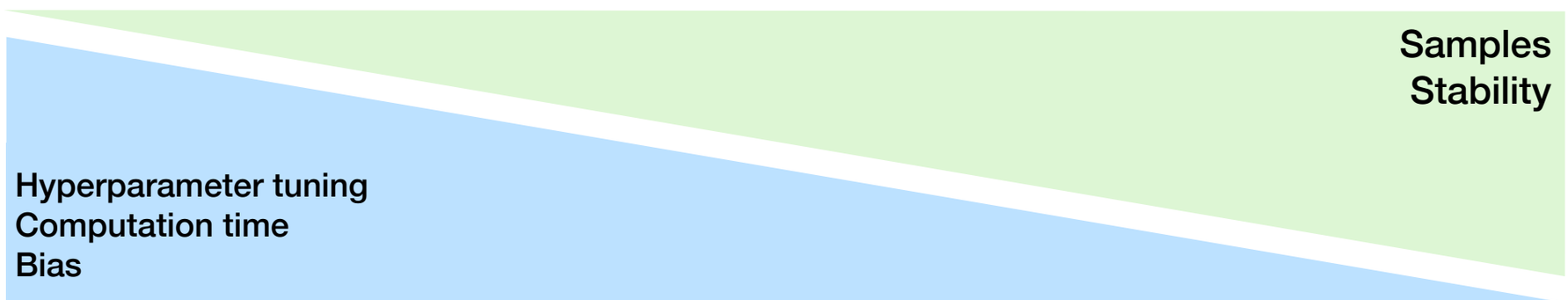


The challenge of sample complexity

- Algorithms take a very long time to solve problems
- Real application impossible
- Accurate simulations take a very long time to solve - slower than real time
- Range for real applications severely limited

Sample complexity vs. „tuning“ complexity





Modern Deep RL world

Make more stable!

Value based relies on Bellman updates

- Biased
- Can be offline
- Brittle
- No guarantees (deadly triad)
- Only Deterministic
- Sample efficient
- Examples: DDPG, TD3, DQN …

Actor/Critic

Reduce variance!

Policy based - stochastic optimisation

- High Variance
- Generally online
- Stable
- Local performance guarantees
- Allows for stochastic policies
- Low computational cost
- Examples: AC, TRPO, PPO…

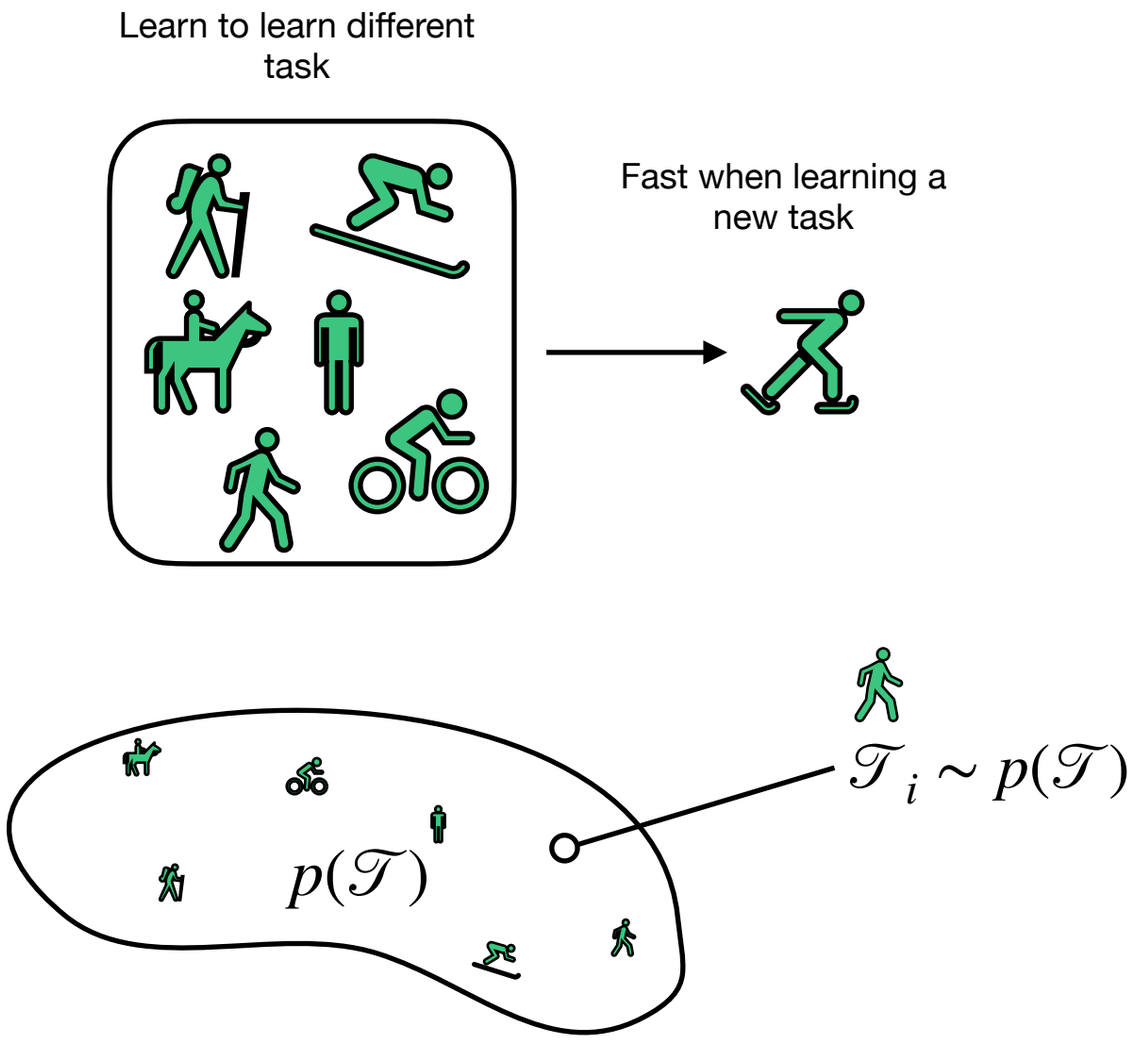
What can research do?

- Better model-based algorithms
- Development of faster algorithms
  - ➡ SAC (very efficient maximum entropy algorithm)
  - ➡ TD3 (simple and effective tricks to speed up DDPG-style algorithms)
- Reuse prior knowledge to speed up RL
  - ➡ RL^2: Fast reinforcement learning via slow reinforcement learning https://arxiv.org/pdf/1611.02779.pdf
  - ➡ Model-agnostic meta-learning http://proceedings.mlr.press/v70/finn17a/finn17a.pdf

Excursion: What is meta-RL?

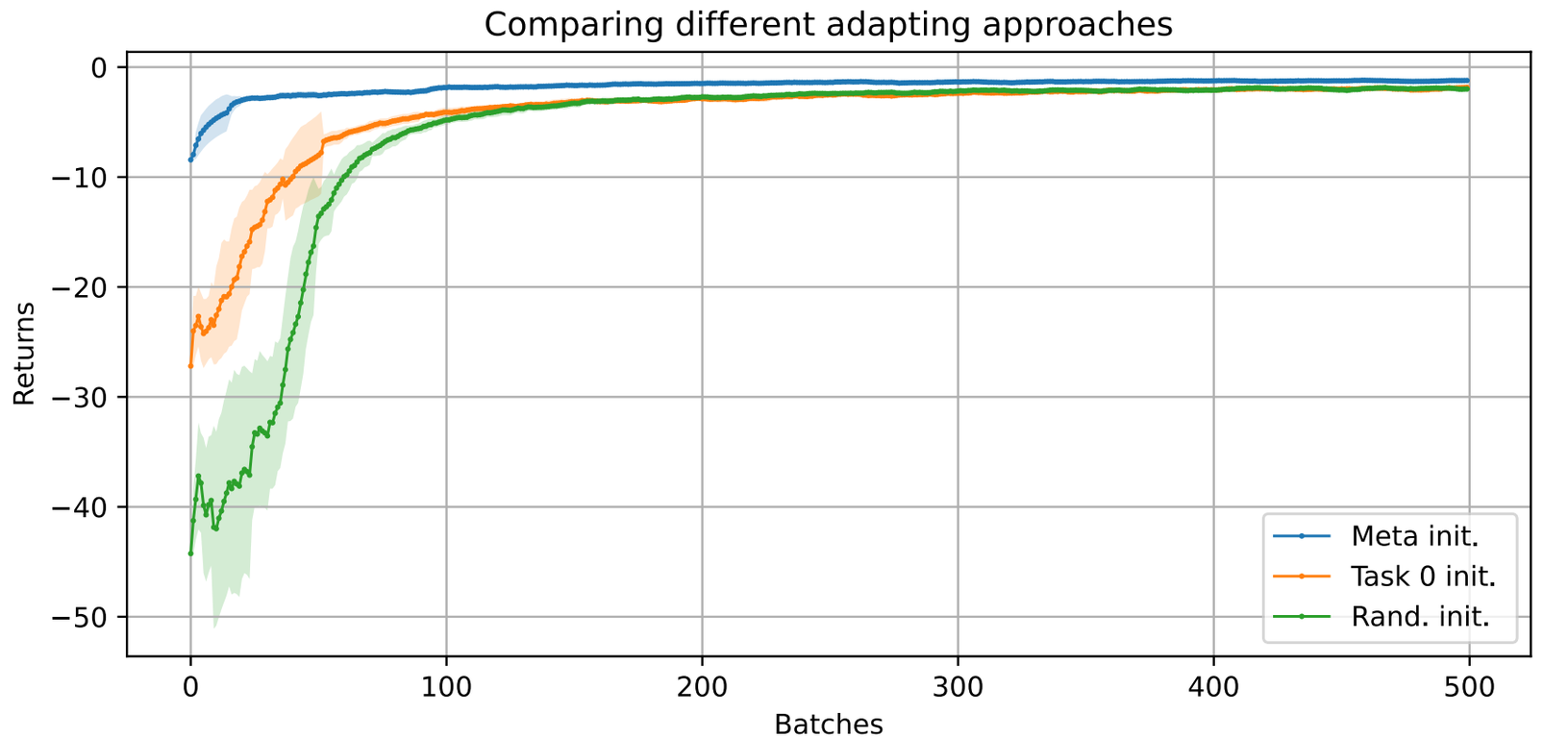
What is meta RL?

- Meta-learning = learn to learn
- Can the knowledge acquired from learning many different tasks be leveraged to expedite and improve the learning process for new tasks?
- Closely related to multi task learning: task provided explicitly
- Meta-learning: explicit focus on rapidly adapting to new task



Example meta RL on AWAKE

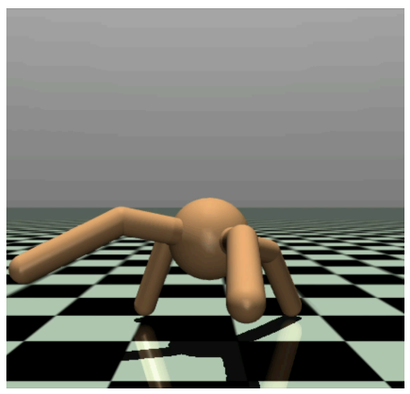
- Using a stable and monotonic algorithm
- Adapt quickly to actual setting - few shot adaption



Demonstrated on the real system

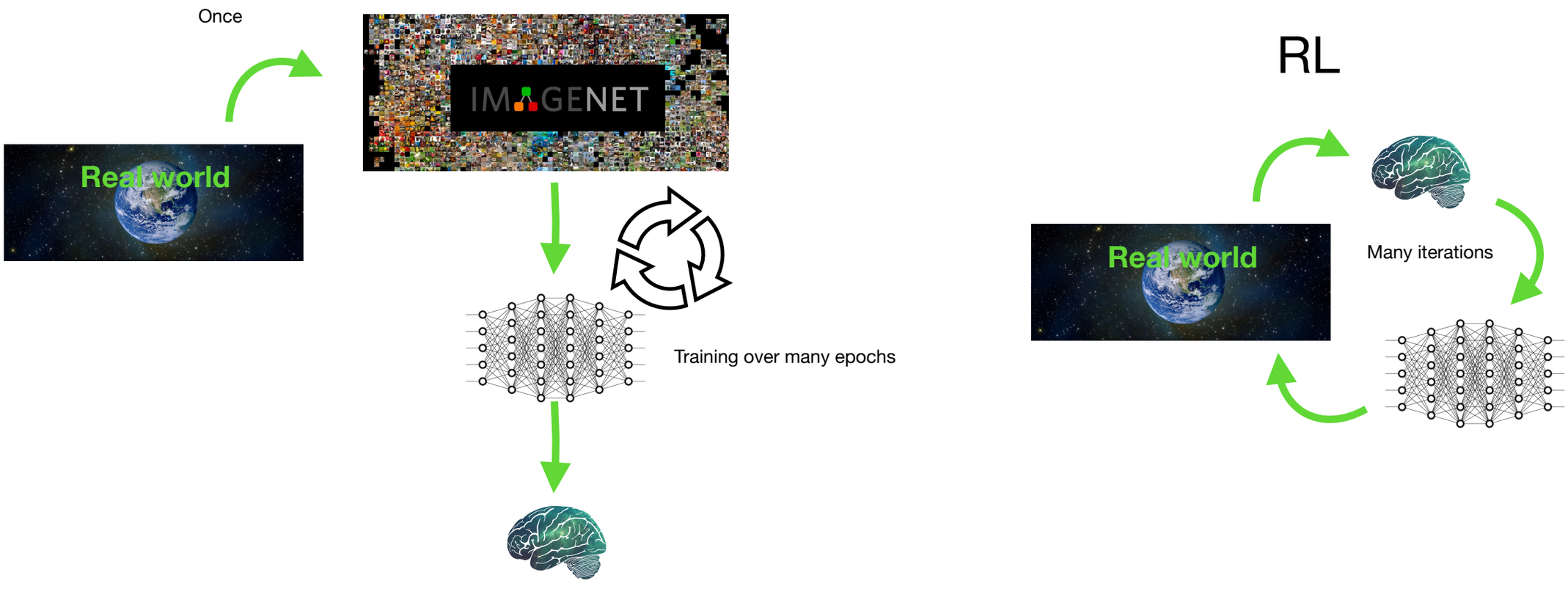
Scaling and generalisation

- Large Scale
- Emphasis on diversity
- Evaluated for generalisation



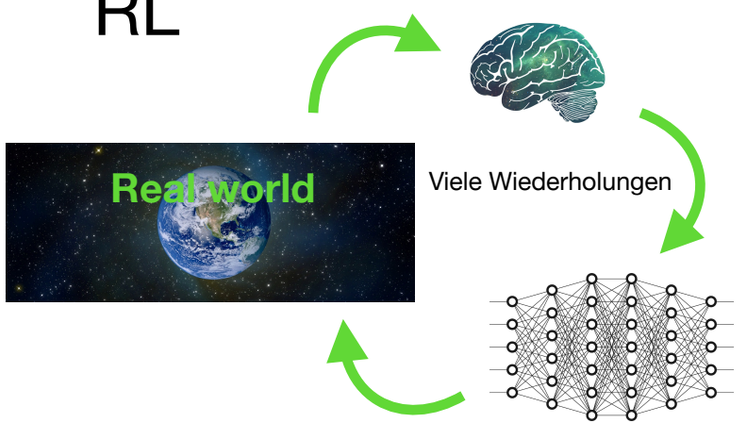
- Small Scale
- Emphasis on max. reward (mastery of a task)
- Evaluated for performance
- Where is the generalisation?

RL has a huge problem

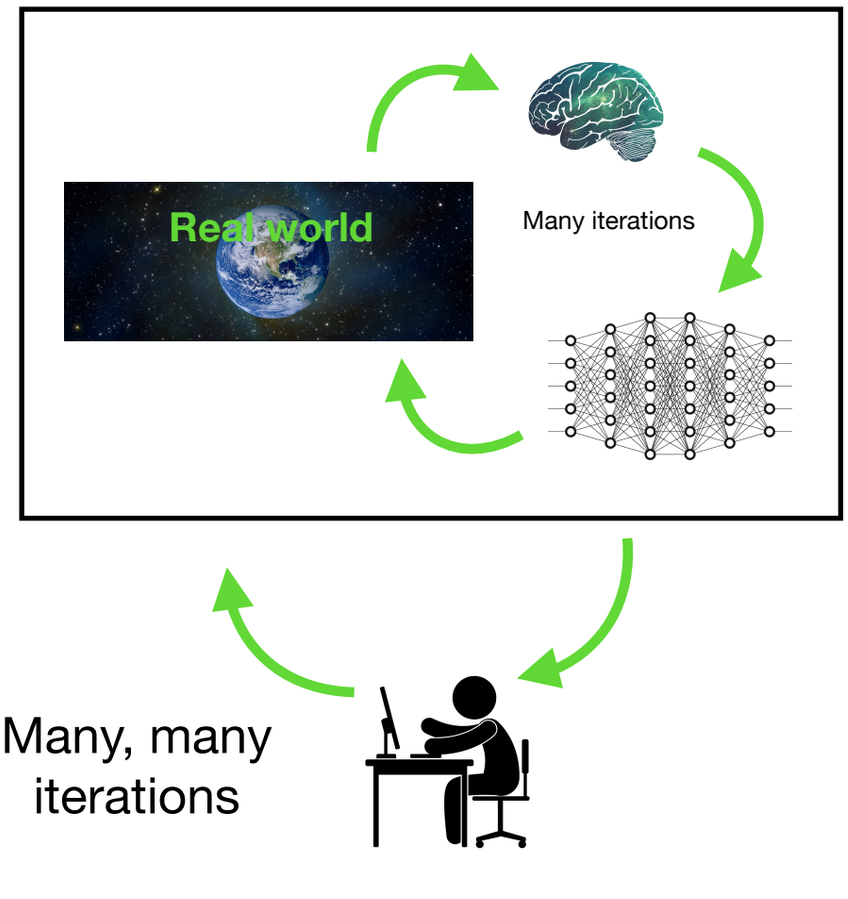


RL has a huge problem

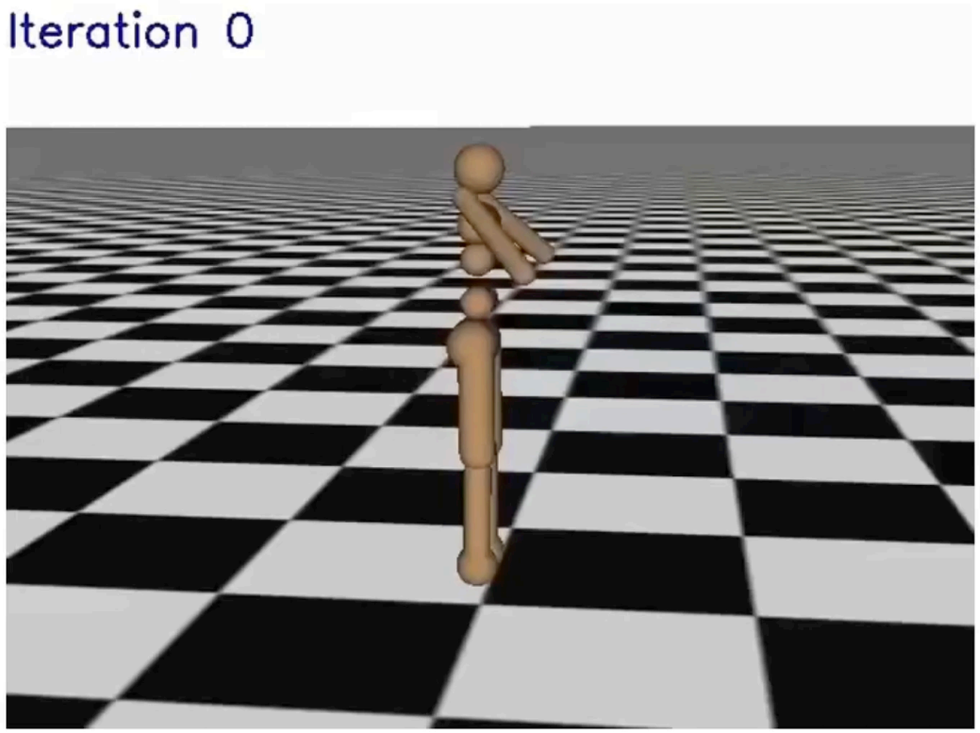
RL



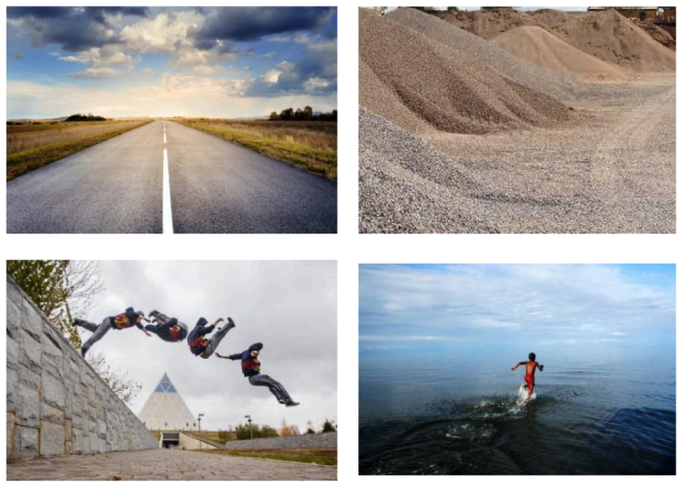
Actual RL



How bad is this problem?



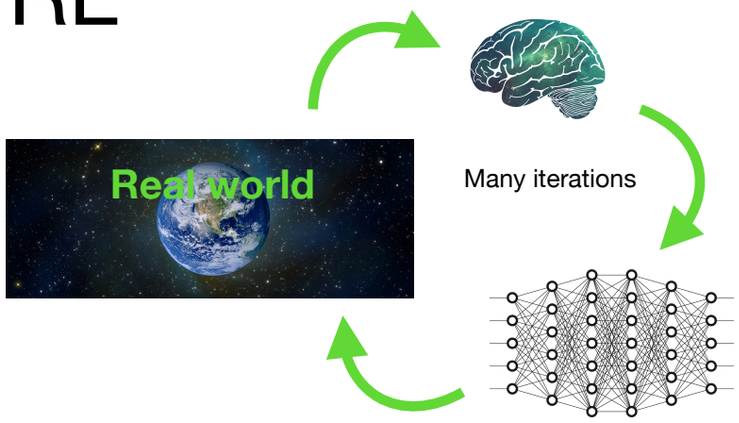
- Very good results
- ≈ 6 hours of real-time training
- ...running on an infinitely extended plane



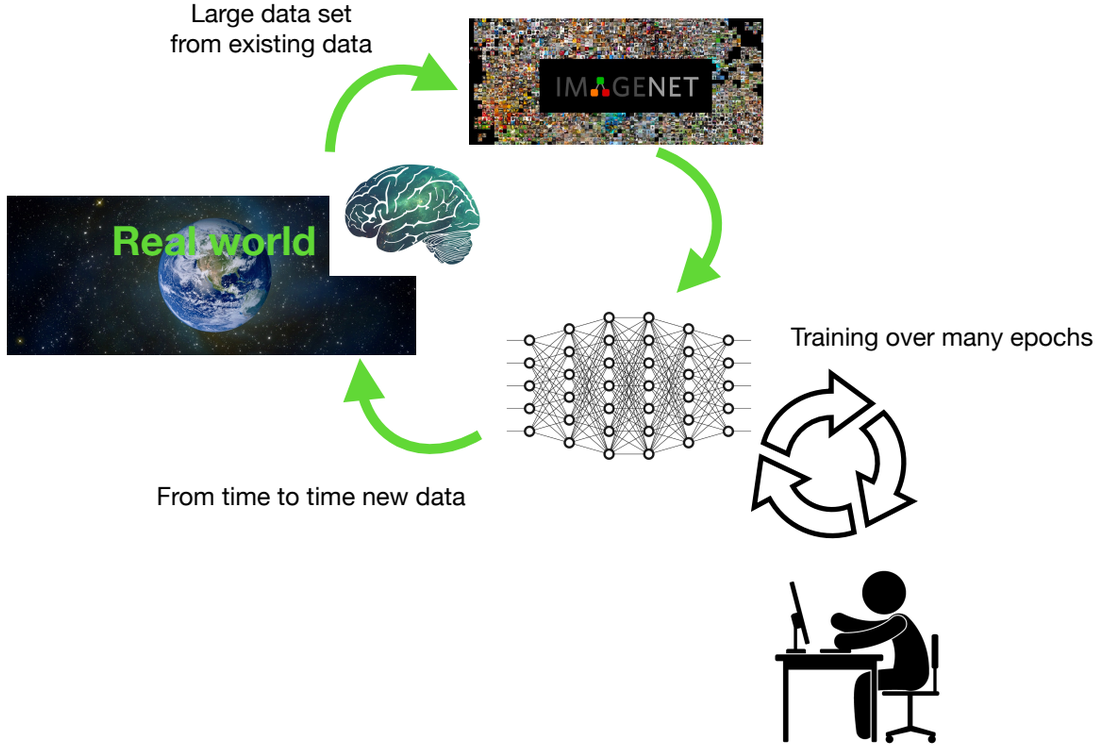
- The reality is not that simple!

Off-line (off-policy) RL?

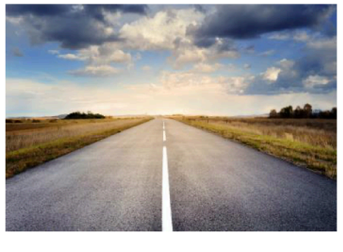
RL

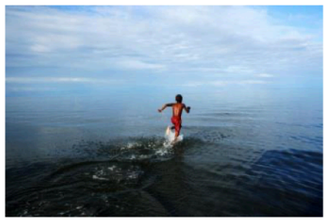


Off-policy RL

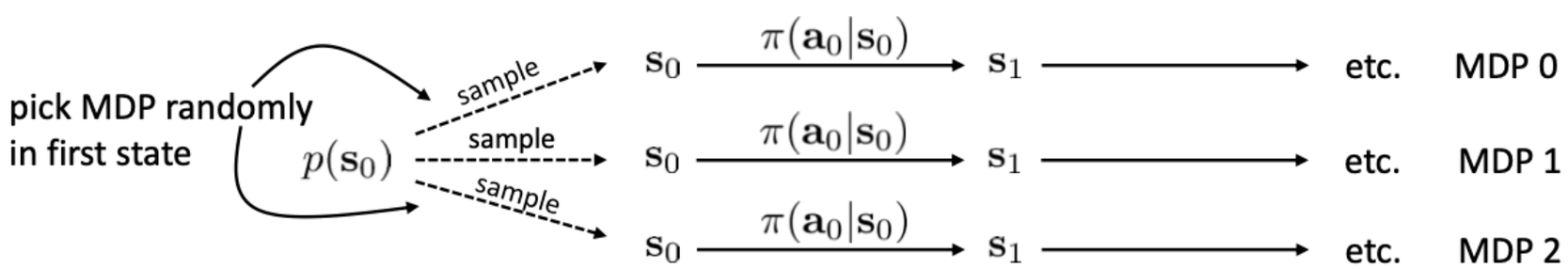


Single Task/Multi Task



- This is where generalisation comes from
- Perhaps no new assumptions necessary, but needs additional studies
  - ➡ Variance
  - ➡ Sample complexity
  - ➡ New methods?



Generalisation of multi-task learning

- Train on different tasks, then try to generalise or fine-tune
  - ➡ Policy Destillation https://arxiv.org/abs/1511.06295v2 (transfer to a smaller network)
  - ➡ Actor-Mimic https://arxiv.org/abs/1511.06342 (transfer from teacher network)
  - ➡ Model-agnostic Meta-Learning http://proceedings.mlr.press/v70/finn17a/finn17a.pdf
  - ➡ …
- Unsupervised or weakly supervised learning of different behaviours
  - ➡ Stochastic neural networks https://arxiv.org/abs/1704.03012v1 (prelearn skills)
  - ➡ RL with deep energy-based Policies https://arxiv.org/abs/1702.08165
  - ➡ …

Outline

- Practical questions in RL
- Some remarks on RL
- What if you know how the world works?
- The basis of the decisions
- Challenges with the basic algorithms
- **Some philosophical perspectives**
- Final remarks

Different perspectives

1. RL as engineering tool
2. RL and the real world
3. RL as ‚universal‘ learning

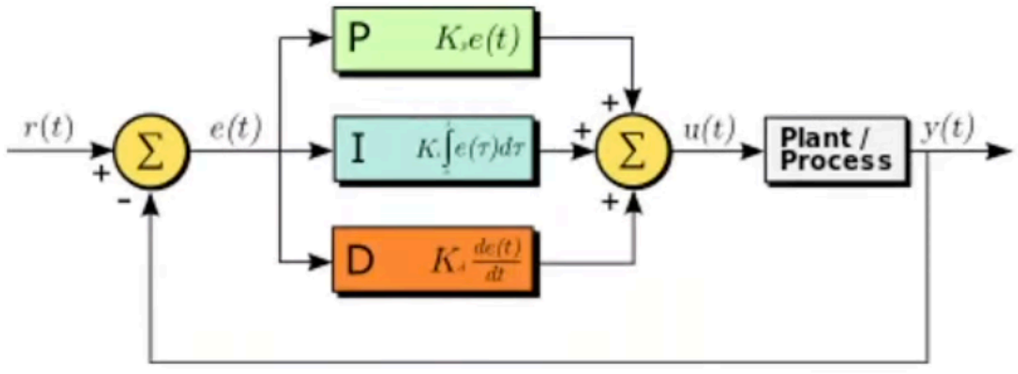
1. RL as engineering tool

$$\mathbf{r} = \mathbf{r}(t) = r\hat{\mathbf{e}}_r$$

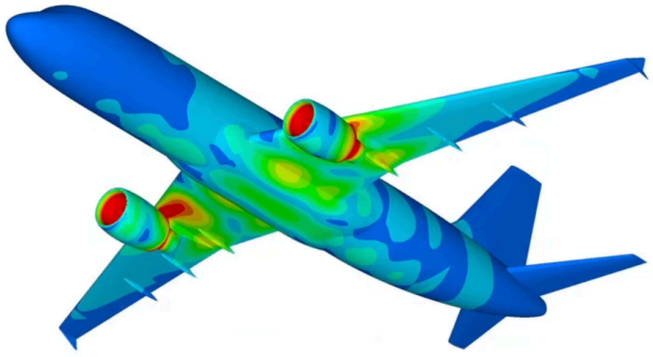
$$\mathbf{v} = v\hat{\mathbf{e}}_r + r\frac{d\theta}{dt}\hat{\mathbf{e}}_\theta + r\frac{d\varphi}{dt}\sin\theta\hat{\mathbf{e}}_\varphi$$

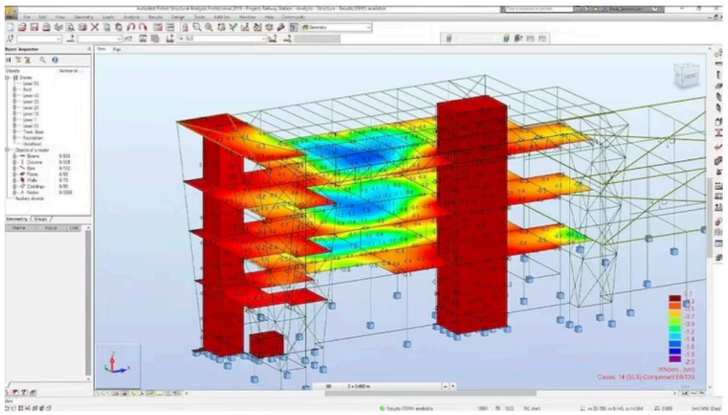
$$\mathbf{a} = \left(a - r\left(\frac{d\theta}{dt}\right)^2 - r\left(\frac{d\varphi}{dt}\right)^2\sin^2\theta\right)\hat{\mathbf{e}}_r + \left(r\frac{d^2\theta}{dt^2} + 2v\frac{d\theta}{dt} - r\left(\frac{d\varphi}{dt}\right)^2\sin\theta\cos\theta\right)\hat{\mathbf{e}}_\theta + \left(r\frac{d^2\varphi}{dt^2}\sin\theta + 2v\frac{d\varphi}{dt}\sin\theta + 2r\frac{d\theta}{dt}\frac{d\varphi}{dt}\cos\theta\right)\hat{\mathbf{e}}_\varphi$$

- Engineering as control
- Inverting physics



Characterisation and simulation





- Run RL in a simulator inverts the physical model using ML
- Powerful tool - inversion engine
- Simulations have to be developed

2. RL in the real world

- Moravec’s paradox

Natural Evolution

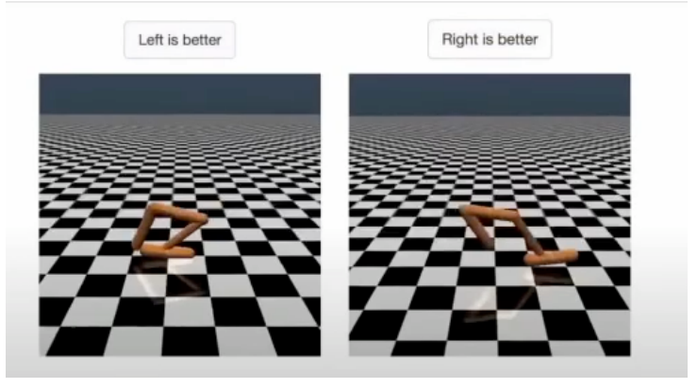
Time → Survive in the world Abstract thinking

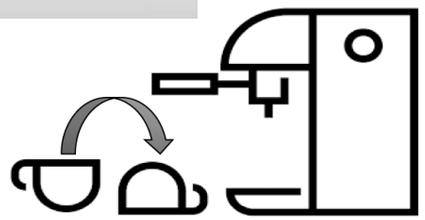
AI Evolution

Time →Abstract calculation Move in the world

Challenges for full real world training

- How do we tell the agents what we want them to do?
- How can we learn fully autonomously?
- How do remain as robust as the environment changes?
- What is the right way to generalise from experienced data?





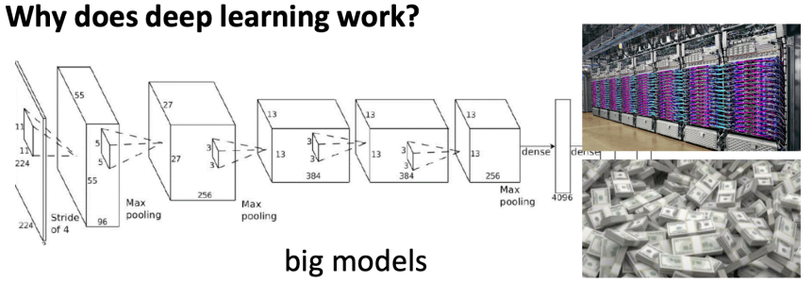
Why is this interesting?

- How do we engineer a system that can deal with the unexpected? Current AI systems are extremely bad at this
- **RL in principle can do this, and nothing else can**
  - ➡ It's exciting to see new solutions of the agent we don't expect
  - ➡ This requires the world they inhabit to admit novel solutions
  - ➡ This means that world must be complicated enough!
- To see **interesting emergent behaviour**, we must train our systems in environments that actually require interesting emergent behaviour!

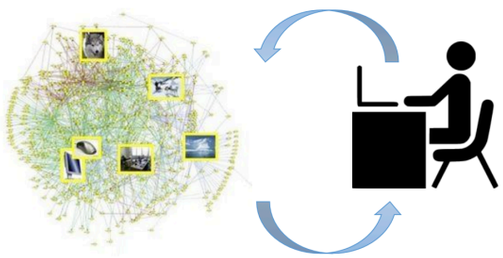
3. RL as ‚universal‘ learning

Large-scale machine learning

Why does deep learning work?



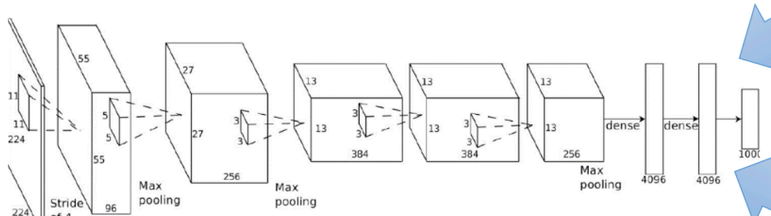
big models



big datasets

labeled

LLMs



this is what self-supervised learning is supposed to do

small labeled dataset

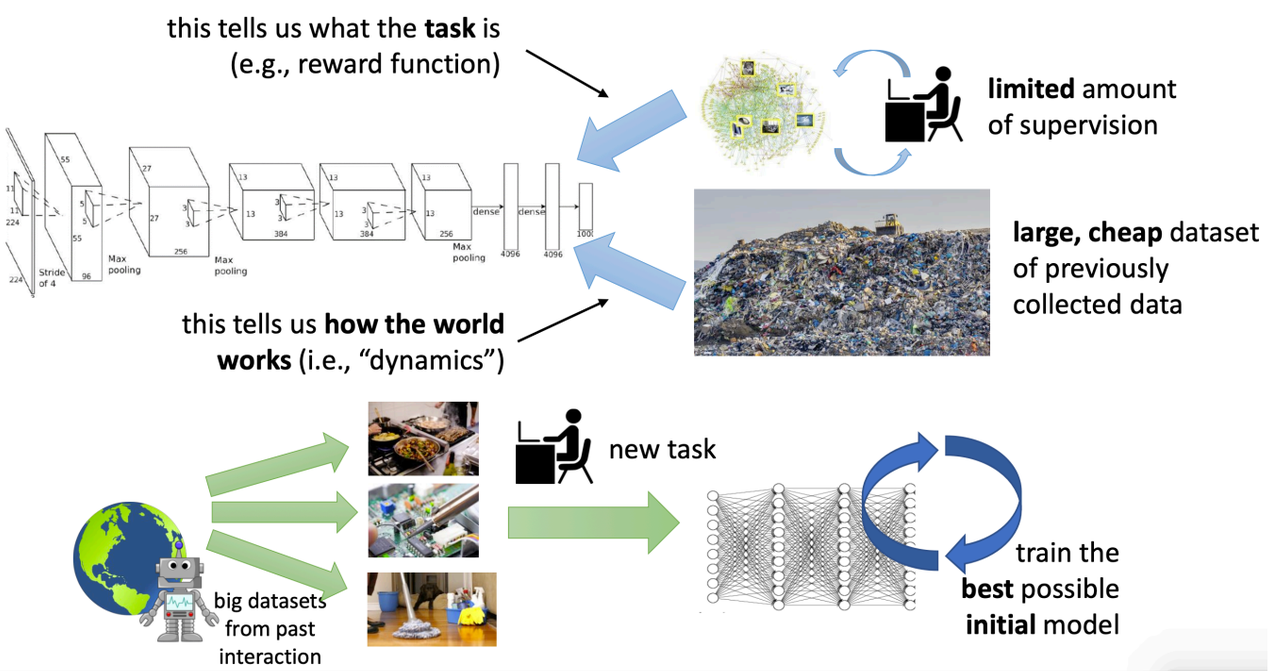
giant unlabeled garbage dataset (aka the Internet)

But then where does the knowledge come from?

“Classic” unsupervised learning: $p_\theta(\mathbf{x})$

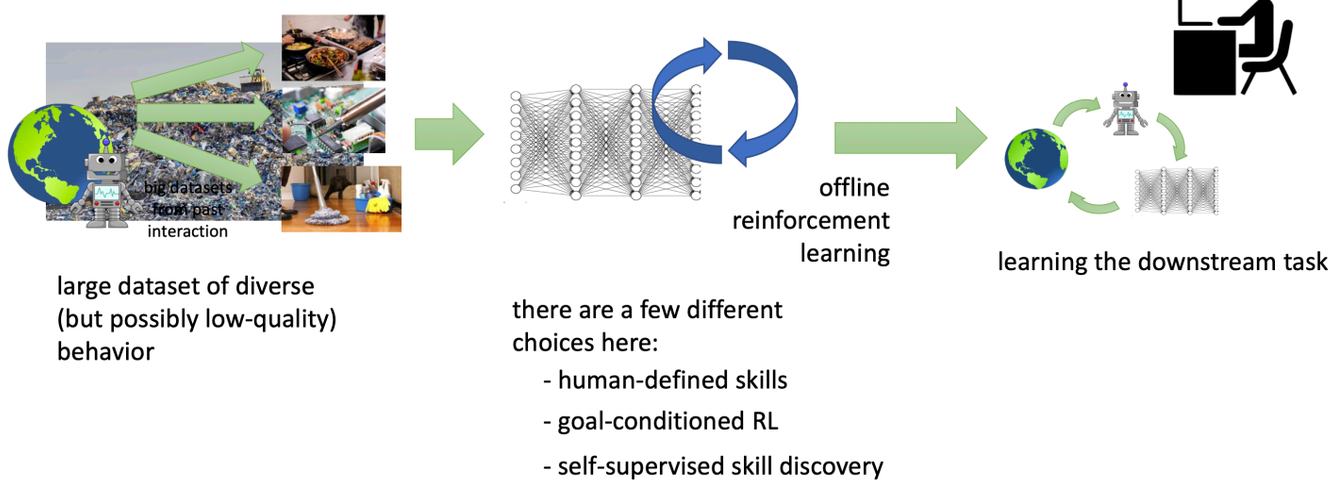
(this is what, for example, large language models do)

RL can do better



- ML is trying to produce adaptable complex decisions

The recipe

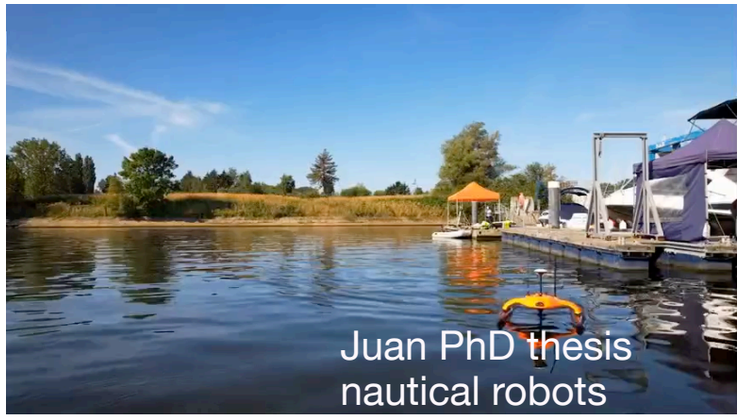


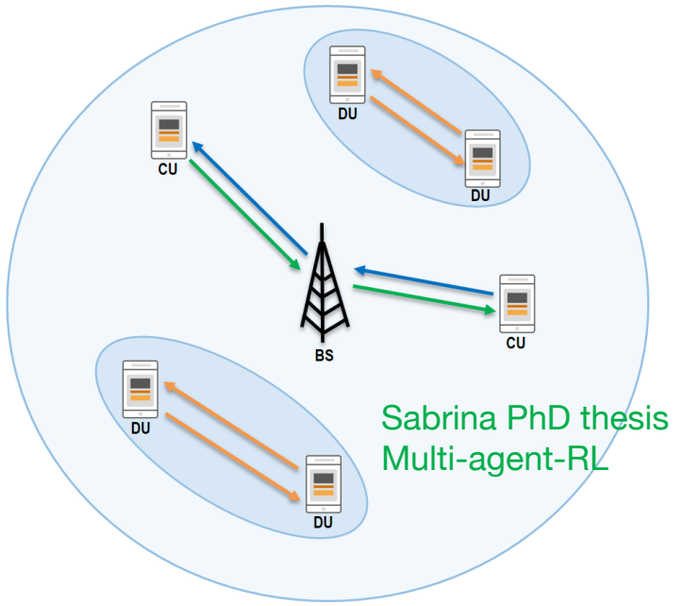
large dataset of diverse (but possibly low-quality) behavior

there are a few different choices here:

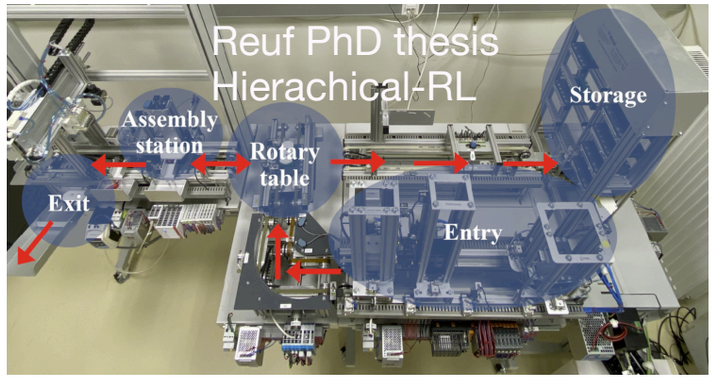
- human-defined skills
- goal-conditioned RL
- self-supervised skill discovery

RL here in Salzburg…



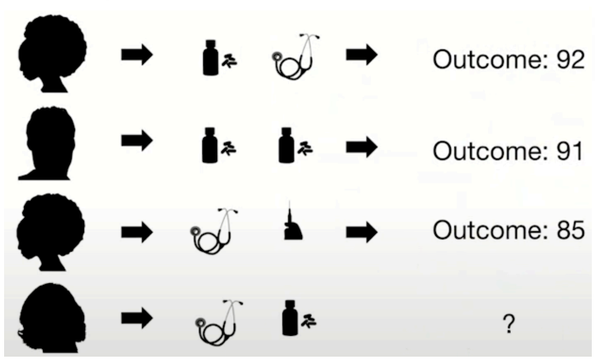


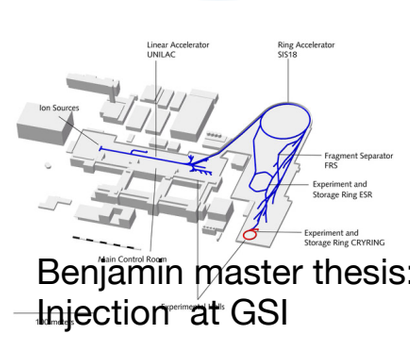
Sabrina PhD thesis
Multi-agent-RL



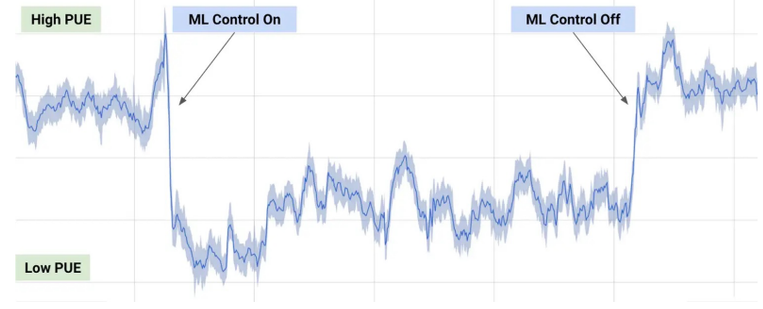
Sarah Reduce power consumption of companies and big plants

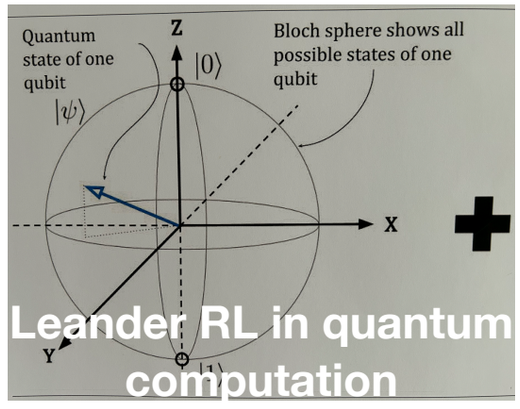
Sascha master thesis
Interpretable RL in medicine





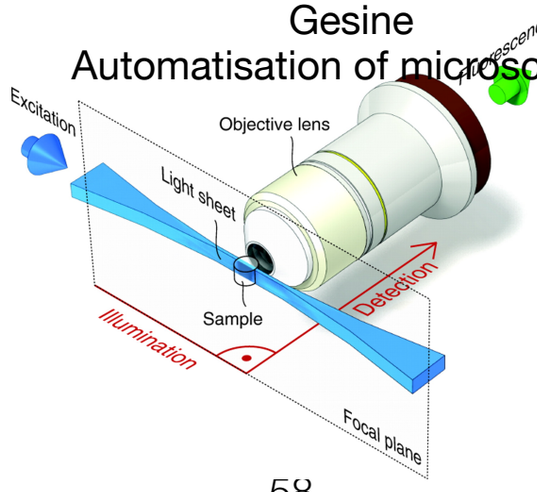
Benjamin master thesis:
Injection at GSI



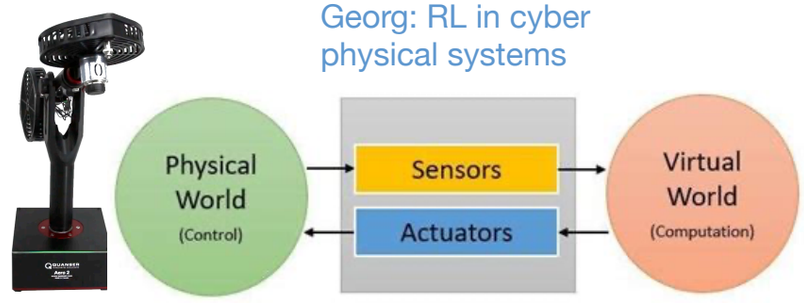


Leander RL in quantum computation

Gesine
Automatisation of microscopes



Georg: RL in cyber physical systems



Control theory and RL:
steer a non-linear system

Final remarks

- Think about your problems carefully (always)
- Decide how to formulate and solve them
- Choose the right algorithms - there is no solution fit it all in RL (ML)
- Think how to evaluate your progress
- Be brave: try to think outside the box if necessary
- Your are not alone! RL needs some experience, reach out to us- the community - if you need help

Thanks for your attention

References

- Courses online:
  - Chelsea Finn (Berkley): Deep Multi-Task and Meta Learning
  - Sergey Levine (Berkley): Deep Reinforcement Learning
  - Emma Brunskill (Stanford): Reinforcement Learning
- Papers:
  - Hirlaender, Simon / Pochaba, Sabrina / Lukas, Lamminger / Garcia, Andrea Santamaria / Xu, Chenran / Kaiser, Jan / Eichler, Annika / Kain, Verena „Deep Meta Reinforcement Learning for Rapid Adaptation In Linear Markov Decision Processes: Applications to CERN's AWAKE Project“, SMPS 2024
  - S. Hirlaender, L. Lamminger, G. Zevi Della Porta, and V. Kain, “Ultra fast reinforcement learning demonstrated at CERN AWAKE,” JACoW IPAC, vol. 2023, p. THPL038, 2023, doi: 10.18429/JACoW-IPAC2023-THPL038 https://cds.cern.ch/record/2886522/files/document.pdf
  - C. Finn, P. Abbeel, and S. Levine, “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.” 2017: https://arxiv.org/abs/1703.03400
  - S. Kamthe and M. Deisenroth, “Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control,” in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, A. Storkey and F. Perez-Cruz, Eds., in Proceedings of Machine Learning Research, vol. 84. PMLR, 2018, pp. 1701–1710. [Online]. Available: https://proceedings.mlr.press/v84/kamthe18a.html
  - A. Girard, C. E. Rasmussen, J. Quinonero- Candela, and R. Murray-Smith. Gaussian Process Priors with Uncertain Inputs-Application to Multiple-Step Ahead Time Series Forecasting. Ad- vances in Neural Information Processing Systems, 2003.
  - J. Beck et al., “A Survey of Meta-Reinforcement Learning.” 2023: https://arxiv.org/abs/2301.08028
  - J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, “Trust Region Policy Optimization.” 2017: https://arxiv.org/abs/1502.05477
  - J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, “High-Dimensional Continuous Control Using Generalized Advantage Estimation.” 2018: https://arxiv.org/abs/1506.02438
- Books:
  - R. S. Sutton, *Reinforcement learning*, Second edition. in Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, 2020 http://incompleteideas.net/book/the-book.html
  - A. Agarwal, N. Jiang, S. M. Kakade, W. Sun: *Reinforcement Learning: Theory and Algorithms, 2022* https://rltheorybook.github.io/
  - K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. https://probml.github.io/pml-book/book1.html
  - K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics.* MIT Press, 2023. http://probml.github.io/book2
  - D. Liberzon, *Calculus of variations and optimal control theory.* Princeton, NJ: Princeton University Pres, 2012, http://liberzon.csl.illinois.edu/teaching/cvoc/cvoc.html
  - F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge: Cambridge University Press, 2017
- Other resources:
  - S. Hirlaender, Advanced concepts in RL, 2023 (RL4AA23 Lecture) https://github.com/RL4AA/RL4AA23/blob/main/slides/Hirlaender_advanced_concepts.pdf