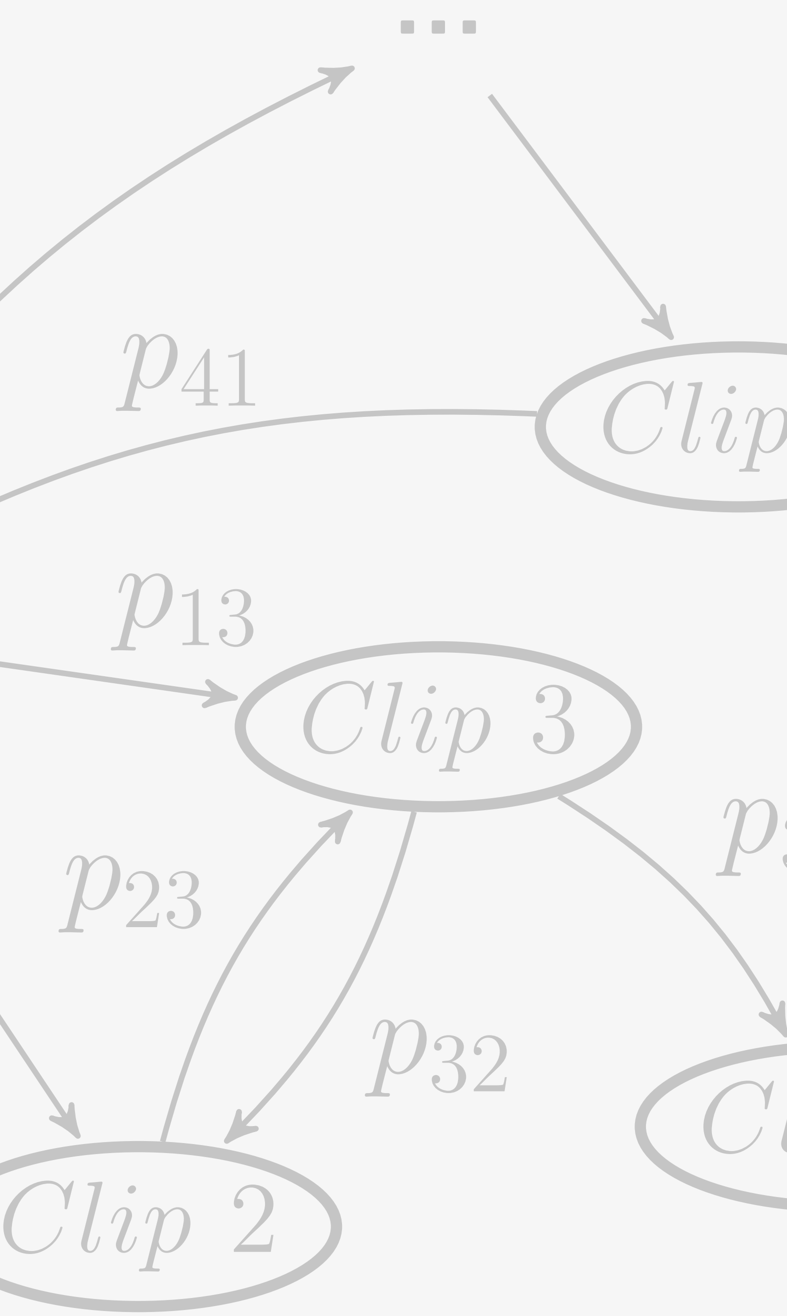
universität innsbruck
Institute for Theoretical Physics

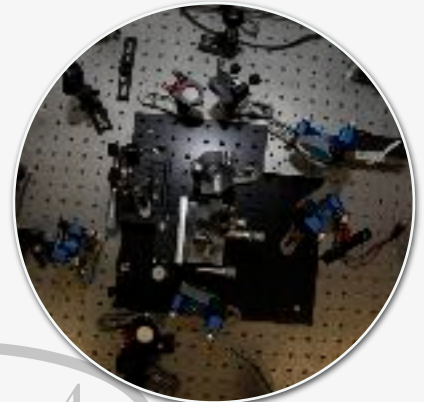
Machine learning for designing new quantum experiments

Alexey Melnikov

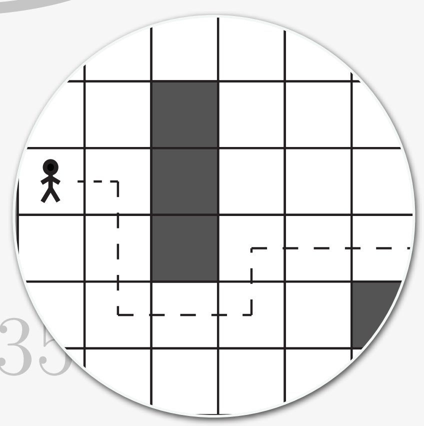
Quantum Machine Learning Plus
Innsbruck, September 18, 2018

FWF Der Wissenschaftsfonds.
TEMPLETON World Charity Foundation, Inc.





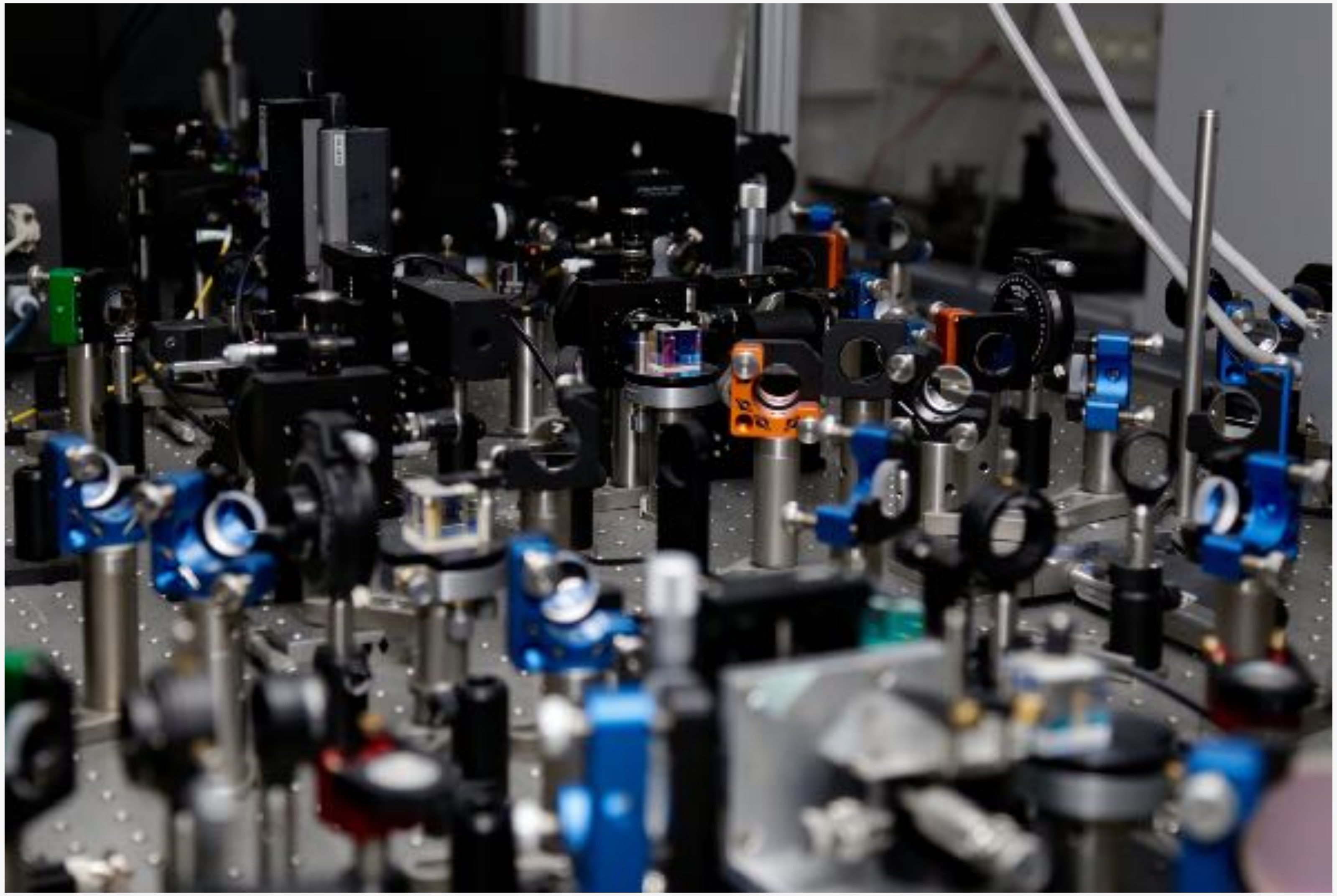
Creating quantum experiments



Reinforcement learning and
the projective simulation model

Learning to design quantum experiments
with projective simulation

Creating novel quantum experiments



Courtesy of Manuel Erhard (University of Vienna)

A typical example of complex arrangement of elements on the optical table.

Creating novel experiments:

- Defining a research goal, given existing tools
- Finding a solution that achieves this goal
- Executing the found solution in the lab:
  - understanding what is needed
  - ordering things
  - writing programs for talking to different devices
  - dealing with problems

Goal: multiphoton entanglement

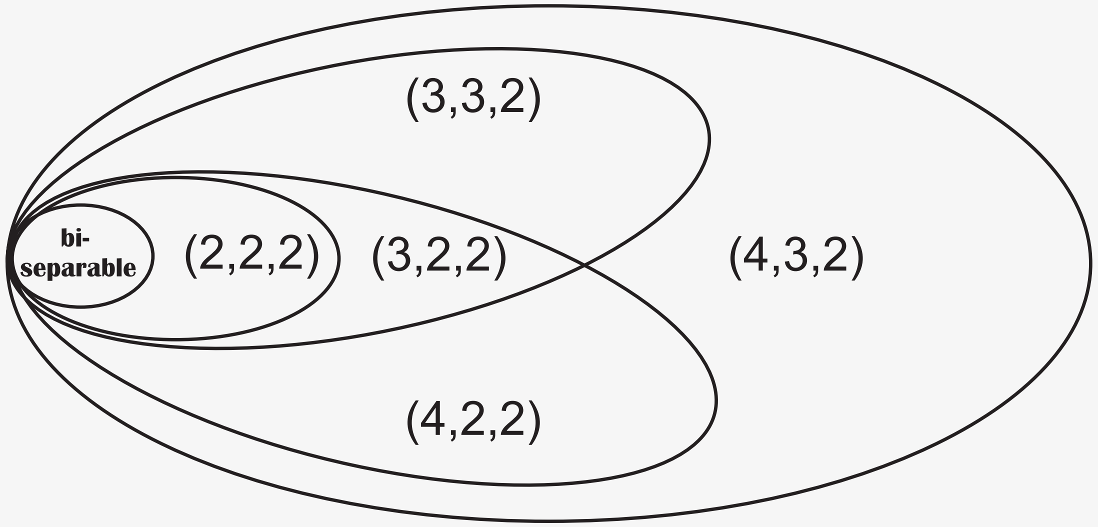
We are interested in new implementations for the *creation and manipulation of complex quantum states*.

For example, it is known how to construct a $|000\rangle + |111\rangle$ state, but how to construct a $|000\rangle + |111\rangle + |222\rangle$ state?

Or other *interesting* high-dimensional tripartite states.

Tripartite states are characterized by 3 numbers forming a Schmidt rank vector (SRV):

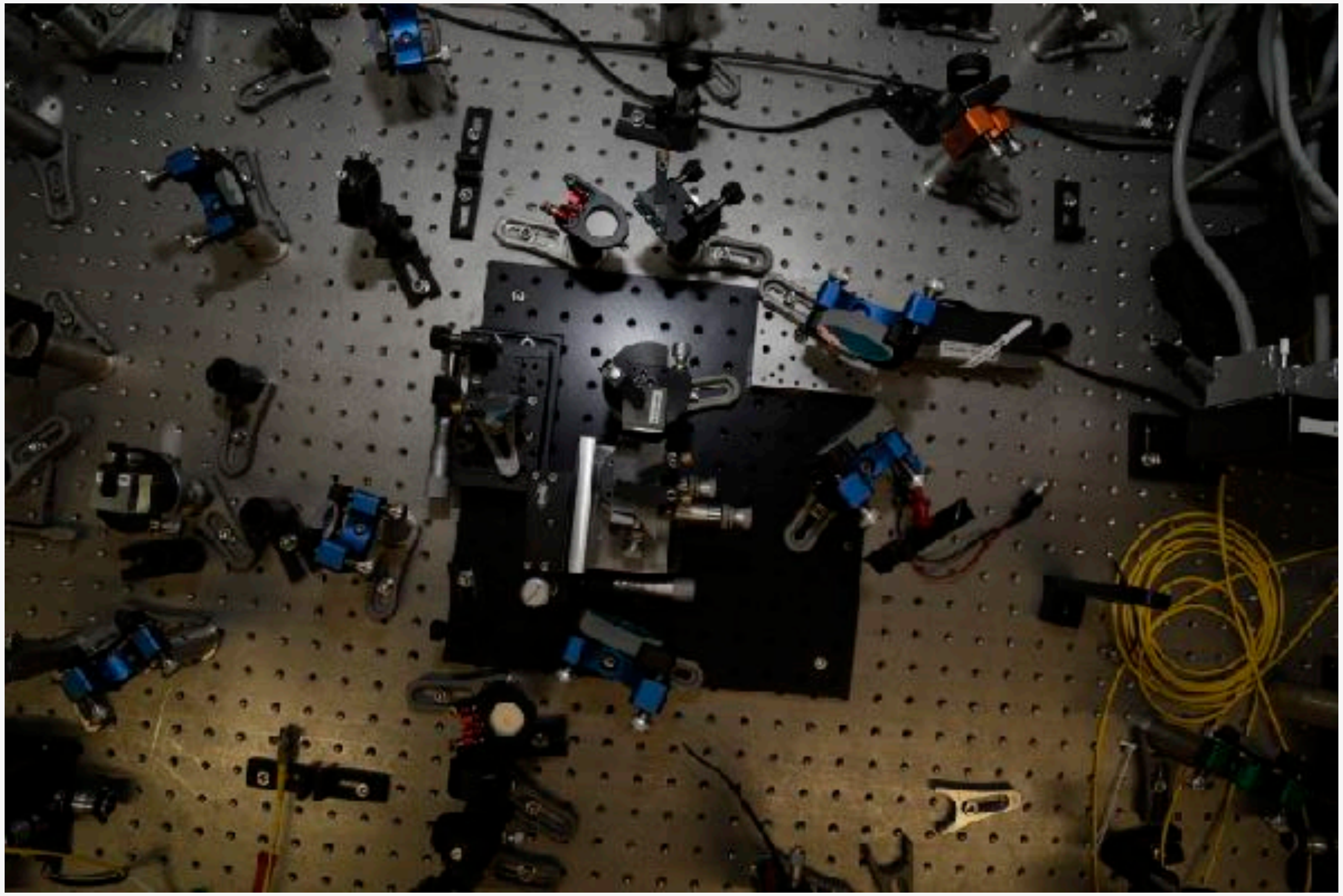
(r_A, r_B, r_C), where $r_n = \operatorname{rank}\left(\operatorname{Tr}_{\bar{n}} |\psi\rangle\langle\psi|\right)$



Structure of multidimensional entanglement in multipartite systems

M. Huber and J. I. de Vicente, Phys. Rev. Lett. 110, 030501 (2013)

Existing tools



Courtesy of Manuel Erhard (University of Vienna)

These optical elements, if arranged in a proper way, can create multiphoton entangled states.

Toolbox of optical elements:

- Photon source
- Nonlinear crystal
- OAM-Hologram
- Dove prism
- Mirror
- Beam splitter
- Detector

Existing tools

- Nonlinear crystal — entangled state initialization

$$|\psi_0\rangle = \frac{1}{\sqrt{3}}\left(|-1_a\rangle|1_b\rangle + |0_a\rangle|0_b\rangle + |1_a\rangle|-1_b\rangle\right)$$

- OAM-Hologram — shifts OAM

$$|l_a\rangle \rightarrow |l_a + n_A\rangle$$

- Mirror

$$|l_a\rangle \rightarrow |-l_a\rangle$$

- Dove prisms

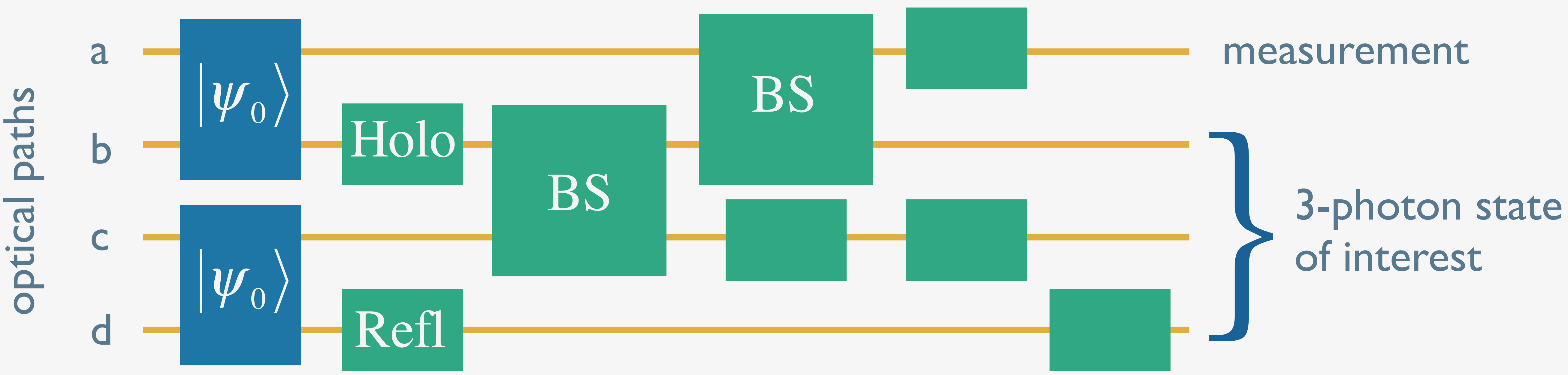
$$|l_a\rangle \rightarrow ie^{i\pi l/n}|-l_a\rangle$$

- Non-polarizing symmetric 50/50 beam splitter — creating superpositions

$$|l_a\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|l_b\rangle + i|-l_a\rangle\right) \qquad |l_b\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|l_a\rangle + i|-l_b\rangle\right)$$

- Detector — triggering a final state

Finding ways to achieve the goal



There is a similarity to:

- Circuit design (e.g., a generative approach by Alejandro Perdomo-Ortiz, a poster by In-Chan Choi)
- Hamiltonian design (e.g., a poster by Luca Innocenti)
- Quantum control

Finding ways to achieve the goal

How difficult is it to find a sequence of optical elements that has e.g $|000\rangle + |111\rangle + |222\rangle$ state as an output?

Turned out to be not that easy

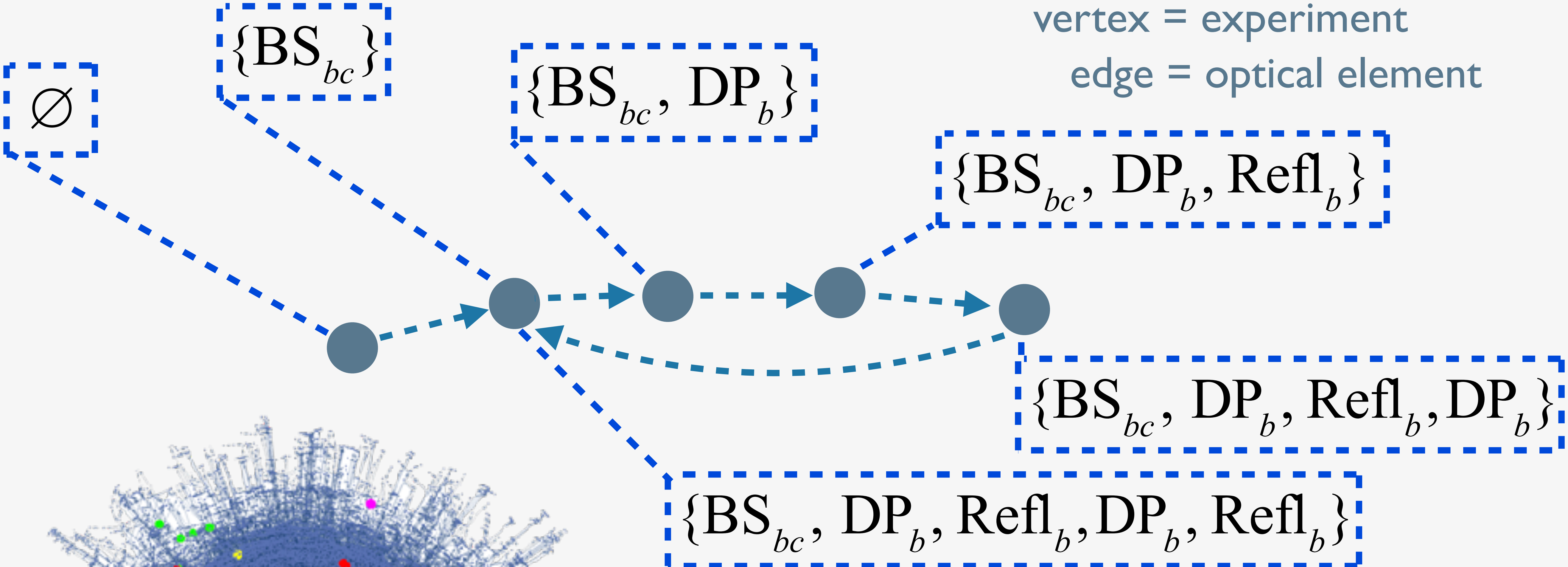
Number of possible configurations grows exponentially with the size of experiment.

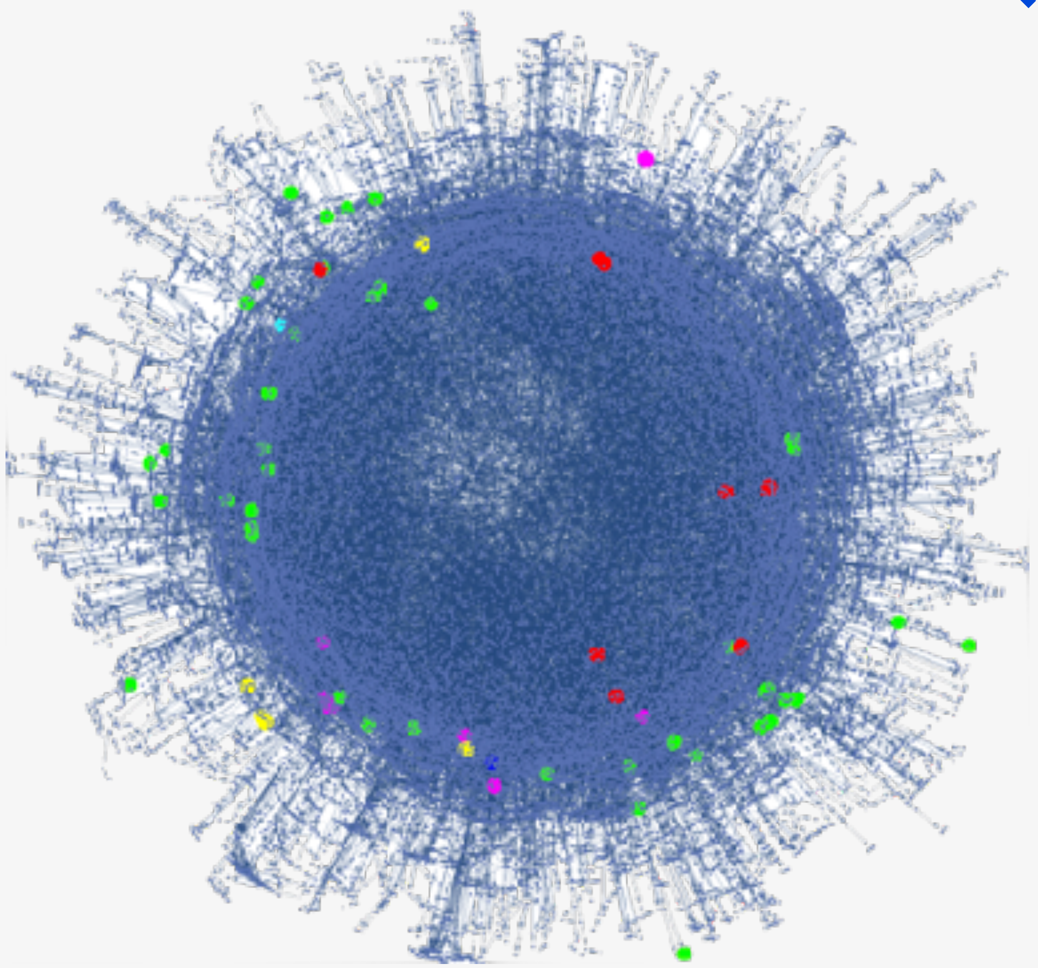
With a reasonable limitation of 12 basic elements on the table, and having just 4 optical paths there are

More than 10^{17} optical setups

The space of possibilities is complex

vertex = experiment
edge = optical element



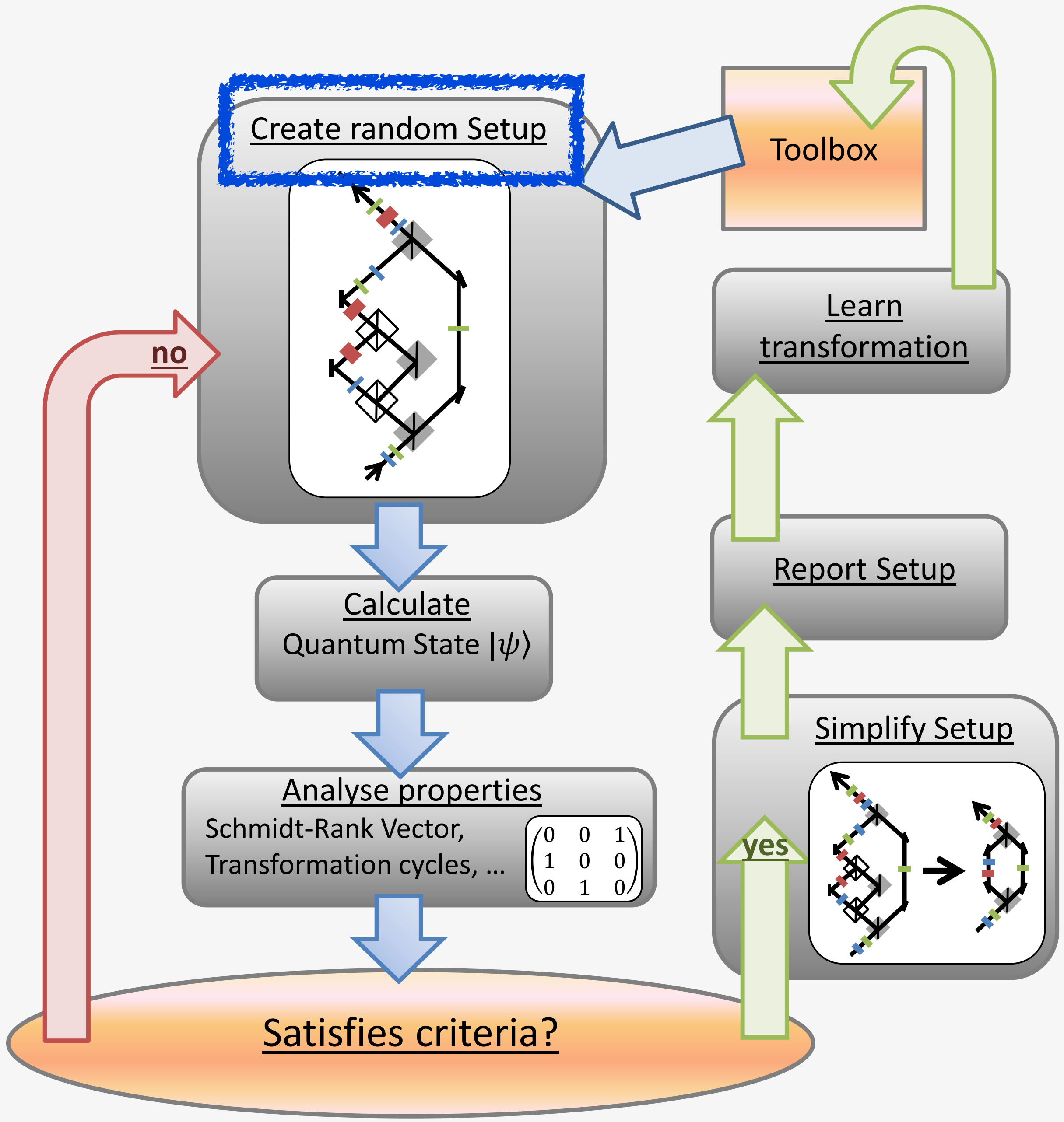


around 45000 randomly generated experiments are shown

only 67 potentially interesting experiments create high-dimensional multiphoton entanglement

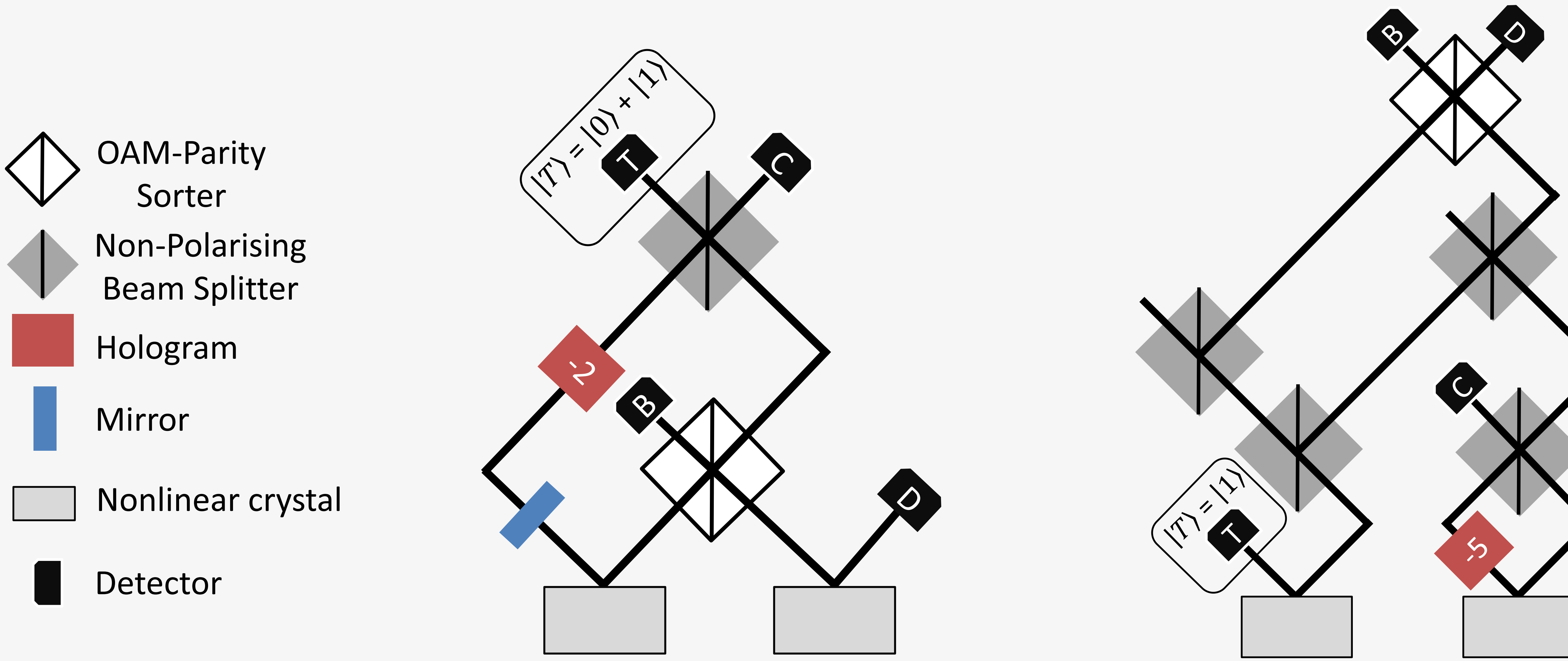
Automating the design of experiments

Working principle of automated search for quantum experiments (MELVIN)



M. Krenn et. al., Phys. Rev. Lett. 116, 090405 (2016)

Automating the design of experiments



3-dimensional GHZ state,
or (3,3,3) state

(10, 6, 5) state

M. Krenn et. al., Phys. Rev. Lett. 116, 090405 (2016)
(3,3,3) state was implemented in arxiv:1708.03881

Is automated random search good enough?

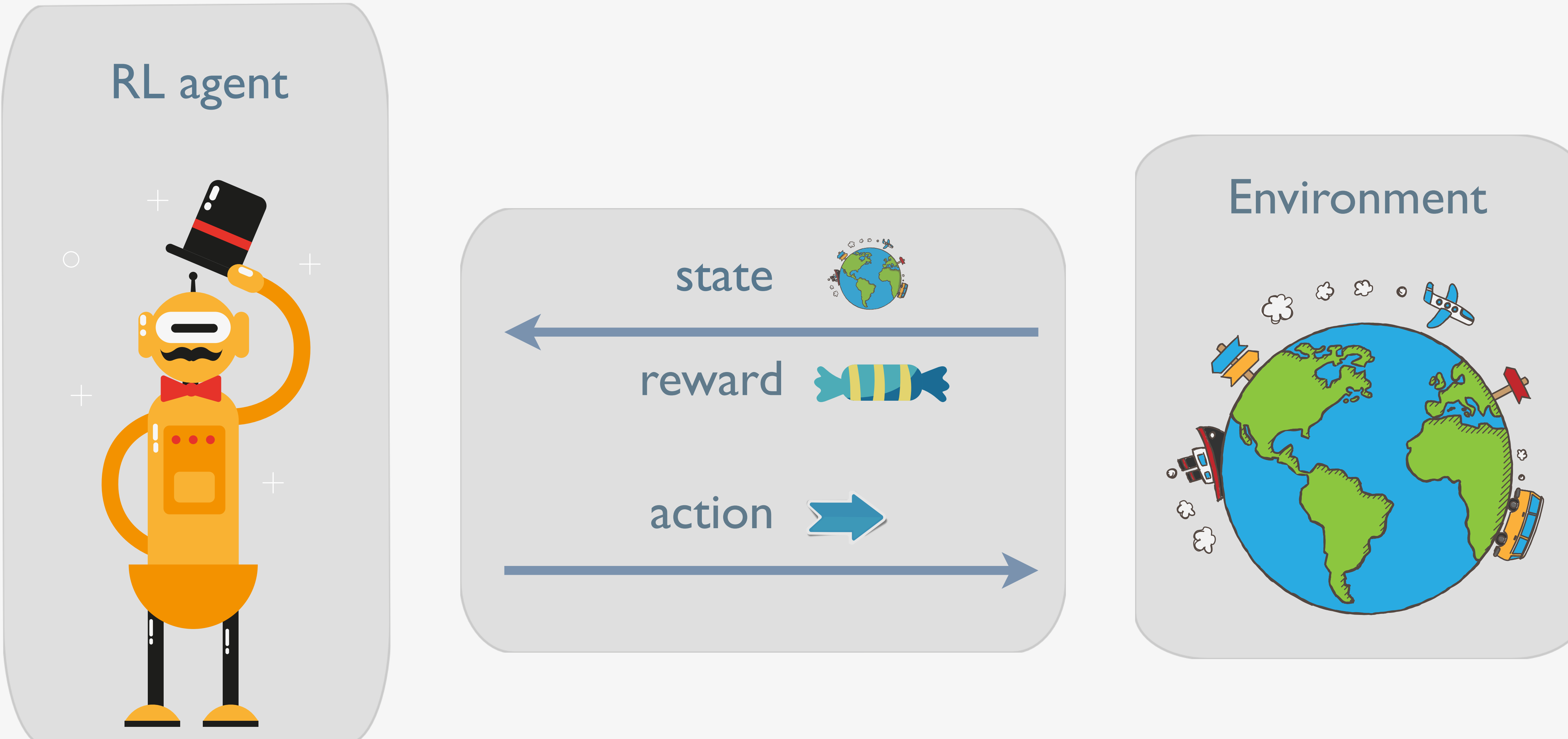
No, we should try to

- find simpler experiments (simplified automatically, not by hand)
- find more interesting experiments
- try to learn from the space of possibilities

How to do these improvements?

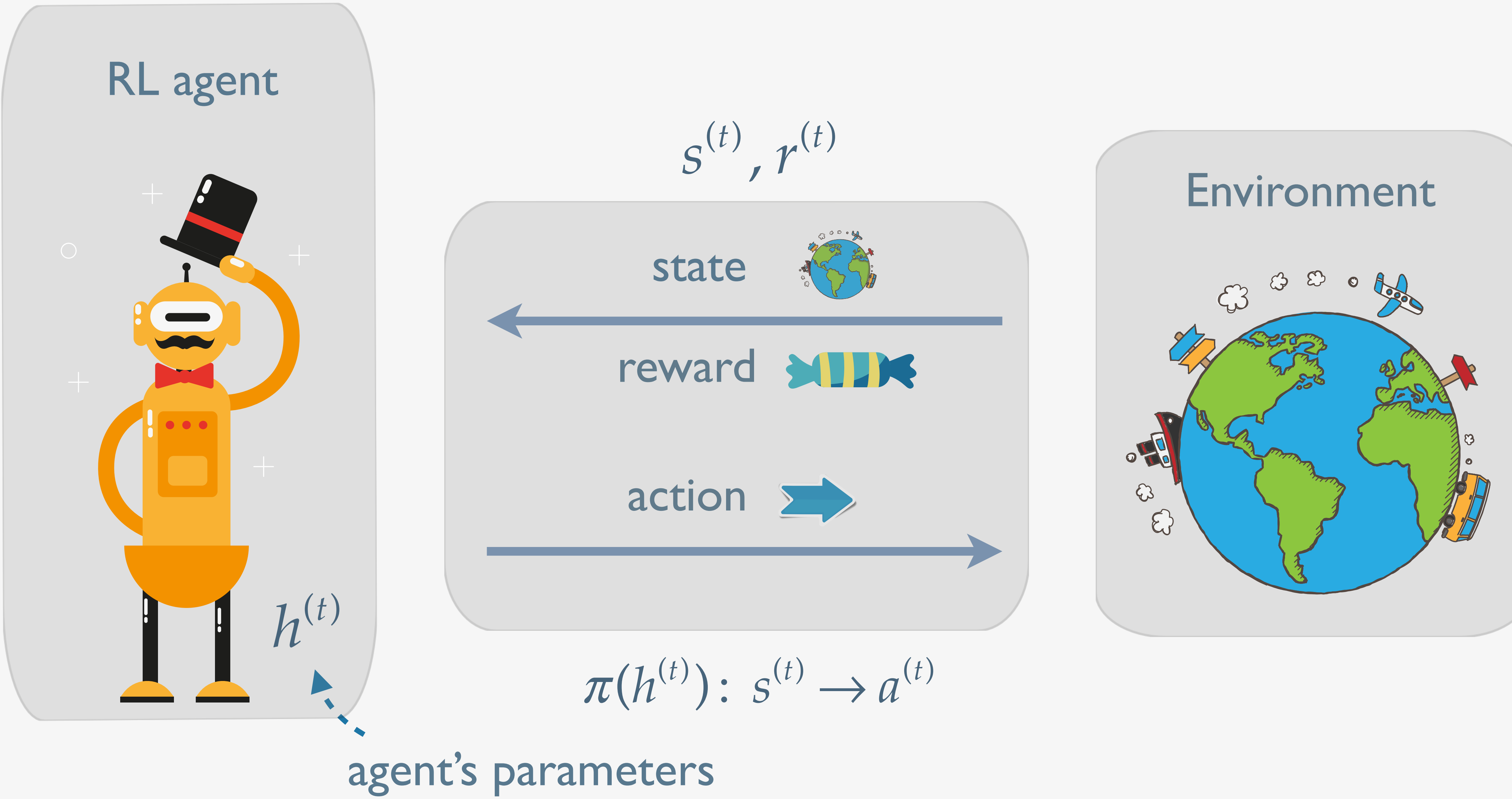
We do it by formulating these problems as *reinforcement learning* problems and solving them with *projective simulation*

Reinforcement learning (RL) framework



Reinforcement learning (RL) framework

The agent acts according to a policy π, which maps input states to actions.

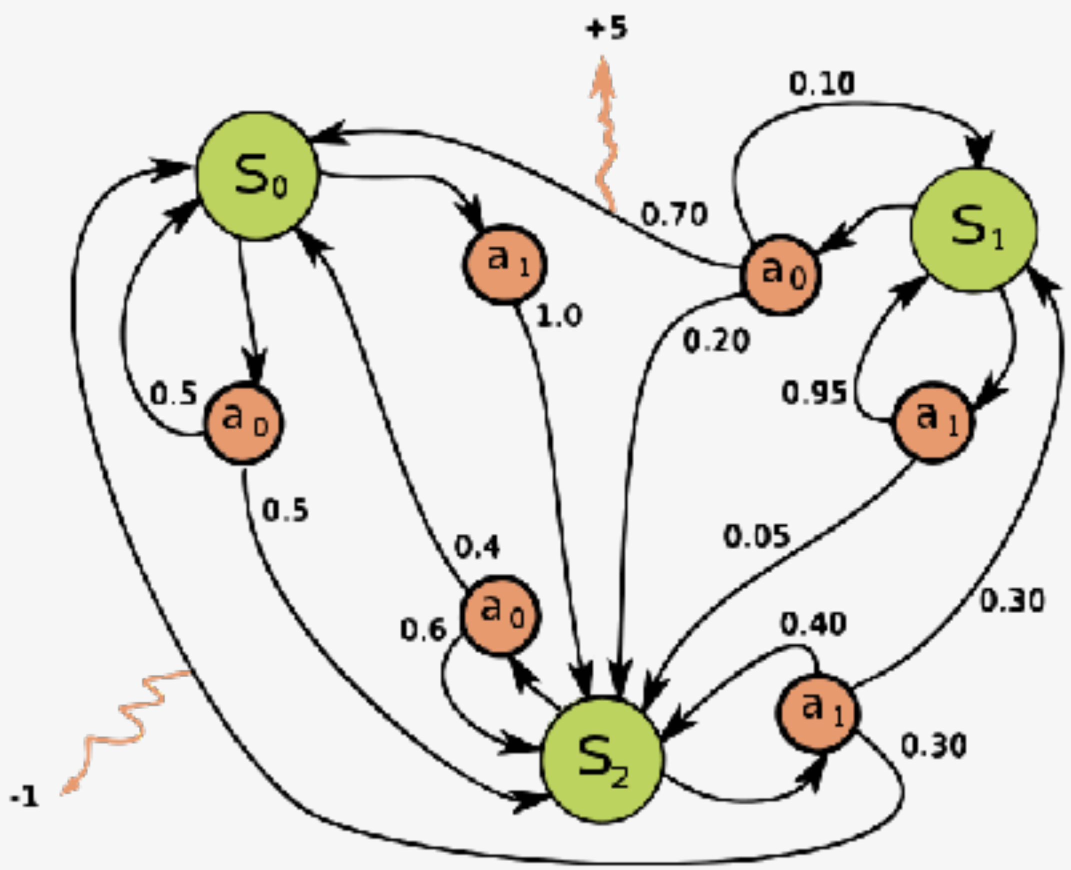


The goal of learning is to modify the agent's parameters, such that the agent produces desired outputs.
Learning algorithm updates these parameters $h^{(t+1)} = L(h^{(t)})$.

Desired outputs

Desired outputs are such that they maximize some function of cumulative reward.

Example:
Markov Decision Processes (MDP), a class of RL environments



commons:Waldoalvarez

One usually maximizes an expected future return

$$R(T) = \sum_{t=T}^{\infty} \gamma^t r^{(t)}$$

It is possible to converge to an optimal policy in the limit

RL is challenging

RL is a *very general* machine learning framework

and *a very challenging*

from an agent's point of view:

- there is no teacher, and no training examples are given, only (often ambiguous) reward signals
- rewards are usually delayed, there are temporal correlations in data
- agent's actions usually affect the environment, hence changing the subsequent data
- number of interactions is limited, which leads to the exploration-exploitation dilemma

RL is challenging

RL is a *very general* machine learning framework

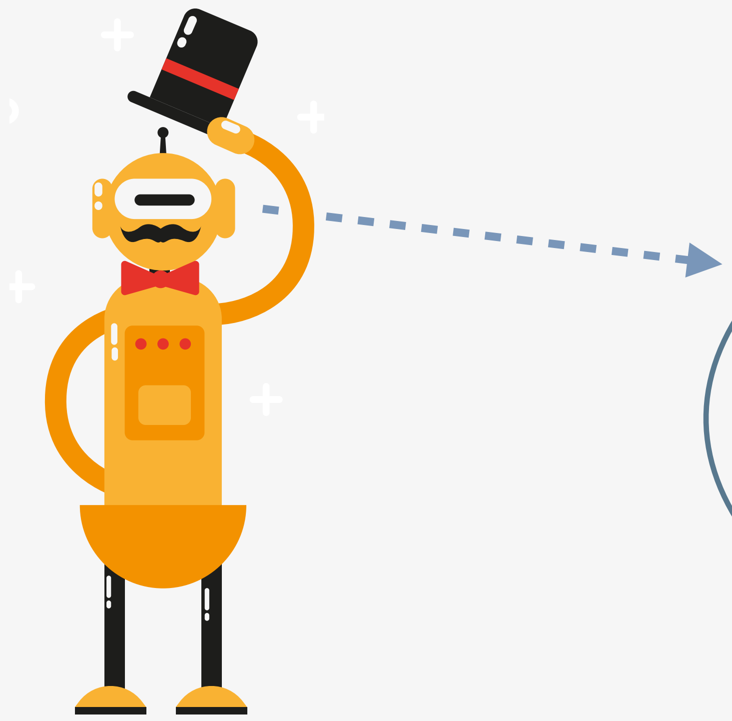
and *a very challenging*

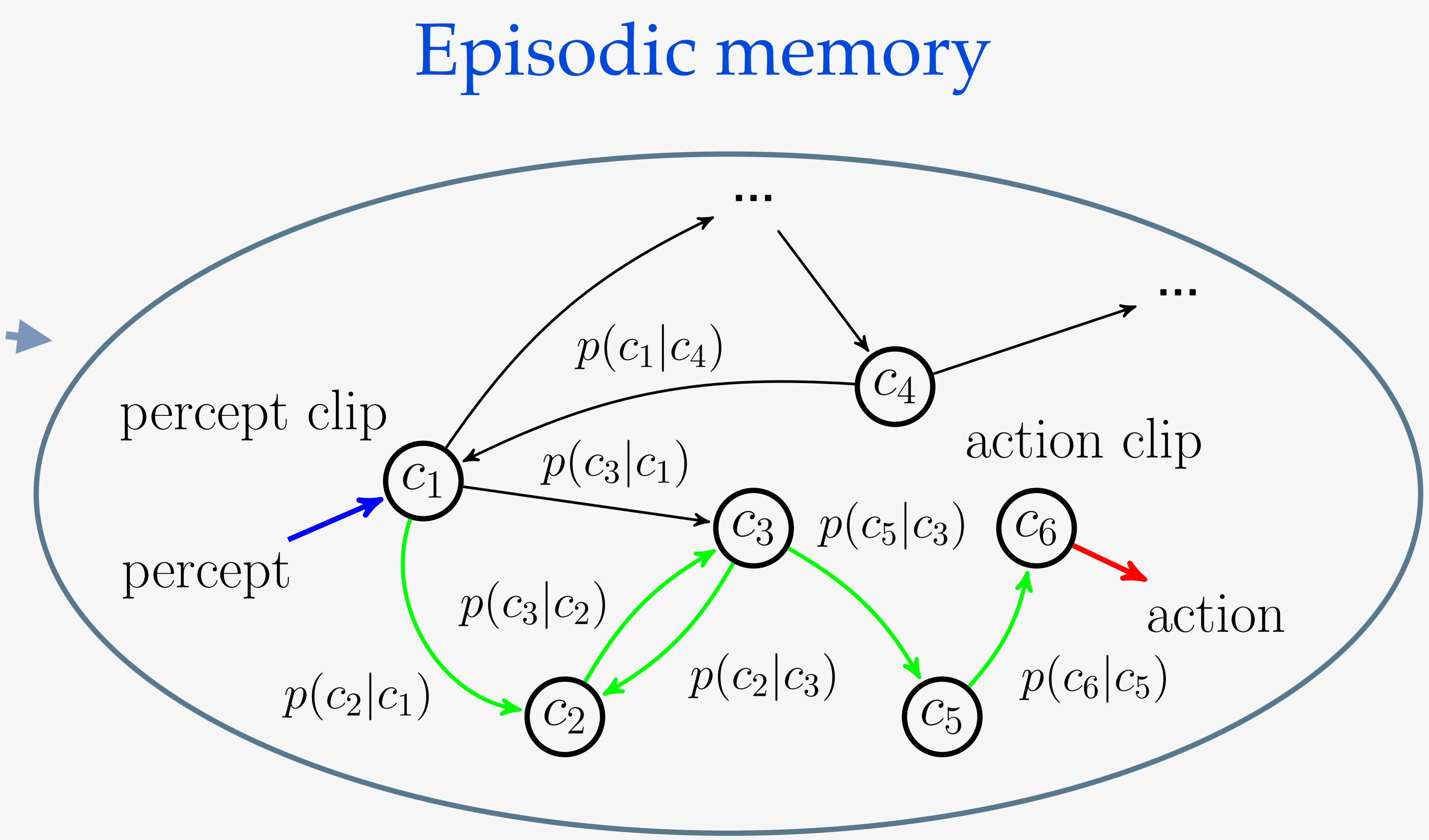
from a developer's point of view:

- agent's performance is usually very sensitive to the choice of meta-parameters
- reward function specification is hard;
  agent usually "hacks" your reward function
- agent's performance is influenced by it's individual interaction history;
  there is usually a fraction of "unlucky" agents;
  you never know if performance is bad, or you are just unlucky

Projective simulation (PS)

The PS model is a physical approach to agency. The PS agent processes information stochastically in a directed, weighted network of clips, where each clip represents a remembered percept, action, or sequences thereof.





Once a percept is observed, the network is activated, invoking a random walk between the clips, until an action clip is hit and couples out as a real action of the agent.

H. J. Briegel and G. De las Cuevas, Sci. Rep. 2, 400 (2012)

Some features of PS

low computational complexity

decision-making process is fast, we can do many trials

low model complexity

easy to choose parameters

interpretability

we can analyze the agent's policy by observing graph properties

conceptually attractive

decision-making on graphs, flexibility of graphs

the framework is general

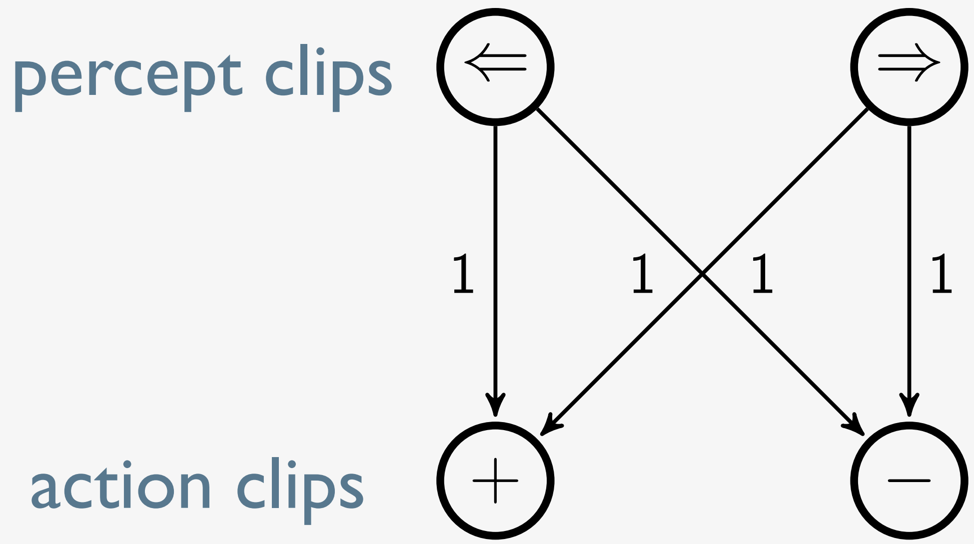
Python code
ProjectiveSimulation.org

clear route to quantization

quantum walks on graphs

Learning in the PS model

Learning is realized by enhancing connections between relevant clips



PS network initially

All edges in the network have initially the same strength $h = 1$.

Strength determine the hopping probabilities between the clips

$$p_{ij}^{(t)} = \frac{h_{ij}^{(t)}}{\sum_k h_{ik}^{(t)}},$$

and so the initial behaviour is random.

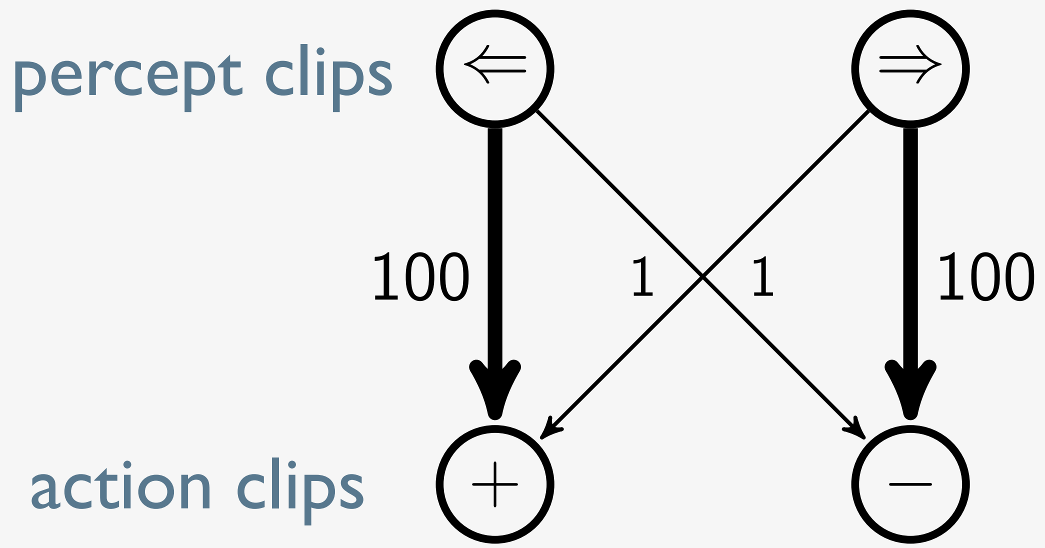
Reward from the environment increases the probability to do the same action.

Learning algorithm

$$h_{ij}^{(t+1)} = h_{ij}^{(t)} + r^{(t)}$$

Learning in the PS model

Learning is realized by enhancing connections between relevant clips



PS network after learning

All edges in the network have initially the same strength $h = 1$.

Strength determine the hopping probabilities between the clips

$$p_{ij}^{(t)} = \frac{h_{ij}^{(t)}}{\sum_k h_{ik}^{(t)}},$$

and so the initial behaviour is random.

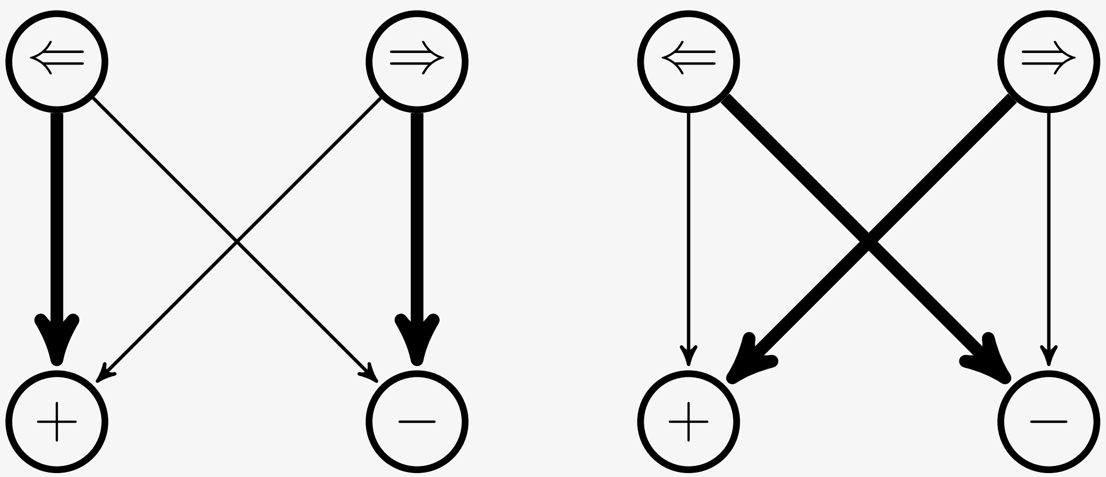
Reward from the environment increases the probability to do the same action.

Learning algorithm

$$h_{ij}^{(t+1)} = h_{ij}^{(t)} + r^{(t)}$$

Changing environments

Environment may change

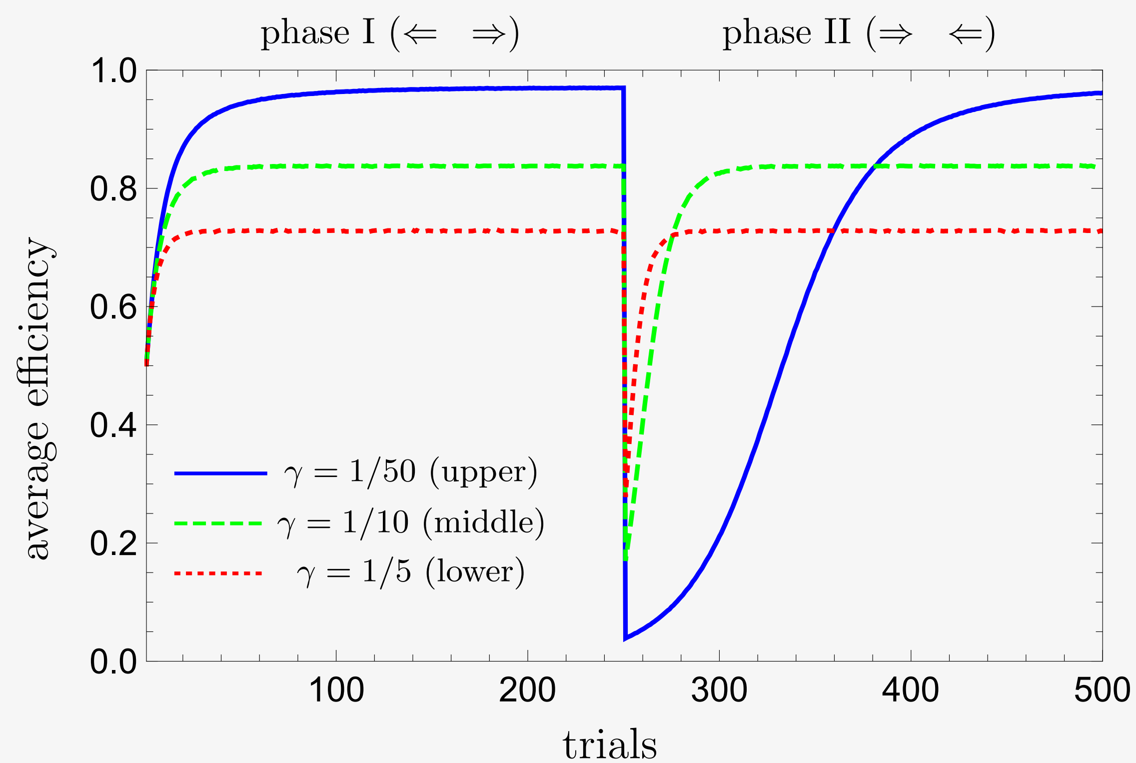


Phase I

Phase II

Learning algorithm

$$h_{ij}^{(t+1)} = h_{ij}^{(t)} - \gamma(h_{ij}^{(t)} - 1) + r^{(t)}$$



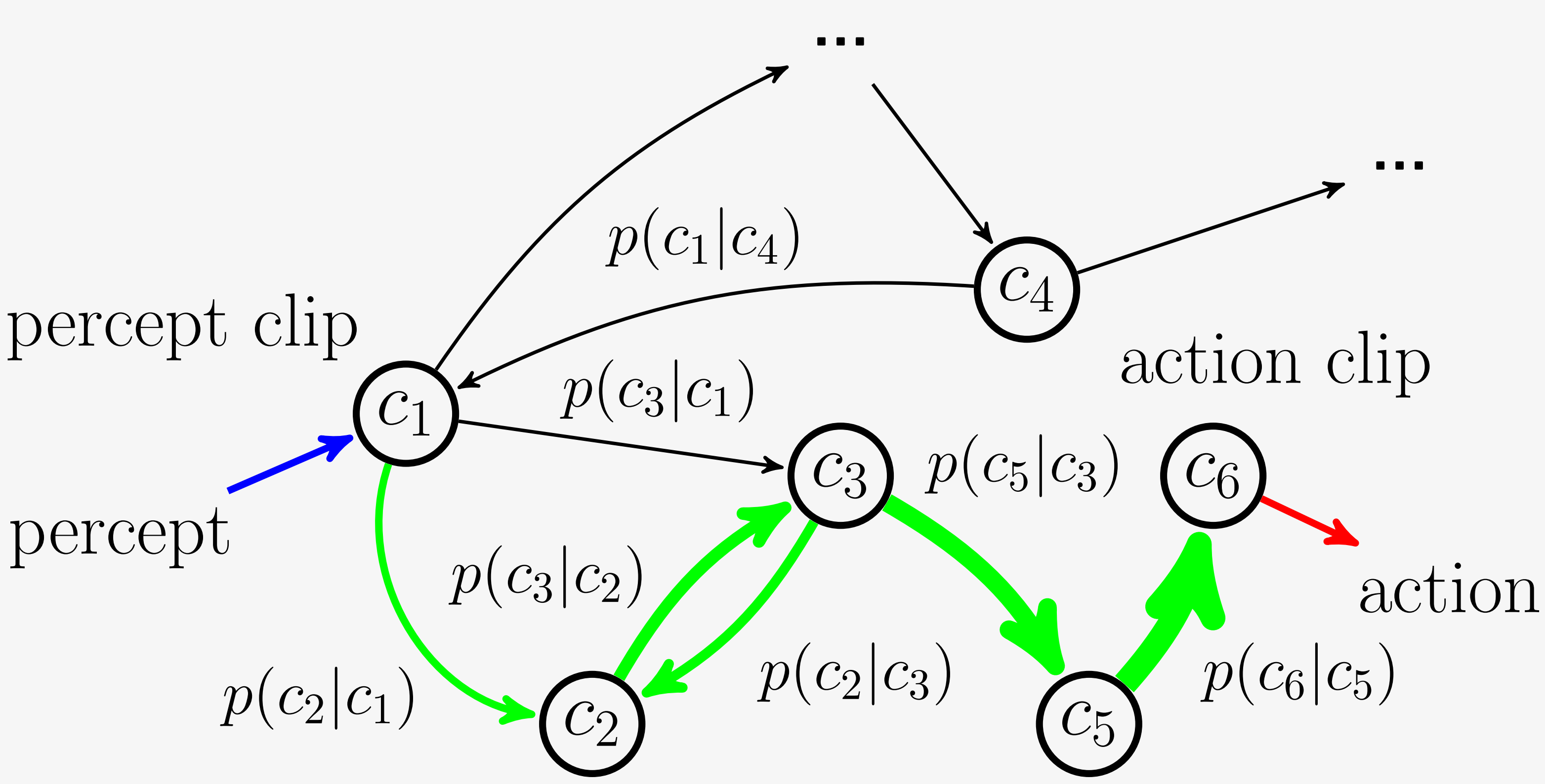
Relearning is much faster, but the success rate is lower

Tradeoff between flexibility and maximum achievable success rate

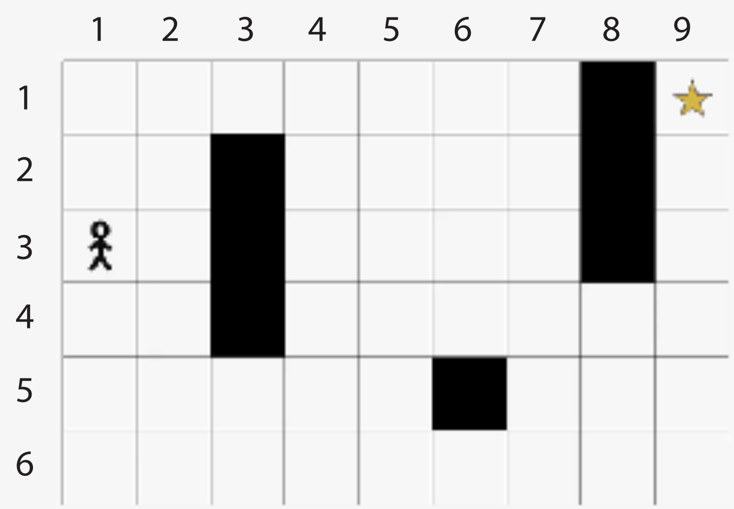
Glow mechanism

Learning algorithm

$$h^{(t+1)} = h^{(t)} - \gamma\left(h^{(t)} - 1\right) + g^{(t+1)}\lambda^{(t)}, \quad g^{(t+1)}(c_i, c_j) = \begin{cases} 1, \text{ if } (c_i, c_j) \text{ was traversed} \\ \left(1 - \eta\right) g^{(t)}(c_i, c_j), \text{ otherwise} \end{cases}$$



RL: navigation problems



Grid world problem

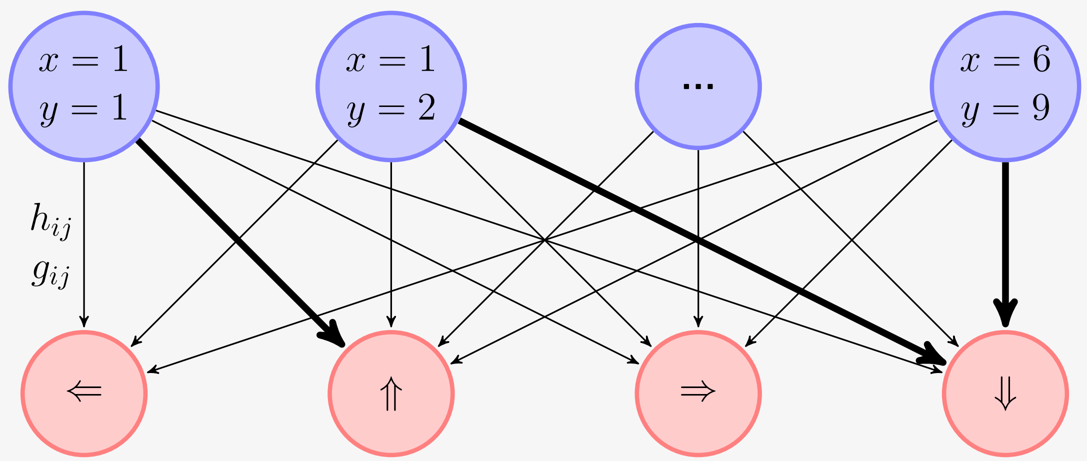
Percept: room coordinates (x, y)

Actions: left, right, up and down

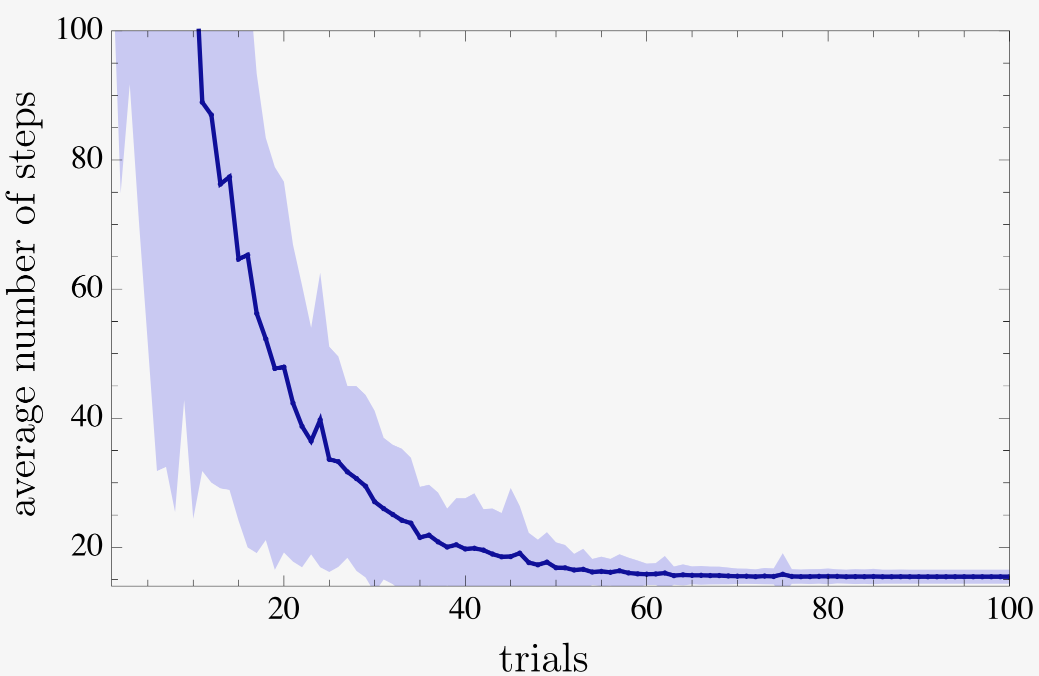
Reward: +1 for reaching the (1, 9) room

Reward awaits at least 14 decisions away from the start

The task is to find the shortest path



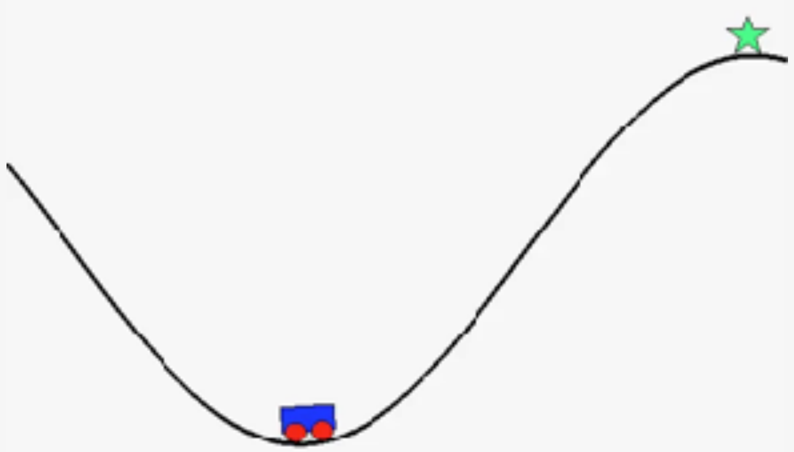
PS network:
directed complete bipartite weighted graph



Learning curve

AAM, A. Makmal, and H. J. Briegel, arXiv:1804.08607, accepted in IEEE Access

RL: navigation problems



Mountain car problem

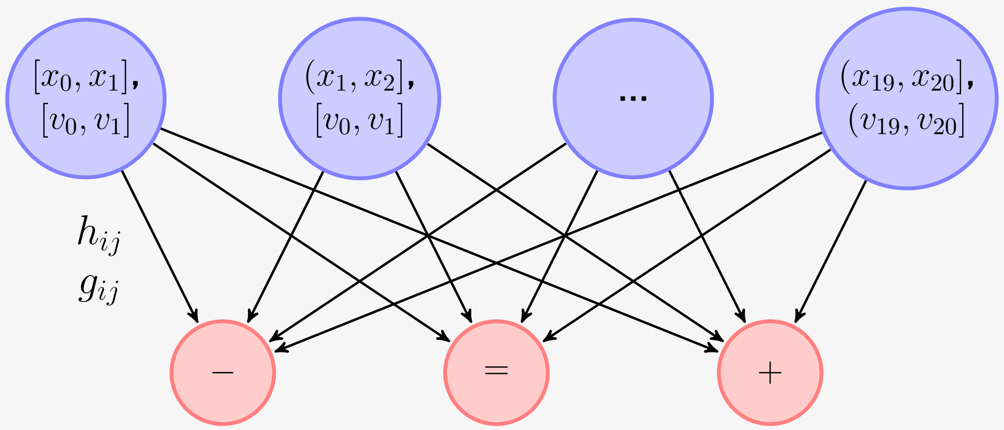
Percept: position and velocity (x, v)

Actions: accelerate to the left, right, or no acceleration

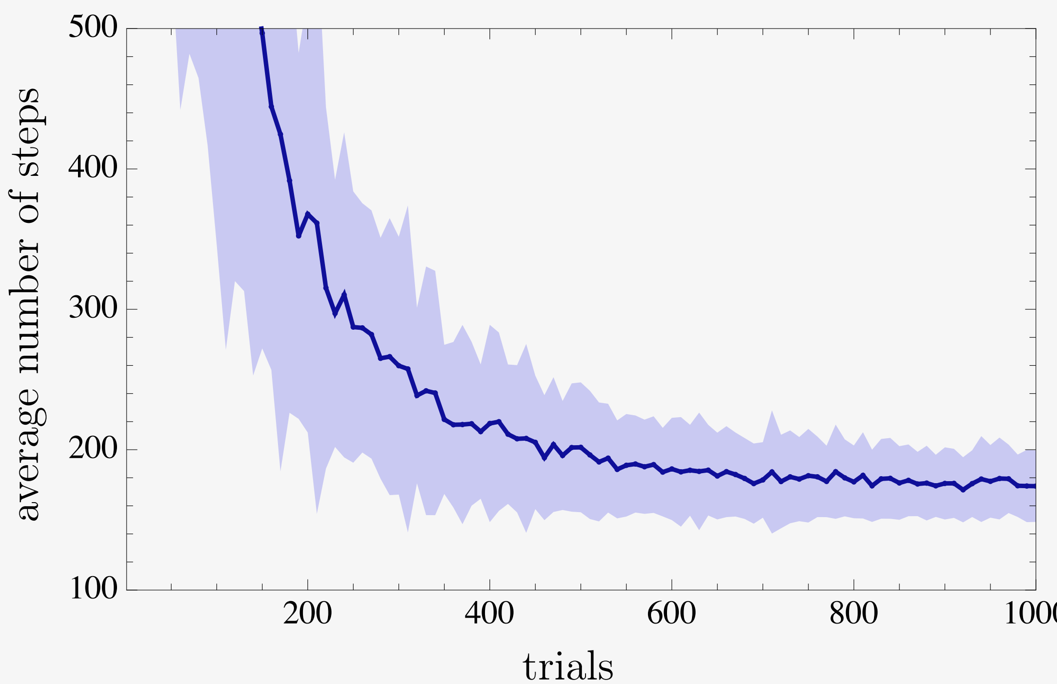
Reward: +1 for reaching the right mountain top

Reward awaits at least 86 decisions away from the start

The task is to find the shortest path



PS network:
directed complete bipartite weighted graph

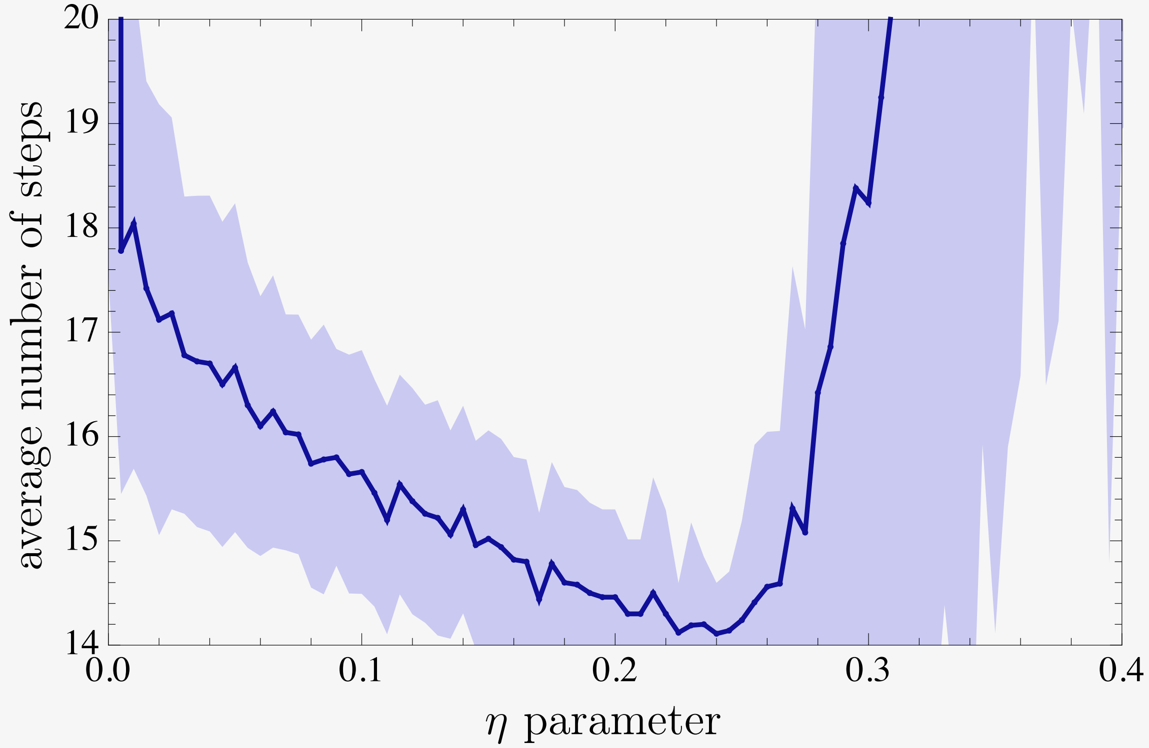


Learning curve

AAM, A. Makmal, and H. J. Briegel, arXiv:1804.08607, accepted in IEEE Access

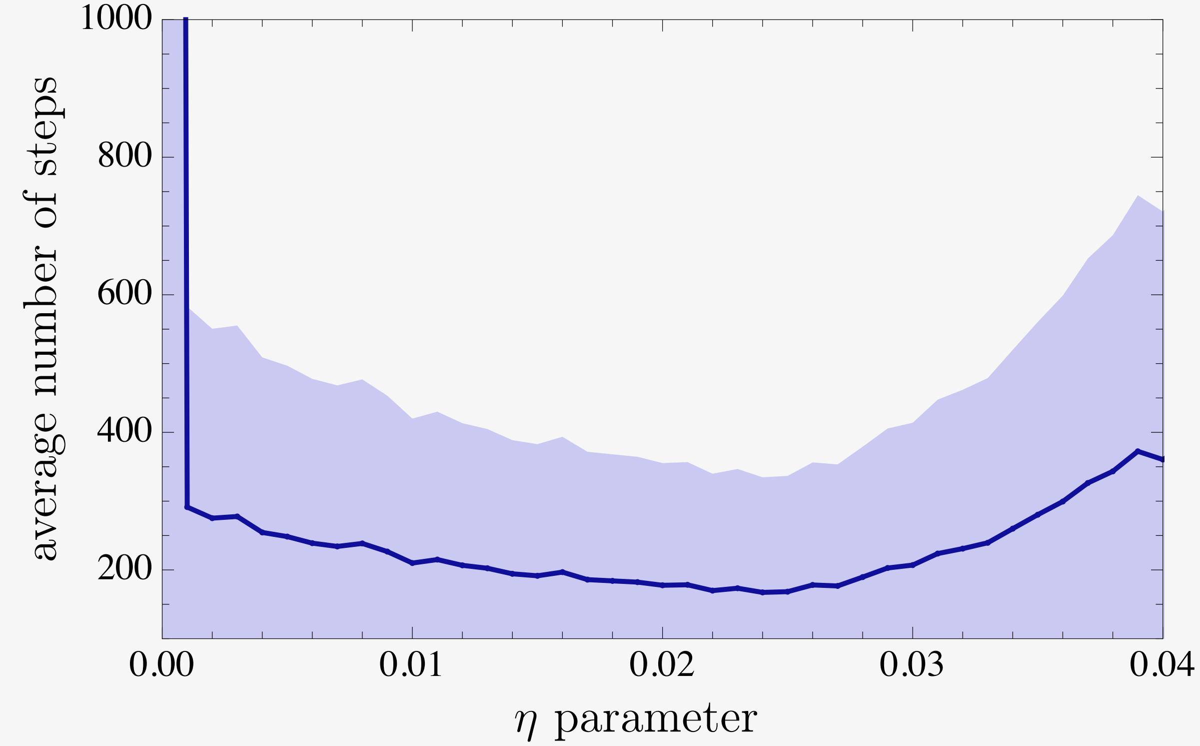
Choice of model parameters

In case of low-dimensional Markov decision processes the choice of parameters is straightforward



Grid world problem

2 hours per agent to check the full parameter range

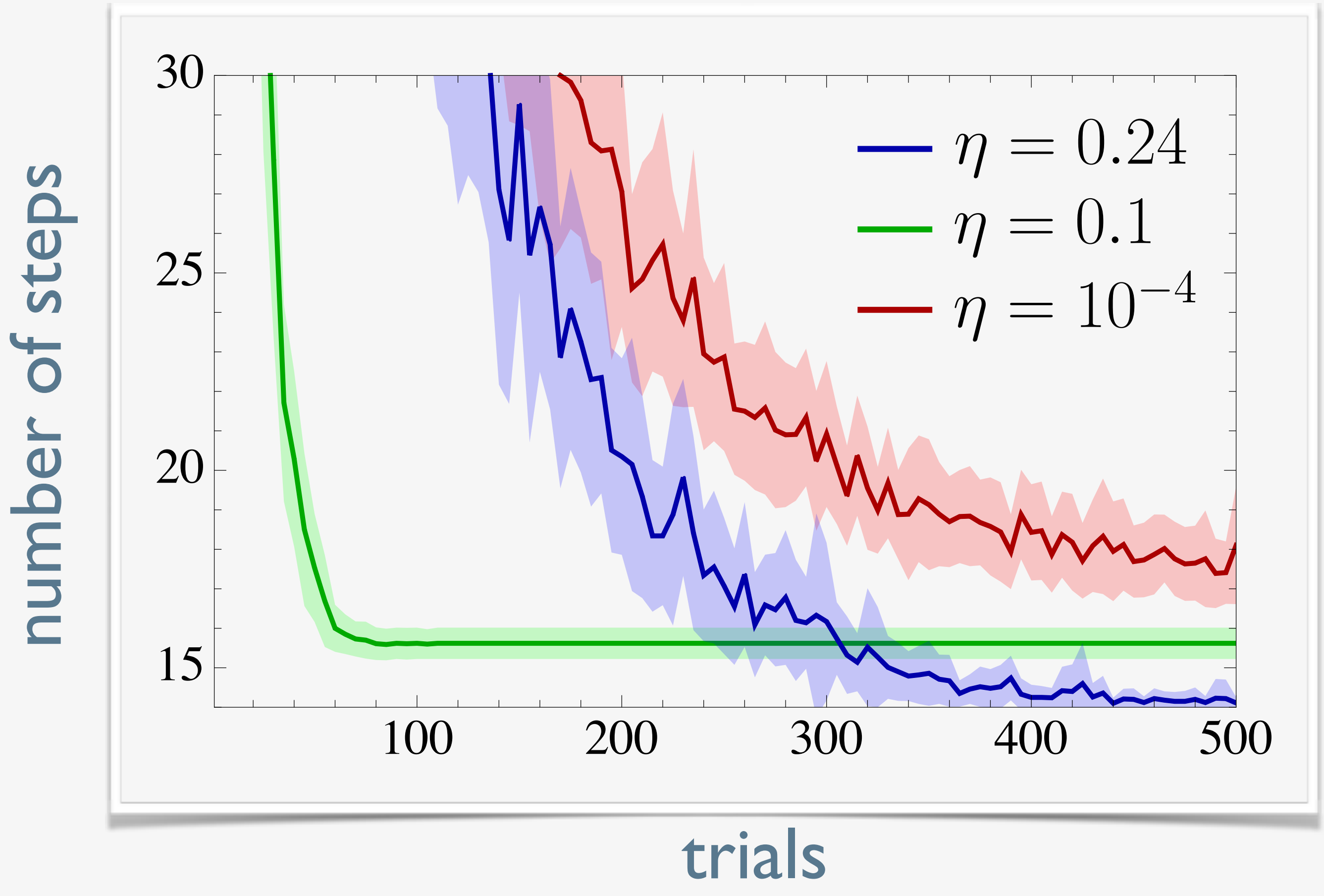


Mountain car problem

6 hours per agent to check the full parameter range

AAM, A. Makmal, and H. J. Briegel, arXiv:1804.08607, accepted in IEEE Access

Glow η parameter: heuristics



Two basic rules

- world size $\uparrow$ — $\eta \downarrow$
  world size $\downarrow$ — $\eta \uparrow$
  given that the maximum number of trials is the same
- maximum number of trials $\uparrow$ — $\eta \uparrow$
  maximum number of trials $\downarrow$ — $\eta \downarrow$
  given that the world size is the same

Glow η parameter: analytics

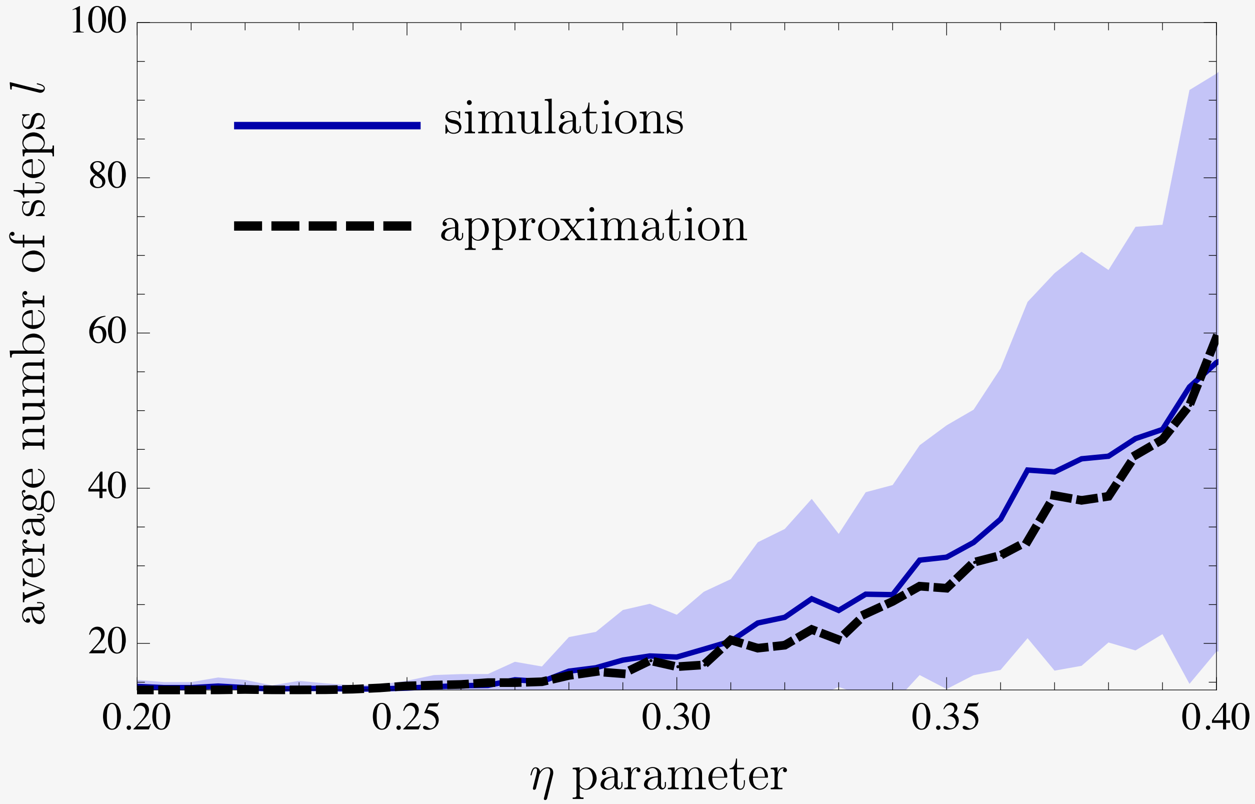
lower bound on success probability

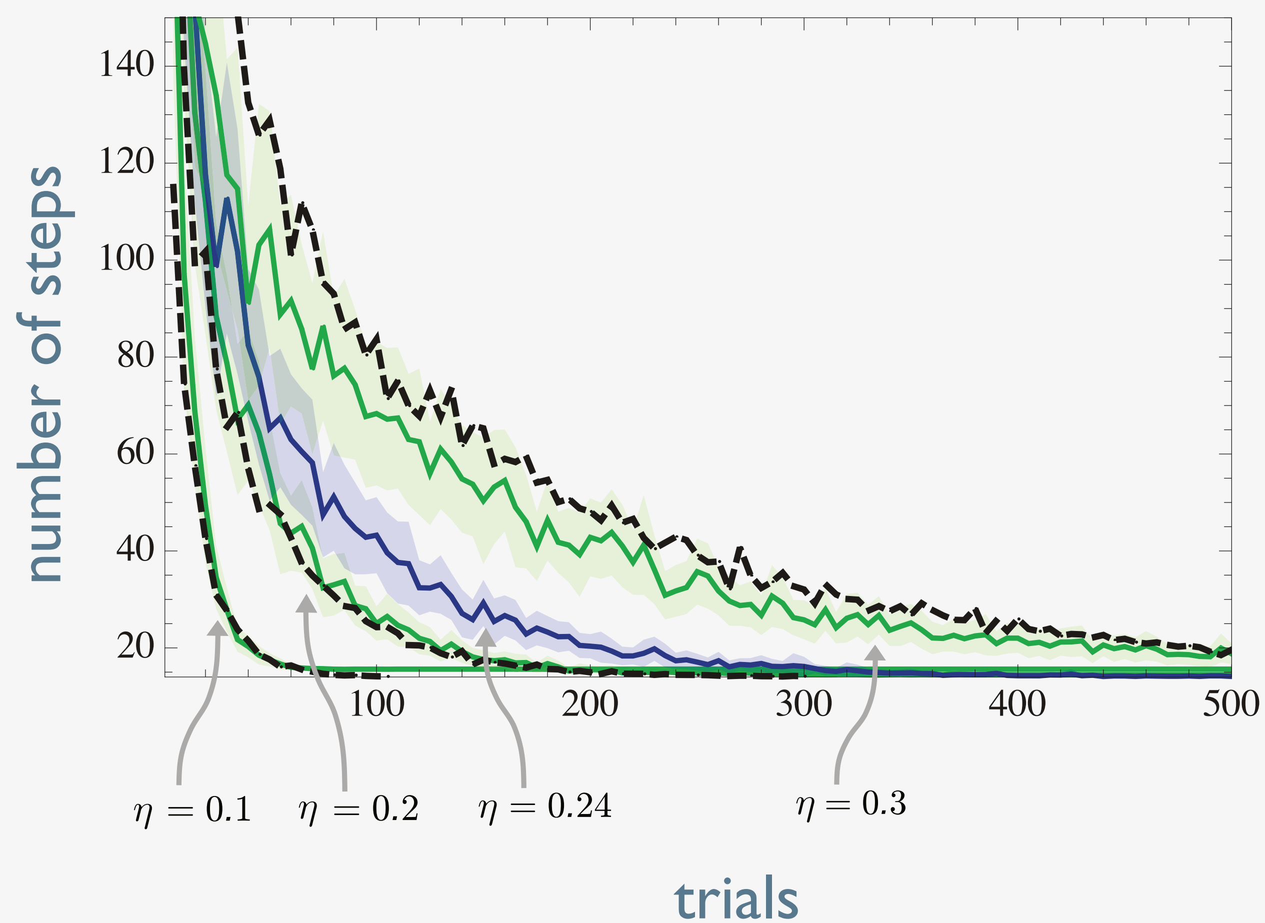
$$p_{l=L}(t) = \prod_{m=0}^{L-1} \frac{1 + t\lambda(1-\eta)^m}{K + t\lambda(1-\eta)^m}$$

relation between learning times

$$\frac{\widetilde{T}(\eta_1)}{\widetilde{T}(\eta_2)} = \frac{\left(\frac{1}{\eta_1} - 1\right)\left((1-\eta_1)^{-L} - 1\right)}{\left(\frac{1}{\eta_2} - 1\right)\left((1-\eta_2)^{-L} - 1\right)}$$

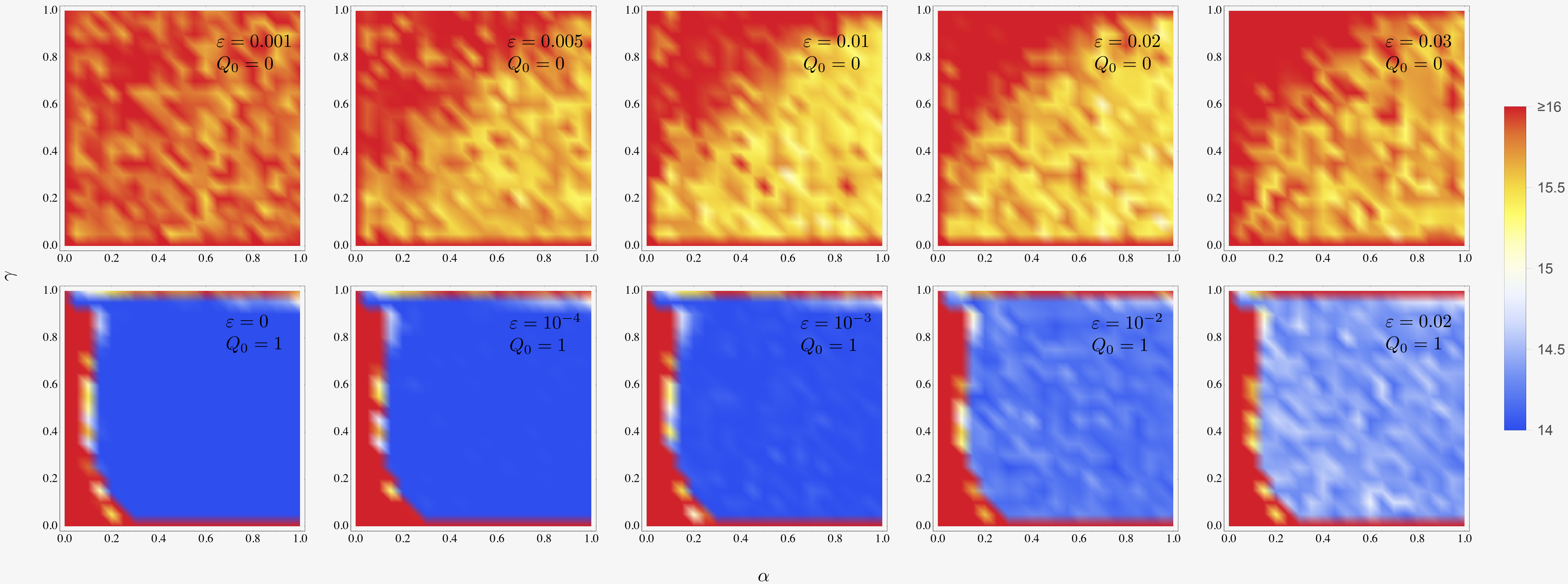
Approximation on the learning curves



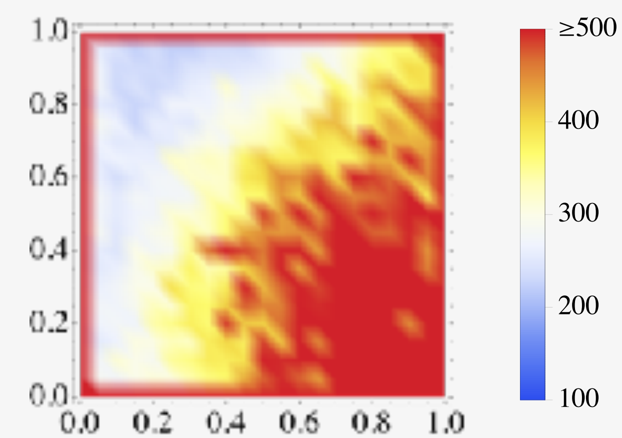


Choice of model parameters

With standard tabular RL approaches it is usually more complex



Q-learning agent



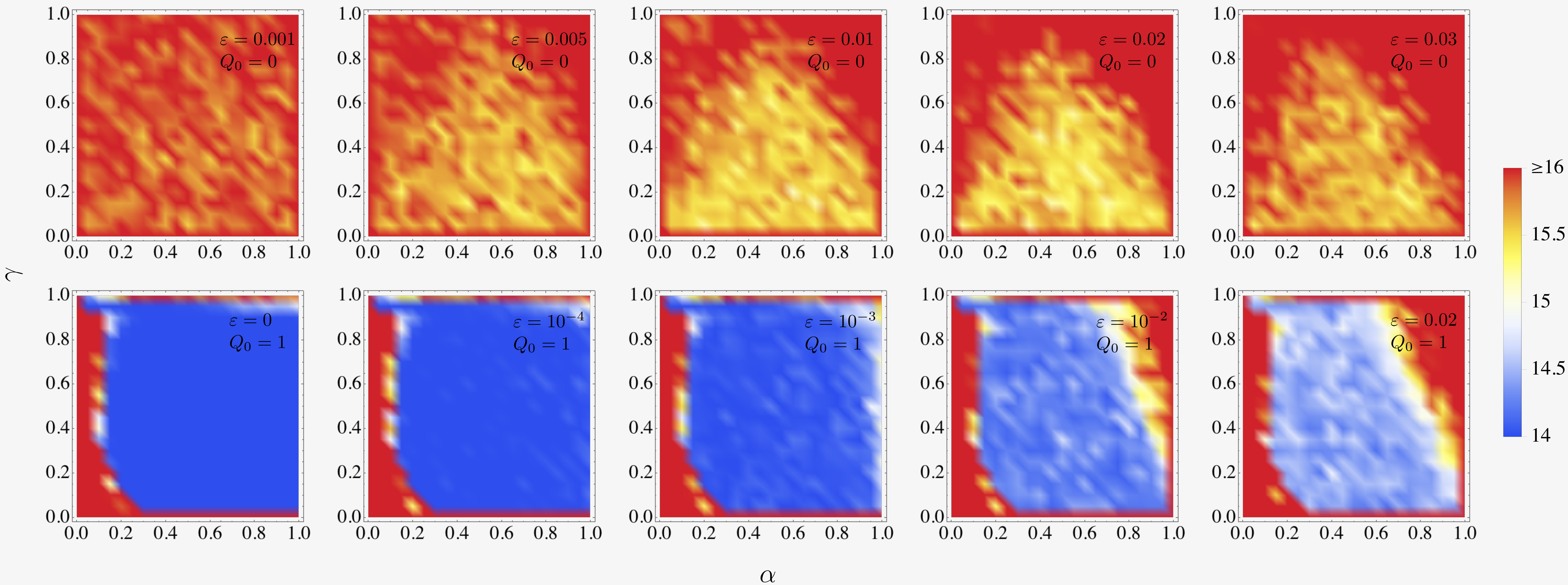
120 hours per agent in the grid world problem

180 hours per agent in the mountain car problem

AAM, A. Makmal, and H. J. Briegel, arXiv:1804.08607, accepted in IEEE Access

Choice of model parameters

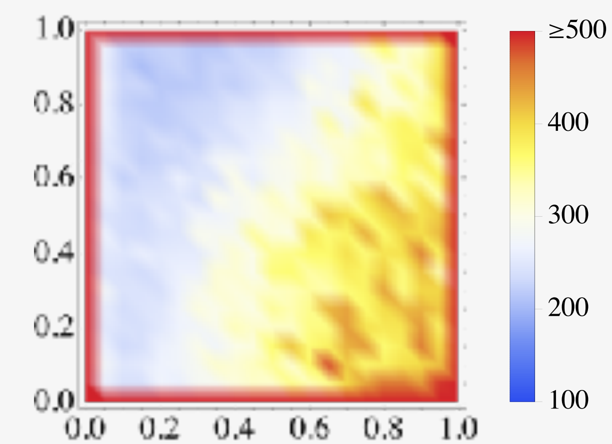
With standard tabular RL approaches it is usually more complex



SARSA agent

120 hours per agent in the grid world problem

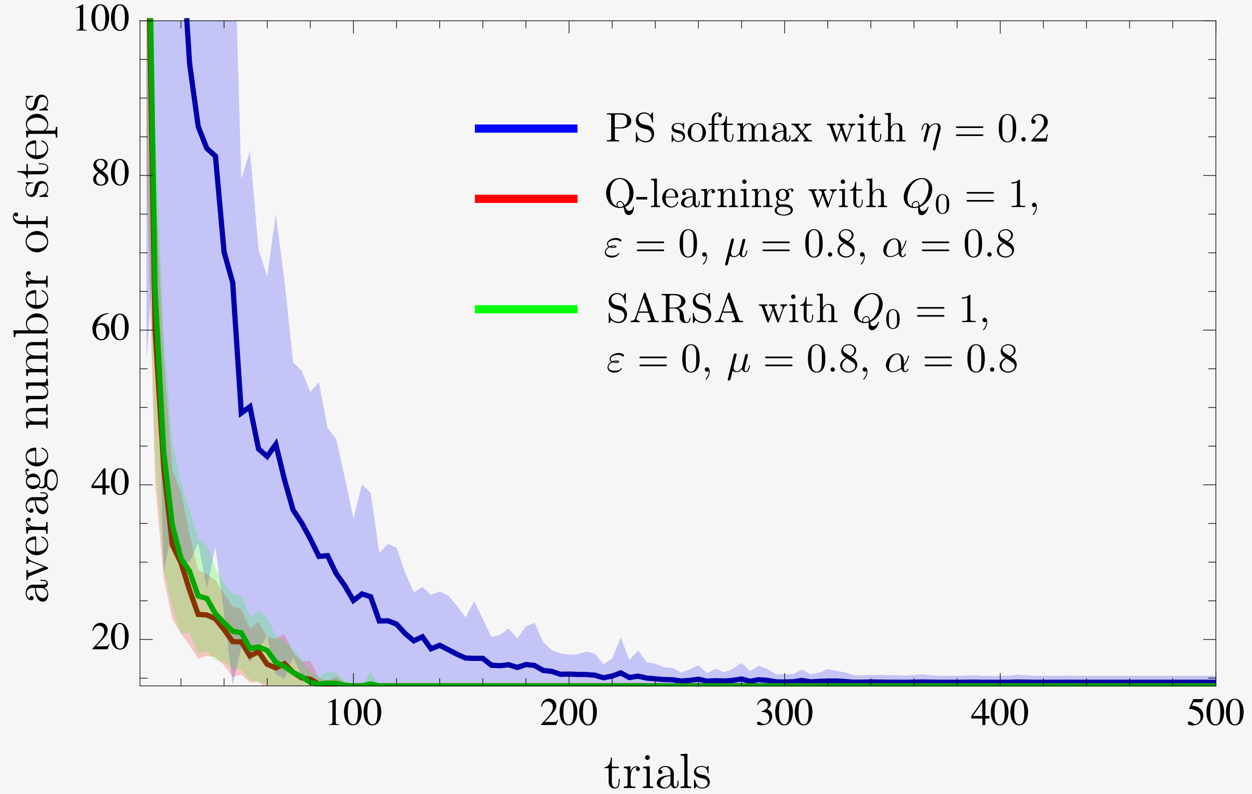
180 hours per agent in the mountain car problem



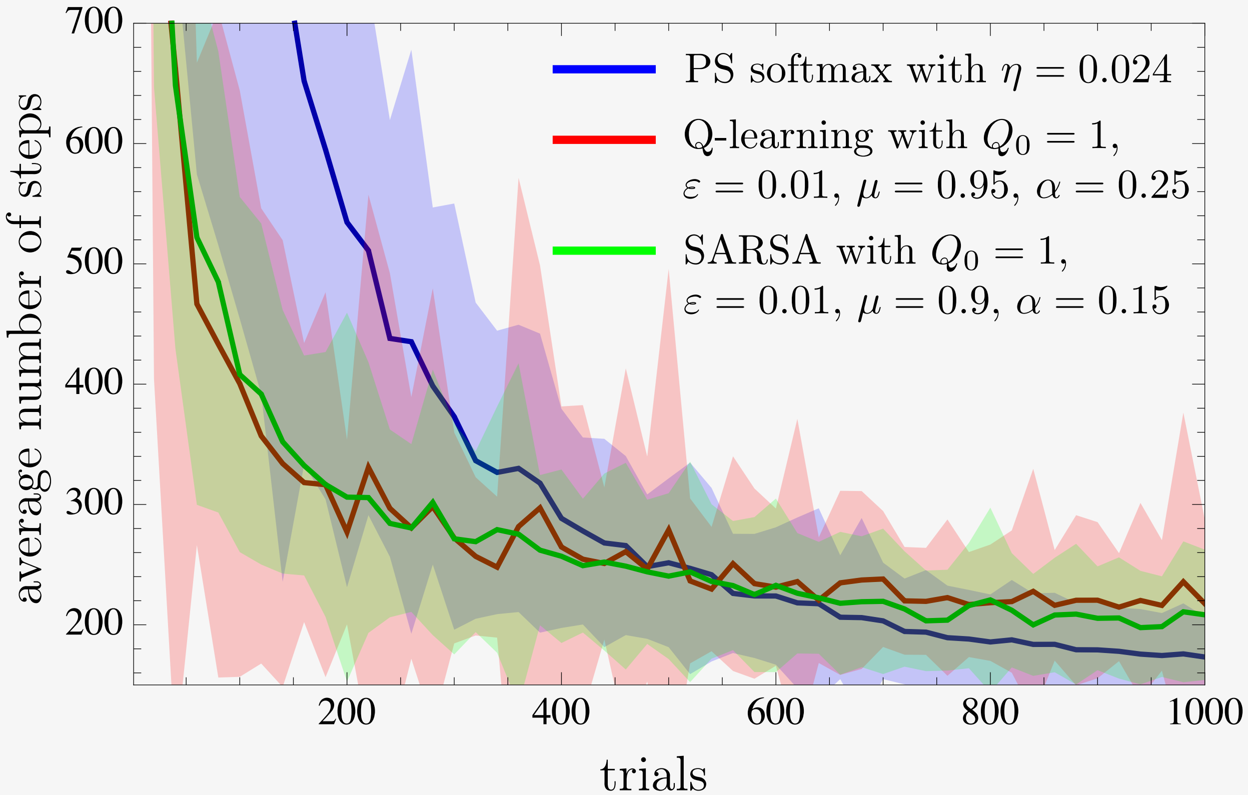
AAM, A. Makmal, and H. J. Briegel, arXiv:1804.08607, accepted in IEEE Access

Choice of model parameters

The performance is qualitatively and quantitatively similar



Grid world problem



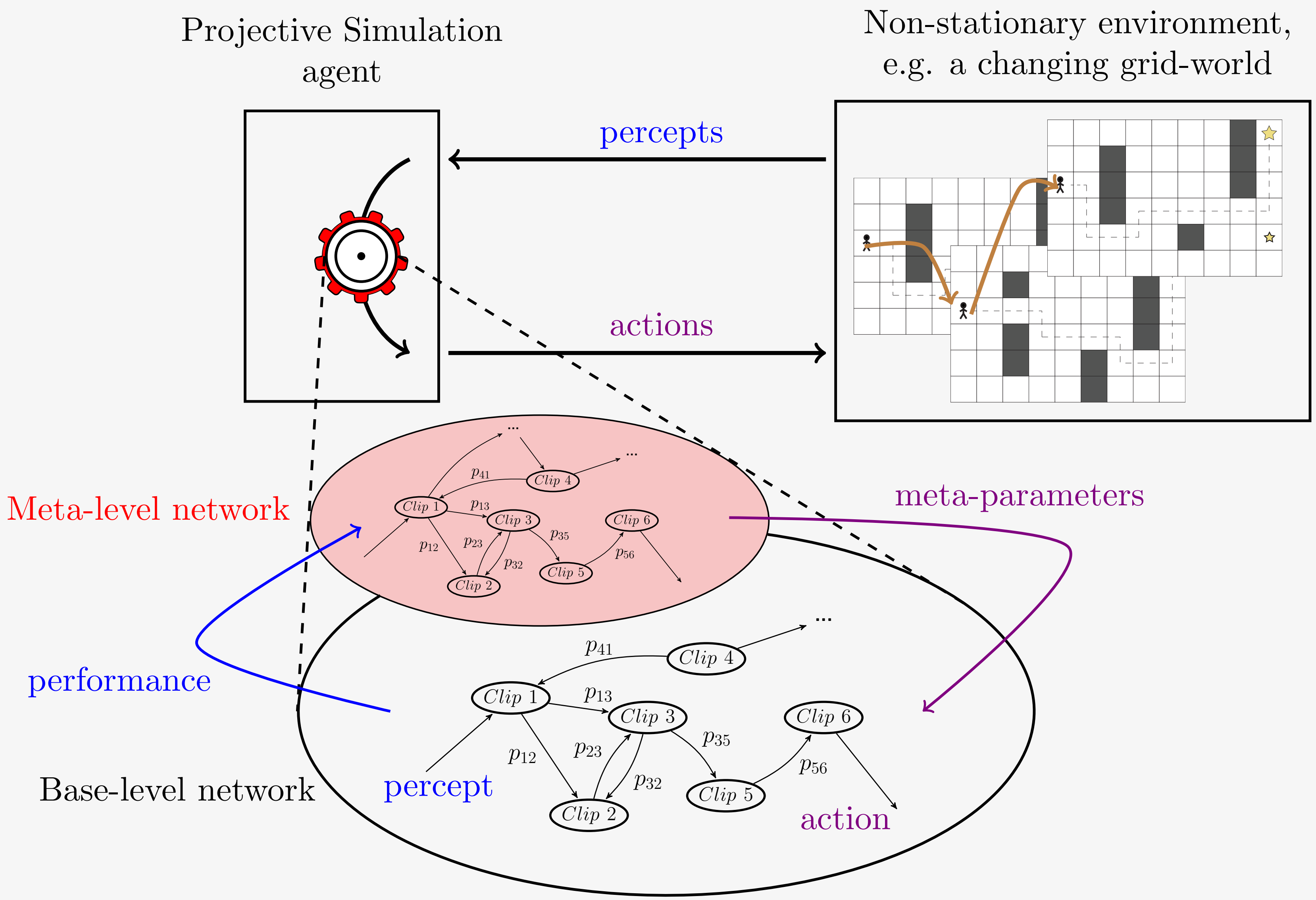
Mountain car problem

AAM, A. Makmal, and H. J. Briegel, arXiv:1804.08607, accepted in IEEE Access

Meta-learning within PS

Aske Plaat: meta-learning is one of the main ML challenges

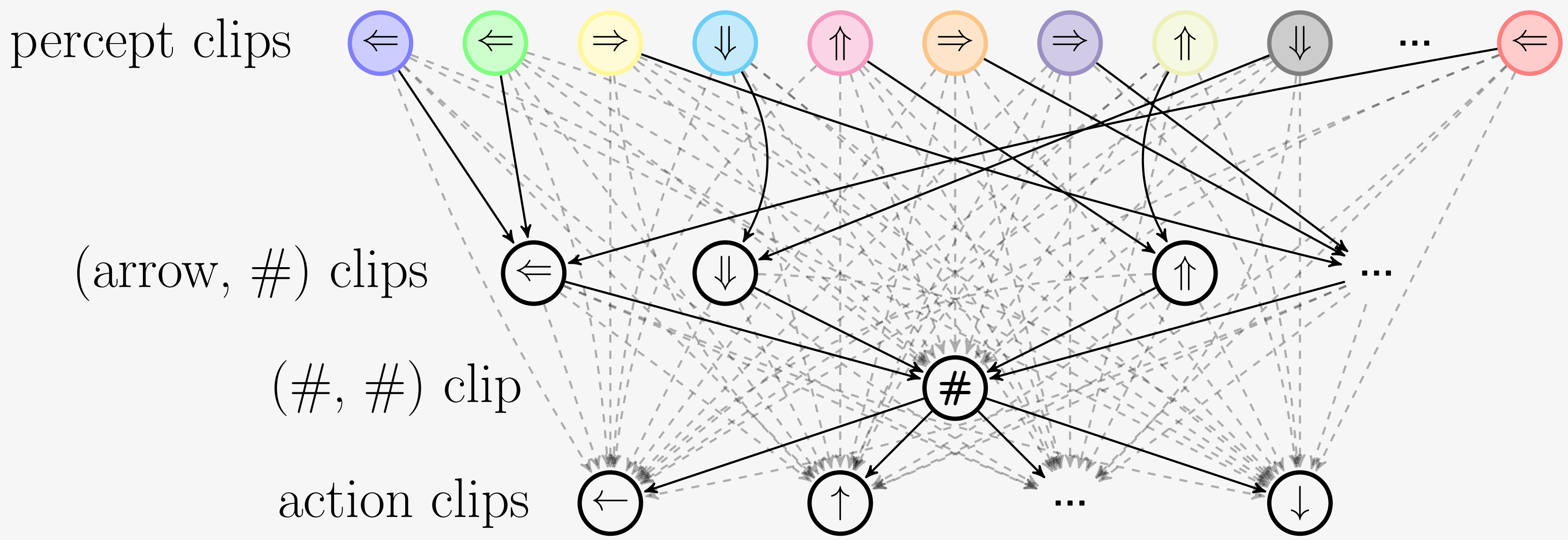
PS can naturally be extended to account for meta-learning in RL



A. Makmal, AAM, V. Dunjko, and H. J. Briegel, IEEE Access 4, 2110 (2016)

Generalization within PS

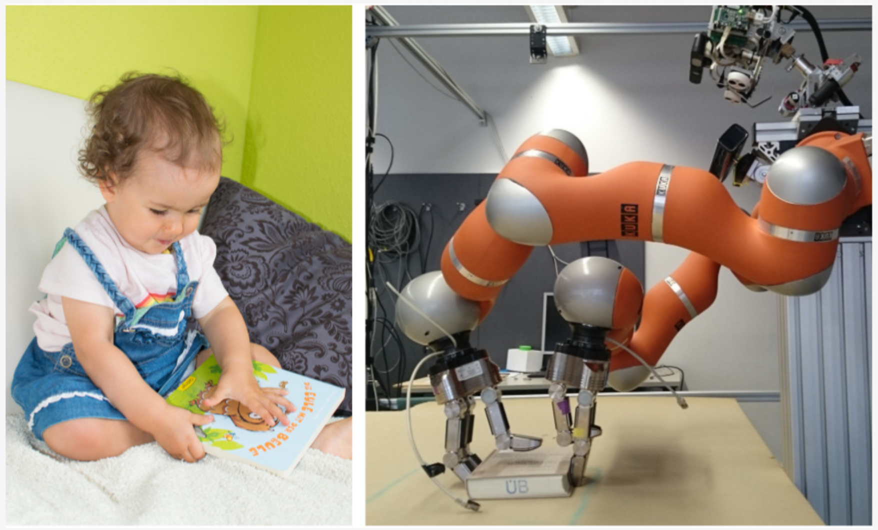
A dynamic and autonomous machinery that enables PS agents to generalize

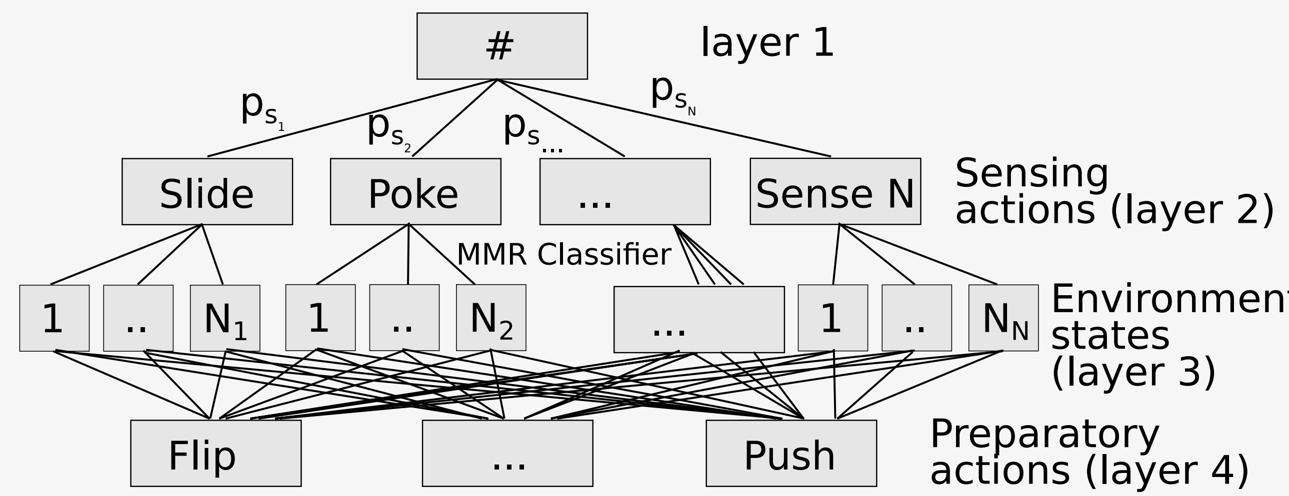


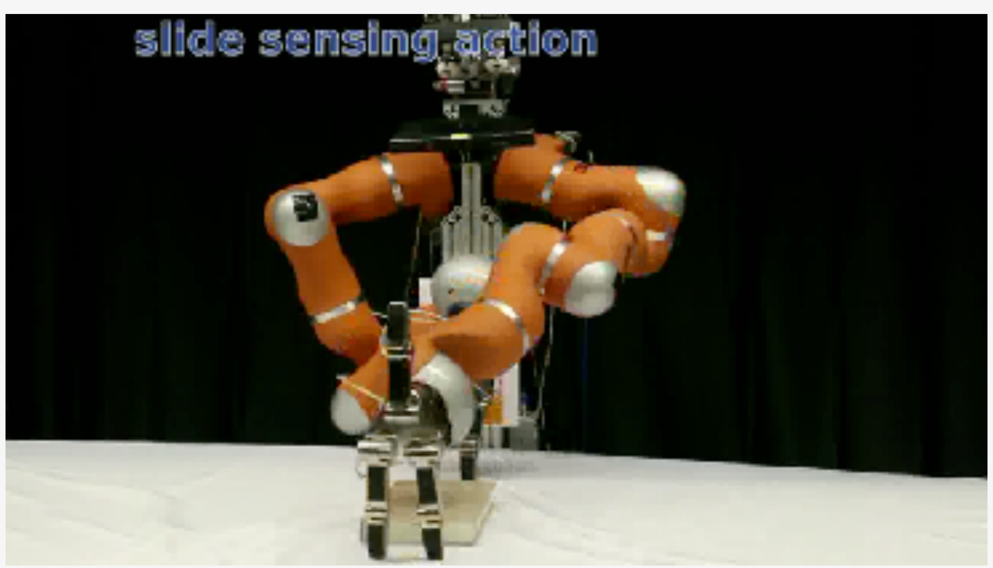
AAM, A. Makmal, V. Dunjko, and H. J. Briegel, Sci. Rep. 7, 14430 (2017)

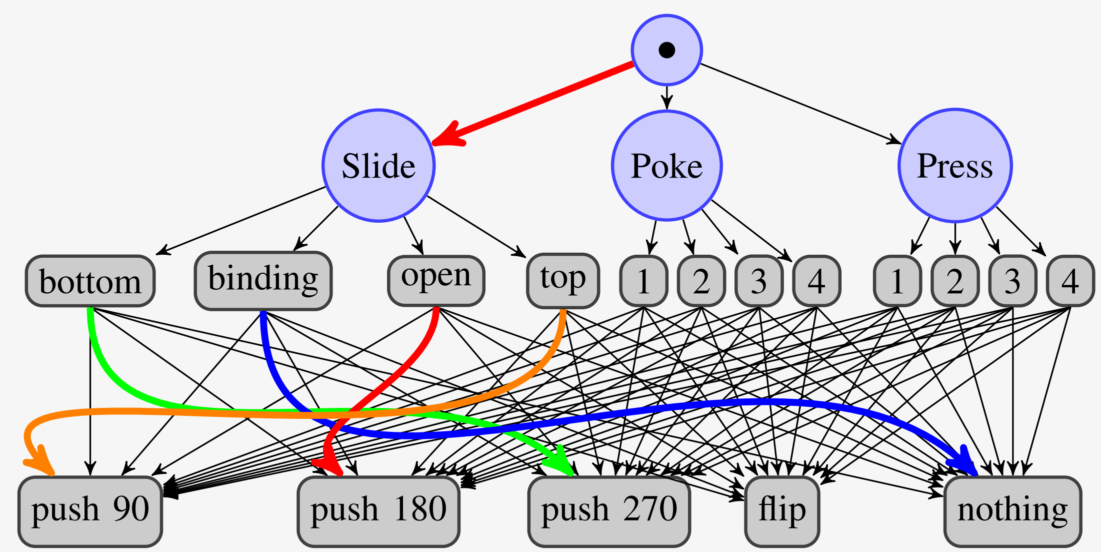
PS in robotics

PS in the problem of learning complex haptic manipulation skills









S. Hangl, E. Ugur, S. Szedmak, and J. Piater, IEEE/RSJ IROS, p. 2799 (2016)

A talk by Justus Piater on Friday

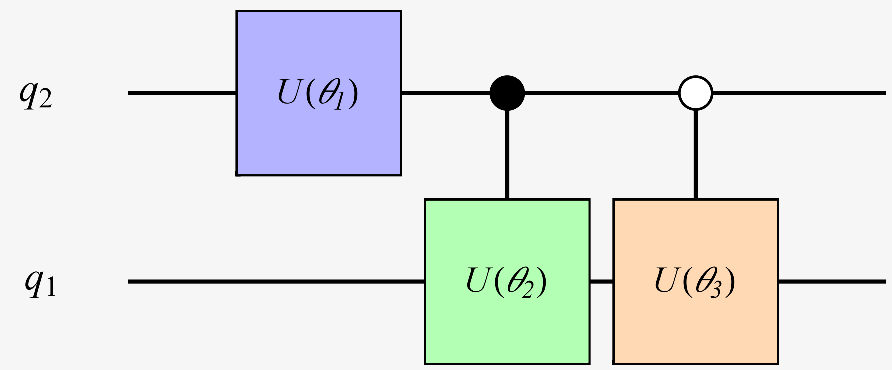
Quantum PS agent

A quantum state of the memory with N clips can be described by a state vector

$$|c_i\rangle = |i\rangle.$$

A quantum walk in the memory is characterised by N unitaries

$$U_i|0\rangle = \sum_{j=1}^{N} \sqrt{p_{ij}}\,|c_j\rangle.$$



Two-qubit probability unitaries for PS network with 4 memory units

G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, H. J. Briegel PRX 4, 031002 (2014)
V. Dunjko, N. Friis, and H. J. Briegel, New J. Phys. 17(2), 023006 (2015)
N. Friis, AAM, G. Kirchmair, and H. J. Briegel, Sci. Rep. 5, 18036 (2015)

Quantum PS agent

- quadratic speed-up in preparing a stationary distribution

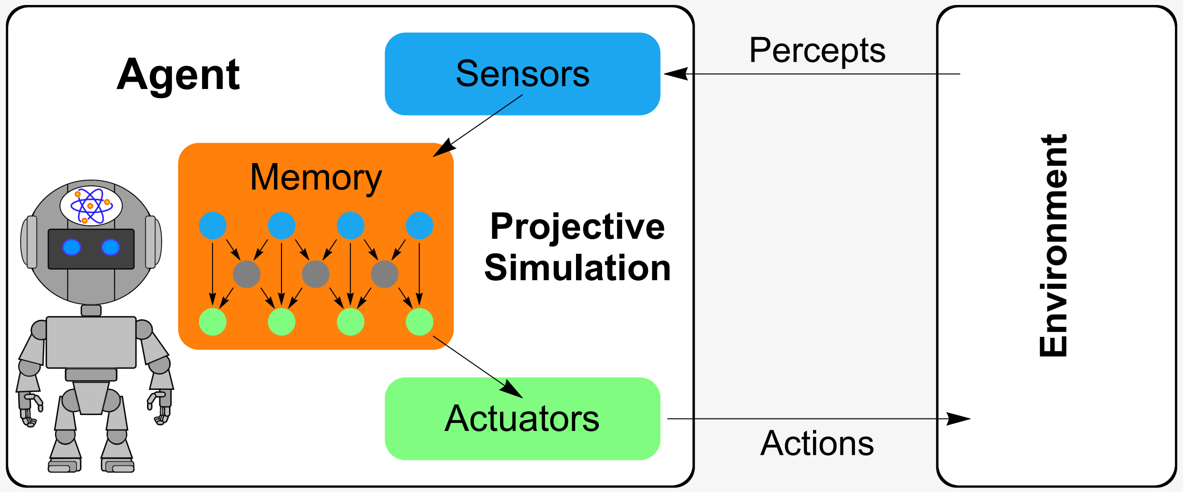
$$O\left(\frac{1}{\sqrt{\delta}}\right)$$

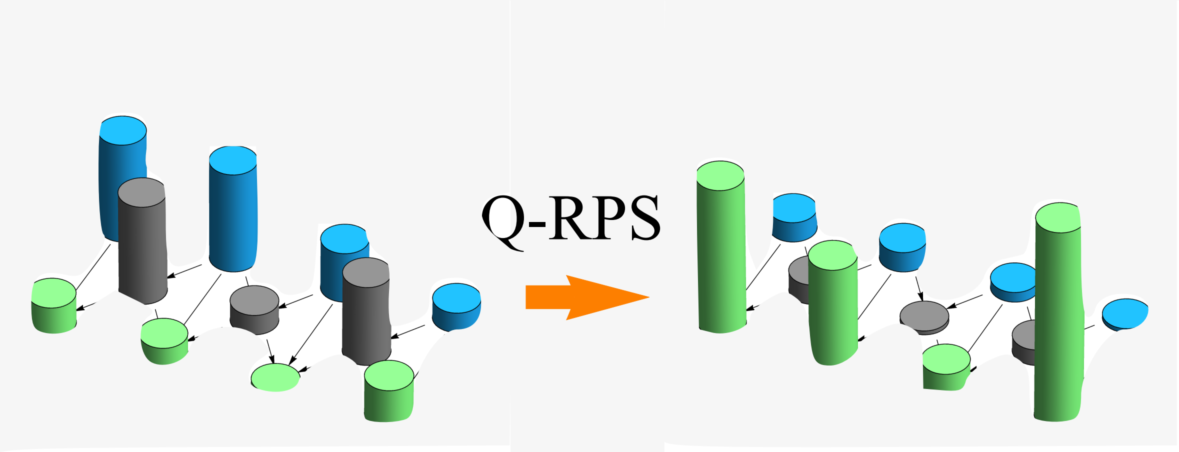
δ - spectral gap of the stochastic matrix

- quadratic speed-up in sampling an action

$$O\left(\frac{1}{\sqrt{\varepsilon}}\right)$$

ε - probability of sampling an action from the stationary distribution



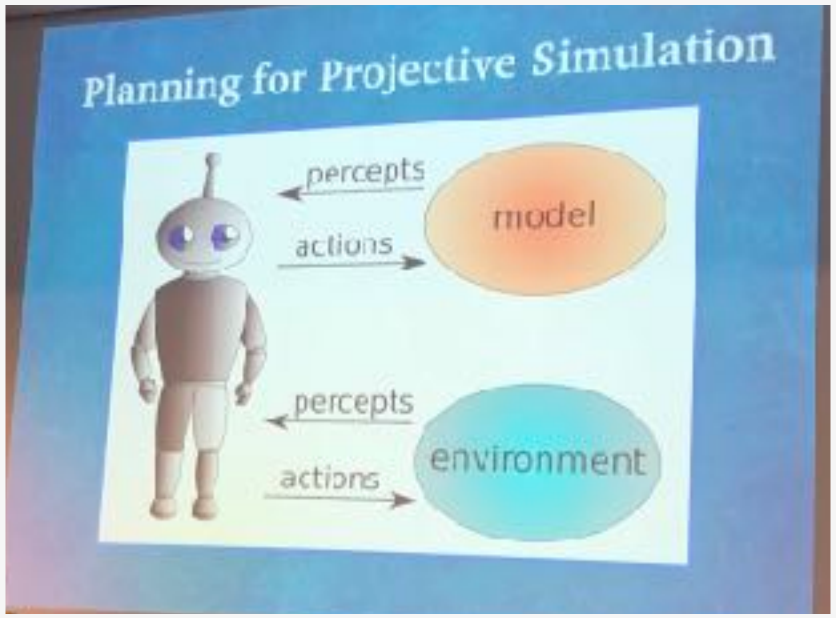


G. D. Paparo, et al., PRX 4, 031002 (2014)

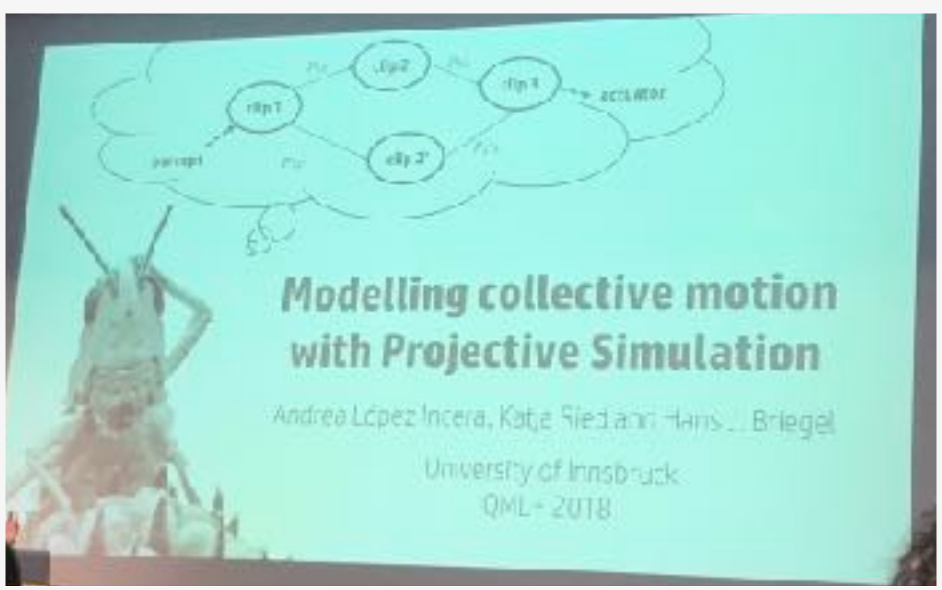
T. Sriarunothai, et al., arXiv:1709.01366

There are several posters about PS

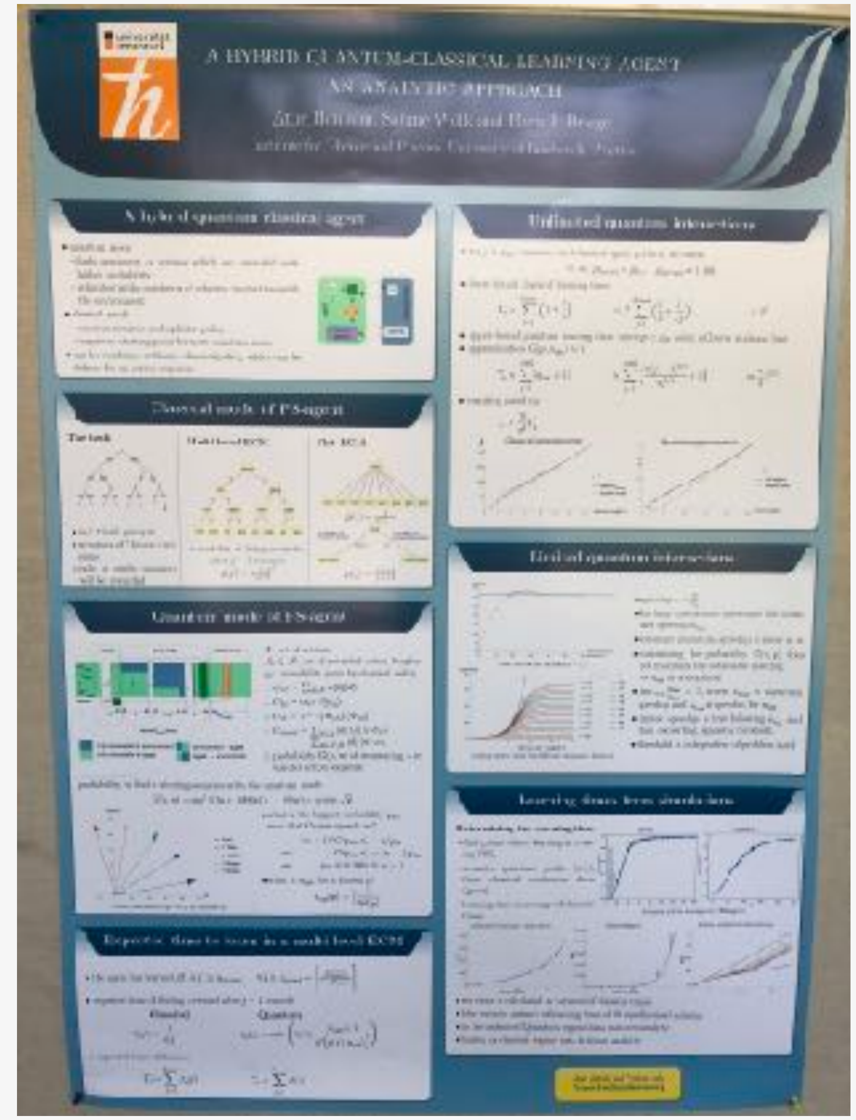
- Lea Trenkwalder



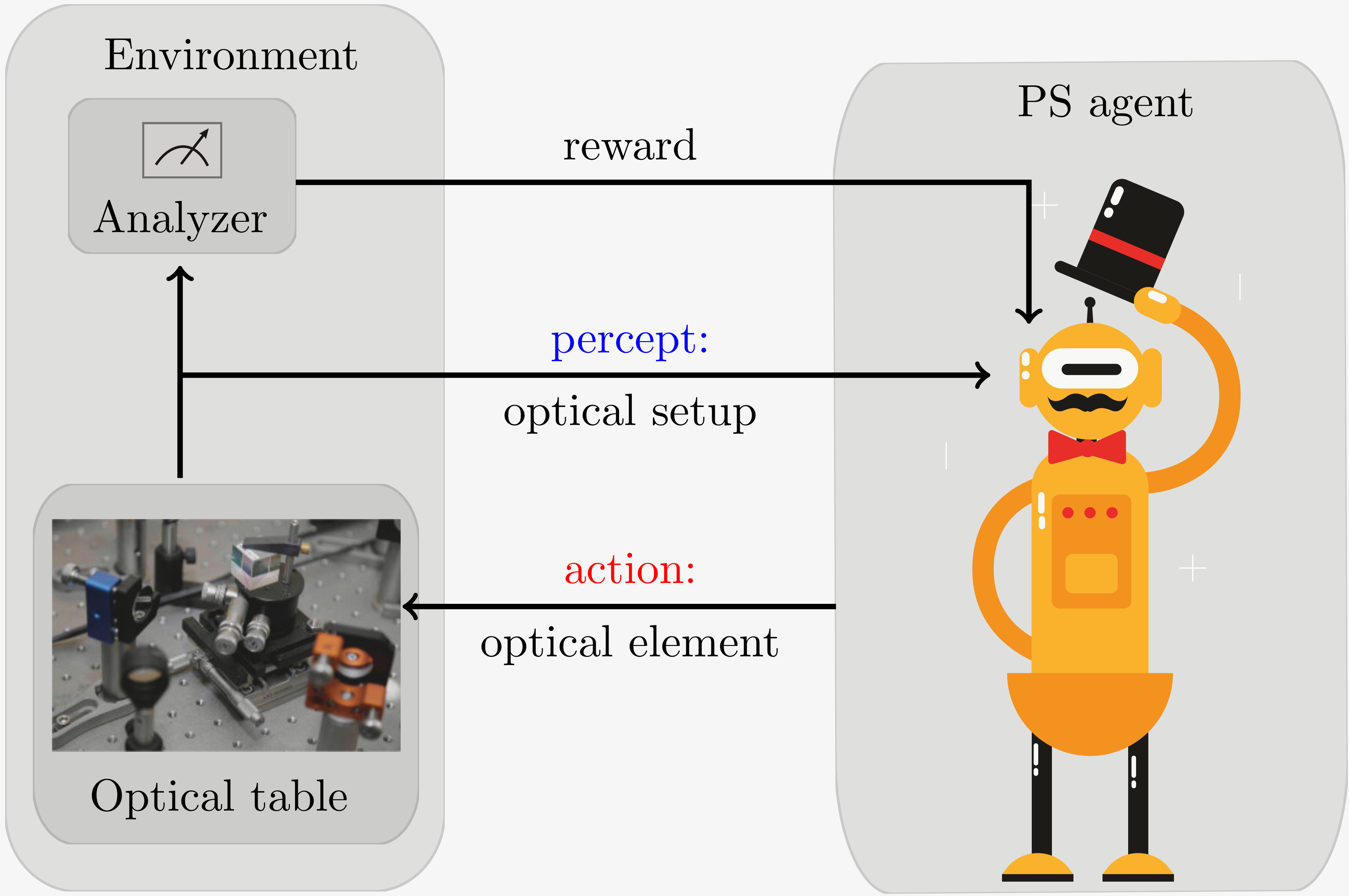
- Andrea López-Incera



- Arne Hamann

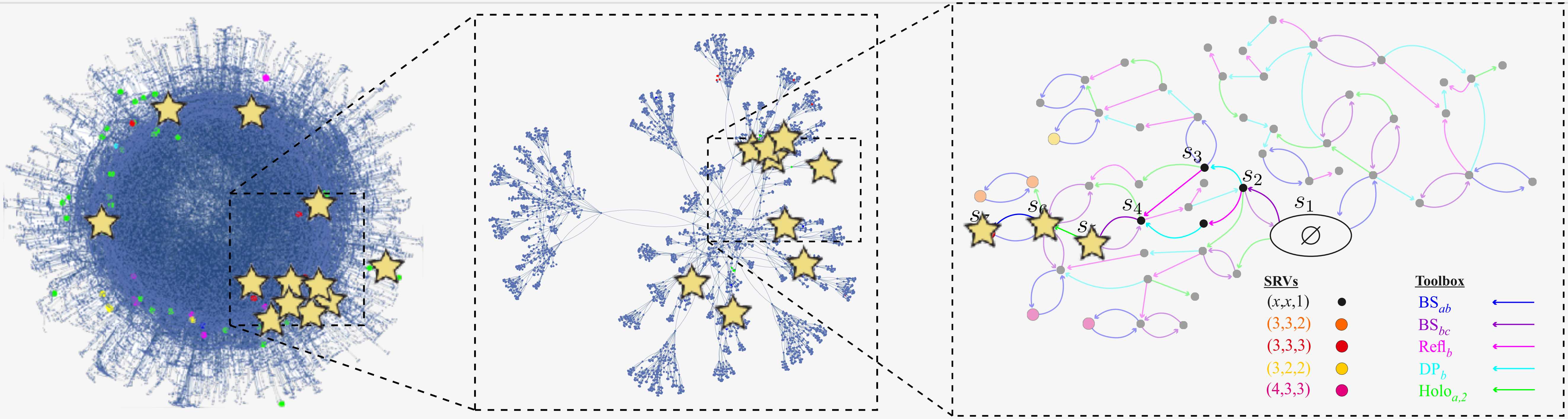


RL in quantum laboratory



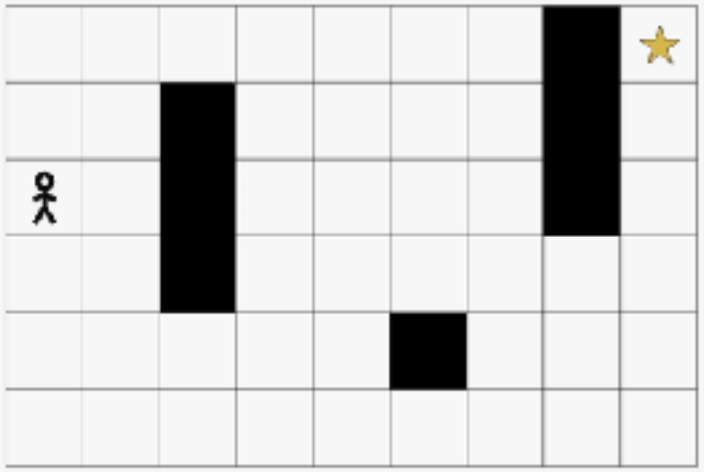
AAM, H. Poulsen Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, PNAS 115, 1221 (2018)

Why RL?

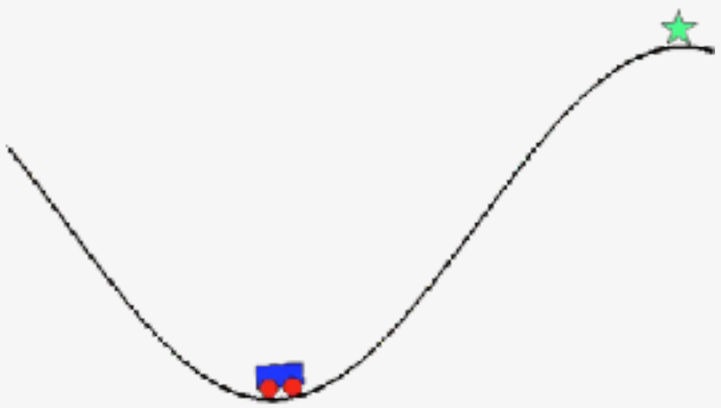


Exploration space *Scale-free network* *Complex maze*

Design of new experiments is a navigation in a complex network

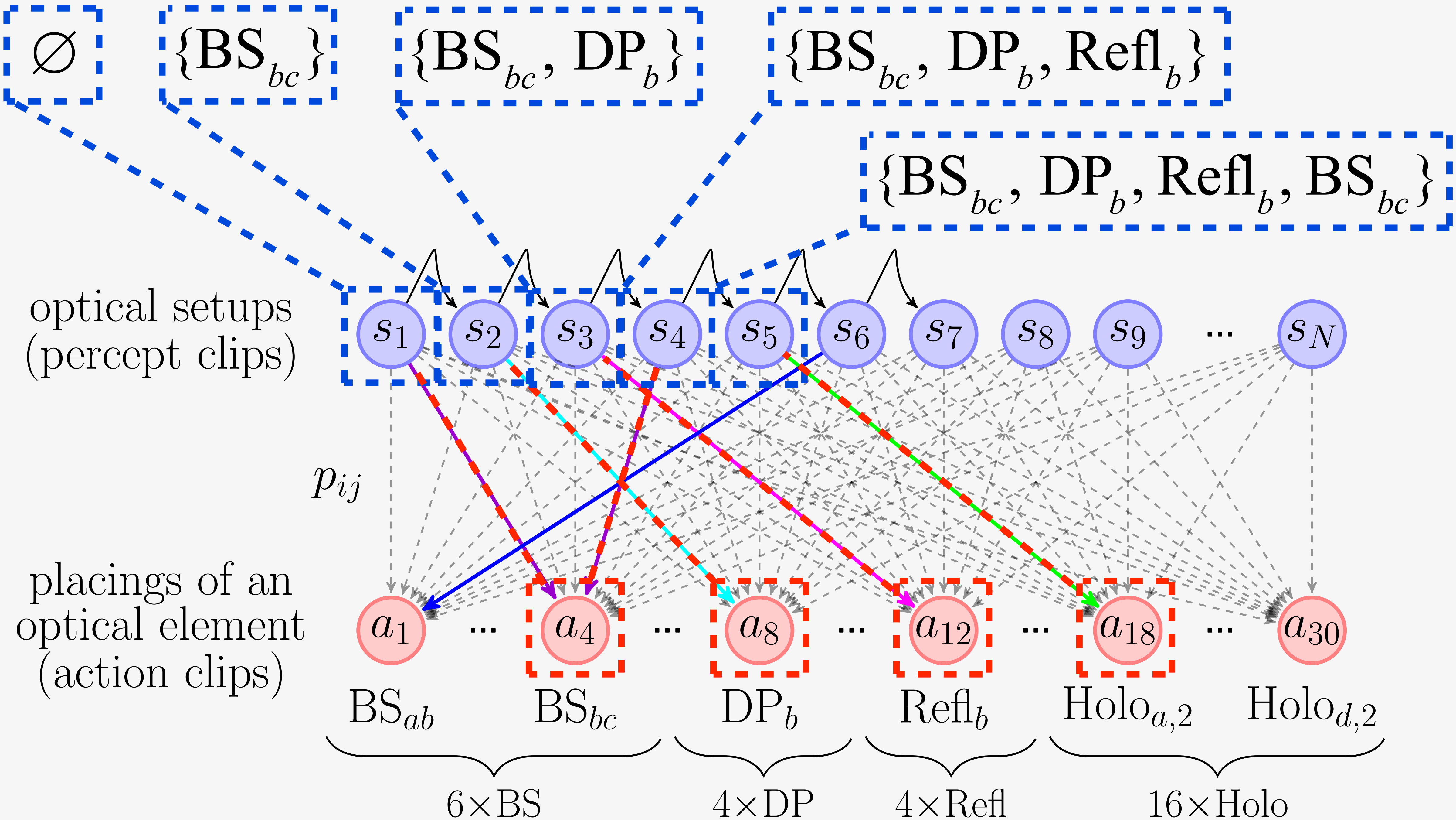


Navigation in the maze



Mountain car task

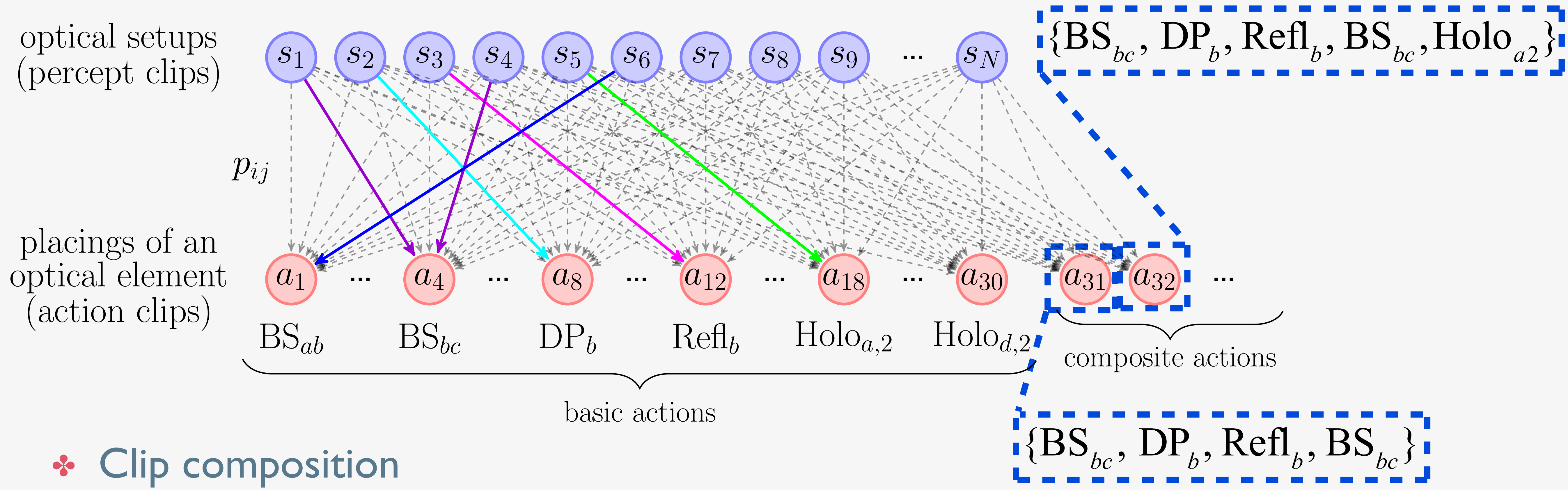
Episodic memory of the PS agent



Learning algorithm

$$h_{ij}^{(t+1)} = h_{ij}^{(t)} - \gamma(h_{ij}^{(t)} - 1) + g_{ij}^{(t)}\lambda, \quad g^{(t+1)} = (1-\eta)g^{(t)}$$

Clip composition and clip deletion



- **Clip composition**

  We create composite actions in case a sequence of actions is rewarded

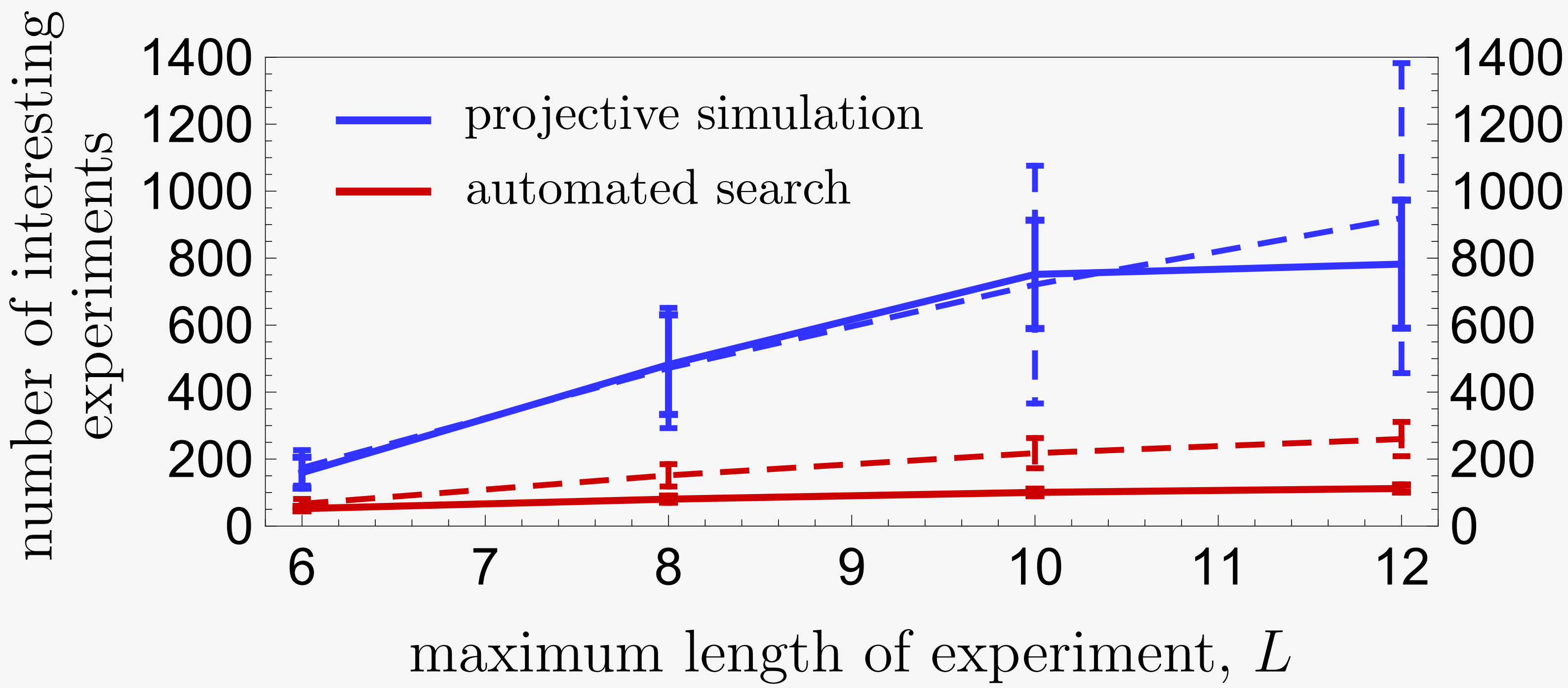
- **Clip deletion**

  We delete percepts (with edges) if the experiment didn't show nontrivial SRV

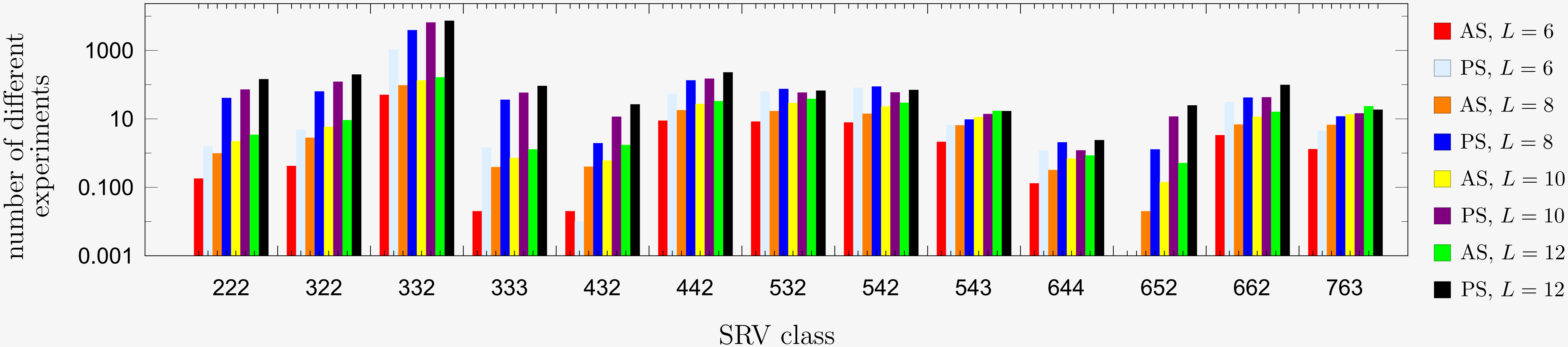
  We delete composite actions stochastically depending on their connectivities

$$p_{a_j}^{\mathrm{del}}(t) = \left(\frac{N(t)}{\sum_{k=1}^{N(t)} h_{kj}(t)} \right)^{N(t)} = \left(\frac{N(t) + N_R(t)}{N(t)} \right)^{-N(t)} \approx 1 - N_R(t)$$

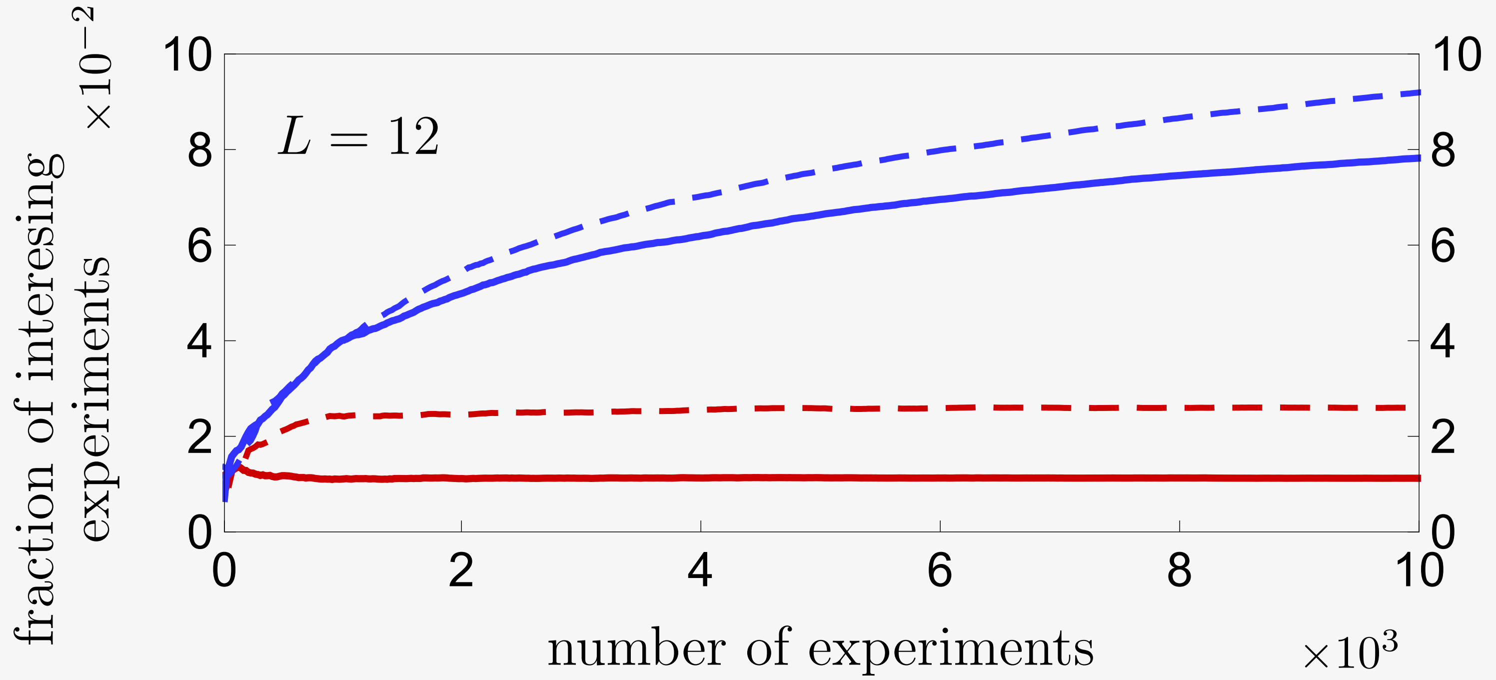
PS agent designs new quantum experiments



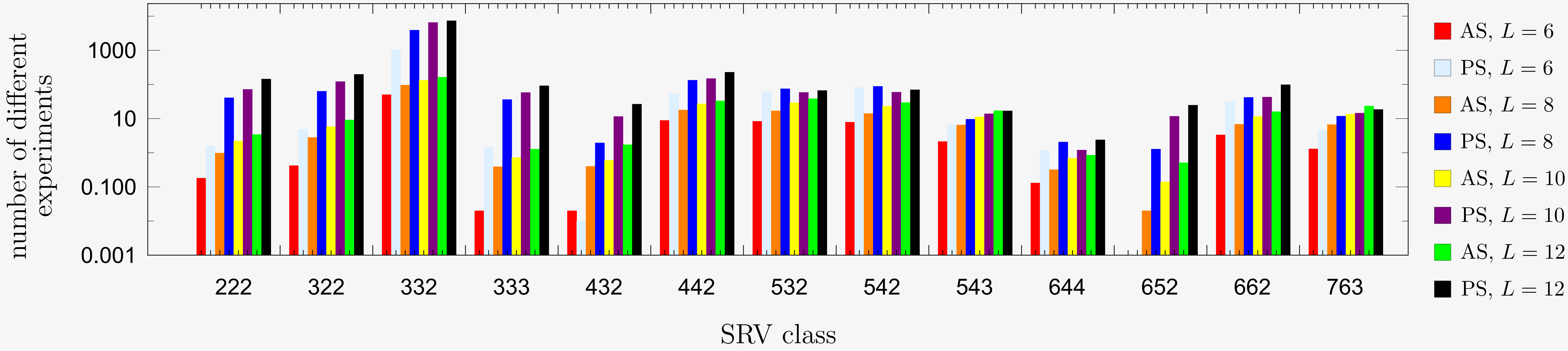
The PS agent has found many more interesting experiments, in comparison to the best previously known approach



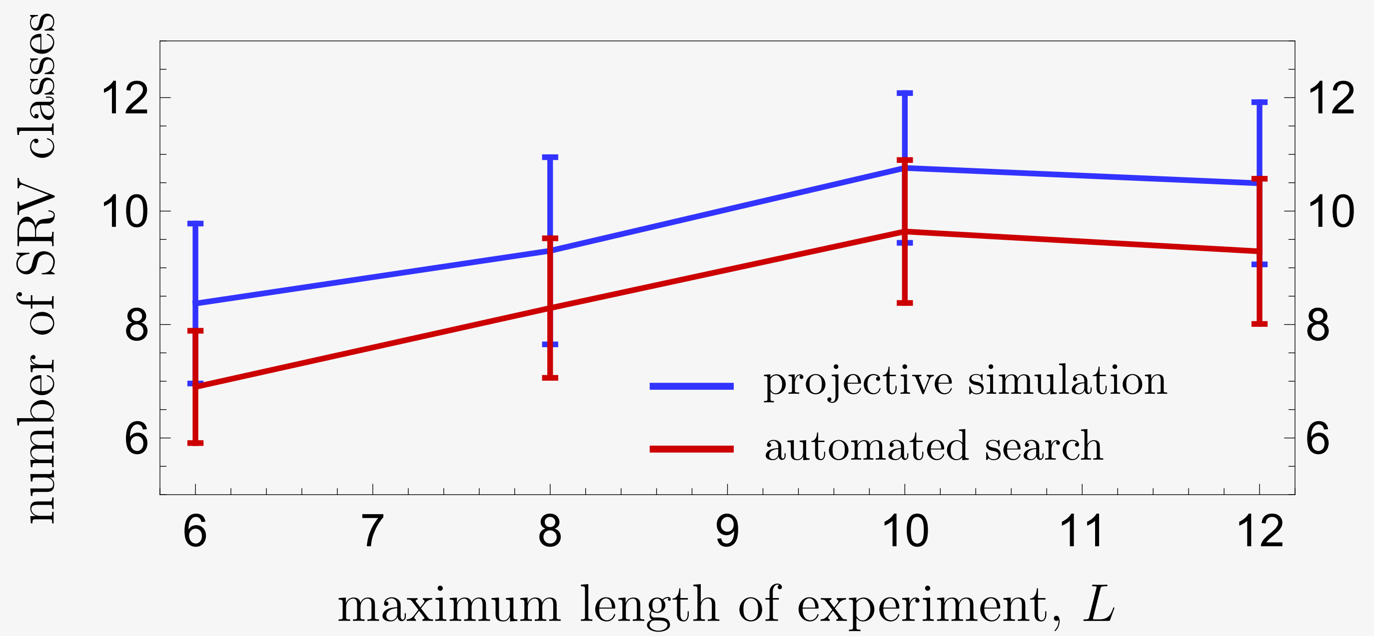
PS agent designs new quantum experiments



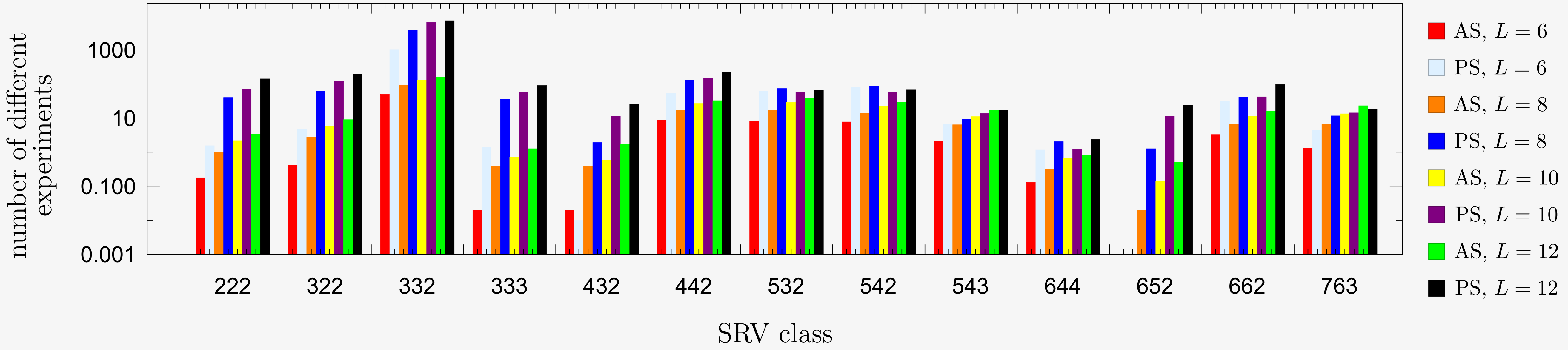
The PS agent has found many more interesting experiments, in comparison to the best previously known approach



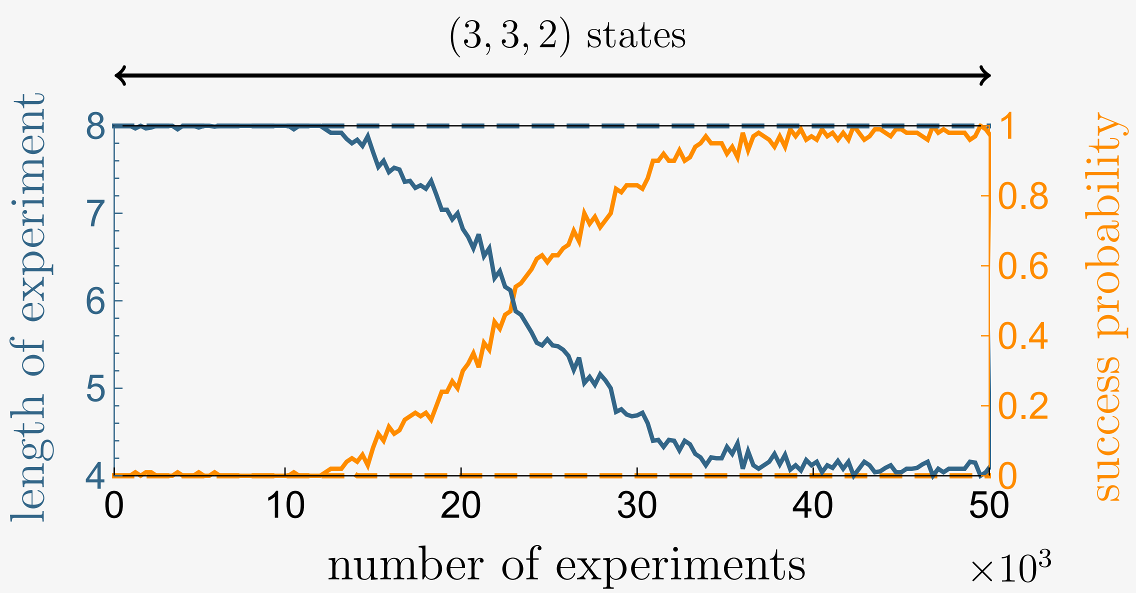
PS agent designs new quantum experiments

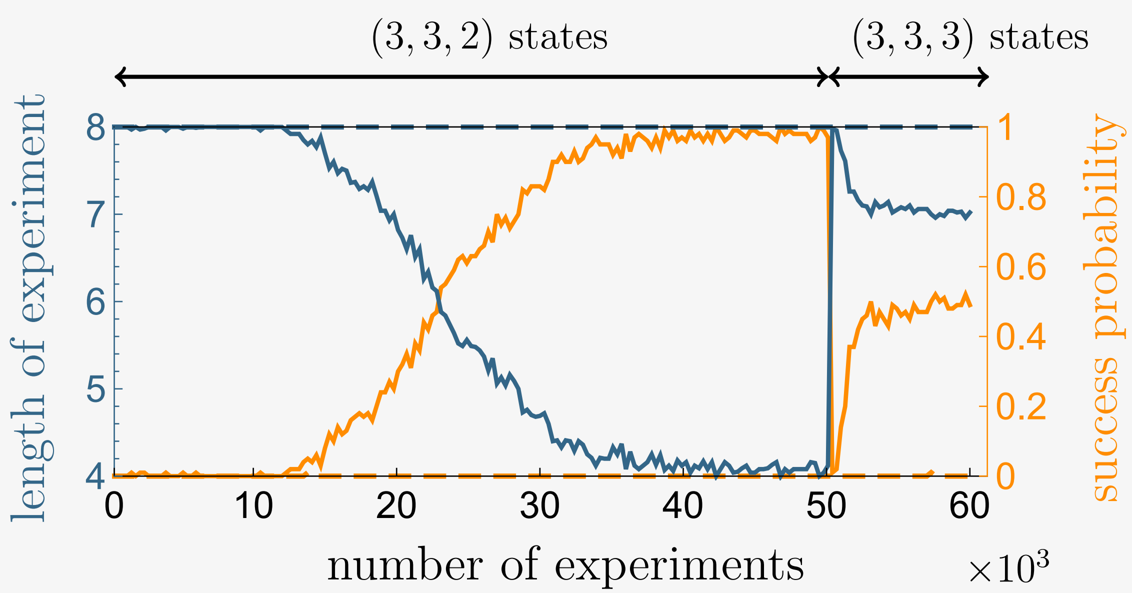


The PS agent has found many more interesting experiments, in comparison to the best previously known approach



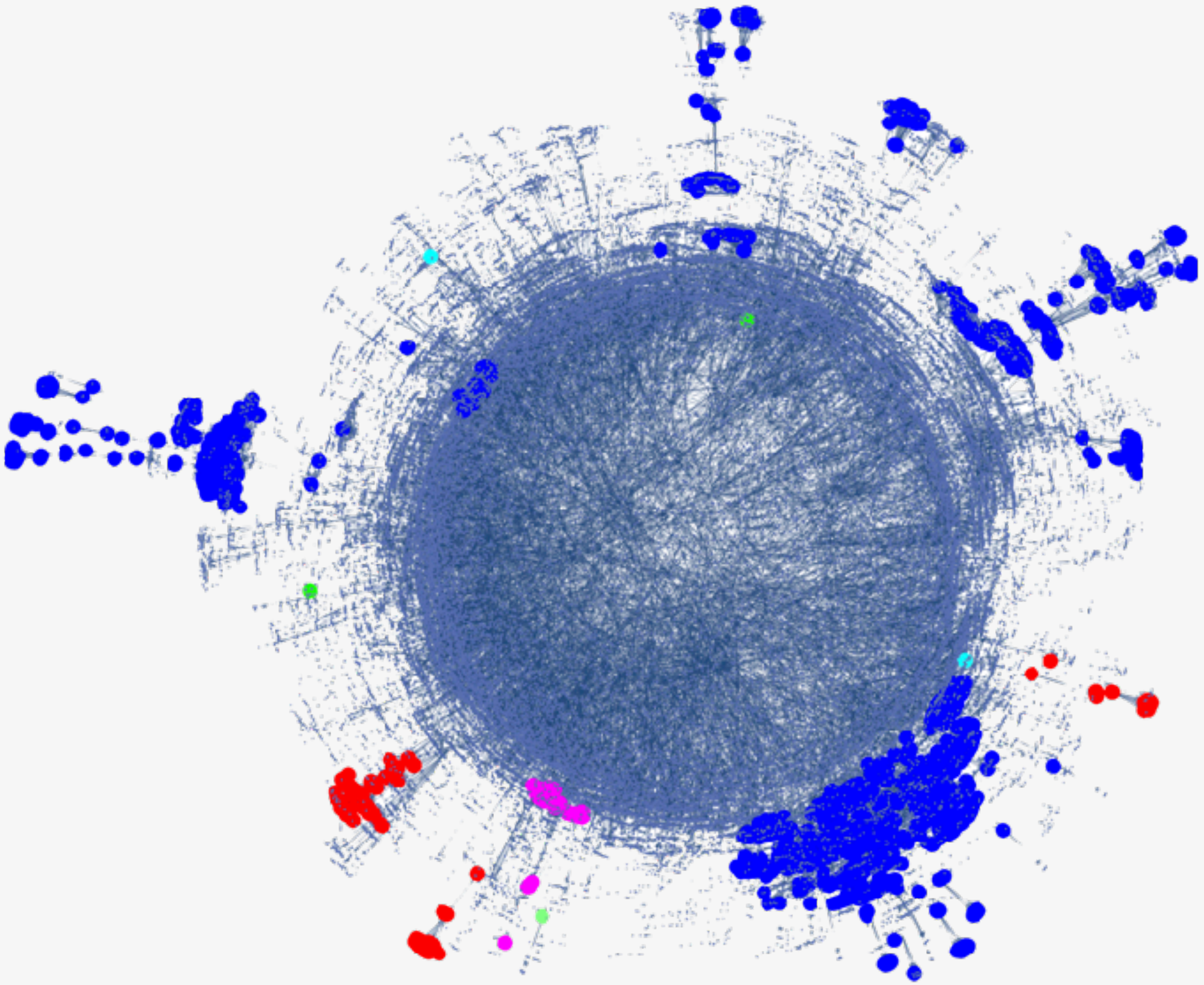
PS agent designs new quantum experiments





1. The PS agent autonomously learned to design target states (success curve)
2. The PS agent automatically learned to optimize the length of those experiments (length curve)
3. The PS agent uses the knowledge of building (3,3,2)-states to construct (3,3,3)-states (second phase curves)

Explored space of experiments



Discovering entanglement classes

{3, 2, 3}	9478	1355	1334	{Holo[1, -2], Refl[3], Holo[4, -1], BS[3, 4], BS[1, 4]}
{3, 4, 5}	662	11216	41	{Holo[1, -2], Refl[3], Holo[4, 2], BS[3, 4], BS[1, 4]}
{3, 6, 7}	235	11873	49	{Holo[1, -2], Refl[3], Holo[4, 3], BS[3, 4], BS[1, 4]}
{3, 2, 2}	2017	14189	126	{Holo[1, -1], BS[2, 3], BS[2, 4], DP[2], Refl[4], BS[2, 4]}
{2, 4, 4}	790	14311	170	{DP[1], Refl[1], BS[1, 3], BS[2, 4], Holo[4, 2], BS[3, 4]}
{3, 2, 4}	553	19043	22	{Holo[3, -2], BS[2, 3], BS[2, 4], DP[2], Refl[4], BS[2, 4]}
{2, 2, 2}	55	20178	8	{Holo[1, 2], BS[2, 4], DP[2], Refl[4], BS[2, 4], BS[1, 4]}
{3, 3, 3}	43	20178	5	{Holo[1, 2], BS[2, 4], DP[2], Refl[4], BS[2, 4], BS[1, 4]}
{3, 5, 2}	1385	22344	283	{Holo[1, 2], BS[1, 3], Holo[2, -1], BS[1, 2], DP[4], Holo[4, -2]}
{2, 6, 6}	691	23051	190	{Holo[1, 2], BS[1, 3], Holo[4, -4], BS[3, 4]}
{4, 2, 5}	1596	31806	453	{Holo[1, 2], BS[1, 4], Holo[1, -2], BS[1, 2], Refl[3], Holo[3, -1]}
{5, 3, 3}	6	84602	3	{Holo[4, 3], BS[1, 4], BS[2, 4], DP[2], Holo[3, 0], Refl[4], BS[2, 4]}

Certain elements combinations appear in different setups

Discovering entanglement classes

{2, 3, 3}	23131	2212	2364	{DP[1], BS[2, 4], Refl[3], BS[1, 3]}
{3, 2, 2}	338	5002	20	{DP[1], BS[2, 4], Refl[3], BS[1, 3], Holo[4, -1], BS[1, 4]}
{3, 5, 2}	1962	5608	474	{Holo[1, -1], Holo[2, 2], BS[2, 3], BS[1, 2]}
{2, 2, 2}	96	13615	8	{DP[1], BS[2, 4], Refl[3], BS[1, 3], Holo[1, 2], BS[1, 4]}
{3, 3, 3}	96	13615	8	{DP[1], BS[2, 4], Refl[3], BS[1, 3], Holo[1, 2], BS[1, 4]}
{4, 4, 2}	198	13618	21	{DP[1], BS[2, 4], Refl[3], BS[1, 3], Holo[2, 2], BS[2, 3]}
{4, 5, 2}	1894	17268	449	{Holo[1, -2], Holo[2, 2], BS[2, 3], BS[1, 2]}
{3, 5, 4}	11	19888	4	{Holo[1, -2], Holo[3, -1], Refl[3], Holo[3, 1], BS[3, 4], BS[1, 3]}
{3, 7, 6}	9	38218	2	{Holo[1, -2], Holo[3, -2], Refl[3], Holo[3, 1], BS[3, 4], BS[1, 3]}
{2, 6, 6}	461	51830	85	{DP[1], BS[2, 4], Refl[3], BS[1, 3], Holo[4, 3], BS[3, 4]}

Different agents have different sequences

These sequences can appear in different parts of an experiment

Episodic memory analysis

Let's look inside the memory of the PS agent
And output clips with the strongest connectivities

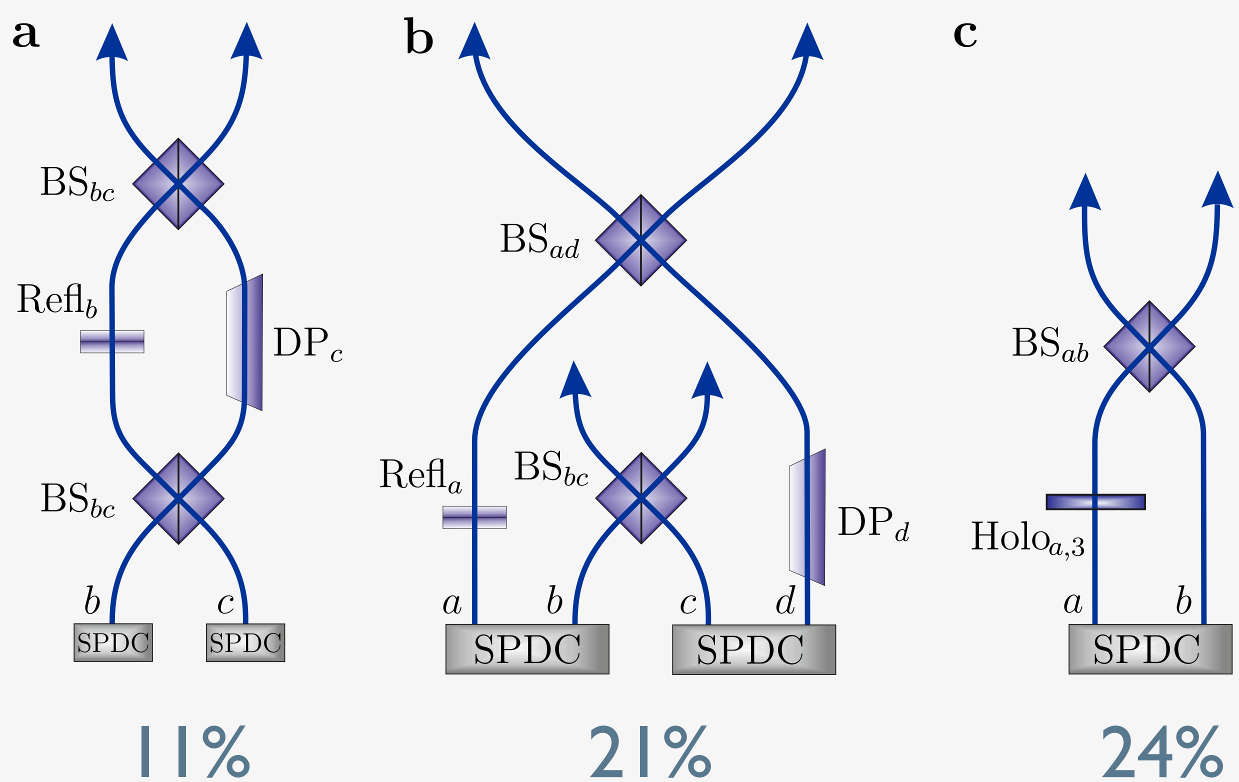
```
{BS[1, 4], DP[1, 1], Refl[4], BS[1, 4]} 2
{Holo[1, -2], Holo[4, -2], Refl[4], BS[3, 4], BS[1, 4]} 1
{BS[1, 3], DP[4, 1], Refl[4], BS[2, 4]} 1
{Holo[2, 1], Refl[3], Holo[4, -1], BS[3, 4], BS[1, 4]}  1
{Holo[1, 2], BS[1, 4], Holo[1, -2], BS[1, 2]}    1
{Holo[1, -1], DP[2, 1], Holo[3, 2], BS[1, 3], BS[3, 4]} 1
{DP[1, 1], Refl[3], Holo[3, 1], BS[3, 4], BS[1, 3]} 1
{Refl[4], Holo[4, -1], BS[3, 4], Refl[4], BS[1, 4]} 1
{Holo[2, 1], Holo[3, 1], Refl[4], BS[3, 4], BS[1, 3]}   2
{BS[1, 3], Holo[3, -2], DP[4, 1], Refl[4], BS[2, 4]}    1
{BS[1, 3], DP[2, 1], Refl[2], BS[2, 4]} 2
{BS[1, 4], DP[1, 1], Refl[1], BS[1, 4]} 2
{Refl[2], Holo[2, -1], BS[1, 2], Holo[1, -2], BS[1, 3]} 1
{Holo[1, -2], Holo[3, -2], Refl[4], BS[3, 4], BS[1, 4]} 1
{Holo[3, 1], Refl[4], BS[3, 4], Holo[3, -1], BS[1, 3]}  1
{Refl[1], Refl[3], Holo[4, -1], BS[3, 4], BS[1, 4]} 1
{DP[1, 1], BS[1, 3], Refl[2], BS[2, 4]} 2
{Holo[1, 2], Refl[2], BS[1, 2], Holo[3, -2], BS[1, 3]}  1
{Holo[1, -1], Refl[2], BS[1, 2], Holo[3, -1], BS[1, 3]} 1
{Refl[1], Holo[1, 2], Holo[2, -1], BS[1, 2], BS[1, 3]}  1
{Refl[1], BS[1, 3], DP[4, 1], BS[2, 4]} 1
{Holo[1, -2], Holo[3, -1], Refl[4], BS[3, 4], BS[1, 4]} 1
{Holo[1, 2], Holo[3, -2], Refl[4], BS[3, 4], BS[1, 3]}  1
{Holo[1, -2], Refl[2], BS[1, 2], Holo[1, 2], BS[1, 3]}  1
{Holo[1, 2], Holo[4, 1], Refl[4], BS[3, 4], BS[1, 4]}   1
```

…

Something surprising

Let's look inside the memory of the PS agent
And output clips with the strongest connectivities

Most connected clips:



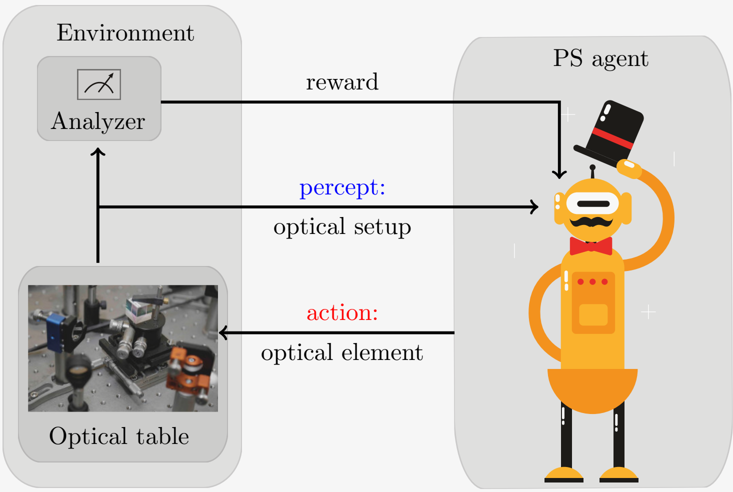
(a) — parity sorter, which was originally designed for a different task

(b) — new parity sorter, equivalent to (a) in the Klyshko wave front picture

(c) — new method to increase dimensionality of photons

Summary

- A search for new quantum experiments can be formulated as a RL problem

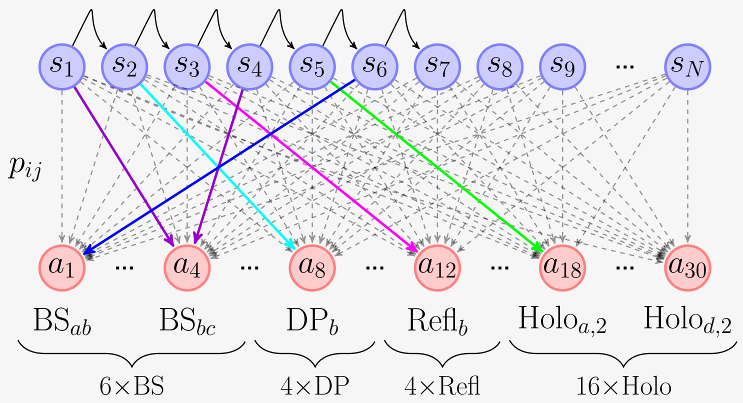


- Solving this RL problem with PS sets a new level of performance

  (1) many more interesting experiments are found

  (2) short implementations of these experiments are learned

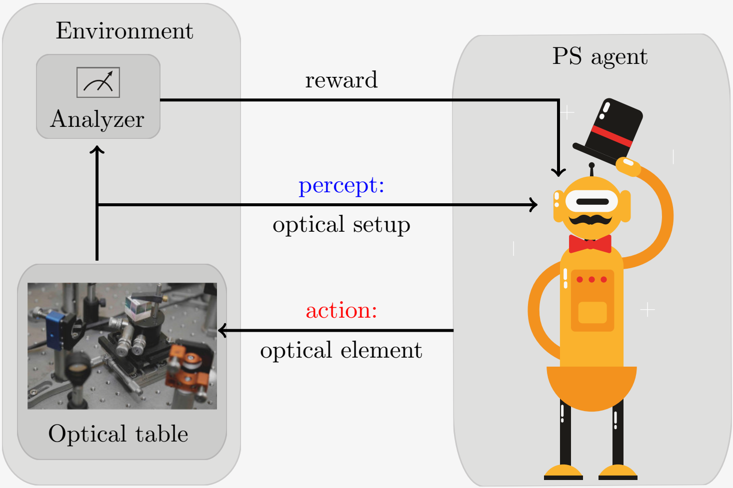
  (3) experimental techniques are discovered



Can machines genuinely contribute to scientific research?

Outlook

The described RL methodology can be applied beyond the considered example



The same can be used if at least one of these things is true:

- there exists a set of goal states, in which these states are correlated
- finding the simplest implementation is of interest, in case of a complex space of possibilities

Thank you for your attention!

Active learning machine learns to create new quantum experiments
PNAS 115, 1221(2018)

ProjectiveSimulation.org