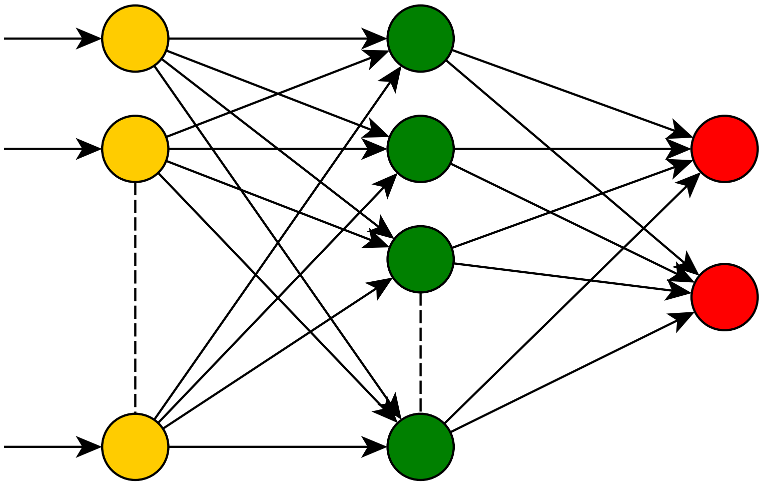
Enhancement of the S-DALINAC Control System with Machine Learning Methods*

J. Hanten#, M. Arnold, J. Birkhan, C. Caliari,
N. Pietralla, M. Steinhorst

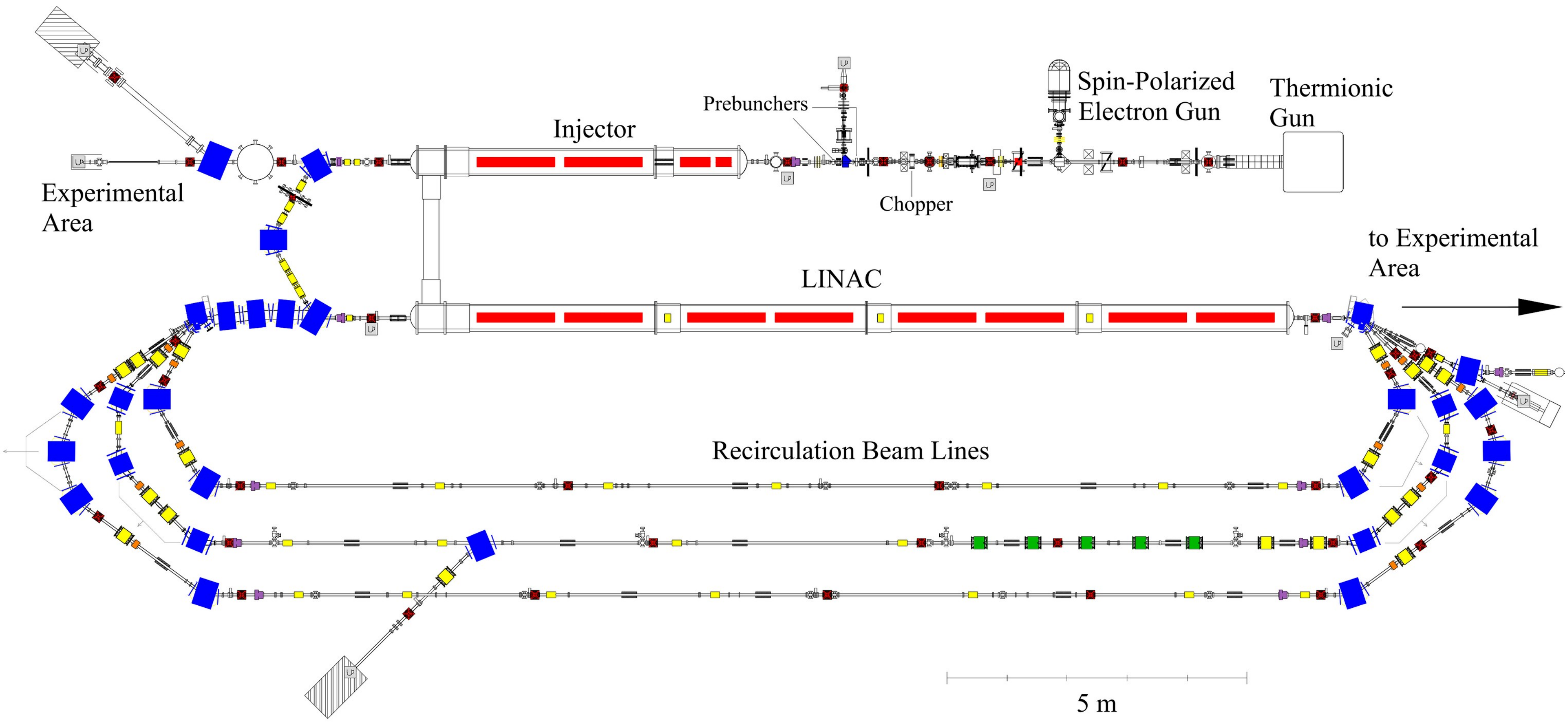
TECHNISCHE UNIVERSITÄT DARMSTADT



*Work supported by DFG through GRK 2128

S-DALINAC

TECHNISCHE UNIVERSITÄT DARMSTADT

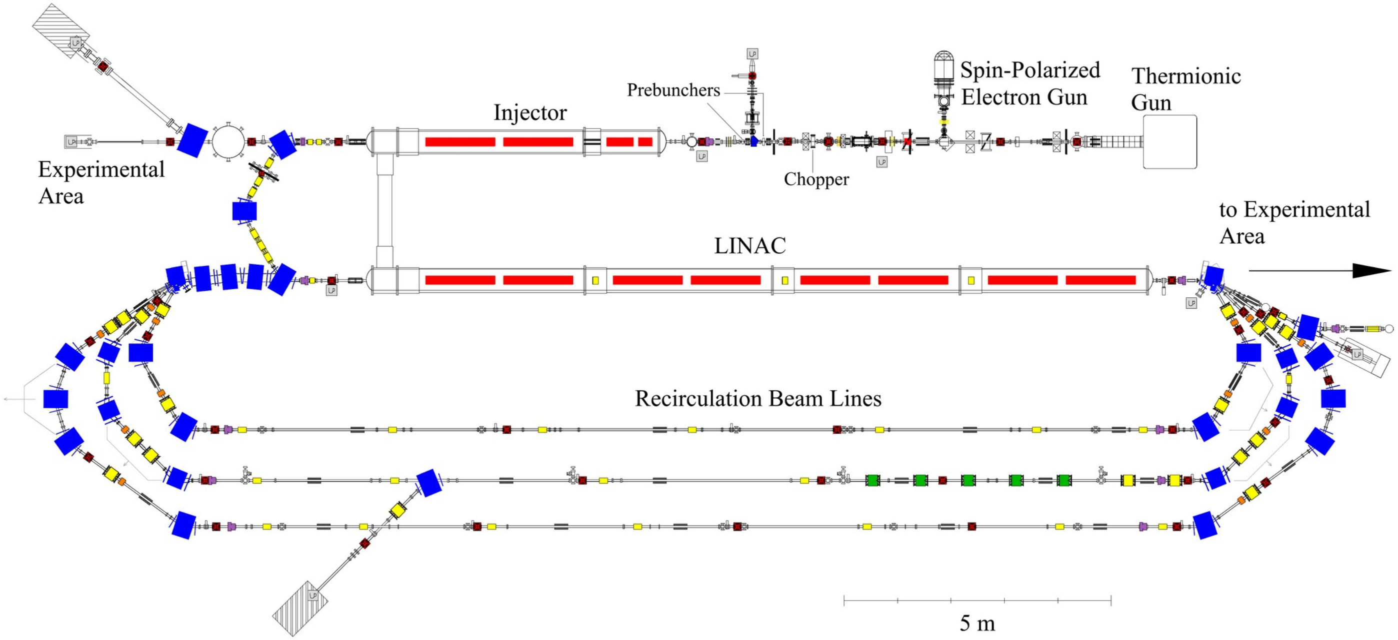


Motivation

Before every beam time:

Providing an electron beam optimized in terms of position, size, transmission and energy resolution at the experimental sites

➜ beam setup

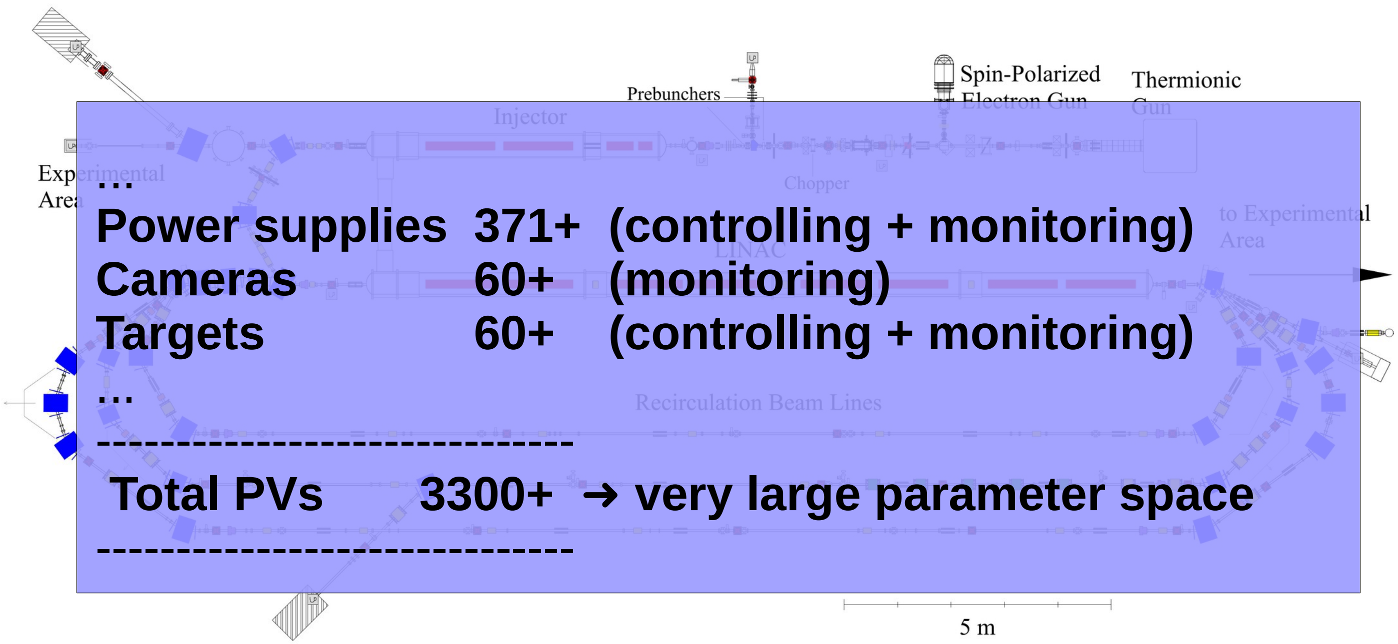


Motivation

Before every beam time:

Providing an electron beam optimized in terms of position, size, transmission and energy resolution at the experimental sites

➜ beam setup



…

Power supplies	**371+**	**(controlling + monitoring)**
Cameras	**60+**	**(monitoring)**
Targets	**60+**	**(controlling + monitoring)**

…

Total PVs 3300+ ➜ very large parameter space

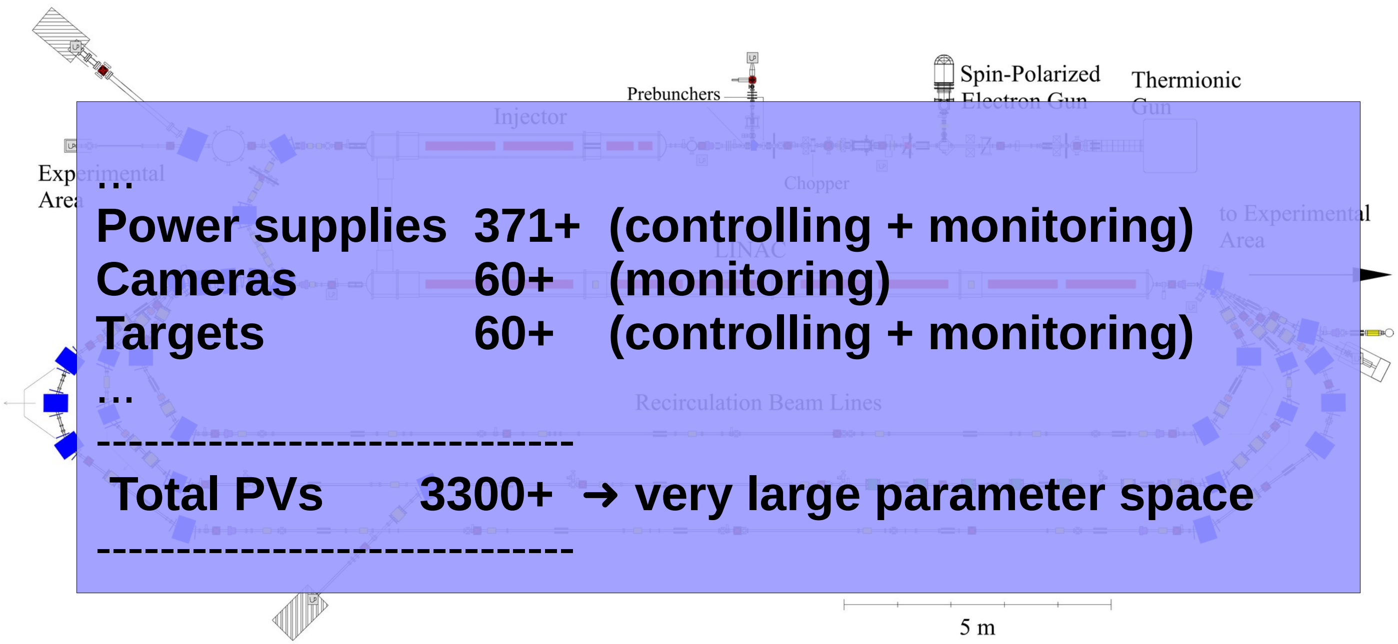
PV: Process Variable

Motivation

Before every beam time:

Providing an electron beam optimized in terms of position, size, transmission and energy resolution at the experimental sites

➜ beam setup



…

Power supplies	**371+**	**(controlling + monitoring)**
Cameras	**60+**	**(monitoring)**
Targets	**60+**	**(controlling + monitoring)**

…

Total PVs 3300+ ➜ very large parameter space

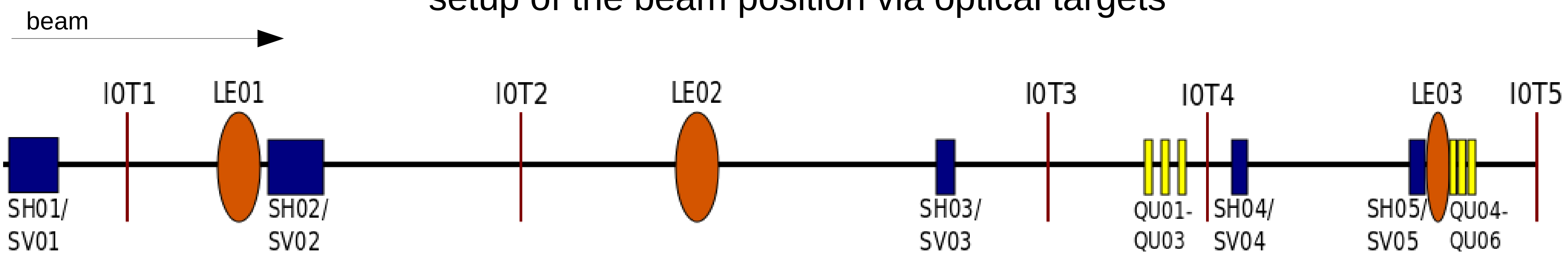
PV: Process Variable

Idea:

Use current machine learning achievements and make the beam setup a reinforcement learning task

Beam Setup

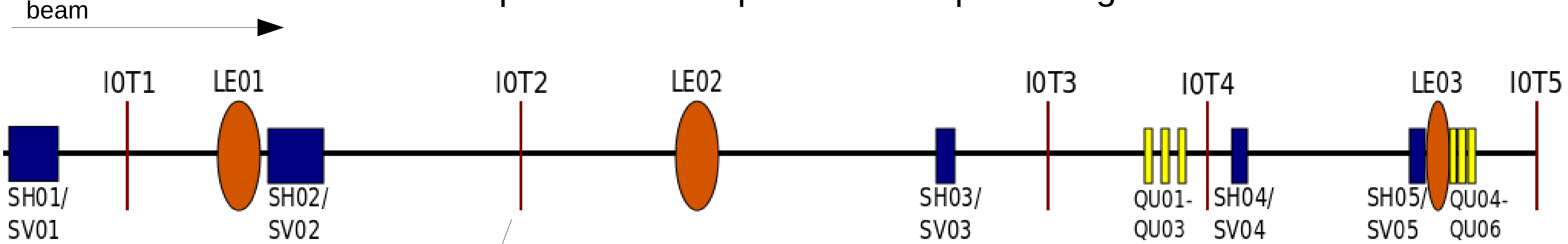
setup of the beam position via optical targets



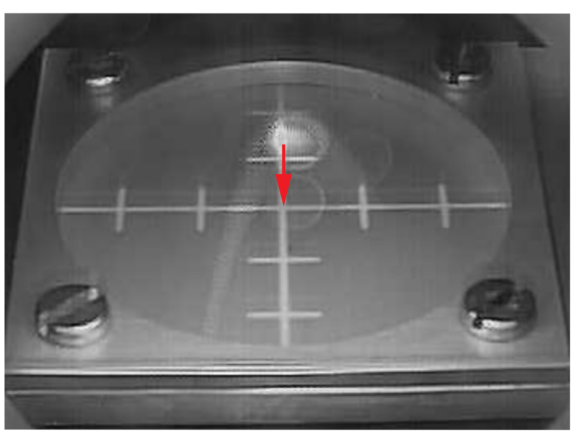
Adjustment by variation ΔI_n of currents

Beam Setup

setup of the beam position via optical targets



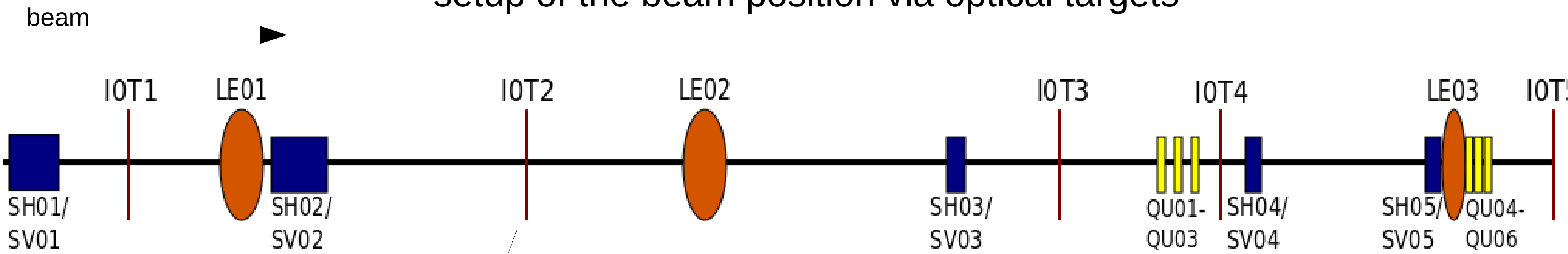
Adjustment by variation ΔI_n of currents



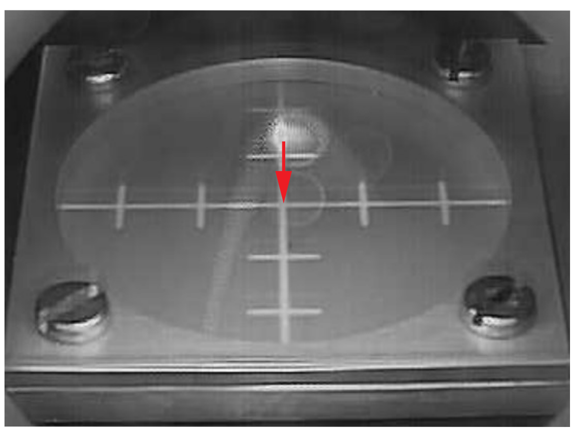
Beam Setup

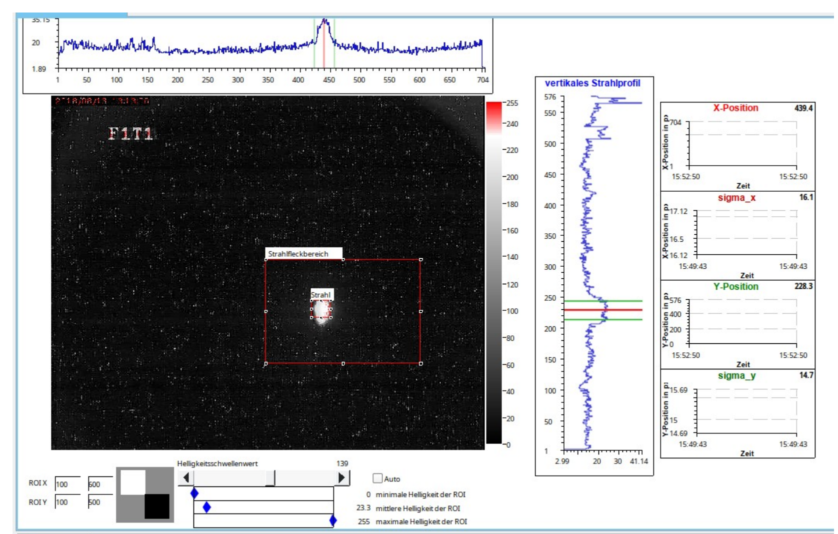
TECHNISCHE
UNIVERSITÄT
DARMSTADT

setup of the beam position via optical targets



Adjustment by variation ΔI_n of currents





Challenges

Previously optimal set points do not always provide the same beam settings and quality after restore

→ Possible loss of beam quality
e.g. 30 keV energy resolution at the spectrometer, instead of 10 keV

→ New adjustment needed ➜ loss of time
+ few days up to weeks

Challenges

Previously optimal set points do not always provide the same beam settings and quality after restore

→ Possible loss of beam quality
e.g. 30 keV energy resolution at the spectrometer, instead of 10 keV

→ New adjustment needed ➜ loss of time
+ few days up to weeks

A possible explanation for deviation in the set points

→ Temperature fluctuations in the environment

→ Resulting e.g. in slightly different properties of rf cables

→ ...

Reinforcement Learning

Software *agents* take *actions* in a *state* of the *environment* to maximize cumulative *reward*

$$R_t := \sum_{i=t}^{T} \gamma^{(i-t)} r\left(s_i, a_i\right), \;\; \gamma \in [0,1]$$

Return for time step t ➜ discounted sum of rewards r until termination

$$Q^*(s,a) := \max_\pi \mathbb{E}\left[R_t \,|\, s_t = s, a_t = a, \pi\right]$$

Optimal action-value function ➜ expected return for best policy

Reinforcement Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

Software *agents* take *actions* in a *state* of the *environment* to maximize cumulative *reward*

$$R_t := \sum_{i=t}^{T} \gamma^{(i-t)} r\left(s_i, a_i\right), \;\; \gamma \in [0,1]$$

Return for time step t ➜ discounted sum of rewards r until termination

$$Q^*(s,a) := \max_{\pi} \mathbb{E}\left[R_t \middle| s_t = s, a_t = a, \pi\right]$$

Optimal action-value function ➜ expected return for best policy

Goal of DQN-Algorithm: Estimate this function

Bellman Equation

reward for last action

Q-value for the best action in the next state (provided by NN)

$$Q_{i+1}(s,a) = \mathbb{E}\left(r + \gamma \cdot \max_{a'} Q_i(s',a') \,\Big|\, s,a\right)$$

updated Q-value for state/action pair (s, a)

discount factor

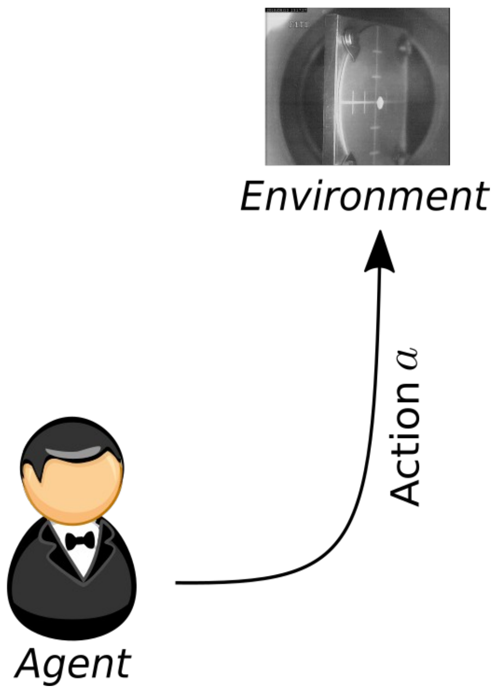
- Bellman equation allows to approximate the action value function iteratively: “learning”
- For continuous state space NN is needed as function approximator ➜ DQN
- Limitation of DQN: Action space still discrete

Deep-Q-Network-Algorithm

Reinforcement learning method

based on finding optimal action-value function $Q^*(s,a)$

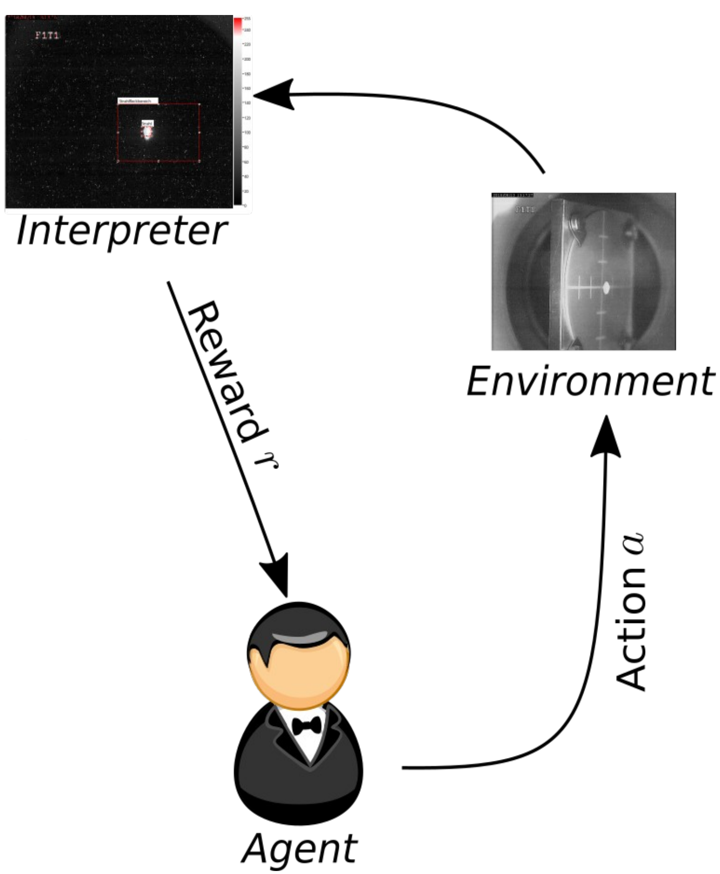
- Agent changes magnet set point (Action)



Deep-Q-Network-Algorithm

Reinforcement learning method

based on finding optimal action-value function $Q^*(s,a)$

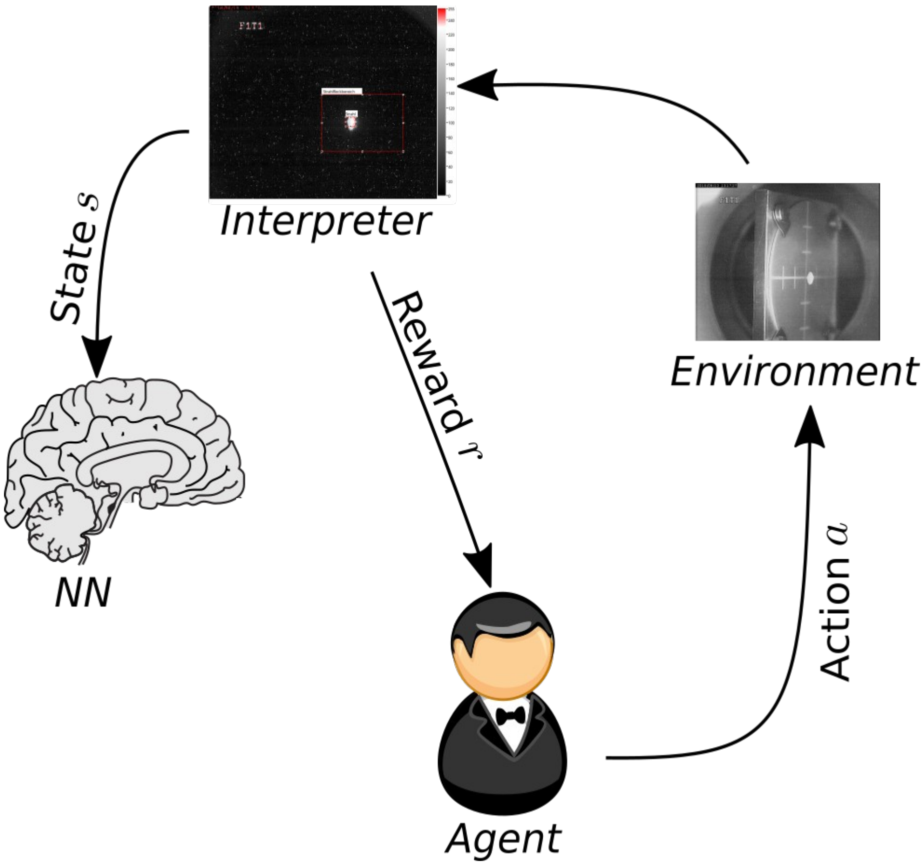


- Agent changes magnet set point (Action)
- AreaDetector gets values from target
- Reward is calculated
- State is evaluated

Deep-Q-Network-Algorithm

Reinforcement learning method

based on finding optimal action-value function $Q^*(s, a)$

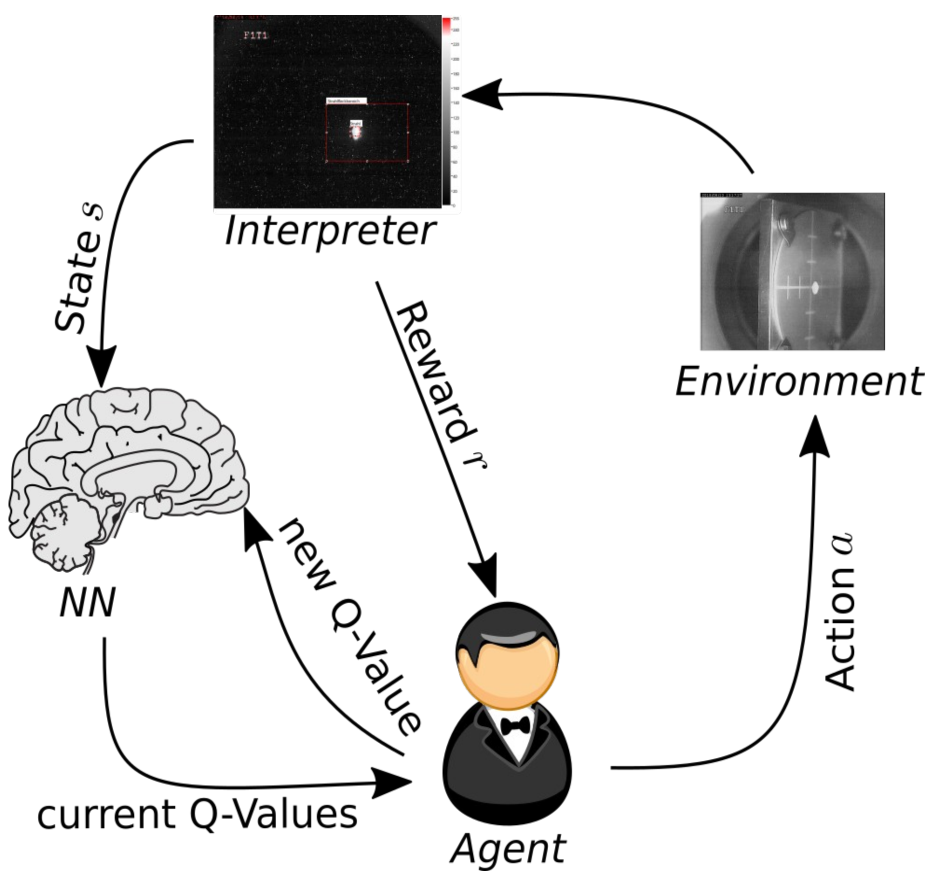


- Agent changes magnet set point (Action)
- AreaDetector gets values from target
- Reward is calculated
- State is evaluated

Deep-Q-Network-Algorithm

Reinforcement learning method

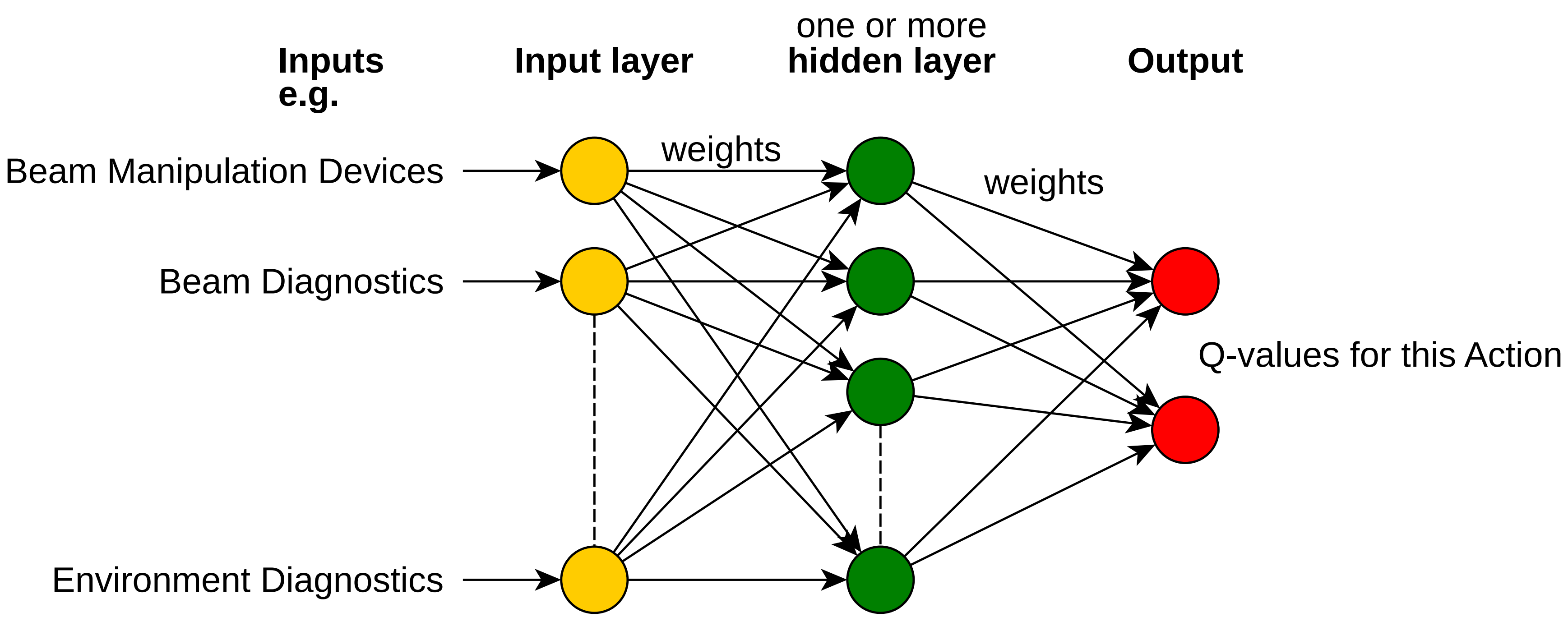
based on finding optimal action-value function $Q^*(s, a)$



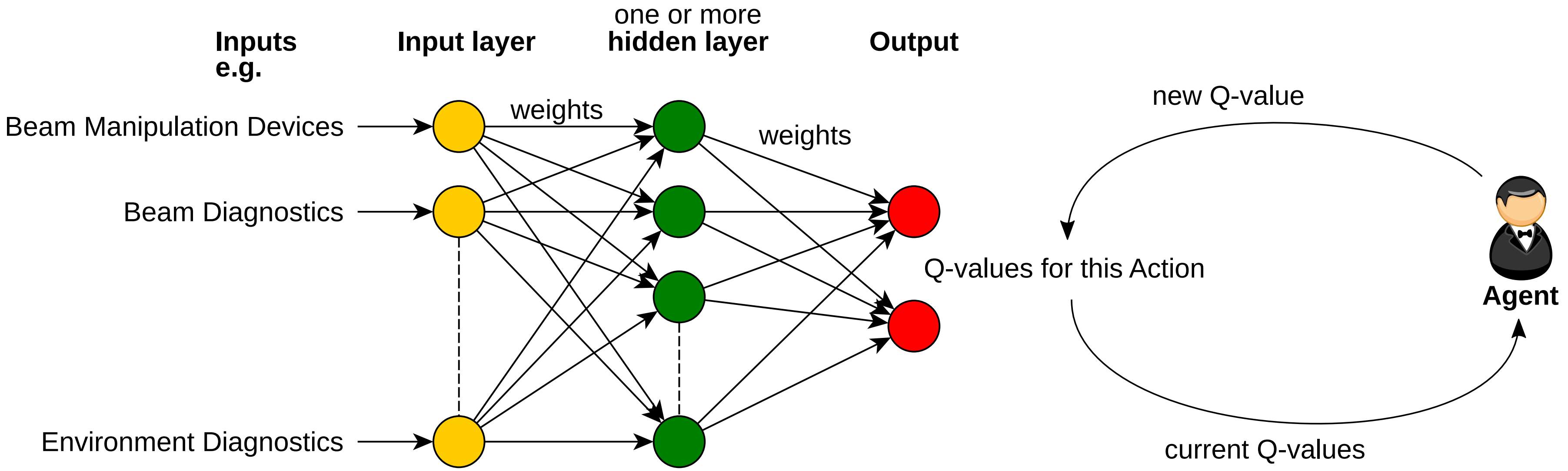
- Agent changes magnet set point (Action)
- AreaDetector gets values from target
- Reward is calculated
- State is evaluated
- NN gives next Q
- New Q for this state/action pair is calculated
- NN is trained with this value

Q-Network

- Each connection has weight w and each neuron has bias b
- Activation function $\sigma\left(\vec{w}\cdot\vec{x}+b\right)$ with activition vector $\vec{x}$



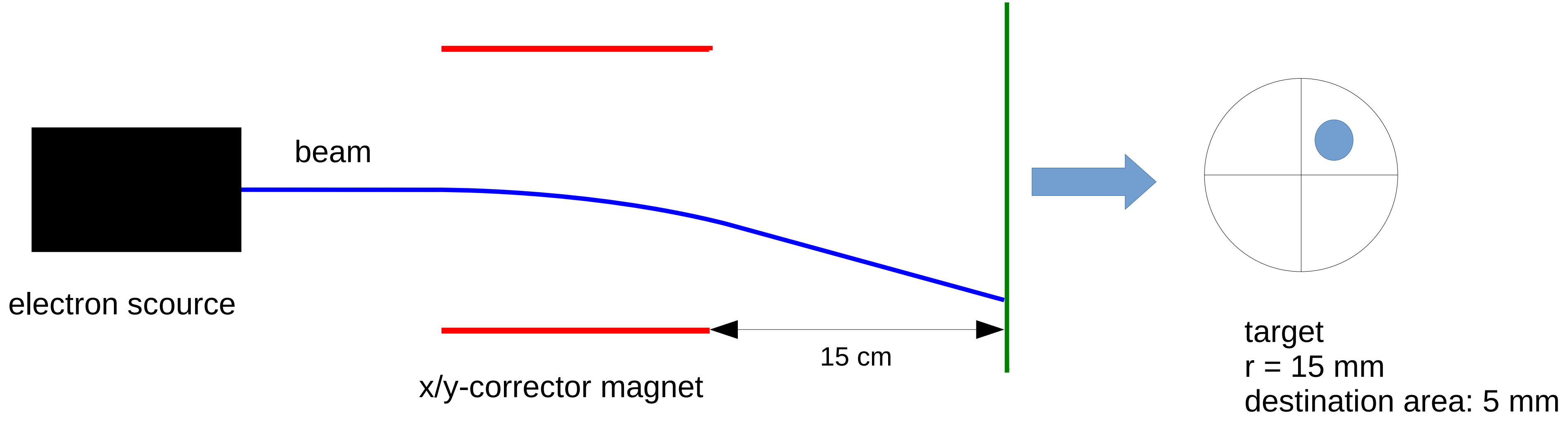
Q-Network



Inputs
e.g.
Input layer
one or more
hidden layer
Output
Beam Manipulation Devices
Beam Diagnostics
Environment Diagnostics
weights
weights
new Q-value
Q-values for this Action
Agent
current Q-values

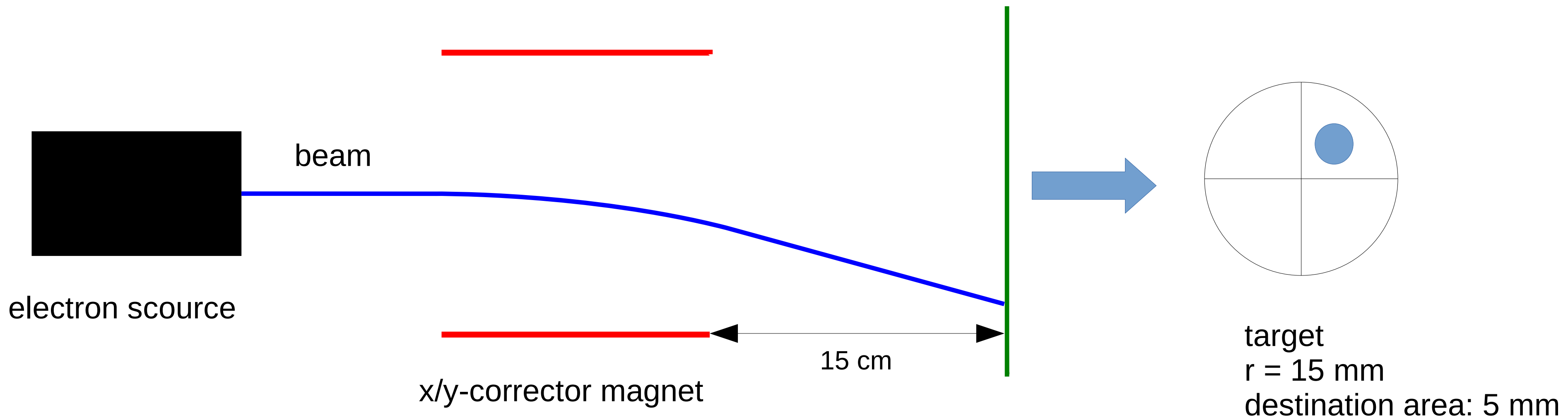
First Test of DQN-Algorithm with *elegant*

TECHNISCHE UNIVERSITÄT DARMSTADT



- Initial magnet deflection angle and destination point chosen randomly
- beam coordinates absolute and relative to destination, current magnet deflection ➜ 6 input neurons
- Reward proportional to reduction of distance to destination

First Test of DQN-Algorithm with *elegant*



Tested with three different action spaces:

$$A_4 = \{-5\,\mathrm{mrad}, 5\,\mathrm{mrad}\}^2$$
$$A_9 = \{-5\,\mathrm{mrad}, 0\,\mathrm{mrad}, 5\,\mathrm{mrad}\}^2$$
$$A_{25} = \{-10\,\mathrm{mrad}, -1\,\mathrm{mrad}, 0\,\mathrm{mrad}, 1\,\mathrm{mrad}, 10\,\mathrm{mrad}\}^2$$

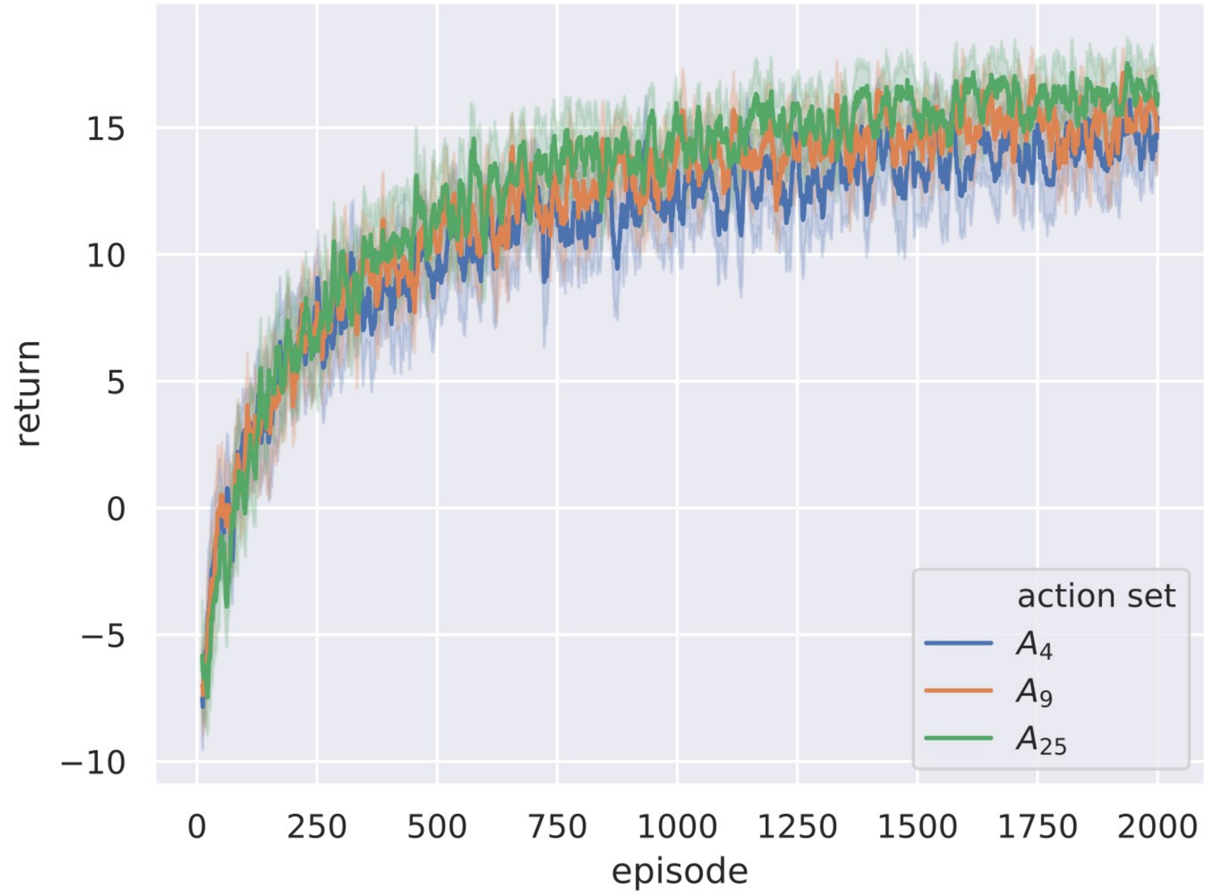
1 mrad ≙ 0.15 mm

Max 50 steps per episode

Learning Curve

TECHNISCHE UNIVERSITÄT DARMSTADT

Mean return in last 10 episodes



return = sum of rewards during episode

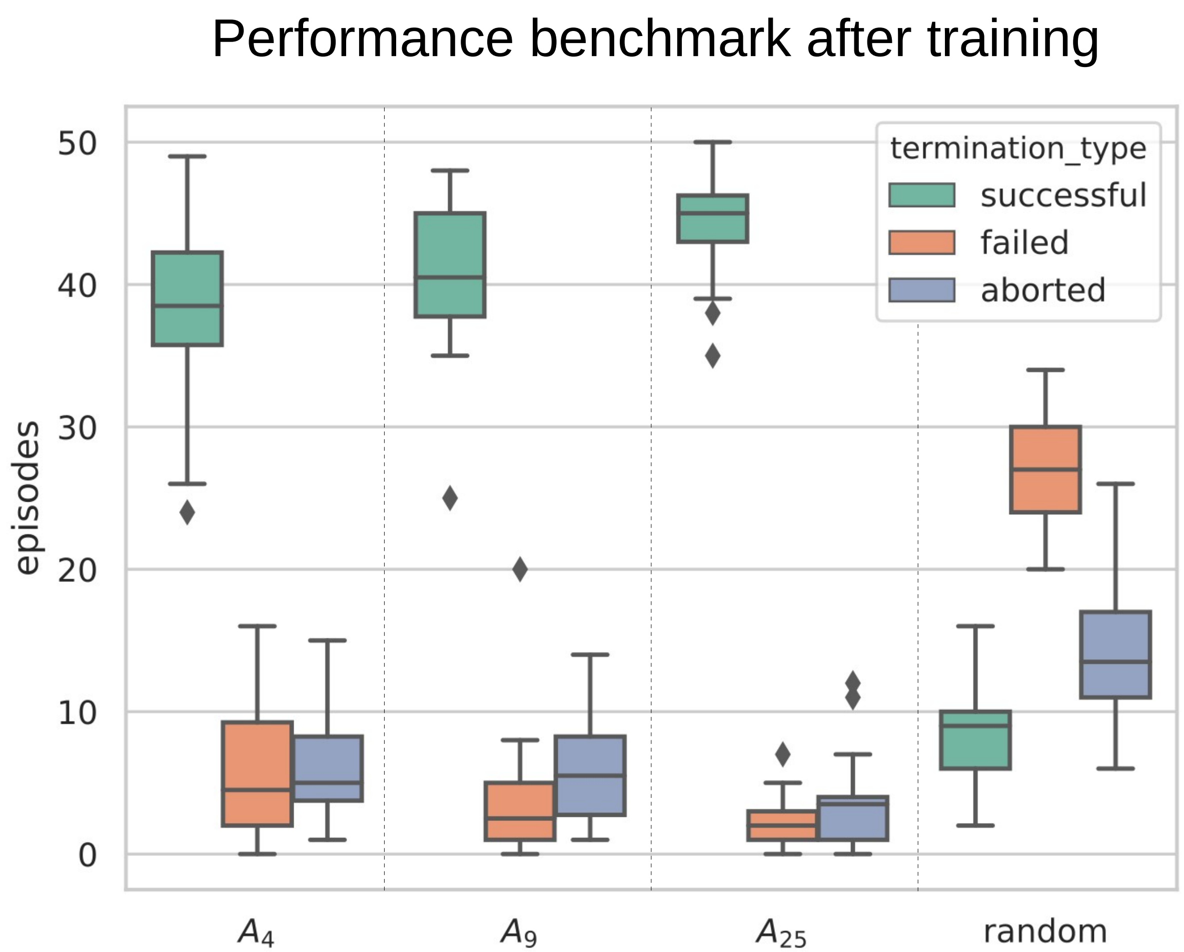
reward/approach = 1/mm

reward for reaching goal = 10

reward for leaving target = -10

$\gamma = 0.25$

Performance of Trained Networks



50 test episodes with trained network

failed: escape from target

aborted: not at destination within 50 steps

Conclusion and Outlook

TECHNISCHE UNIVERSITÄT DARMSTADT

- DQN was able to learn in a simple test case
- discretization will be problematic with higher number of elements
  ➜ Action space grows exponentially

- *Deep deterministic policy gradient* (DDPG) might be a solution
- Continuous action space possible
- Test of RL-algorithms with more complex simulation scenarios
- Test at operation

Thank you for your attention!

Goals

TECHNISCHE UNIVERSITÄT DARMSTADT

globally and reproducibly optimized settings per experiment	compensation of temporal fluctuations due to environmental influences

Autonomous beam optimization

- Reinforcement learning with artificial neural networks (NN) (e.g. Deep-Q-Networks DQN)

Correlation study of process variables using methods of data mining

- Transfer entropy
- Singular value decomposition

Goals

globally and reproducibly optimized settings per experiment | compensation of temporal fluctuations due to environmental influences

Autonomous beam optimization

- Reinforcement learning with artificial neural networks (NN) (e.g. Deep-Q-Networks DQN)

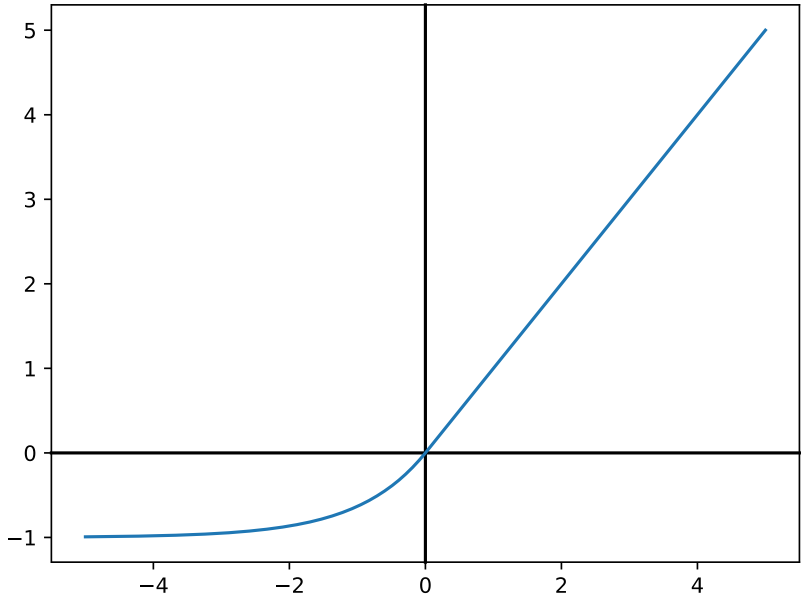
Correlation study of process variables using methods of data mining

- Transfer entropy
- Singular value decomposition

ELU vs. ReLU

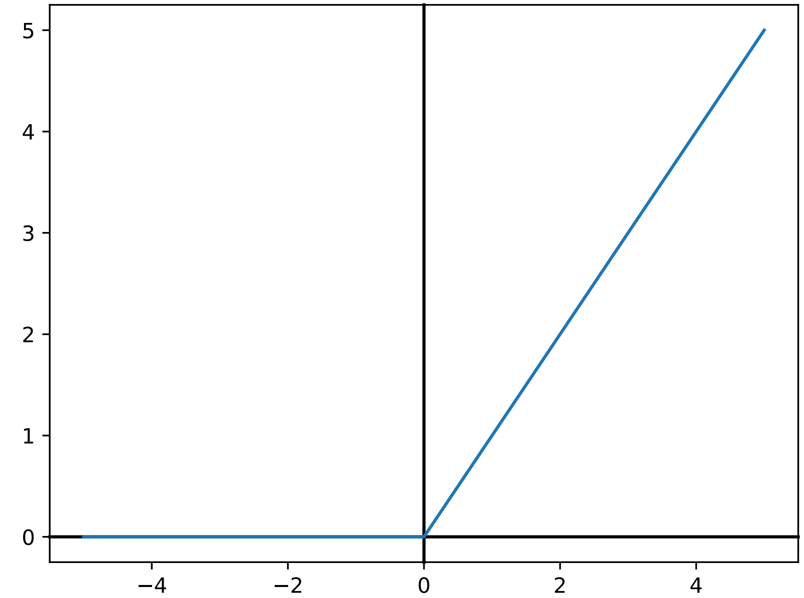
TECHNISCHE UNIVERSITÄT DARMSTADT

Exponential Linear Unit



$$\sigma(x) = \begin{cases} x & \text{if } x > 0 \\ (e^x - 1) & \text{otherwise} \end{cases}$$

Rectified Linear Unit



$$\sigma(z) = \max(0, z)$$

NN Training

TECHNISCHE UNIVERSITÄT DARMSTADT

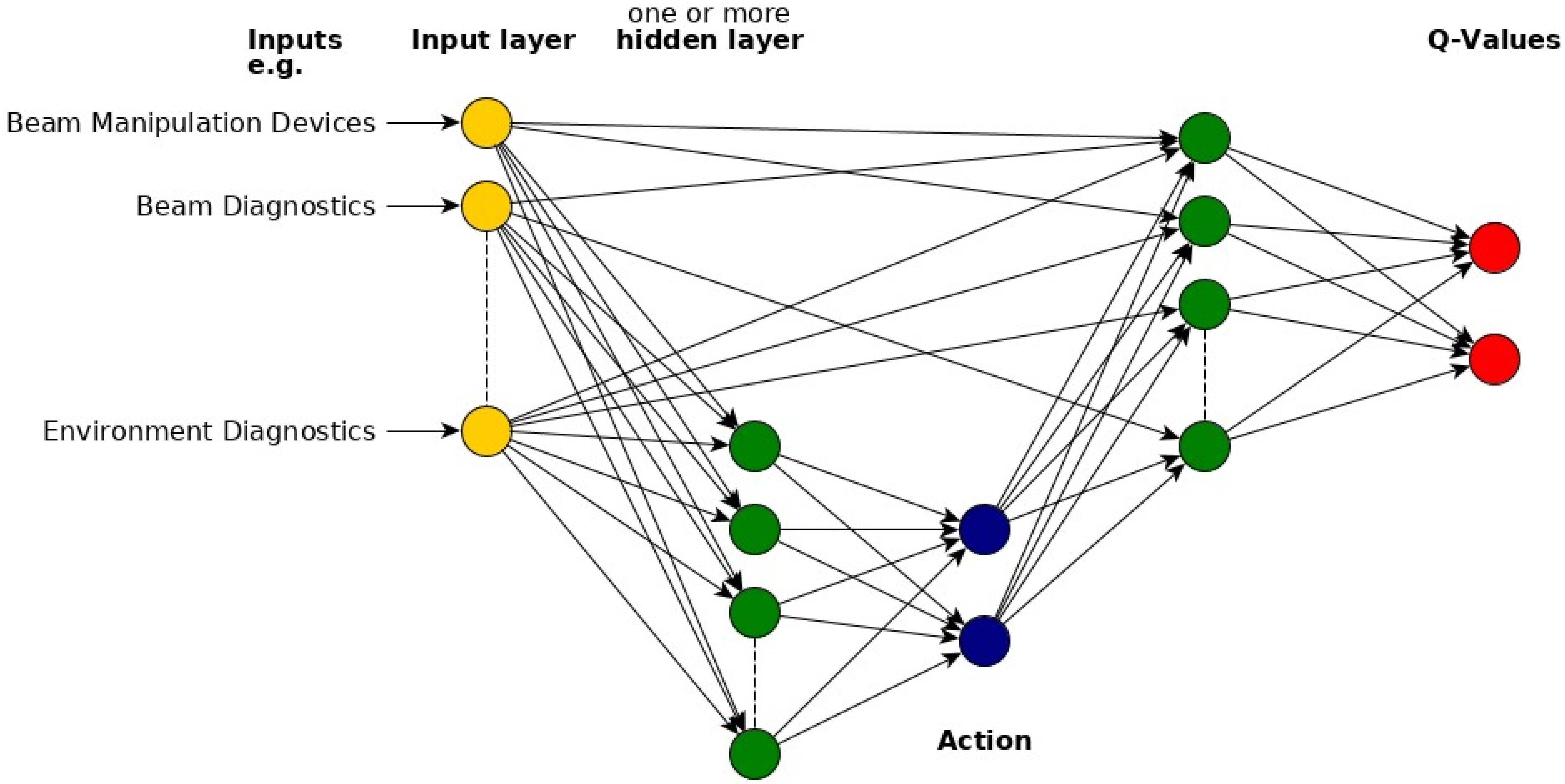
$$R = \left|r_{\text{target}} - r_{\text{before}}\right| - \left|r_{\text{target}} - r_{\text{after}}\right|$$

$$L_t\left(\theta_t\right) = \begin{cases} \frac{1}{2}\left(y_t - Q\left(s_t, a_t; \theta_i\right)\right)^2 & \text{for } \left|y_t - Q\left(s_t, a_t; \theta_i\right)\right| < 1 \\ \left|y_t - Q\left(s_t, a_t; \theta_i\right)\right| - 0.5 & \text{otherwise} \end{cases}$$

$$b_k \rightarrow b_k - \eta \frac{\partial C}{\partial b_k}$$

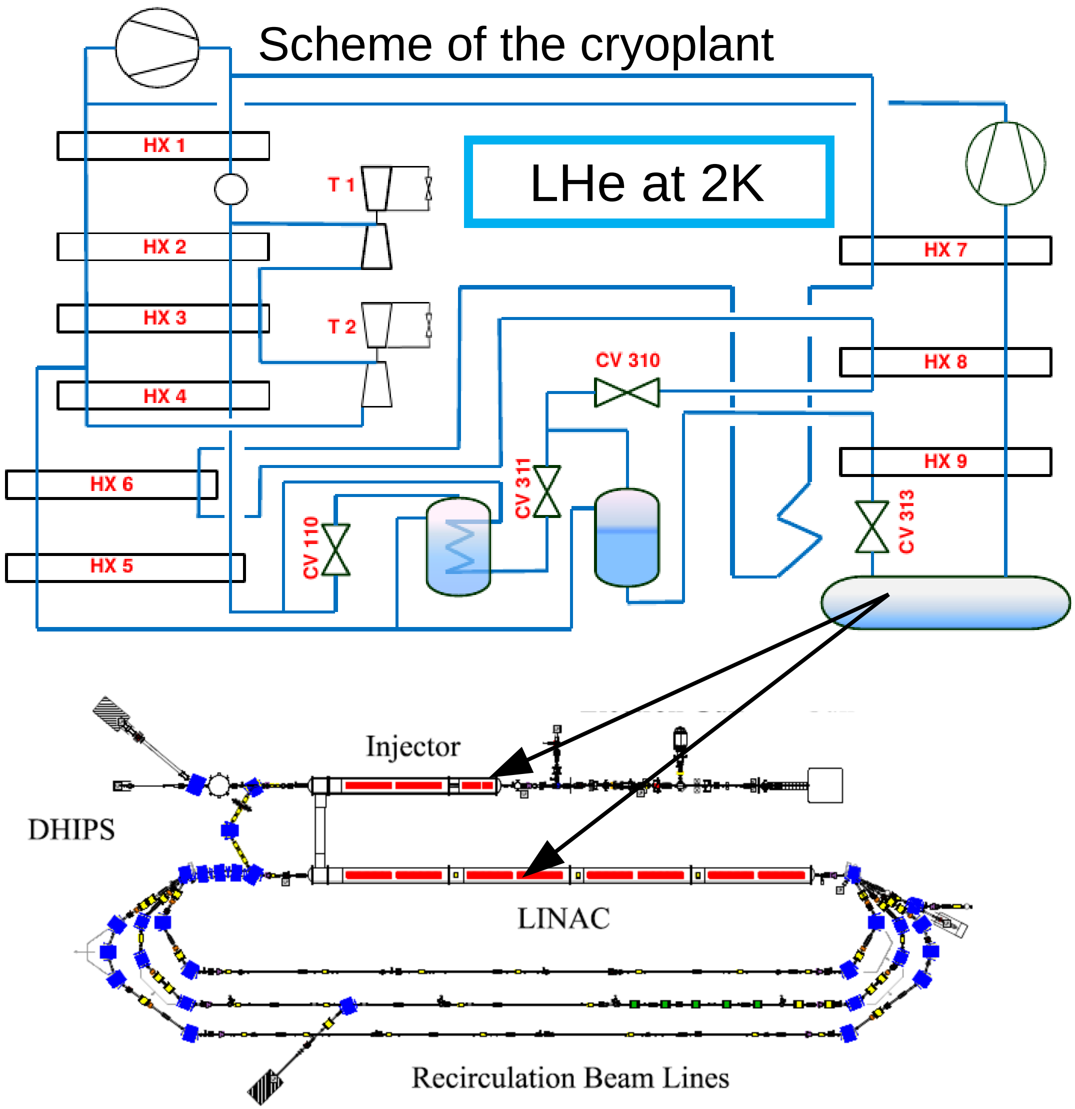
$$w_k \rightarrow w_k - \eta \frac{\partial C}{\partial w_k}$$

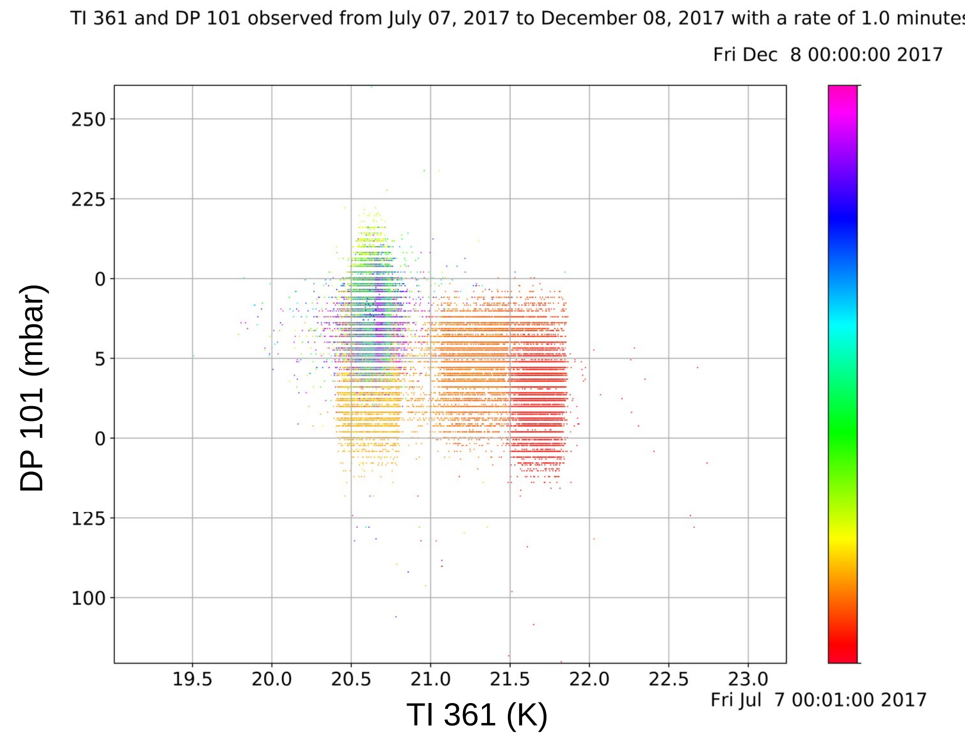
DDPG


Inputs e.g.
Input layer
one or more hidden layer
Q-Values
Beam Manipulation Devices
Beam Diagnostics
Environment Diagnostics
Action

Correlation Analysis of Process Variables (Cryoplant)

Scheme of the cryoplant





Correlation diagram of one beam time

Correlation Analysis of Process Variables (Cryoplant)