New Techniques for Operational Control and Performance Optimization @ BESSY II

L. Vera Ramírez, T. Birke, D. Engel, G. Hartmann, V. Laux, T. Mertens, M. Ries, A. Schälicke, P. Schnizer, J. Viefhaus

Helmholtz-Zentrum Berlin (HZB), Germany

Data Science and Machine Learning Workshop, ICALEPCS 2019,

06/10/2019

Motivation

At the **synchrotron radiation source BESSY II** (Helmholtz-Zentrum Berlin, HZB) both the **beamline** and the **machine groups** have started working towards setting up the infrastructure to introduce **modern analysis, optimization and automation** in order to improve the performance and the experimental setups.

In this talk we introduce some of these first tools - based among others on **ML techniques** - concerning data acquisition, simulation, machine measurement prediction and parameter tuning. More detailed results on ML- experiments will be presented at talk TUCPL01.

- The **Helmholtz Association** (the largest scientific organisation in Germany) has initiated the implementation of the **Data Management and Analysis** concept across its centers.
- Further HZB colaborations:
  - Accelerating Machine Learning for physics (AMaLea) (DESY, KIT, HZDR and HZB) - *more about this at talk TUCPL06*.
  - Advanced Computational Tools (ACT) (TU Darmstadt, HZB).
  - Findable, Accessible, Inter-operable and Re-usable Data Infrastructure for material research (FAIRmat/FAIRDI) (IRIS Berlin, HU Berlin, Max Planck Institute and HZB).

Figs: Helmholtz-Gemeinschaft Deutscher Forschungszentren - https://commons.wikimedia.org/w/index.php?curid=56032104

Table of Contents

Table of Contents

Advanced data taking and processing

Advanced data taking and processing

Advanced tools for data taking and preprocessing are currently being incorporated: Singularity, Ophyd, Bluesky, ElasticSearch...

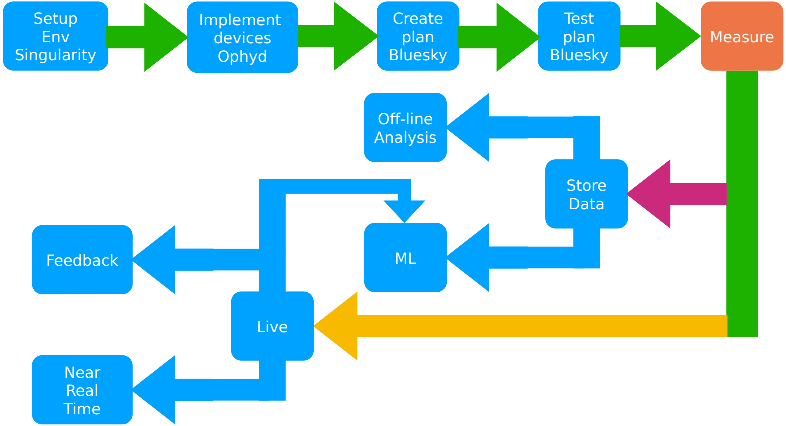


Table of Contents

ML tools @ BESSY II

Device

- Injector Ring accelerator
- Undulator
- Beamline
- Experiment

Data

- Archive, Diagnostic
- Simulations
- Online optimization

- Diagnostic
- Raytracing
- Scans/online data

- Demands
- Simulations
- Beamtimes

Methods

- SVR-RFF
- DNN
- Deep-RL-Control
- RNN, LSTM

- Autoencoder
- CNN, MLP, GBoost
- Dataloader
- Tensor product
- kNN, auto-diff.

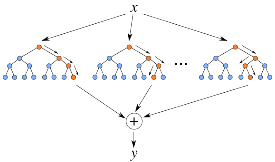
- Reasonable random generator

Agent

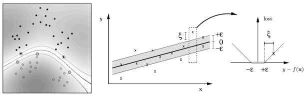
- Operator
- Beamline scientist
- (Random-) User

Table of Contents

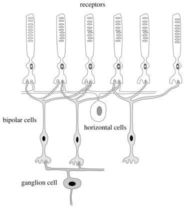
- **Ensemble methods**: Random Forests, Extremely Randomized Trees... ([Bre01], [GEW06]). For regression, **MSE** as loss $\rightarrow$ variance as impurity measure. **Self-explaining**: allow individual analysis of each variable's behavior.



- **Support Vector Regression** [Smo98] with **Random Fourier Features** ([RR08]). SVR extends traditional SVM (for classification) via Vapnik's ϵ-insensitive loss function ([Vap95]).

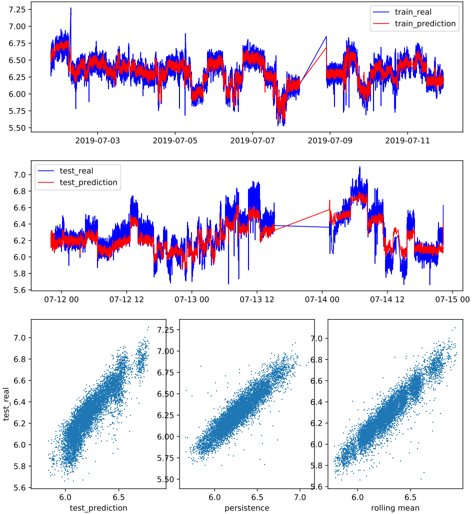


- **Neural Networks** (e.g. see [Roj96]). Feed-forward NNs for regression (i.e. MSE as loss function).



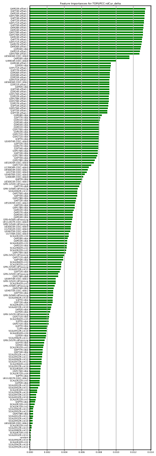
Figs. from https://dsc-spidal.github.io/harp/docs/examples/rf/, [SS03], [Roj96].

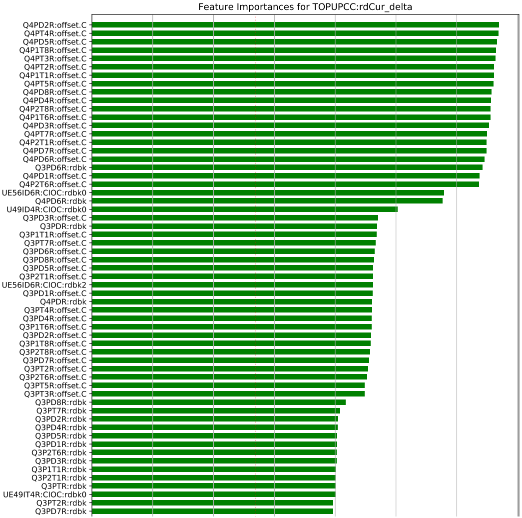
HZB Helmholtz Zentrum Berlin



- Prediction model w.r.t. 185 machine readbacks:
  - Gap and shift of insertion devices
  - Quadrupoles power supplies and offsets,
  - Local beam loss fractions
  - ...
- We **omit** previous lifetime observations to avoid an excessive reliance on them, force the model to identify further patterns and reuse the gained information.
- E.g. SVR with Random Fourier Features forecasts the trends in **ca. 3 days of completely unseen data** ($R^2 = 0.724506 \pm 0.015291$).

HZB Helmholtz Zentrum Berlin





- Analysis with `RandomForest`.
- Evenly distributed feature importances - **quadrupoles** (offsets) and **insertion devices** stand out.

Table of Contents

- **Deep Deterministic Policy Gradient** ([LHP+16]): Actor-critic Reinforcement Learning algorithm for continuous environments.
- Off-policy data and the Bellman equation used to learn the Q-function.
- Q-function used to learn the policy.
- *Approximated* with NNs.

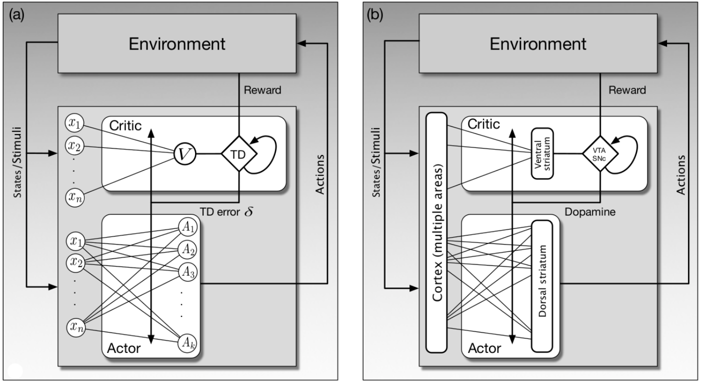
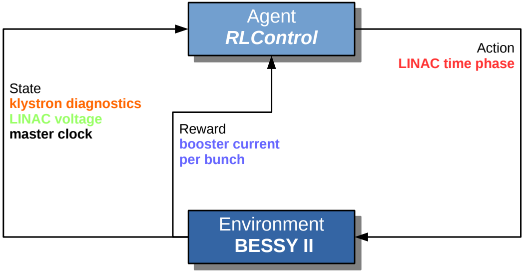
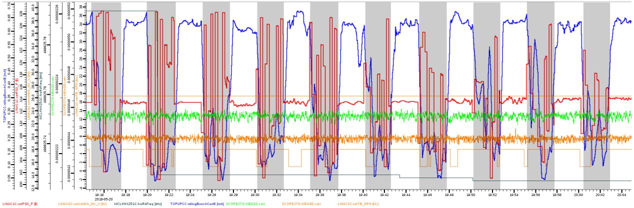
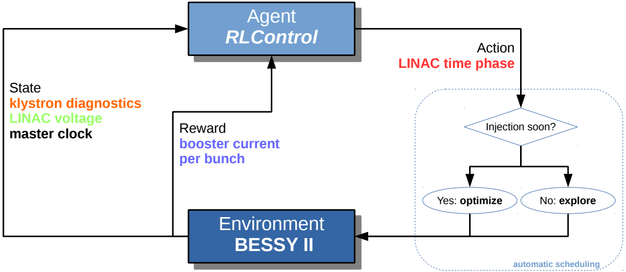


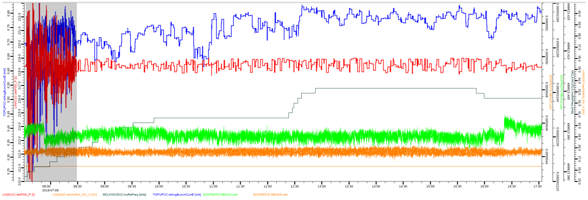
Figure: From [SB18]





- First use case: **booster current optimization.**
- Nowadays manual parameter tuning is required - low booster current after long interruptions.
- Test with periods of **pure exploration** (shaded) and optimization **learning with no exploration.** Some issues:
  - Non-optimal exploration → Parameter Space Noise ([PHD+17])
  - Long training time and normalization problems → pretraining with historical data (inspired by [ZM18]).





- **Automatic scheduling**:
  - Pure exploration is scheduled in the meantime between injections to avoid disturbing user activity.
  - Optimization activated shortly before each injection.
- Successful 8.5 hours long test **during user operation.**
- More detailed results and further use cases (**injection efficiency optimization**) will be presented during this conference - talk TUCPL01.

Table of Contents

ML tools @ BESSY II: Building the digital twin

The *accelerator/source* side: OCELOT surrogate models for RL ([AGTZ14]). Some tests with toy-examples (emittance, orbit-correction...) in small lattices already carried out. Challenge: *export* a RL-agent trained with the virtual lattice to the real accelerator.

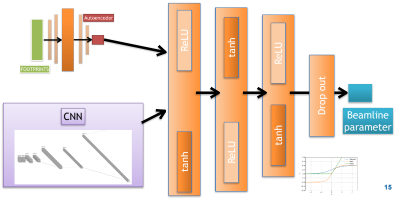
The *beamline/x-ray* side: general initiative to take advantage of ML methods also for the **retrieval of scientific data** from the measurements. Close coordination of activities developing toolsets for the accelerator and the beamlines ([M$^+$19]). Focus on improving the prerequisites for common ML methodologies.

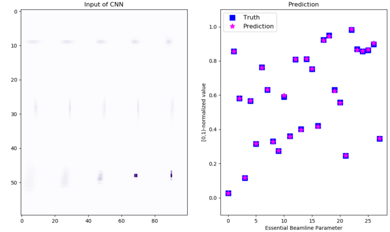
Table of Contents

ML tools @ BESSY II: Understanding beamline performance

HZB Helmholtz Zentrum Berlin

- Today's beamlines have **couple of hundred parameters** - impossible to map with traditional simulation tools.
- Approach: **ML methods** to learn raytracing as well as beamline parameter prediction from photon diagnostics.
- Test - **inversion** of beamline raytracing:
  - Diagnostic: **photon footprint screens** at three different positions in the beamline (intermediate focus, exit slit and experimental focus).
  - These are used from a neural network to **predict 28 essential beamline parameters** (100 parameters varied in the training).





Current development by Dr. Gregor Hartmann - further info: talk MOCPL02 ([M^{+}19]).

Table of Contents

Next steps

- Measurement prediction:
  - Build models for further target variables: purity...
  - Classification approach.
  - Surrogate models.
- *RLControl*:
  - Further tests, investigation and use cases (injection efficiency with more state and action variables, orbit correction with OCELOT pretraining...).
  - Integration of advanced data collection tools such as Bluesky.
  - User interfaces.
- Beamline raytracing:
  - Benchmark and optimize beamline designs.
  - Determine the current state of a beamline.
  - Include more ”building blocks” (e-beam, undulator, experiment, spectrometers).
  - Determine at which position photon detection screens are best placed.

Table of Contents

Advanced data taking and processing

ML tools @ BESSY II

- Measurement prediction framework
- RLControl - Parameter tuning with Deep RL
- Building the digital twin
- Understanding beamline performance

Next steps

References

Backup

HZB Helmholtz Zentrum Berlin

[AGTZ14] Ilya Agapov, Gianluca Geloni, Sergey Tomin, and Igor Zagorodnov. OCELOT: A software framework for synchrotron light source and FEL studies. *Nuclear instruments & methods in physics research / A*, 768:151 – 156, 2014. (c) Elsevier B.V.

[Bre01] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[GEW06] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, Apr 2006.

[IF18] Masaaki Imaizumi and Kenji Fukumizu. Deep neural networks learn non-smooth functions effectively. 2018.

[LHP+16] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[LTZ08] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM ’08, pages 413–422, Washington, DC, USA, 2008. IEEE Computer Society.

Bibliography II

[LWSG13] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, pages 431–439, USA, 2013. Curran Associates Inc.

[M+19] Roland Müller et al. Modernization of experimental data taking at BESSY II. ICALEPCS, 2019.

[Pea01] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[PHD+17] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *CoRR*, abs/1706.01905, 2017.

[Roj96] Raul Rojas. *Neural networks : a systematic introduction*. Springer-Verlag, Berlin ;;New York, 1996.

[RR08] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc., 2008.

[RT17] David Rolnick and Max Tegmark. The power of deeper networks for expressing natural functions. *CoRR*, abs/1705.05502, 2017.

[SB18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[SLH+14] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 387–395, Bejing, China, 22–24 Jun 2014. PMLR.

[Smo98] Alexander Johannes Smola. Learning with kernels, 1998.

[SS03] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. Technical report, STATISTICS AND COMPUTING, 2003.

[Vap95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

[ZM18] Xiaoqin Zhang and Huimin Ma. Pretraining deep actor-critic reinforcement learning algorithms with expert demonstrations. *CoRR*, abs/1801.10459, 2018.

- **Ensemble methods**: Random Forests, Extremely Randomized Trees... ([Bre01], [GEW06])
- Based on different randomization techniques:
  - Bootstraping
  - Random variable subsets per node
  - Random threshold selection per node
- For regression, MSE as loss $\rightarrow$ variance as impurity measure.
- Self-explaining: allow individual analysis of each variable's behaviour.
- Can be used for several tasks (outlier detection, feature selection...).

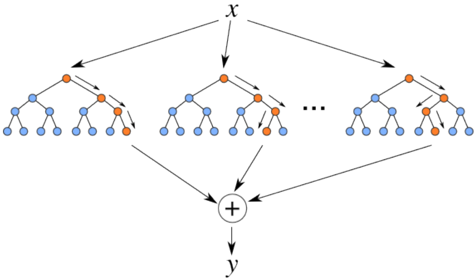


Figure: From https://dsc-spidal.github.io/harp/docs/examples/rf/

- **Support Vector Regression**: Linear and RBF Kernel, Random Fourier Features... ([Smo98], [RR08])
- SVR extends traditional SVM (conceived for classification) via Vapnik's ϵ-insensitive loss function ([Vap95])

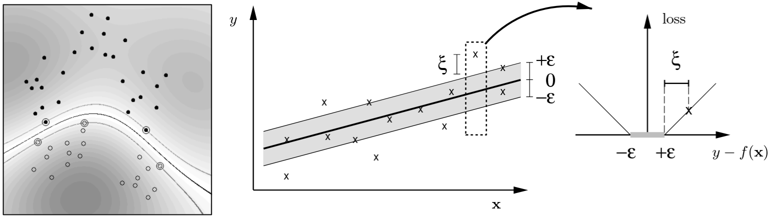


Figure: From [SS03]

- **Neural Networks** (e.g. introduction in [Roj96])
- (Deep) Feed-forward NNs for regression (i.e. MSE as loss function).
- Flexible and powerful, but with a huge meta-parameter space, including network architecture.
- Some recent works investigate this problematic from a theoretical point of view (e.g. [RT17], [IF18]) - specially interesting for our regression scenario.

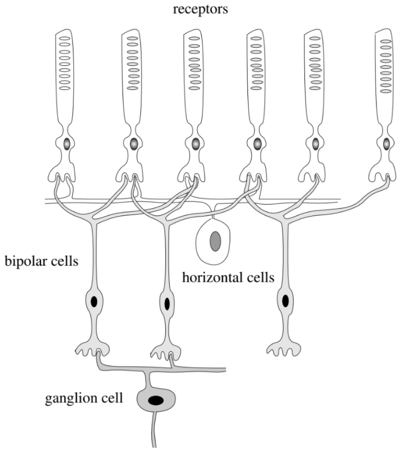


Figure: From [Roj96]

- **Isolation forest** ([LTZ08])
- Classification with an unsupervised method.
- Builds an ensemble of random trees for a given data set where each point gets a single leaf.
- Anomalies are those instances which have *short* average path lengths on the trees.

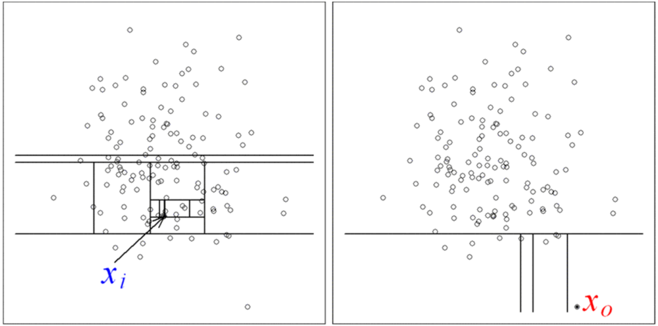
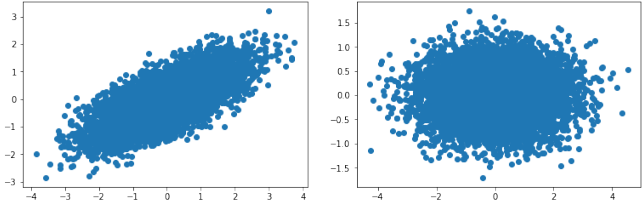


Figure: [LTZ08]

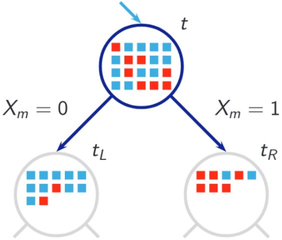
- Traditional **PCA** ([Pea01] (!!)): orthogonal transformation that creates linearly uncorrelated variables (principal components):

$$T = WX \text{ with } w_k := \underset{|w|=1}{\arg\max} \sum_i \langle [x_i - \sum_{c=1}^{k-1} \langle x_i, w_c \rangle], w \rangle^2$$



- The distribution of the transformation coefficients $\{w_i\}$ reveals the inner variable structure of the data.

- Filtering via **feature importance** with randomized forests ([LWSG13]) - based on internal impurity decrease per feature:



$$i(t) = 0.97$$

$$i(t_L) = 0.65$$

$$i(t_R) = 0.81$$

$$\Delta i(t) = i(t) - \frac{12}{20} i(t_L) - \frac{8}{20} i(t_R)$$

$$= 0.25$$

Figure: From [LWSG13] (poster)

- Theoretically conceived for totally randomized forests, can be approximated with RandomForests and/or ExtraTrees restricting the number of admitted features per node.

HZB Helmholtz Zentrum Berlin

- **Bellman equation** with a deterministic target policy μ_θ:

$$Q^\mu(s_t, a_t) = \mathbb{E}_{s_{t+1}\sim E}\left[r(s_t, a_t) + \gamma Q^\mu(s_{t+1}, \mu(s_t))\right]$$

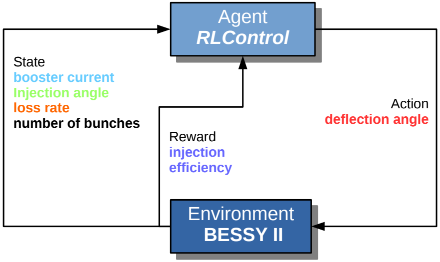
- **Critic update** (parametrized approximation Q_ϕ) through SGD with loss:

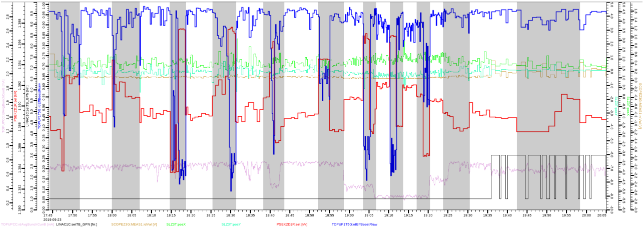
$$L(\phi) = \mathbb{E}_{s_t\sim\rho^\beta, a_t, r_t, s_{t+1}\sim E}\left[\Big(Q_\phi(s_t, a_t) - (r_t + \gamma Q_{\tilde{\phi}}(s_{t+1}, \mu_{\tilde{\theta}}(s_t))\Big)^2\right]$$

- **Actor update** - (off-policy) Deterministic Policy Gradient Theorem ([SLH⁺14]): for a performance objective $J_\beta(\mu_\theta) = \mathbb{E}_{s\sim\rho^\beta}\left[Q^{\mu_\theta}(s, \mu_\theta(s))\right]$,

$$\nabla_\theta J_\beta(\mu_\theta) \approx \mathbb{E}_{s\sim\rho^\beta}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^{\mu_\theta}(s, a)\big|_{a=\mu_\theta(s)}\right]$$

- Implementation tricks: delayed target networks ($Q_{\tilde{\phi}}$, $\mu_{\tilde{\theta}}$), replay buffer.





- Further experiments with *RLControl*: **injection efficiency optimization** (work in progress).

- Beamline raytracing is a powerful tool to understand X-ray-beam propagation and for optimizing beam properties for the experimental requirements.
- The amount of parameters is impossible to map with traditional simulation tools.