**Cheetah** provides **orders of magnitude faster** and **differentiable** beam dynamics simulations to **speed up reinforcement learning** and support applications like gradient-based optimisation.

## Start using Cheetah today!

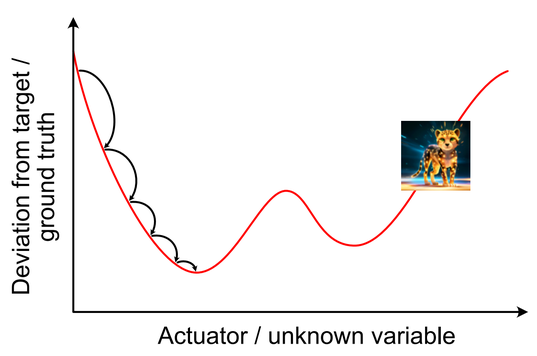
Scan for repository

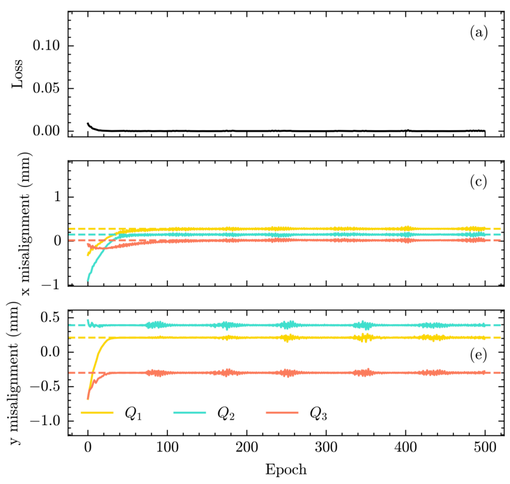
Discord


```
pip install cheetah-accelerator
```

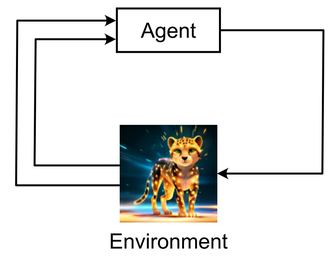
```
# Load initial beam distribution from ASTRA tracking
incoming = ParticleBeam.from_astra("beam_in.ini")

# Create a FODO lattice
segment = Segment(
  [
    Drift(length=torch.tensor(0.2)),
    Quadrupole(length=torch.tensor(0.2), name="Q1"),
    Drift(length=torch.tensor(0.4)),
    Quadrupole(length=torch.tensor(0.2), name="Q2"),
    Drift(length=torch.tensor(0.2)),
  ]
)

# Change the magnet strengths
segment.Q1.k1 = torch.tensor(10.0)
segment.Q2.k1 = torch.tensor(-9.0)

# Tracking through the segment
outgoing = segment.track(incoming)
```

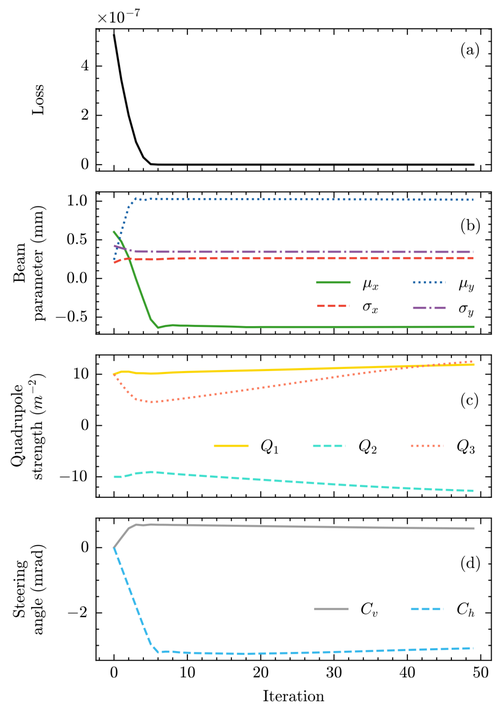
## Gradient-based diagnostics




## RLO tuning example




# Cheetah: A High-Speed Differentiable Beam Dynamics Simulator for Machine Learning Applications.

Jan Kaiser*, Chenran Xu, Annika Eichler, Andrea Santamaria Garcia

### Motivation

- Machine learning has emerged as a powerful solution to the modern challenges in accelerator physics.
- Limited availability of beam time, the computational cost of simulations, and the high-dimensionality of optimisation problems pose significant challenges in generating the required data for training state-of-the-art machine learning models.

### Beam dynamics simulations in PyTorch

- We introduce Cheetah, a beam dynamics simulation Python package written in PyTorch.
- Cheetah is up to 7 orders of magnitude faster than other simulation codes, making use of automatic and opt-in optimisations as well as PyTorch's built-in GPU acceleration.
- Being based on PyTorch, all models built in Cheetah support automatic differentiation, making Cheetah differentiable.

### Example applications

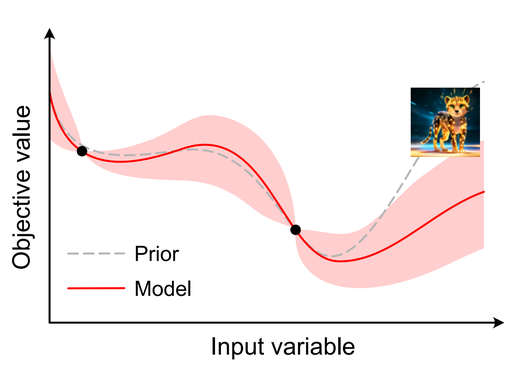
- Two learning-based optimisation methods have found particular interest in the accelerator community:
  - A **reinforcement learning-trained optimisation** policy has been trained for 6 million interactions in feasible time using Cheetah and was successfully transferred zero-shot to the real ARES accelerator.
  - **Gradient-based actuator tuning** on a Cheetah model can be used to tune the magnets at ARES to achieve desired transverse beam parameters.
  - **Gradient-based system identification** can be used to build virtual diagnostics. At ARES for example, we can utilise Cheetah to identify quadrupole misalignments zero-shot from parasitic measurements.
  - The built-in differentiation features of Cheetah, allow Cheetah models to be used as **physics-based priors for Bayesian optimisation**.
  - Cheetah's implementation based on PyTorch allows it to seamlessly integrate with **modular neural network surrogate models**. This allows, for example, for **fast computation of collective effects**. Moreover, this integration preserves gradients.

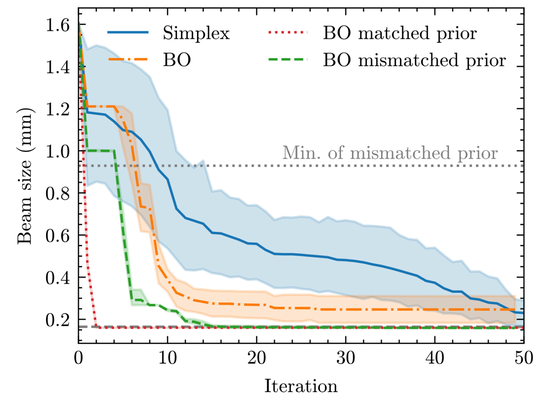
### Start using Cheetah yourself!

- Cheetah is available on ***GitHub*** (https://github.com/desy-ml/cheetah) and can be installed through `pip install cheetah-accelerator`.

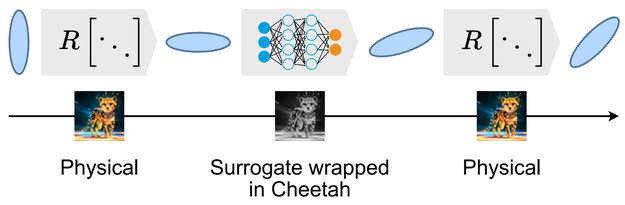
***Corresponding author:** jan.kaiser@desy.de

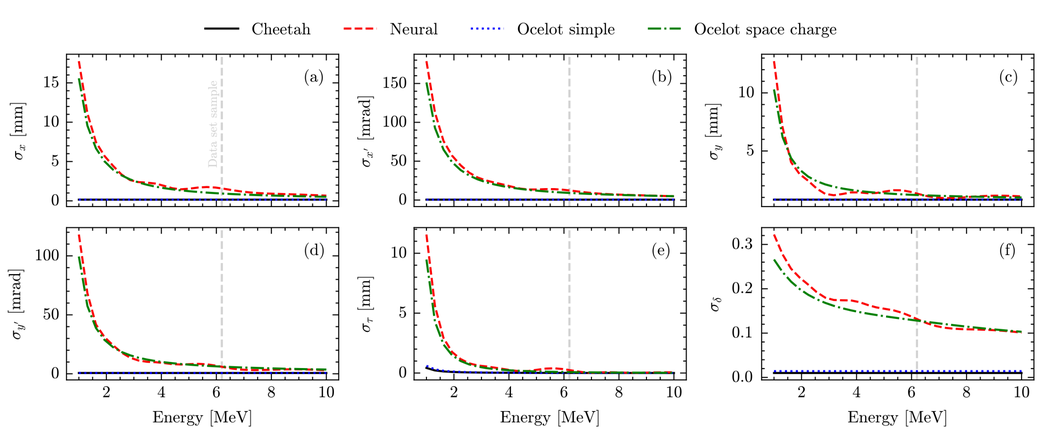
## BO prior example




## Neural network space charge surrogate example




Scan me to download the full paper!

**Deutsches Elektronen-Synchrotron DESY**
**Research Centres of the Helmholtz Association**

UNIVERSITY OF LIVERPOOL · The Cockcroft Institute of Accelerator Science and Technology · BERKELEY LAB · DESY. · TUHH Hamburg University of Technology · THE UNIVERSITY OF CHICAGO · SLAC · KIT Karlsruher Institut für Technologie