Cheetah – A High-speed Differentiable Beam Dynamics Simulation for Machine Learning Applications

MT ARD ST3 Meeting
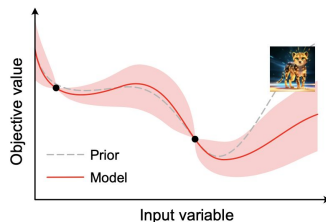
Jan Kaiser, Chenran Xu, Annika Eichler and Andrea Santamaria Garcia
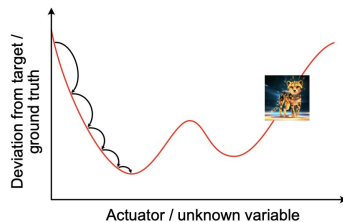Darmstadt, 5 July 2024

Cheetah

Differentiable Beam Dynamics Python Package

- **What is Cheetah?** 🐆
  - An easy-to-use Python package for fast and differentiable beam dynamics simulations.
  - Specifically designed for machine learning applications.
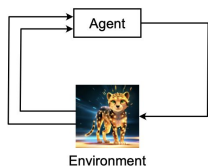
- **What can you do with it?** 🚀
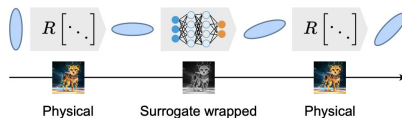

(a) Bayesian optimisation prior

Objective value / Input variable
- - - Prior
— Model

(b) Gradient-based tuning / system identification

Deviation from target / ground truth / Actuator / unknown variable

(c) Reinforcement learning

Agent → Environment

(d) Integrate module neural network surrogates

$R\left[\cdot\cdot\right]$ — $R\left[\cdot\cdot\right]$

Physical — Surrogate wrapped in Cheetah — Physical

+ Gradient-based reinforcement learning

+ **All the things you come up with!**
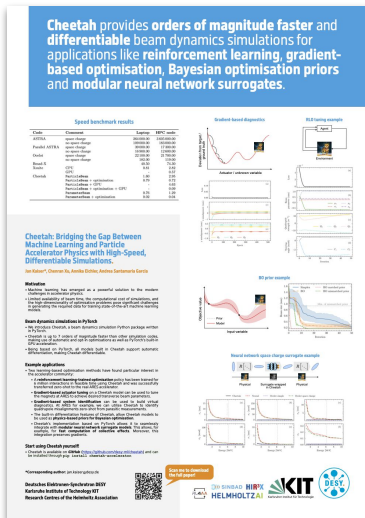
Now you!

Come to the poster.

- **Come to the poster!**
- **Read the paper!**
  - *"Bridging the Gap Between Machine Learning and Particle Accelerator Physics with High-Speed, Differentiable Simulations" in PRAB*

- **Use Cheetah!**

```
pip install cheetah-accelerator
```



```python
# Load initial beam distribution from ASTRA tracking
beam_in = 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
beam_out = segment.track(beam_in)
```
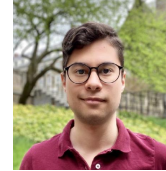
Contact

DESY. Deutsches Elektronen-Synchrotron

www.desy.de

Jan Kaiser
Machine Beam Controls (MSK)
jan.kaiser@desy.de