# Optimal control for rapid switching of beam energies for the ATR line at BNL

*Jonathan Edelen, Nathan Cook, Kevin Brown, and Philip Dyer*

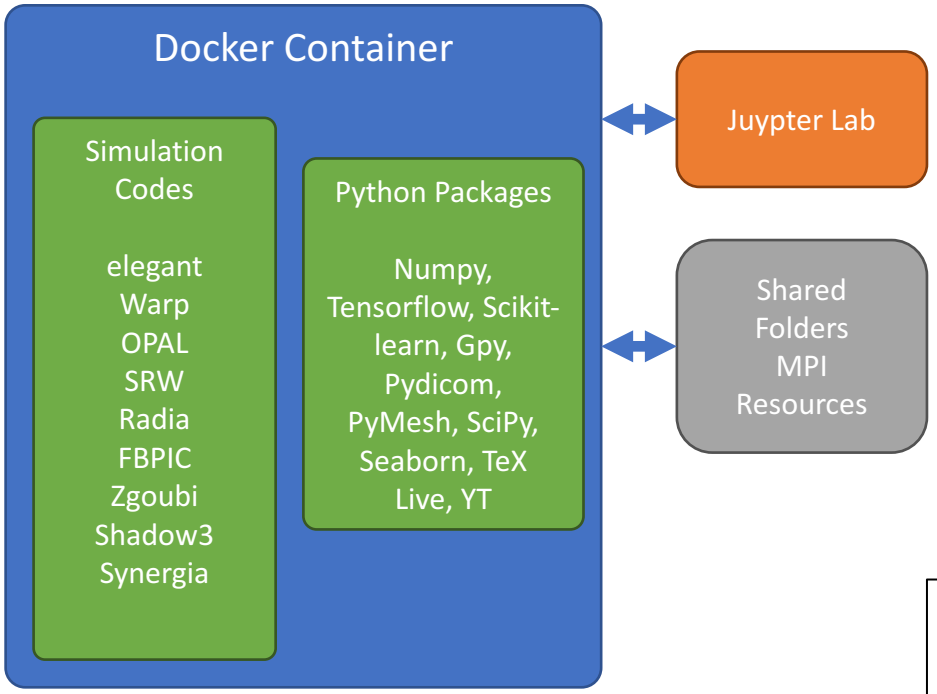
radiasoft

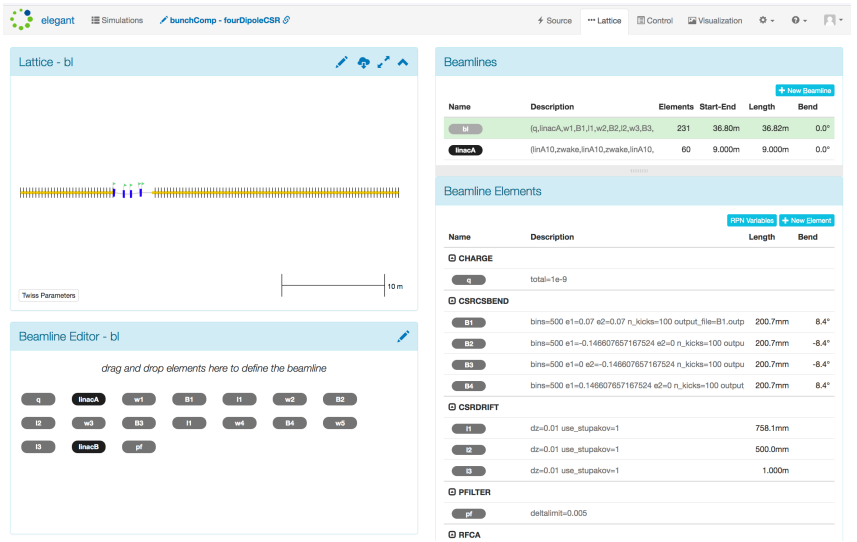
BROOKHAVEN NATIONAL LABORATORY

**presented at ICALEPCS 2019: Feedback Control and Process Tuning**
(Brooklyn, NY)
8 Oct 2019

# Jupyter/Hub

Sirepo

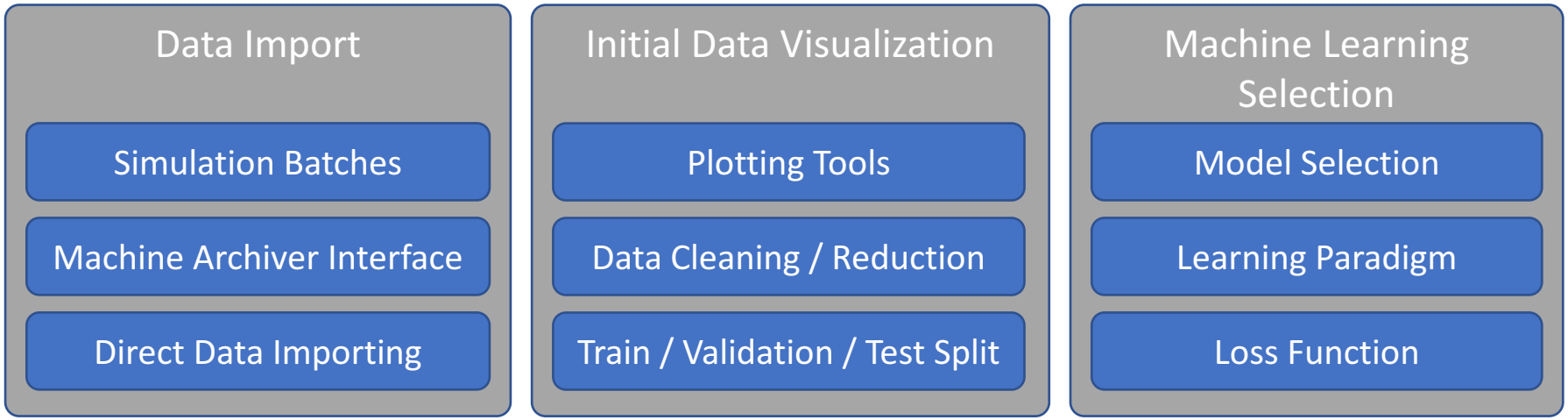


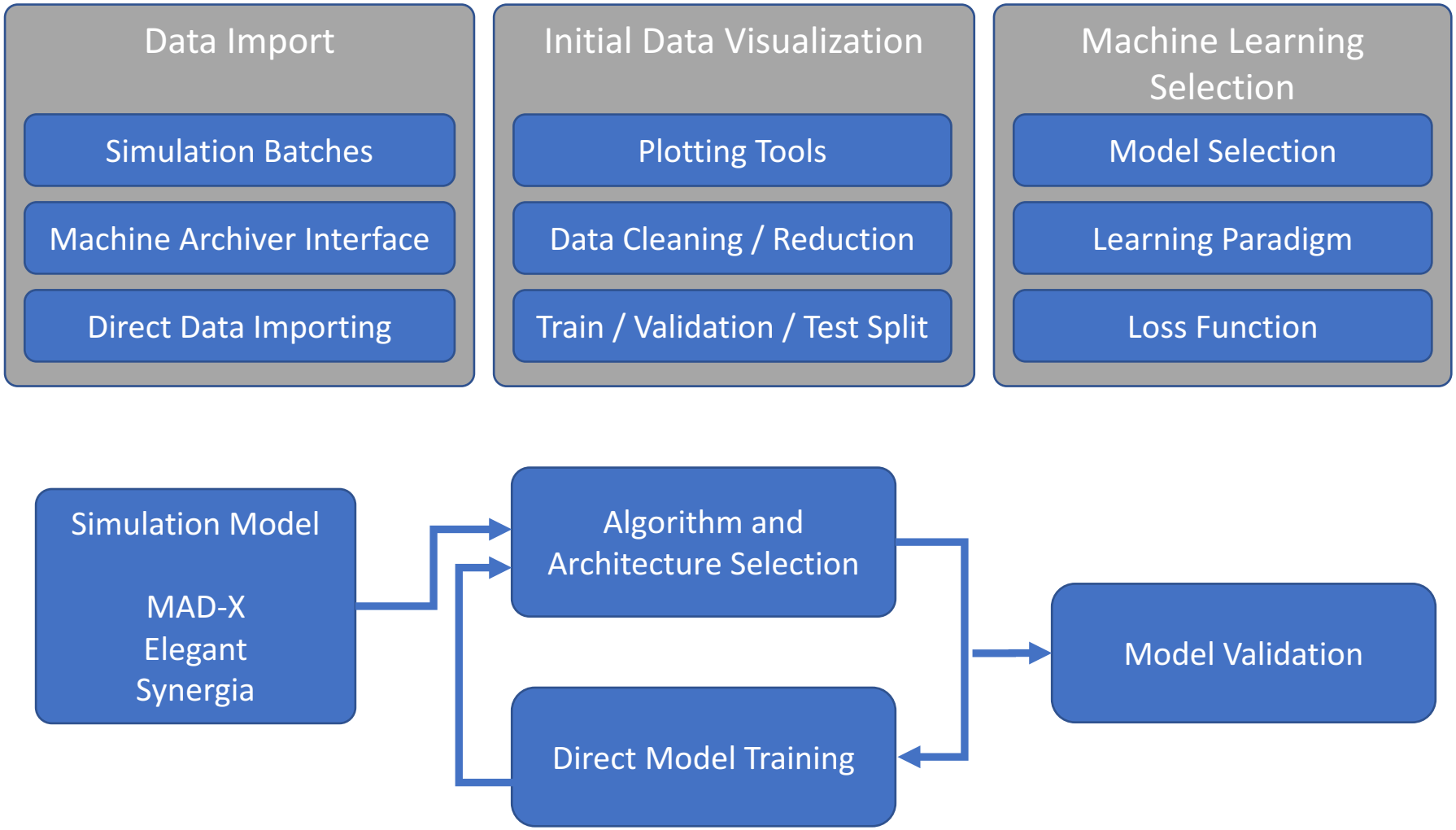

D.L. Bruhwiler *et al.*, "Knowledge Exchange Within the Particle Accelerator Community via Cloud Computing," in *IPAC* (2019).

https://jupyter.radiasoft.org, and https://sirepo.com are free Scientific Gateways

# Developing machine learning tools @ Radiasoft



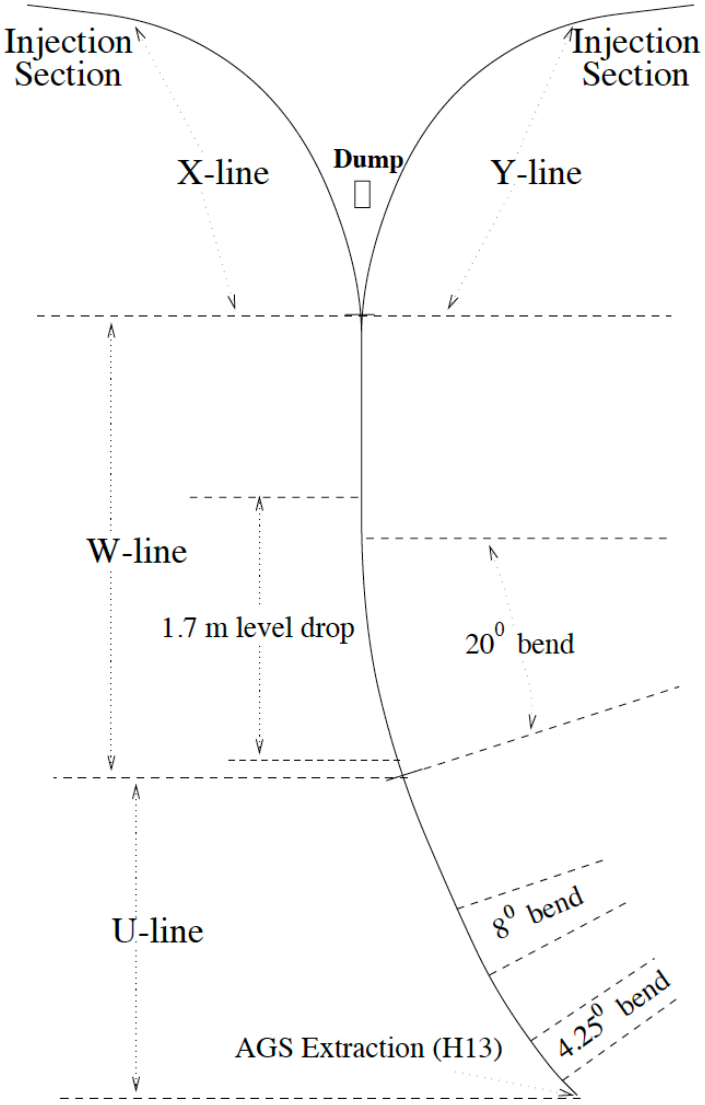
# Developing machine learning tools @ Radiasoft



# The AGS to RHIC transfer line

- 600 Meter transfer line brings beams from the Alternating Gradient Synchrotron (AGS) to the Relativistic Heavy Ion Collider (RHIC)

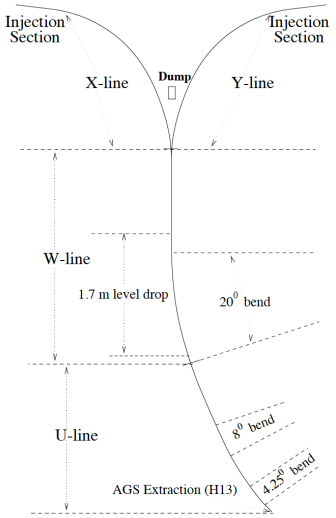
- Energy scan requires re-tuning of the ATR line
  - *Match the beam trajectory*
  - *Match transverse optics*
  - *Make spin transparent*

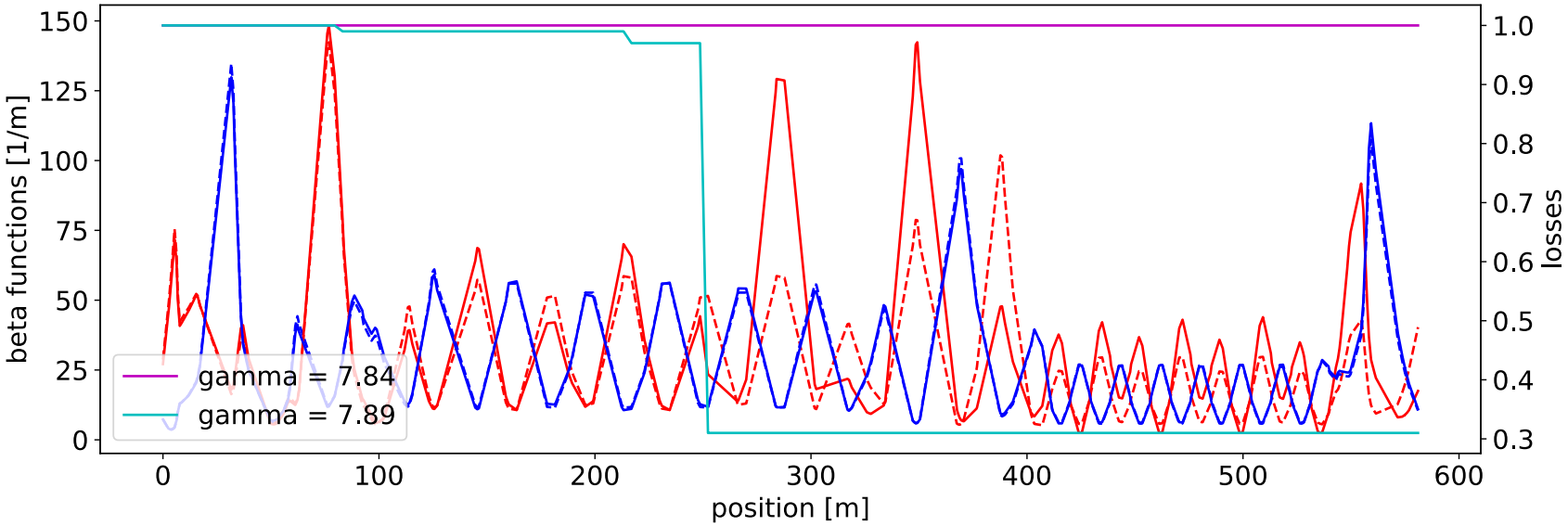


# The AGS to RHIC transfer line

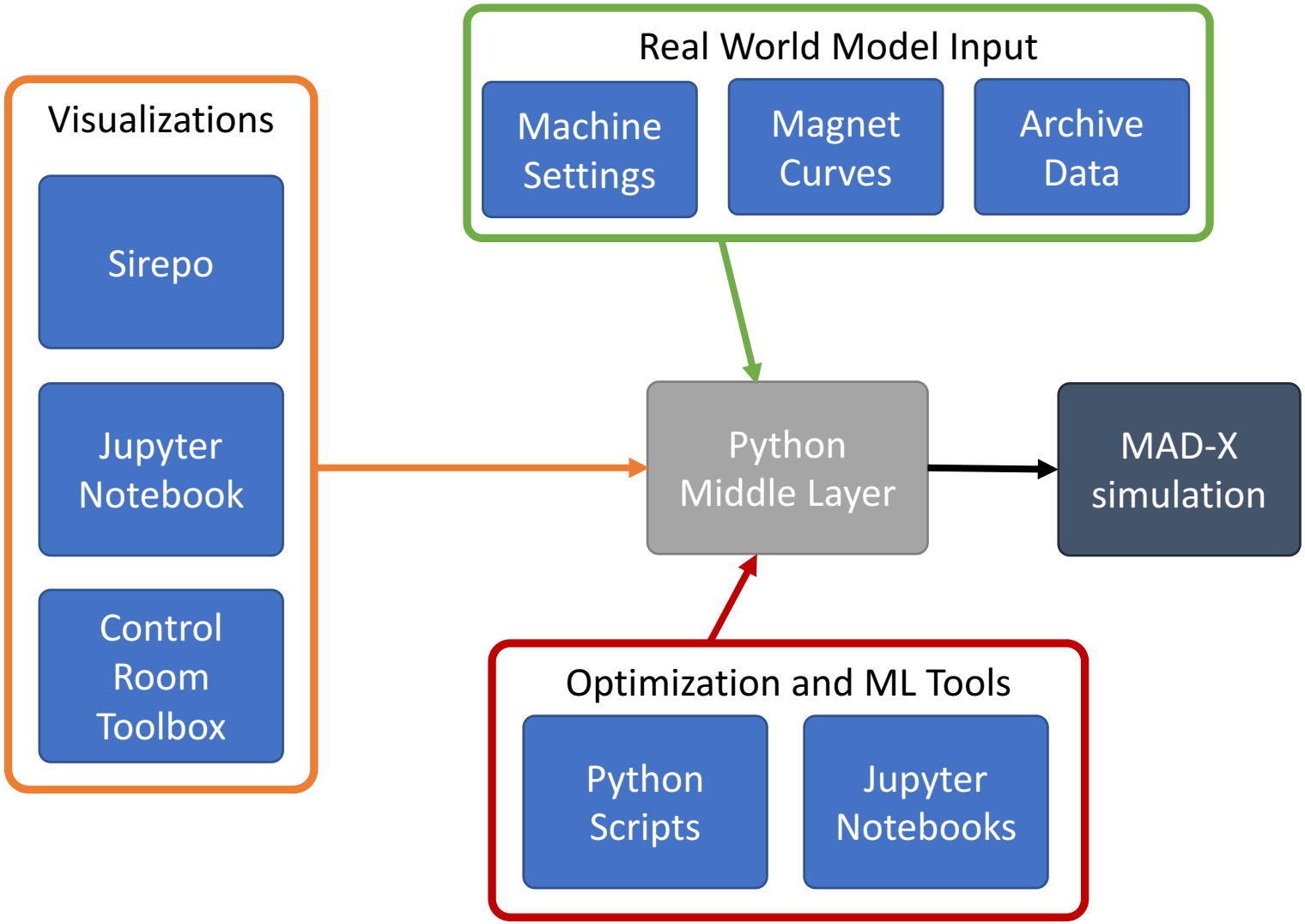
- 600 Meter transfer line brings beams from the Alternating Gradient Synchrotron (AGS) to the Relativistic Heavy Ion Collider (RHIC)

- Energy scan requires re-tuning of the ATR line
  - *Match the beam trajectory*
  - *Match transverse optics*
  - *Make spin transparent*



# Developing a toolbox for simulations and ML



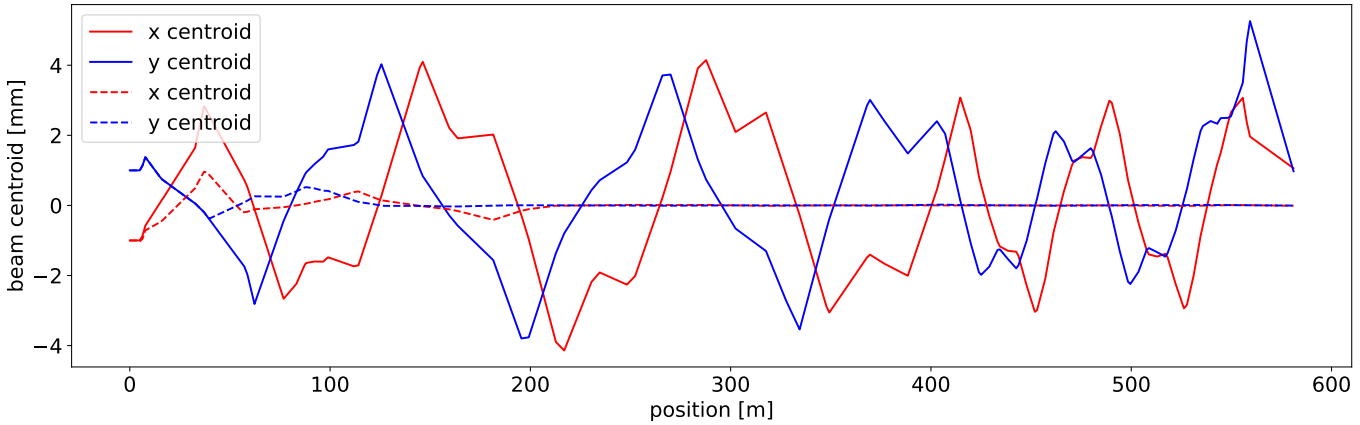
# Beam steering in the ATR

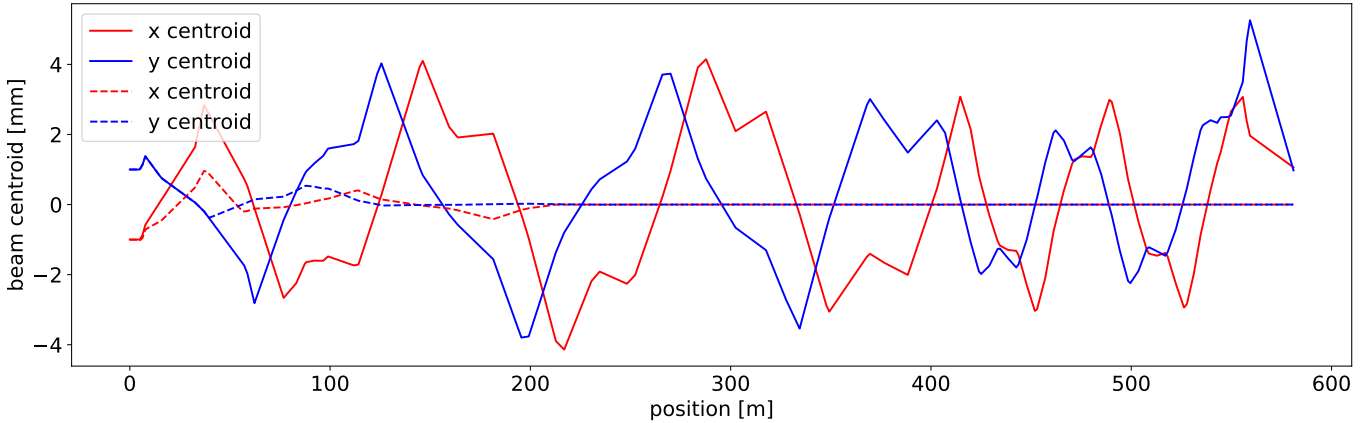
- Optimization
  - *Connect MAD-X simulation to python optimization tools using our middle layer*
  - *Study convergence rate for tuning the trajectory over a range of initial offsets*

- Machine Learning
  - *Build neural network model of corrector settings to bpm-readings*
    - Use with optimization
  - *Build inverse model of bpm-readings to corrector settings*
    - Make feed-forward correction

# Beam steering in the ATR with optimization

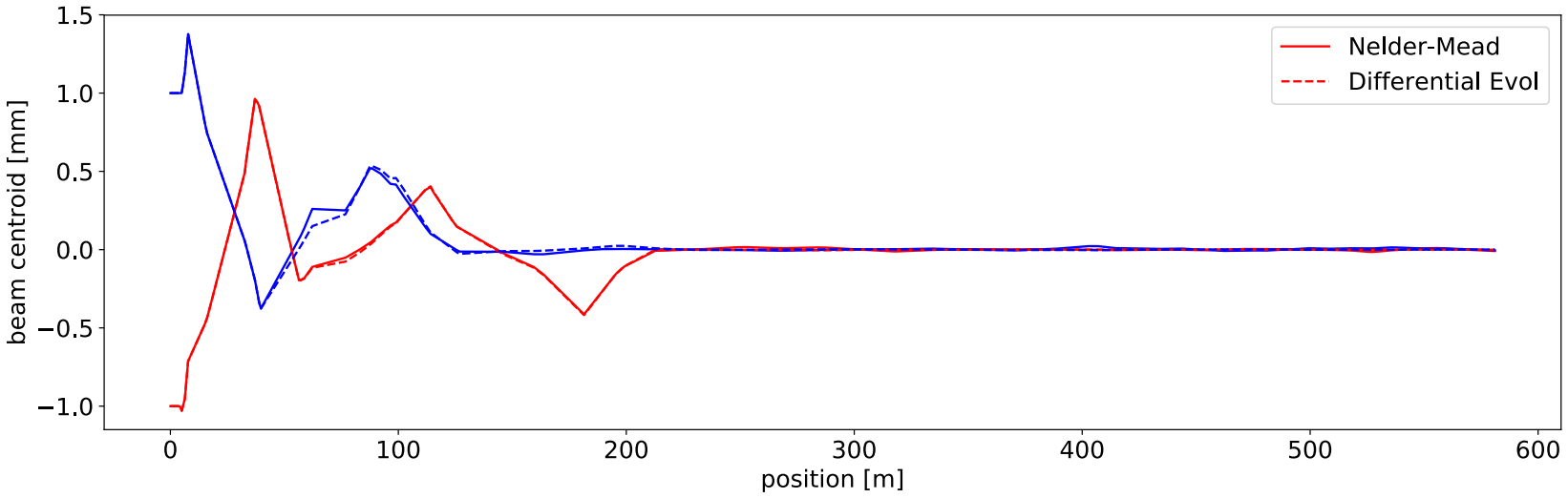
- Optimization
  - *Connect MAD-X simulation to python optimization tools using our middle layer*
  - *Study convergence rate for tuning the trajectory over a range of initial offsets*
  - *Nelder-Mead: Converges in ~ 2500 function evaluations (top)*
  - *Differential Evolution: Converged in ~11000 (bottom)*



# Beam steering in the ATR with optimization

- Optimization
  - *Connect MAD-X simulation to python optimization tools using our middle layer*
  - *Study convergence rate for tuning the trajectory over a range of initial offsets*
  - *Nelder-Mead: Converges in ~ 2500 function evaluations*
  - *Differential Evolution: Converged in ~11000*



# Beam steering in the ATR

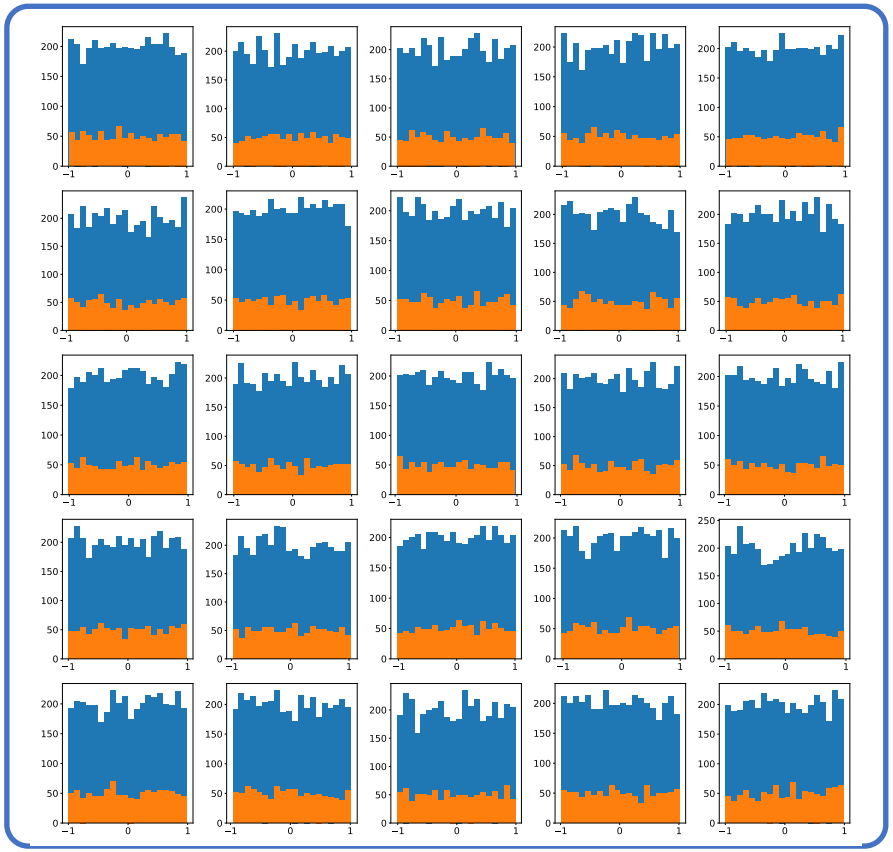
- Optimization
  - *Connect MAD-X simulation to python optimization tools using our middle layer*
  - *Study convergence rate for tuning the trajectory over a range of initial offsets*

- Machine Learning
  - *Build neural network model of corrector settings to bpm-readings*
    - Use with optimization
  - *Build inverse model of bpm-readings to corrector settings*
    - Make feed-forward correction

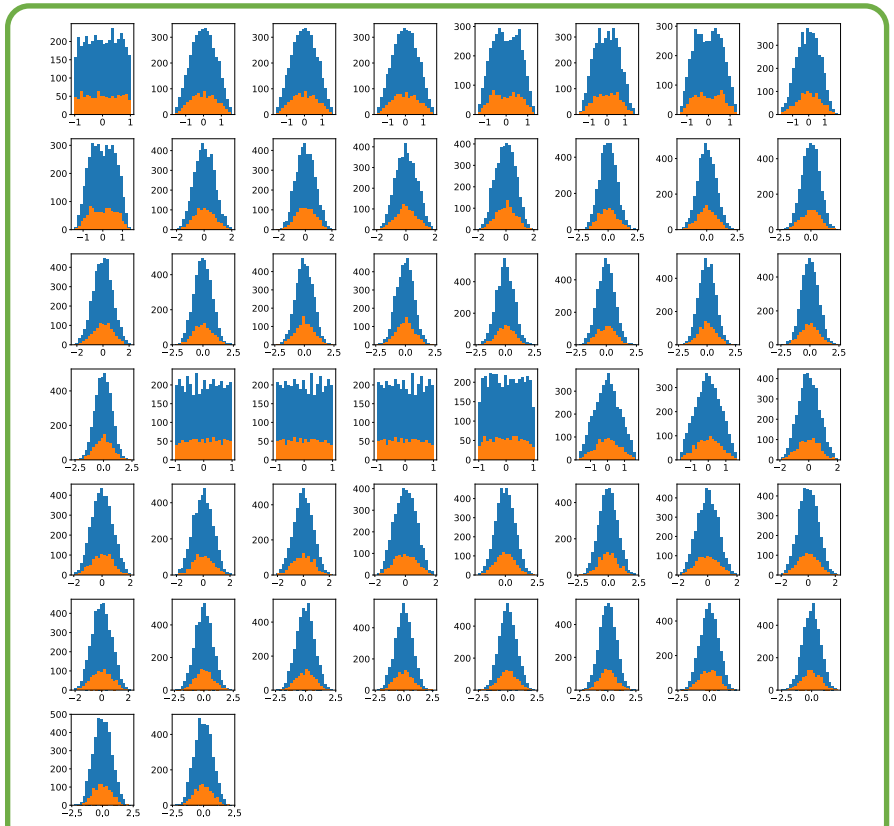
# Beam steering in the ATR with machine learning

- Train neural network to predict BPM readings from initial offset and corrector settings

Inputs to the model (beam offset and corrector settings)

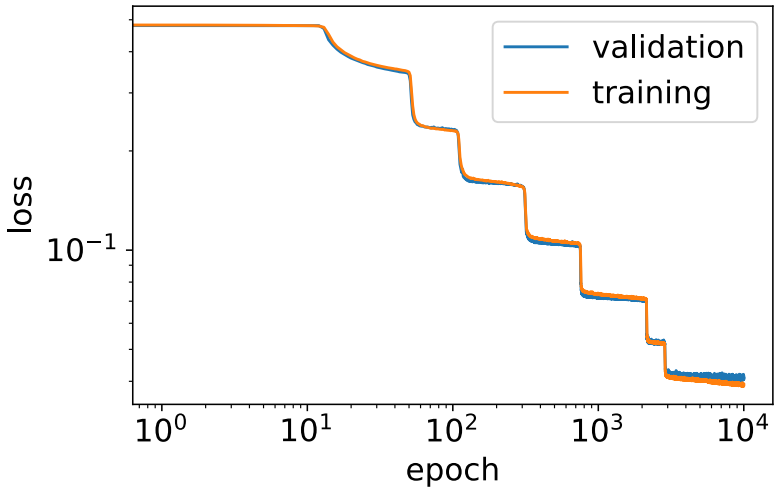


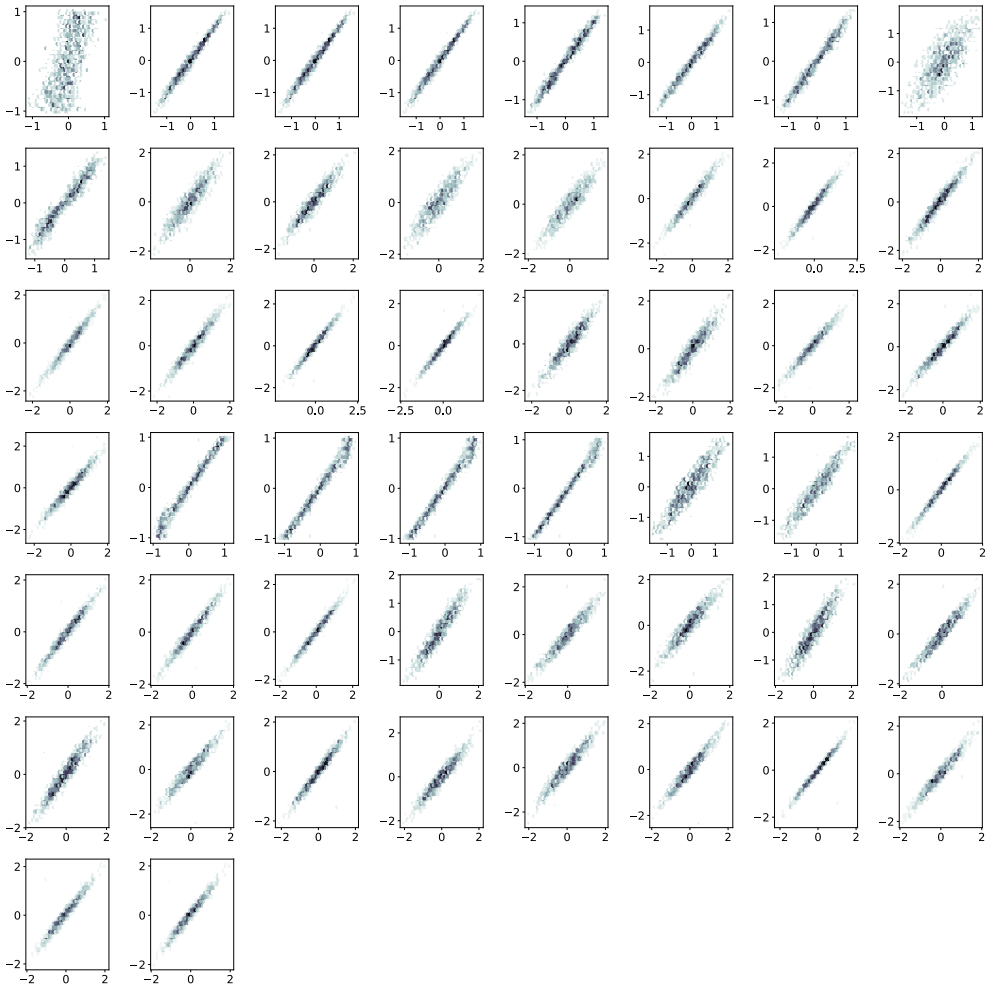
Outputs from the model (beam position readings)



# Beam steering in the ATR with Machine Learning

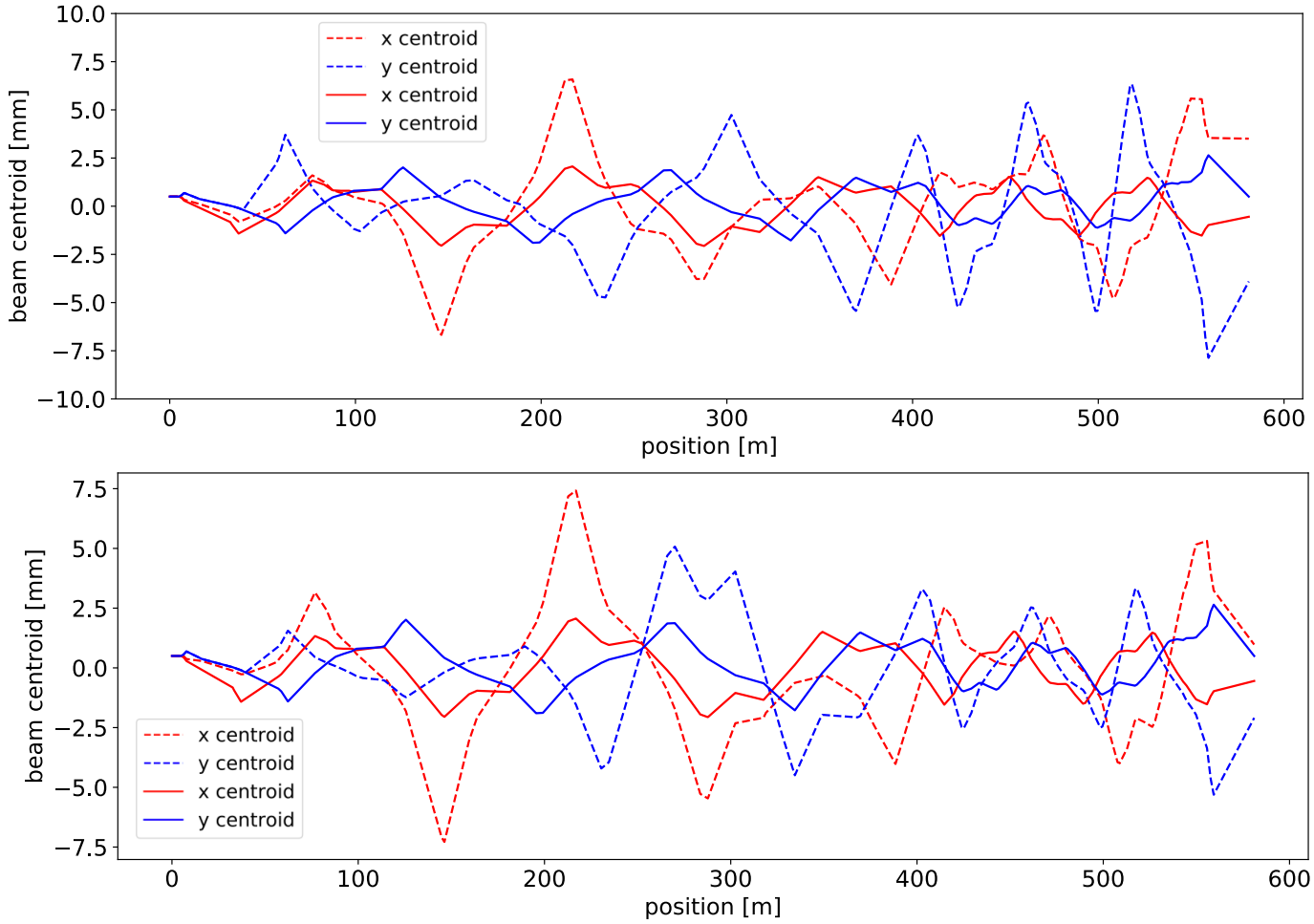
- 15 layers, 15 nodes per layer
- "relu" activation functions
- Gaussian noise layers



# Optimization with neural network surrogate

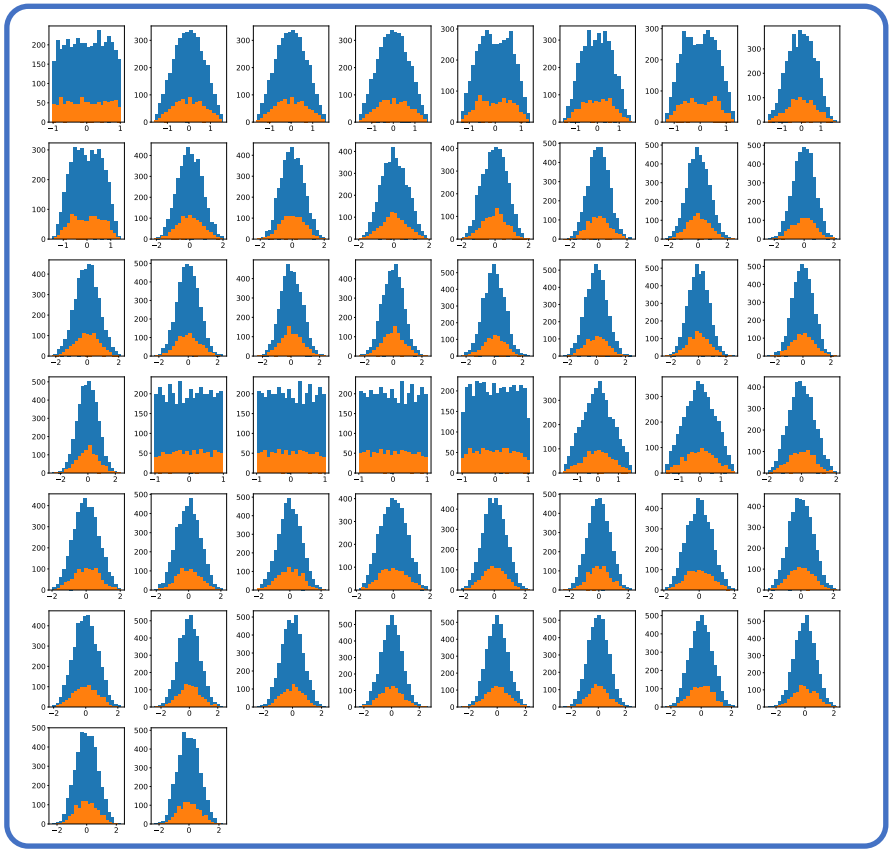
- Neural network evaluates orders of magnitude faster than MADX simulation
- In general our network does not work for this application



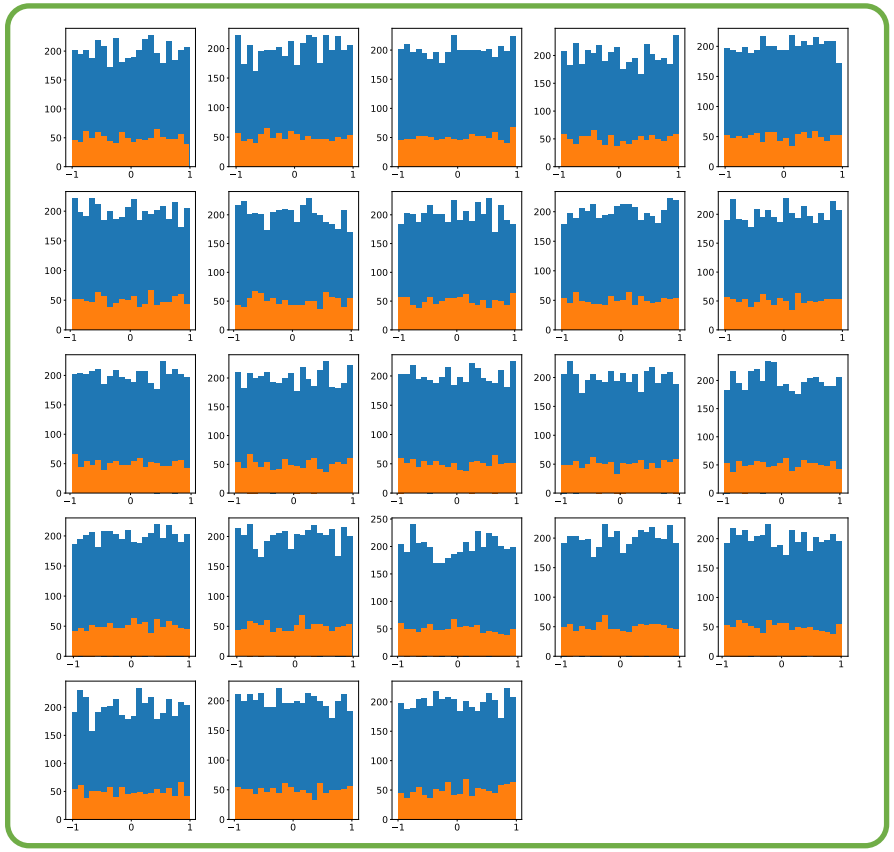
# Beam steering in the ATR with Machine Learning

- Inverse model: train neural network to predict corrector settings from BPM readings

Inputs to the model (beam offset and beam position readings)

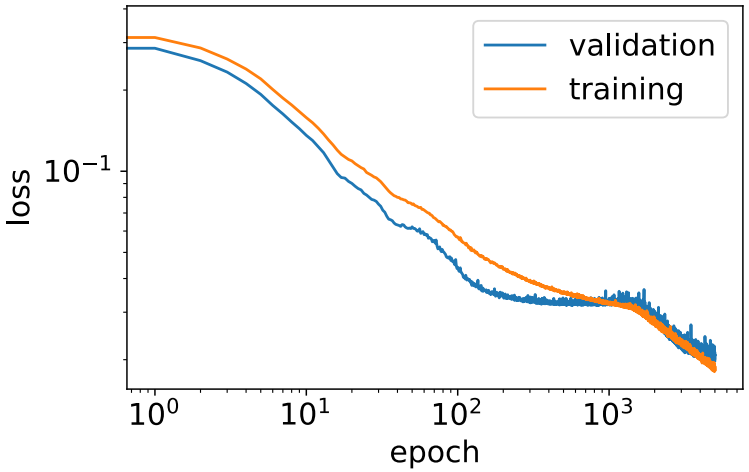


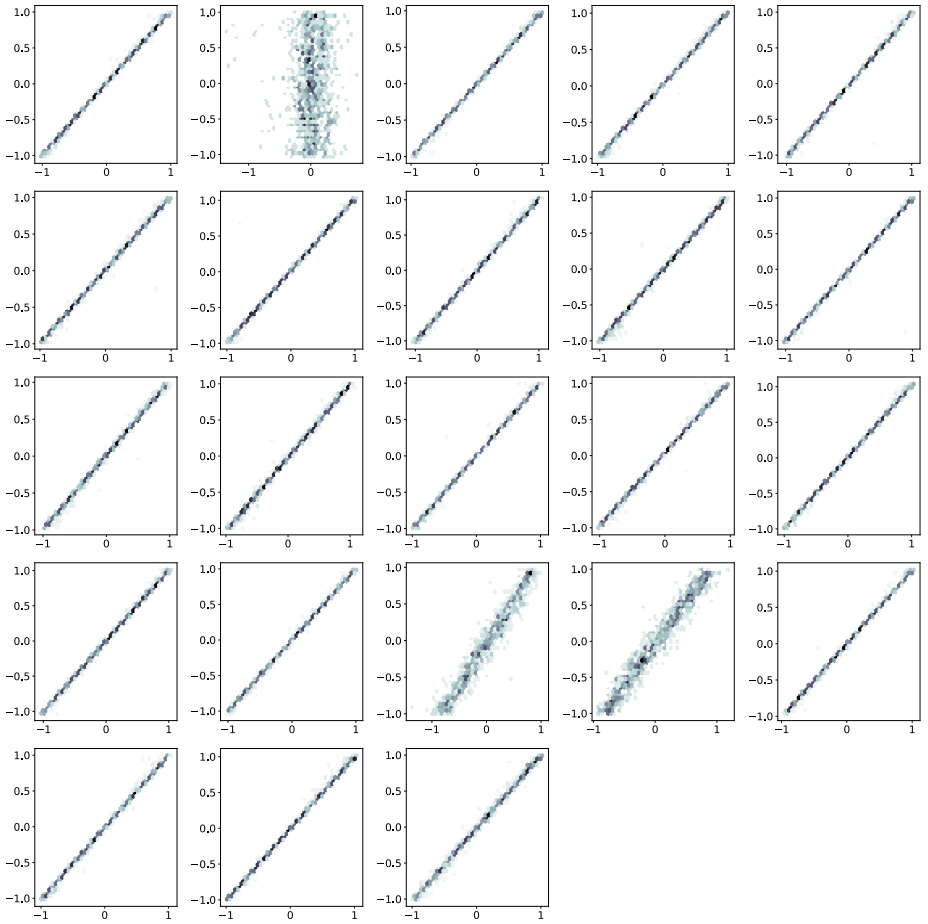
Outputs from the model (corrector settings)



# Beam steering in the ATR with Machine Learning

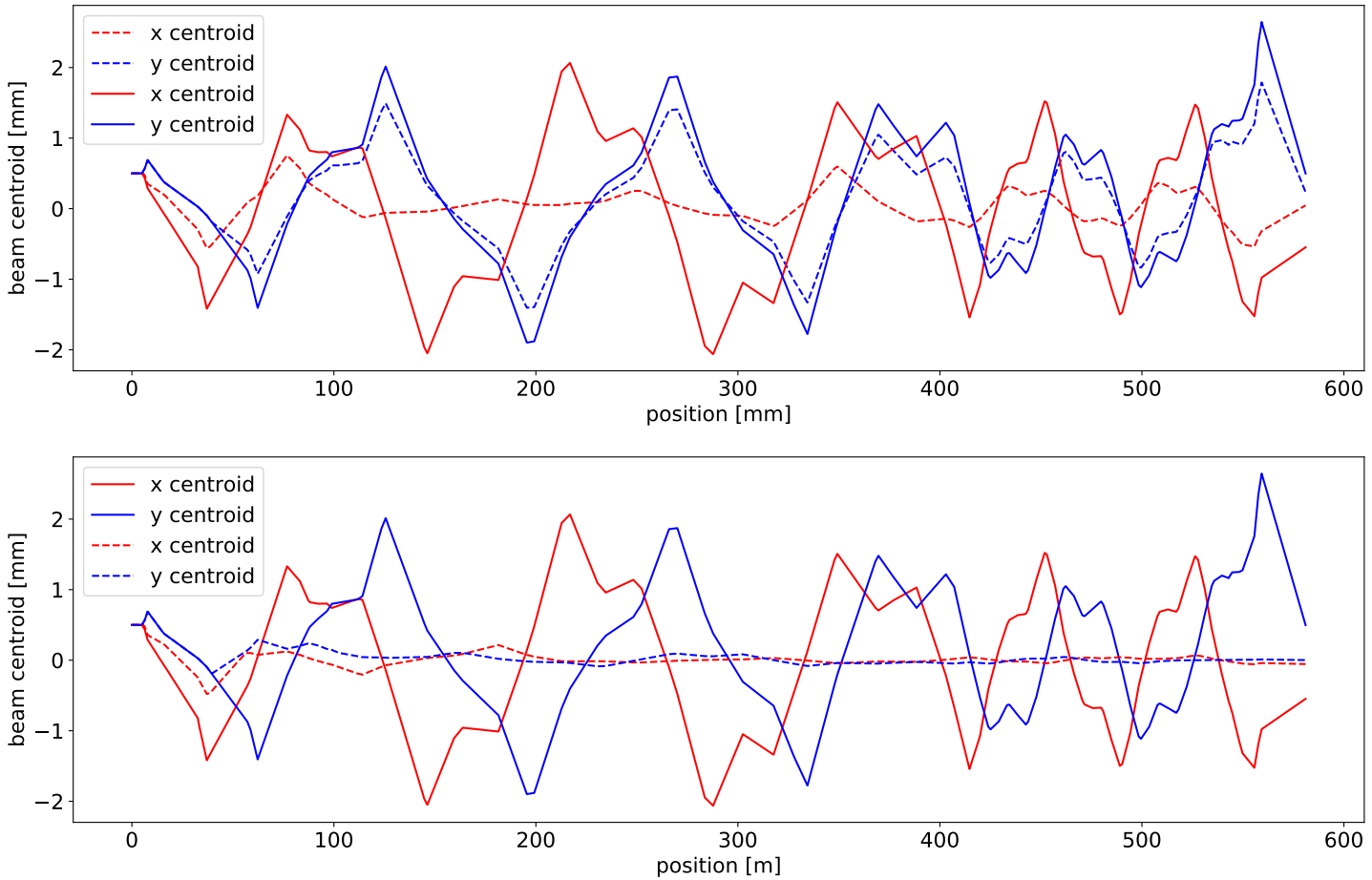
- 3 layers, (40,50,50)
- "relu" activation functions
- Gaussian noise layers



# Using inverse model for trajectory control

- Compute BPM settings from desired input trajectory
- Use as starting point for optimization



# Summary and Conclusions

- We want to provide tools for the community that enable wider / easier use of advanced methods for control
  - *Continuing to pursue new collaborations*

- Developing tools to improve the adoption of ML techniques for Accelerators

- Working towards using ML for control problems at BNL and also at Fermilab