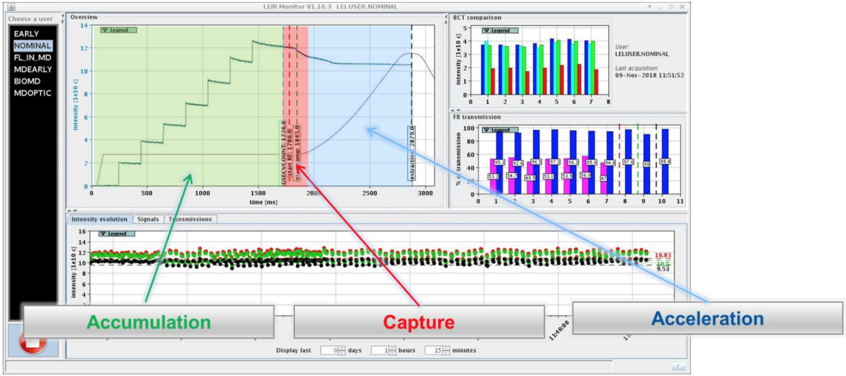
Status of Reinforcement Learning Studies for E-Cooler Operation

N. Biancacci V. Kain A. Latina ▪ N. Madysa

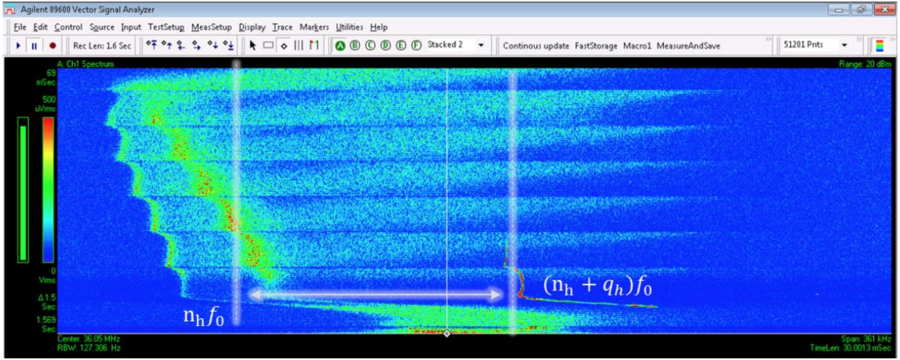
E-BEAM Meeting, 9 September 2020

LEIR Operation



- accumulates injections from Linac 3
- 7 injections every 200 ms, each taking 200 µs
- cooling throughout flat bottom ⇒ fill phase space efficiently
- bunching via RF Capture, acceleration, transfer to PS

LEIR Operation



- accumulates injections from Linac 3
- 7 injections every 200 ms, each taking 200 µs
- cooling throughout flat bottom ⇒ fill phase space efficiently
- bunching via RF Capture, acceleration, transfer to PS

Goal

given the previous process:

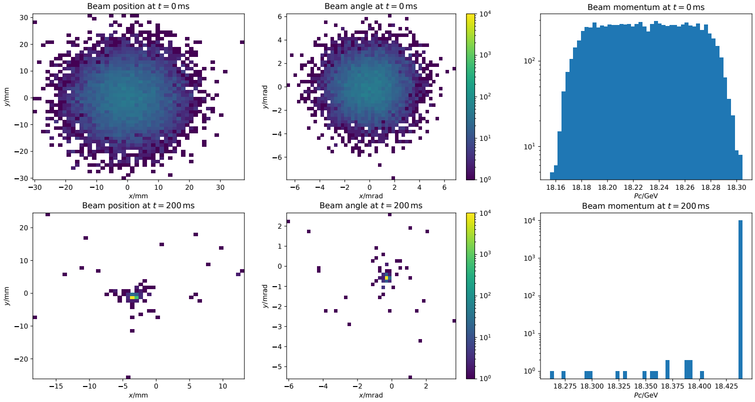
- cooling parameters,
- beam properties before cooling,
- beam properties after cooling,

adjust parameters so that beam after cooling follows some distribution

We will work our way up:

1. optimize for minimal $\sigma[p_{\mathrm{out}}]$
2. optimize for given $\mu[p_{\mathrm{out}}], \sigma[p_{\mathrm{out}}]$
3. optimize other variables than p_{out}
4. work with distributions
5. work with Schottky spectra

Training Data



- simulation via RFTrack
- simulates the cooler, ignores rest of the accelerator

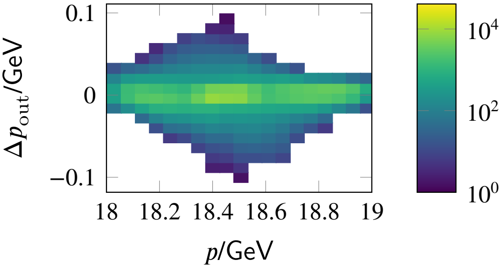
input: random sample of macroparticles

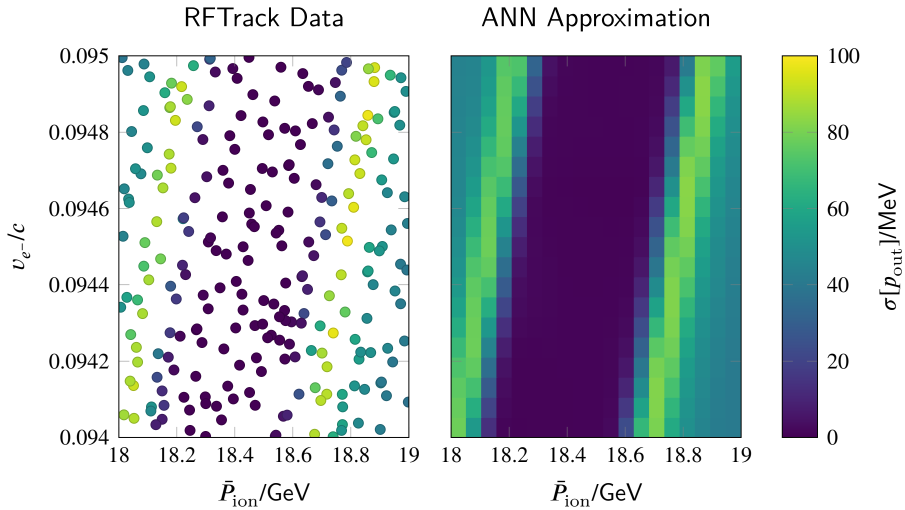
output: cooled macroparticles

Momentum Predictor

- problem: RFTrack is very complex, takes a long time to complete
- solution: train an ANN to predict outcome of RFTrack simulation

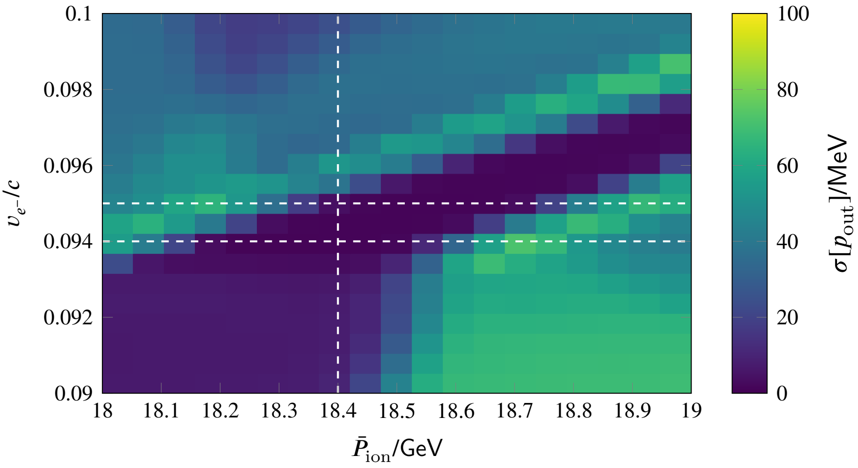
⇒ supervised learning, function approximation



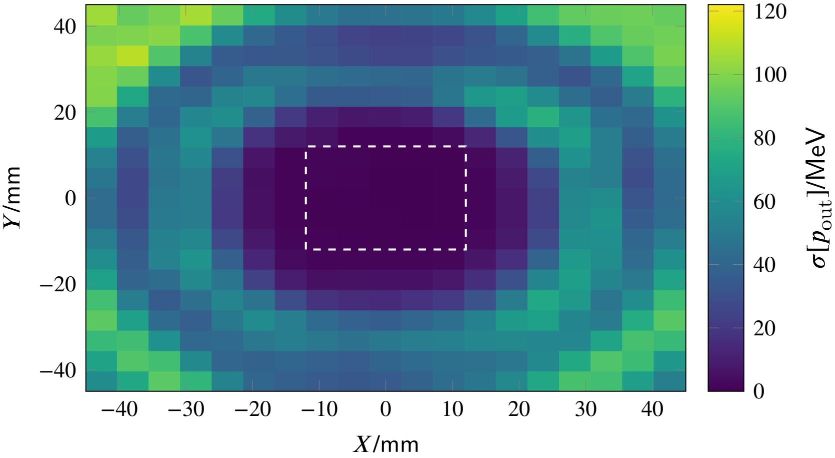


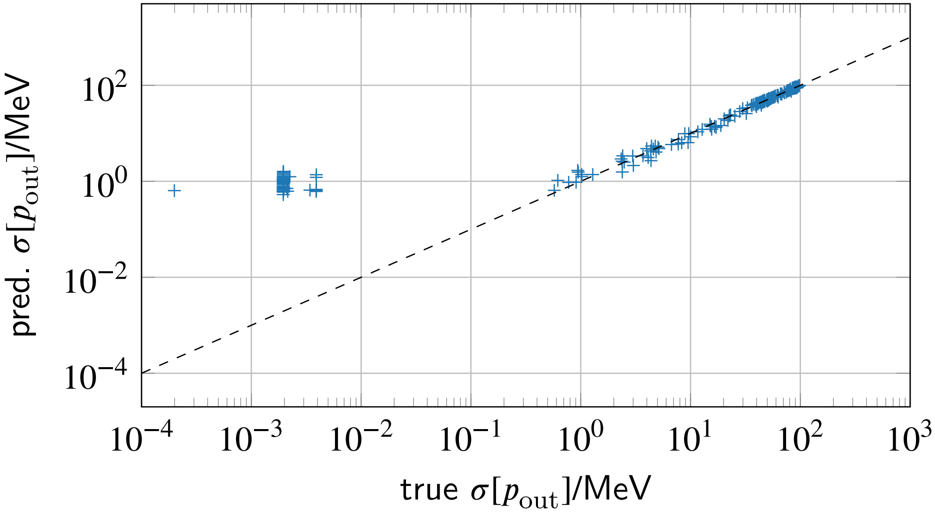
RFTrack Data
ANN Approximation
0.095
0.0948
0.0946
0.0944
0.0942
0.094
v_{e^-}/c
18
18.2
18.4
18.6
18.8
19
$\bar{P}_{\text{ion}}$/GeV
100
80
60
40
20
0
$\sigma[p_{\text{out}}]$/MeV

ANN Approximation (extended)



ANN Approximation (extended)

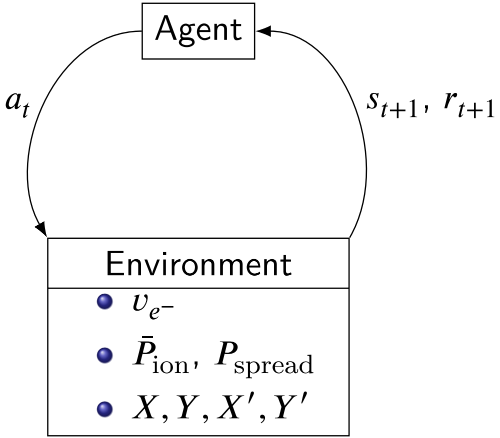




pred. $\sigma[p_{out}]$/MeV
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-4} 10^{-3} 10^{-2} 10^{-1} 10^{0} 10^{1} 10^{2} 10^{3}
true $\sigma[p_{out}]$/MeV

Cooling Agent

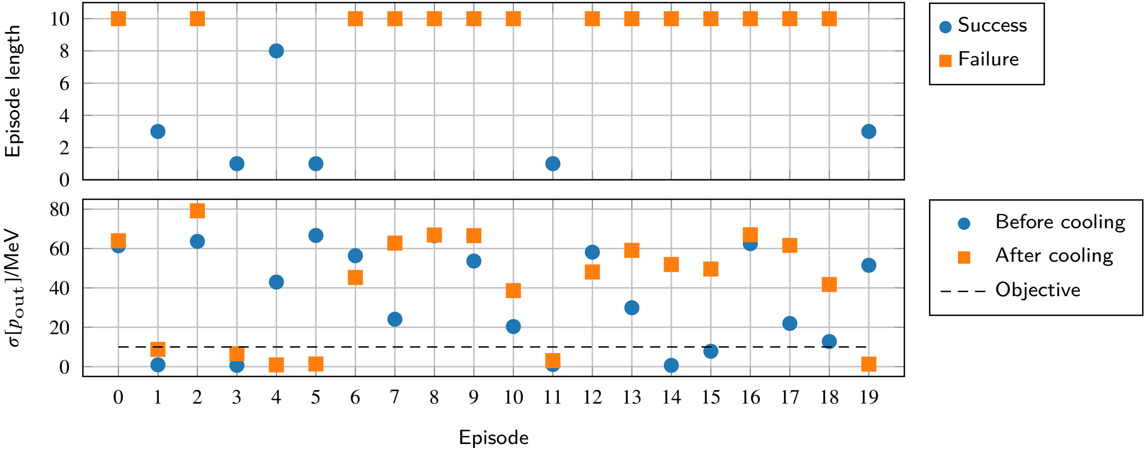
- with predictor, we can generate as many data samples as we need
- use it to train reinforcement learning agent
- agent sends cooler settings corrections (Δv_{e^-}, ...) to environment
- environment returns next cooling result and a reward



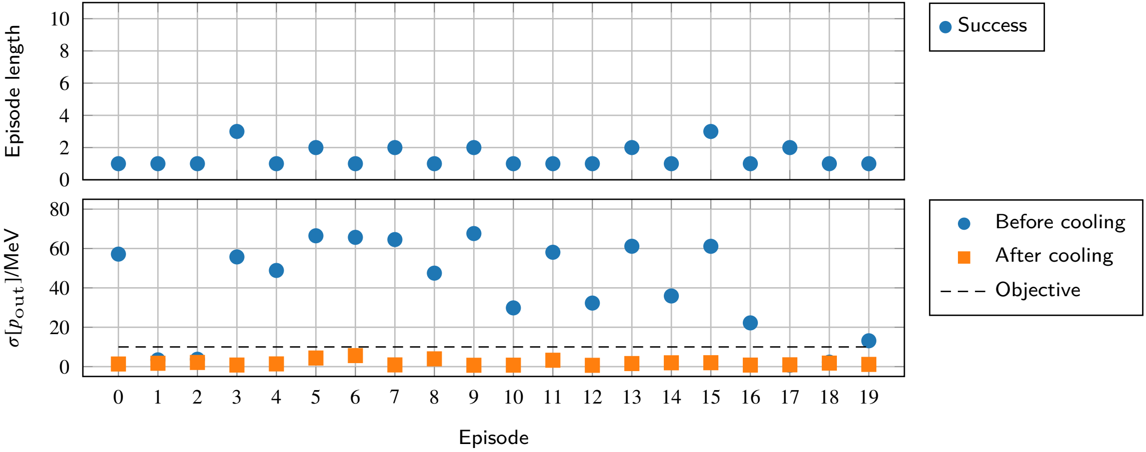
OpenAI Gym API

```python
import gym
from stable_baselines3 import TD3

class LeirEnv(gym.Env): ...

env = LeirEnv(...)
agent = TD3('MlpPolicy', env, ...)
agent.learn(10000)

obs = env.reset()
done = False
while not done:
    action, _ = agent.predict(obs)
    obs, reward, done, info = obs.step(action)
    ...
```

Picking Random Actions

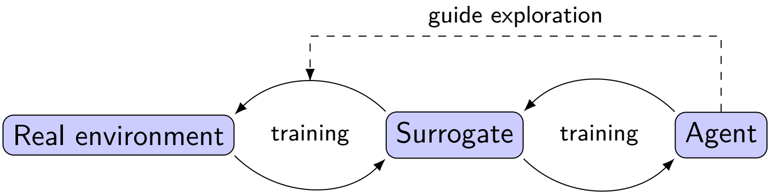


TD3 Algorithm

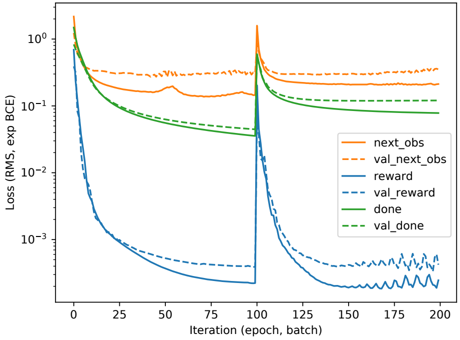


Dyna Architecture

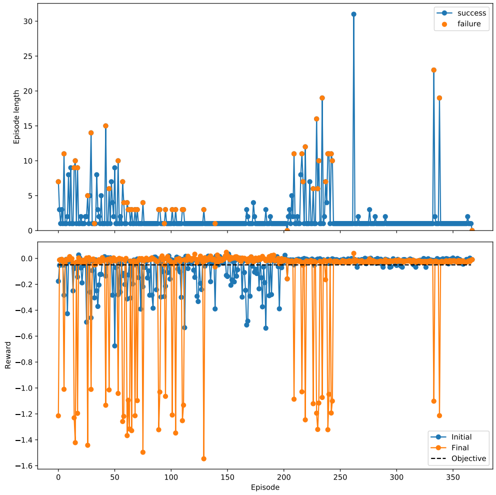
- general approach to improve sample efficiency of model-free agents
- approximate slow real environment with a fast *surrogate* (avoid machine interactions)
- train RL agent on surrogate
- switch between training both



Dyna Surrogate Training

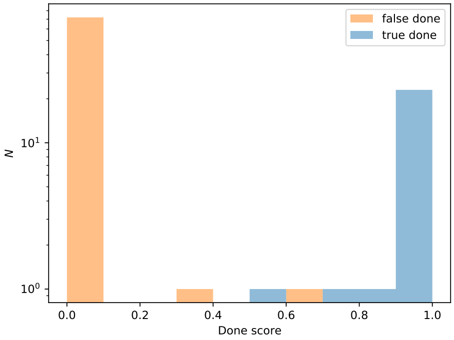


Training of surrogate

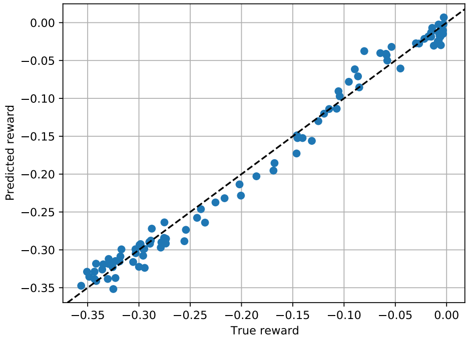


Training of agent on surrogate

Dyna Surrogate Training

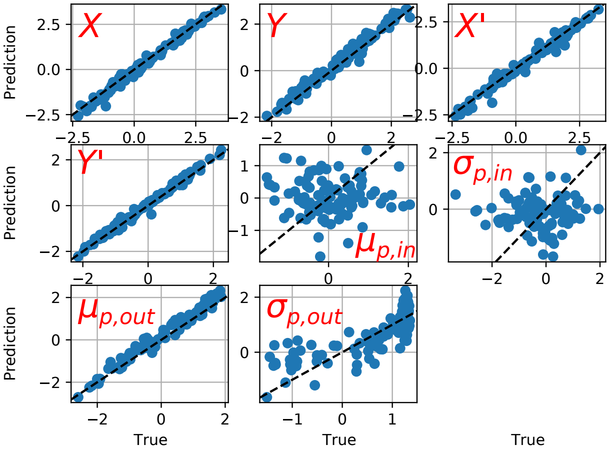


Prediction of end of episode



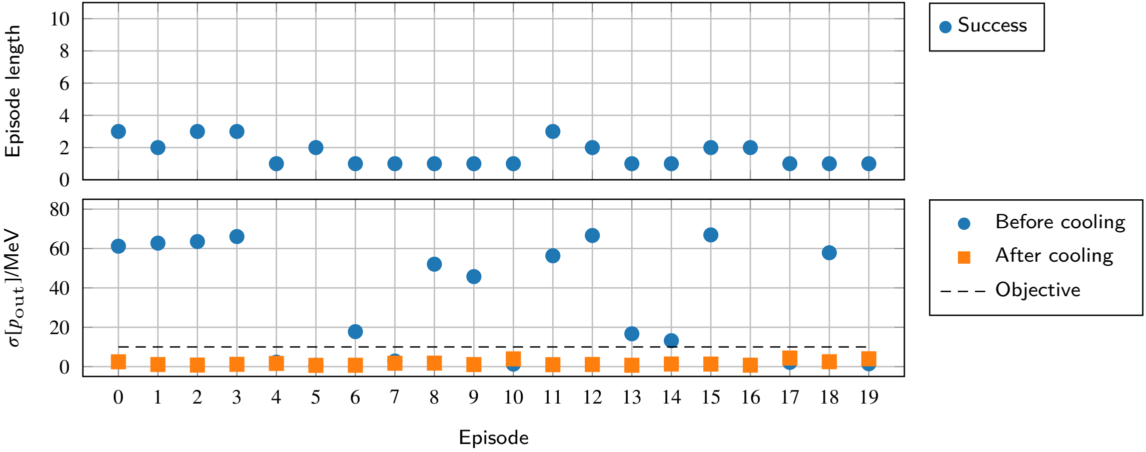
Prediction of reward for each step

Dyna Surrogate Training



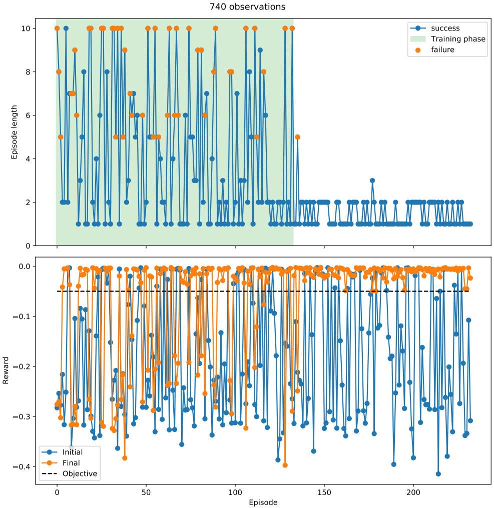
Prediction of next observation (standardized)

Dyna+TD3

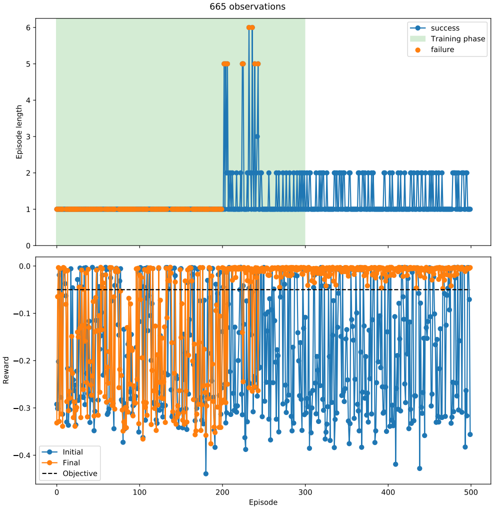


Episode length
10
8
6
4
2
0
Success
80
60
40
20
0
$\sigma[p_{\mathrm{out}}]$/MeV
Before cooling
After cooling
Objective
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Episode

Plain TD3 vs. Dyna+TD3



Plain TD3



Dyna+TD3

Conclusions:

- replaced RFTrack simulation with an ANN
- trained RL agent on simulated data
- improved sample efficiency with model-based RL

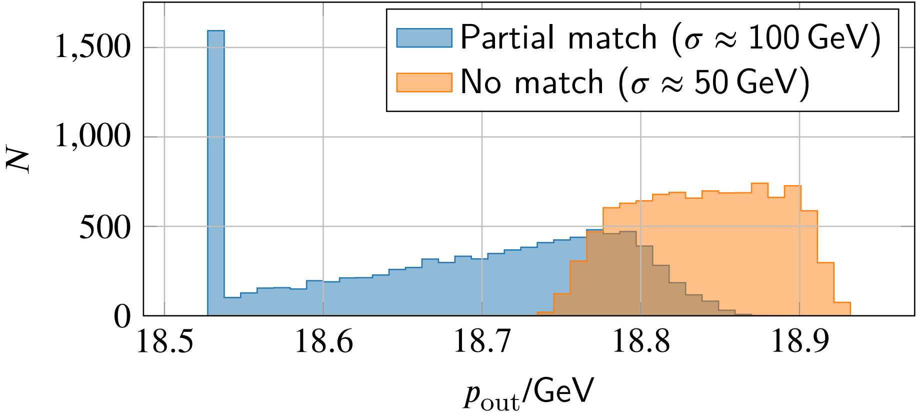
Next steps:

- transition predictor from individual particles to beam statistics
- extend optimization problem: $\sigma[x_{\mathrm{out}}]$, $\sigma[y_{\mathrm{out}}]$, $\mu[p_{\mathrm{out}}]$

Eventually:

- transition to full distributions of p, x, y, ...
- transition to Schottky spectra

Backup



Reason for ridge structure:

- distribution stays the same if β_{e^-} and $\bar{P}_{\text{ion}}$ don't match
- distribution is torn apart if β_{e^-} and $\bar{P}_{\text{ion}}$ match partially