

Dynamic/Momentum Aperture Optimization with MultipointBAX

Apr 9, 2025

D. Ratner on behalf of Z. Zhang, X. Huang
SLAC National Accelerator Laboratory

Multi-point optimization

Classic problem from aeronautics design:
how to optimize an airplane wing?

1. Choose a design
2. Simulate range of conditions
3. Combine into single metric / multiple metrics
4. Loop to step 1

Very slow process!
Very information inefficient!

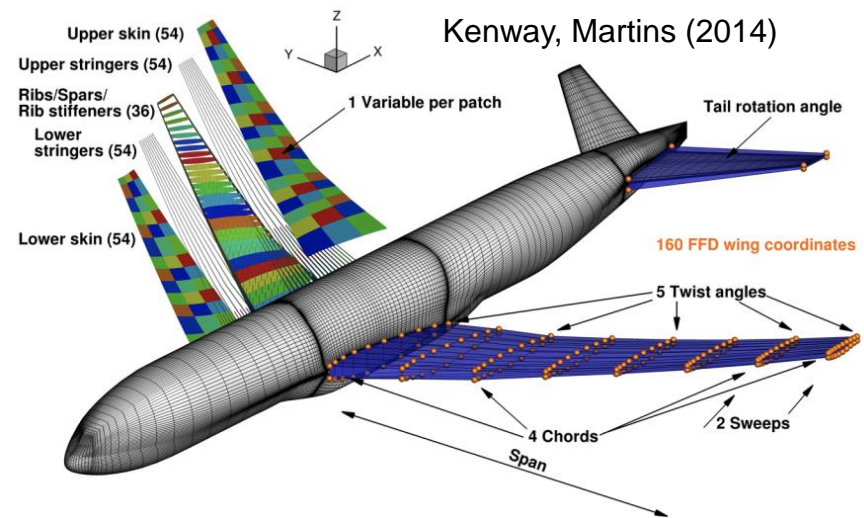
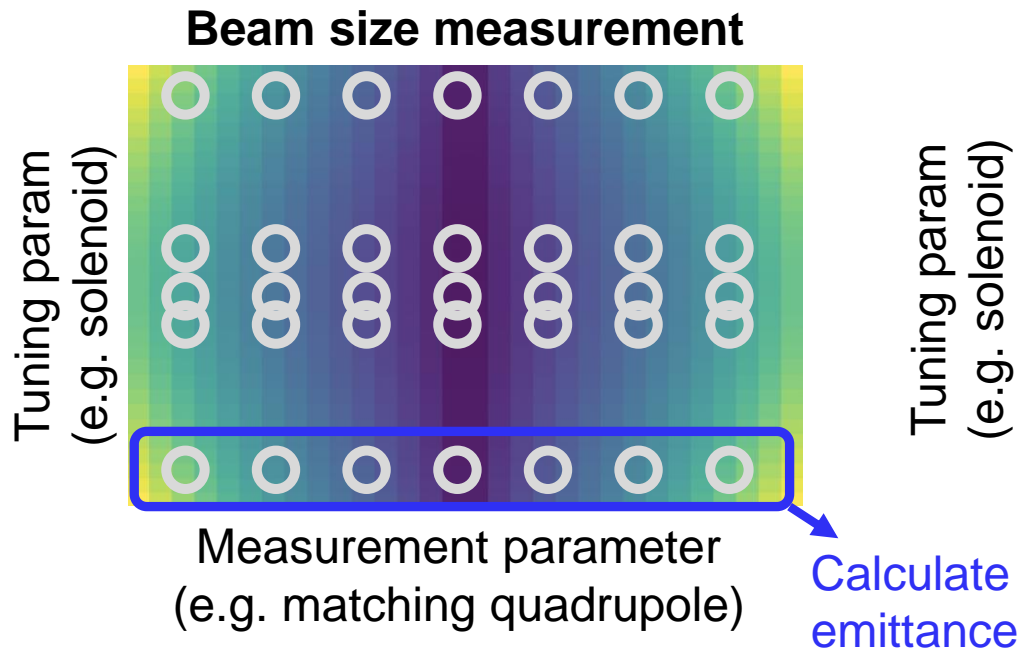


Image courtesy FAA

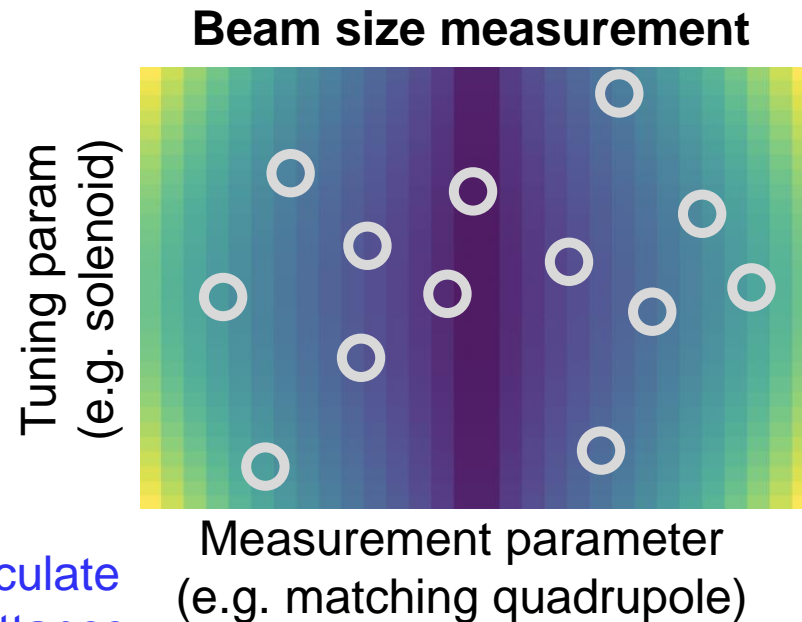
Emittance example

Accelerator example: emittance optimization



BO has multiple problems:

- Sampling inefficient
- Information inefficient
- Difficult to model



Models beam size efficiently...

...but we want to minimize emittance, not beam size

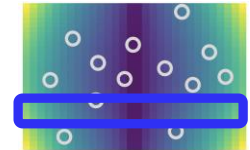
Bayesian Algorithmic Execution (BAX)

Standard BO: model and find optimal point in black box function $f(x)$

BAX: model $f(x)$, and find optimal output of $A[f(x)]$, for known algorithm $A[\cdot]$

Emittance example:

- Model beam size behavior, $f(x)$
- Algorithm, $A[\cdot]$, outputs minimum emittance from $f(x)$
- Measure new beam size that has the biggest impact on output of $A[\cdot]$



Note: We never actually measure emittance... just calculate from a virtual model!

(For computer scientists, infoBAX acquisition function calculated from mutual information of algorithm output and model posterior)

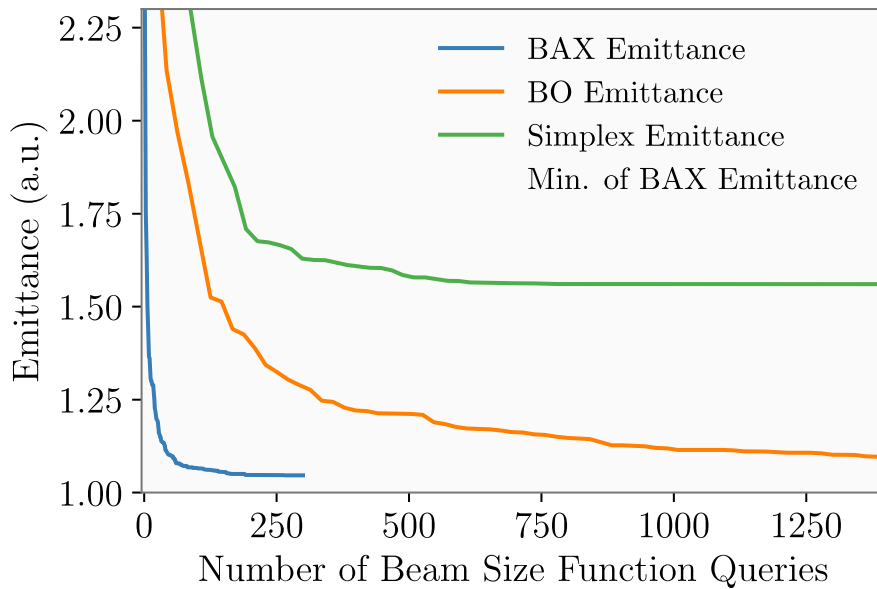
→ **BAX:** <https://willieneis.github.io/bax-website/>

W. Neiswanger et al., ICML, 2021

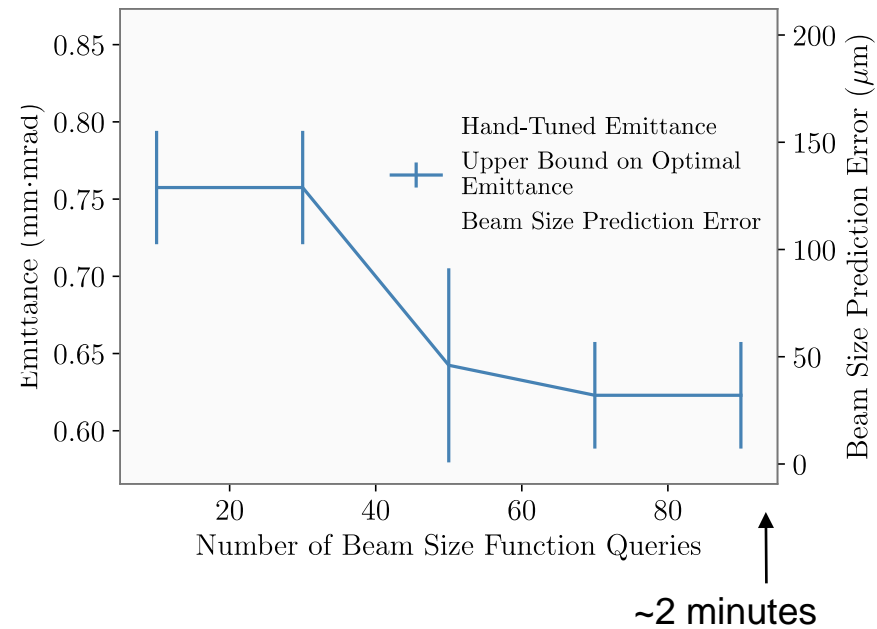
Emittance example

Accelerator example: emittance optimization

Simulated optimization, LCLS Injector



Experimental optimization, LCLS Injector



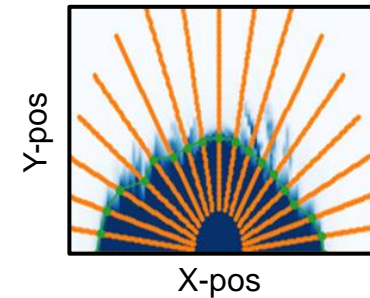
Application to dynamic/momentum aperture

Now consider dynamic aperture (DA) optimization for storage rings

Goal: maximize dynamic aperture
(i.e. the dark area):

DA is a hard problem for two reasons:

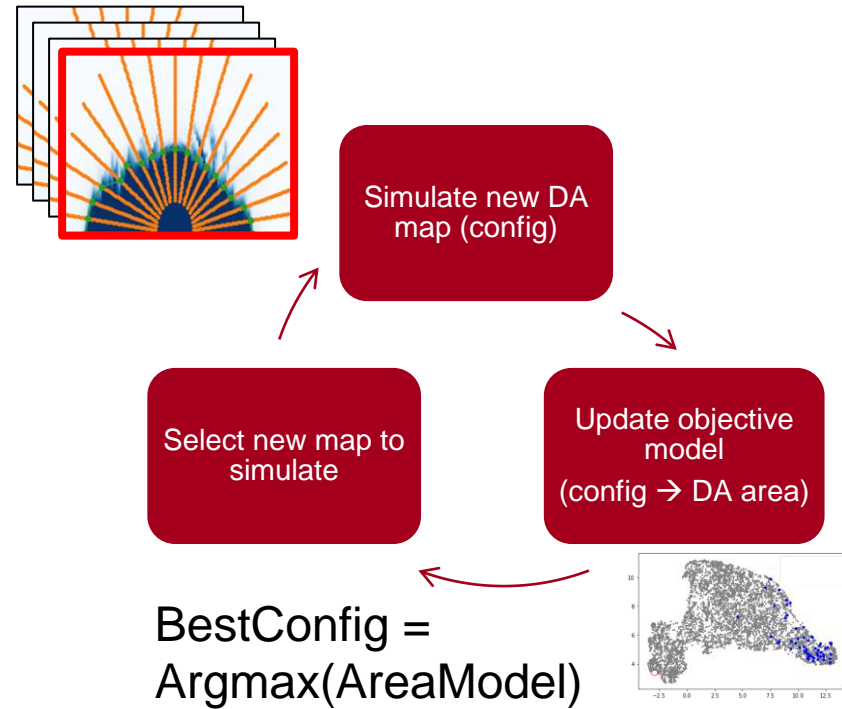
- High dimensional optimization (typically 10s to 100s of parameters)
- Acquisition is expensive: Each calculation of the objective function requires tracking 100s -1000s of particles for 1000s of turns



ML for DA/MA Optimization

ML can address both challenges:

- Model-driven optimization can improve search efficiency:
 - Eg. Bayesian Optimization (BO) more efficient than genetic algorithms
 - surrogate guides search by modeling the **Objective**
 - But each acquisition is slow



- Surrogate models (e.g. NN) can accelerate slow simulations:
 - But how to train surrogate models? Pre-training on random simulations eats-away at simulation savings.



Physics sim → NN surrogate

Application to dynamic/momentum aperture

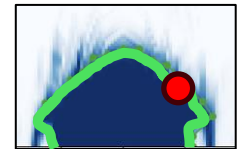
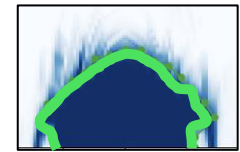
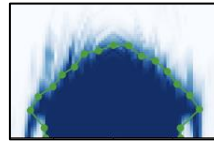
multipointBAX combines both advantages:

Standard BO: model and find optimal point in black box function $f(x)$

BAX: model $f(x)$, and find optimal output of $A[f(x)]$, for known algorithm $A[\cdot]$

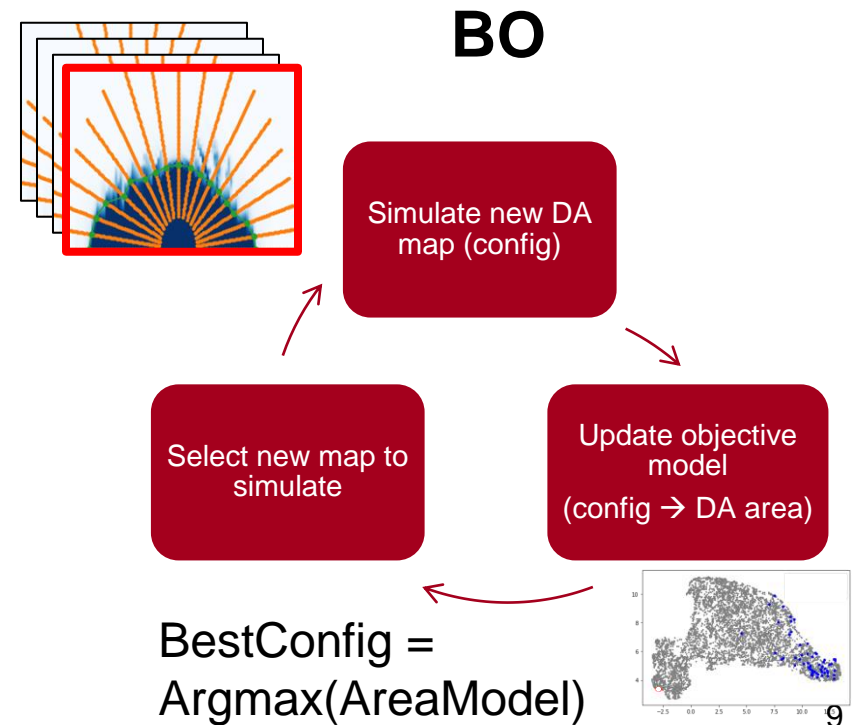
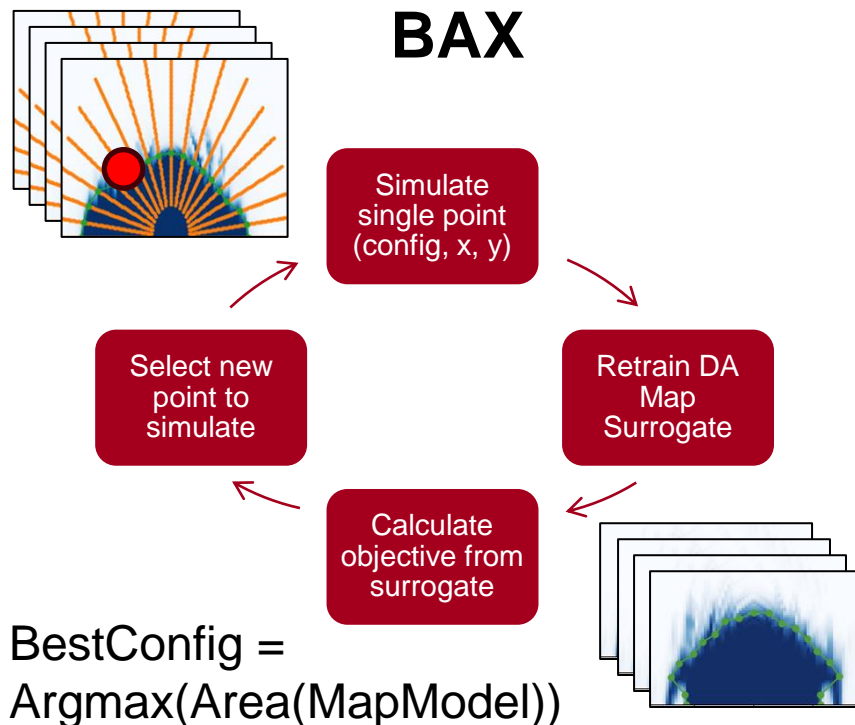
DAMA example:

- Model full DA/MA maps $f(x)$
- Algorithm, $A[\cdot]$, outputs area of surviving particles from $f(x)$
- Simulate new particle that has the biggest impact on output of $A[\cdot]$



Application to dynamic/momentum aperture

- MultipointBAX for DA:
 - Each acquisition is a single point (1 config, 1 particle)
 - Surrogate model predicts the DA map
 - Algorithm calculates the blue area of the map
 - BAX acquisition: picks the most informative acquisition point



Test problem: Ring design for SSRL-X

- Tuning 6 sextuple families

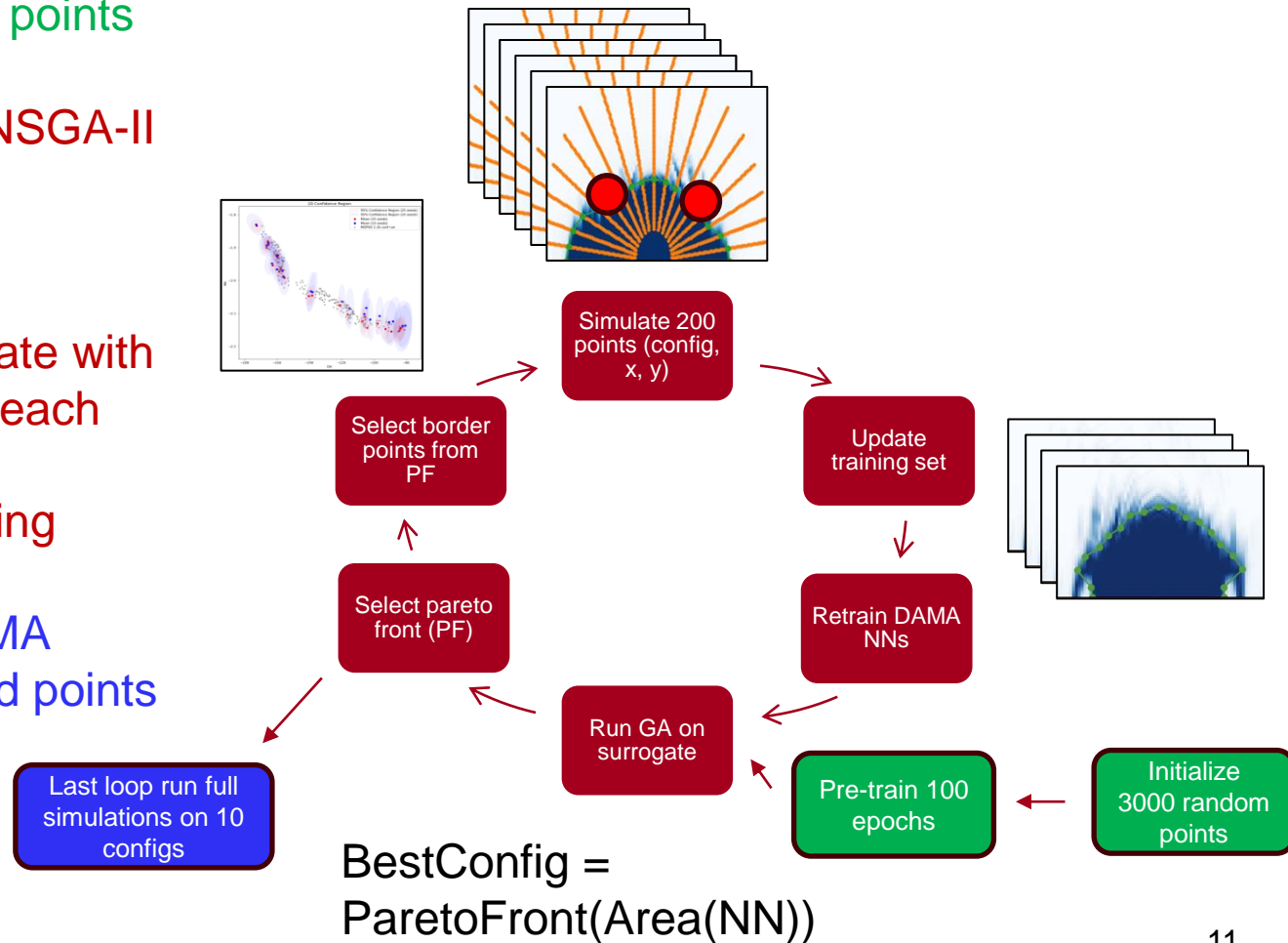
Traditional approach

- Parallel particle tracking to evaluate DA and MA
 - 1.3k particle tracking for DA
 - 1.0k particle tracking for MA
- MOPSO:
 - 200+ pop/generation
 - 100+ generations
 - 3-4M+ trackings for both DA and MA

SSRL-X example

Full BAX procedure

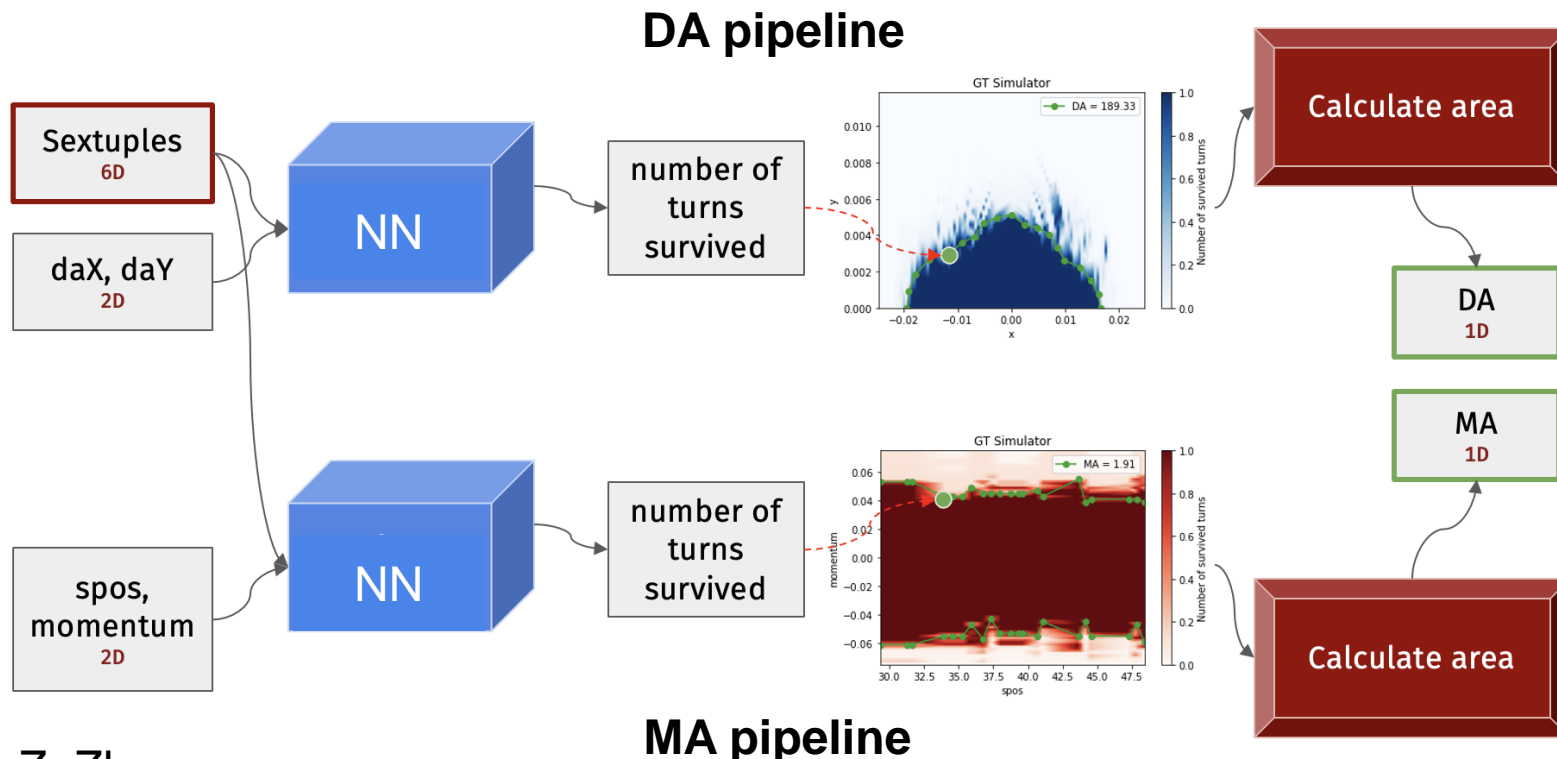
1. Initialize NN with 3000 points from random configs
2. Run 100 generations NSGA-II on NN
3. Select Pareto front
4. Select simulate 200 configs/points to simulate with random error seed for each point
5. Run 10 epochs retraining
6. Return to step 2
7. Final loop run full DA/MA simulations on selected points



SSRL-X example

NN architecture:

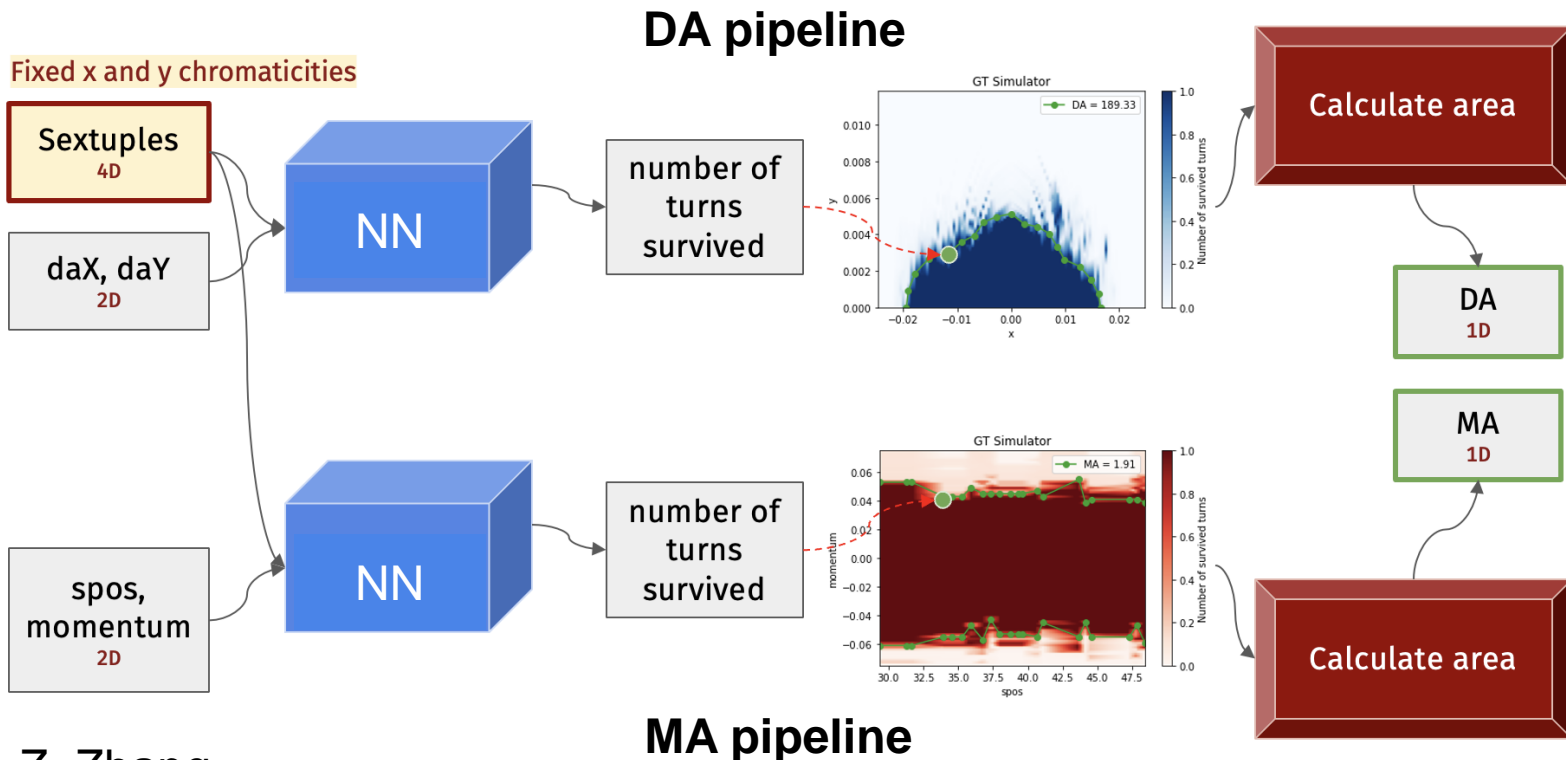
- Neural field: config + coordinate \rightarrow # of turns survived
- Moderate size: 6 layers, ~ 8 M parameters, not heavily optimized



SSRL-X example

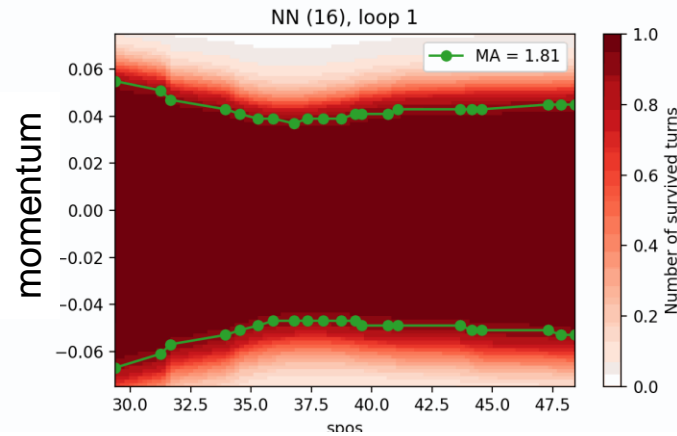
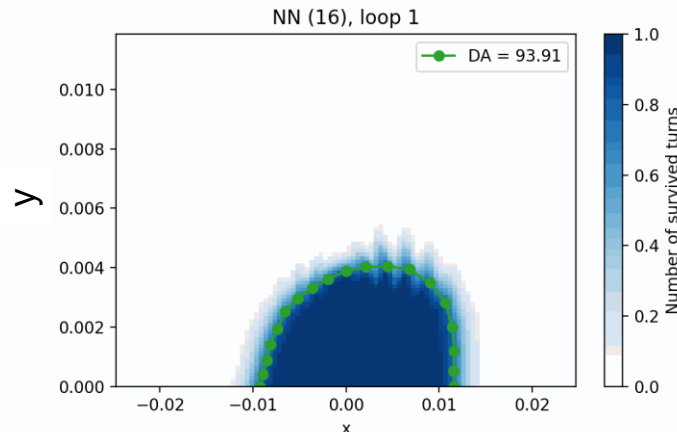
NN architecture:

- Neural field: config + coordinate \rightarrow # of turns survived
- Moderate size: 6 layers, ~ 8 M parameters, not heavily optimized

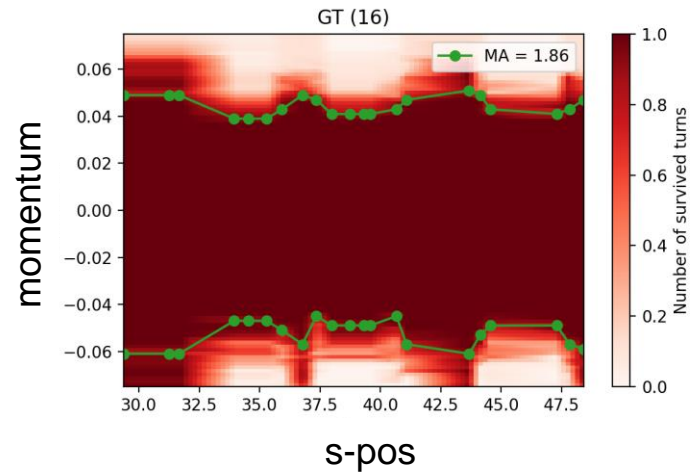
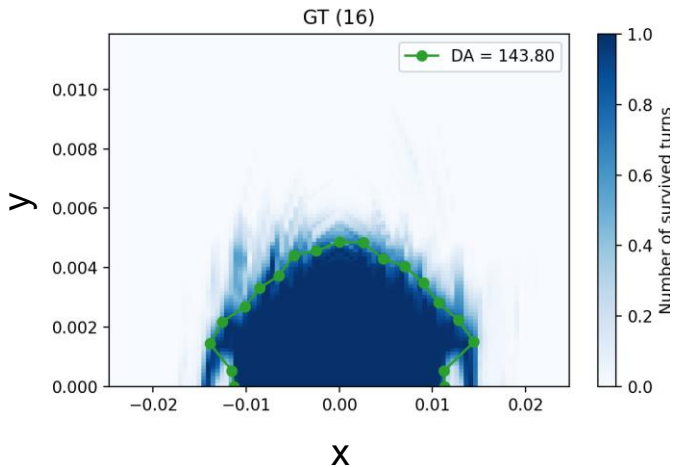


Evolution of the NN prediction vs. BAX loop

NN prediction



GT Simulation



SSRL-X example

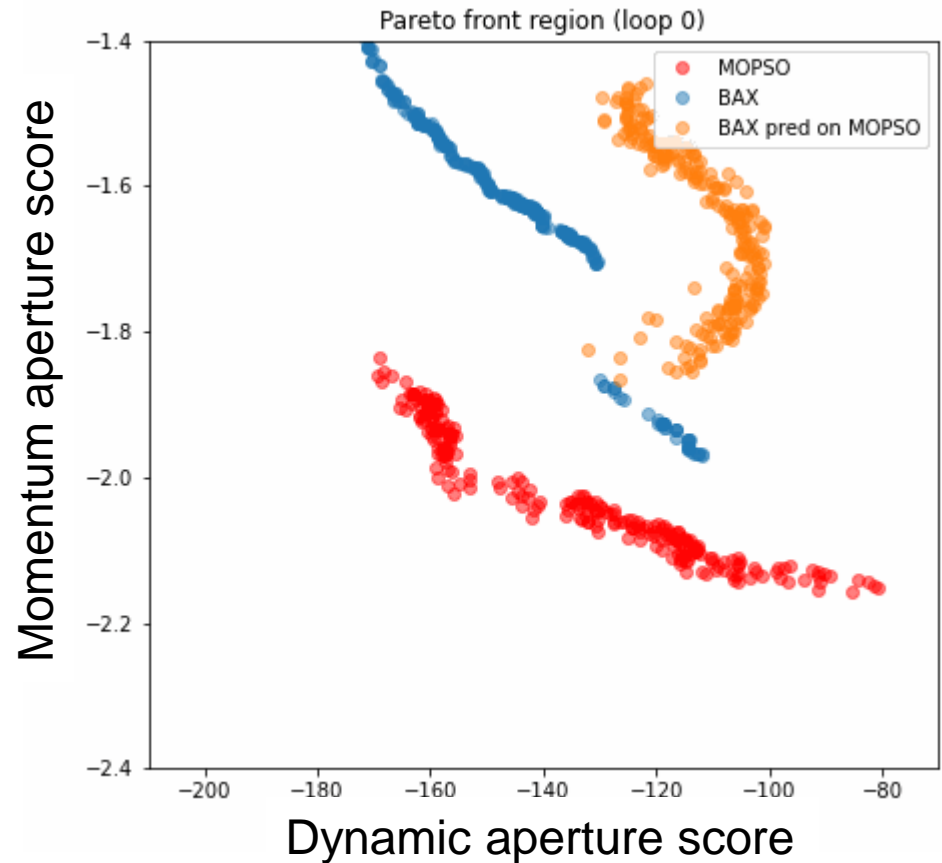
Our goal: optimize Pareto-front hyper volume:

Red: MOPSO baseline run

Blue: BAX model prediction

Yellow: BAX model on MOPSO simulations

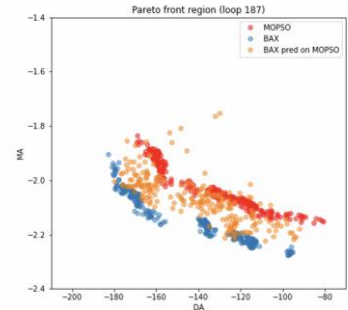
Yellow predictions give sense of model accuracy... but need full simulation to confirm



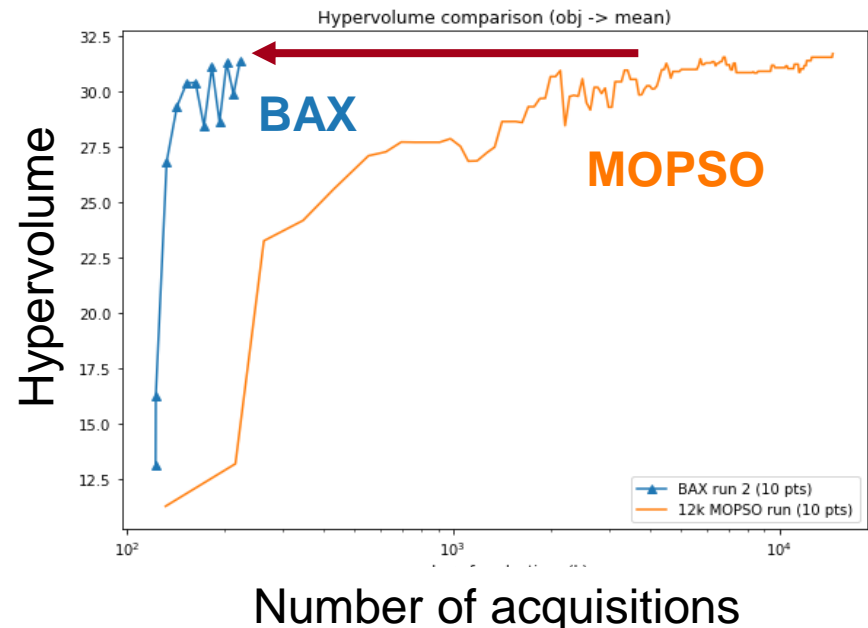
BAX vs. MOPSO Hypervolume

- Select 10 configurations on PF
- Run full simulations (10 error seeds) for each configuration
- Calculate hypervolume

Cost **includes** final simulation!



20x speedup



Conclusion

Current status:

- Proof-of-principle study on 4-dimension optimization at SSRL-X
- Factor of 20x speed-up compared to MOPSO

Next steps:

- Shift NN from predicting mean to distribution
- Introduce robustness, e.g. minimizing sensitivity to error seeds
- Tackle more challenging problems, e.g. FCCee?

Thanks for listening!

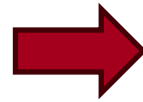
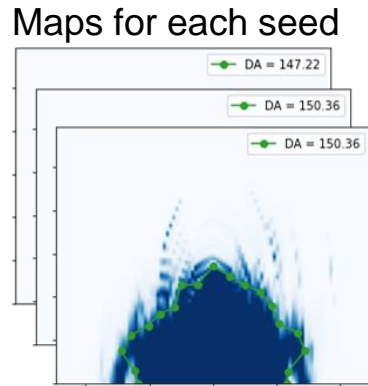
Work completed by Zhe Zhang (main author)

Thanks to Ilya Agapov, Sathya Chitturi, Sean Gasiorowski, Thorsten Hellert, Xiaobiao Huang, and Willie Neiswanger for helpful discussions

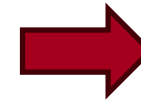
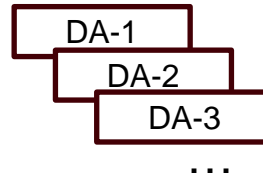
Next steps

Scoring: mean(objective) vs. objective(mean map)

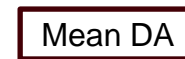
MOPSO score



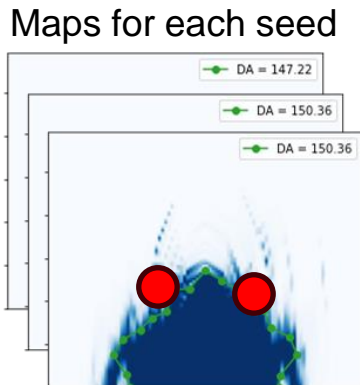
Area for each seed



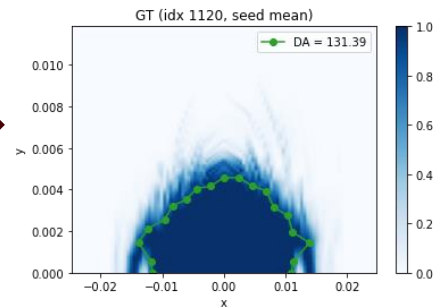
Final score



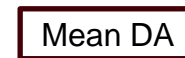
BAX Score



Average map



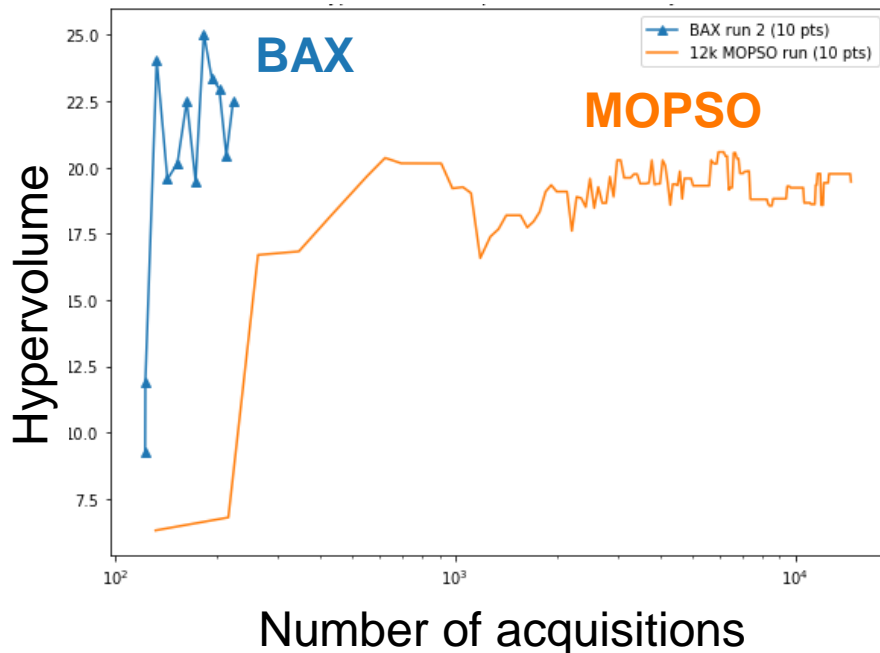
Final score



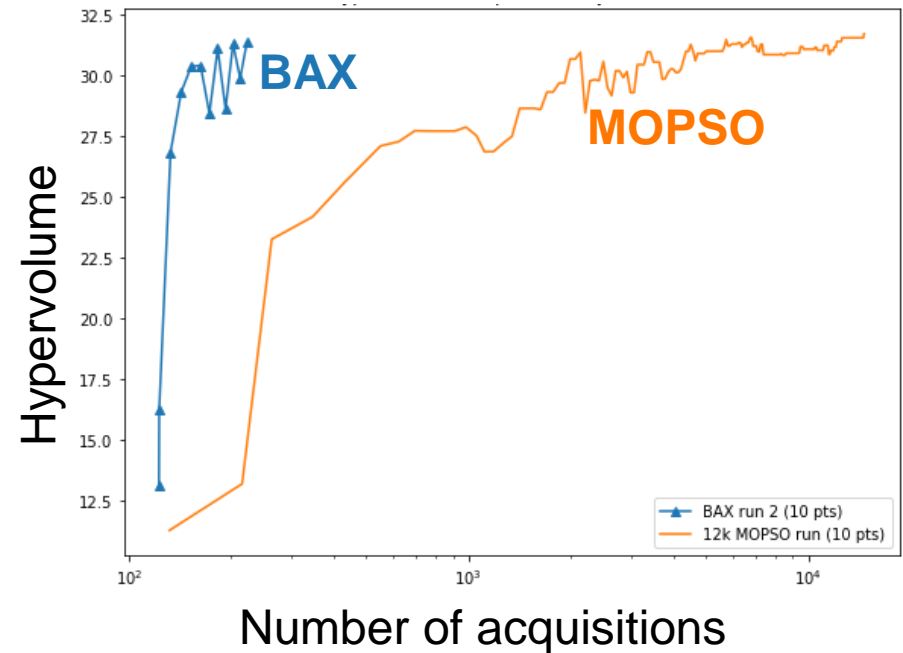
Scoring: mean(objective) vs. objective(mean map)

- Better solution found when using the BAX algorithm objective
 - ➔ matching the true objective will improve results
- Requires NN to predict map distribution, not mean map

Using BAX Score

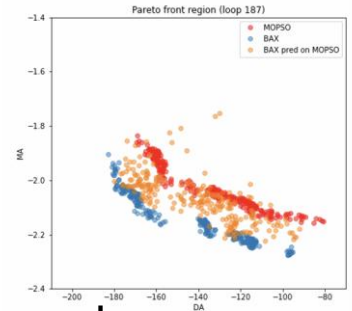


Using MOPSO Score



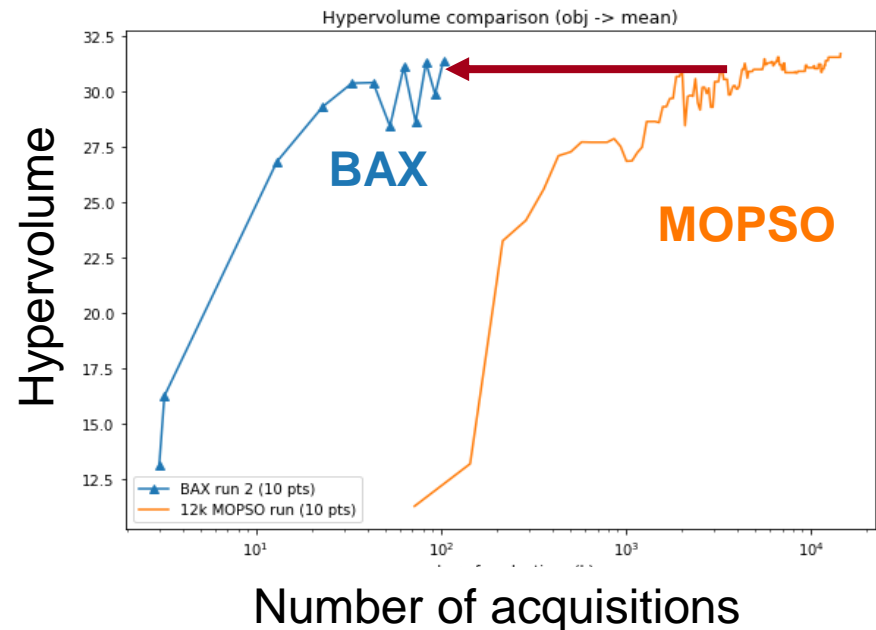
BAX vs. MOPSO Hypervolume

- Select 10 configurations on PF
- Run full simulations (10 error seeds) for each configuration
- Calculate hypervolume



40x speedup

Cost **without** final simulation!



Application to dynamic/momentum aperture

Two ways to estimate advantage of BAX:

- BO models 1 scalar for each 1000 simulations → up to **1000x gains**
- BO simulations are redundant, i.e. could predict map with fewer sims → more than **10x gains.**
- → Expect somewhere 10x to 1000x gain in efficiency from BAX

