# ONLINE ACCELERATOR TUNING WITH ADAPTIVE BAYESIAN OPTIMIZATION

**NIKITA KUKLEV** (ADVANCED PHOTON SOURCE, ANL, USA)
Yine Sun
Michael Borland
Hairong Shang
Greg Fystro

**THXD4**
NAPAC 2022
August 11, 2022

# ML AND ACCELERATORS

Accelerator performance requirements **continue to increase**

- Tighter tolerances
- New physics
- Fast reconfiguration

Machine learning (ML) methods a promising solution

- Anomaly detection/prediction
- Optimization
- Fast heuristics (surrogate models)

# ACCELERATOR OPTIMIZATION

Many approaches with various tradeoffs

Humans

Classic methods

Artificial Intelligence

Machine Learning

Deep Learning

Nelder-Mead Simplex
RCDS
Extremum Seeking

Genetic Algorithm
Particle Swarm

Bayesian Optimization

Reinforcement Learning
Deep Kernel Learning

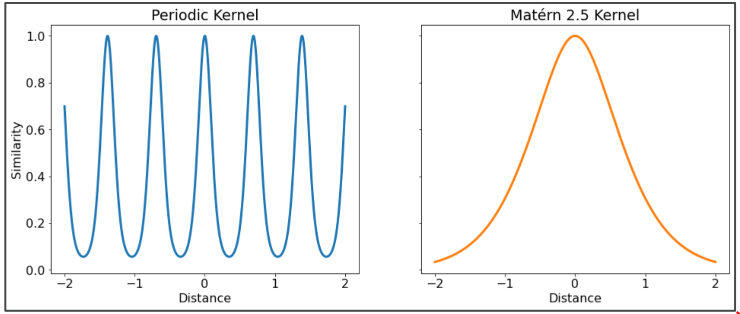
THIS TALK

# BAYESIAN OPTIMIZATION

INPUTS

$\mathbf{x}$

MACHINE BLACK BOX




$$\mathbf{y} = f(\mathbf{x}) + \varepsilon$$

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

GAUSSIAN PROCESS (GP) MODEL

$$p(\mathbf{f} \mid \mathbf{y}, \mathbf{x})$$

$$\mathrm{UCB}(\mathbf{x}) = \mu(\mathbf{x}) + \sqrt{\beta} * \sigma(\mathbf{x})$$

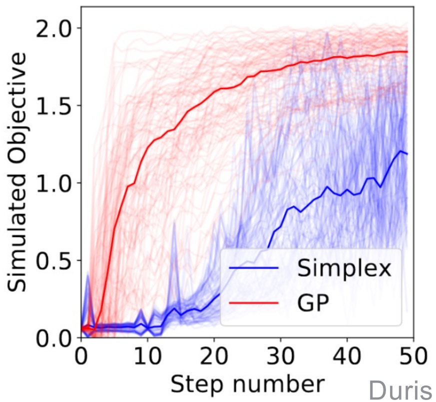
MAXIMIZE ACQUISITION FUNCTION

# BAYESIAN OPTIMIZATION

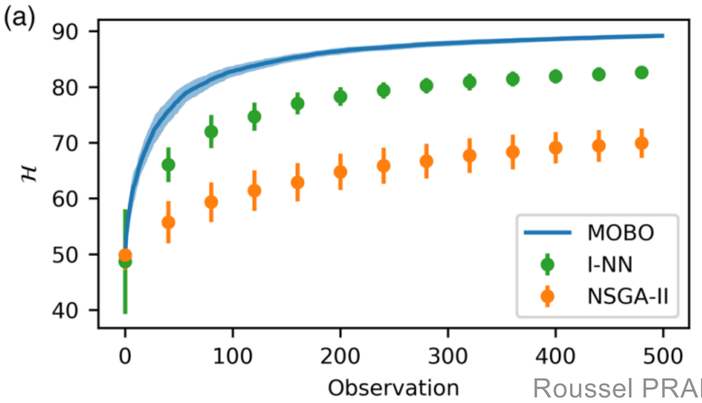
**Bayesian optimization** (BO) a promising method for ***expensive problems***

- Model-based and can encode expert knowledge
- Interpretable and scalable

Previous work showed good performance in *time-invariant* tasks




Duris PRL 2020

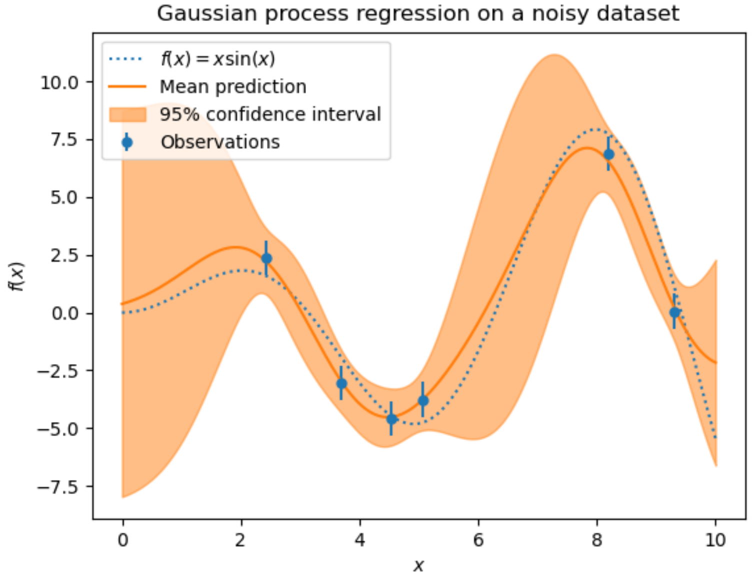



Roussel PRAB 2021

# ADAPTIVE BAYESIAN OPTIMIZATION

Many accelerators have **time-dependent performance: *f(t,x)***

- External factors (temperature, etc.)
- Device drift / degradation




A challenge for conventional BO

- Without time model, **drift appears as noise**
- Convergence to average *suboptimal* state
- Common solution – run local optimizer after BO
- Can drift be modelled explicitly?

# ADAPTIVE BAYESIAN OPTIMIZATION KERNEL

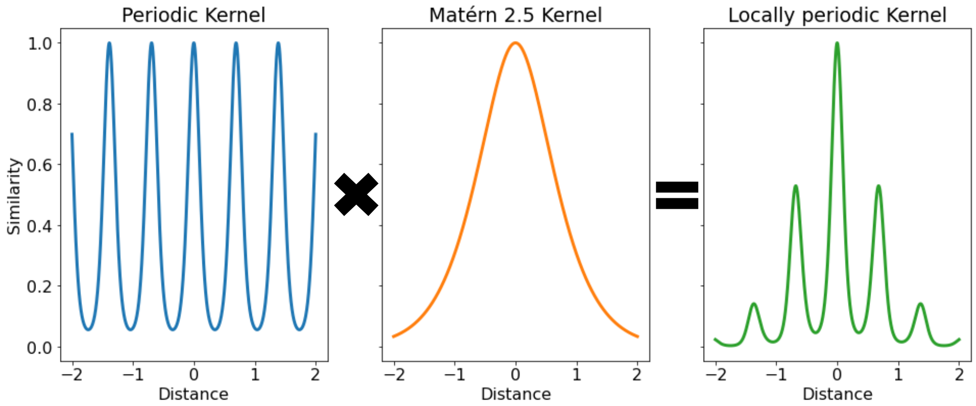
Consider time as another 'input' - has **very different properties**

- Periodic on various timescales (minutes, hours, days)
- Overall linear/polynomial trends

Can compose sub-kernels along any subspace - **what is the right one?**

K1 + K2 = LOGICAL **OR**

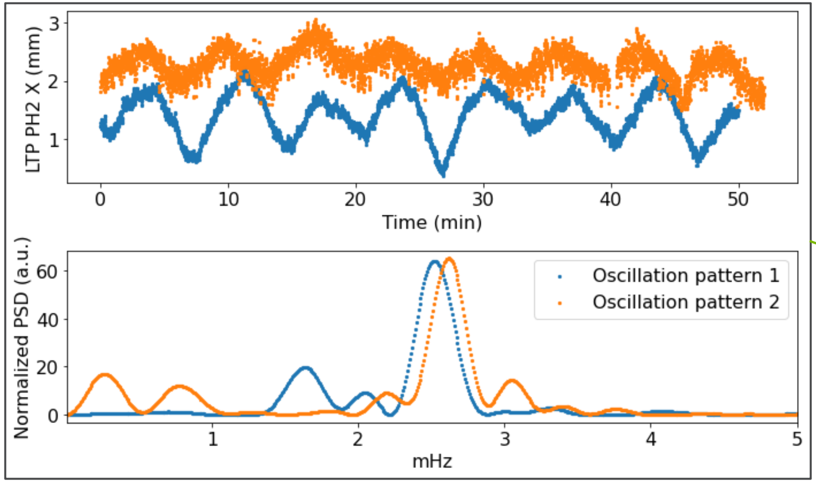
K1 * K2 = LOGICAL **AND**

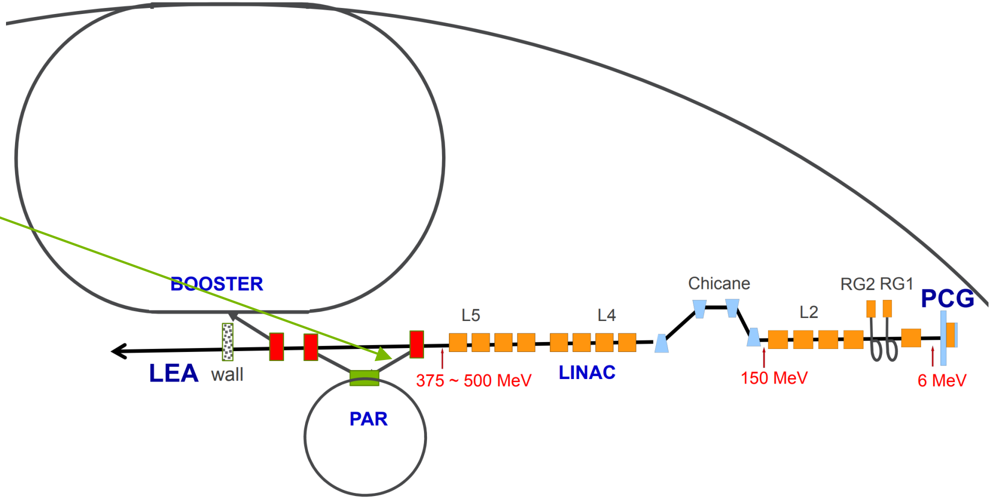


# ADAPTIVE ML MOTIVATION @ APS

APS injector supplies beam to storage ring and linac extension area

- **Proportional feedback** used to compensate drifts but has **high jitter**
- Drift spectrum varies day to day - requires **time-aware** and **time-adaptive** control



# ADAPTIVE BAYESIAN OPTIMIZATION KERNEL

ABO choice: **Spectral Mixture Kernel**

- Express as spectral density of several Gaussians
- Can approximate **any** stationary kernel
- Starting point for **Deep Kernel Learning**

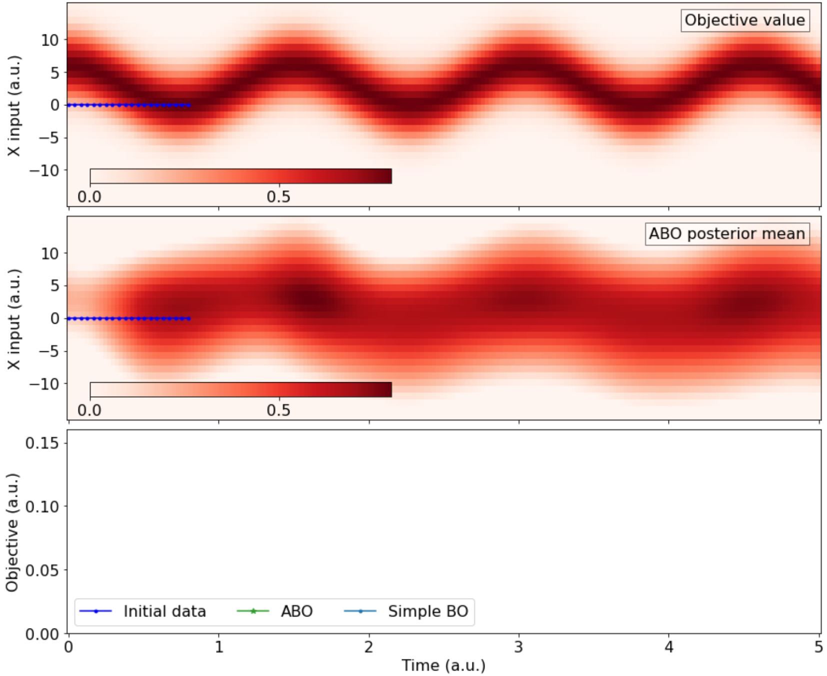
$$k(\tau) = \int S(s) e^{2\pi i s^T \tau} ds,$$

$$S(s) = \sum_{i=1}^{Q} w_i^2 \left[ \mathcal{N}(s|\mu_i, \sigma_i^2) + \mathcal{N}(s|-\mu_i, \sigma_i^2) \right].$$

**Final model** $k_{ABO}(t, t', x, x') = (k_{SM}(t, t') + k_l(t, t')) \times \sigma^2 k_{Mt}(\mathbf{x}, \mathbf{x}')$

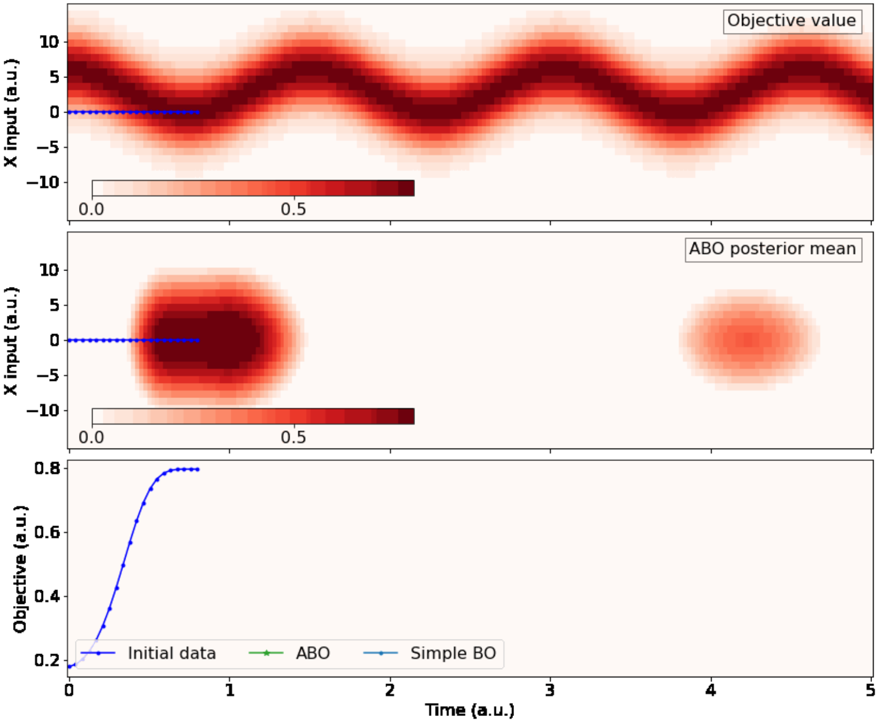
# RESULTS - SIMULATION

Example: sinusoidal signal with noise + initial steady state sampling



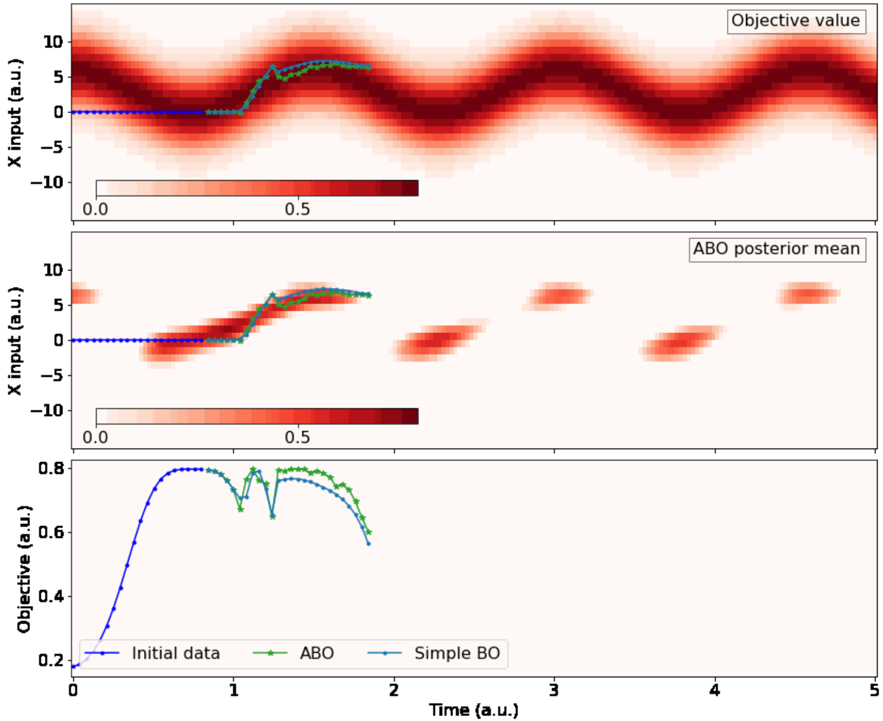
# RESULTS - SIMULATION

Example: sinusoidal signal with noise + initial steady state sampling



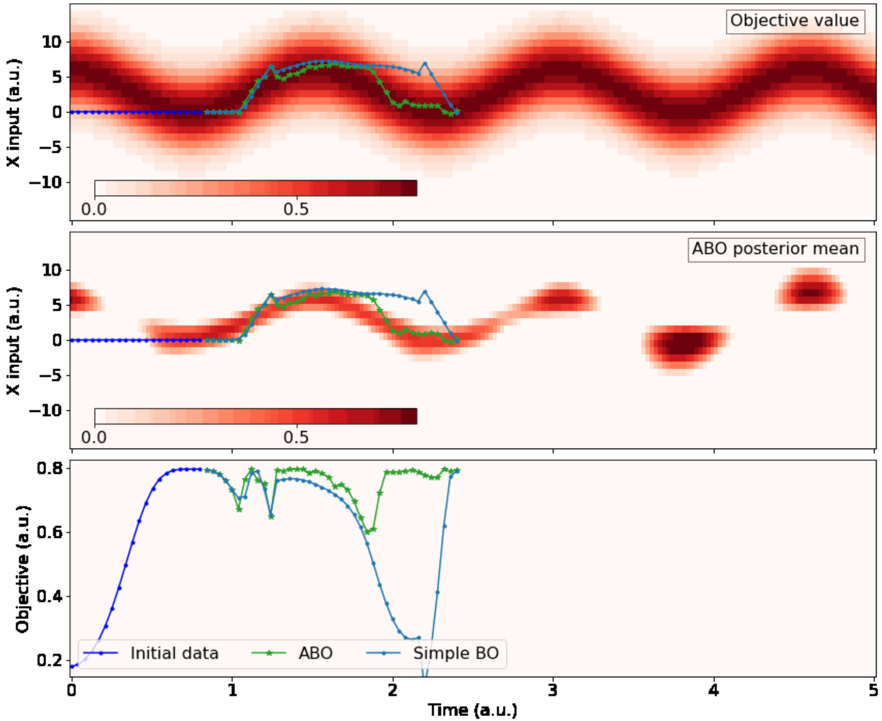
# RESULTS - SIMULATION

Example: sinusoidal signal with noise + initial steady state sampling



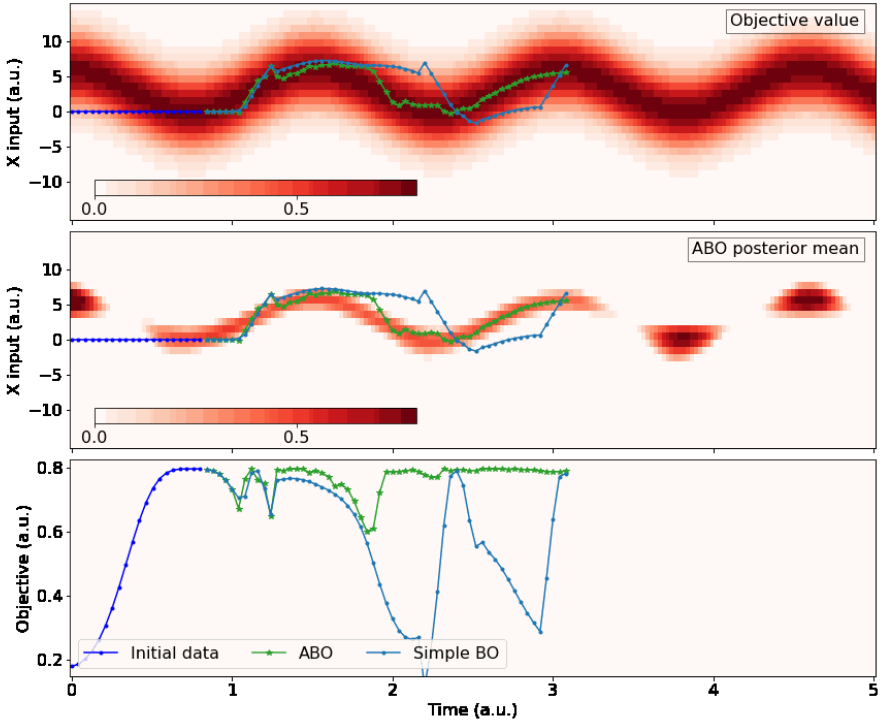
# RESULTS - SIMULATION

Example: sinusoidal signal with noise + initial steady state sampling



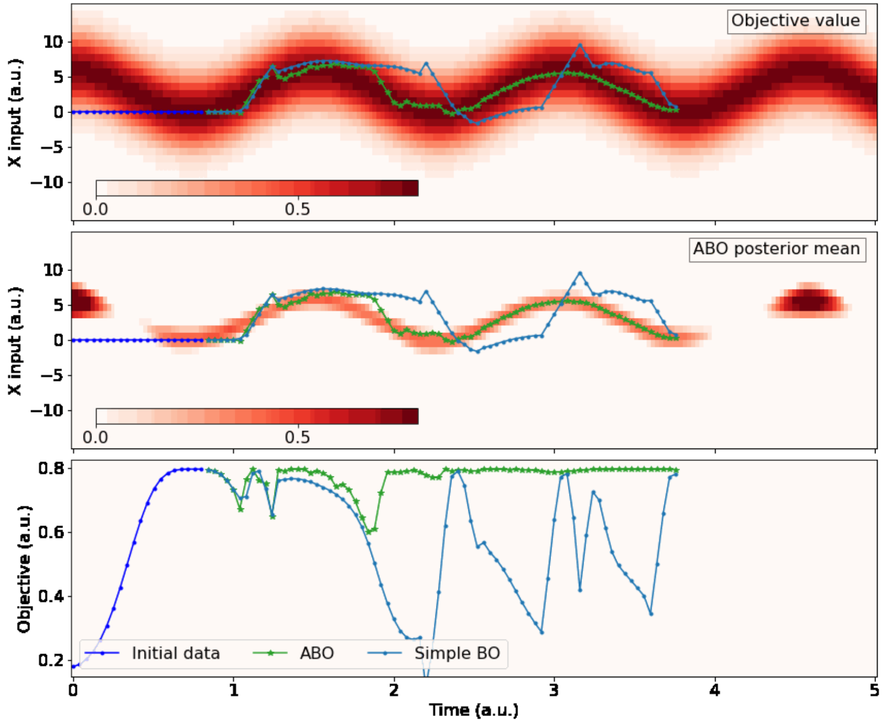
# RESULTS - SIMULATION

Example: sinusoidal signal with noise + initial steady state sampling



# RESULTS - SIMULATION

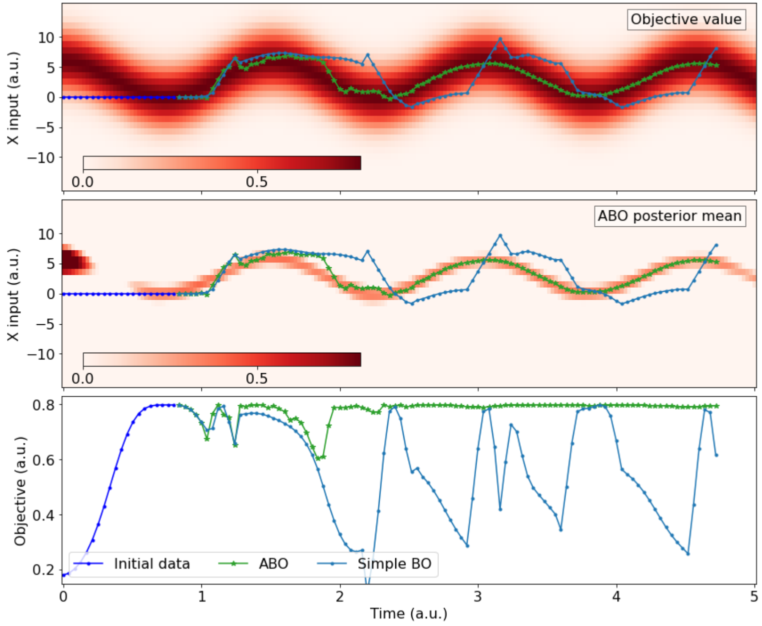
Example: sinusoidal signal with noise + initial steady state sampling

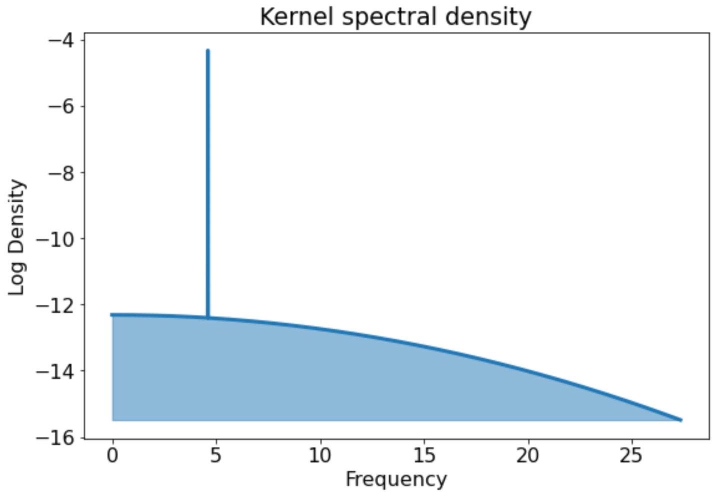


# RESULTS - SIMULATION

ABO finds correct oscillation within 1 period

- Kernel density reflects broad noise + oscillation frequency



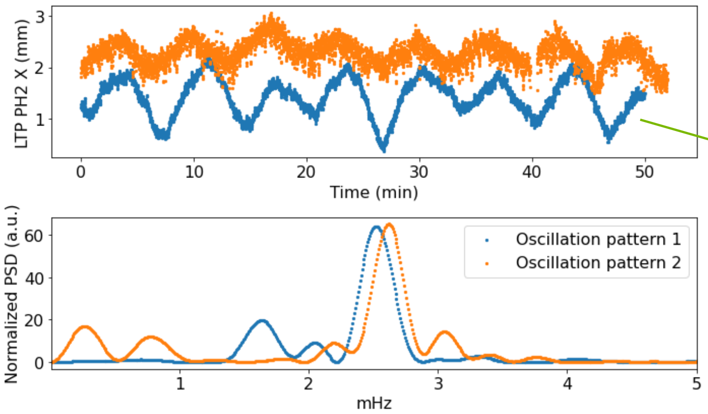



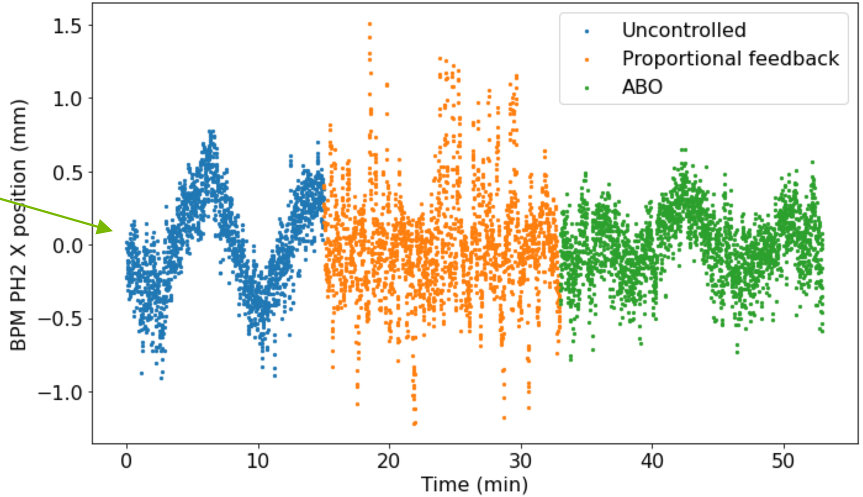
# CONTINUOUS USE CONSIDERATIONS

| | Performance | Safety |
|---|---|---|
| | Needs to run faster than drift rate<br>Poor complexity scaling ($N^3$) | Must respect constraints<br>Must behave conservatively |
| **Considered** | Old data cutoff<br>Sparse GP<br>Scalable GPU methods | Constrained acquisition functions |
| **ABO** | Time-biased bandpass importance subsampling | Constraint auxiliary models<br>Slew rate + hard limits<br>Proximity bias |

See TUPA24 for related digital twin model work

# RESULTS - EXPERIMENT




Several tests in APS linac – trajectory MSE objective

- 10s cycle limit
- Train with last 20 minutes of history

Overall **2x jitter** improvement (0.21/0.36/0.33)!

# OPERATIONAL IMPLEMENTATION

We are developing libraries to work with operational APS systems

- **APSopt** – optimizer algorithms + SDDS toolkit command line interface
- **pySDDS** – native SDDS format reader/writer
- **pybeamtools** – soft IOC, surrogate model, and archiver interface

Can make end-to-end virtual accelerators, test with real data, and deploy operationally

# CONCLUSION

**Adaptive ML** key in enabling long-term operational use

**ABO** a good adaptive optimizer when:

- Both optimization and stabilization are desired
- Drift likely explained by (unknown) model

Future work:

- Better use of extra inputs (i.e. air temperature) and historic data – we have 20 years!
- Integration with local optimization methods
- Transition to full deep kernel learning

# THANK YOU!