



U.S. DEPARTMENT
of ENERGY

Blop: Beamline Optimization Package

Thomas Hopkins (thopkins1@bnl.gov)

Max Rakitin (mrakitin@bnl.gov)

2025/11/06

X f i @BrookhavenLab

What is Blop?

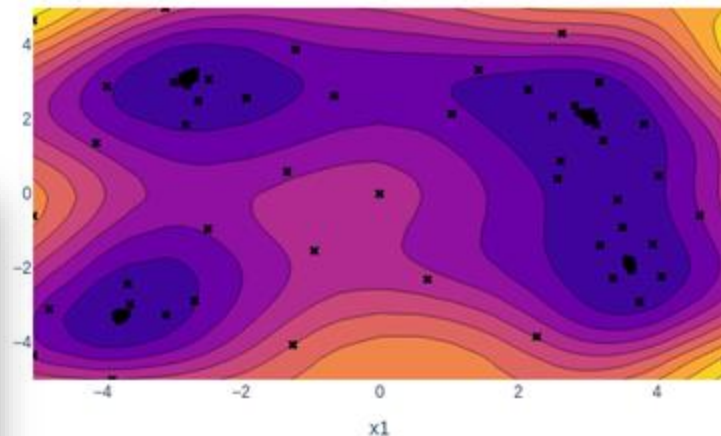
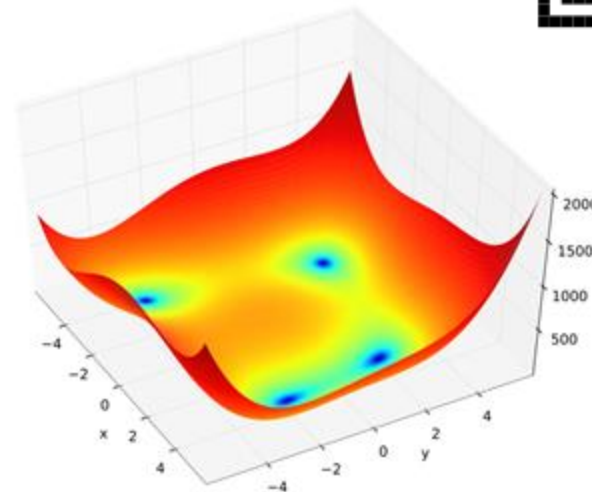
An open-source Python library for optimization using Bluesky

- A bridge between optimization algorithms and Bluesky
- Simple, practical interface for data-driven optimization
- Tools for beamline optimization problems

<https://github.com/NSLS-II/blop>

<https://nsls-ii.github.io/blop/>

<https://pypi.org/project/blop>

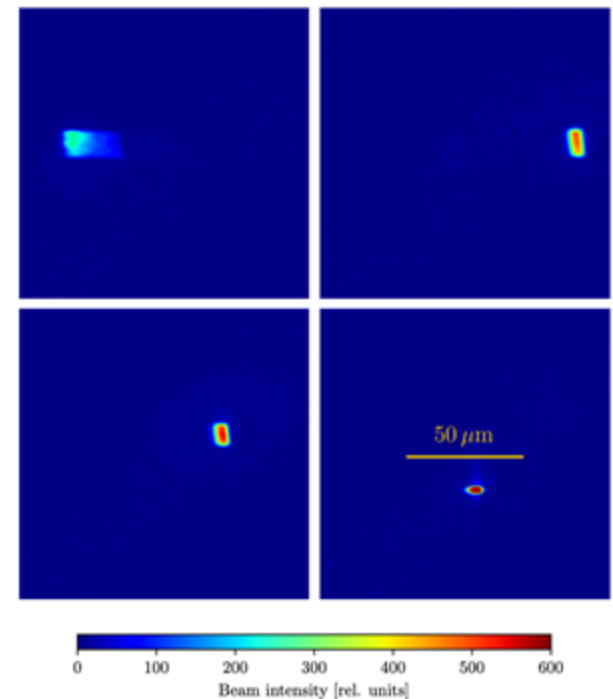
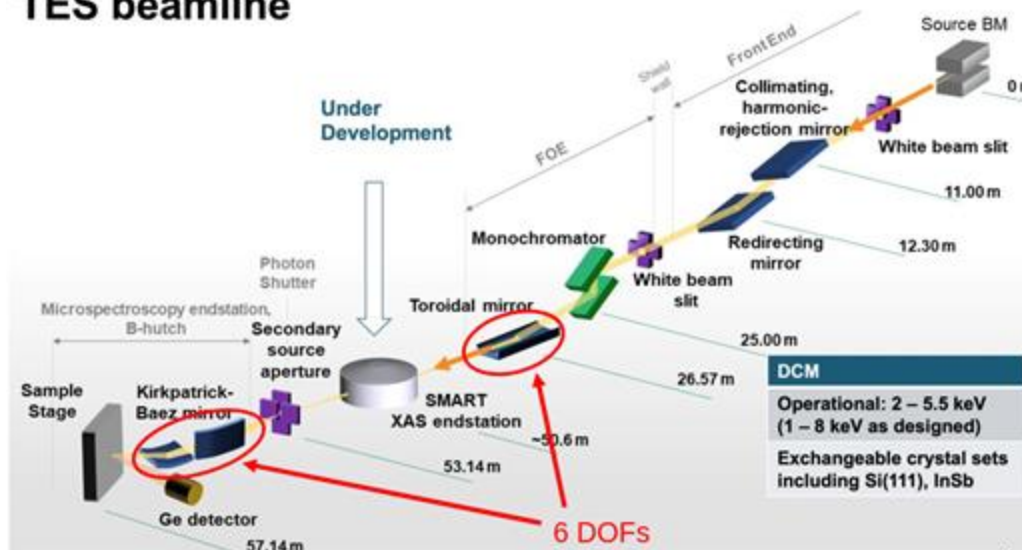


<https://doi.org/10.1107/S1600577524008993>

Some History...

- **2020:** differential evolution for optimization at NSLS-II's TES and in Sirepo
- **2021-24:** Bayesian optimization and real beamline experiments
- **2025:** package refactor and software development, XRT support

TES beamline



Development Team



Thomas
Morris



Thomas
Hopkins



Jennefer
Maldonado



Jessica
Moylan



Abigail
Giles



Roman
Chernikov

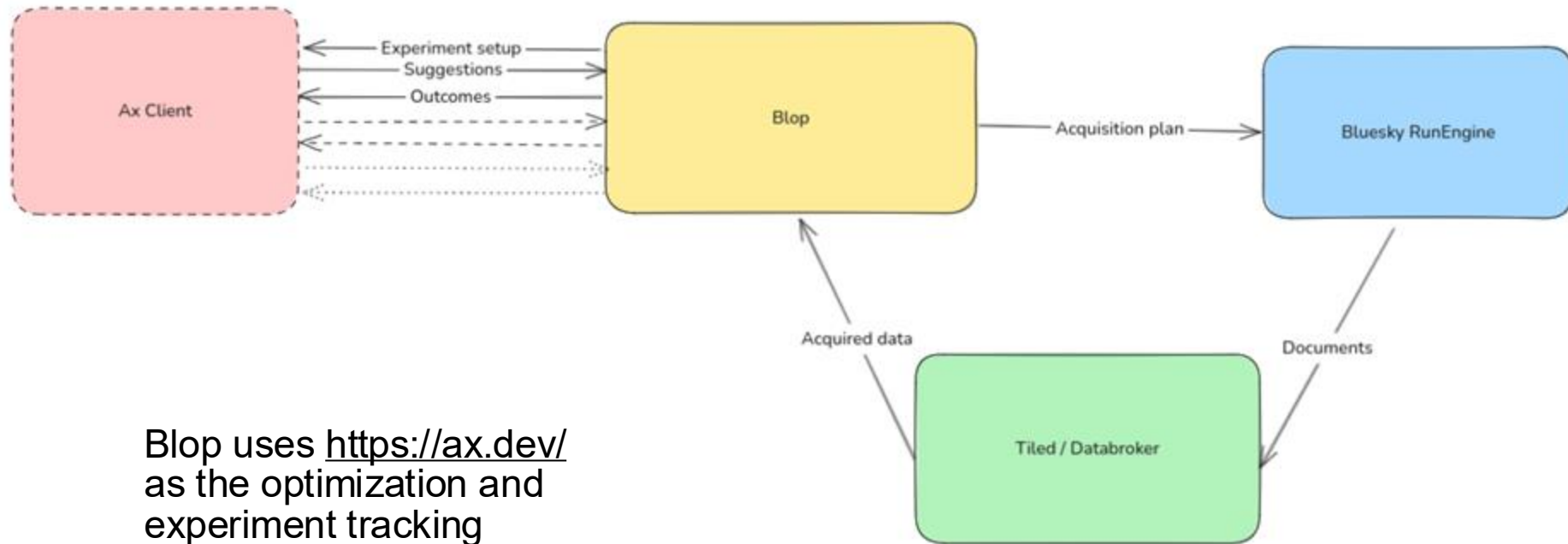


Phil
Maffettone

Acknowledgements

- Thanks to NSLS-II, ALS, LCLS, BESSY-II Staff for the interest
- ILLUMINE - Intelligent Learning for Light Source and Neutron Source User Measurements Including Navigation and Experiment Steering (LAB 23-3030, 2023-2028)
- BNL's LDRD-22-031 project "Simulation-aided Instrument Optimization using Artificial Intelligence and Machine Learning Methods" (2021-2023)
- DOE SBIR project (award No. DE-SC00020593, 2020-2023)
- DOE SULI Internship Program (2020)

Core Architecture (v0.8.0)



Blop uses <https://ax.dev/> as the optimization and experiment tracking backend

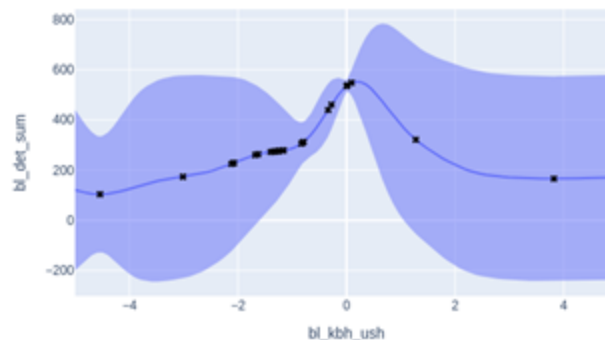
No strict Ophyd dependency!

Project Goals

- Make beamline optimization **practical**
- Integrate with the full **Bluesky** ecosystem
- Enable **flexible** choice of optimization algorithms
- Solve **real beamline** optimization problems

Practical Beamline Optimization

- Safety
- High performance
- Human-in-the-loop control
- Experiment tracking



trial_index	arm_name	trial_status	generation_node	bl_det_sum	bl_det_wid_x	bl_det_wid_y	bl_kbv_dsv	bl_kbh_dsh
0	0	0_0	COMPLETED	CenterOfSearchSpace	22139.112374	16.301732	41.147863	40000.000000
1	1	1_0	COMPLETED	Sobol	477.833153	373.000000	79.500000	43163.089752
2	2	2_0	COMPLETED	Sobol	559.728522	74.965847	132.113333	30869.154837
3	3	3_0	COMPLETED	Sobol	5810.577930	285.713095	39.503152	35459.523015
4	4	4_0	COMPLETED	Sobol	861.663585	170.595238	109.275000	48026.437219
5	5	5_0	COMPLETED	MBM	22641.851151	86.500641	20.667179	38244.515736
6	6	6_0	COMPLETED	MBM	23721.790561	14.193780	18.323182	38039.083619
7	7	7_0	COMPLETED	MBM	22445.220181	105.261905	21.618578	38090.557765
8	8	8_0	COMPLETED	MBM	23714.594206	16.357143	20.070909	38311.940065
9	9	9_0	COMPLETED	MBM	23757.491811	15.827273	19.123182	38145.243855
10	10	10_0	COMPLETED	MBM	23665.904300	14.506667	18.979798	38230.932182
11	11	11_0	COMPLETED	MBM	23764.704280	13.776786	18.525455	38335.309712
12	12	12_0	COMPLETED	MBM	23824.781969	13.932619	18.300000	38147.871212
13	13	13_0	COMPLETED	MBM	23698.656408	14.678571	19.181818	38263.930406
14	14	14_0	COMPLETED	MBM	23756.129665	15.346154	19.110000	38096.511708

Initial sample

Random exploration

Bayesian optimization

Integrate with Bluesky

- Optimization-focused plans, plan-stubs, callbacks, suspenders
- *bluesky-adaptive* support
- *bluesky-queueserver* support
- *tilde/databroker* support

Bluesky protocols
(*ophyd[-async]* support)



```
x1 = MovableSignal("x1", initial_value=0.1)
x2 = MovableSignal("x2", initial_value=0.23)
det = ReadableSignal("det")

dofs = [
    DOF(movable=x1, search_domain=(-5, 5)),
    DOF(movable=x2, search_domain=(-5, 5)),
]
objectives = [
    Objective(name="himmelblau_2d", target="min"),
]
readables = [det]
```

[5] ✓ 0.0s

```
agent = Agent(readables=readables, dofs=dofs, objectives=objectives, db=tilde_client, digestion=himmelblau_2d_digestion)
agent.configure_experiment(name="simple_experiment", description="A simple experiment.")
RE(agent.learn(iterations=100))
```

Flexible Optimizer Choice

- Ax-backend custom “generation strategies”
- Protocols for optimization-focused Bluesky plans
- Standard generators support (adopted by *Xopt*, *optimas*, *libEnsemble*, and *rsopt*)



Beamline Optimization Problems

- XRT simulations of varying complexity
- Benchmarking custom kernels, models, algorithms, etc.
- **Deploy at NSLS-II beamlines**

