Tutorial:
Multi-Objective Bayesian Optimization with BoTorch and Ax

2ND WORKSHOP ON AI FOR THE ELECTRON ION COLLIDER (AI4EIC)

OCTOBER 10, 2022

BoTorch Ax

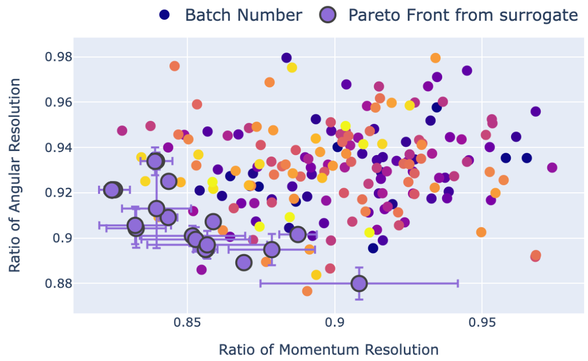
Maximilian Balandat

Adaptive Experimentation team, Meta

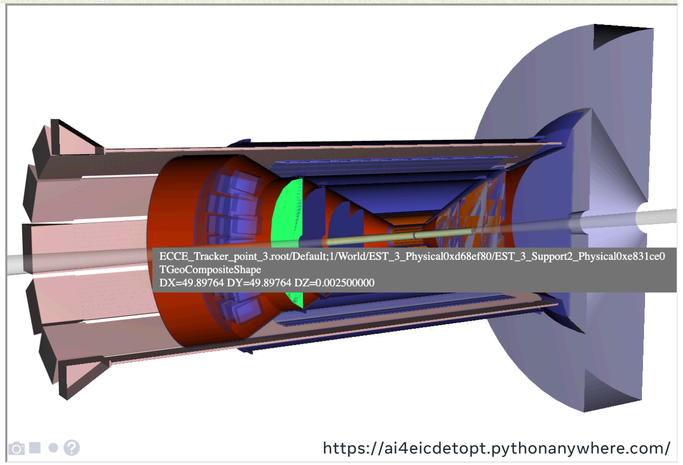
Motivation: Accelerator Detector Design

- Domain: 11+ parameters (location of tracking layers, etc.)
- Multiple objectives:
  - resolution (momentum, polar angular)
  - efficiency
  - cost
  - ...
- Constraints on geometry (to avoid overlap)
- Evaluating objectives requires **computationally expensive** simulation for each design point (parameter configuration)

Multi Objective Bayesian Optimization



GEANT4 Visualization of the design

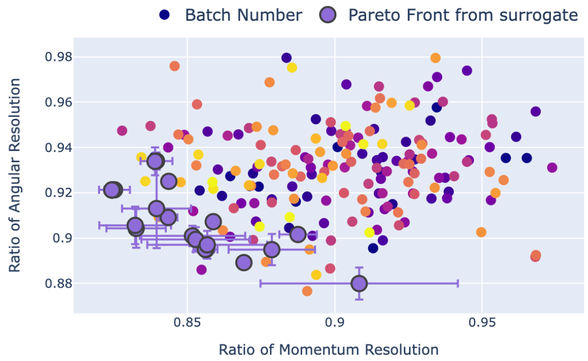


Adkins et al. Design of the ECCE Detector for the Electron Ion Collider. ArXiv 2022.

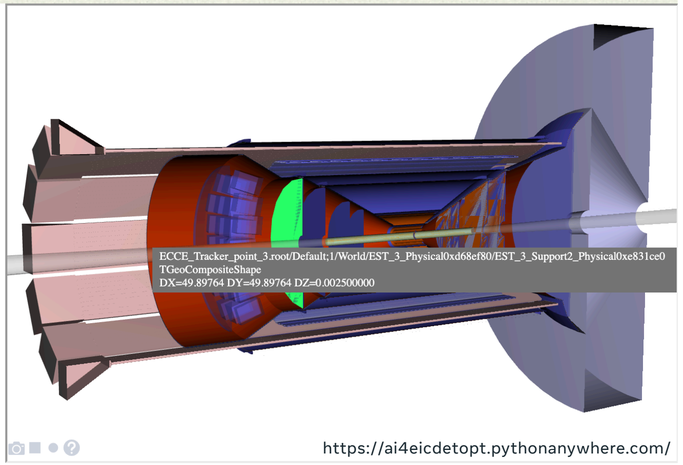
Motivation: Accelerator Detector Design

- Domain: 11+ parameters (location of tracking layers, etc.)
- Multiple objectives:
  - resolution (momentum, polar angular)
  - efficiency
  - cost
  - ...
- Constraints on geometry (to avoid overlap)
- Evaluating objectives requires **computationally expensive** simulation for each design point (parameter configuration)

Multi Objective Bayesian Optimization



GEANT4 Visualization of the design



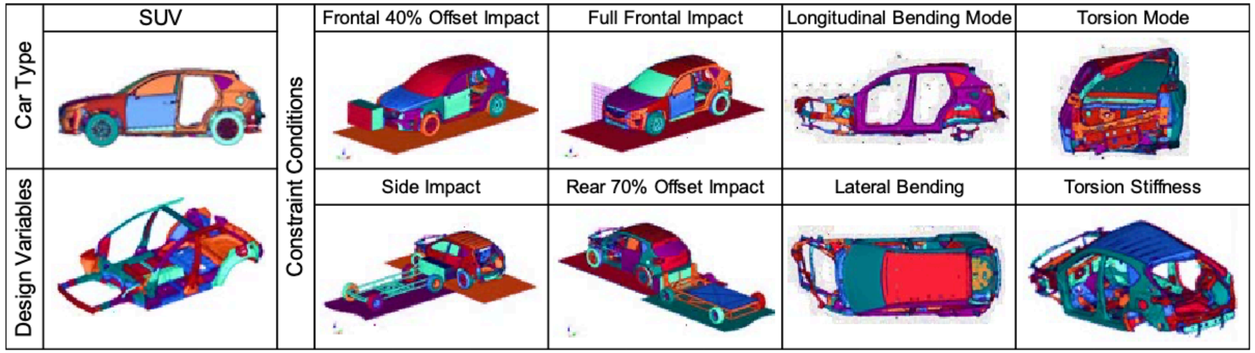
Goal: Explore a complex design space to optimally trade off multiple competing objectives that are computationally expensive-to-evaluate.

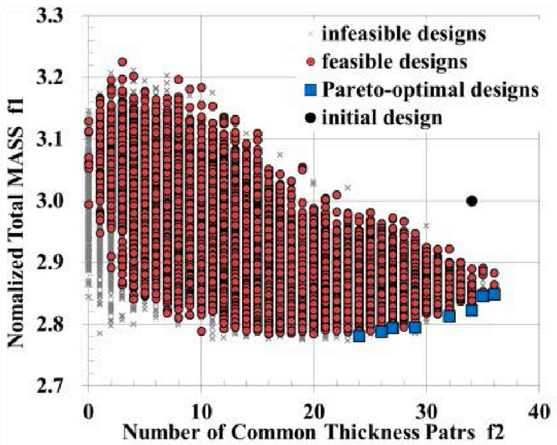
Adkins et al. Design of the ECCE Detector for the Electron Ion Collider. ArXiv 2022.

Motivation: Vehicle Design Optimization

- Domain: **222 parameters** (properties of structural parts)
- Goals:
  - **minimize** total vehicle mass of three vehicles (Mazda 3, 6, CX-5)
  - **maximize** number of parts shared across vehicles
  - satisfy **54 black-box outcome constraints** (performance & safety requirements)
- Evaluation requires solving a complex physics simulation
- Originally solved on the (at the time) world's fastest super computer in **>3,000 CPU years**

→ **56 outcomes**





T. Kohira et al. Proposal of benchmark problem based on real-world car structure design optimization. GECCO 2018.

Hot Take:

Every optimization problem is fundamentally a multi-objective optimization problem.

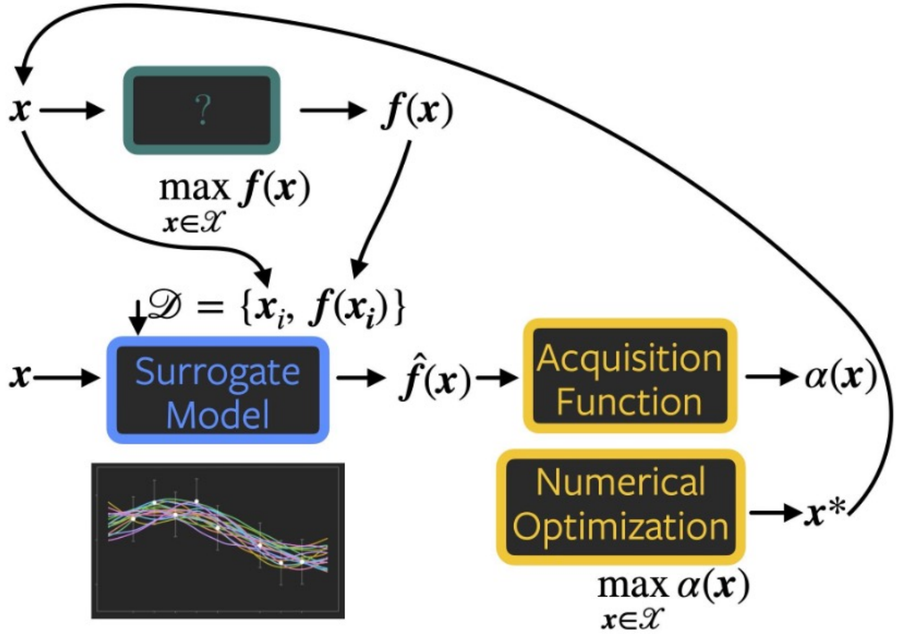
Outline

1. Motivation & Intro
2. Background: Methodology
3. Applications
4. Tutorial: Hands-On Examples
5. Advanced Features

2 Background

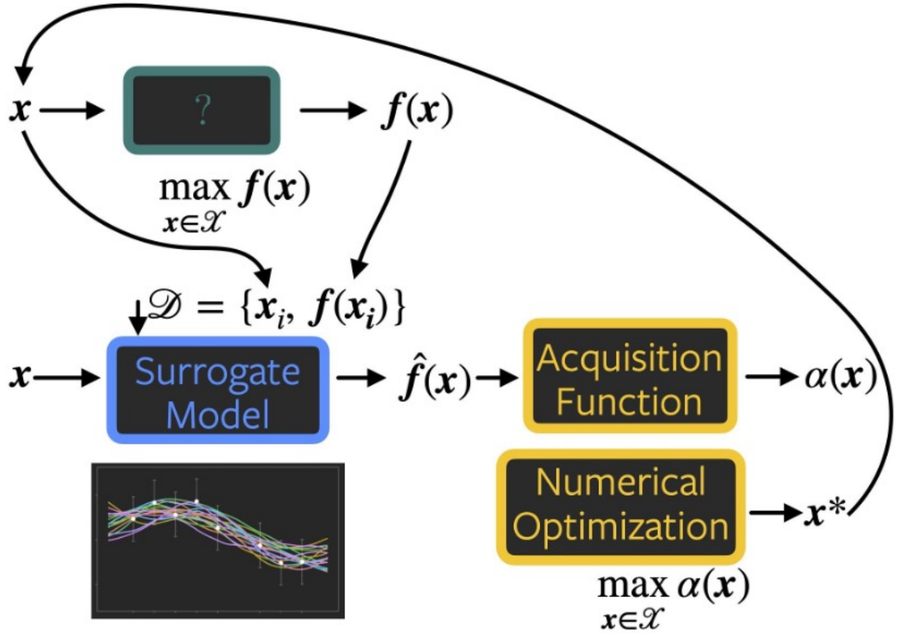
Bayesian Optimization (BO)

- Goal: Sample-efficient optimization of “black-box” function: $\max_{x \in \mathbb{X}} f(x)$, where f is expensive to evaluate, w/o gradients
- Strategy: Leverage
  - a probabilistic **surrogate model** $\hat{f}$
  - an **acquisition function** $x \mapsto \alpha(\hat{f}(x))$ to guide exploration



Bayesian Optimization (BO)

- Goal: Sample-efficient optimization of “black-box” function: $\max_{x \in \mathbb{X}} f(x)$, where f is expensive to evaluate, w/o gradients
- Strategy: Leverage
  - a probabilistic **surrogate model** $\hat{f}$
  - an **acquisition function** $x \mapsto \alpha(\hat{f}(x))$ to guide exploration



Typical surrogate: Gaussian Process (GP)

- Defines a distribution over functions:

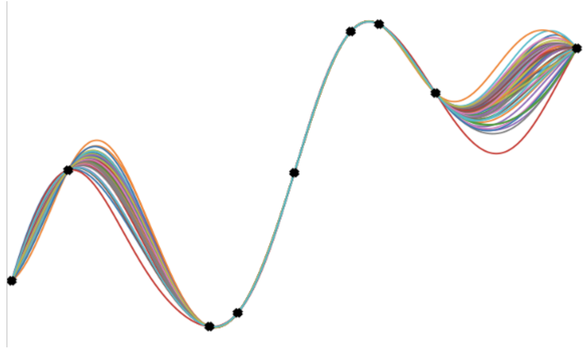
$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$$

- Covariance function, e.g.

$$k_{SE}(x, x') = s^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{d}\left(\frac{x_i - x_i'}{\ell_i}\right)^2\right)$$

lengthscale ℓ_i measures the distance for being uncorrelated along x_i

draws from the prior

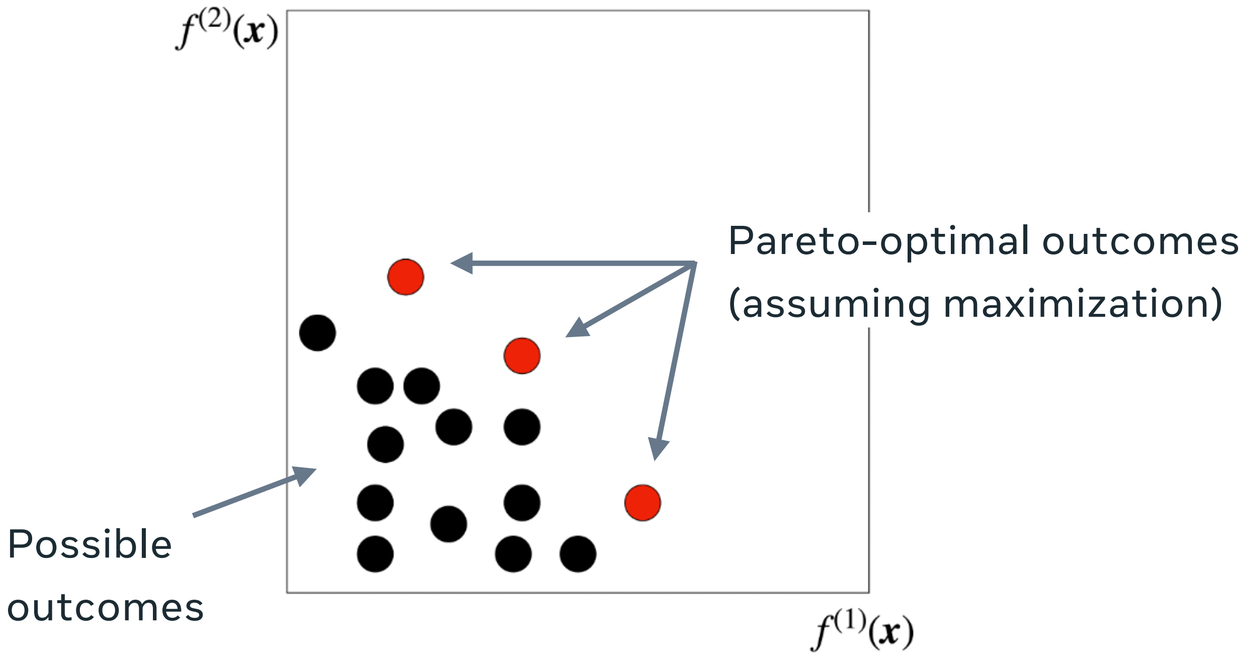


draws from the posterior

Multi-Objective Optimization (MOO)

- Goal: Optimize a **vector-valued black-box function** $f(x) = (f^{(1)}(x), \ldots, f^{(m)}(x))$.
  - Typically, no single best solution that simultaneously maximizes all objectives exists.
  - Instead, we identify the **Pareto frontier** of optimal trade-offs

- Various approaches:
  - Domain-specific heuristics
  - Evolutionary algorithms (e.g. NSGA-II)
  - Multi-Objective BO (MOBO)
    - via random scalarizations (e.g. ParEGO)
    - Information-theoretic acquisition functions
    - based on hypervolume improvement

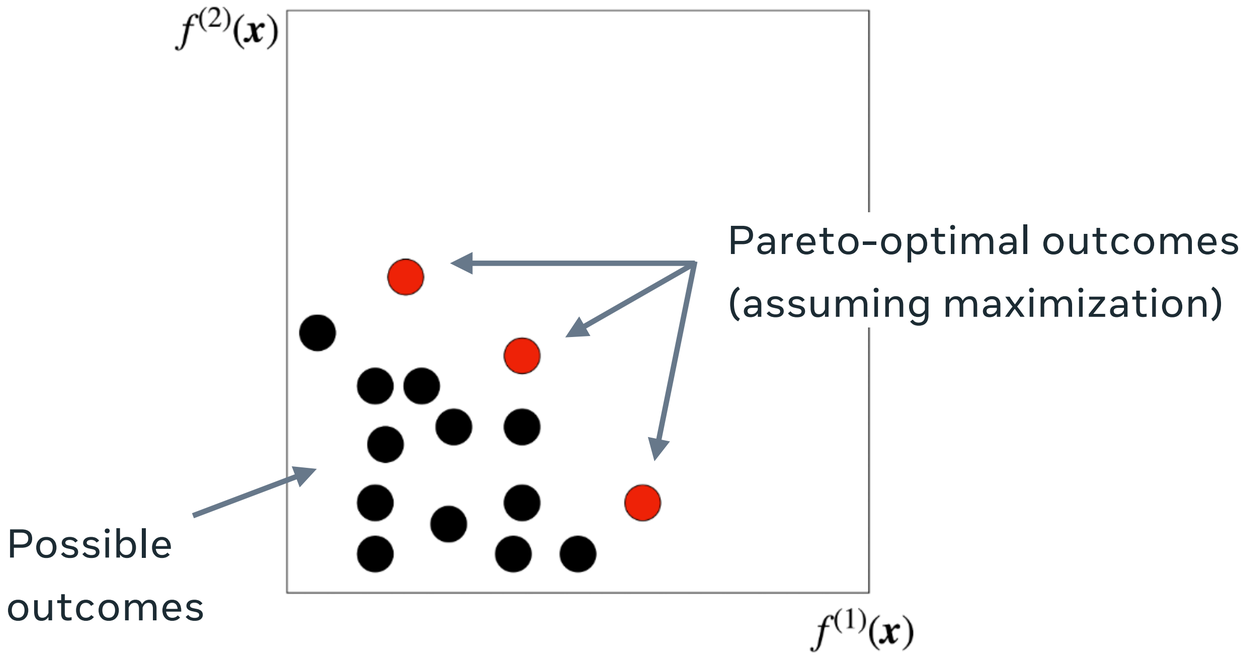


Multi-Objective Optimization (MOO)

- Goal: Optimize a **vector-valued black-box function** $f(x) = (f^{(1)}(x), \ldots, f^{(m)}(x))$.
  - Typically, no single best solution that simultaneously maximizes all objectives exists.
  - Instead, we identify the **Pareto frontier** of optimal trade-offs

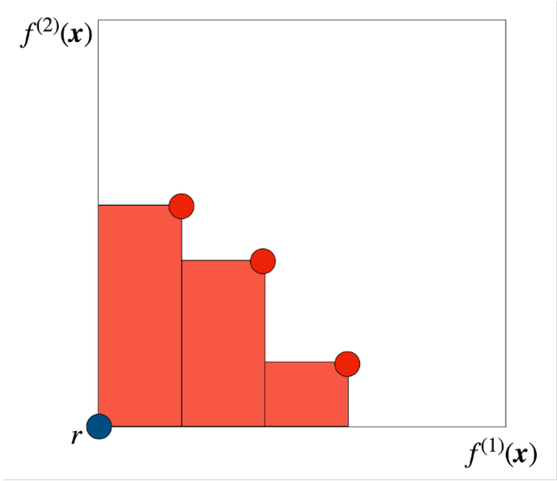
- Various approaches:
  - Domain-specific heuristics
  - Evolutionary algorithms (e.g. NSGA-II)
  - Multi-Objective BO (MOBO)
    - via random scalarizations (e.g. ParEGO)
    - Information-theoretic acquisition functions
    - based on hypervolume improvement

offers superior sample-efficiency



Expected Hypervolume Improvement

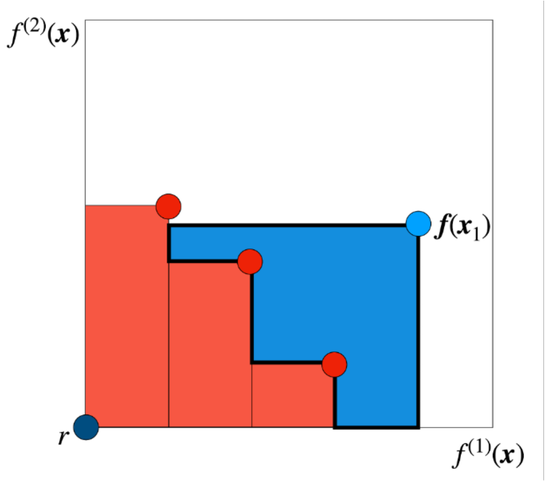
- **Hypervolume (HV)**: Measure of the quality of a Pareto frontier
  - Volume dominated by the Pareto frontier and bounded by a reference point r
  - Larger HV -> better Pareto frontier
- **Hypervolume improvement**: increase in HV from a new point or set of points



S. Daulton, M. Balandat, E. Bakshy. Differentiable Expected Hypervolume Improvement for parallel multi-objective Bayesian optimization. NeurIPS 2020.
S. Daulton, M. Balandat, E. Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. NeurIPS 2021.

Expected Hypervolume Improvement

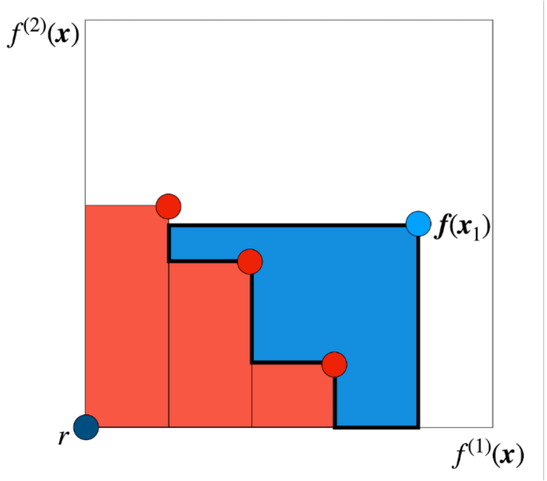
- **Hypervolume (HV)**: Measure of the quality of a Pareto frontier
  - Volume dominated by the Pareto frontier and bounded by a reference point r
  - Larger HV -> better Pareto frontier
- **Hypervolume improvement**: increase in HV from a new point or set of points



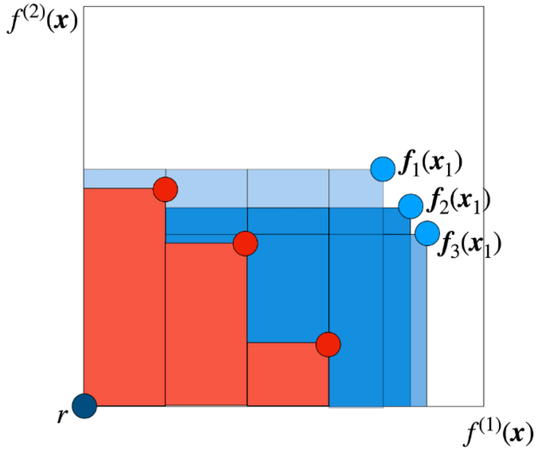
S. Daulton, M. Balandat, E. Bakshy. Differentiable Expected Hypervolume Improvement for parallel multi-objective Bayesian optimization. NeurIPS 2020.
S. Daulton, M. Balandat, E. Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. NeurIPS 2021.

Expected Hypervolume Improvement

- **Hypervolume (HV)**: Measure of the quality of a Pareto frontier
  - Volume dominated by the Pareto frontier and bounded by a reference point r
  - Larger HV -> better Pareto frontier
- **Hypervolume improvement**: increase in HV from a new point or set of points



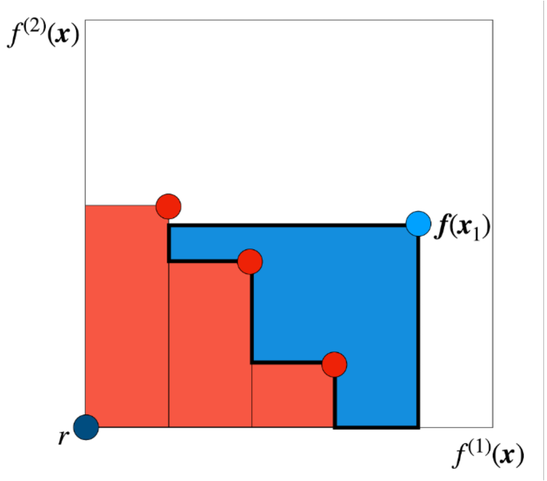
- **q-Expected Hypervolume Improvement (qEHVI)**
  - One-step Bayes-optimal acquisition function for HV criterion
  - Computes the expected increase in HV under the posterior distribution of the surrogate model



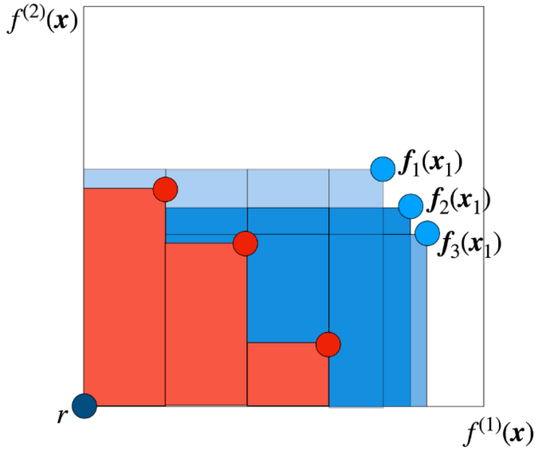
S. Daulton, M. Balandat, E. Bakshy. Differentiable Expected Hypervolume Improvement for parallel multi-objective Bayesian optimization. NeurIPS 2020.
S. Daulton, M. Balandat, E. Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. NeurIPS 2021.

Expected Hypervolume Improvement

- **Hypervolume (HV)**: Measure of the quality of a Pareto frontier
  - Volume dominated by the Pareto frontier and bounded by a reference point r
  - Larger HV -> better Pareto frontier
- **Hypervolume improvement**: increase in HV from a new point or set of points



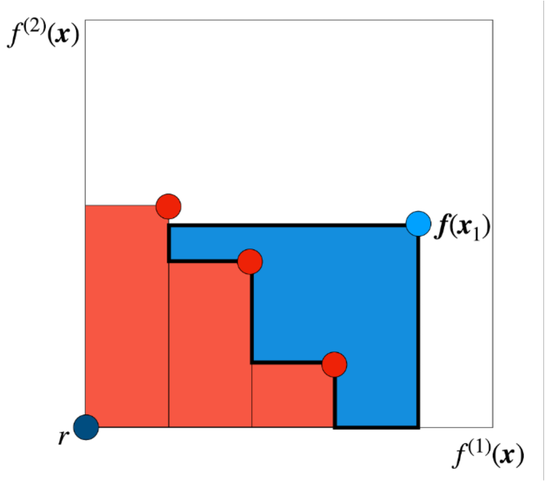
- **q-Expected Hypervolume Improvement (qEHVI)**
  - One-step Bayes-optimal acquisition function for HV criterion
  - Computes the expected increase in HV under the posterior distribution of the surrogate model
- **q-Noisy Expected Hypervolume Improvement (qNEHVI)**
  - Like qEHVI, but integrates over uncertainty about the value of noisy observations



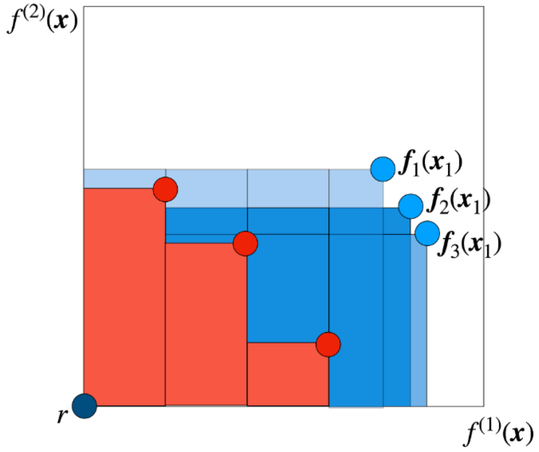
S. Daulton, M. Balandat, E. Bakshy. Differentiable Expected Hypervolume Improvement for parallel multi-objective Bayesian optimization. NeurIPS 2020.
S. Daulton, M. Balandat, E. Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. NeurIPS 2021.

Expected Hypervolume Improvement

- **Hypervolume (HV)**: Measure of the quality of a Pareto frontier
  - Volume dominated by the Pareto frontier and bounded by a reference point r
  - Larger HV -> better Pareto frontier
- **Hypervolume improvement**: increase in HV from a new point or set of points



- **q-Expected Hypervolume Improvement (qEHVI)**
  - One-step Bayes-optimal acquisition function for HV criterion
  - Computes the expected increase in HV under the posterior distribution of the surrogate model



- **q-Noisy Expected Hypervolume Improvement (qNEHVI)**
  - Like qEHVI, but integrates over uncertainty about the value of noisy observations

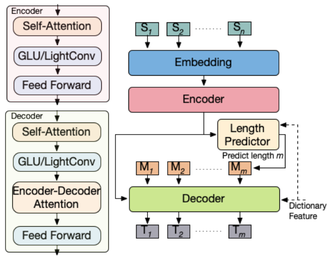
Default in Ax & BoTorch

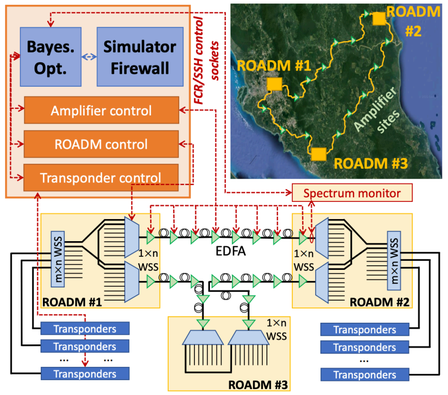
S. Daulton, M. Balandat, E. Bakshy. Differentiable Expected Hypervolume Improvement for parallel multi-objective Bayesian optimization. NeurIPS 2020.
S. Daulton, M. Balandat, E. Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. NeurIPS 2021.

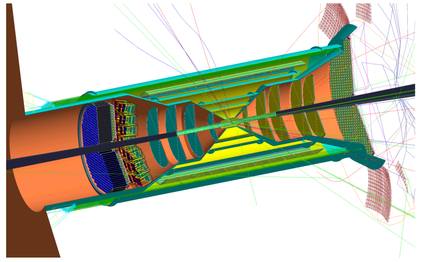
3 Applications

Application Areas (non-exhaustive)

- **AutoML**
  - e.g. Hyperparameter Optimization, Neural Architecture Search, Inference optimization
- **Infrastructure Optimization**
  - e.g. tuning parameters of large distributed systems, optimizing fiber backend networks
- Sequential Experimentation in large-scale **online A/B tests**
- **Chemistry & Material Science**
  - e.g. process optimization, Automated Chemical Design, material discovery
- **Engineering / Simulation Optimization**
  - e.g. design of optical systems for augmented reality (AR) see-through displays
- Health Sciences (Medicine, Neuroscience)
- …







Research built on Ax & BoTorch

ODBO: Bayesian Optimization with Search Space Prescreening for Directed Protein Evolution

Bayesian optimization of comprehensive two-dimensional liquid chromatography separations

Jim Boelrijk[a,d,*], Bob Pirok[a,b], Bernd Ensing[a,c], Patrick Forré[a,d]

[a] AI4Science Lab, University of Amsterdam, The Netherlands
[b] Analytical Chemistry Group, Van 't Hoff Institute for Molecular Sciences, University of Amsterdam, The Netherlands
[c] Computational Chemistry Group, Van 't Hoff Institute for Molecular Sciences, University of Amsterdam, The Netherlands
[d] AMLab, Informatics Institute, University of Amsterdam, The Netherlands

High-dimensional Automated Radiation Therapy Treatment Planning via Bayesian Optimization

Qingying Wang[1,2, a], Ruoxi Wang[1, a], Jiacheng Liu[1,2], Fan Jiang[1], Haizhen Yue[1], Yi Du[1], Hao Wu[1,2,b]

[1] Key Laboratory of Carcinogenesis and Translational Research (Ministry of Education/Beijing), Department of Radiation Oncology, Beijing Cancer Hospital & Institute, Beijing, China

[2] Institute of Medical Technology, Peking University Health Science Center

Characterization of Exercise-Induced Myocardium Growth Using Finite Element Modeling and Bayesian Optimization

Yiling Fan[1,2,3], Jaume Coll-Font[1,4,5], Maaike van den Boomen[1,4,5], Joan H. Kim[1], Shi Chen[1], Robert Alan Eder[1], Ellen T. Roche[2,3,5*] and Christopher T. Nguyen[1,4,5*]

[1] Cardiovascular Bioengineering and Imaging Laboratory, Cardiology Division, Massachusetts General Hospital, Charlestown, MA, United States, [2] Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, United States, [3] Institute for Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA, United States, [4] Athinoula A. Martinos Center for Biomedical Imaging, Charlestown, MA, United States, [5] Harvard Medical School, Boston, MA, United States

GAUCHE: A Library for Gaussian Processes in Chemistry

Ryan-Rhys Griffiths[*1] Leo Klarner[*2] Henry B. Moss[*3] Aditya Ravuri[*1] Sang Truong[*4] Bojana Rankovic[*5] Yuanqi Du[*6] Arian Jamasb[1] Julius Schwartz[1] Austin Tripp[1] Gregory Kell[7] Anthony Bourached[8] Alex J. Chan[1] Jacob Moss[1] Chengzhi Guo[1] Alpha A. Lee[1] Philippe Schwaller[5] Jian Tang[9 10 11]

Advanced Operating Conditions Search applied in Analog Circuit Verification

Cristian Manolache[1], Alexandru Caranica[1], Marius Stănescu[1], Horia Cucu[1], Andi Buzo[2], Cristian Diaconu[2], Georg Pelz[2]
[1]University "Politehnica" of Bucharest, Romania
[2]Infineon Technologies, Munich, Germany
{cristian.manolache, alexandru.caranica, adrian.stanescu, horia.cucu}@upb.ro
{andi.buzo, cristian-vasile.diaconu, georg.pelz}@infineon.com

Rapidly Inferring Personalized Neurostimulation Parameters with Meta-Learning: A Case Study of Individualized Fiber Recruitment in Vagus Nerve Stimulation

Ximeng Mao[1,2], Yao-Chuan Chang[3], Stavros Zanos[3], Guillaume Lajoie[1,4,5,*]

ARTICLE

https://doi.org/10.1038/s41467-022-28580-6 OPEN

Check for updates

A self-driving laboratory advances the Pareto front for material properties

Benjamin P. MacLeod[1,2,5], Fraser G. L. Parlane[1,2,5], Connor C. Rupnow[1,2,3], Kevan E. Dettelbach[1], Michael S. Elliott[1], Thomas D. Morrissey[1,2], Ted H. Haley[1], Oleksii Proskurin[1], Michael B. Rooney[1], Nina Taherimakhsousi[1], David J. Dvorak[2], Hsi N. Chiu[1], Christopher E. B. Waizenegger[1], Karry Ocean[1], Mehrdad Mokhtari[1] & Curtis P. Berlinguette[1,2,3,4]

Bayesian Optimisation for Sequential Experimental Design with Applications in Additive Manufacturing

Mimi Zhang[1,4], Andrew Parnell[2,4], Dermot Brabazon[3,4], Alessio Benavoli[1]

[1]School of Computer Science and Statistics, Trinity College Dublin, Ireland
[2]Hamilton Institute, Maynooth University, Ireland
[3]School of Mechanical & Manufacturing Engineering, Dublin City University, Ireland
[4]I-Form Advanced Manufacturing Research Centre, Science Foundation Ireland

Noisy Bayesian optimization for variational quantum eigensolvers

Giovanni Iannelli[a,b,c,d,*] and Karl Jansen[a]

Active Bayesian Causal Inference

Christian Toth, TU Graz; Lars Lorch, ETH Zürich; Christian Knoll, TU Graz; Andreas Krause, ETH Zürich; Franz Pernkopf, TU Graz

Robert Peharz*, TU Graz; Julius von Kügelgen*, MPI for Intelligent Systems, Tübingen, University of Cambridge

Tracking Blobs in the Turbulent Edge Plasma of a Tokamak Fusion Device

Woonghee Han[1,*], Randall A. Pietersen[2], Rafael Villamor-Lora[2], Matthew Beveridge[3], Nicola Offeddu[4], Theodore Golfinopoulos[1], Christian Theiler[4], James L. Terry[1], Earl S. Marmar[1], and Iddo Drori[3]

[1]MIT Plasma Science and Fusion Center, Cambridge, Massachusetts 02139, USA
[2]MIT, Civil and Environmental Engineering, Cambridge, Massachusetts 02139, USA
[3]MIT, Electrical Engineering and Computer Science, Cambridge, Massachusetts 02142, USA
[4]École Polytechnique Fédérale de Lausanne (EPFL), Swiss Plasma Center (SPC), CH-1015 Lausanne, Switzerland
*harryhan@mit.edu

ARTICLE

https://doi.org/10.1038/s41467-021-25757-3 OPEN

Check for updates

Turn-key constrained parameter space exploration for particle accelerators using Bayesian active learning

Ryan Roussel[1], Juan Pablo Gonzalez-Aguilera[1], Young-Kee Kim[1], Eric Wisniewski[2], Wanming Liu[2], Philippe Piot[2,3], John Power[2], Adi Hanuka[4] & Auralee Edelen[4]

Bayesian optimization of distributed neurodynamical controller models for spatial navigation

Armin Hadzic[a,*], Grace M. Hwang[a,b,1], Kechen Zhang[c], Kevin M. Schultz[a], Joseph D. Monaco[c]

POLYHYMNIA Mood – Empowering people to cope with depression through music listening

Eduardo Coutinho, Applied Music Research Lab, University of Liverpool, United Kingdom, e.coutinho@liverpool.ac.uk

Ayesh Alshukri, Oxford Clinical Trials Research Unit, University of Oxford, United Kingdom, ayesh.alshukri@ndorms.ox.ac.uk

Jacopo de Berardinis, School of Computer Science, University of Manchester, United Kingdom, jacopo.deberardinis@manchester.ac.uk

Chris Dowrick, Department of Primary Care and Mental Health, University of Liverpool, United Kingdom, cfd@liverpool.ac.uk

Adaptive nonparametric psychophysics

Lucy Owen, Dartmouth College[2]
Jonathan Browder, Facebook Reality Labs
Benjamin Letham, Facebook
Gideon Stocek, Facebook Reality Labs
Chase Tymms, Facebook Reality Labs
Michael Shvartsman, Facebook Reality Labs

Benchmarking Structured Policies and Policy Optimization for Real-World Dexterous Object Manipulation

Niklas Funk[*1], Charles Schaff[*2], Rishabh Madan[*3], Takuma Yoneda[*2], Julen Urain De Jesus[1], Joe Watson[1], Ethan K. Gordon[4], Felix Widmaier[5], Stefan Bauer[5], Siddhartha S. Srinivasa[4], Tapomayukh Bhattacharjee[3], Matthew R. Walter[2], Jan Peters[1]

Article

Lost in Optimization of Water Distribution Systems: Better Call Bayes

Antonio Candelieri[1,*], Andrea Ponti[2,3], Ilaria Giordani[2,4] and Francesco Archetti[3,4]

4 Tutorial: Hands-on Examples

The Ax & BoTorch Tech Stack

Ax

All-in-one blackbox optimization
- High-level APIs
- Storage backend
- Scheduling capabilities
- ...

BoTorch

Bayesian Optimization research
- Modular / composable / extensible

Developed & maintained by the Adaptive Experimentation team

GPyTorch

Gaussian processes for *modern* machine learning systems.

Gaussian Process research
- Scalable GP models
- GPU acceleration

Co-maintained by the Adaptive Experimentation team

PyTorch

Part or the Linux Foundation (technical governance: Meta)

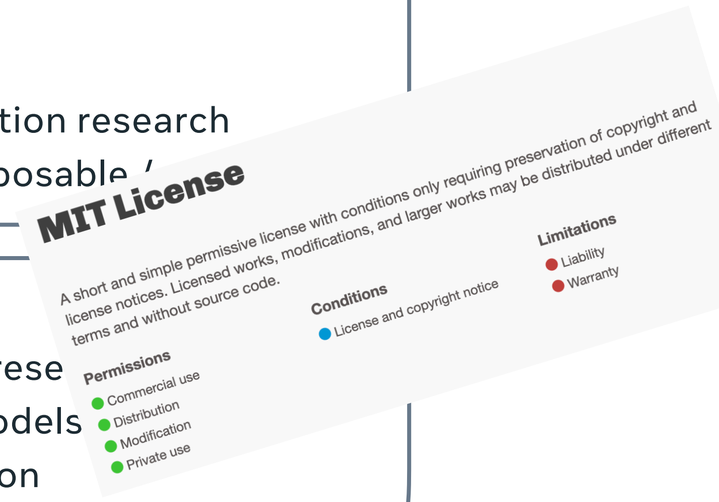
The Ax & BoTorch Tech Stack

Ax

All-in-one blackbox optimization
- High-level APIs
- Storage backend
- Scheduling capabilities
- ...

BoTorch

Bayesian Optimization research
- Modular / composable /



Developed & maintained by the Adaptive Experimentation team

GPyTorch

Gaussian processes for *modern* machine learning systems.

Gaussian Process rese
- Scalable GP models
- GPU acceleration

Co-maintained by the Adaptive Experimentation team

PyTorch

Part or the Linux Foundation (technical governance: Meta)

Ax APIs: Service

- Ask-tell interface
- **Simple yet powerful**
- Scheduling & Evaluation of trials is responsibility of the user
- **Should be the go-to** if additional advanced scheduling / deployment functionalities are not required

```python
from ax.service.ax_client import AxClient
from ax.utils.measurement.synthetic_functions import branin

ax_client = AxClient()
ax_client.create_experiment(
    name="branin_test_experiment",
    parameters=[
        {
            "name": "x1",
            "type": "range",
            "bounds": [-5.0, 10.0],
        },
        {
            "name": "x2",
            "type": "range",
            "bounds": [0.0, 10.0],
        },
    ],
    objective_name="branin",
    minimize=True,
)

for _ in range(15):
    parameters, trial_index = ax_client.get_next_trial()
    result = branin(parameters["x1"], parameters["x2"])
    ax_client.complete_trial(
        trial_index=trial_index,
        raw_data= result,
    )

best_parameters, metrics = ax_client.get_best_parameters()
```

(synthetic) black-box function f

Inferred to be “float”.
Can also define integer or categorical parameters.

Here is where we evaluate f

Ax APIs: Developer

- Highly flexibly and customizable (thus more complex)
- Runner and Metric abstractions allow implementing **trial deployment** / management and **data fetching** capabilities for **different backends**

```
from ax import *

class MockRunner(Runner):
    def run(self, trial):
        return {"name": str(trial.index)}


search_space = SearchSpace(
    parameters=[
        RangeParameter(
            name="x1", parameter_type=ParameterType.FLOAT, lower=-5, upper=10
        ),
        RangeParameter(
            name="x2", parameter_type=ParameterType.FLOAT, lower=0, upper=15
        ),
    ]
)
exp = Experiment(
    name="test_branin",
    search_space=search_space,
    optimization_config=OptimizationConfig(
        objective=Objective(
            metric=BraninMetric(name="branin", param_names=["x1", "x2"]),
            minimize=True,
        ),
    ),
    runner=MockRunner(),
)

sobol = Models.SOBOL(exp.search_space)
for _ in range(5):
    trial = exp.new_trial(generator_run=sobol.gen(n=1))
    trial.run()
    trial.mark_completed()

for _ in range(15):
    gp_ei = Models.GPEI(experiment=exp, data=exp.fetch_data())
    generator_run = gp_ei.gen(n=1)
    best_arm, _ = generator_run.best_arm_predictions
    trial = exp.new_trial(generator_run=generator_run)
    trial.run()
    trial.mark_completed()

exp.fetch_data()
best_parameters = best_arm.parameters
```

Usually used to deploy to backend (e.g. simulation cluster)

Example synthetic metric.

Quasi-random initial exploration

GP model + Expected Improvement

Ax APIs: Scheduler

- Orchestrates **fully automated closed-loop optimization** in a **backend-agnostic fashion** (leveraging `Metric` and `Runner` implementations)
- Supports advanced scheduling capabilities including
  - Parallel trial evaluations
  - Fully asynchronous trial evaluations
  - Early stopping of trials that provide partial results during the run
  - Early stopping of the overall optimization based on customizable stopping strategies

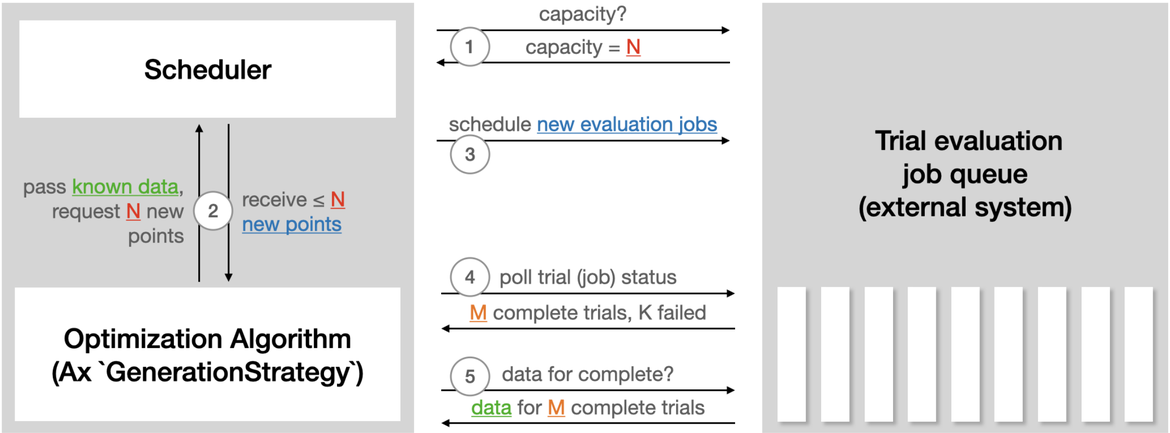
```python
from ax import *
from ax.modelbridge.generation_strategy import GenerationStrategy
from ax.service import Scheduler

# Full `Experiment` and `GenerationStrategy` instantiation
# omitted for brevity, refer to tutorials for detail.
experiment = Experiment(...)
generation_strategy = GenerationStrategy(...)

scheduler = Scheduler(
    experiment=experiment,
    generation_strategy=generation_strategy,
    options=SchedulerOptions(),  # Configurations for how to run the experiment
)

scheduler.run_n_trials(100)  # Automate running 100 trials and reporting results
```

can be chosen automatically



Getting Started: Multi-Objective BO w/ Ax

https://ax.dev/tutorials/multiobjective_optimization.html

Ax Docs Tutorials API GitHub Search v: stable

Tutorials
- Overview

API Comparison
- Loop API
- Service API
- Developer API

Deep Dives
- Visualizations
- Generation Strategy
- Scheduler
- Modular `BoTorchModel`

Bayesian Optimization
- Hyperparameter Optimization for PyTorch
- Hyperparameter Optimization via Raytune

Multi-Objective Optimization Ax API

Using the Service API

For Multi-objective optimization (MOO) in the `AxClient`, objectives are specified through the `ObjectiveProperties` dataclass. An `ObjectiveProperties` requires a boolean `minimize`, and also accepts an optional floating point `threshold`. If a `threshold` is not specified, Ax will infer it through the use of heuristics. If the user knows the region of interest (because they have specs or prior knowledge), then specifying the thresholds is preferable to inferring it. But if the user would need to guess, inferring is preferable.

To learn more about how to choose a threshold, see Set Objective Thresholds to focus candidate generation in a region of interest. See the Service API Tutorial for more infomation on running experiments with the Service API.

In [1]:

```
from ax.service.ax_client import AxClient
from ax.service.utils.instantiation import ObjectiveProperties

import torch
```

Under the Hood: Multi-Objective BO w/ BoTorch

https://botorch.org/tutorials/multi_objective_bo

BoTorch Docs Tutorials API Reference Papers GitHub Search v: stable

Tutorials

Overview

Using BoTorch with Ax

Using a custom BoTorch model

Writing a custom acquisition function

Full Optimization Loops

q-Noisy Constrained EI

Bayesian optimization with pairwise comparison data

Bayesian optimization with preference exploration

Trust Region Bayesian Optimization (TuRBO)

Scalable Constrained Bayesian Optimization (SCBO)

Noisy, Parallel, Multi-Objective BO in BoTorch with qEHVI, qNEHVI, and qNParEGO¶

In this tutorial, we illustrate how to implement a simple multi-objective (MO) Bayesian Optimization (BO) closed loop in BoTorch.

In general, we recommend using Ax for a simple BO setup like this one, since this will simplify your setup (including the amount of code you need to write) considerably. See here for an Ax tutorial on MOBO. If desired, you can use a custom BoTorch model in Ax, following the Using BoTorch with Ax tutorial. Given a `MultiObjective`, Ax will default to the qNEHVI acquisiton function. If desired, this can also be customized by adding `"botorch_acqf_class": <desired_botorch_acquisition_function_class>,` to the `model_kwargs`.

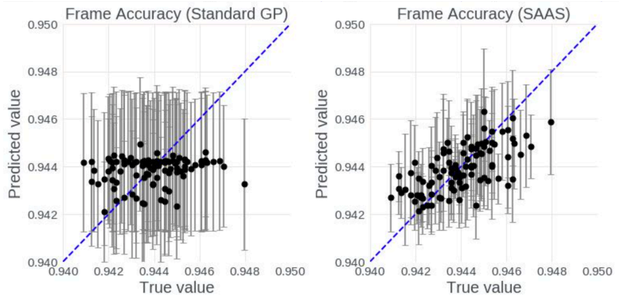
We use the parallel ParEGO (qParEGO) [1], parallel Expected Hypervolume Improvement (qEHVI) [1], and parallel Noisy Expected Hypervolume Improvement (qNEHVI) [2] acquisition functions to optimize a synthetic BraninCurrin problem test function with additive Gaussian observation noise over a 2-parameter search space [0,1]^2. See `botorch/test_functions/multi_objective.py` for details on BraninCurrin. The noise standard deviations are 15.19 and 0.63 for each objective, respectively.

Since botorch assumes a maximization of all objectives, we seek to find the Pareto frontier, the set of

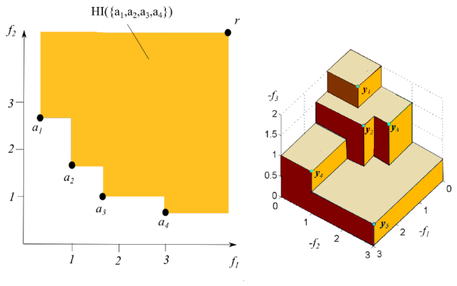
Common Gotchas: Scalability

- In the number of data points / function **evaluations (n)**:
  - Model fitting is O(n^3) with standard GP models / algorithms ⇒ typically limited to a few hundred observations (more scalable models / methods exist)

- In the dimensional of the search space / number of **parameters (d)**:
  - Computational scaling is relatively benign (kernel method!)
  - But: Statistical efficiency & model quality becomes an issue!
  - SAASBO (sparsity-inducing prior) and TuRBO (trust-region approach) can help scale to hundreds of parameters

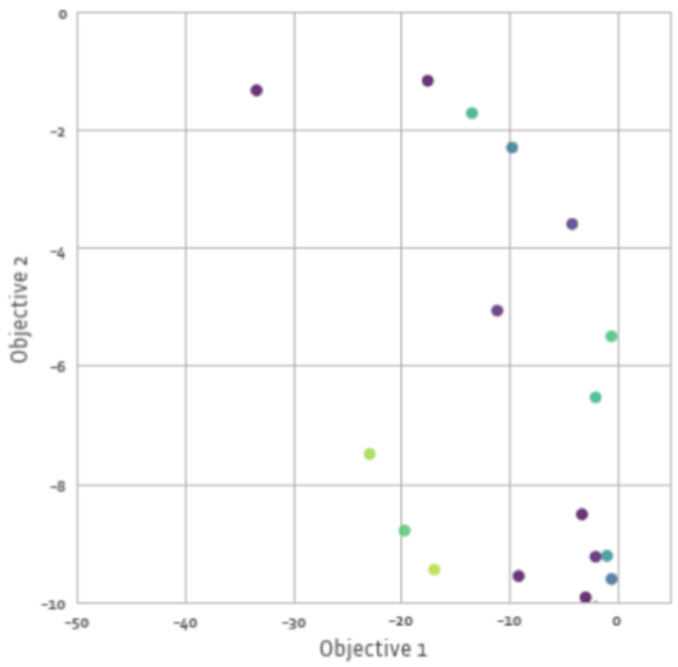


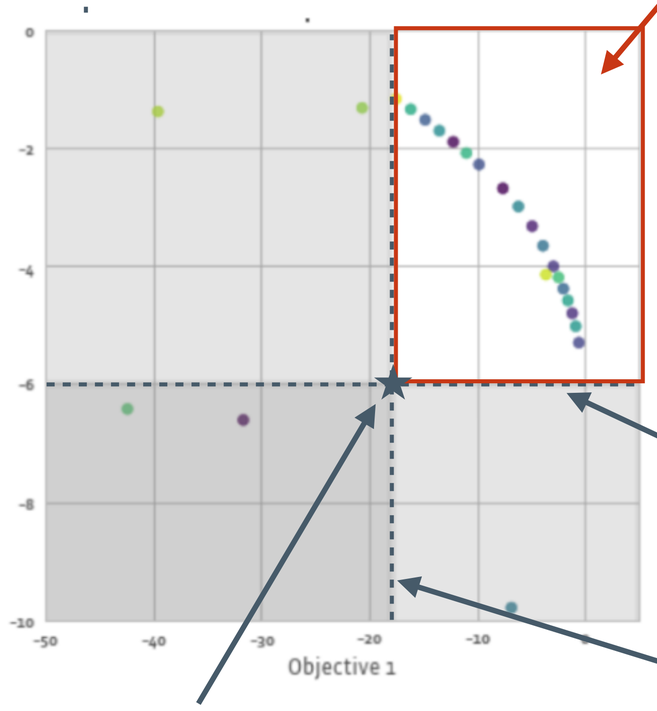
- In the number of **objectives (m)** for MOO:
  - Computing HV is super-polynomial in m ⇒ does not scale to more than 3-4 objectives (scaling in outcome constraints on other outcomes is more graceful)
- Random scalarization (e.g. qParEGO) can work with larger m (though less sample-efficient)



Common Gotchas: Setting Objective Thresholds

- Objective thresholds (Ax) = Reference point (BoTorch)
- These **define the "area of interest"** to the MOO algorithm **in the outcome space**
- Setting these properly is important for the efficiency of the optimization





Portion of the Pareto Frontier explored by the algorithm

Threshold on objective 1

Threshold on objective 2

Reference point

Other Common Gotchas

Noisy observations

- In practice, outcomes are often subject to **observation noise** (either known or unknown)
  - If noise level is unknown, is important for the model to understand if it needs to **infer a noise level** (in Ax that means passing a `NaN`/`None` variance observation, in BoTorch it means using a `SingleTaskGP` or similar)
  - If noise level is known, providing it to model can be very helpful

Numerical precision

- Inference with GPs often involves solving ill-conditioned linear systems
  - This is especially problematic if there are many repeated or very similar design points
  - Always **work in double precision if possible**!

5 Advanced Features

High-Dimensional Multi-Objective BO

Limitations of existing Multi-Objective Bayesian Optimization methods (e.g. qEHVI):

- Poor performance of GP models in **high-dimensional domains** ($\gtrsim$ 15 parameters)
- May not scale well to the **large-batch / high-throughput setting** with many observations
- Often do not support **black-box constraints**

High-Dimensional Multi-Objective BO

Limitations of existing Multi-Objective Bayesian Optimization methods (e.g. qEHVI):

- Poor performance of GP models in **high-dimensional domains** ($\gtrsim$ 15 parameters)
- May not scale well to the **large-batch / high-throughput setting** with many observations
- Often do not support **black-box constraints**

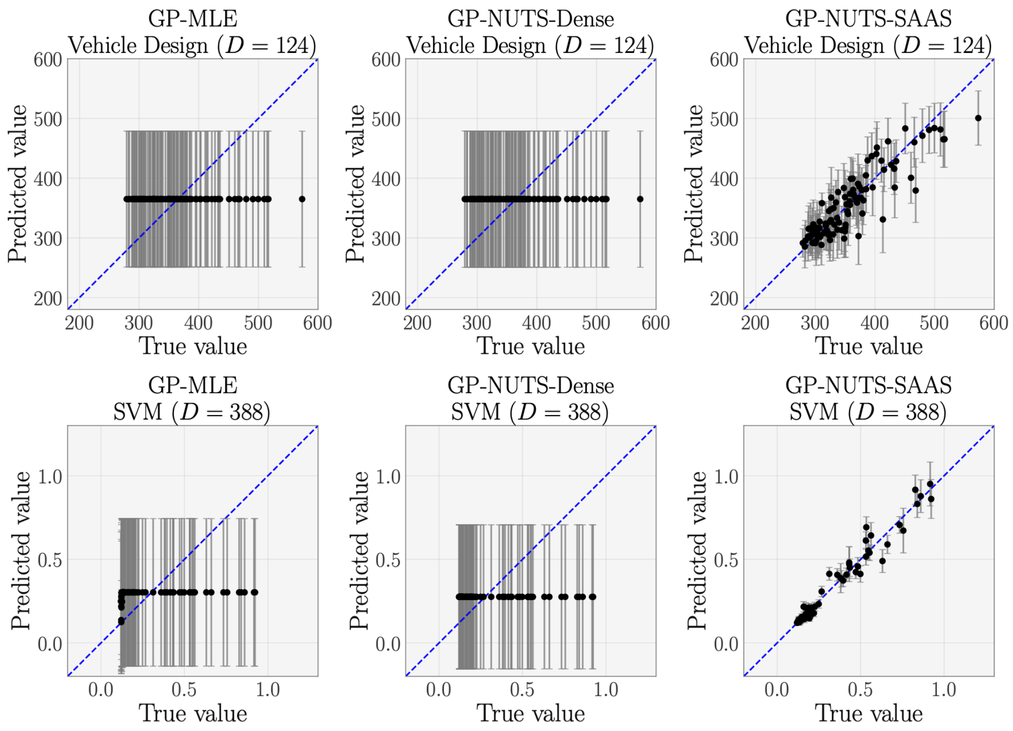
Our contributions:

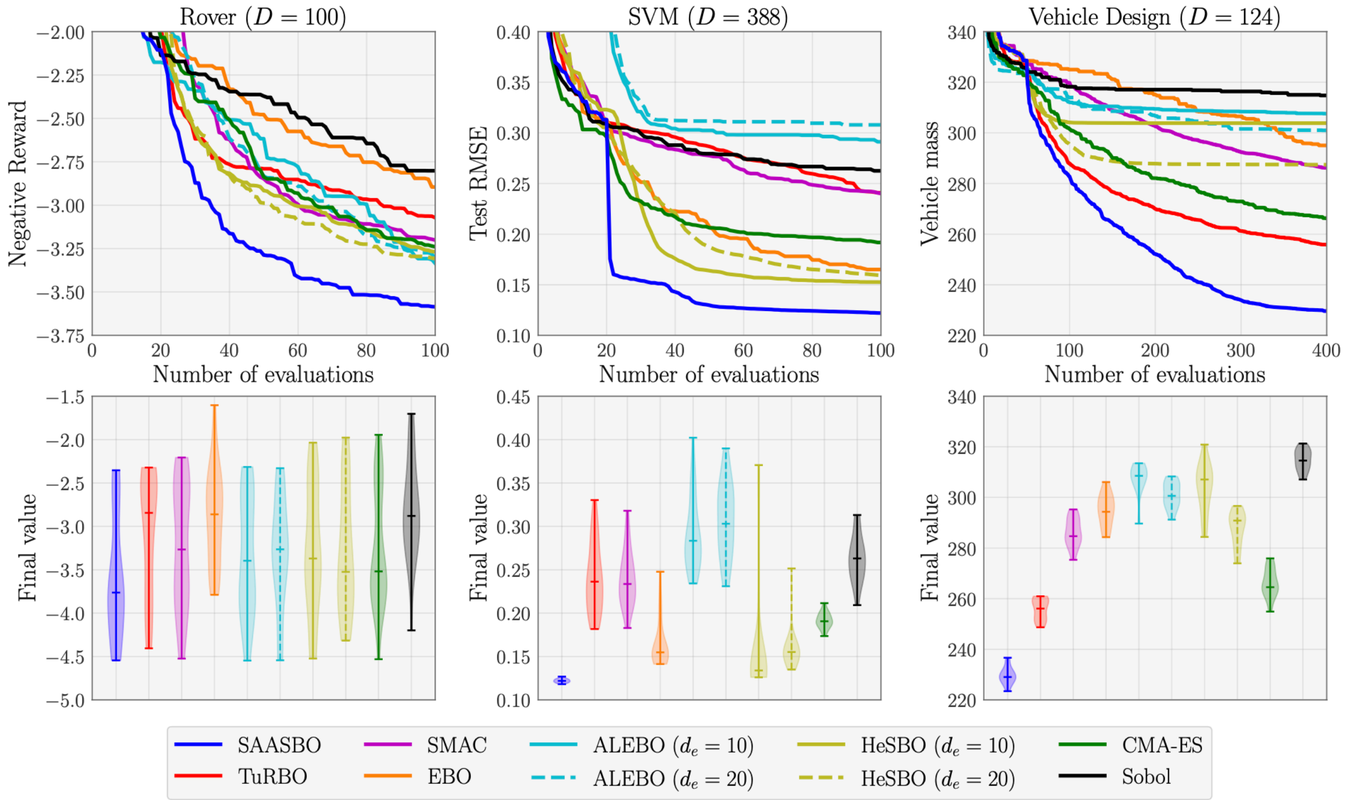
- **SAASBO:** "Sparse Axis-Aligned Subspace Bayesian Optimization"
  - Sparsity-inducing prior on the kernel lenghtscales **identifies the most important parameters**
  - Allows model to scale to **hundreds of dimensions**
- **MORBO**: "Multi-Objective trust Region Bayesian Optimization"
  - The first scalable MO-BO algorithm that is **practical for high-dim problems w/ thousands of evaluations**
  - **Efficient scaling** in both dimension and # of evaluations; **well-distributed high-quality Pareto frontiers**

D. Eriksson and M. Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. UAI 2021.
S. Daulton, D. Eriksson, M. Balandat, E. Bakshy. Multi-objective bayesian optimization over high-dimensional search spaces. UAI 2022.

SAASBO: Improved Surrogate Modeling

- In many high-dimensional problems, only a rather small set of the parameters have outsized influence
- **Idea:** Place a sparsity inducing prior over the length scales of the GP kernel
  - Inference performed via MCMC (NUTS)
- Can be combined with MOO



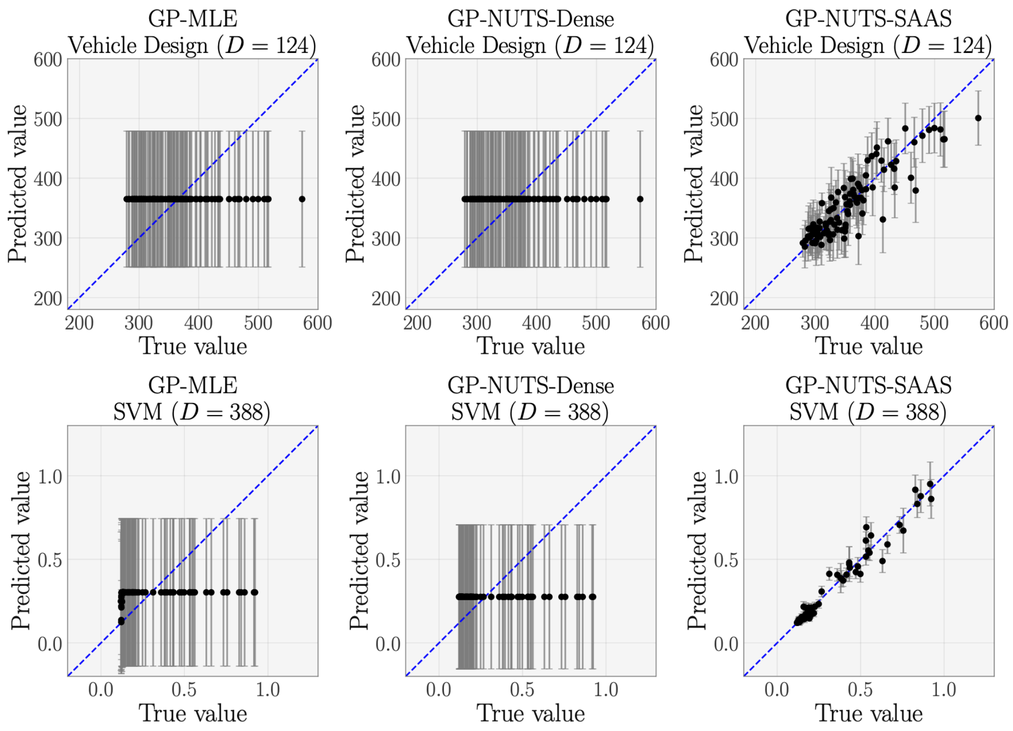


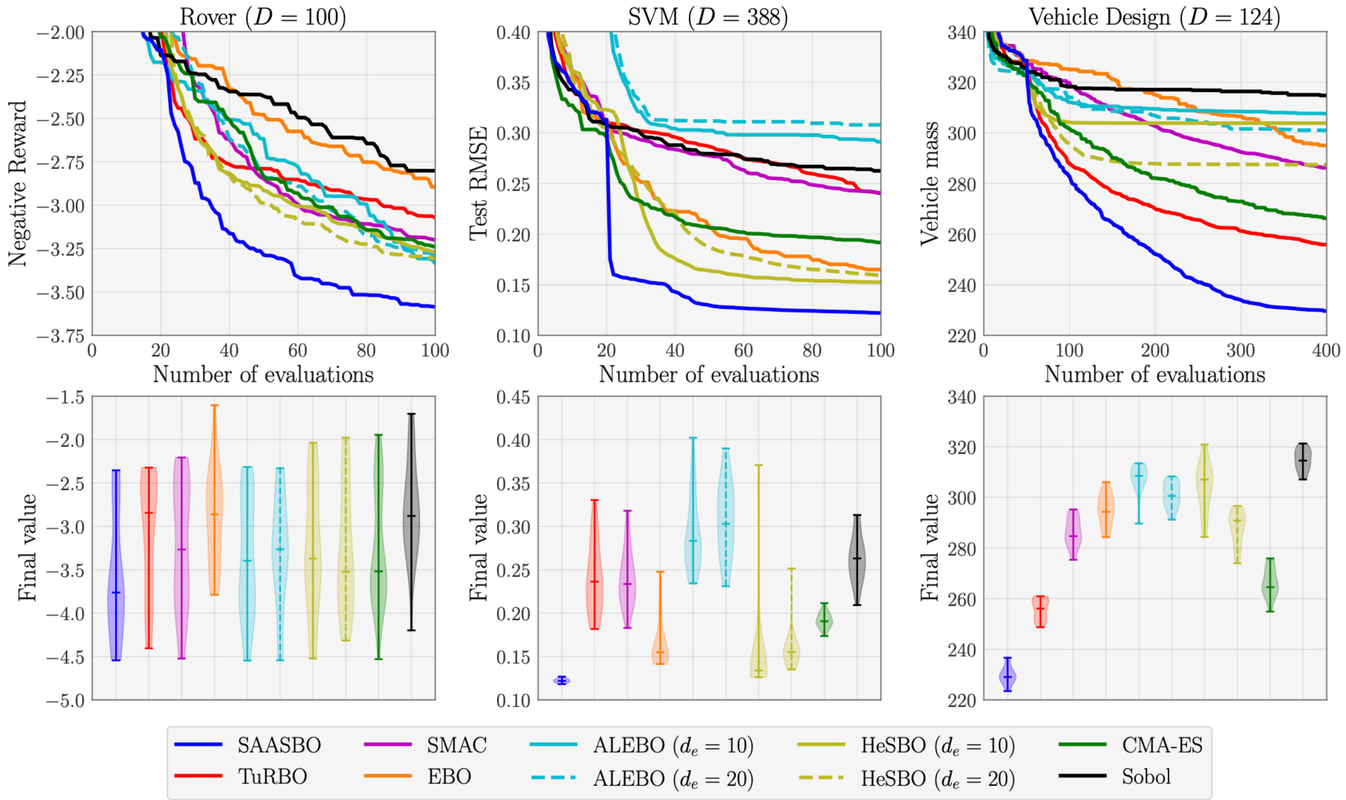
D. Eriksson and M. Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. UAI 2021.

SAASBO: Improved Surrogate Modeling

Available in Ax & BoTorch!

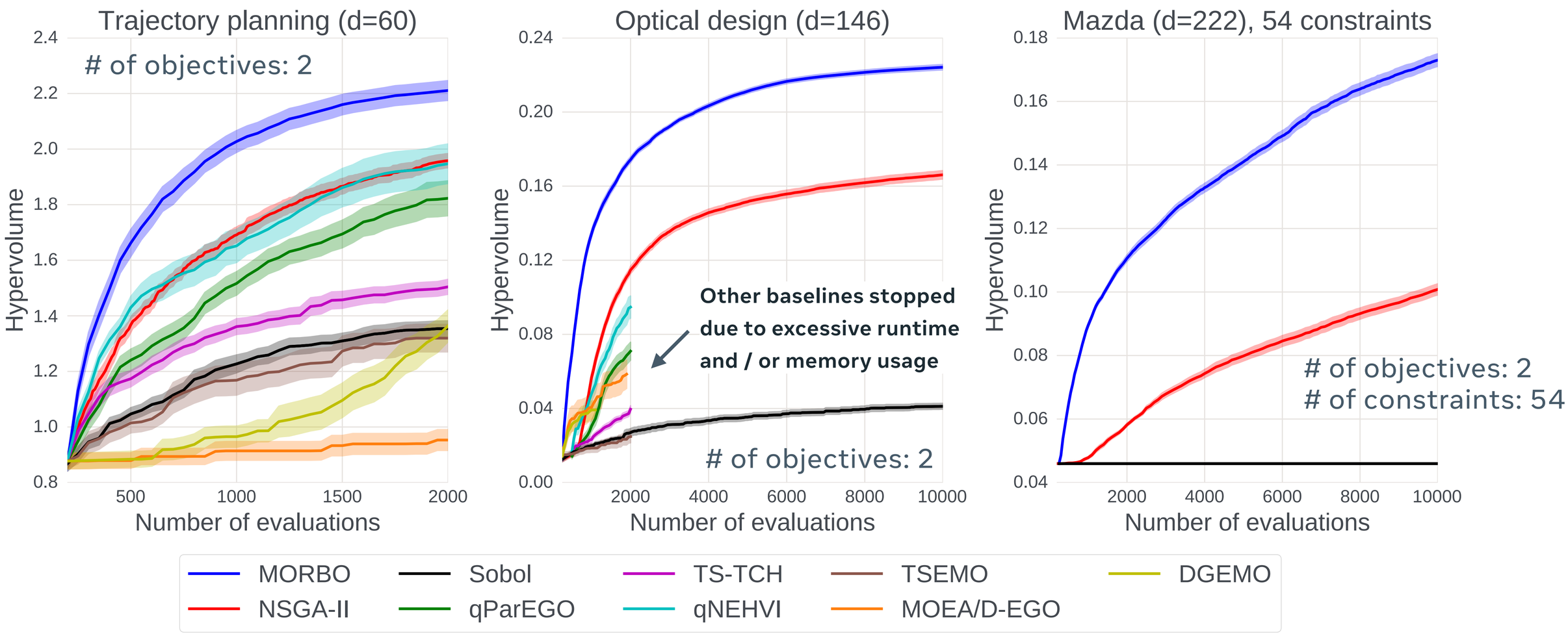
- In many high-dimensional problems, only a rather small set of the parameters have outsized influence
- **Idea:** Place a sparsity inducing prior over the length scales of the GP kernel
  - Inference performed via MCMC (NUTS)
- Can be combined with MOO





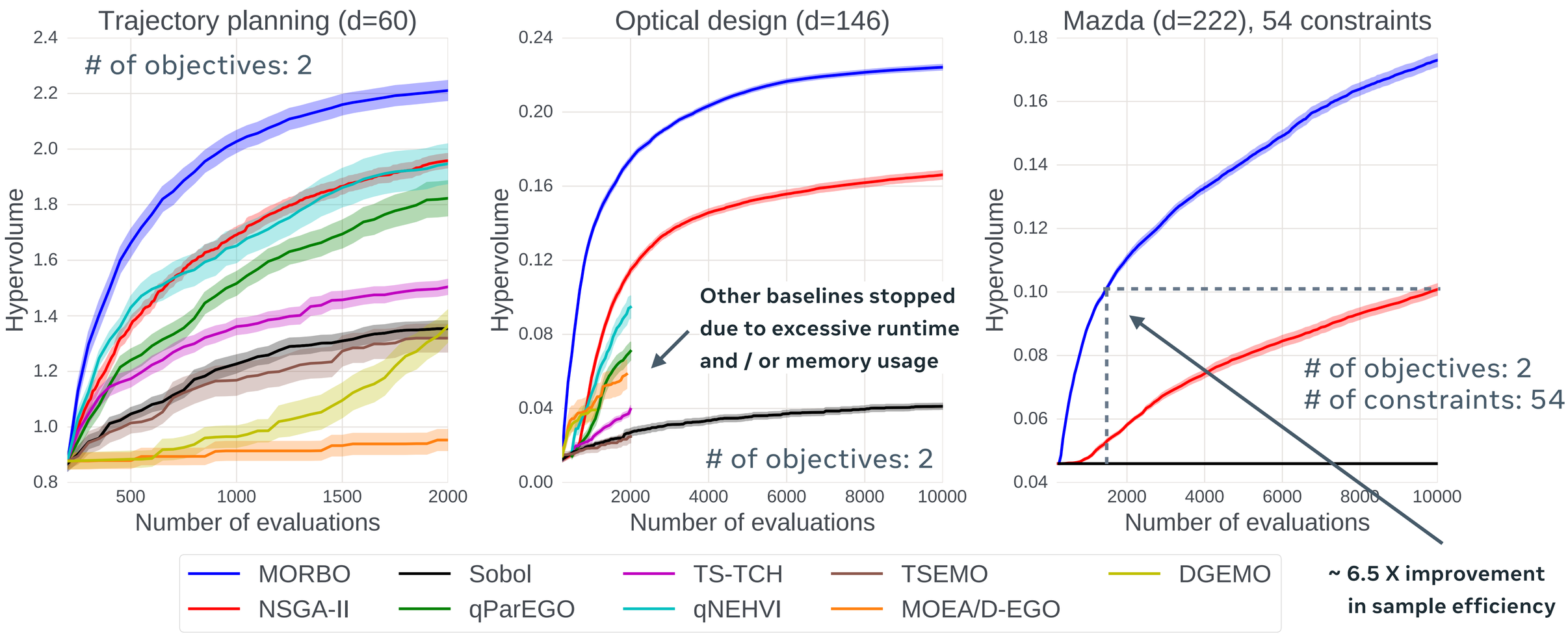
D. Eriksson and M. Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. UAI 2021.

MORBO: High-Throughput MOBO



S. Daulton, D. Eriksson, M. Balandat, E. Bakshy. Multi-objective bayesian optimization over high- dimensional search spaces. UAI 2022.

MORBO: High-Throughput MOBO



~ 6.5 X improvement in sample efficiency

S. Daulton, D. Eriksson, M. Balandat, E. Bakshy. Multi-objective bayesian optimization over high- dimensional search spaces. UAI 2022.

Robust Multi-Objective Optimization

- In practice, we may not be able to realize a design $x \in \mathbb{X}$ exactly
  - e.g. we cannot build 100% to spec due to manufacturing tolerances, environmental conditions, etc.
- How can we be robust to this input noise? How to even formulate this in a multi-objective setting?

S. Daulton, S. Cakmak, M. Balandat, M. A. Osborne, E. Zhou, and E. Bakshy. Robust multi-objective Bayesian optimization under input noise. ICML 2022.

Robust Multi-Objective Optimization

- In practice, we may not be able to realize a design $x \in \mathbb{X}$ exactly
  - e.g. we cannot build 100% to spec due to manufacturing tolerances, environmental conditions, etc.
- How can we be robust to this input noise? How to even formulate this in a multi-objective setting?

Definition 4.1. Given input noise $\boldsymbol{\xi} \sim P(\boldsymbol{\xi})$ where $\boldsymbol{\xi} \in \mathbb{R}^d$ and a confidence level $\alpha \in [0, 1]$, the value-at-risk for a given point $\boldsymbol{x}$ is:

$$\mathrm{VaR}_\alpha\big[f(\boldsymbol{x} \diamond \boldsymbol{\xi})\big] = \sup\{z \in \mathbb{R} : P\big[f(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq z\big] \geq \alpha\}.$$

S. Daulton, S. Cakmak, M. Balandat, M. A. Osborne, E. Zhou, and E. Bakshy. Robust multi-objective Bayesian optimization under input noise. ICML 2022.

Robust Multi-Objective Optimization

- In practice, we may not be able to realize a design $x \in \mathbb{X}$ exactly
  - e.g. we cannot build 100% to spec due to manufacturing tolerances, environmental conditions, etc.
- How can we be robust to this input noise? How to even formulate this in a multi-objective setting?

Definition 4.1. Given input noise $\boldsymbol{\xi} \sim P(\boldsymbol{\xi})$ where $\boldsymbol{\xi} \in \mathbb{R}^d$ and a confidence level $\alpha \in [0, 1]$, the value-at-risk for a given point $\boldsymbol{x}$ is:

$$\text{VaR}_\alpha\big[f(\boldsymbol{x} \diamond \boldsymbol{\xi})\big] = \sup\{z \in \mathbb{R} : P\big[f(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq z\big] \geq \alpha\}.$$

Definition 4.2. The MVaR of $\boldsymbol{f}$ for a given point $\boldsymbol{x}$ and confidence level $\alpha \in [0, 1]$ is:

$$\boxed{\begin{aligned}&\text{MVaR}_\alpha\big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\big] = \\ &\qquad \text{Pareto}\big(\{\boldsymbol{z} \in \mathbb{R}^M : P\big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}\big] \geq \alpha\}\big).\end{aligned}}$$

The MVaR set over X specifies objective vectors $\boldsymbol{z}$ such that there exists a known design $\boldsymbol{x} \in X$ with corresponding random objectives $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ under $P(\boldsymbol{\xi})$ that dominate $\boldsymbol{z}$ with probability $\geq \alpha$.

S. Daulton, S. Cakmak, M. Balandat, M. A. Osborne, E. Zhou, and E. Bakshy. Robust multi-objective Bayesian optimization under input noise. ICML 2022.

Robust Multi-Objective Optimization

- In practice, we may not be able to realize a design $x \in \mathbb{X}$ exactly
  - e.g. we cannot build 100% to spec due to manufacturing tolerances, environmental conditions, etc.
- How can we be robust to this input noise? How to even formulate this in a multi-objective setting?

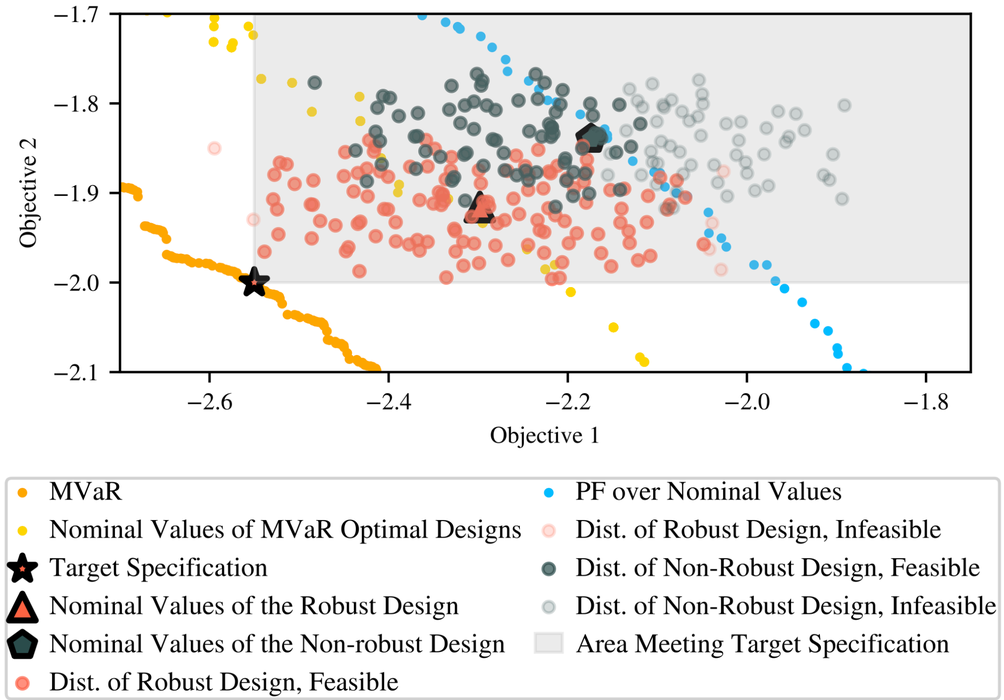
Definition 4.1. Given input noise $\boldsymbol{\xi} \sim P(\boldsymbol{\xi})$ where $\boldsymbol{\xi} \in \mathbb{R}^d$ and a confidence level $\alpha \in [0, 1]$, the value-at-risk for a given point $\boldsymbol{x}$ is:

$$\mathrm{VaR}_\alpha\big[f(\boldsymbol{x} \diamond \boldsymbol{\xi})\big] = \sup\{z \in \mathbb{R} : P\big[f(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq z\big] \geq \alpha\}.$$

Definition 4.2. The MVaR of $\boldsymbol{f}$ for a given point $\boldsymbol{x}$ and confidence level $\alpha \in [0, 1]$ is:

$$\boxed{\begin{aligned}&\mathrm{MVaR}_\alpha\big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\big] = \\ &\qquad \mathrm{Pareto}\big(\{\boldsymbol{z} \in \mathbb{R}^M : P\big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}\big] \geq \alpha\}\big).\end{aligned}}$$

The MVaR set over X specifies objective vectors $\boldsymbol{z}$ such that there exists a known design $\boldsymbol{x} \in X$ with corresponding random objectives $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ under $P(\boldsymbol{\xi})$ that dominate $\boldsymbol{z}$ with probability $\geq \alpha$.



- Contribution: MVAR Approximation based on Random Scalarizations (MARS)

S. Daulton, S. Cakmak, M. Balandat, M. A. Osborne, E. Zhou, and E. Bakshy. Robust multi-objective Bayesian optimization under input noise. ICML 2022.

Robust Multi-Objective Optimization

Available in Ax & BoTorch!

- In practice, we may not be able to realize a design $x \in \mathbb{X}$ exactly
  - e.g. we cannot build 100% to spec due to manufacturing tolerances, environmental conditions, etc.
- How can we be robust to this input noise? How to even formulate this in a multi-objective setting?

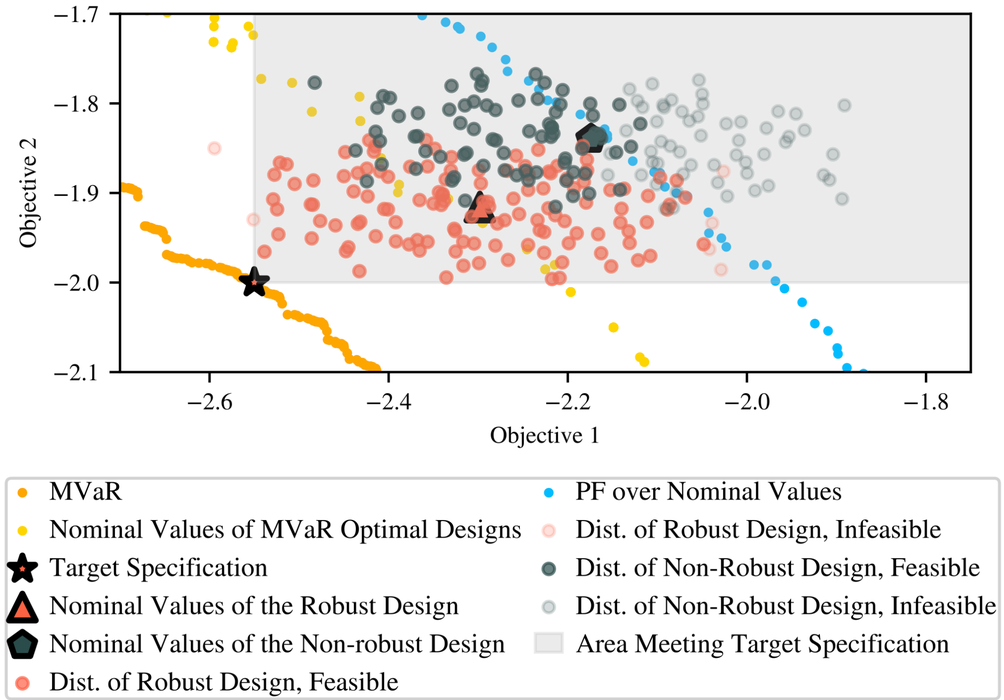
Definition 4.1. Given input noise $\boldsymbol{\xi} \sim P(\boldsymbol{\xi})$ where $\boldsymbol{\xi} \in \mathbb{R}^d$ and a confidence level $\alpha \in [0, 1]$, the value-at-risk for a given point $\boldsymbol{x}$ is:

$$\mathrm{VaR}_\alpha\big[f(\boldsymbol{x} \diamond \boldsymbol{\xi})\big] = \sup\{z \in \mathbb{R} : P\big[f(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq z\big] \geq \alpha\}.$$

Definition 4.2. The MVaR of $\boldsymbol{f}$ for a given point $\boldsymbol{x}$ and confidence level $\alpha \in [0, 1]$ is:

$$\boxed{\begin{aligned}&\mathrm{MVaR}_\alpha\big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\big] = \\ &\qquad \mathrm{Pareto}\big(\{\boldsymbol{z} \in \mathbb{R}^M : P\big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}\big] \geq \alpha\}\big).\end{aligned}}$$

The MVaR set over X specifies objective vectors $\boldsymbol{z}$ such that there exists a known design $\boldsymbol{x} \in X$ with corresponding random objectives $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ under $P(\boldsymbol{\xi})$ that dominate $\boldsymbol{z}$ with probability $\geq \alpha$.



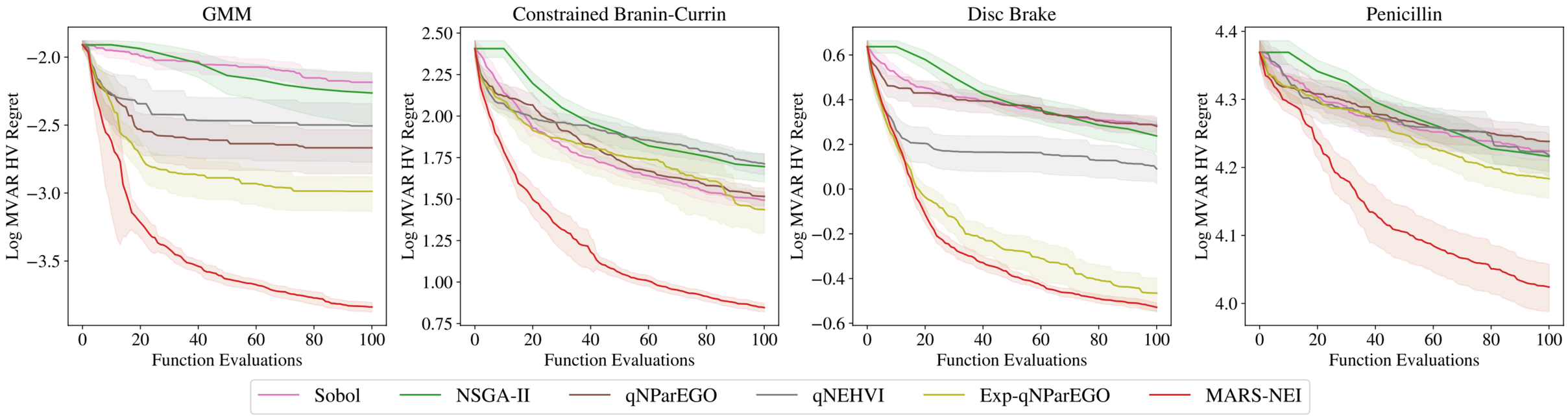
- Contribution: MVAR Approximation based on Random Scalarizations (MARS)

S. Daulton, S. Cakmak, M. Balandat, M. A. Osborne, E. Zhou, and E. Bakshy. Robust multi-objective Bayesian optimization under input noise. ICML 2022.

Robust Multi-Objective Optimization

Available in Ax & BoTorch!

- MARS uses a **novel theoretical connection** between the MVAR and the VAR of a particular scalarization of the objectives
  - This motivates a family of **computationally efficient BO methods for identifying MVAR**
- MARS allows to **identify optimal tradeoffs subject to a constraint on the yield** (% of feasible designs)



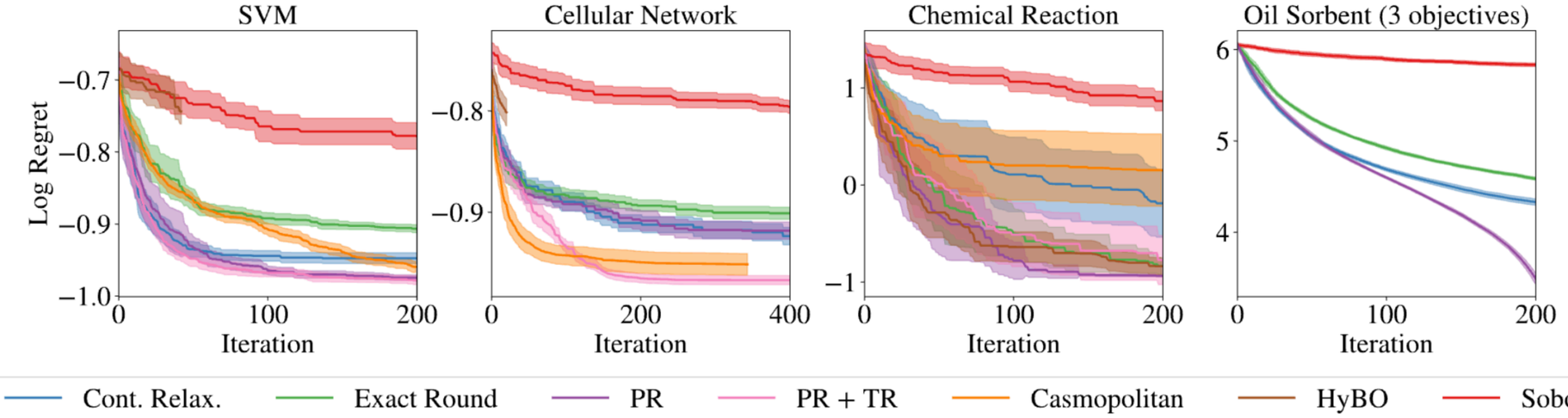
S. Daulton, S. Cakmak, M. Balandat, M. A. Osborne, E. Zhou, and E. Bakshy. Robust multi-objective Bayesian optimization under input noise. ICML 2022.

Mixed Discrete & Continuous Spaces

- Many optimization problems include discrete (ordered or categorical) parameters: $\mathbb{X} = \mathscr{X} \times \mathscr{Z}$
- Ax and BoTorch support discrete parameters out of the box
  - However, common approaches (such as one-hot encoding or continuous relaxation) do not scale well and have other issues
- New approach: **Probabilistic Reparameterization**
  - Key idea: Instead of optimizing over discrete Z, **optimize over probability distribution** $Z \sim p(Z \mid \theta)$**,** with θ a (continuous) parameter
  - Optimizer of reparameterization recovers optimizer of discrete problem.

Algorithm 1 BO with PR

1: Input: black-box objective $f : \mathcal{X} \times \boldsymbol{Z} \to \mathbb{R}$
2: Initialize $\mathcal{D}_0 \leftarrow \emptyset$, $\mathrm{GP}_0 \leftarrow \mathrm{GP}(\mathbf{0}, k)$
3: **for** $n = 1$ **to** $N_{\text{iterations}}$ **do**
4: $\quad (\boldsymbol{x}_n, \boldsymbol{\theta}_n) \leftarrow \arg\max_{(\boldsymbol{x},\boldsymbol{\theta}) \in \mathcal{X} \times \Theta} \mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]$
5: $\quad$ Sample $\boldsymbol{z}_n \sim p(\boldsymbol{Z}|\boldsymbol{\theta}_n)$
6: $\quad$ Evaluate $f(\boldsymbol{x}_n, \boldsymbol{z}_n)$
7: $\quad \mathcal{D}_n \leftarrow \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_n, \boldsymbol{z}_n, \boldsymbol{f}(\boldsymbol{x}_n, \boldsymbol{z}_n))\}$
8: $\quad$ Update posterior GP_n given $\mathcal{D}_n$
9: **end for**



S. Daulton, X. Wan, D. Eriksson, M. Balandat, M. A. Osborne, and E. Bakshy. Bayesian Optimization over Discrete and Mixed Spaces via Probabilistic Reparameterization. ICML 2022.

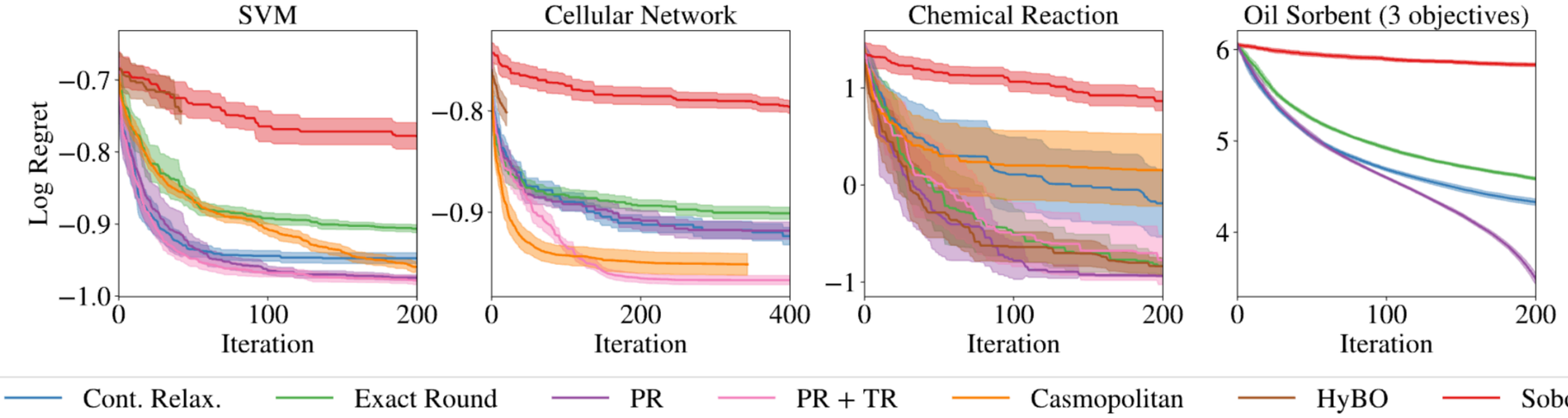
Mixed Discrete & Continuous Spaces

Improved support in Ax & BoTorch currently in the works!

- Many optimization problems include discrete (ordered or categorical) parameters: $\mathbb{X} = \mathcal{X} \times \mathcal{Z}$
- Ax and BoTorch support discrete parameters out of the box
  - However, common approaches (such as one-hot encoding or continuous relaxation) do not scale well and have other issues
- New approach: **Probabilistic Reparameterization**
  - Key idea: Instead of optimizing over discrete Z, **optimize over probability distribution** $Z \sim p(Z \mid \theta)$**,** with θ a (continuous) parameter
  - Optimizer of reparameterization recovers optimizer of discrete problem.

Algorithm 1 BO with PR

1: Input: black-box objective $f : \mathcal{X} \times \boldsymbol{Z} \to \mathbb{R}$
2: Initialize $\mathcal{D}_0 \leftarrow \emptyset$, $\mathrm{GP}_0 \leftarrow \mathrm{GP}(\mathbf{0}, k)$
3: **for** $n = 1$ **to** $N_{\text{iterations}}$ **do**
4: $\quad (\boldsymbol{x}_n, \boldsymbol{\theta}_n) \leftarrow \arg\max_{(\boldsymbol{x},\boldsymbol{\theta}) \in \mathcal{X} \times \Theta} \mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]$
5: $\quad$ Sample $\boldsymbol{z}_n \sim p(\boldsymbol{Z}|\boldsymbol{\theta}_n)$
6: $\quad$ Evaluate $f(\boldsymbol{x}_n, \boldsymbol{z}_n)$
7: $\quad \mathcal{D}_n \leftarrow \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_n, \boldsymbol{z}_n, \boldsymbol{f}(\boldsymbol{x}_n, \boldsymbol{z}_n))\}$
8: $\quad$ Update posterior GP_n given $\mathcal{D}_n$
9: **end for**



S. Daulton, X. Wan, D. Eriksson, M. Balandat, M. A. Osborne, and E. Bakshy. Bayesian Optimization over Discrete and Mixed Spaces via Probabilistic Reparameterization. ICML 2022.

Other Advanced Features

- Hierarchical Search Spaces (HSS)
  - Certain parameters may only be relevant conditional on the value of other parameters

- Multi-fidelity optimization
  - Trial evaluations can be run at lower fidelity (e.g. simulation resolution) to produce cheaper, but noisy and potentially biased observations

- Trial-level early stopping
  - By monitoring partial evaluation results during the trial run, we can identify unpromising runs and stop them early to save evaluation budget.

Resources

- Website (general docs, tutorials, API docs):
  - ax.dev
  - botch.org
- GitHub (issues, discussions, latest features):
  - https://github.com/facebook/Ax
  - https://github.com/pytorch/botorch

PRs welcome

- PyTorch Ax Multi-Objective Neural Architecture Search tutorial:
  - https://pytorch.org/tutorials/intermediate/ax_multiobjective_nas_tutorial.html

Ax BoTorch