The Center for BRIGHTBEAMS
A National Science Foundation Science & Technology Center

Online Multi-Objective Optimization Using Gaussian Processes

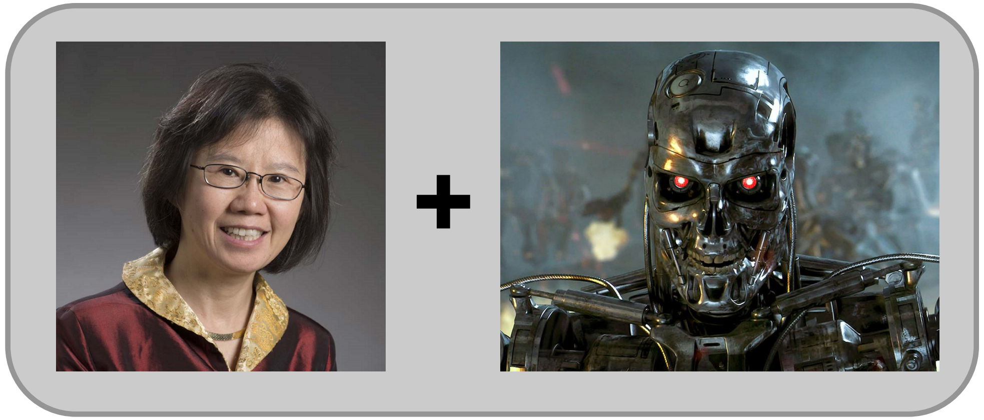
Ryan Roussel, Adi Hanuka, Auralee Edelen, Luis Hidalgo

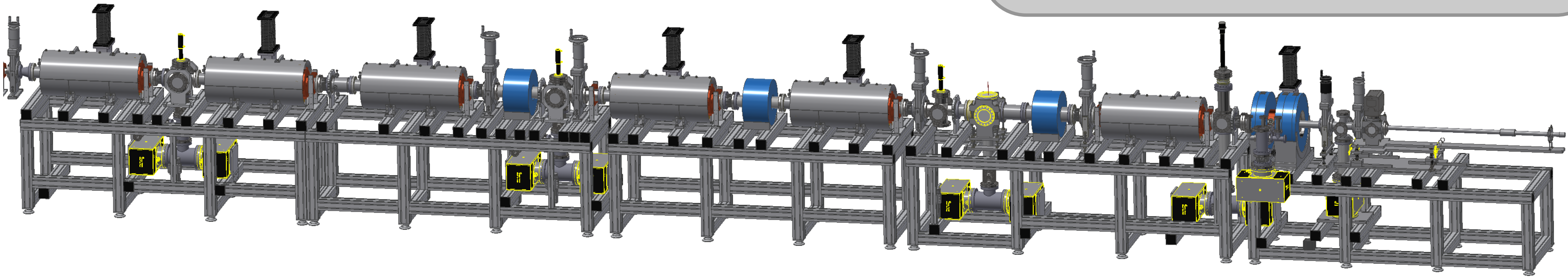
SLAC NATIONAL ACCELERATOR LABORATORY
THE UNIVERSITY OF CHICAGO
ENRICO FERMI INSTITUTE

Handing off Control/Tuning to the Machines

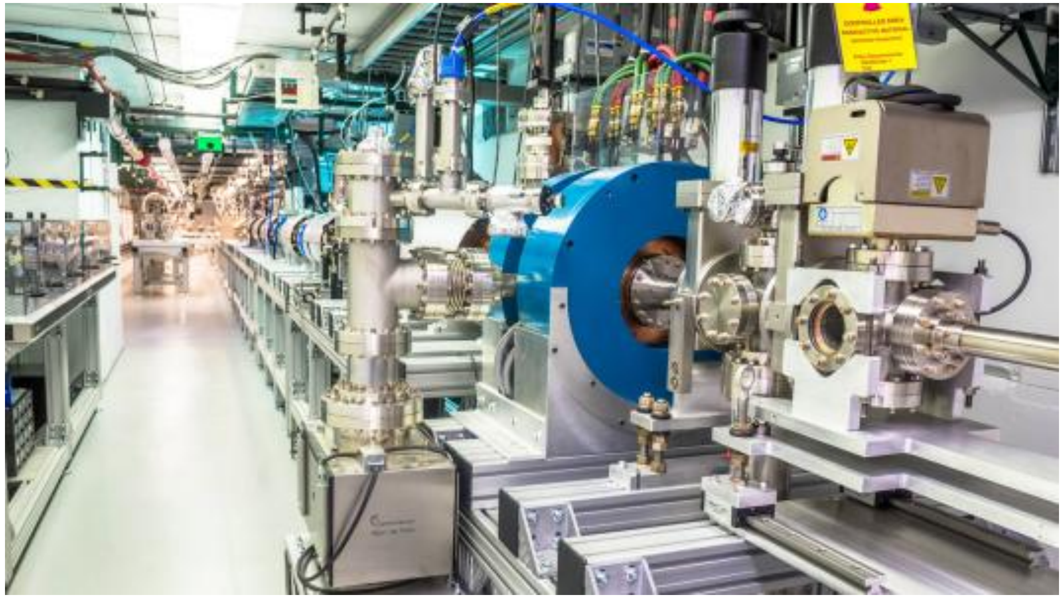
What do we want from a fully (or almost) autonomous system for controlling an accelerator?

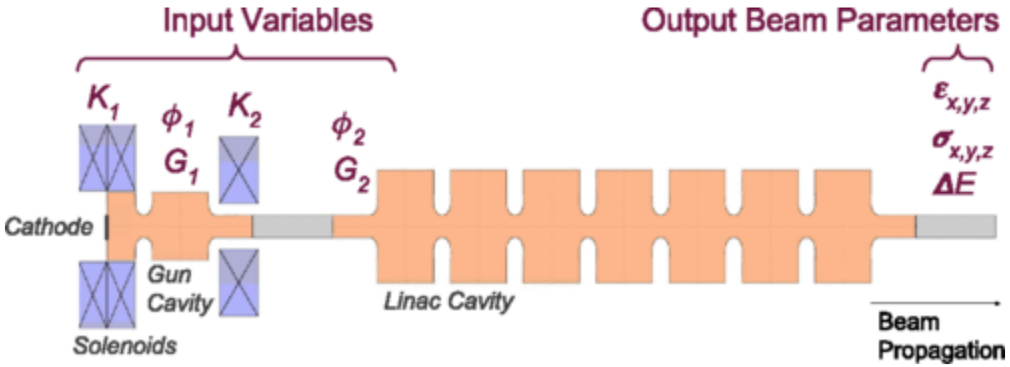
- Able to optimize towards **multiple objectives** at once
- Satisfy multiple operating **constraints**
- **Quickly** find reasonable solutions in extremely high dimensional systems (> 30D, < 1 hr tuning time)
- Handle parameter/observation **noise and drift**
- **Learn** from past experiments and simulations





Photoinjector Optimization





For online photoinjector optimization we wish to simultaneously:
- Minimize **emittances (3x)**
- Minimize **bunch sizes (3x)**
- Minimize **energy spread (1x)**

7 objectives

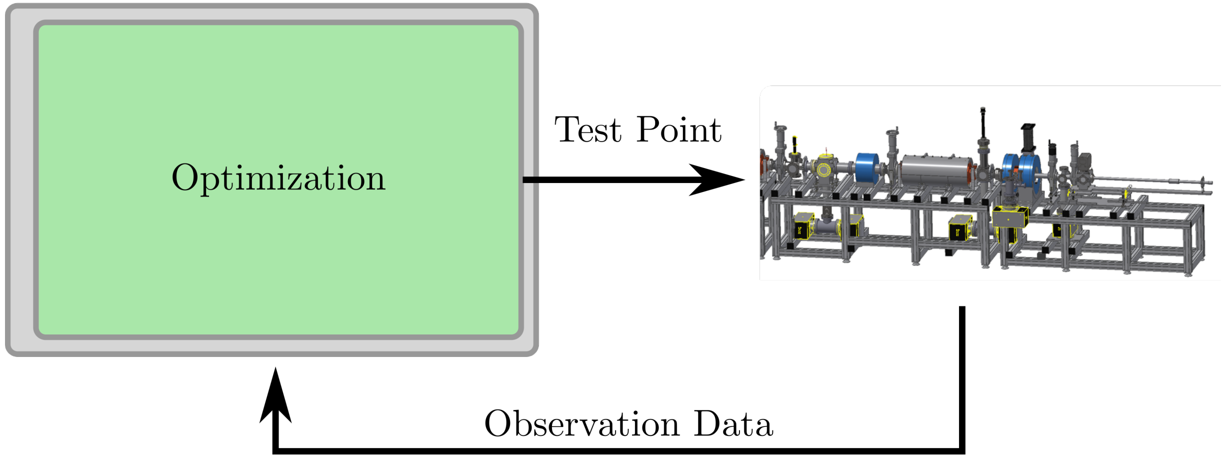
Tuning knobs:
- Solenoid strengths **(2x)**
- RF Amplitudes **(2x)**
- RF Phases **(2x)**

6 input parameters

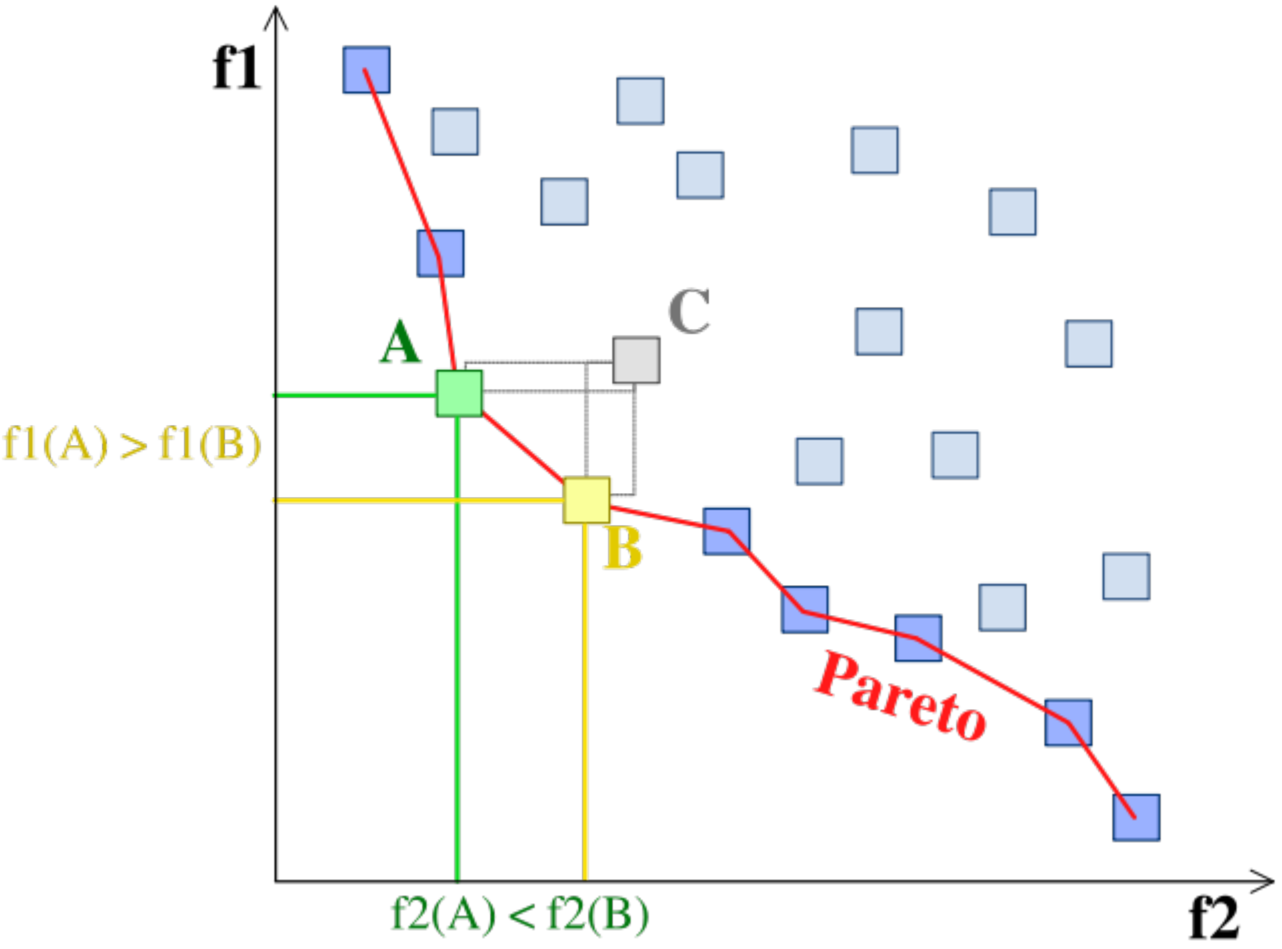
Edelen, Auralee, et al. PRAB 23.4 (2020): 044601.

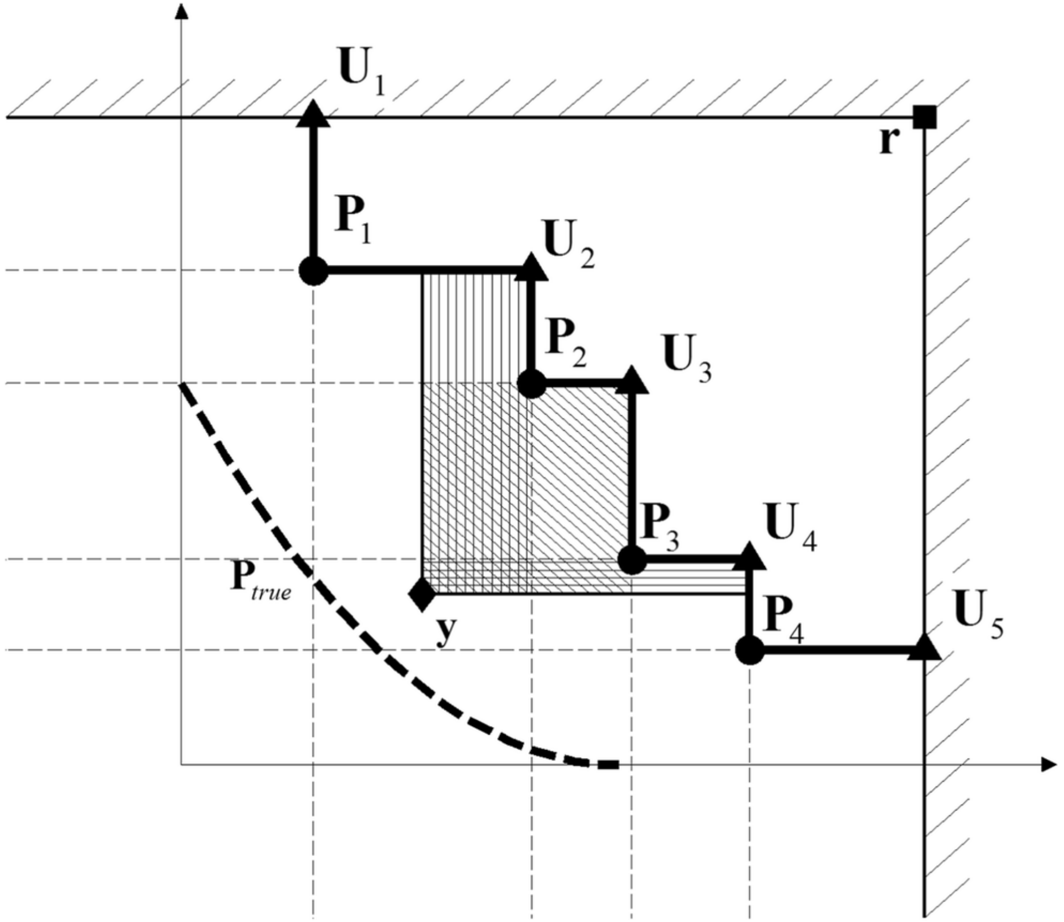
Normal Accelerator Optimization

Normal Optimization Algorithm



Multiple Objectives: Pareto Front

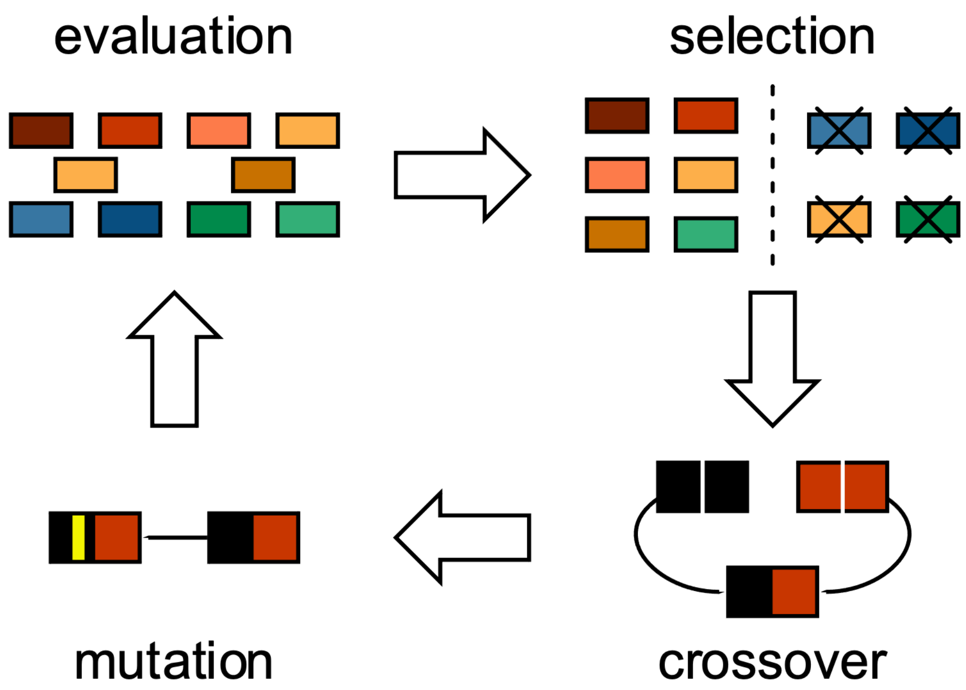




Li, Zheng, et al. *Structural and Multidisciplinary Optimization* 58.5 (2018): 1961-1979.

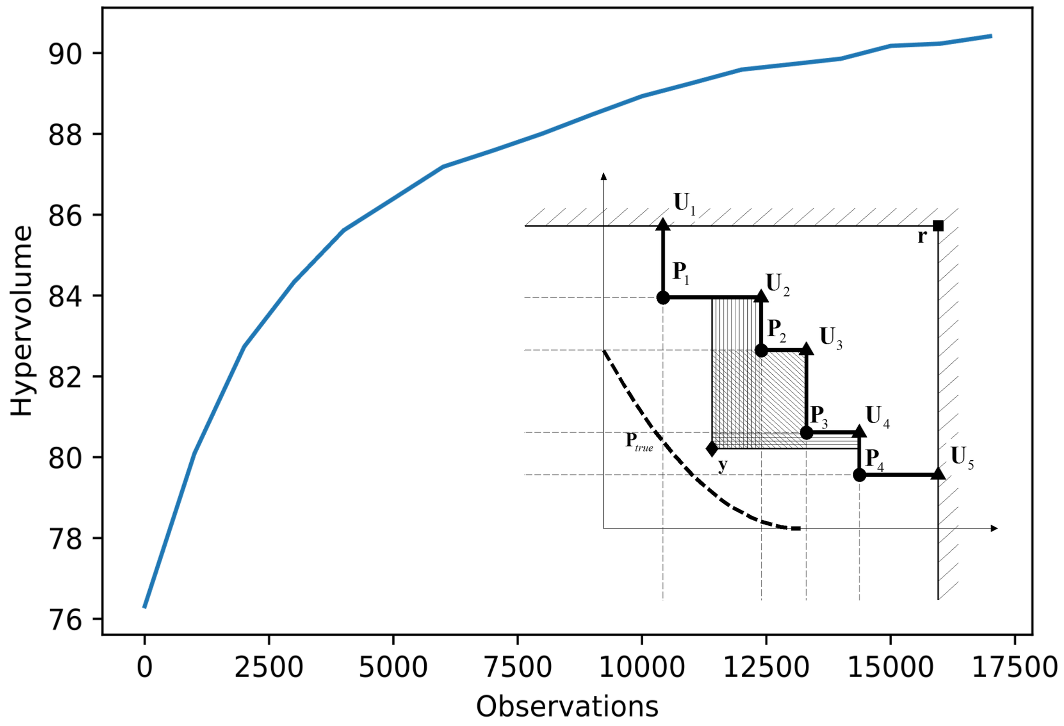
Genetic Optimization of PF

NSGA-II



https://www.strong.io/blog/evolutionary-optimization

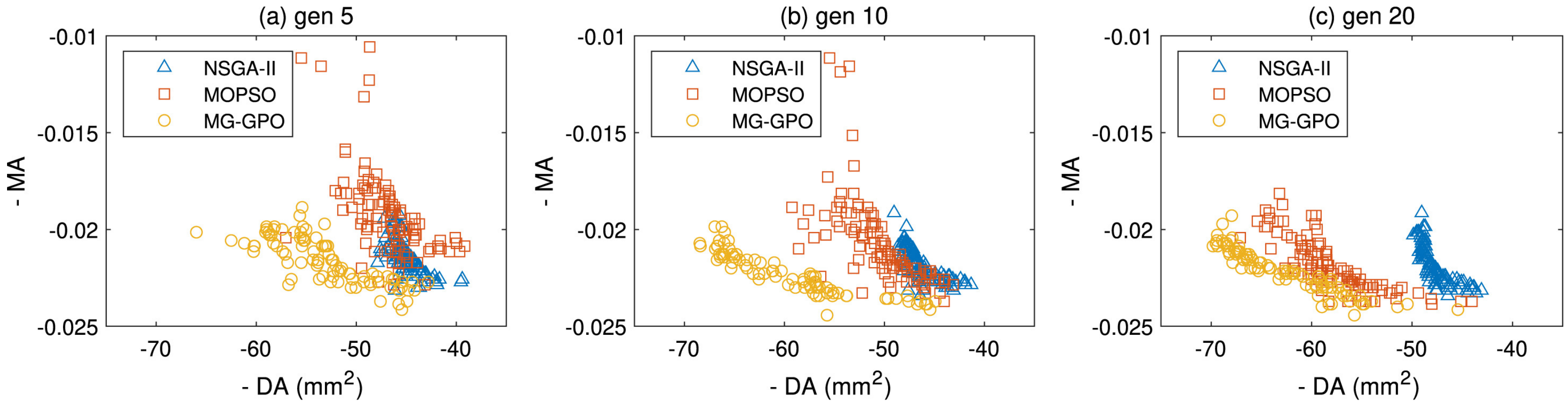
175 generations,100 individuals



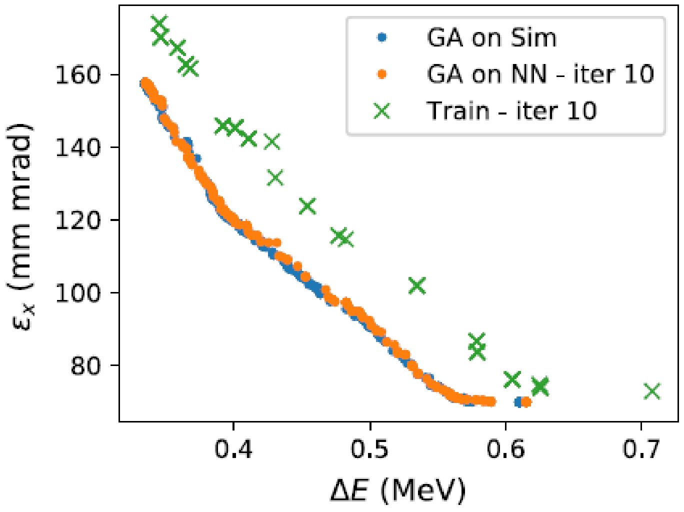
At 5 seconds per observation, optimization time > 24 hrs

Hybrid Genetic Optimization at SLAC

NSGA-II + Gaussian processes to evaluate GA created candidates



Song, Minghao, et al. NIMA (2020): 164273.

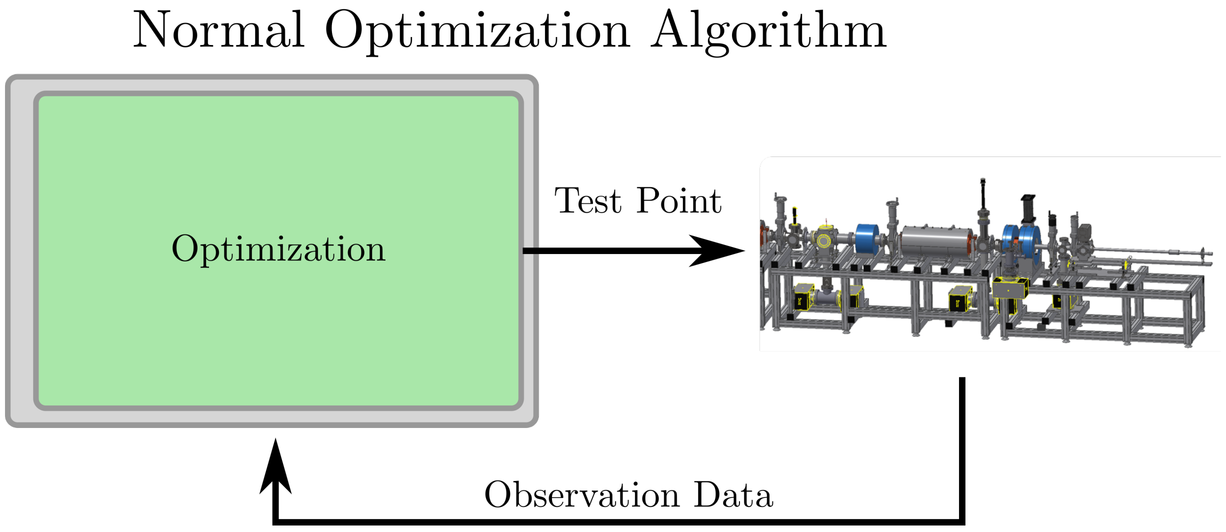


Edelen, Auralee, et al. *PRAB* 23.4 (2020): 044601.

Iterated neural network optimized by NSGA-II to propose ideal points

Large improvements in optimization speedup, but both methods rely on **batch observations**, suited best for **parallel optimization**

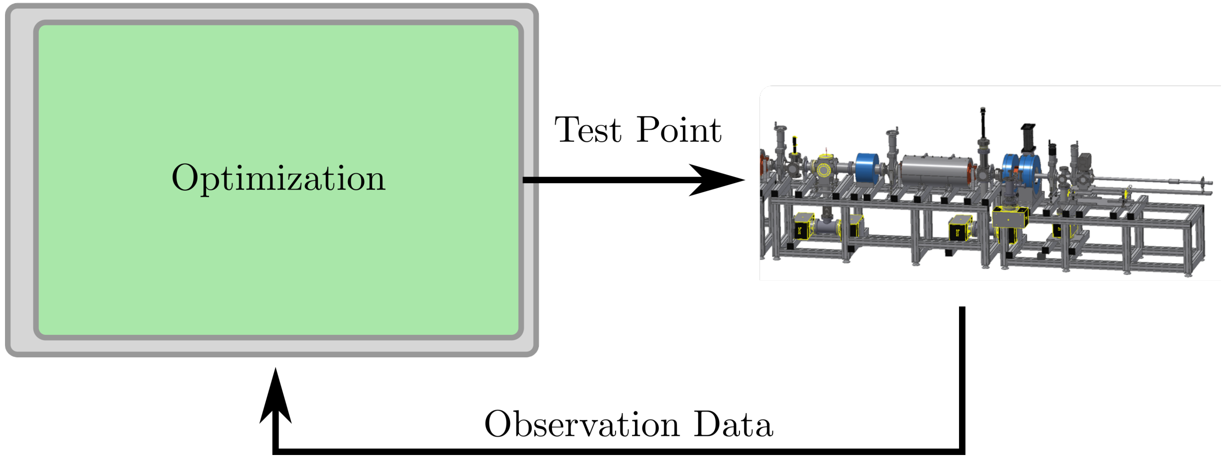
Accelerator Optimization w/Surrogates



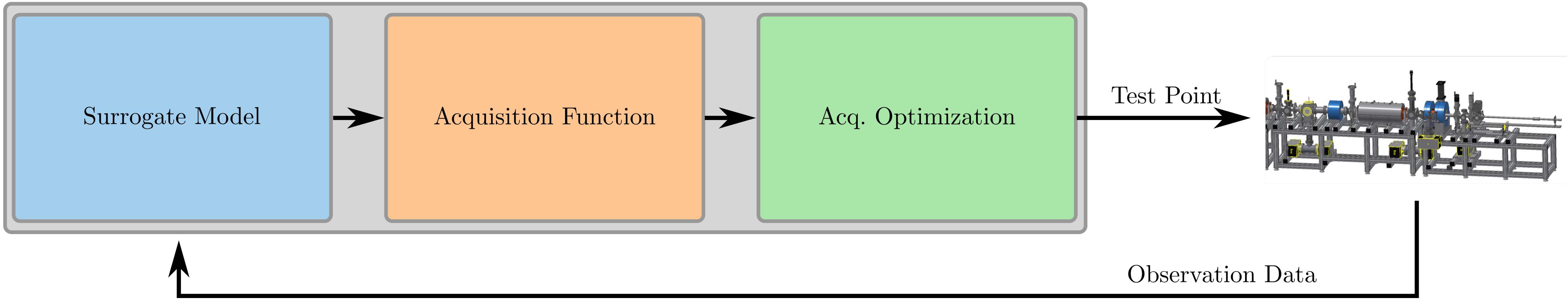
Can we do the same optimization with **less observations?**
Need an **efficient** method!

Accelerator Optimization w/Surrogates

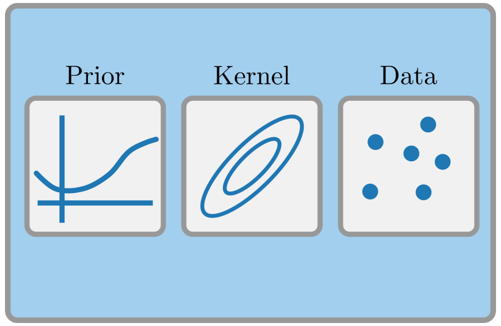
Normal Optimization Algorithm

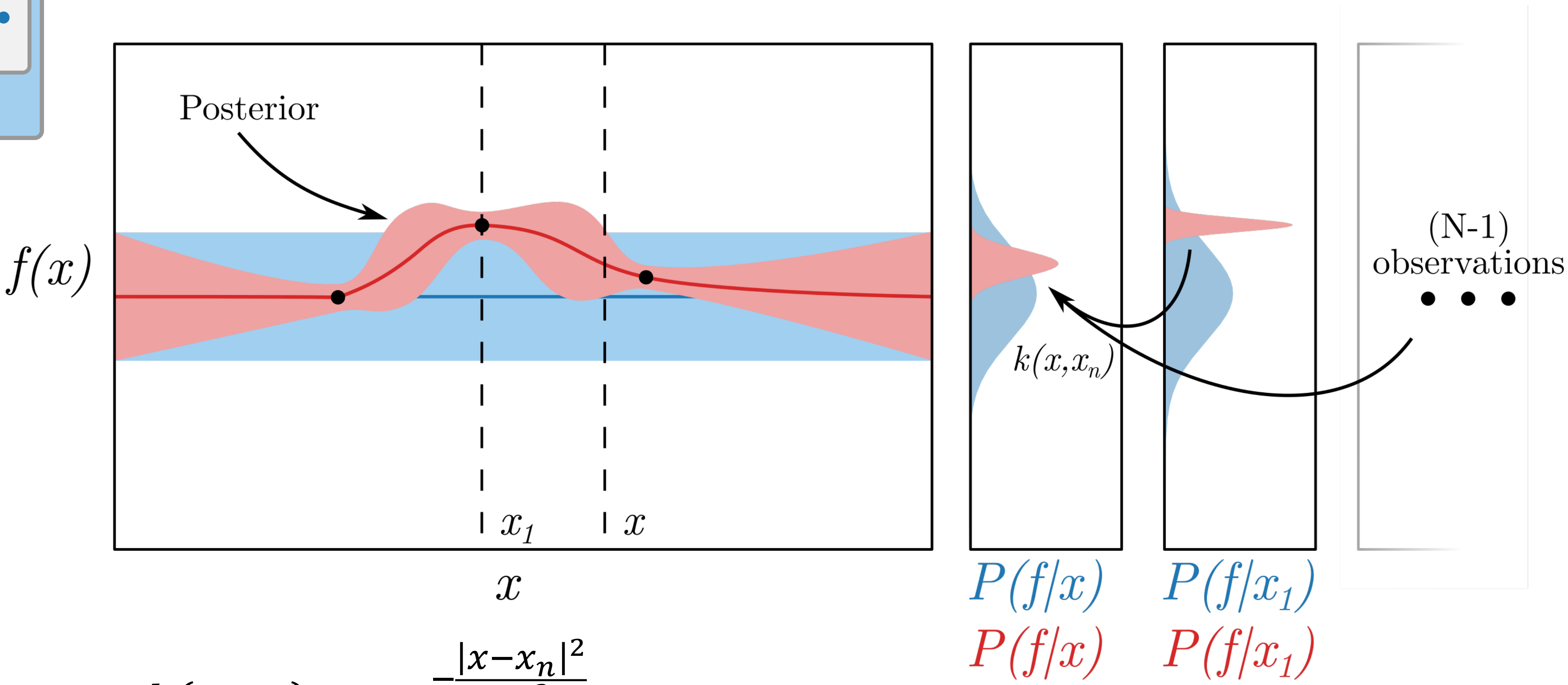


Bayesian Optimization Algorithm



Gaussian Processes as a Surrogate

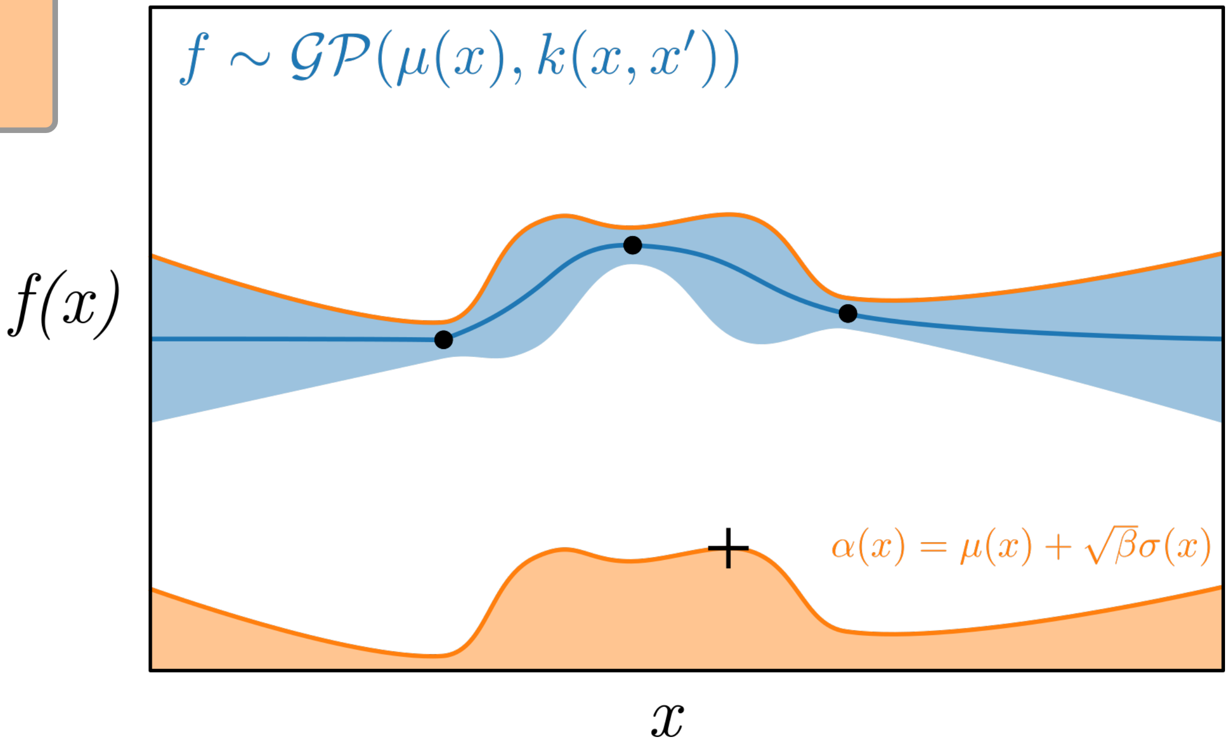




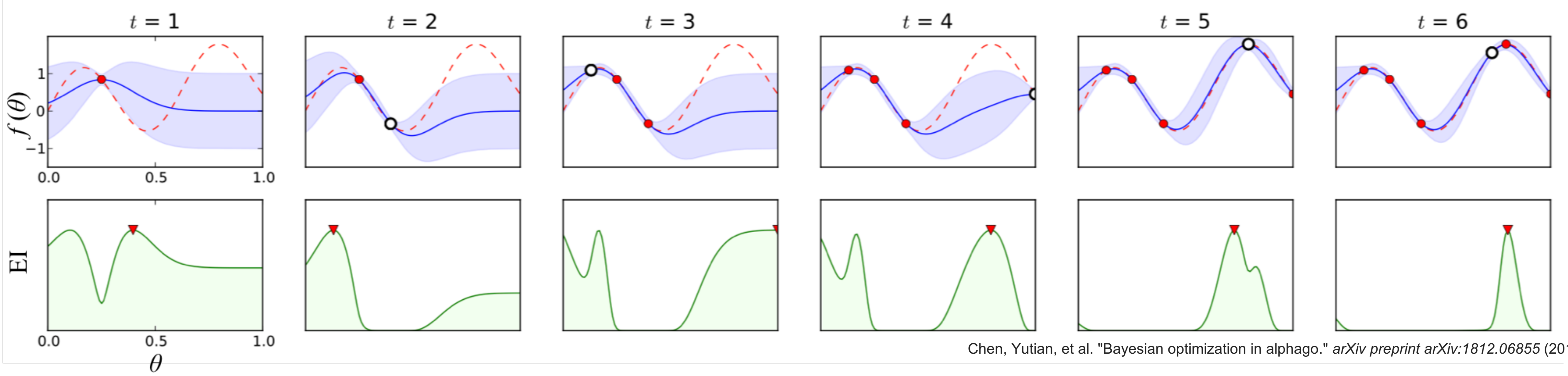
$$k(x, x_n) = \sigma e^{-\frac{|x-x_n|^2}{2\lambda^2}}$$

Gaussian Processes for Optimization

Acquisition Function



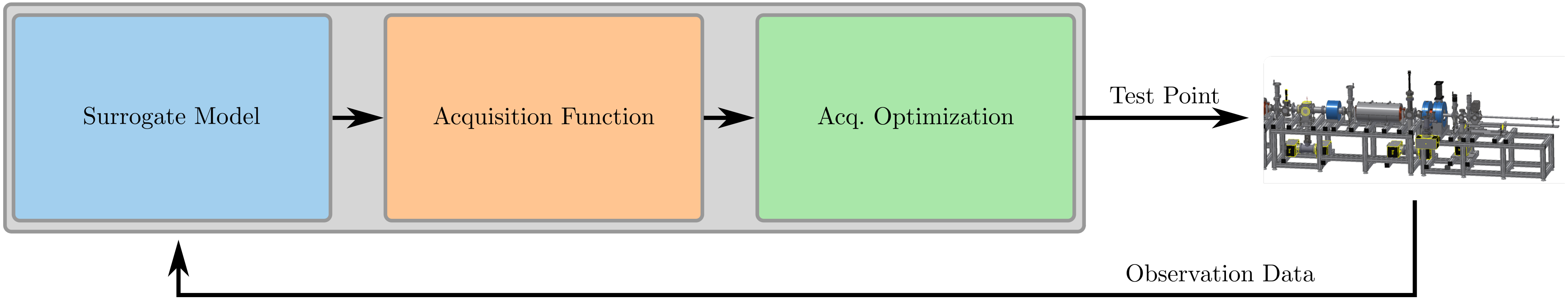
Gaussian Processes for Optimization



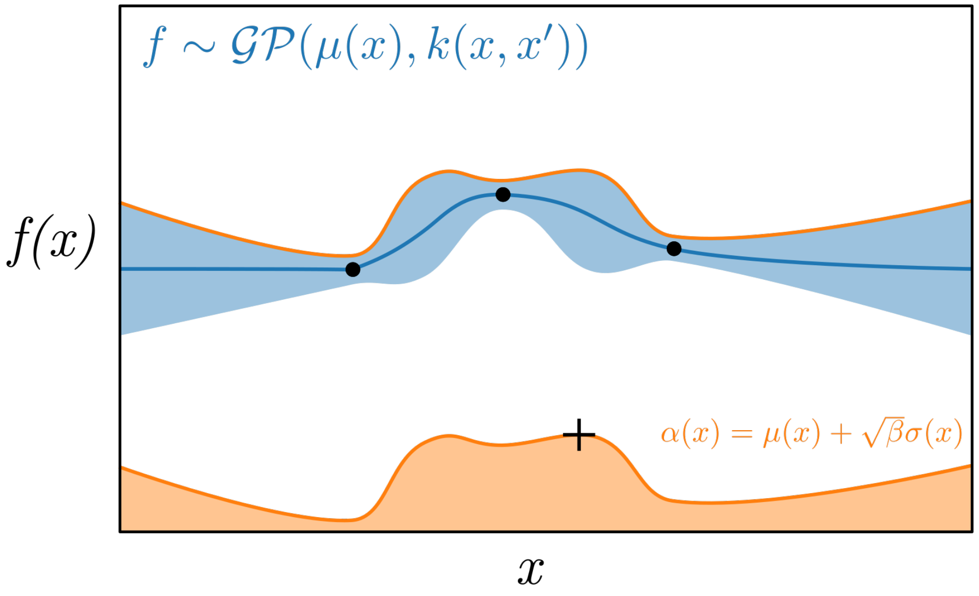
Chen, Yutian, et al. "Bayesian optimization in alphago." *arXiv preprint arXiv:1812.06855* (2018).

Gaussian Processes for Optimization

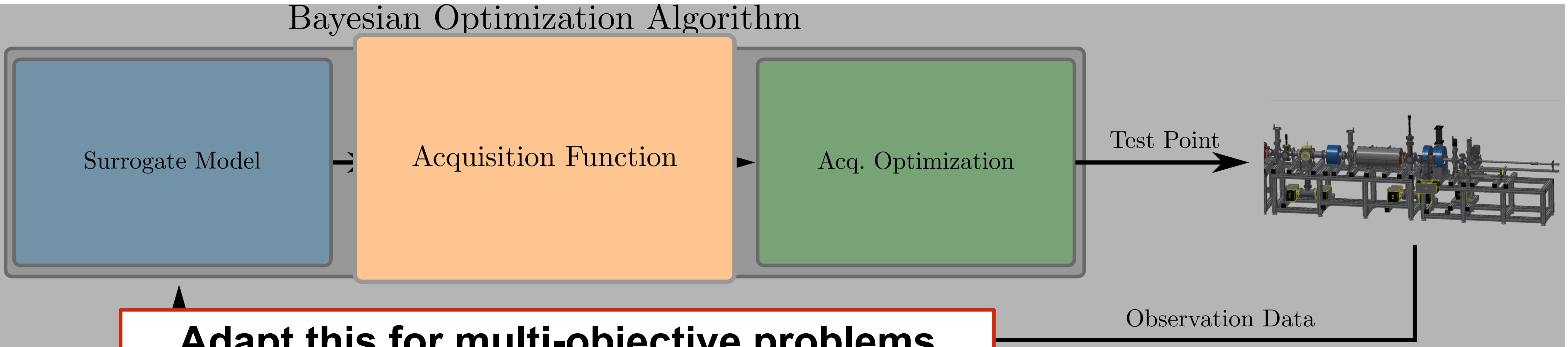
Bayesian Optimization Algorithm



Benefits:
- Specify tradeoff between **exploration** and **exploitation**
- Inherently **improves model accuracy** in **regions of interest**
- Enables parallel optimization strategies

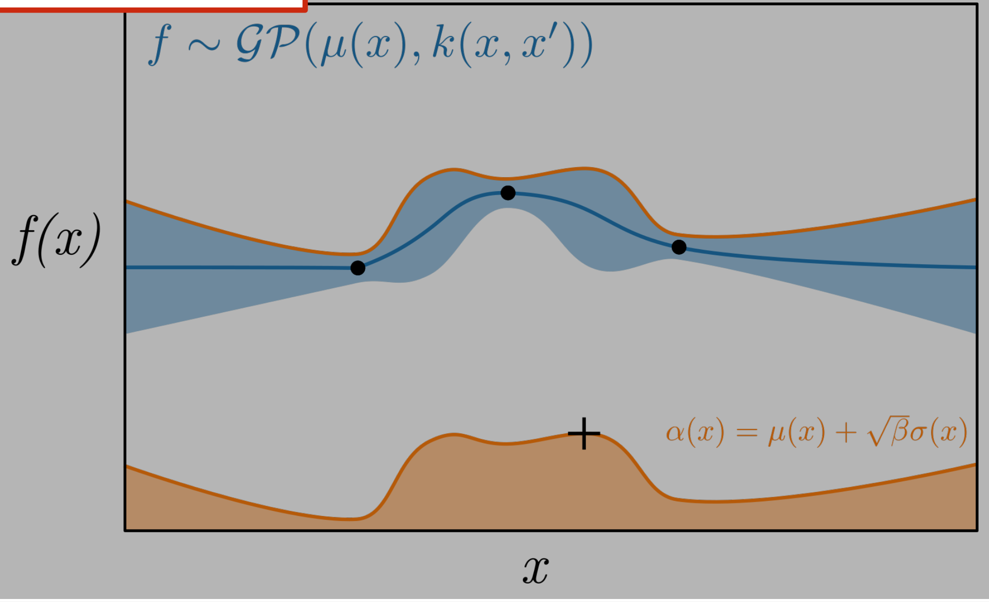


Gaussian Processes for Optimization



Adapt this for multi-objective problems

Benefits:
- Specify tradeoff between **exploration** and **exploitation**
- Inherently **improves model accuracy** in **regions of interest**
- Enables parallel optimization strategies

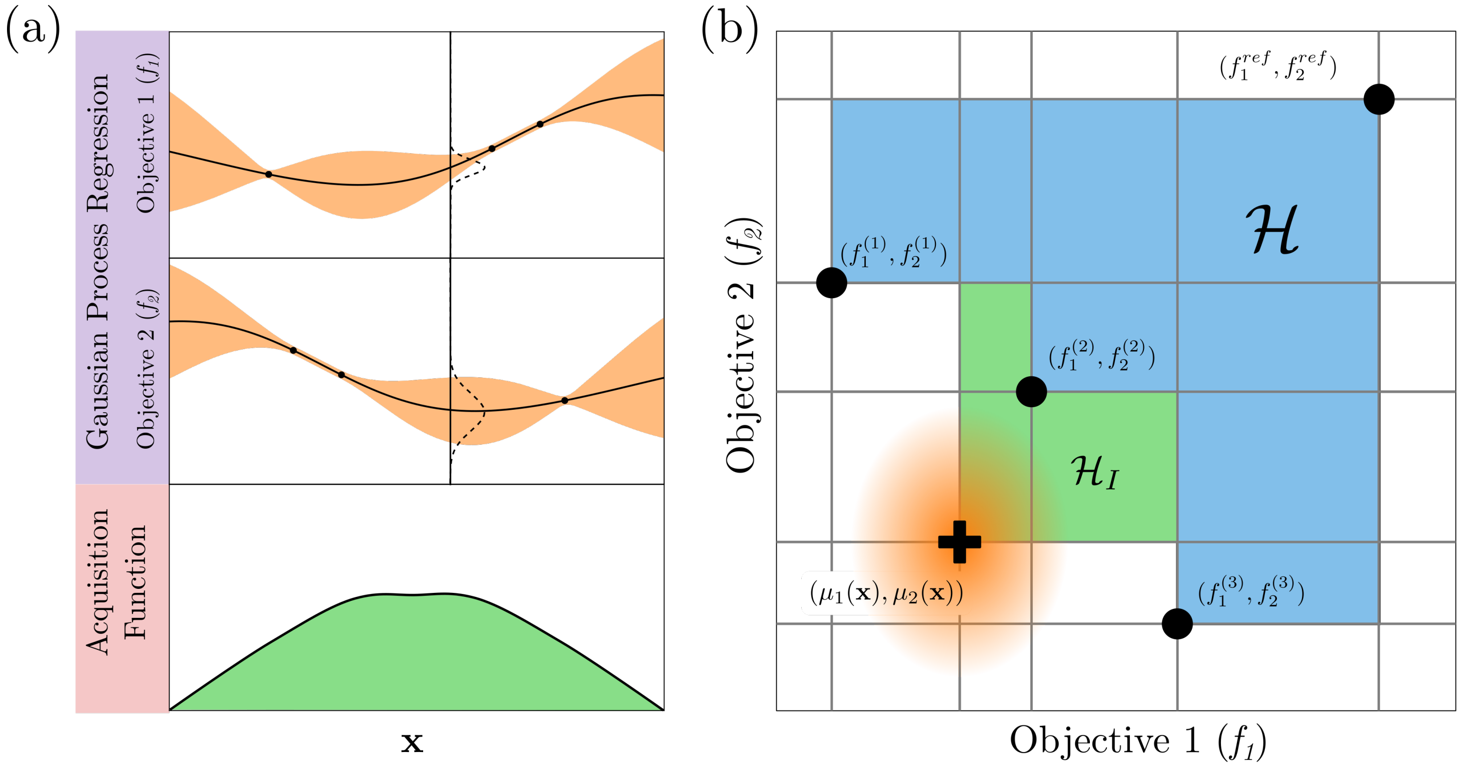


Extension to Multiple Objective Optimization

We wish to find the set of **Pareto-optimal points** in objective space by **maximizing** the contained **Pareto front hypervolume**

- Each objective has a GP surrogate model
- Using the surrogates we calculate the **Expected Hypervolume Improvement (EHVI)** as a function of the input
- Find a point that maximizes the EHVI and use as our next measurement point

This allows us to find the Pareto front **w/ a small number of measurements** unlike genetic or swarm optimization methods

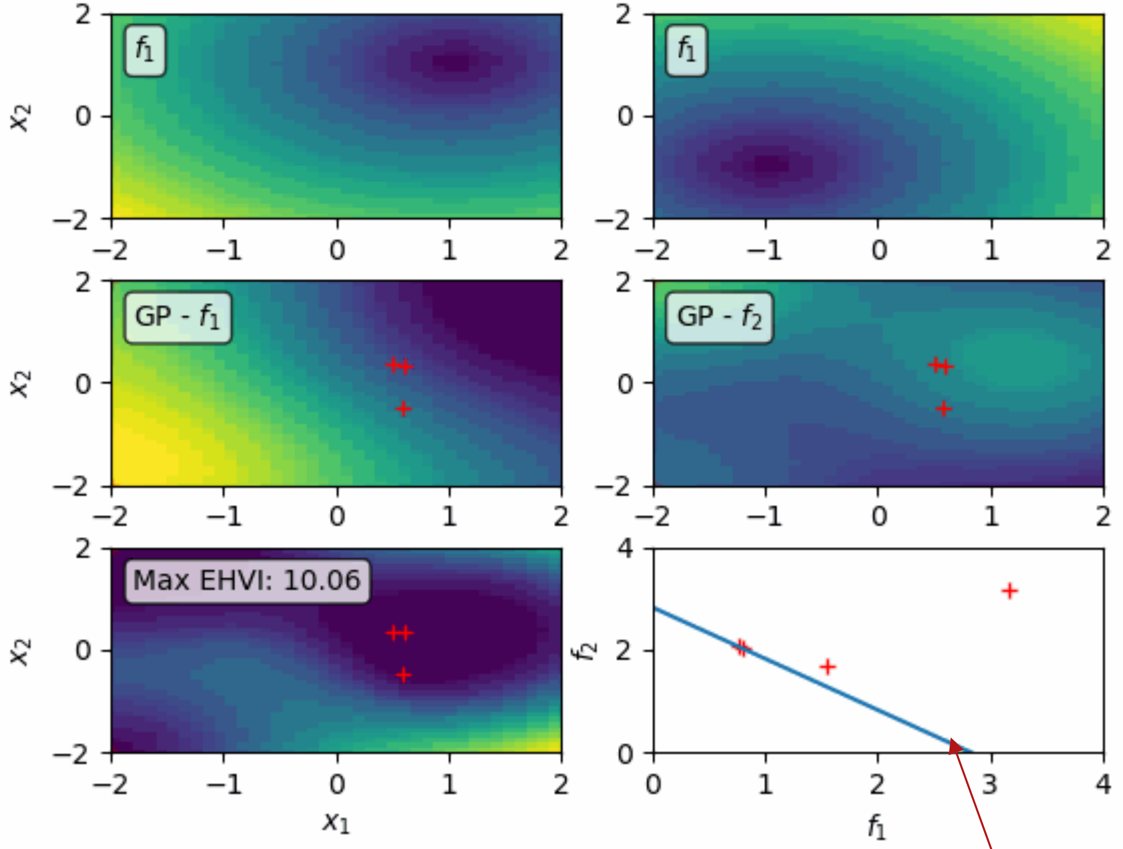


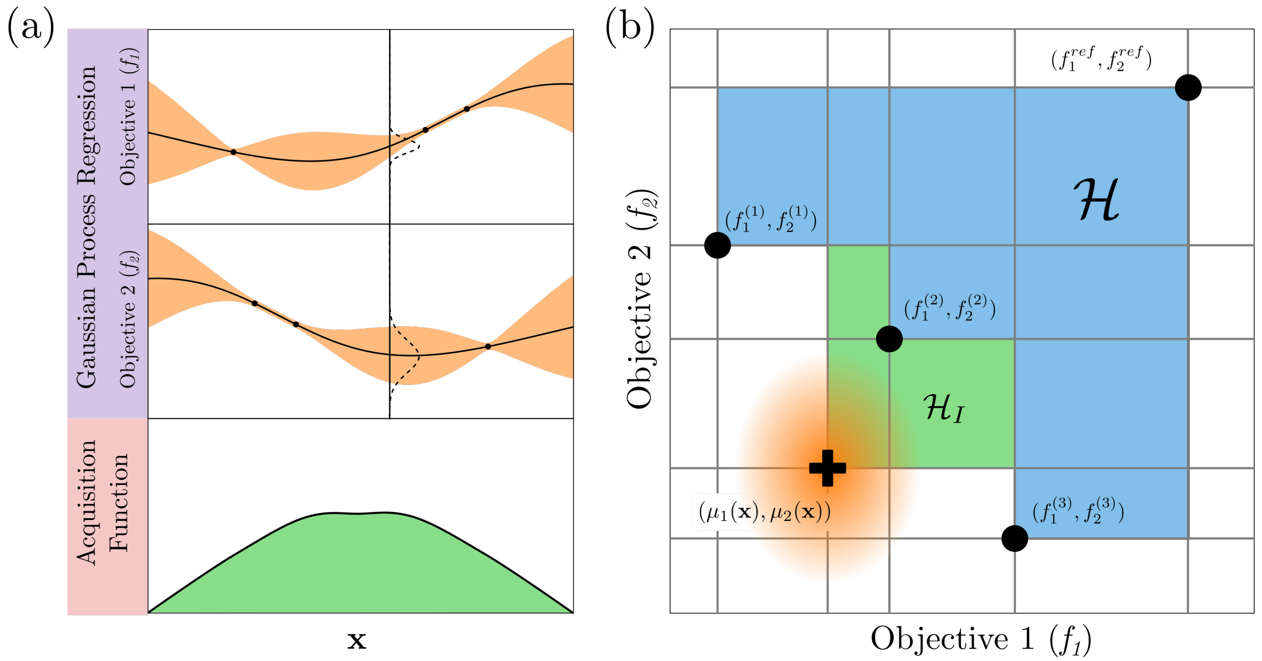
$$\alpha_{EHVI}(\boldsymbol{\mu},\boldsymbol{\sigma},\mathcal{P},\boldsymbol{r}) := \int_{\mathbb{R}^P} \mathrm{HVI}(\mathcal{P},\boldsymbol{y},\boldsymbol{r}) \cdot \boldsymbol{\xi}_{\mu,\boldsymbol{\sigma}}(\boldsymbol{y}) d\boldsymbol{y}$$

Yang, Kaifeng, et al. *Swarm and evolutionary computation* 44 (2019): 945-95

Two Objective Example Optimization

$f_1 = |\boldsymbol{x} - (1,1)|^2$ $f_2 = |\boldsymbol{x} - (-1,-1)|^2$





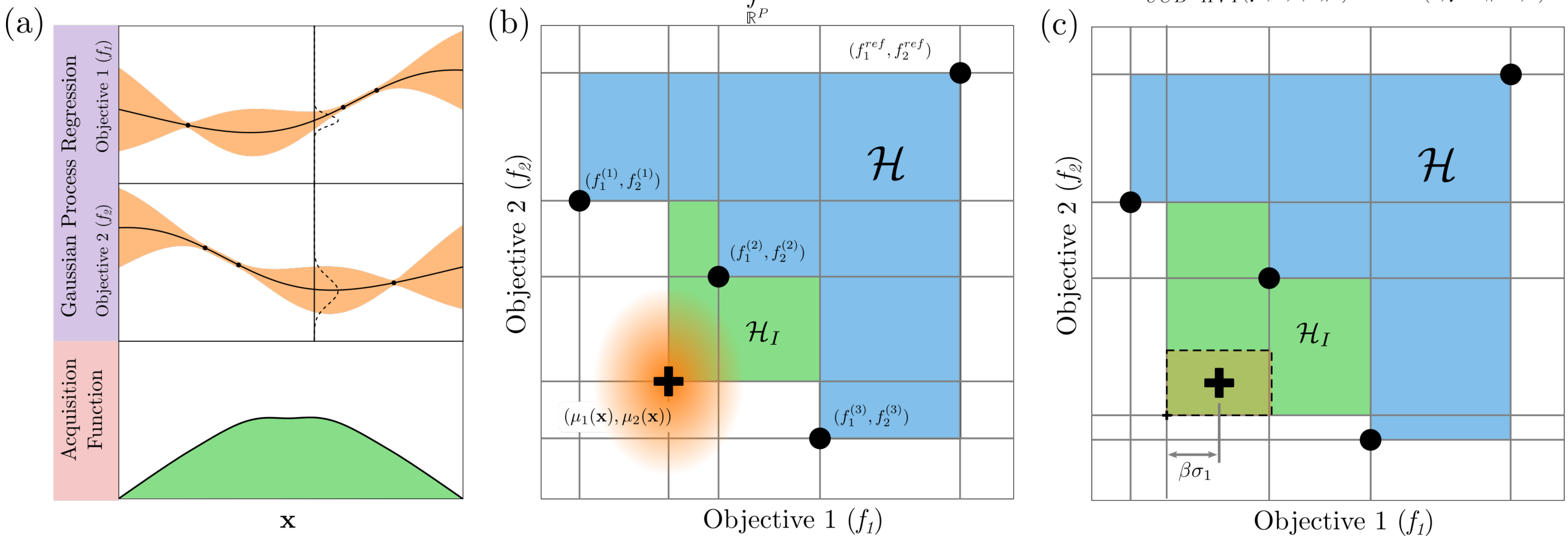
Red cross – observation points Analytical Pareto front

Alternative Hypervolume Improvement Metrics

Large number of objectives -> too expensive for EHVI calculation **~ O(N^D)**

$$\alpha_{EHVI}(\boldsymbol{\mu},\boldsymbol{\sigma},\mathcal{P},\boldsymbol{r}) := \int_{\mathbb{R}^P} \mathrm{HVI}(\mathcal{P},\boldsymbol{y},\boldsymbol{r}) \cdot \boldsymbol{\xi}_{\mu,\sigma}(\boldsymbol{y})\, d\boldsymbol{y}$$

$$\alpha_{UCB-HVI}(\boldsymbol{\mu},\boldsymbol{\sigma},\mathcal{P},\boldsymbol{r},\beta) := \mathrm{HVI}(\mathcal{P},\boldsymbol{\mu}-\sqrt{\beta}\boldsymbol{\sigma},\boldsymbol{r})$$

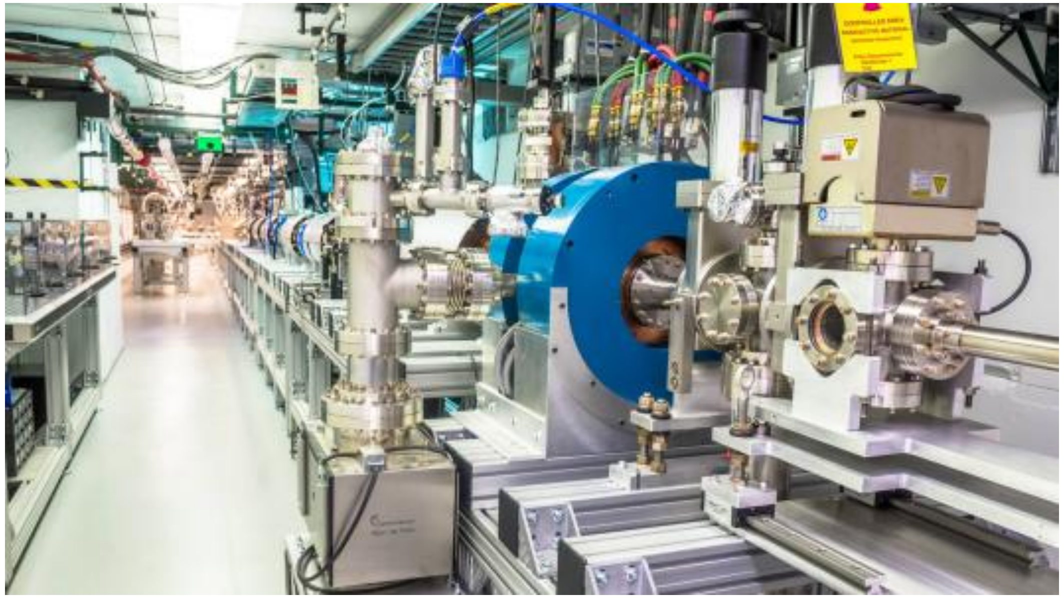


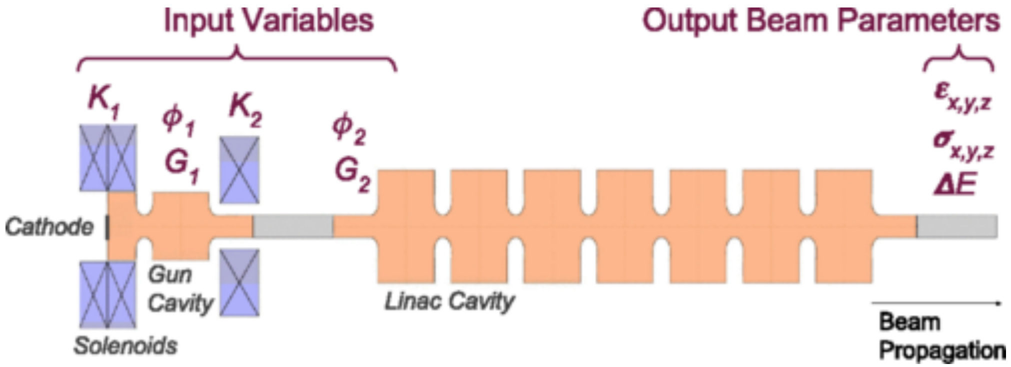
Slow! >>

Fast! >>>>>>>>>>

While, Lyndon, Lucas Bradstreet, and Luigi Barone. *IEEE Transactions on Evolutionary Computation* 16.1 (2011): 86-95.

Photoinjector Optimization





For online photoinjector optimization we wish to simultaneously:

- Minimize **emittances (3x)**
- Minimize **bunch sizes (3x)**
- Minimize **energy spread (1x)**

7 objectives

Tuning knobs:

- Solenoid strengths **(2x)**
- RF Amplitudes **(2x)**
- RF Phases **(2x)**

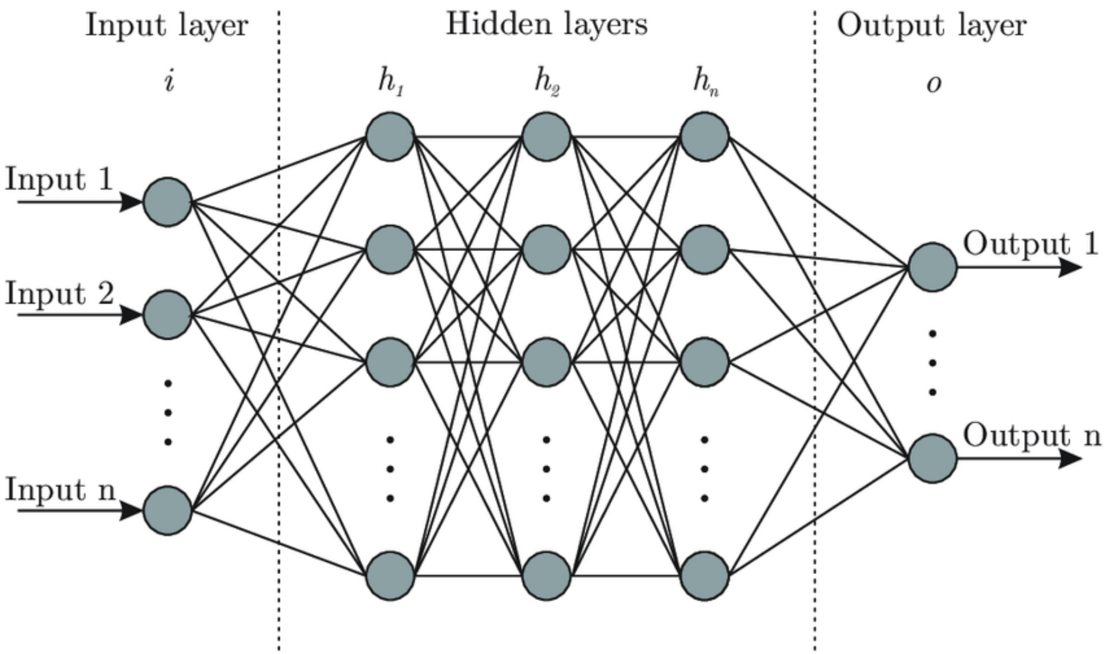
6 input parameters

Edelen, Auralee, et al. PRAB 23.4 (2020): 044601.

Simulated Photoinjector Optimization

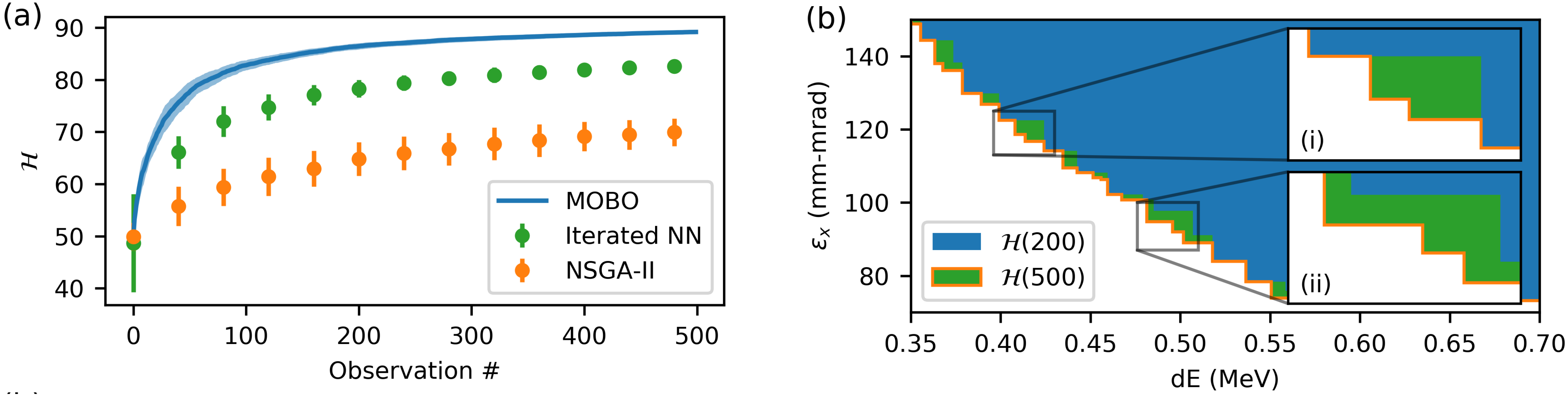
OPAL simulation of AWA photoinjector
6 mins on HPC cluster

NN surrogate of AWA photoinjector
< 1 sec on a laptop



Edelen, Auralee, et al. PRAB 23.4 (2020): 044601.

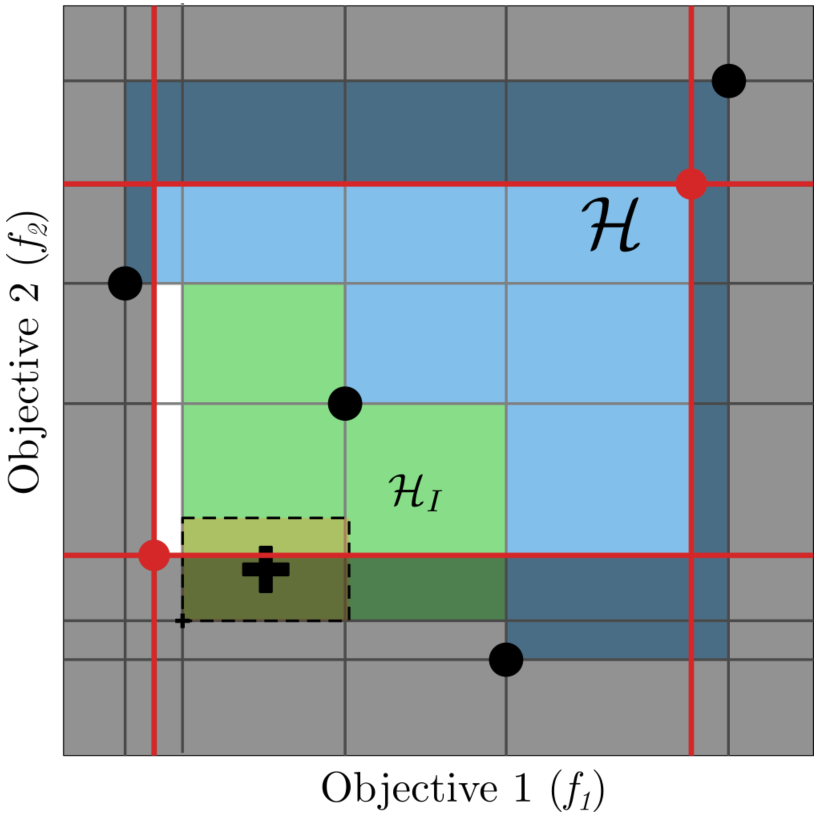
Simulated Photoinjector Optimization

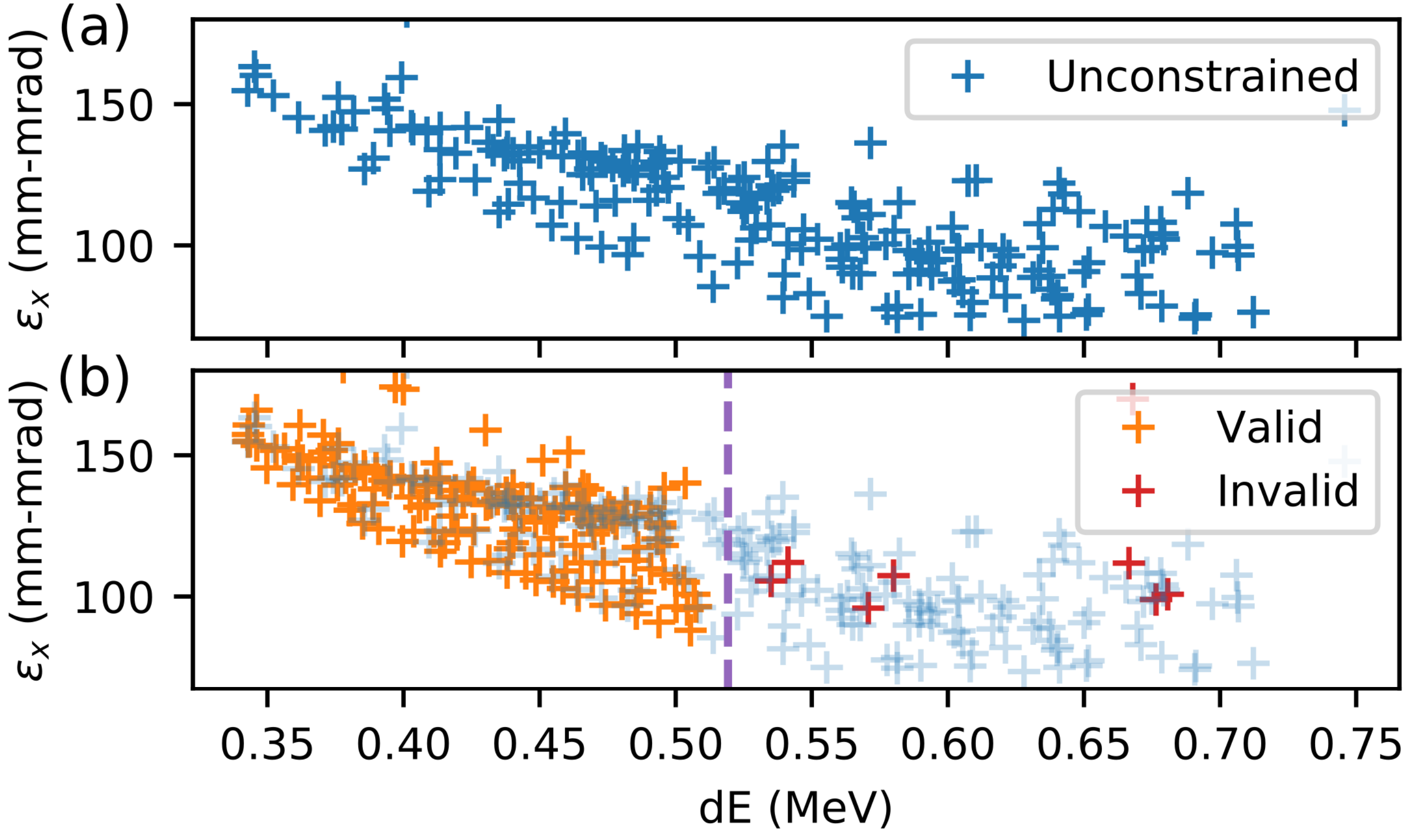


- 10 optimization runs
- 20 initial points each
- Pk hypervolume ~ 90 in < 500 steps (NSGA-II ~ 17.5k) factor of 35x speedup, tuned in < 45 mins!

Adding Objective Preferences

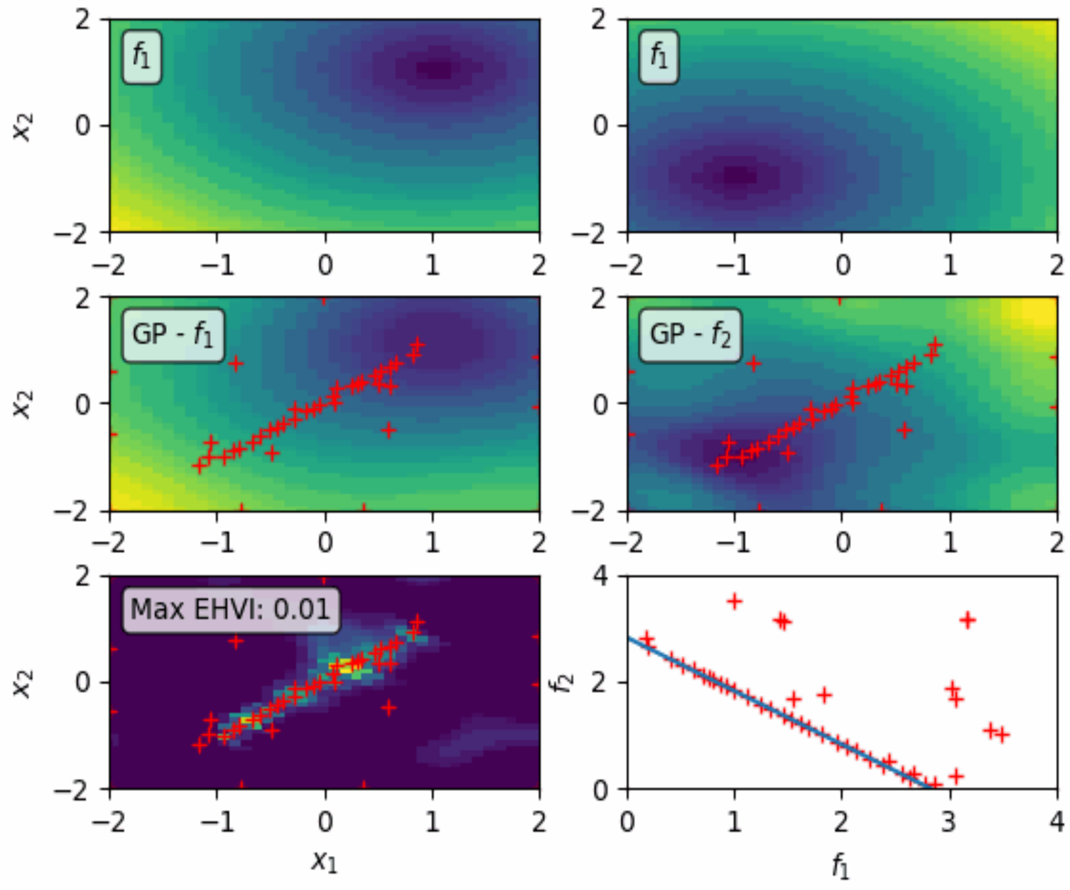
Preferences: only calculate HV inside objective region



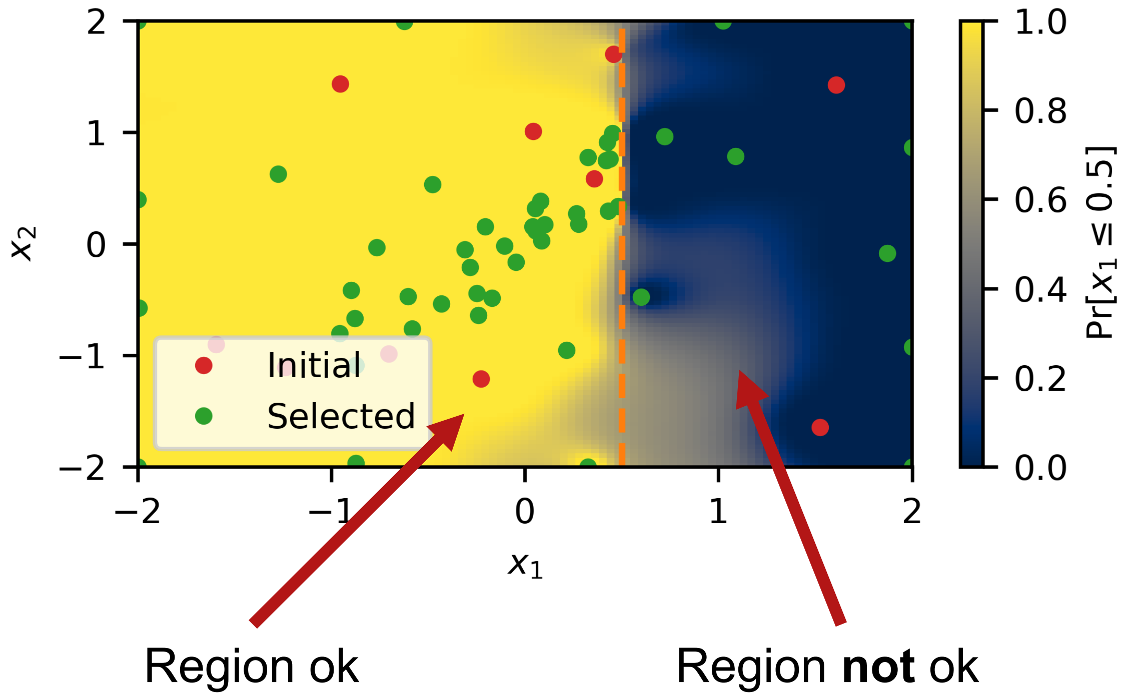


$$\alpha_{TUCB-HVI}(\boldsymbol{\mu},\boldsymbol{\sigma},\mathcal{P},\beta,\boldsymbol{A},\boldsymbol{B}) := \begin{cases} \mathrm{HVI}(\mathcal{P},\boldsymbol{y},\boldsymbol{B}) & \boldsymbol{y}\in\mathcal{T} \\ 0 & \text{otherwise} \end{cases}$$

Adding Constraints



Constraints: model probability that a constraint is satisfied



$$P_g(\boldsymbol{x}) := Pr[g(\boldsymbol{x}) \leq h] = \int_{-\infty}^{h} p(g(\boldsymbol{x})|\mathcal{D}_g)\, dg(\boldsymbol{x})$$

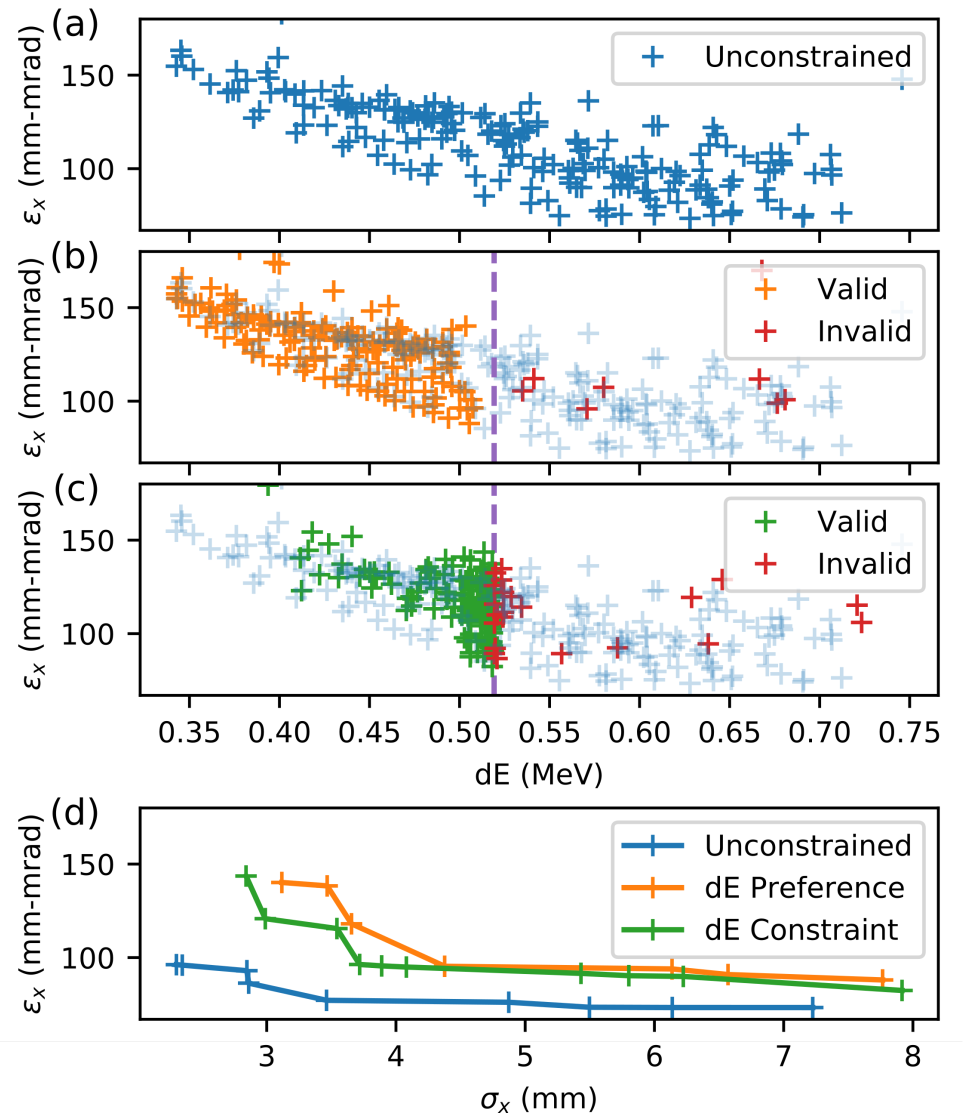
$$\widehat{\alpha}(\boldsymbol{x}) = \alpha(\boldsymbol{x}) P_g(\boldsymbol{x})$$

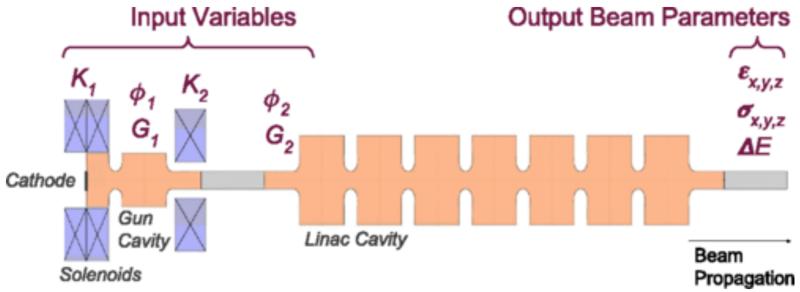
Adding Preferences vs. Constraints

Control

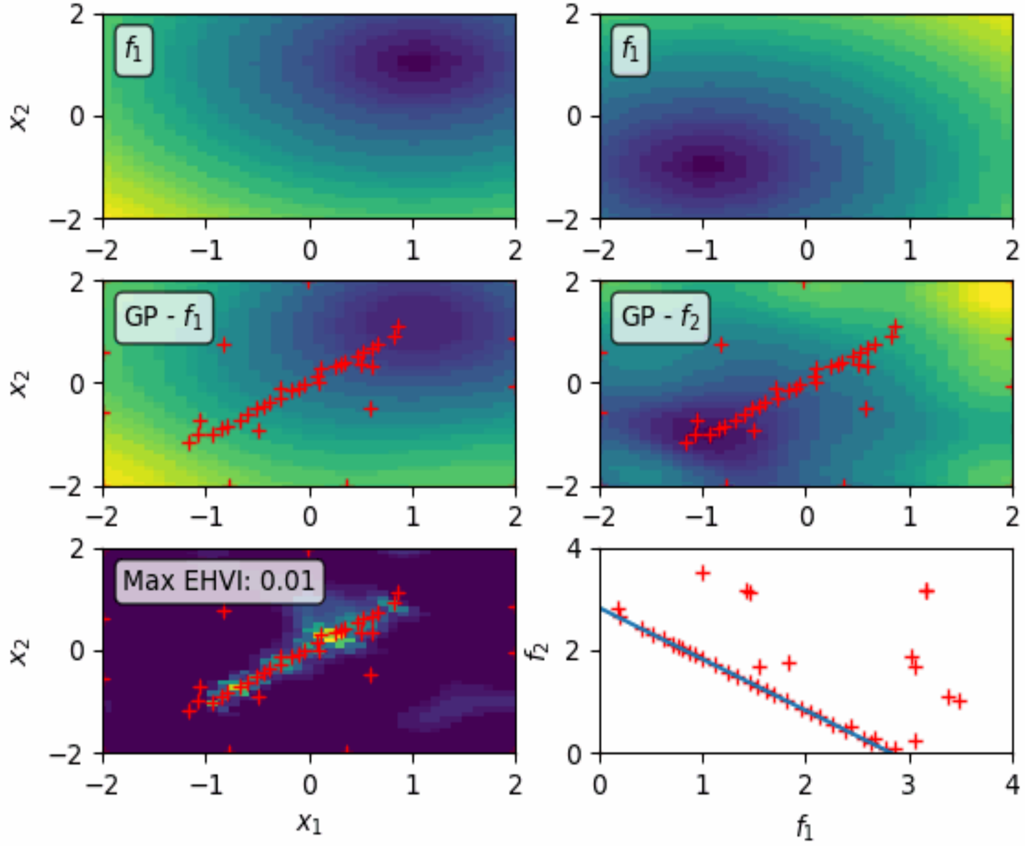
Preference: dE < 0.52 MeV

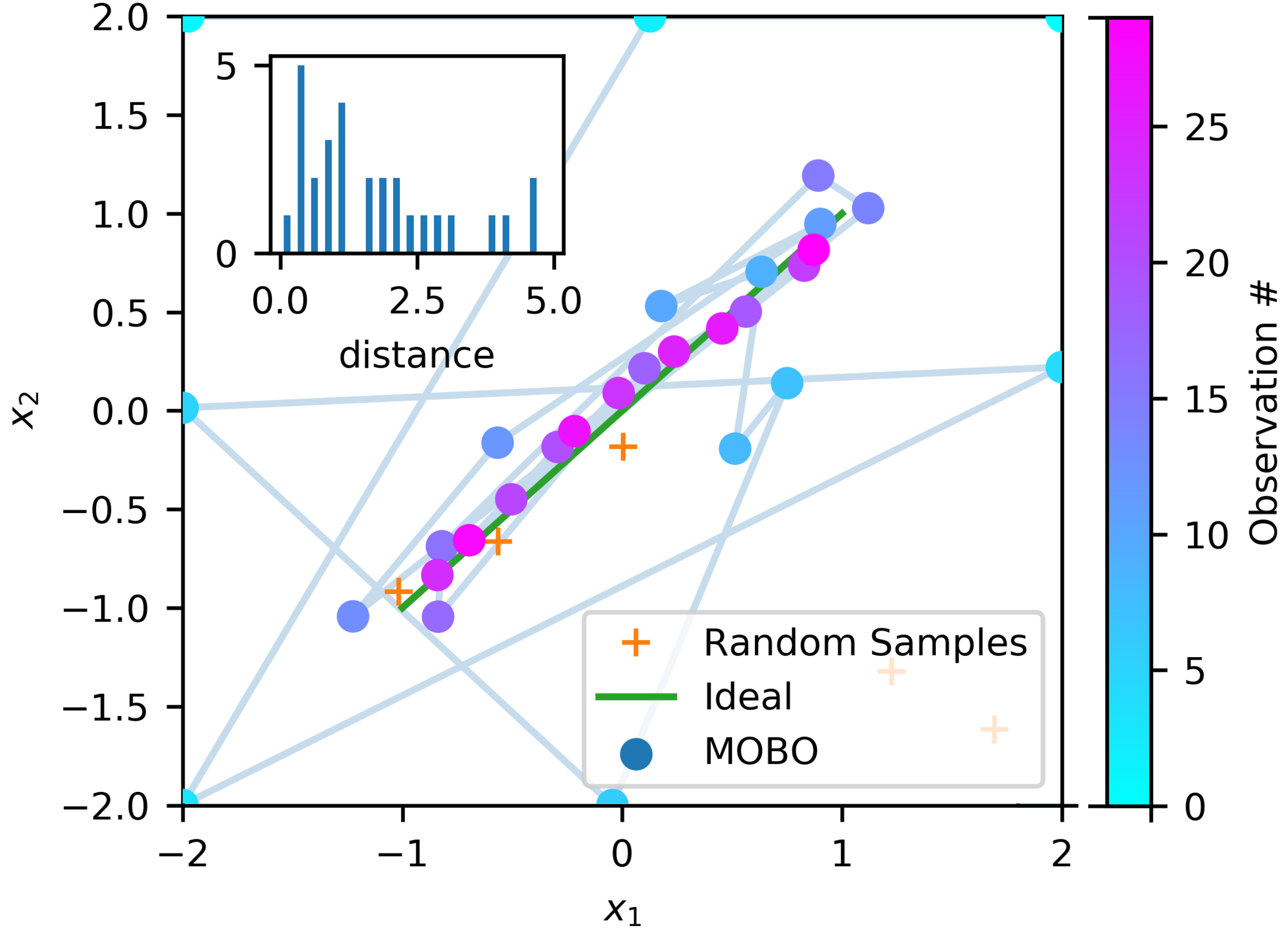
Constraint: dE < 0.52 MeV



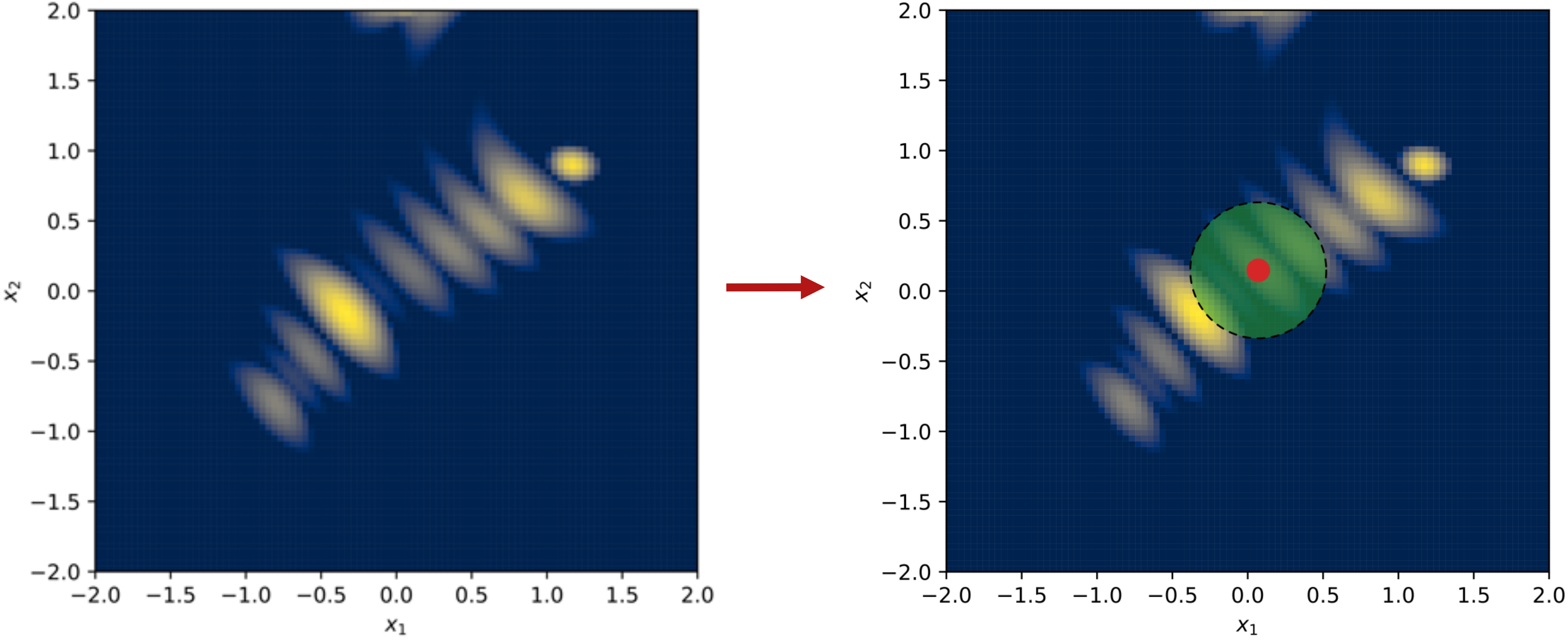


Smooth (Localized) Exploration



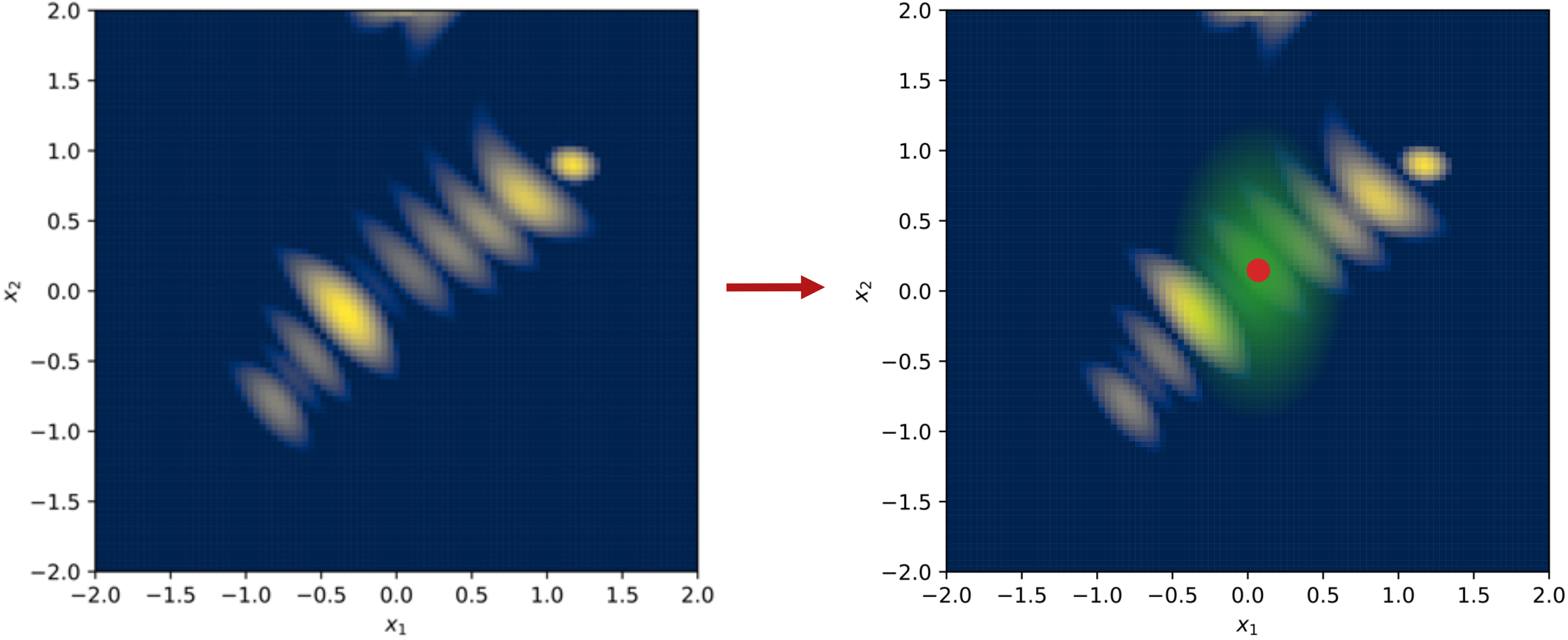


Smooth (Localized) Exploration



Smooth (Localized) Exploration

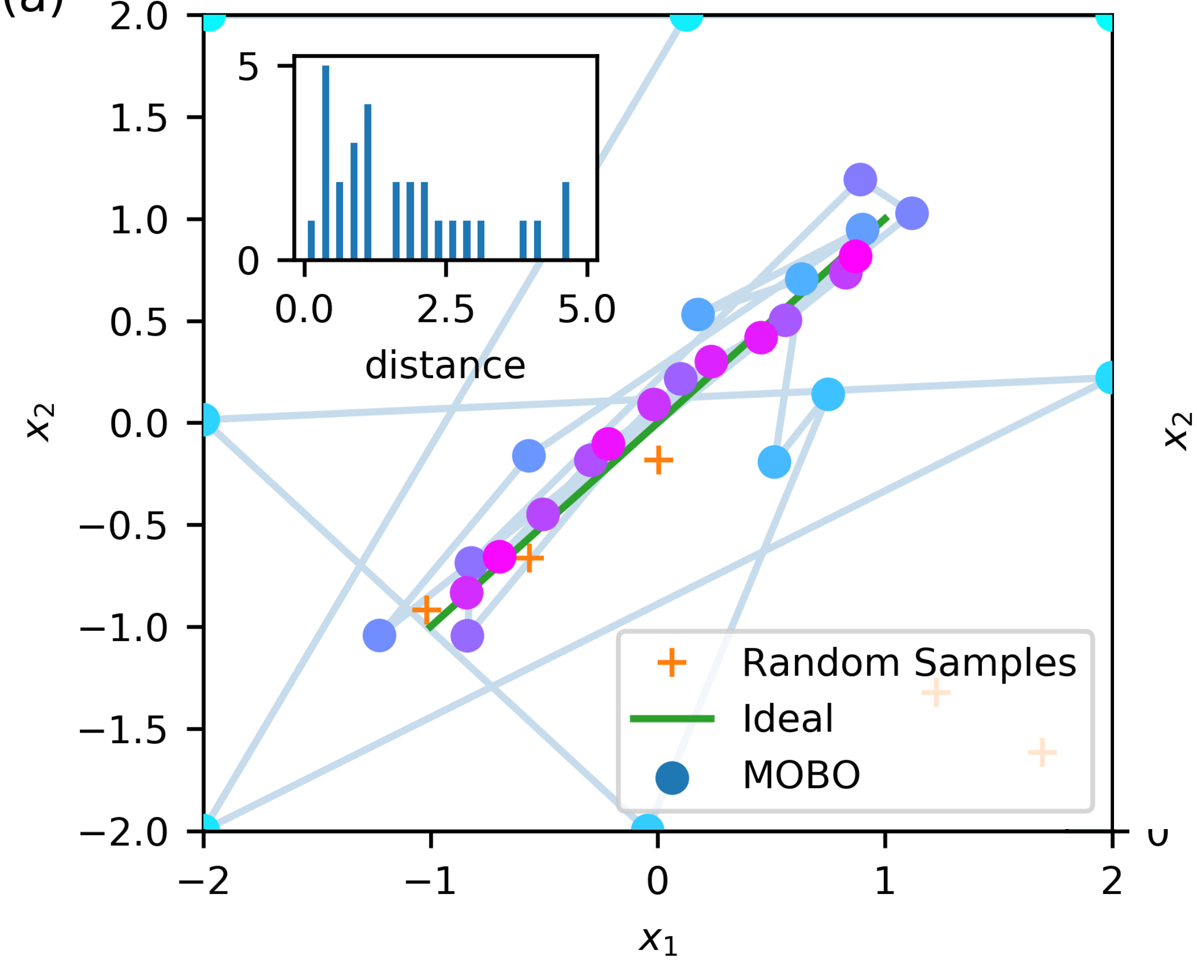
$$\tilde{\alpha}(\boldsymbol{x},\boldsymbol{x}_0)=\alpha(\boldsymbol{x})\exp\left[-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_0)^T\boldsymbol{\Sigma}(\boldsymbol{x}-\boldsymbol{x}_0)\right]$$

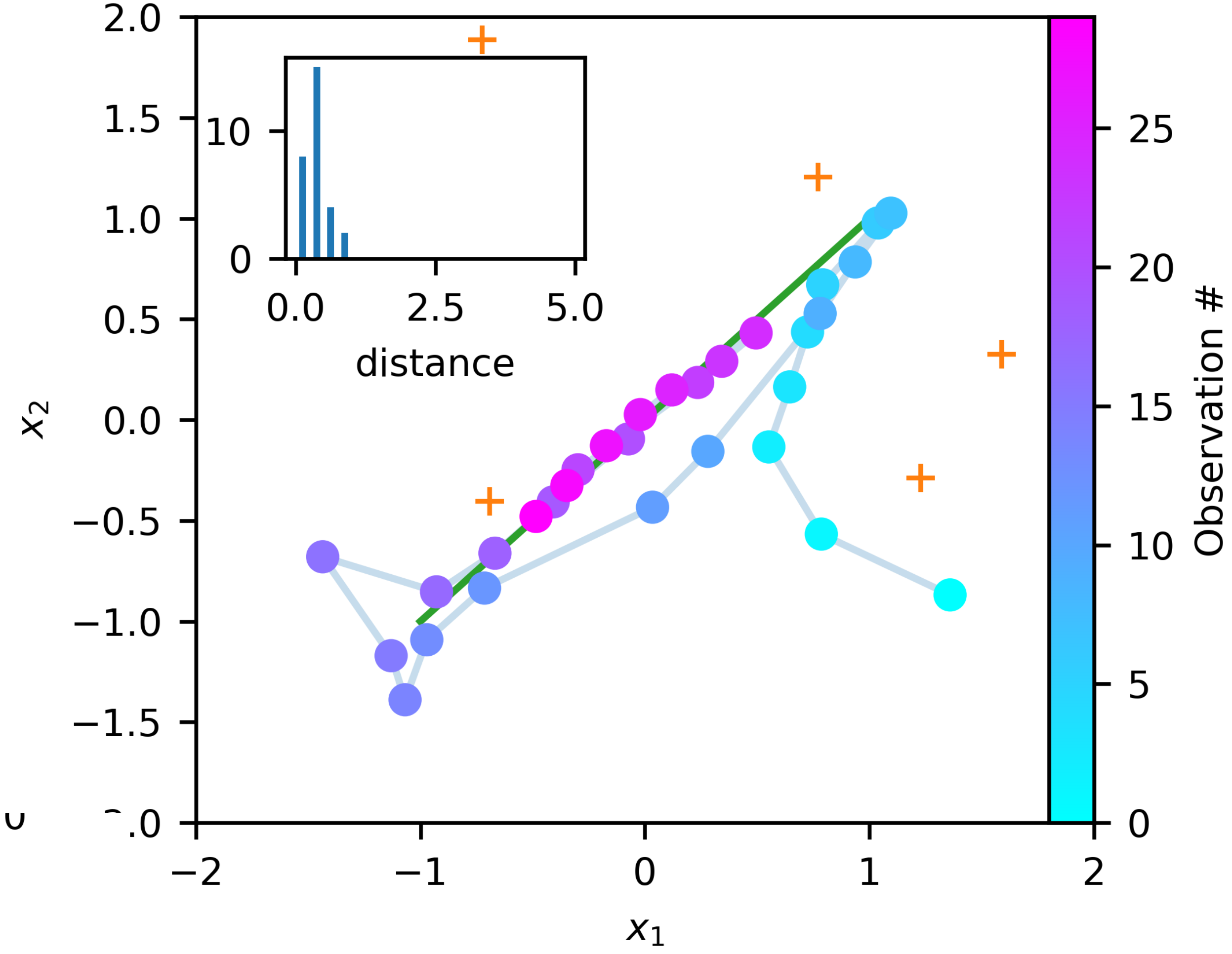


Smooth (Localized) Exploration

$$\widetilde{\alpha}(\boldsymbol{x},\boldsymbol{x}_0)=\alpha(\boldsymbol{x})\exp\left[-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_0)^T\boldsymbol{\Sigma}(\boldsymbol{x}-\boldsymbol{x}_0)\right]$$

(a)



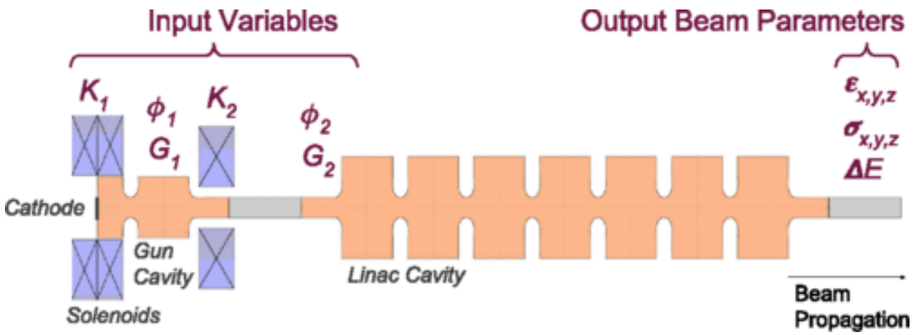


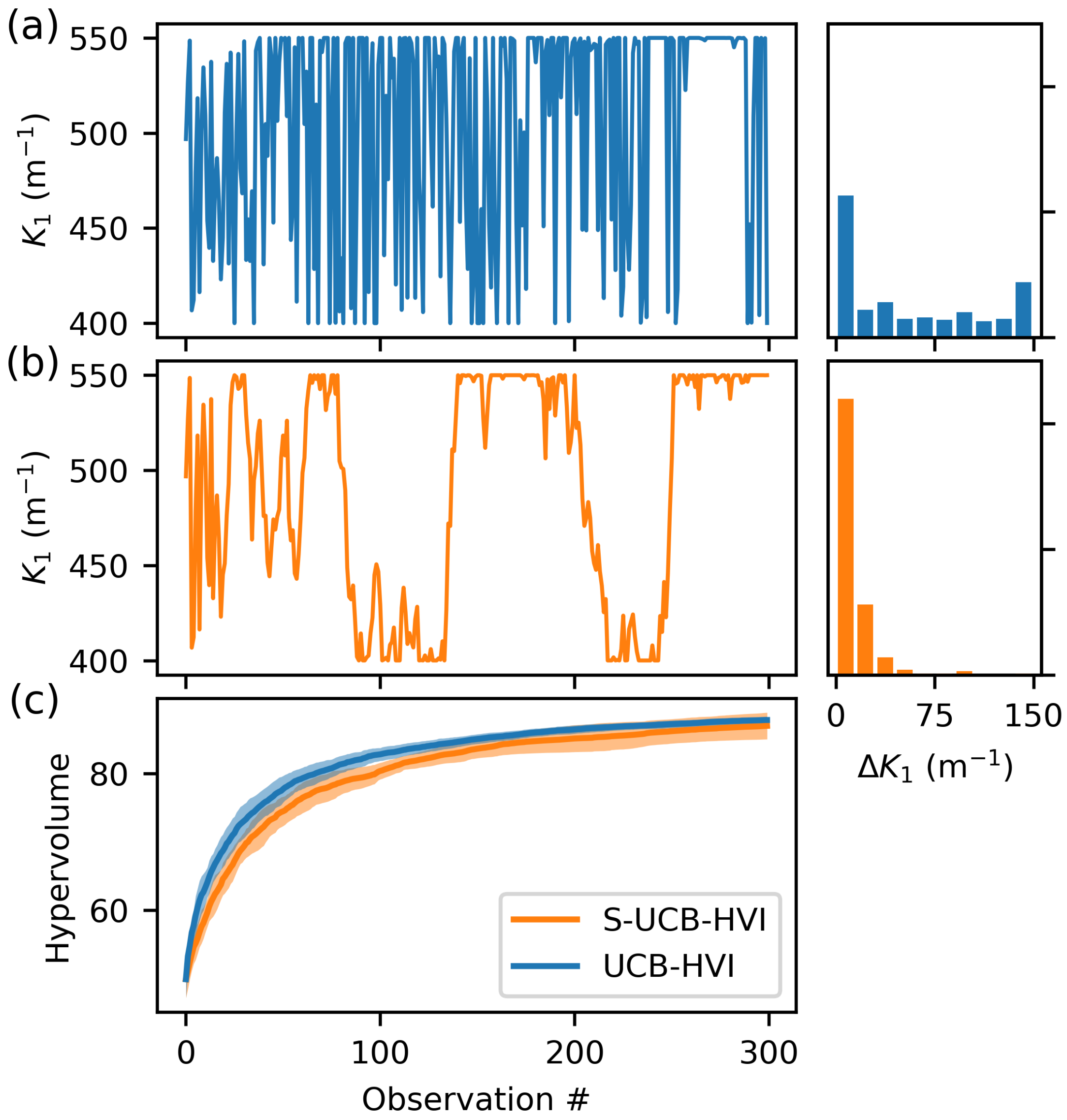
Smooth Exploration (AWA)

No modification

Localized acquisition function

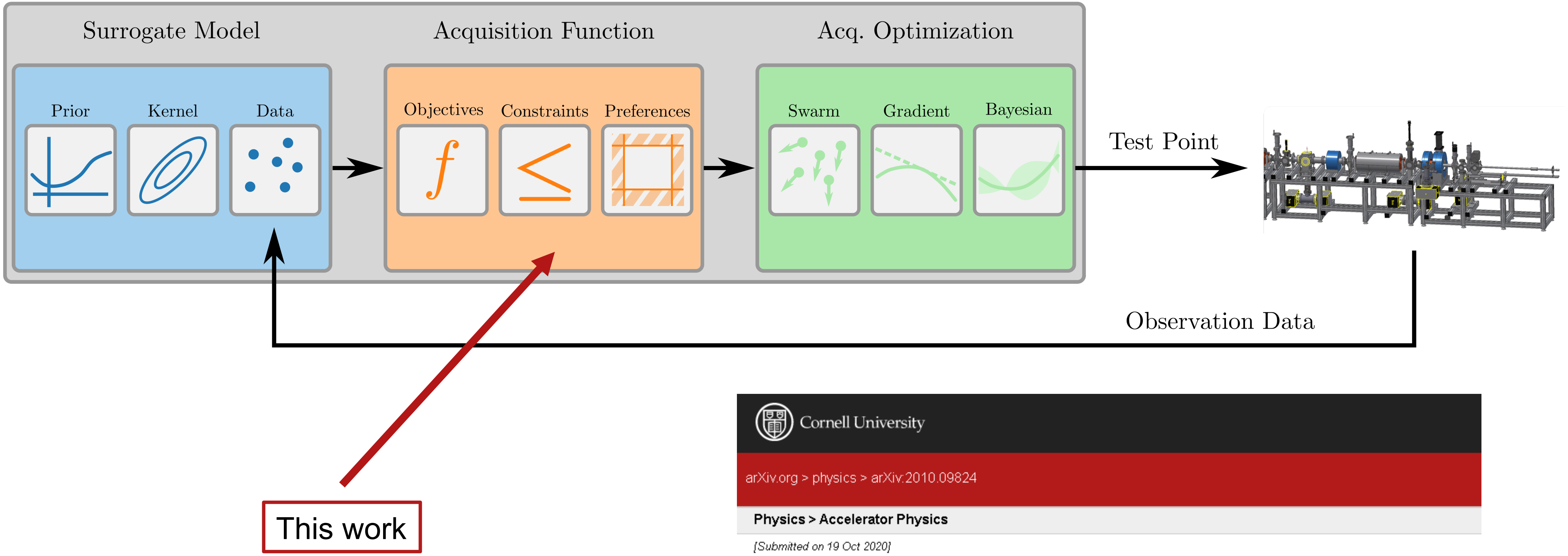
$$\tilde{\alpha}(\boldsymbol{x},\boldsymbol{x}_0)=\alpha(\boldsymbol{x})\exp\left[-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_0)^T\boldsymbol{\Sigma}(\boldsymbol{x}-\boldsymbol{x}_0)\right]$$





Gaussian Processes for Accelerator Optimization

Bayesian Optimization Algorithm



Cornell University

arXiv.org > physics > arXiv:2010.09824

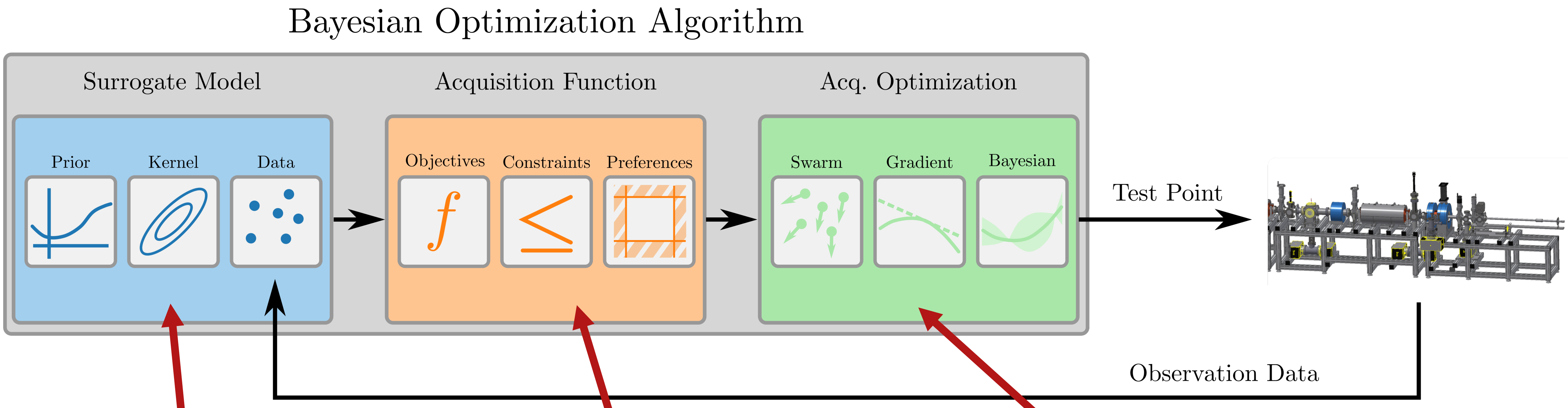
Physics > Accelerator Physics

[Submitted on 19 Oct 2020]

Multi-Objective Bayesian Optimization for Accelerator Tuning

Ryan Roussel, Adi Hanuka, Auralee Edelen

Gaussian Processes for Accelerator Optimization



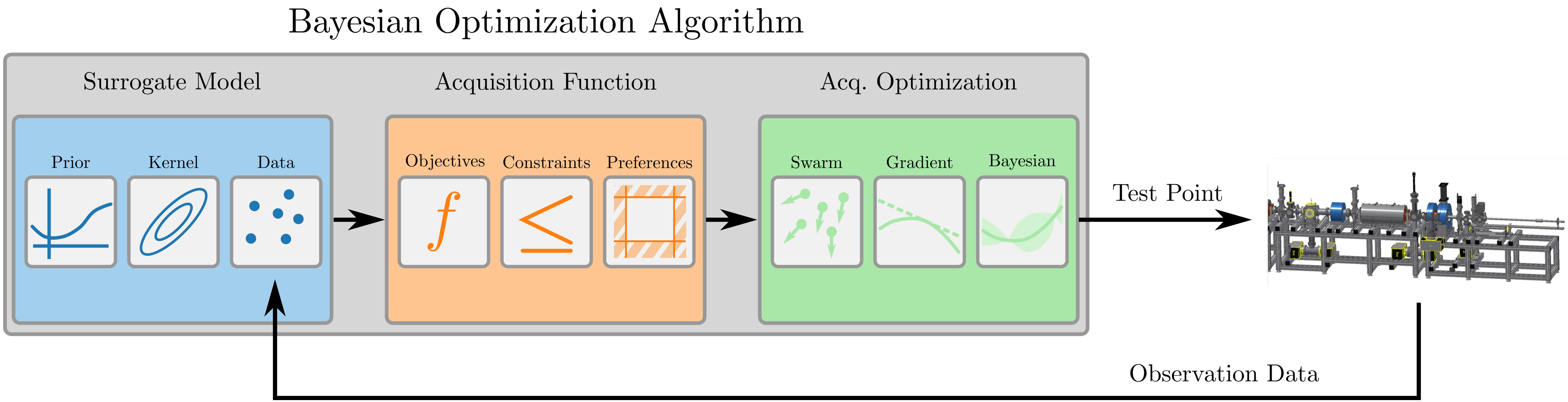
Improve predictive accuracy
- multi-fidelity simulation results
- neural network engine
- manifold GP

Expand capabilities
- Include time dependent drift and noise

Future work

Improve optimization speed
- seeded swarm optimization
- hierarchical Bayesian optimization

Conclusion



Experimental demonstration coming soon!