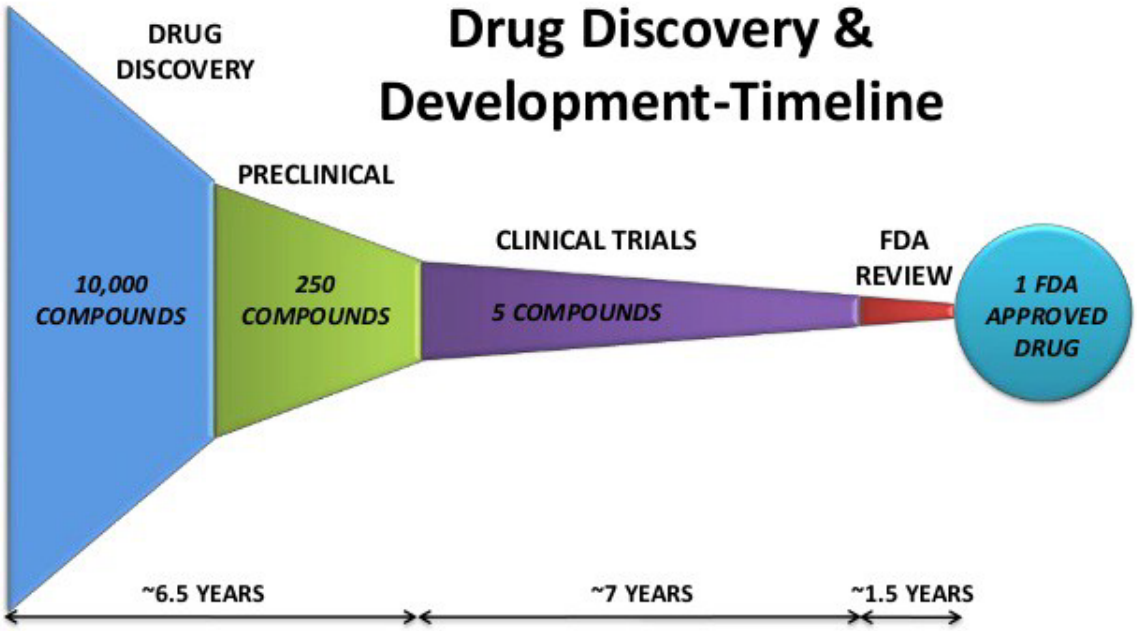
Multi-Objective Bayesian Optimization

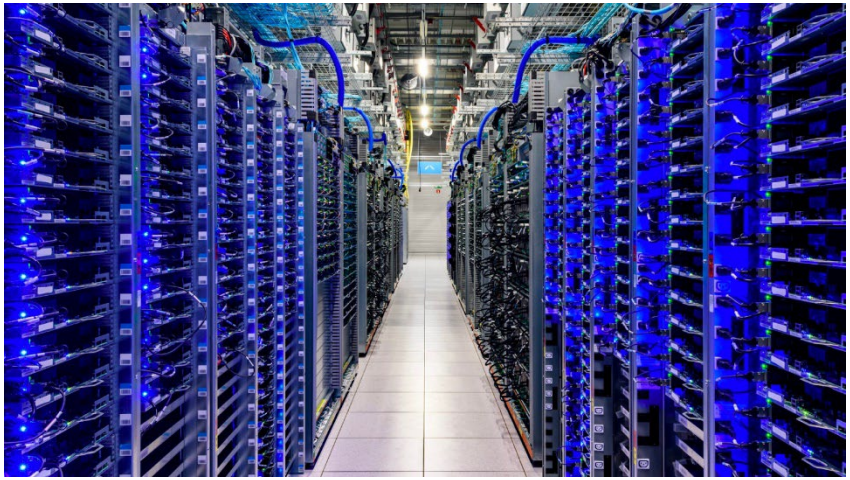
Application #1: Drug/Vaccine Design



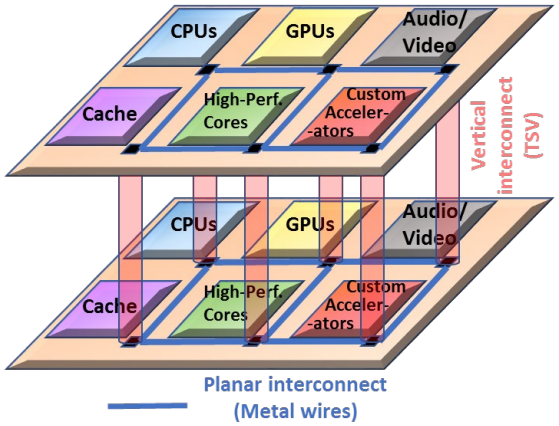
Credit: MIMA healthcare

- Accelerate the discovery of promising designs

Application #2: Hardware Design for Datacenters



High-performance and Energy-efficient manycore chips



America's Data Centers Are Wasting Huge Amounts of Energy

By 2020, data centers are projected to consume roughly 140 billion kilowatt-hours annually, costing American businesses $13 billion annually in electricity bills and emitting nearly 150 million metric tons of carbon pollution

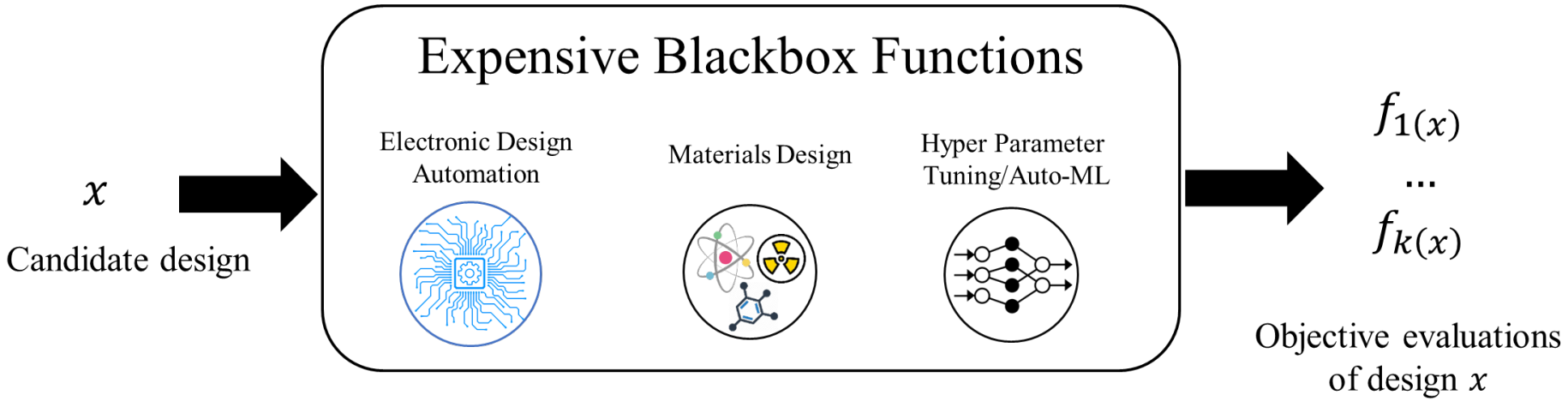
Report from Natural Resources Defense Council:.
https://www.nrdc.org/sites/default/files/data-center-efficiency-assessment-IB.pdf

Performance ↑

Reliability ↑

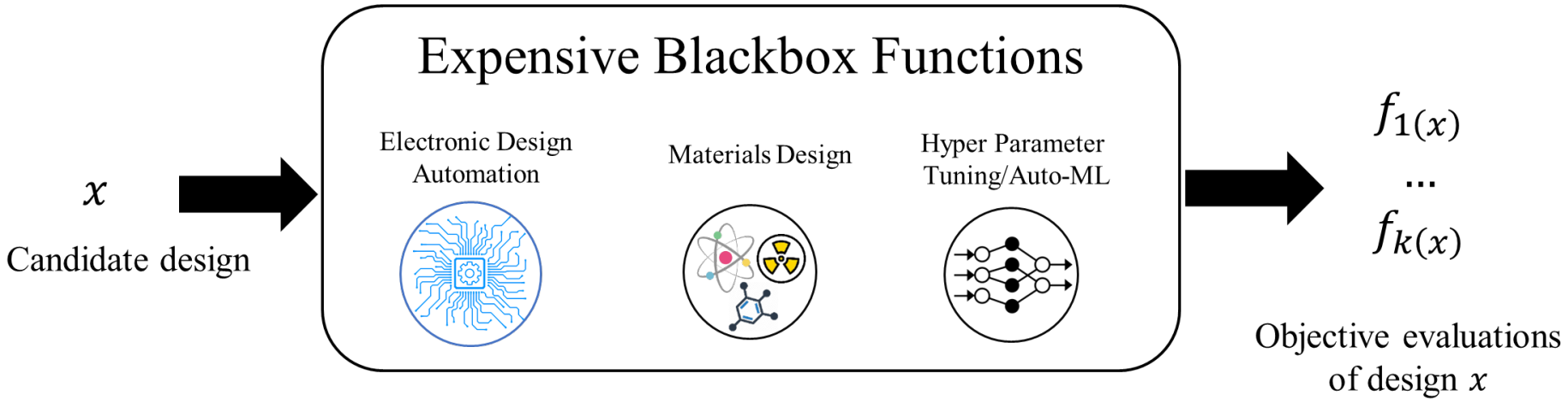
Power ↓

Multi-Objective Optimization: The Problem



- Goal: Find designs with optimal trade-offs by minimizing the total resource cost of experiments

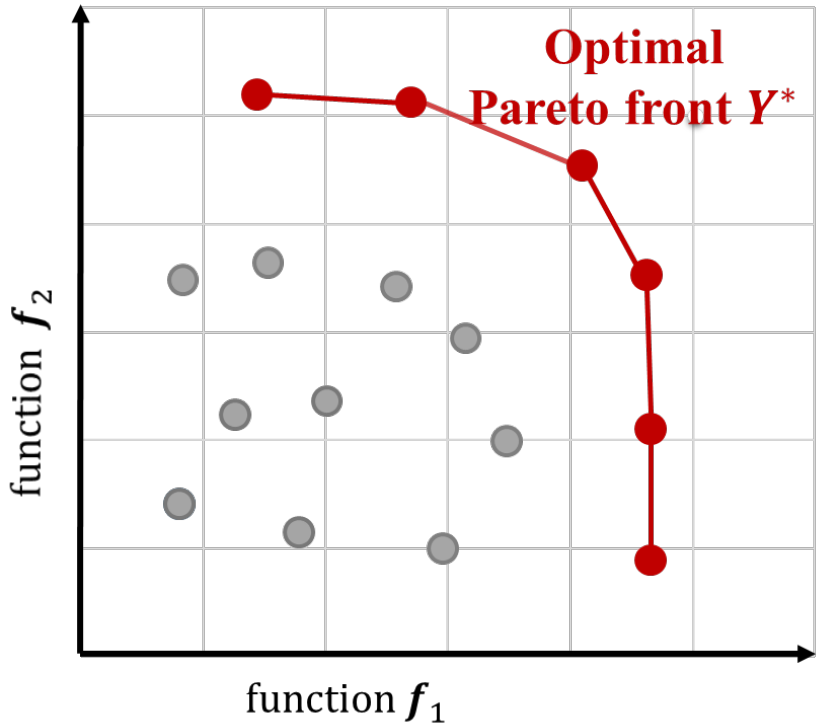
Multi-Objective Optimization: Key Challenge



- Optimize multiple conflicting objective functions

Multi-Objective Optimization: The Solution

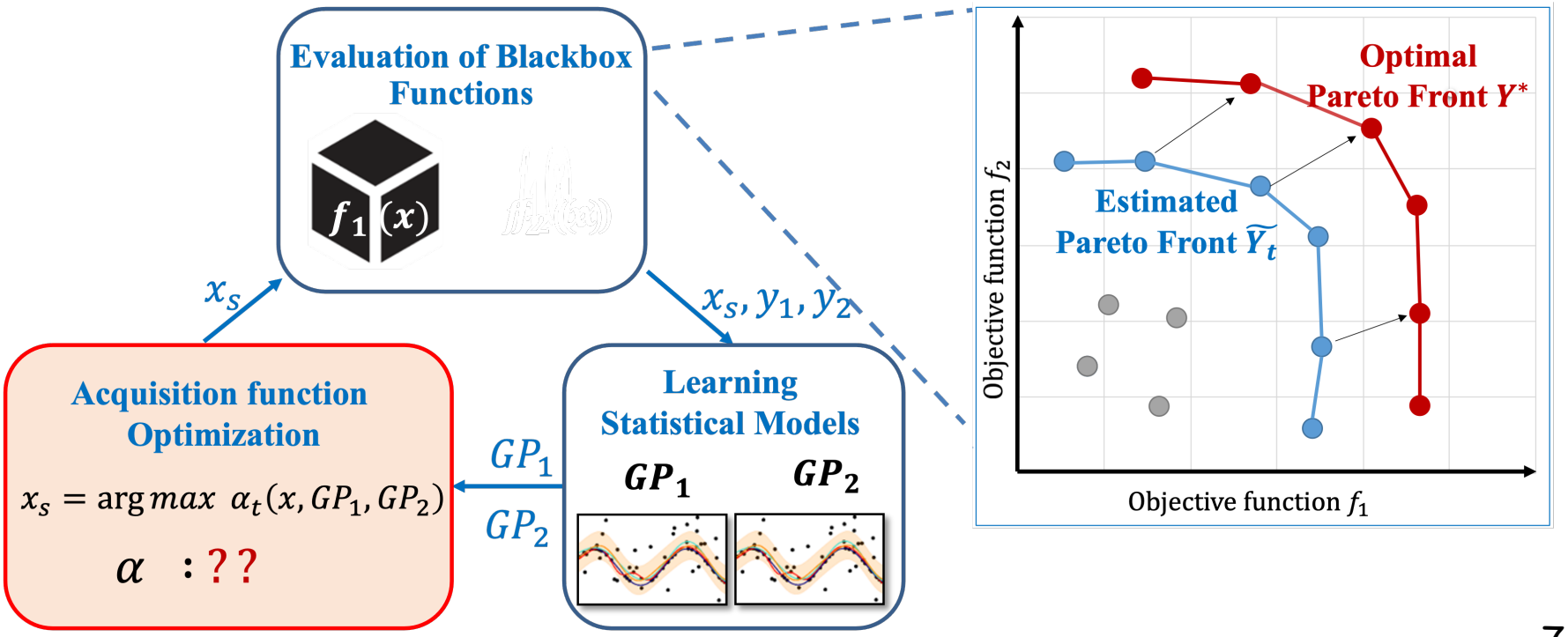
- Set of input designs with optimal trade-offs called the optimal Pareto set χ^*
- Corresponding set of function values called optimal pareto front Pareto front Y^*



- Pareto hypervolume measures the quality of a Pareto front

Single => Multi-Objective BO

- Challenge #1: Statistical modeling
  - Typically, one GP model for each objective function (tractability)
- Challenge #2: Acquisition function design
  - Capture the trade-off between multiple objectives



Multi-Objective BO: Summary of Approaches

- Reduction to single-objective via scalarization
  - ParEGO [Knowles et al., 2006] and MOBO-RS [Paria et al., 2019]
- Hypervolume improvement
  - EHI [Emmerich et al., 2008] , SUR [Picheny et al., 2015] , SMSego [Ponweiser et al., 2008] , qEHVI [Daulton et al., 2020], DGEMO [Lukovic et al. 2020]
- Wrapper methods via single-objective acquisition functions
  - USeMO [Belakaria et al., 2020]
- Information-theoretic methods
  - ϵ-PAL [Zuluaga et al., 2013] , PESMO [Hernandez-Lobato et al., 2016] , MESMO [Belakaria et al., 2019]

Multi-Objective BO: Summary of Approaches

- Reduction to single-objective via scalarization
  - ParEGO [Knowles et al., 2006] and MOBO-RS [Paria et al., 2019]
- Hypervolume improvement
  - EHI [Emmerich et al., 2008] , SUR [Picheny et al., 2015] , SMSego [Ponweiser et al., 2008] , qEHVI [Daulton et al., 2020], DGEMO [Lukovic et al. 2020]
- Wrapper methods via single-objective acquisition functions
  - USeMO [Belakaria et al., 2020]
- Information-theoretic methods
  - ϵ-PAL [Zuluaga et al., 2013] , PESMO [Hernandez-Lobato et al., 2016] , MESMO [Belakaria et al., 2019]

Reduction via Random Scalarization

- Reduce the problem to single objective optimization

- ParEGO [Knowles et al., 2006]
  - BO over scalarized objective function using EI

$$f(x) = \sum_{i=1}^{k} \lambda_i . f_i(x)$$

  - Scalar weights are sampled from a uniform distribution

- MOBO-RS [Paria et al., 2019]
  - Optimize scalarized objective function over a set of scalar weight-vectors using a prior specified by the user

Reduction via Random Scalarization

- ParEGO [Knowles et al., 2006]
  - BO over scalarized objective function using EI

$$f(x) = \sum_{i=1}^{k} \lambda_i . f_i(x)$$

  - Scalar weights are sampled from a uniform distribution
- MOBO-RS [Paria et al., 2019]
  - Optimize scalarized objective function over a set of scalar weight-vectors using a prior specified by the user

Hard to define the scalars or specify priors over scalars, which can lead to sub-optimal results

Multi-Objective BO: Summary of Approaches

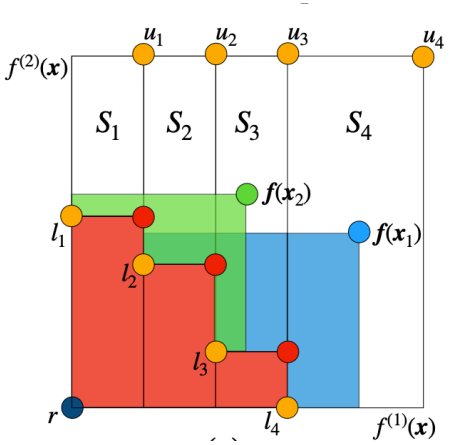
- Reduction to single-objective via scalarization
  - ParEGO [Knowles et al., 2006] and MOBO-RS [Paria et al., 2019]
- Hypervolume improvement
  - EHI [Emmerich et al., 2008] , SUR [Picheny et al., 2015] , SMSego [Ponweiser et al., 2008] , qEHVI [Daulton et al., 2020], DGEMO [Lukovic et al. 2020]
- Wrapper methods via single-objective acquisition functions
  - USeMO [Belakaria et al., 2020]
- Information-theoretic methods
  - ϵ-PAL [Zuluaga et al., 2013] , PESMO [Hernandez-Lobato et al., 2016] , MESMO [Belakaria et al., 2019]

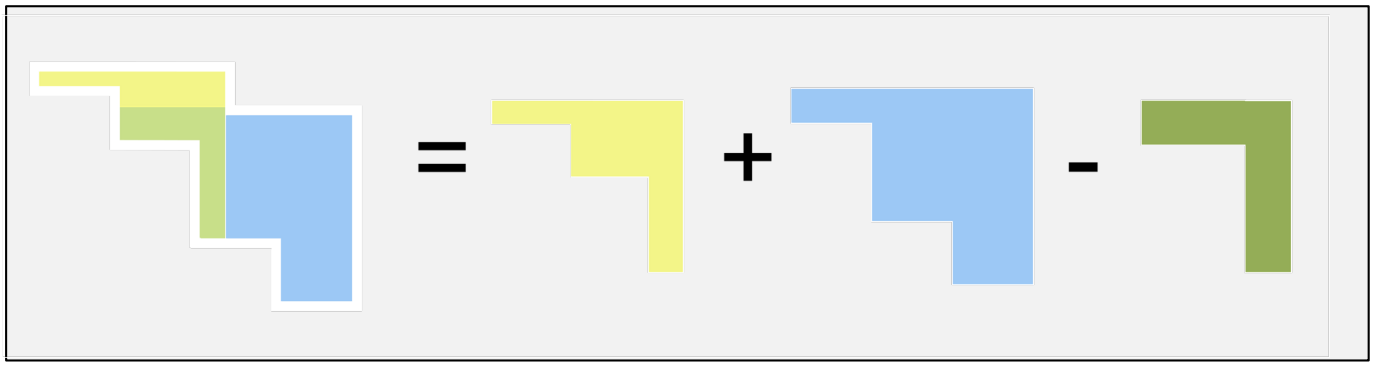
Hypervolume Improvement Approaches

- EHI: Expected improvement in PHV [Emmerich et al., 2008]
- SUR: Probability of improvement in PHV [Picheny et al., 2015]
- SMSego [Ponweiser et al., 2008]
  - Improves the scalability of PHV computation by automatically reducing the search space
- qEHVI [Daulton et al., 2020]
  - Differentiable hypervolume improvement

qEHVI Algorithm [Daulton et al., 2020]

- Parallel EHVI via the Inclusion-Exclusion Principle





- Practical since q is usually small
- The computation of all intersections be parallelized
- The formulation simplifies computation of overlapping hypervolumes

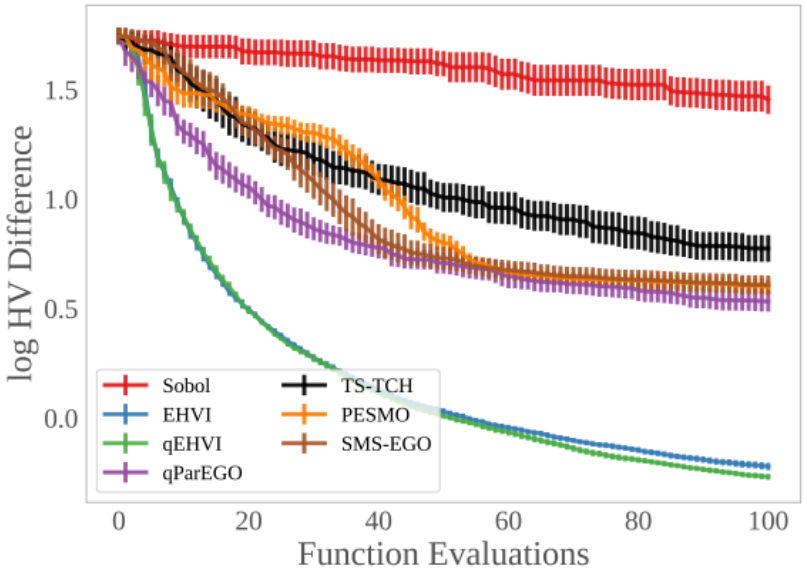
qEHVI Algorithm [Daulton et al., 2020]

- Differentiable Hypervolume Improvement
  - Sample path gradients via the reparameterization trick
  - Unbiased gradient estimator

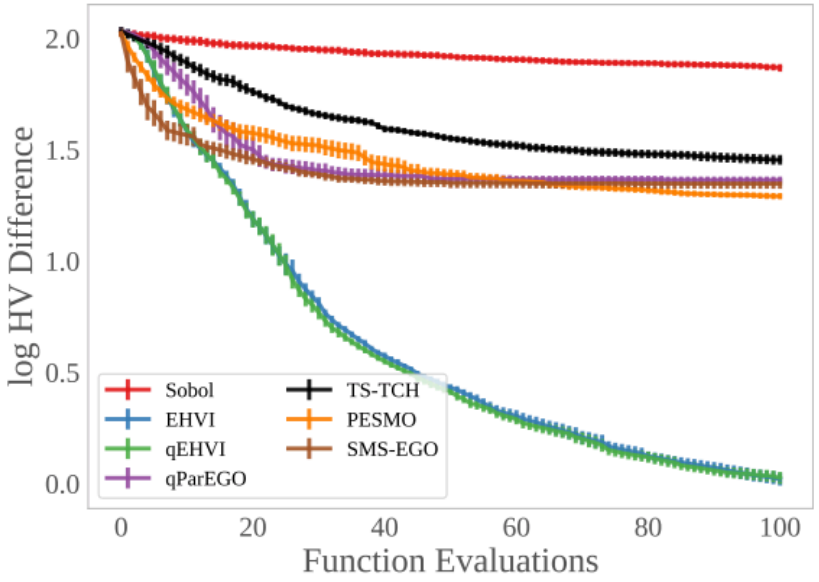
$$\mathbb{E}[\nabla_{\boldsymbol{x}} \hat{\alpha}_{qEHVI}(\boldsymbol{x})] = \nabla_{\boldsymbol{x}} \alpha_{qEHVI}(\boldsymbol{x})$$

qEHVI Algorithm [Daulton et al., 2020]

Vehicle Crash Safety



Branin-Currin



Hypervolume Improvement Approaches

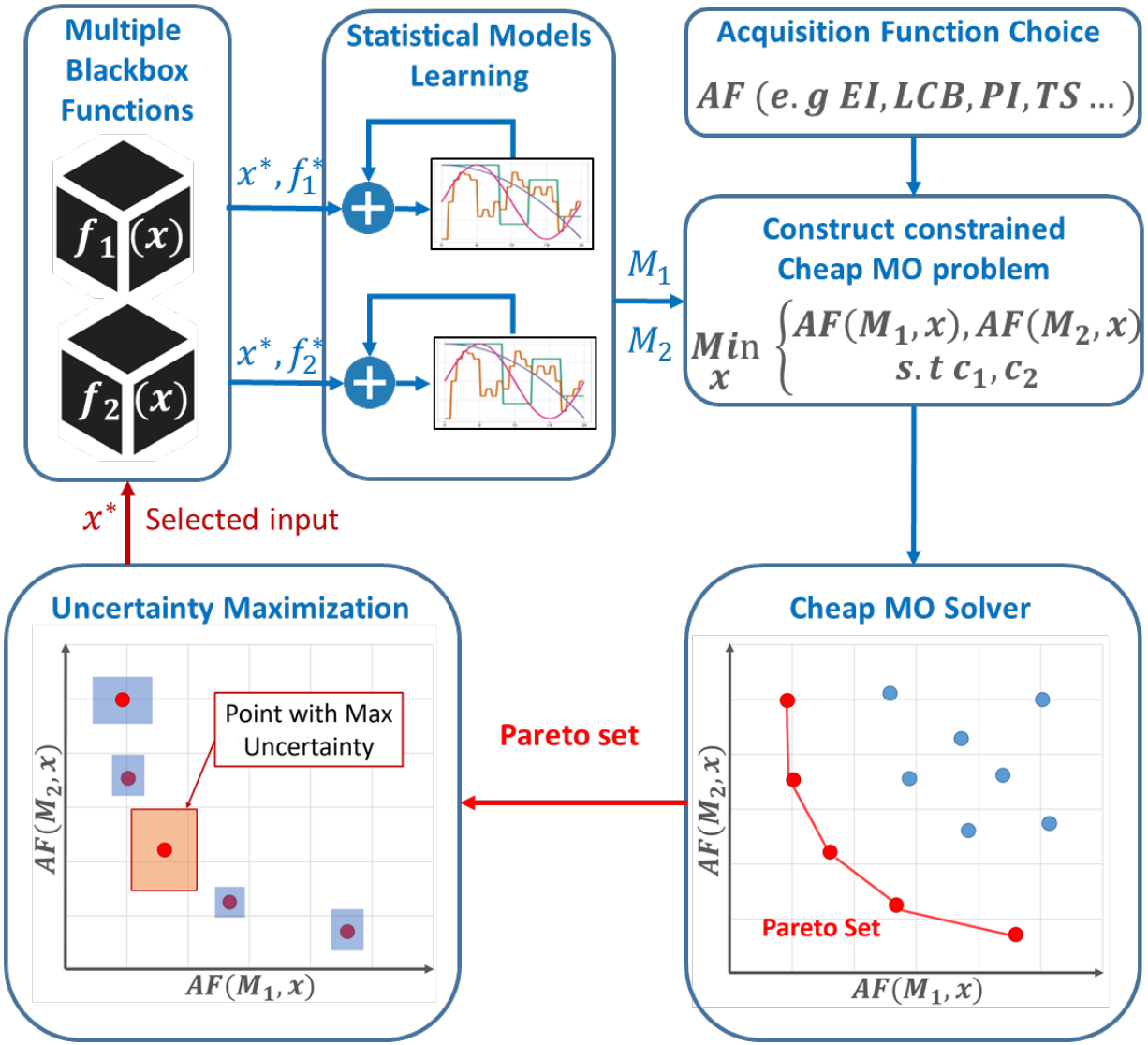
- EHI: Expected improvement in PHV [Emmerich et al., 2008]
- SUR: Probability of improvement in PHV [Picheny et al., 2015]
- SMSego [Ponweiser et al., 2008]
  - Improves the scalability of PHV computation by automatically reducing the search space
- qEHVI [Daulton et al., 2020]
  - Differentiable hypervo

Can potentially lead to more exploitation behavior resulting in sub-optimal solutions

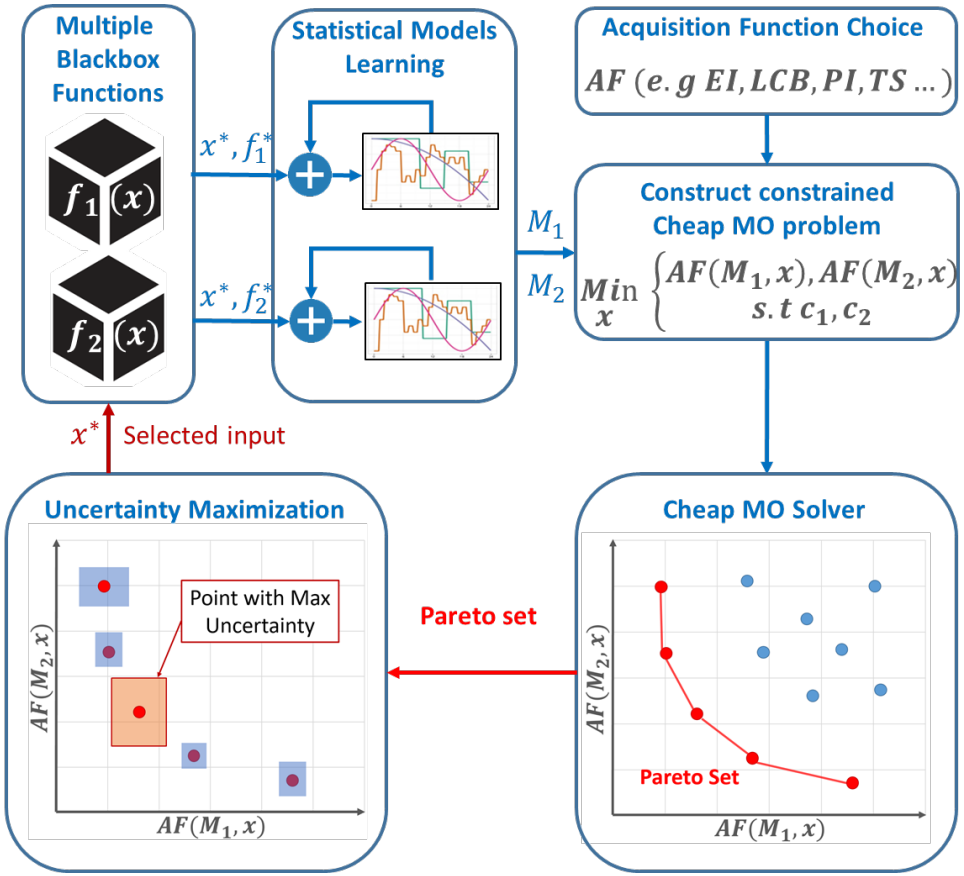
Multi-Objective BO: Summary of Approaches

- Reduction to single-objective via scalarization
  - ParEGO [Knowles et al., 2006] and MOBO-RS [Paria et al., 2019]
- Hypervolume improvement
  - EHI [Emmerich et al., 2008] , SUR [Picheny et al., 2015] , SMSego [Ponweiser et al., 2008] , qEHVI [Daulton et al., 2020]
- Wrapper methods via single-objective acquisition functions
  - USeMO [Belakaria et al., 2020]
- Information-theoretic methods
  - ϵ-PAL [Zuluaga et al., 2013] , PESMO [Hernandez-Lobato et al., 2016] , MESMO [Belakaria et al., 2019]

USeMO Framework [Belakaria et al., 2020]

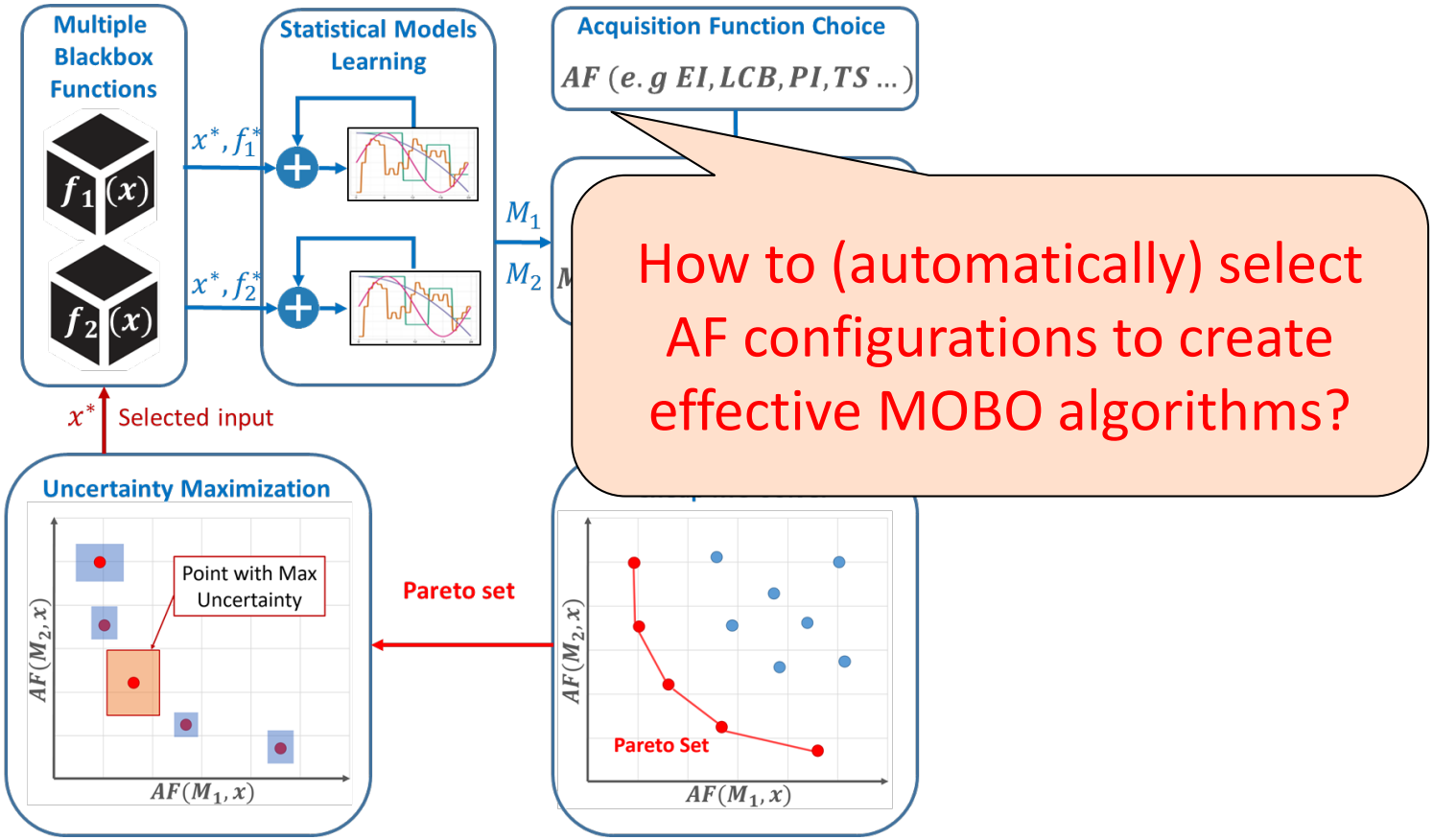


USeMO Framework [Belakaria et al., 2020]



- Allows us to leverage acquisition functions from single-objective BO to solve multi-objective BO problems

USeMO Framework [Belakaria et al., 2020]



- Allows us to leverage acquisition functions from single-objective BO to solve multi-objective BO problems

Multi-Objective BO: Summary of Approaches

- Reduction to single-objective via scalarization
  - ParEGO [Knowles et al., 2006] and MOBO-RS [Paria et al., 2019]
- Hypervolume improvement
  - EHI [Emmerich et al., 2008] , SUR [Picheny et al., 2015] , SMSego [Ponweiser et al., 2008] , qEHVI [Daulton et al., 2020]
- Wrapper methods via single-objective acquisition functions
  - USeMO [Belakaria et al., 2020]
- Information-theoretic methods
  - ϵ-PAL [Zuluaga et al., 2013] , PESMO [Hernandez-Lobato et al., 2016] , MESMO [Belakaria et al., 2019]

ϵ-PAL Algorithm [Zuluaga et al., 2013]

- Classifies candidate inputs into three categories using the learned GP models
  - Pareto-optimal
  - Not Pareto-optimal
  - Uncertain

- In each iteration, selects the candidate input for evaluation to minimize the size of uncertain set

- Accuracy of pruning depends critically on ϵ value

ϵ-PAL Algorithm [Zuluaga et al., 2013]

- Classifies candidate inputs into three categories using the learned GP models
  - Pareto-optimal
  - Not Pareto-optimal
  - Uncertain

Limited applicability as it works only for discrete set of candidate inputs

- In each iteration, selects the candidate input for evaluation to minimize the size of uncertain set

- Accuracy of pruning depends critically on ϵ value

PESMO Algorithm [Hernandez-Lobato et al., 2016]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto set χ^*

- Reminder: Set of input designs with optimal trade-offs is called the optimal Pareto set χ^*

PESMO Algorithm [Hernandez-Lobato et al., 2016]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto set χ^*

$$\begin{aligned}\alpha(x) &= I(\{x, y\}, \chi^* | D) \\ &= H(\chi^* | D) - \mathbb{E}_y[H(\chi^* | D \cup \{x, y\})] \\ &= H(y | D, x) - \mathbb{E}_{\chi^*}[H(y | D, x, \chi^*)]\end{aligned}$$

PESMO Algorithm [Hernandez-Lobato et al., 2016]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto set χ^*

$$\alpha(x) = I(\{x, y\}, \chi^* | D)$$
$$= H(\chi^* | D) - \mathbb{E}_y[H(\chi^* | D \cup \{x, y\})]$$
$$= H(y | D, x) - \mathbb{E}_{\chi^*}[H(y | D, x, \chi^*)]$$

Equivalent to expected reduction in entropy over the pareto set χ^*

PESMO Algorithm [Hernandez-Lobato et al., 2016]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto set χ^*

$$\alpha(x) = I(\{x, y\}, \chi^* | D)$$
$$= H(\chi^* | D) - \mathbb{E}_y[H(\chi^* | D \cup \{x, y\})]$$
$$= H(y | D, x) - \mathbb{E}_{\chi^*}[H(y | D, x, \chi^*)]$$

Due to symmetric property of information gain

PESMO Algorithm [Hernandez-Lobato et al., 2016]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto set χ^*

$$\alpha(x) = I(\{x, y\}, \chi^* | D)$$
$$= H(\chi^* | D) - \mathbb{E}_y[H(\chi^* | D \cup \{x, y\})]$$
$$= H(y | D, x) - \mathbb{E}_{\chi^*}[H(y | D, x, \chi^*)]$$

Entropy of factorizable Gaussian distribution

PESMO Algorithm [Hernandez-Lobato et al., 2016]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto set χ^*

input dimension d

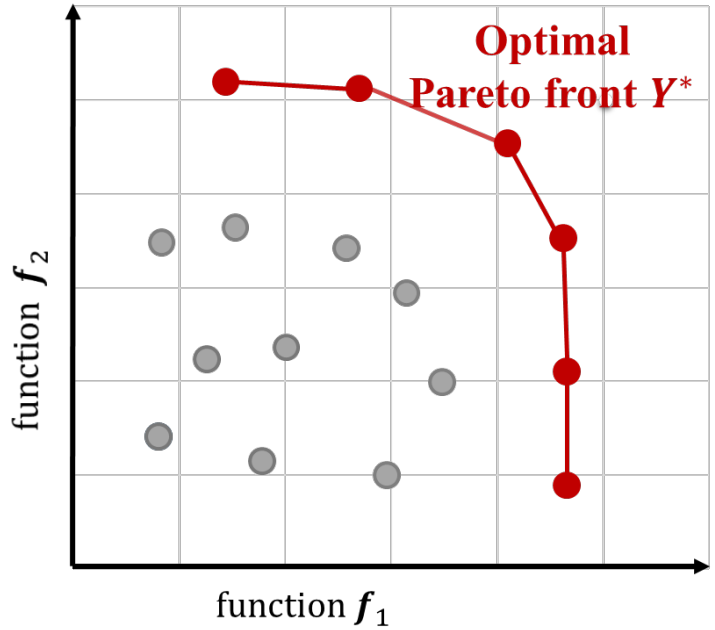
$$\alpha(x) = I(\{x, y\}, \chi^* | D)$$
$$= H(\chi^* | D) - \mathbb{E}_y[H(\chi^* | D \cup \{x, y\})]$$
$$= H(y | D, x) - \mathbb{E}_{\chi^*}[H(y | D, x, \chi^*)]$$

Requires computationally expensive approximation using expectation propagation

MESMO Algorithm [Belakaria et al., 2019]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto front Y^*

- Reminder: Set of function values corresponding to the optimal Pareto set χ^* is called the optimal Pareto front Y^*



MESMO Algorithm [Belakaria et al., 2019]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto front Y^*

$$\begin{aligned}\alpha(x) &= I(\{x, y\}, \boxed{Y^*} | D) \\ &= H(Y^* | D) - \mathbb{E}_y[H(Y^* | D \cup \{x, y\})] \\ &= H(y | D, x) - \mathbb{E}_{Y^*}[H(y | D, x, Y^*)]\end{aligned}$$

MESMO Algorithm [Belakaria et al., 2019]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto front Y^*

$$\alpha(x) = I(\{x, y\}, Y^* | D)$$
$$= H(Y^* | D) - \mathbb{E}_y[H(Y^* | D \cup \{x, y\})]$$
$$= H(y | D, x) - \mathbb{E}_{Y^*}[H(y | D, x, Y^*)]$$

Equivalent to expected reduction in entropy over the pareto front Y^*

MESMO Algorithm [Belakaria et al., 2019]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto front Y^*

$$\alpha(x) = I(\{x, y\}, Y^* | D)$$
$$= H(Y^* | D) - \mathbb{E}_y[H(Y^* | D \cup \{x, y\})]$$
$$= H(y | D, x) - \mathbb{E}_{Y^*}[H(y | D, x, Y^*)]$$

Due to symmetric property of information gain

MESMO Algorithm [Belakaria et al., 2019]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto front Y^*

$$\alpha(x) = I(\{x, y\}, \boxed{Y^*} | D)$$
$$= H(Y^*|D) - \mathbb{E}_y[H(Y^* | D \cup \{x, y\})]$$
$$= H(y|D, x) - \mathbb{E}_{Y^*}[H(y | D, x, Y^*)]$$

Entropy of factorizable Gaussian distribution

MESMO Algorithm [Belakaria et al., 2019]

- Key Idea: select the input that maximizes the information gain about the optimal Pareto front Y^*

Output dimension $k \ll d$

$$\alpha(x) = I(\{x, y\}, \boxed{Y^*} | D)$$
$$= H(Y^* | D) - \mathbb{E}_y[H(Y^* | D \cup \{x, y\})]$$
$$= H(y | D, x) - \boxed{\mathbb{E}_{Y^*}[H(y | D, x, Y^*)]}$$

Closed form using properties of entropy and truncated Gaussian distribution

MESMO Algorithm [Belakaria et al., 2019]

$$\alpha(x) = H(y|D,x) - \mathbb{E}_{Y^*}[H(y|D,x,Y^*)]$$

- The first term is the entropy of a factorizable k-dimensional Gaussian distribution $P(y|D,x)$

$$H(y|D,x) = \frac{K(1+\ln(2\pi))}{2} + \sum_{j=1}^{k} \ln(\sigma_j(x))$$

MESMO Algorithm [Belakaria et al., 2019]

$$\alpha(x) = H(y|D,x) - \mathbb{E}_{Y^*}[H(y\,|D,x,Y^*)]$$

- We can approximately compute the second term via Monte-Carlo sampling (S is the number of samples)

$$\mathbb{E}_{Y^*}[H(y\,|D,x,Y^*)] \approx \frac{1}{S}\sum_{s=1}^{S} H(y\,|D,x,Y_s^*)$$

MESMO Algorithm [Belakaria et al., 2019]

- Approximate computation via Monte-Carlo sampling

$$\mathbb{E}_{Y^*}[H(y \mid D, x, Y^*)] \approx \frac{1}{S} \sum_{s=1}^{S} H(y \mid D, x, Y_s^*)$$

- **Two key steps**
  - How to compute Pareto front samples Y_s^* ?
  - How to compute the entropy with respect to a given Pareto front sample Y_s^*?

MESMO Algorithm [Belakaria et al., 2019]

- Approximate computation via Monte-Carlo sampling

$$\mathbb{E}_{Y^*}[H(y \mid D, x, Y^*)] \approx \frac{1}{S} \sum_{s=1}^{S} H(y \mid D, x, Y_s^*)$$

- How to compute Pareto front samples Y_s^* ?
  - Sample functions from posterior GPs via random Fourier features
  - Solve a cheap MO problem over the sampled functions $\tilde{f}_1 \ldots \tilde{f}_k$ to compute sample Pareto front

MESMO Algorithm [Belakaria et al., 2019]

- How to compute the entropy with respect to a given Pareto front sample Y_s^*?

$$Y_s^* = \{\boldsymbol{v^1}, \dots, \boldsymbol{v^l}\} \; with \; \boldsymbol{v^i} = \{v_1^i, \dots, v_K^i\},$$
$$y_j \leq y_{j_s}^* = \max\{v_1^1, \dots, v_j^l\} \; \forall j \in \{1, \dots, K\}$$

  - Decompose the entropy of a set of independent variables into a sum of entropies of individual variables

  - Model each component y_j as a truncated Gaussian distribution

MESMO Algorithm [Belakaria et al., 2019]

- How to compute the entropy with respect to a given Pareto front sample Y_s^*?

$$Y_s^* = \{\boldsymbol{v^1}, \dots, \boldsymbol{v^l}\} \; with \; \boldsymbol{v^i} = \{v_1^i, \dots, v_K^i\},$$
$$y_j \leq y_{j_s}^* = \max\{v_1^1, \dots, v_j^l\} \; \forall j \in \{1, \dots, K\}$$

$$H(y \,|D, x, Y_s^*) \approx \sum_{j=1}^{K} H\left(y_j |\, D, x, y_{j_s}^*\right)$$

MESMO Algorithm [Belakaria et al., 2019]

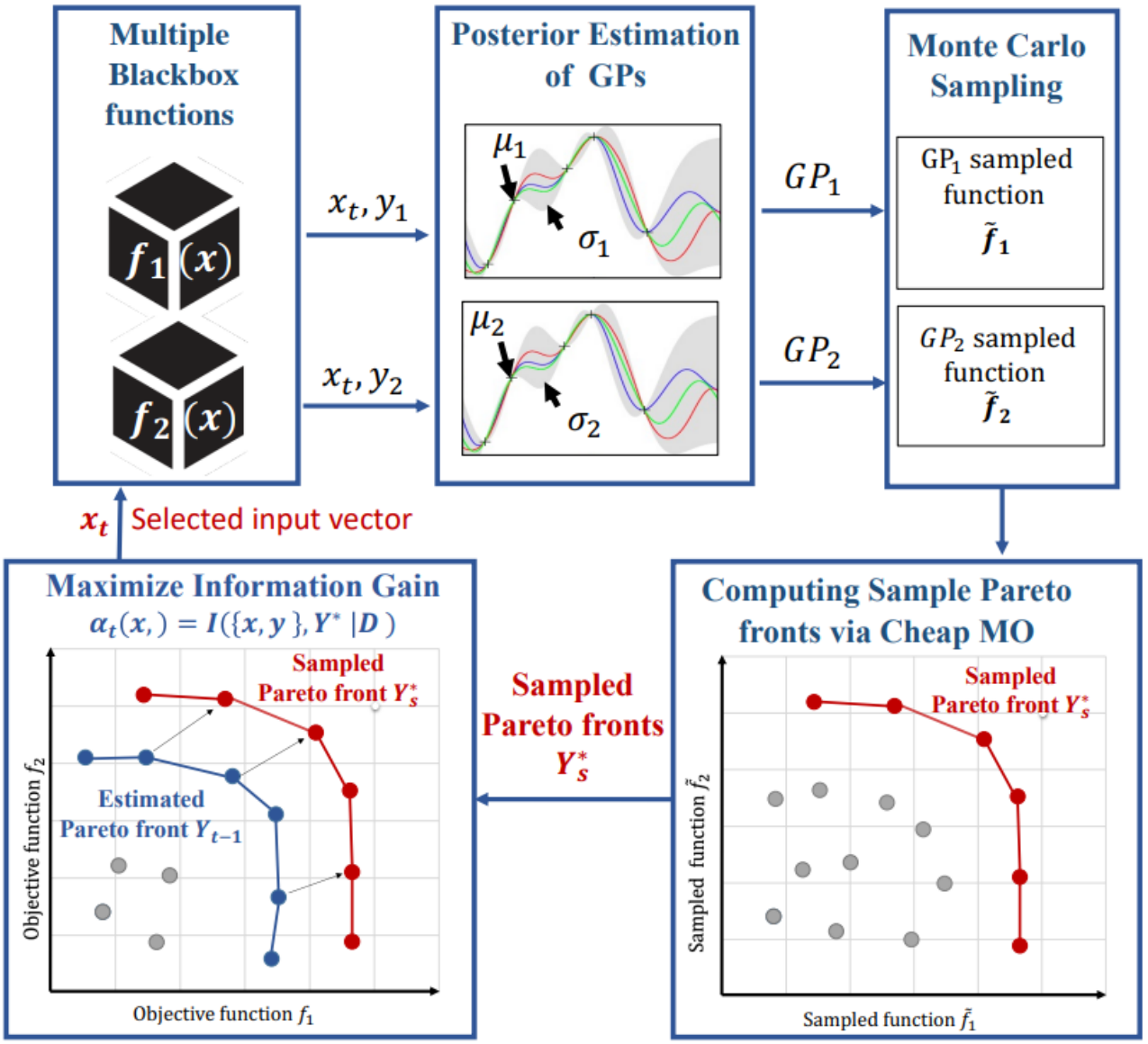
- Final acquisition function

$$\alpha(x) \approx \frac{1}{S} \sum_{s=1}^{S} \sum_{j=1}^{K} \left[\frac{\gamma_s^j(x)\phi\left(\gamma_s^j(x)\right)}{2\Phi\left(\gamma_s^j(x)\right)} - \ln\Phi\left(\gamma_s^j(x)\right) \right]$$

Closed form

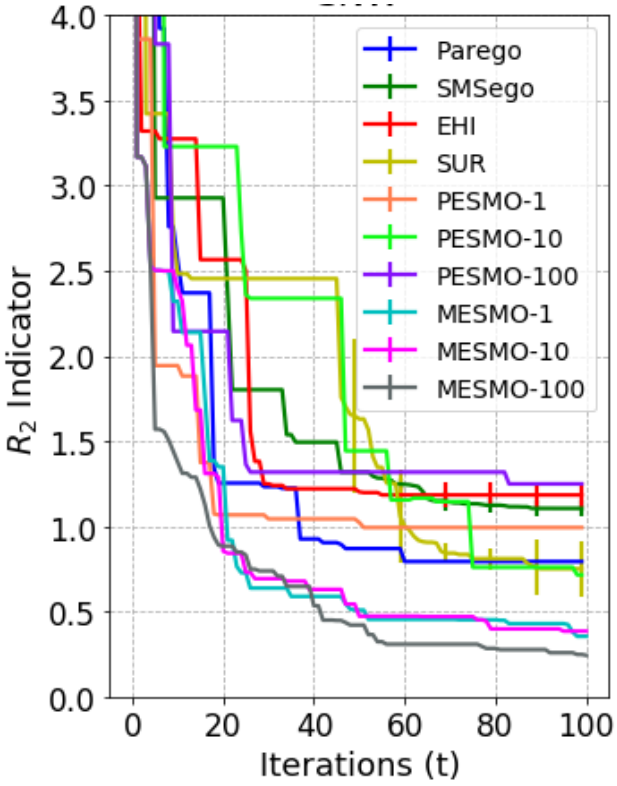
where $\gamma_s^j(x) = \frac{y_{js}^* - \mu_j(x)}{\sigma_j(x)}$, ϕ and Φ are the p.d.f and c.d.f of a standard normal distribution

MESMO Algorithm [Belakaria et al., 2019]

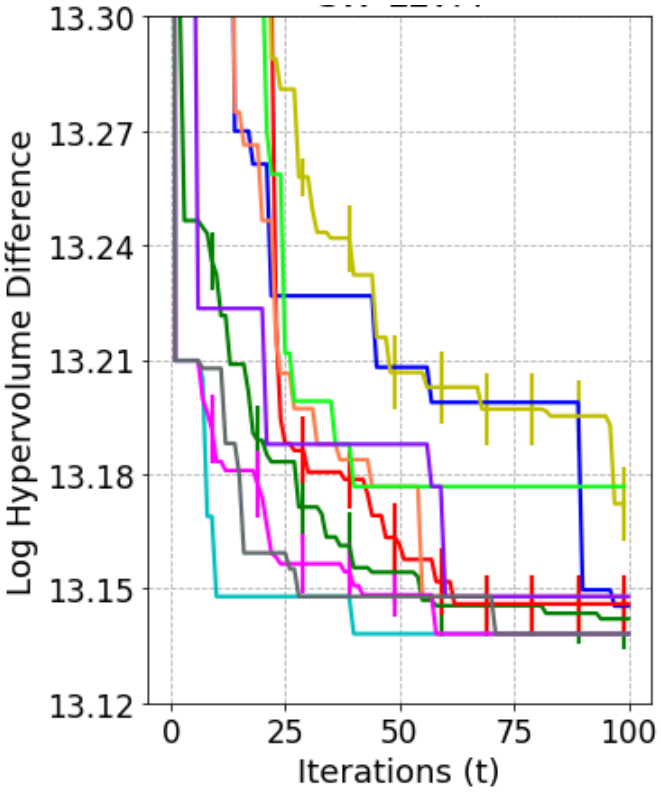


MOBO Experiments and Results #1

Network on Chip Design

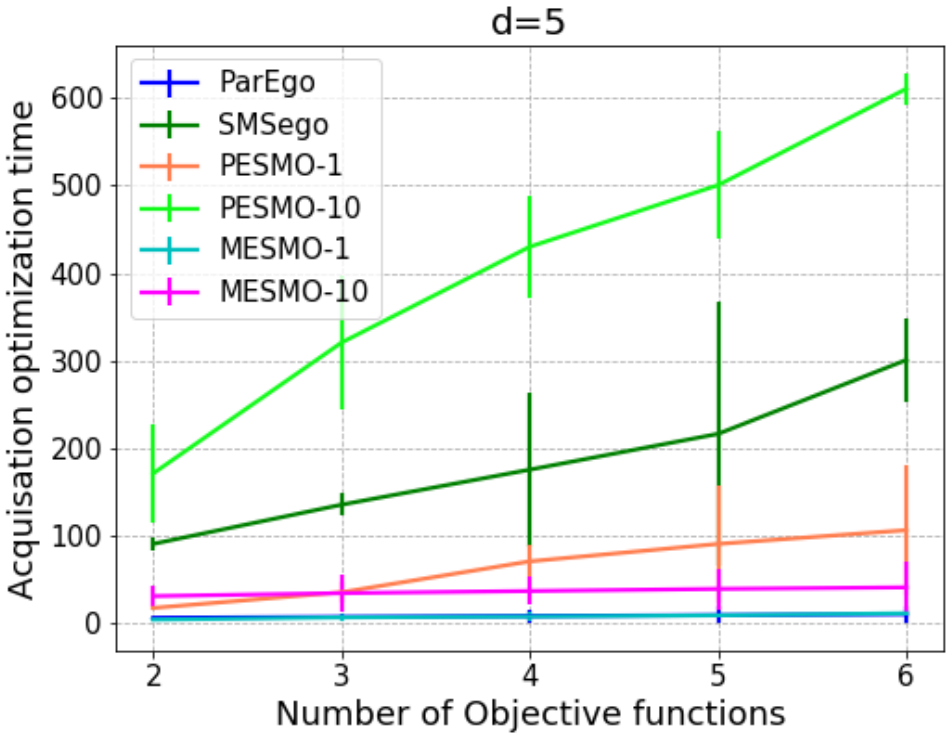


Compiler Settings Optimization



- MESMO is better than PESMO
- MESMO converges faster
- MESMO is robust to the number of samples (even a single sample!)

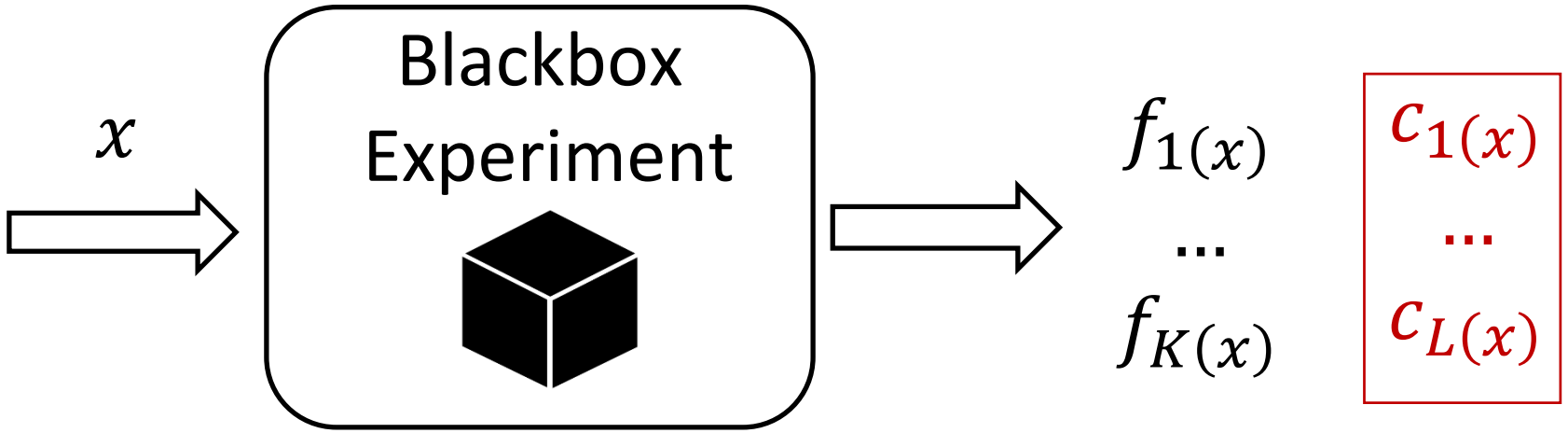
MOBO Experiments and Results #2



- MESMO is highly scalable when compared to PESMO
- MESMO with one sample is comparable to ParEGO
- Time for PESMO and SMSego increases significantly with the number of objectives

Multi-Objective Bayesian Optimization With Black-Box Constraints

MOBO with Black-Box Constraints: The Problem



Objectives and constraints evaluation of design x

- Goal: find the approximate (optimal) constrained Pareto set by minimizing the total resource cost of experiments

MOBO with Black-Box Constraints: The Problem

Amazon Prime Air autonomous unmanned aerial vehicle (UAV)

- Electrified aviation power system design for UAVs [Belakaria et al., 2021]
  - Multiple Objectives: total energy and mass
  - Safety constraints: thresholds for motor temperature and voltage of cells

MESMOC Algorithm [Belakaria et al., 2021]

- Extension of MESMO for constrained setting

$$\alpha(x) \approx \frac{1}{S} \sum_{s=1}^{S} \left[\sum_{j=1}^{K} \frac{\gamma_s^{f_j}(x)\phi\left(\gamma_s^{f_j}(x)\right)}{2\Phi\left(\gamma_s^{f_j}(x)\right)} - \ln\Phi\left(\gamma_s^{f_j}(x)\right) + \sum_{j=1}^{L} \frac{\gamma_s^{c_j}(x)\phi\left(\gamma_s^{c_j}(x)\right)}{2\Phi\left(\gamma_s^{c_j}(x)\right)} - \ln\Phi\left(\gamma_s^{c_j}(x)\right) \right]$$

Closed form

MESMOC Algorithm [Belakaria et al., 2021]

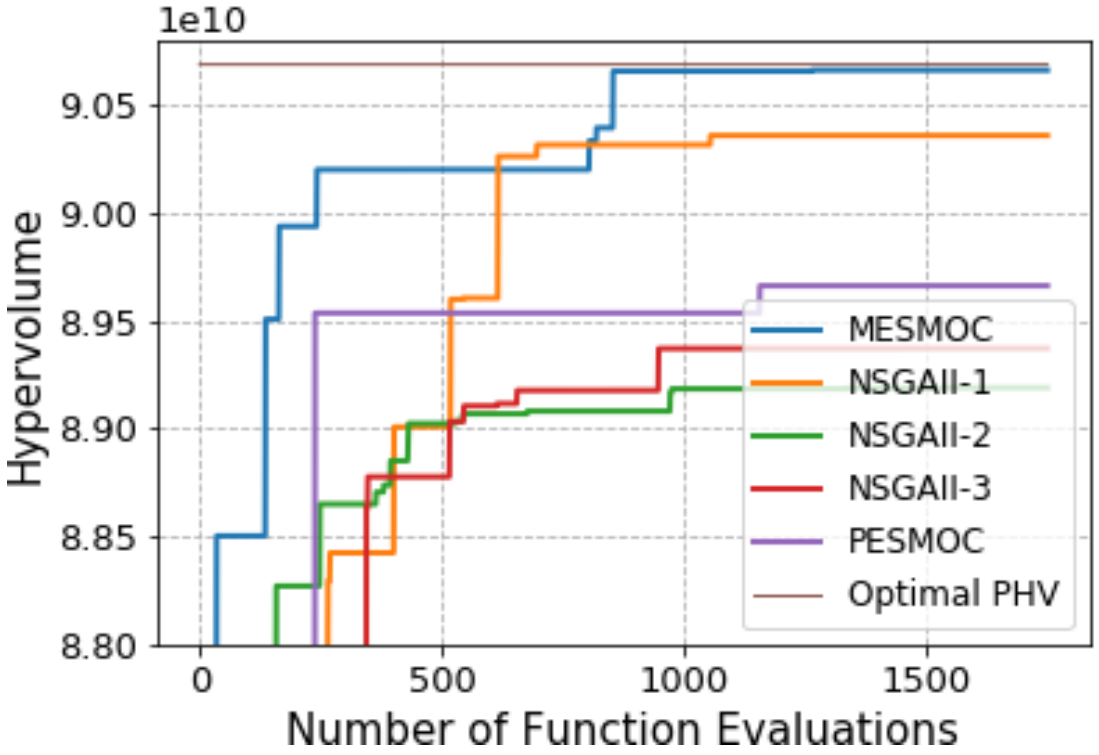
- Solves a cheap MOO over sampled functions ($\widetilde{f_1}, \dots, \widetilde{f_K}$) constrained by sampled constraints ($\widetilde{c_1}, \dots, \widetilde{c_L}$)

$$Y_s^* \leftarrow \arg\max_{x \in \chi} (\widetilde{f_1}, \dots, \widetilde{f_K})$$
$$\text{s.t. } (\widetilde{c_1} \geq 0, \dots, \widetilde{c_L} \geq 0)$$

- Acquisition function optimization constrained by predictive mean of constraints

$$x_t \leftarrow \arg\max_{x \in \chi} \alpha_t$$
$$\text{s.t. } (\mu_{c_1}(x) \geq 0, \dots, \mu_{c_L}(x) \geq 0)$$

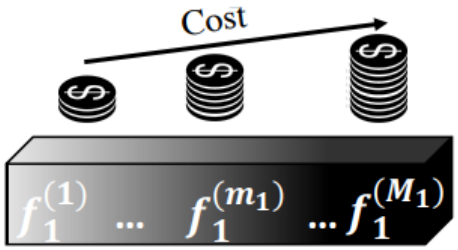
MESMOC Experiments and Results



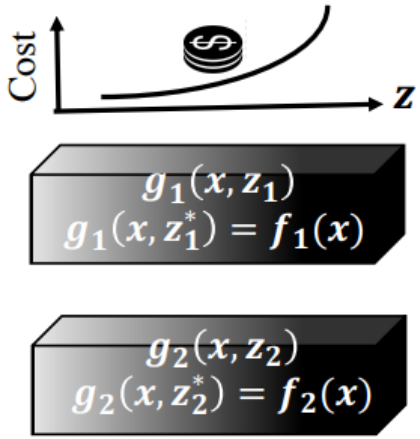
- MESMOC finds near-optimal Pareto front in ~250 evaluations out of ~168,000 designs (<1%)
- 95% of the inputs selected by MESMOC are valid, while the best among baselines was only 39%

Multi-Objective Bayesian Optimization With Multi-Fidelity Function Evaluations

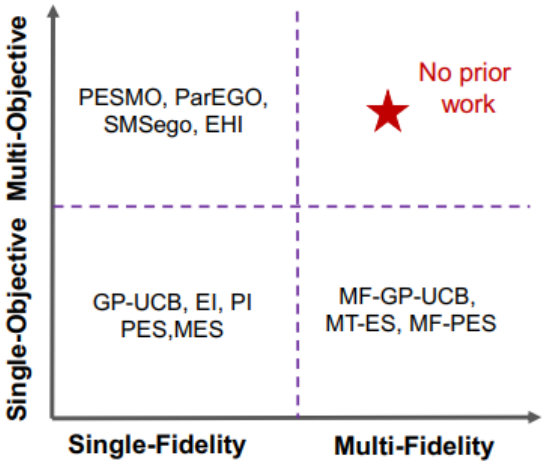
Multi-Fidelity Multi-Objective BO: The Problem



Discrete fidelity



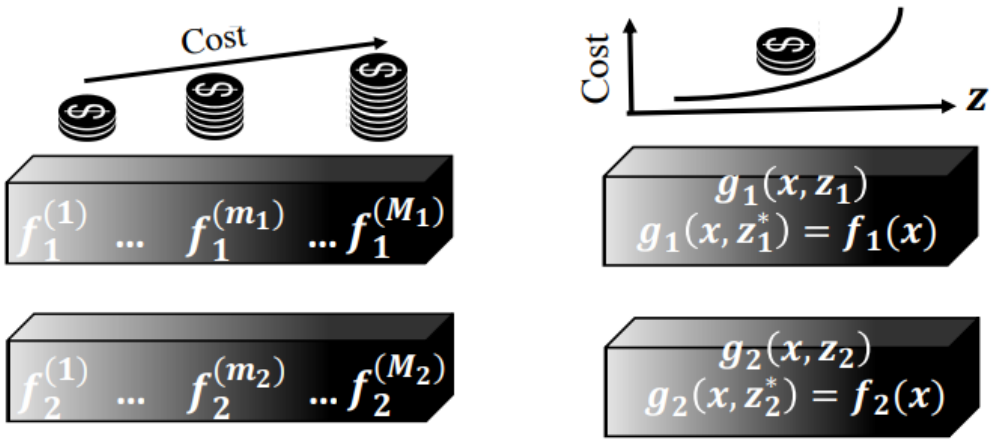
Continuous fidelity

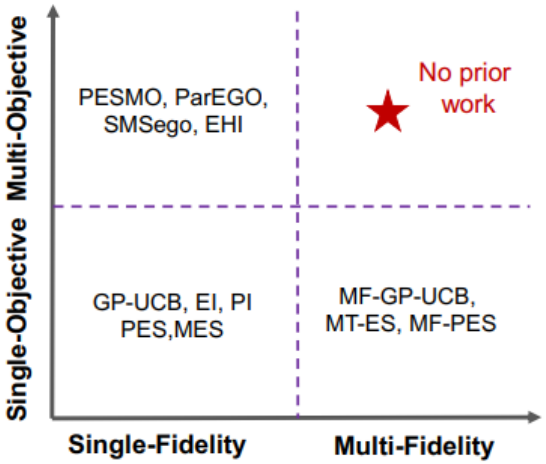


- Continuous-fidelity is the most general case
  - Discrete-fidelity is a special case

- Goal: find the approximate (optimal) Pareto set by minimizing the total resource cost of experiments

Multi-Fidelity Multi-Objective BO: Key Challenges





- How to model functions with multiple fidelities?
- How to join[Already covered] and fidelity-vector pair in each BO iteration?
- How to progressively select higher fidelity experiments?

iMOCA Algorithm [Belakaria et al., 2021]

- Key Idea: Select the input and fidelity-vector that maximizes information gain per unit resource cost about the optimal Pareto front Y^*

$$\alpha(\boldsymbol{x},\boldsymbol{z}) = I(\{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\}, Y^*|D)/C(\boldsymbol{x},\boldsymbol{z})$$
$$= (H(\boldsymbol{y}|D,\boldsymbol{x},\boldsymbol{z}) - \mathbb{E}_{Y^*}[H(\boldsymbol{y}|D,\boldsymbol{x},\boldsymbol{z},Y^*)])/C(\boldsymbol{x},\boldsymbol{z})$$

$$= (\sum_{j=1}^{K} \ln\left(\sqrt{2\pi e}\,\sigma_{g_j}(\boldsymbol{x},z_j)\right)$$
$$-\frac{1}{S}\sum_{s=1}^{S}\sum_{j=1}^{K} H(y_j|D,\boldsymbol{x},z_j,f_s^{j*}))/C(\boldsymbol{x},\boldsymbol{z})$$

where $C(\boldsymbol{x},\boldsymbol{z}) = \sum_{j=1}^{K} \frac{C(\boldsymbol{x},z_j)}{C(\boldsymbol{x},z_j^*)}$ is the normalized cost over different functions

iMOCA Algorithm [Belakaria et al., 2021]

- **Assumption**: Values at lower fidelities are smaller than maximum value of the highest fidelity $y_j \leq f_s^{j*} \; \forall j \in \{1, \dots, K\}$

- Truncated Gaussian approximation (Closed-form)

$$\alpha(\boldsymbol{x}, \boldsymbol{z}) \approx \frac{1}{C(\boldsymbol{x},\boldsymbol{z})S} \sum_{s=1}^{S} \sum_{j=1}^{K} \left[\frac{\gamma_s^{(g_j)} \phi\left(\gamma_s^{(g_j)}\right)}{2\Phi\left(\gamma_s^{(g_j)}\right)} - \ln\Phi\left(\gamma_s^{(g_j)}\right) \right]$$

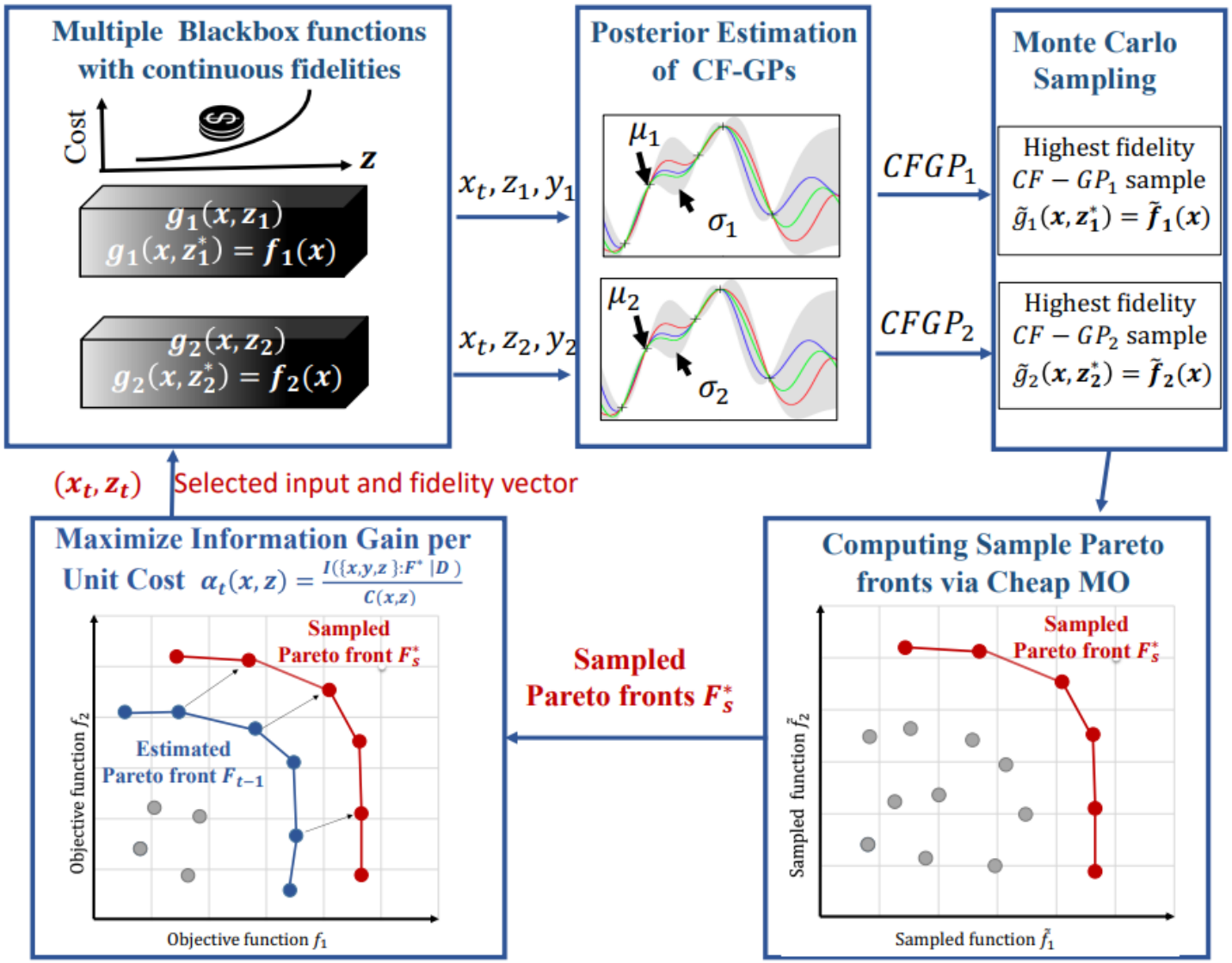
Where $\gamma_s^{(g_j)} = \frac{f_s^{j*} - \mu_{g_j}}{\sigma_{g_j}}$, $\boldsymbol{\phi}$ and $\boldsymbol{\Phi}$ are the p.d.f and c.d.f of a standard normal distribution

iMOCA Algorithm [Belakaria et al., 2021]

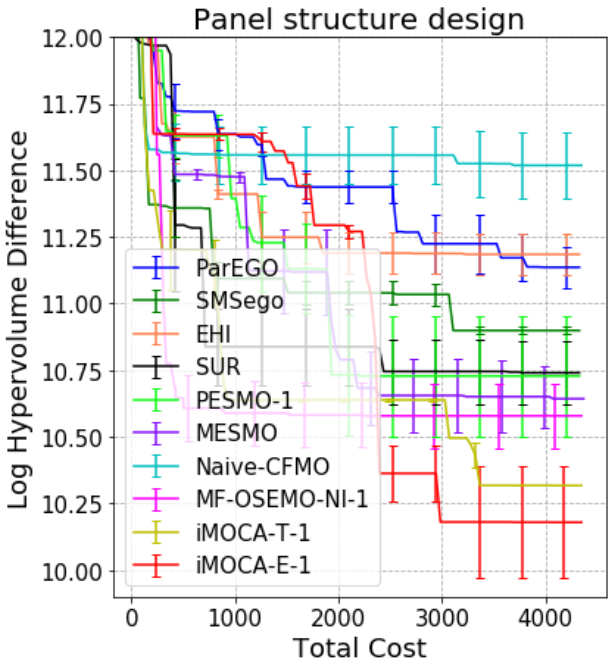
- Challenges of large (potentially infinite) fidelity space
  - Select costly fidelity with less accuracy
  - Tendency to select lower fidelities due to normalization by cost

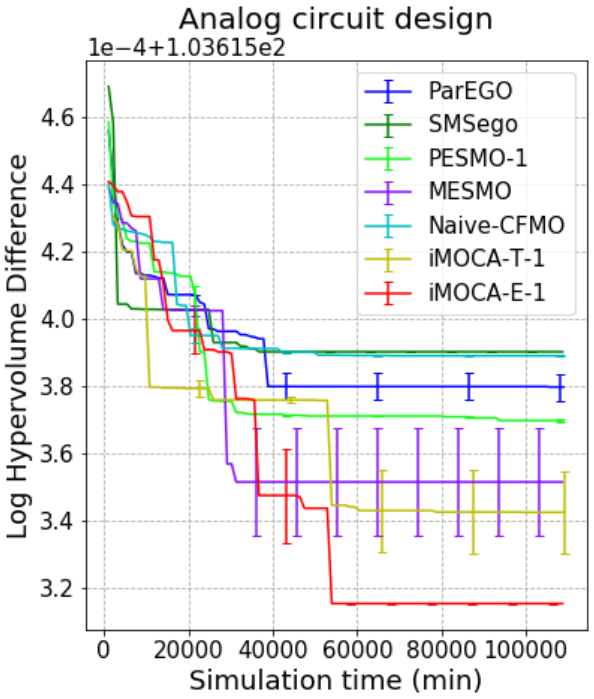
- iMOCA reduces the fidelity search space using a scheme similar to the BOCA algorithm

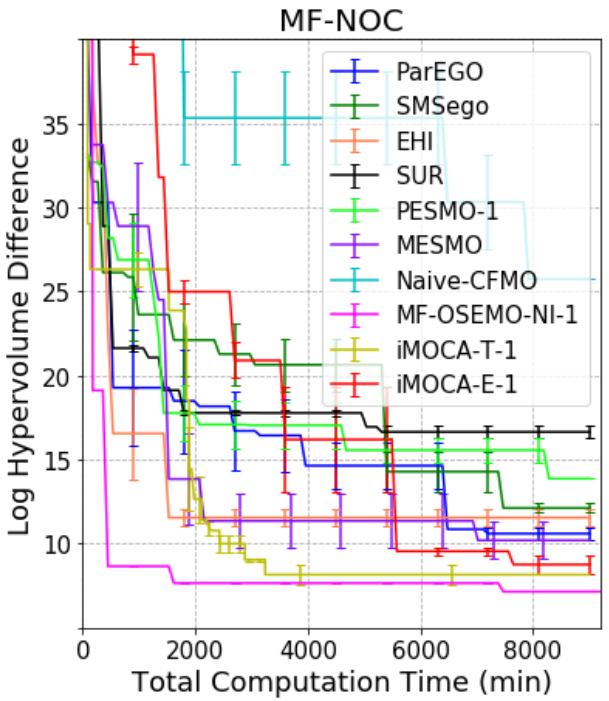
iMOCA Algorithm [Belakaria et al., 2021]



iMOCA Experiments and Results







- iMOCA performs better than all baselines
- Both variants of iMOCA converge at a much lower cost
- Robust to the number of samples

iMOCA Experiments and Results

- **Cost reduction factor**
  - Although the metric gives advantage to baselines, the results in the table show a consistently high gain ranging from 52% to 85%

Name	BC	ARS	Circuit	Rocket
$\mathcal{C}_B$	200	300	115000	9500
$\mathcal{C}$	30	100	55000	2000
$\mathcal{G}$	85%	66.6%	52.1%	78.9%

Table: *Best* convergence cost from all baselines $\mathcal{C}_B$, *Worst* convergence cost for iMOCA $\mathcal{C}$, and cost reduction factor $\mathcal{G}$.

Software and code

- github.com/HIPS/Spearmint/tree/PESM
- github.com/belakaria/MESMO
- github.com/belakaria/USeMO
- botorch.org/tutorials/multi_objective_bo
- github.com/yunshengtian/DGEMO
- github.com/belakaria/MESMOC
- github.com/belakaria/MF-OSEMO
- github.com/belakaria/iMOCA

Questions ?