Learning the Privacy-Utility Trade-off with Bayesian Optimization

Borja Balle

Joint work with B. Avent, J. Gonzalez, T. Diethe and A. Paleyes


Utility
Privacy

Theory *vs* Practice

$$O\left(\frac{\sqrt{d\log(1/\delta)}}{n\varepsilon}\right)$$

Plot from J. M. Abowd "Disclosure Avoidance for Block Level Data and Protection of Confidentiality in Public Tabulations" (CSAC Meeting, December 2018)

Example: DP-SGD

Input: dataset $z = (z_1, \ldots, z_n)$
Hyperparameters: learning rate η, mini-batch size m, number of epochs T, noise variance σ^2, clipping norm L
Initialize $w \leftarrow 0$
for $t \in [T]$ **do**
  for $k \in [n/m]$ **do**
    Sample $S \subset [n]$ with $|S| = m$ uniformly at random
    Let $g \leftarrow \frac{1}{m} \sum_{j \in S} \mathsf{clip}_L(\nabla \ell(z_j, w)) + \frac{2L}{m} \mathcal{N}(0, \sigma^2 I)$
    Update $w \leftarrow w - \eta g$
return w

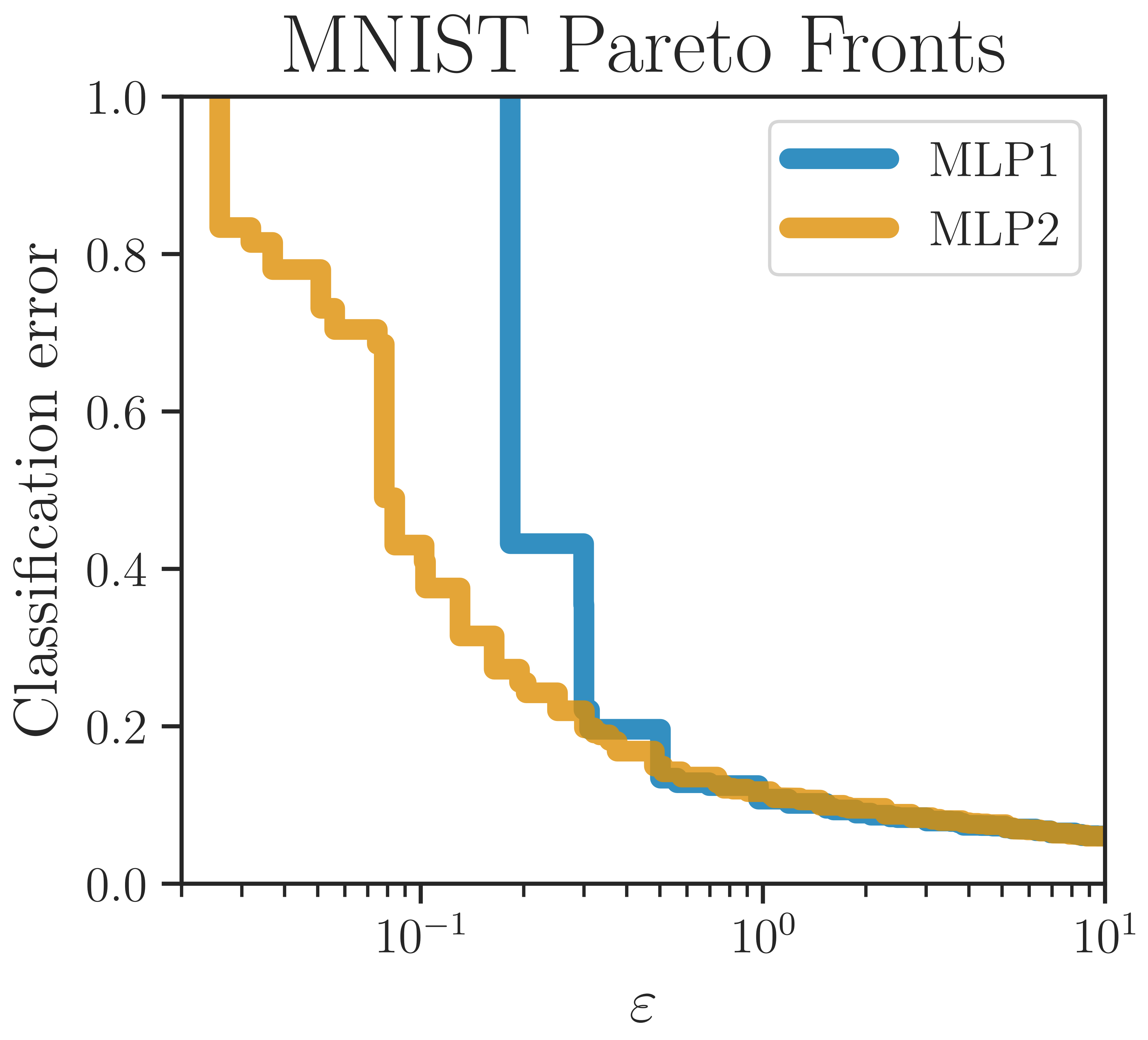
- 5+ hyper-parameters affecting both privacy and utility
- For convex problems can be set to achieve near-optimal rates
- For deep learning applications we don't have (good) utility bounds

[Bassily et al. 2014; Abadi et al. 2016]

Privacy-Utility Pareto Front

Desiderata

1. Efficient to compute
2. Use empirical utility measurements
3. Enable fine-grained comparisons



Problem Formulation

Parametrized Algorithm Class

$$\mathcal{A} = \{A_\lambda : Z \to W \mid \lambda \in \Lambda\}$$

Eg. DP-SGD

Error (Utility) Oracle

$$E : \Lambda \to [0, 1]$$

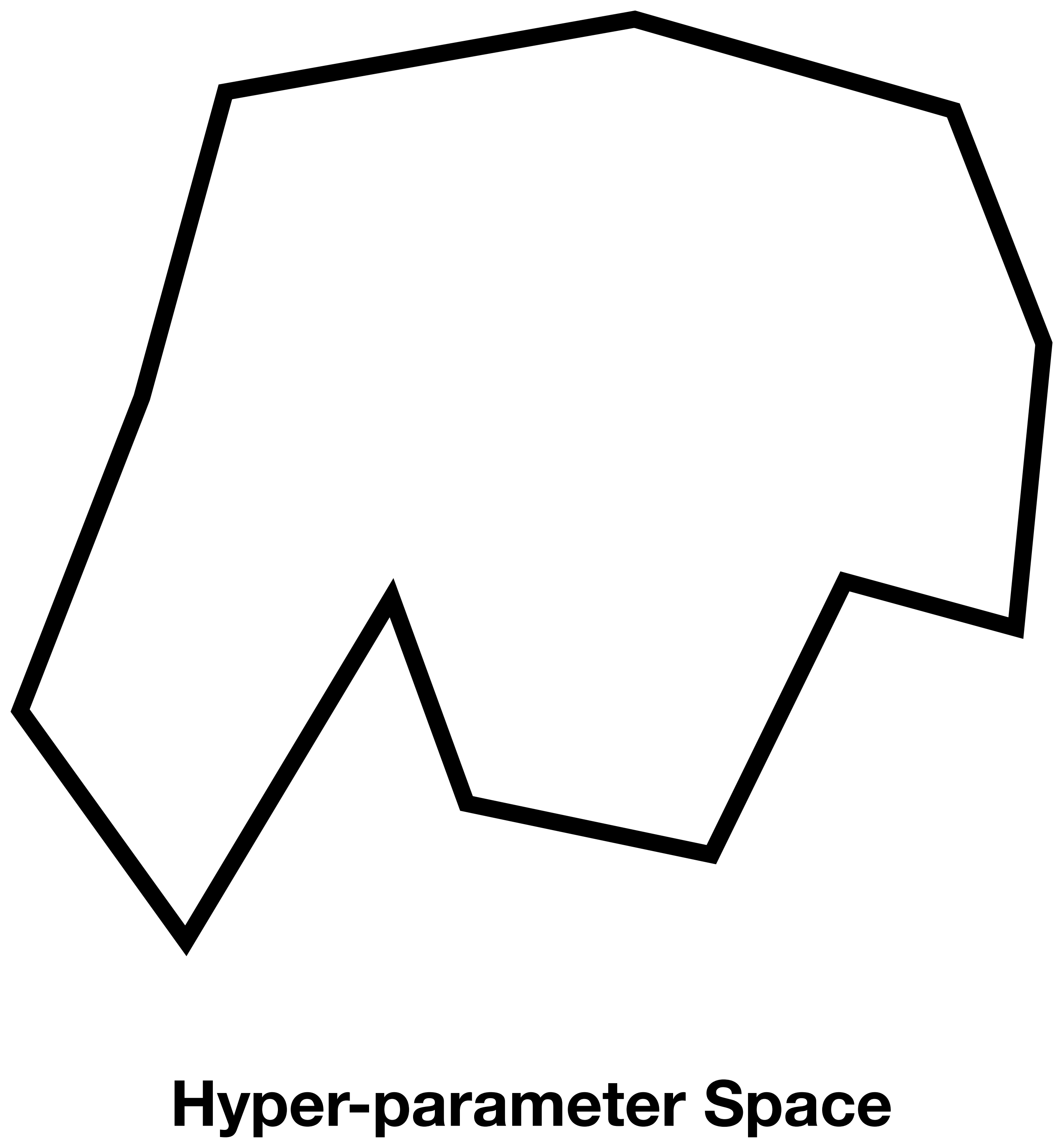
Eg. Expected classification error

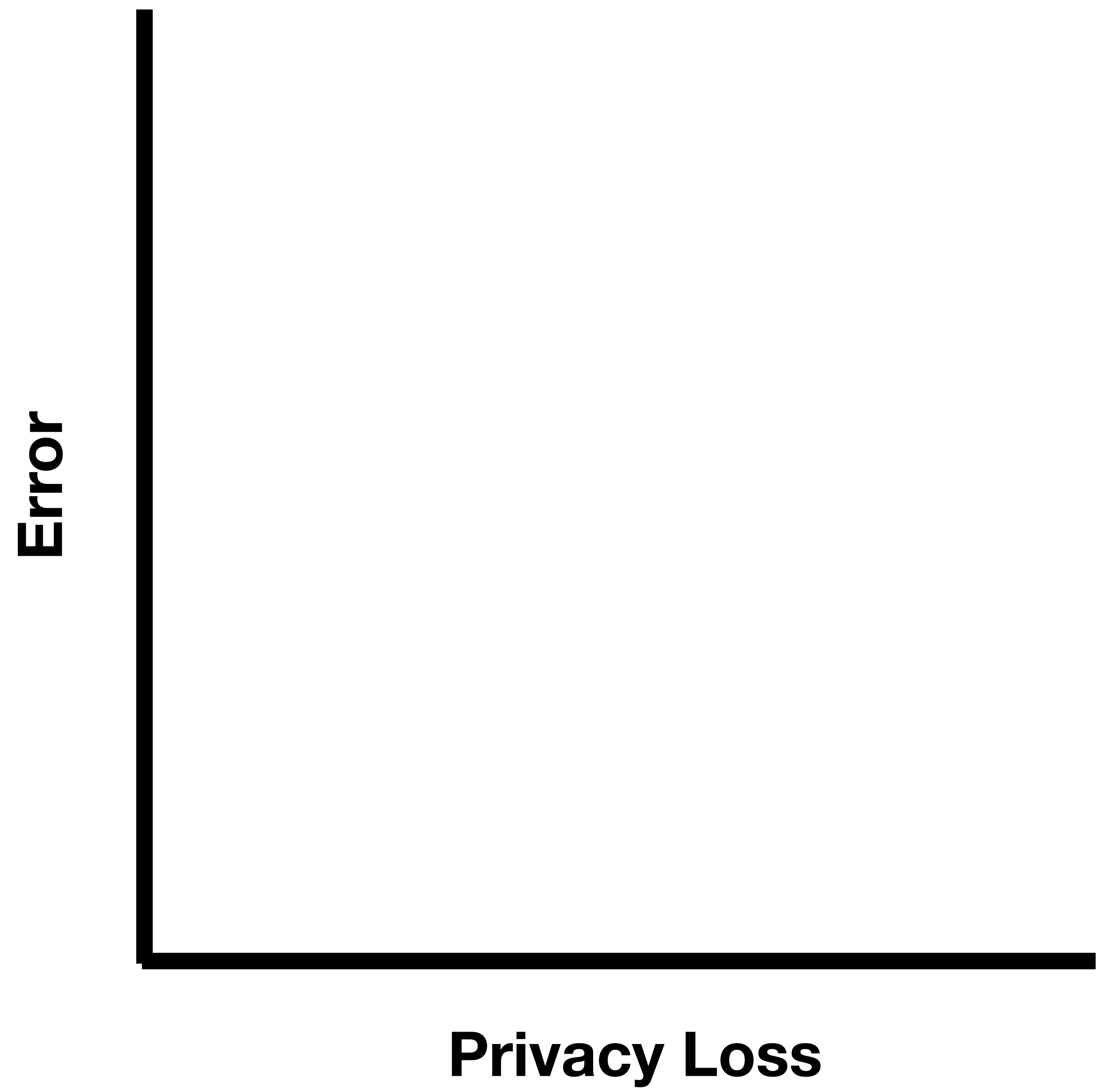
Privacy Oracle

$$P : \Lambda \to [0, \infty)$$

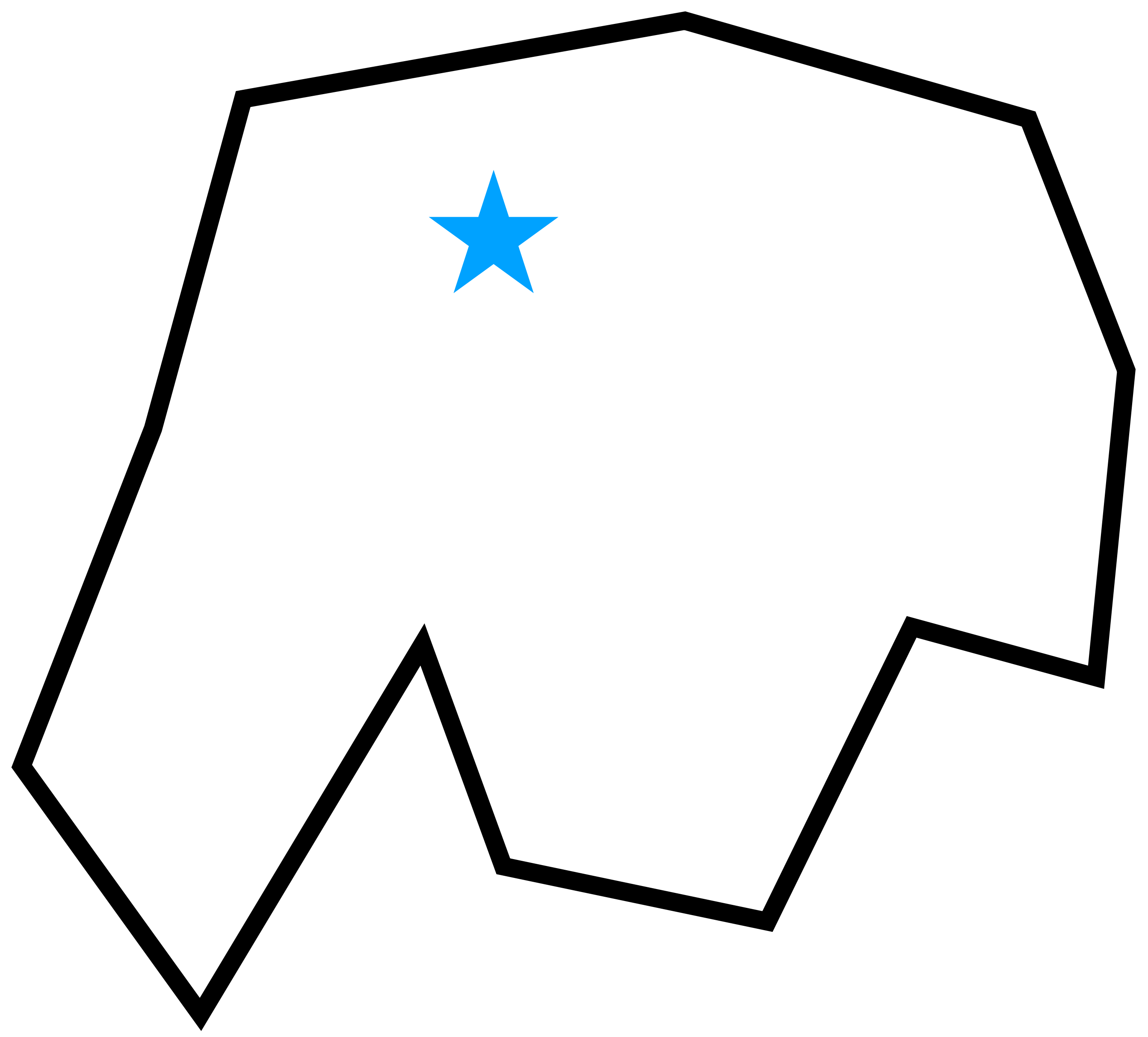
Eg. Epsilon for fixed delta

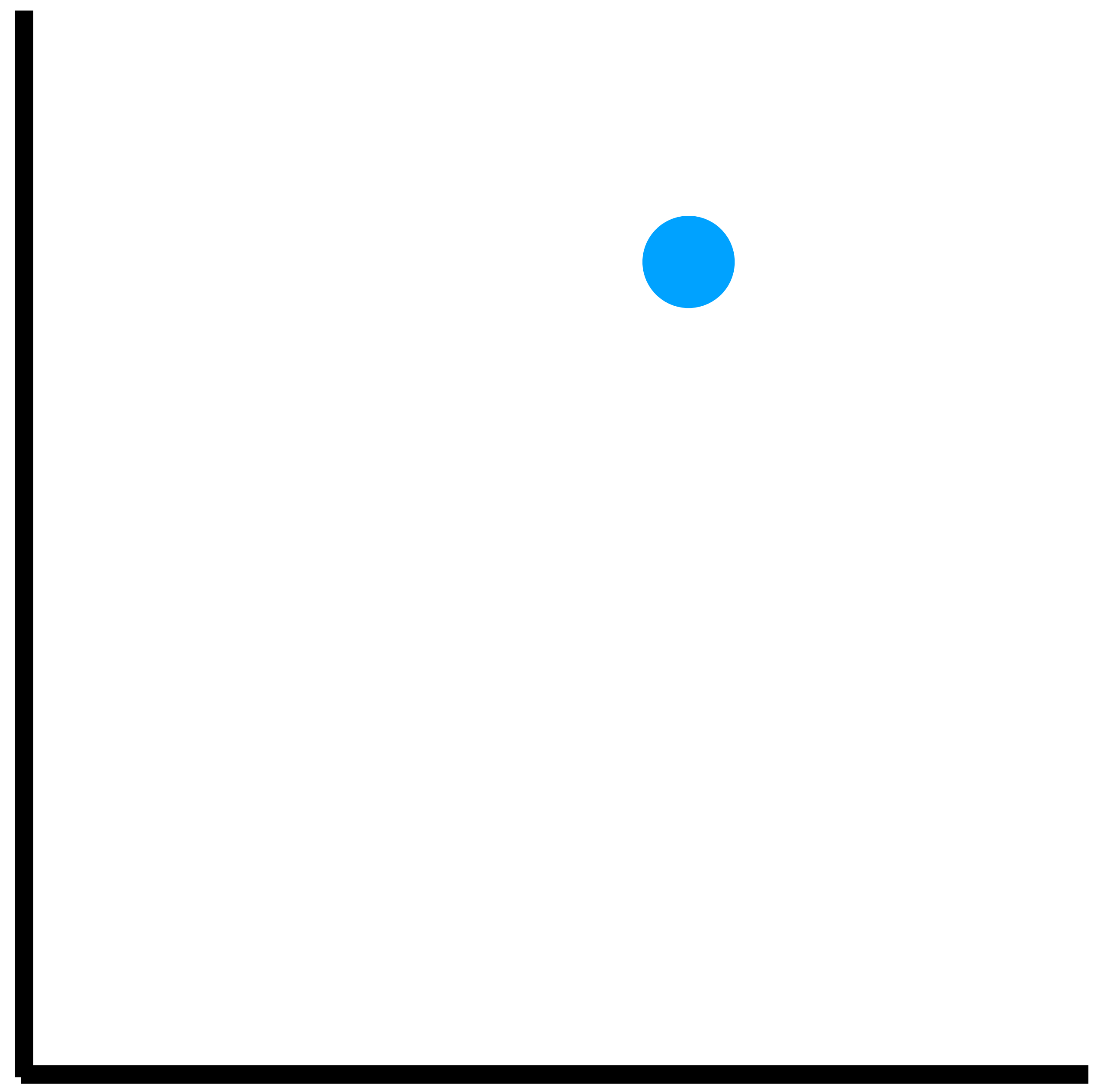
Pareto-Optimal Points



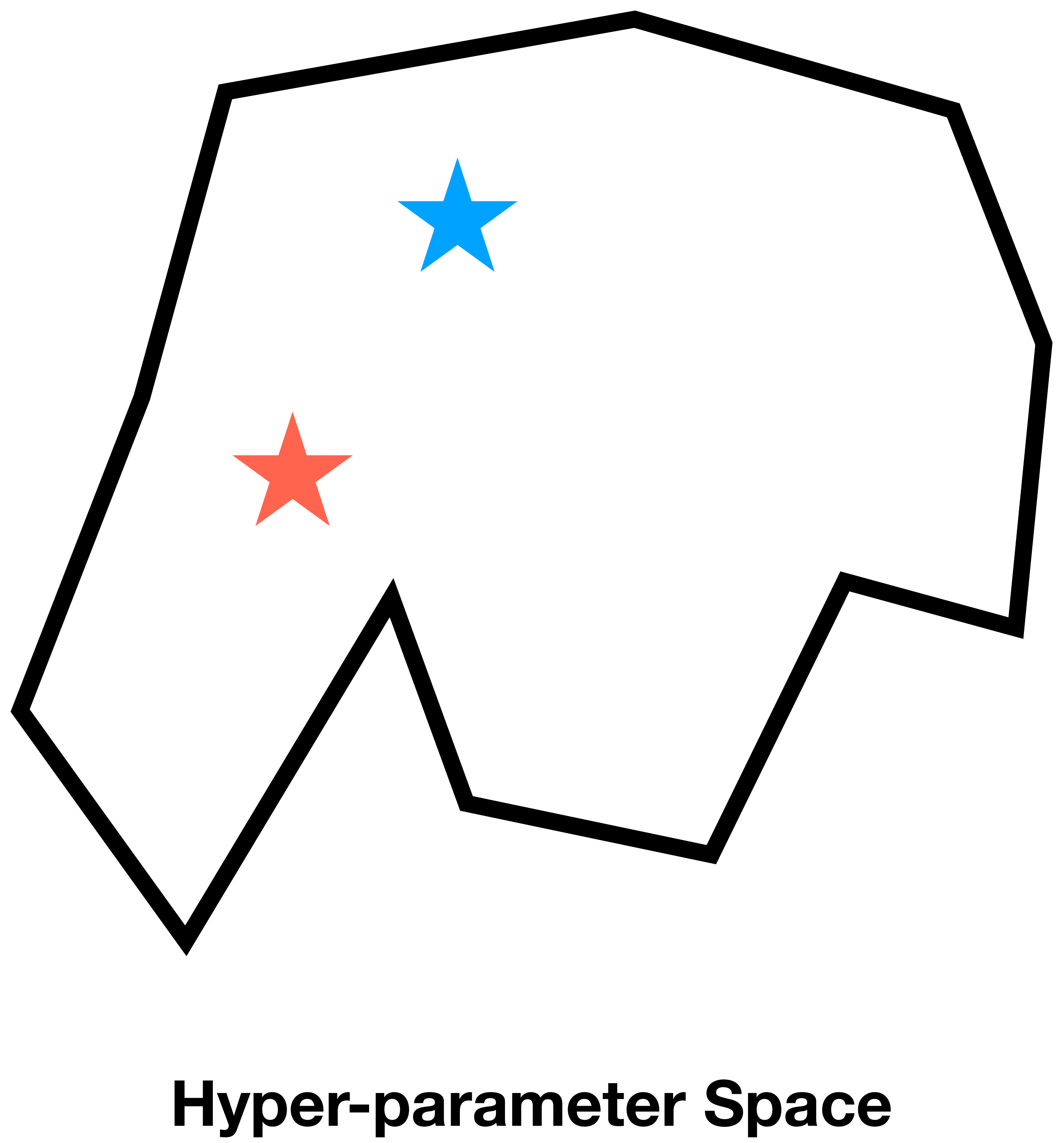


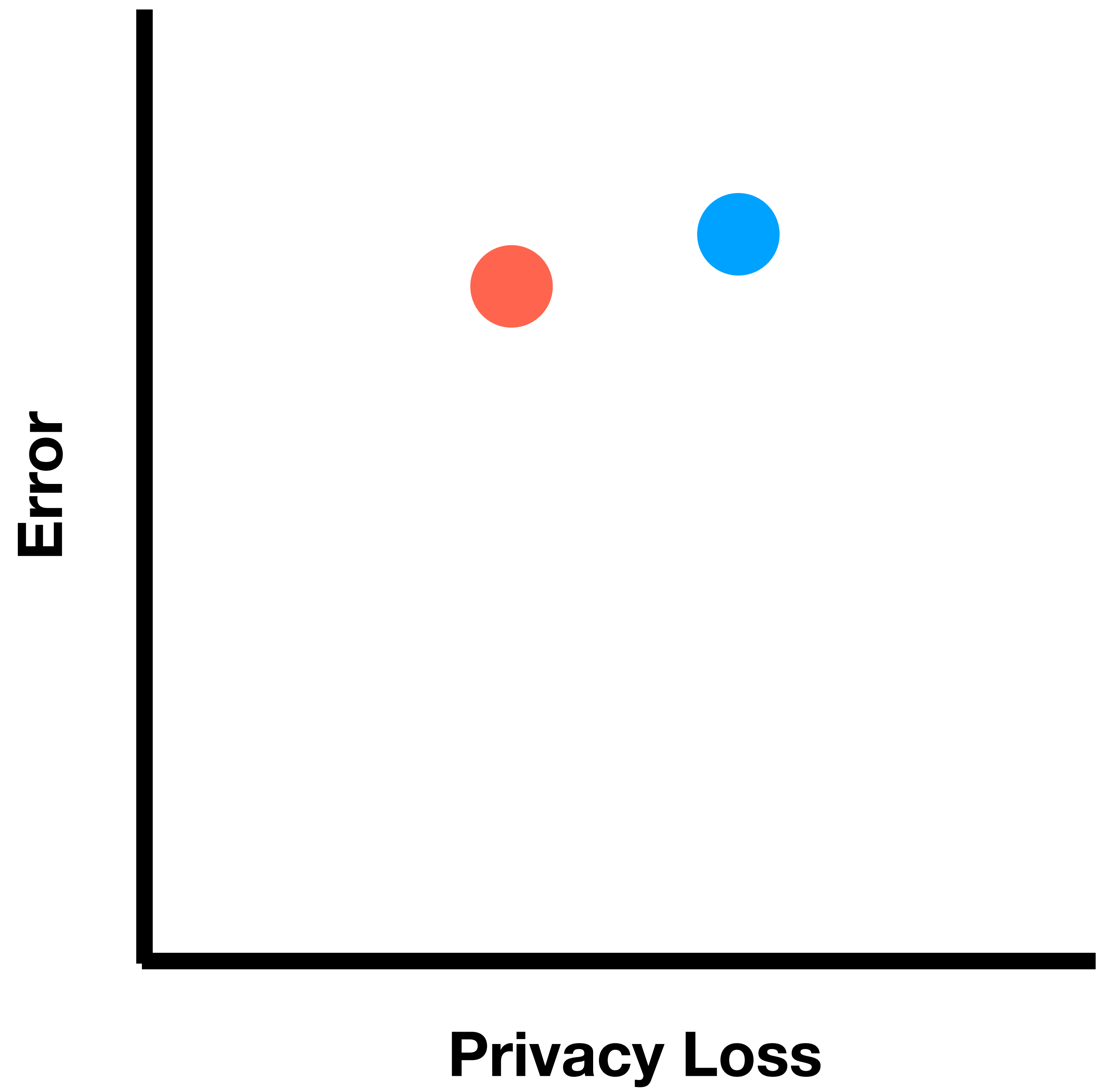
Pareto-Optimal Points



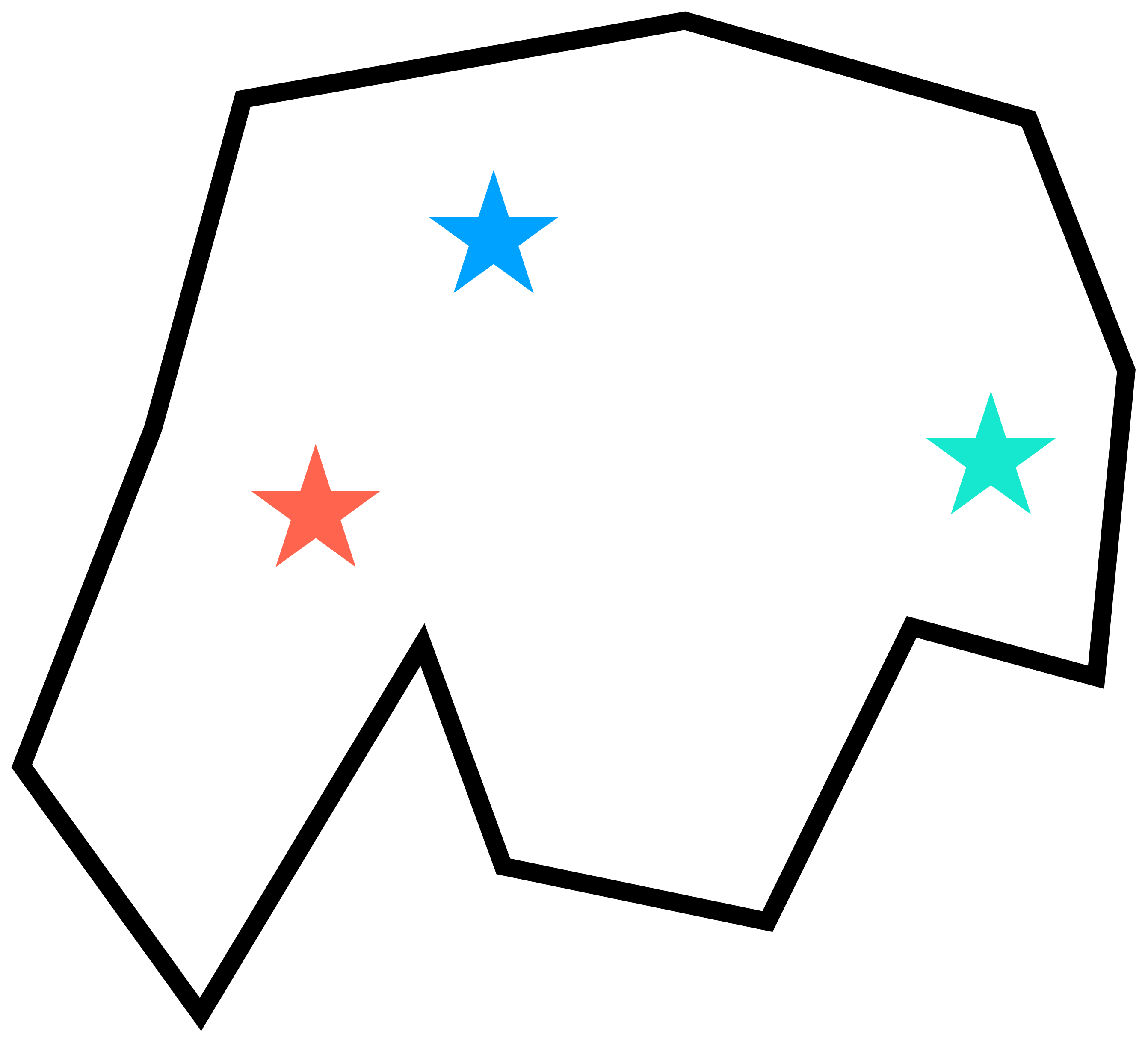


Pareto-Optimal Points

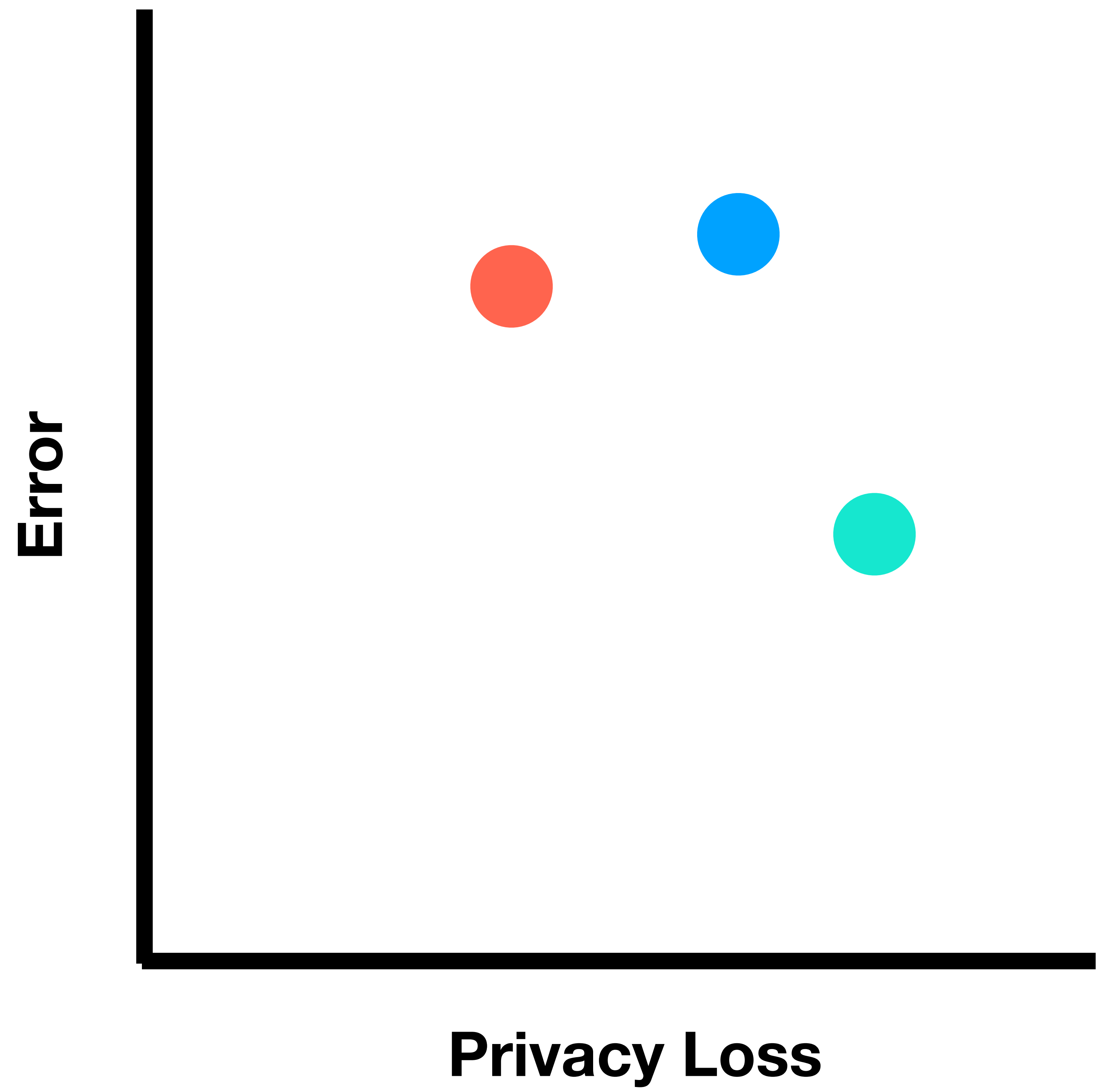




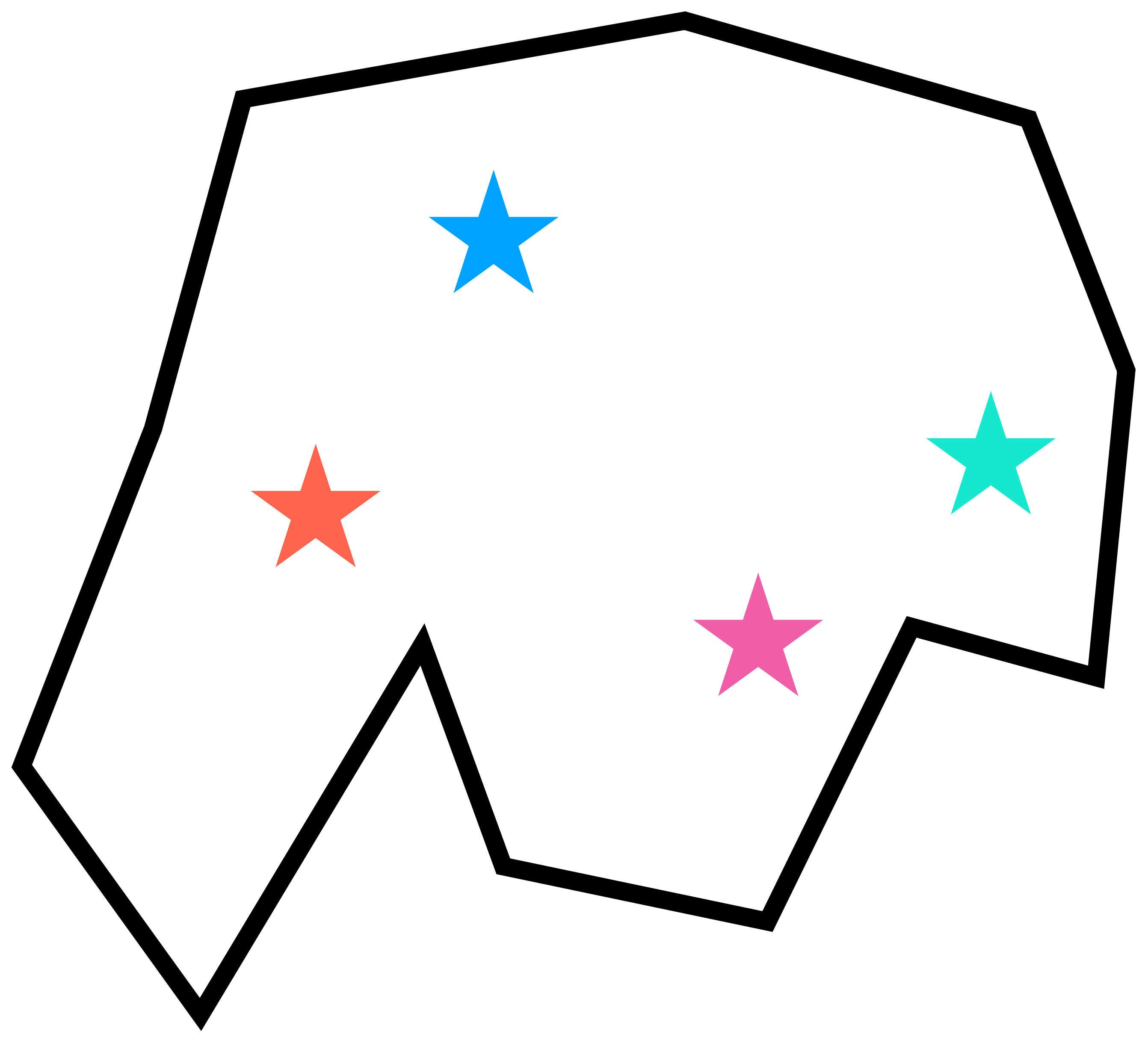
Pareto-Optimal Points



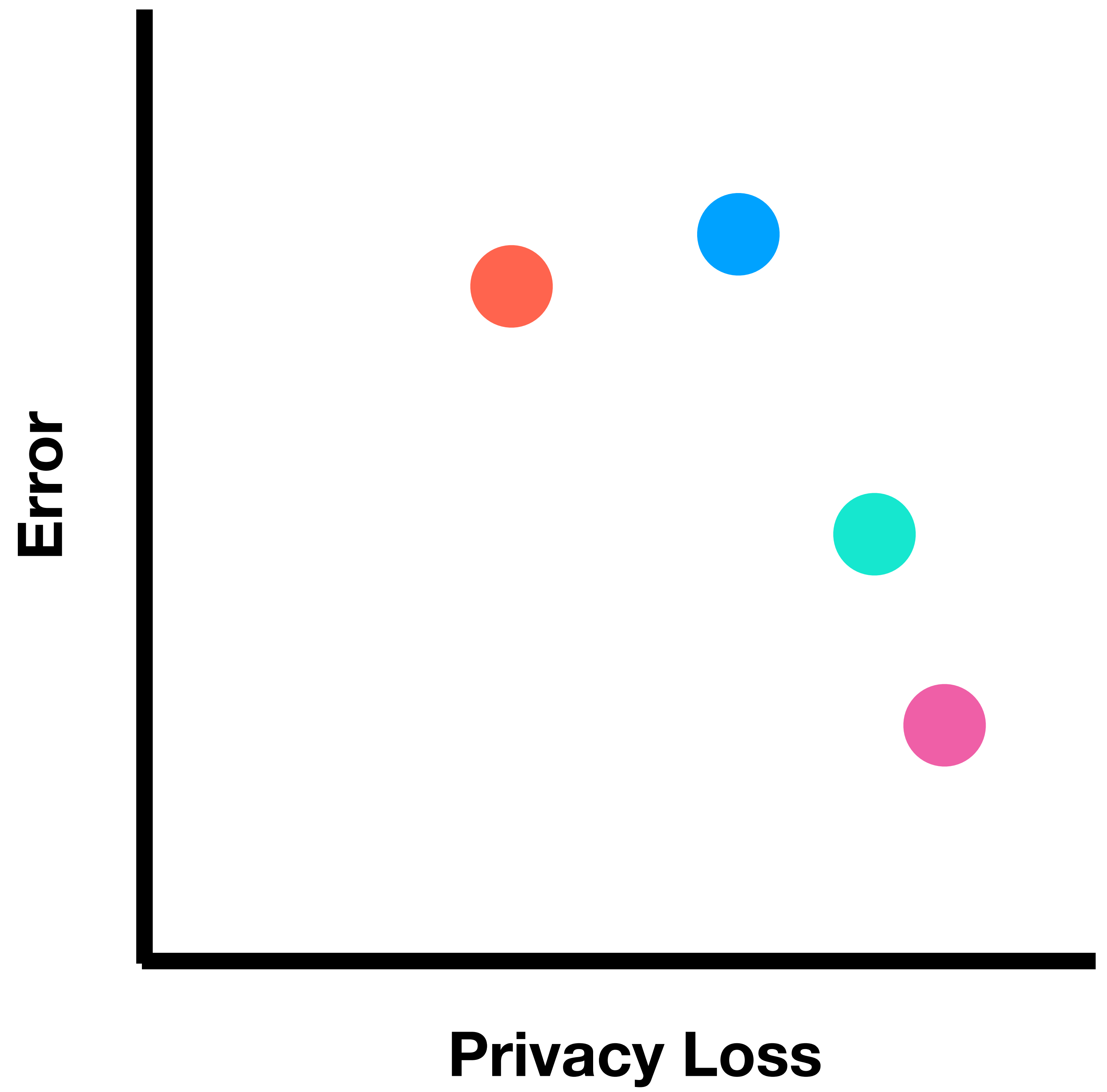
Hyper-parameter Space



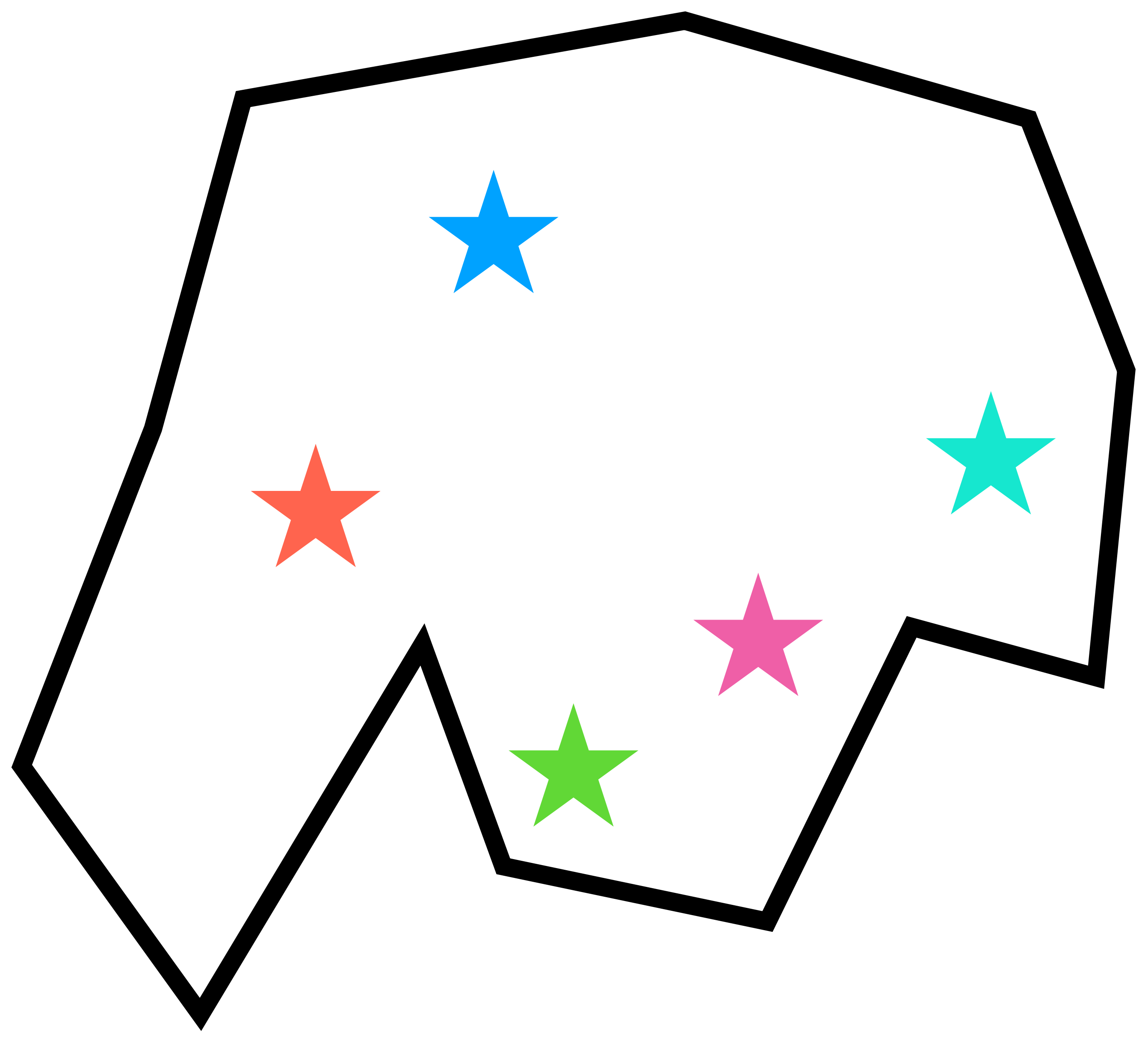
Pareto-Optimal Points



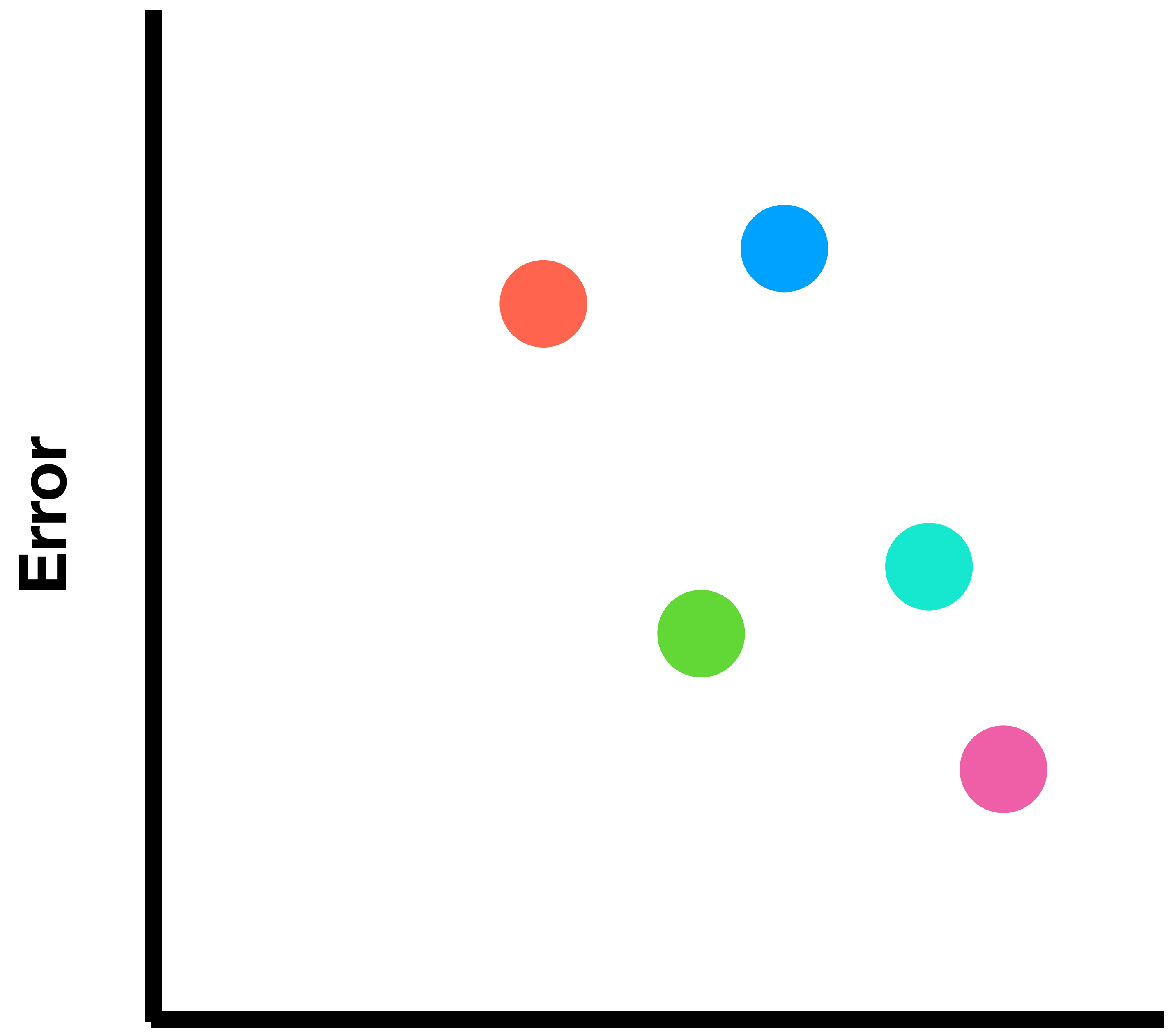
Hyper-parameter Space



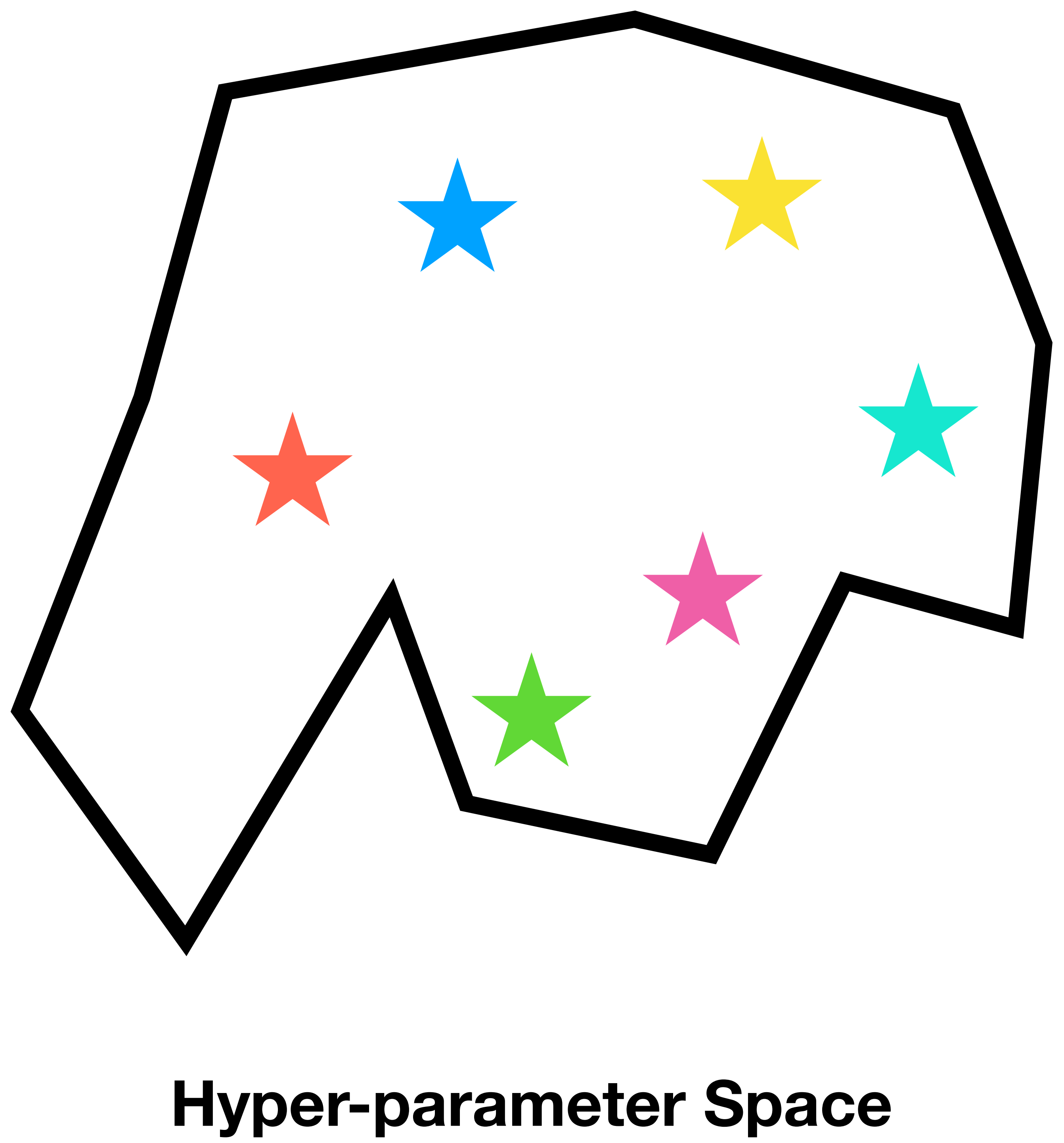
Pareto-Optimal Points

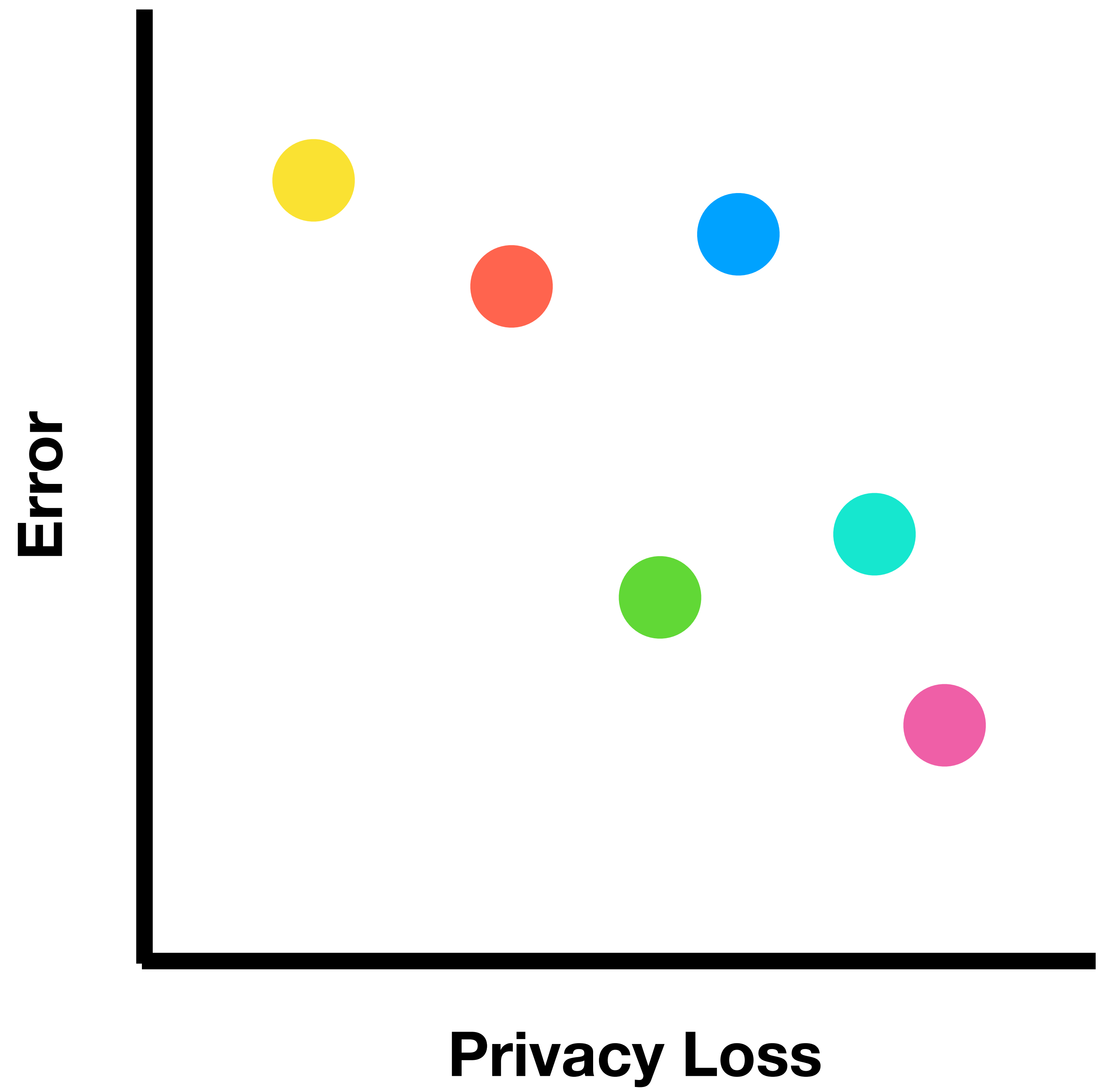


Hyper-parameter Space

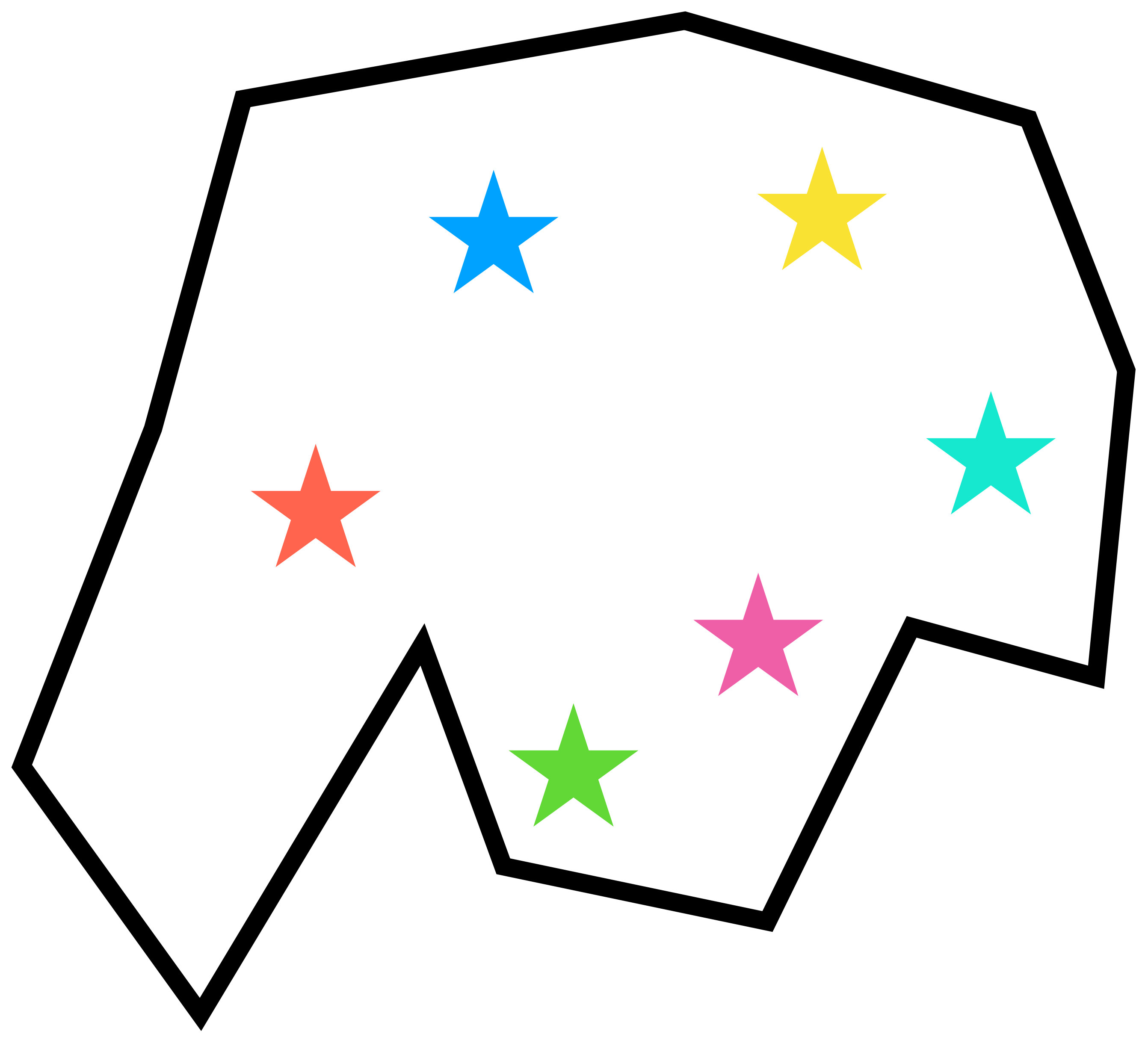


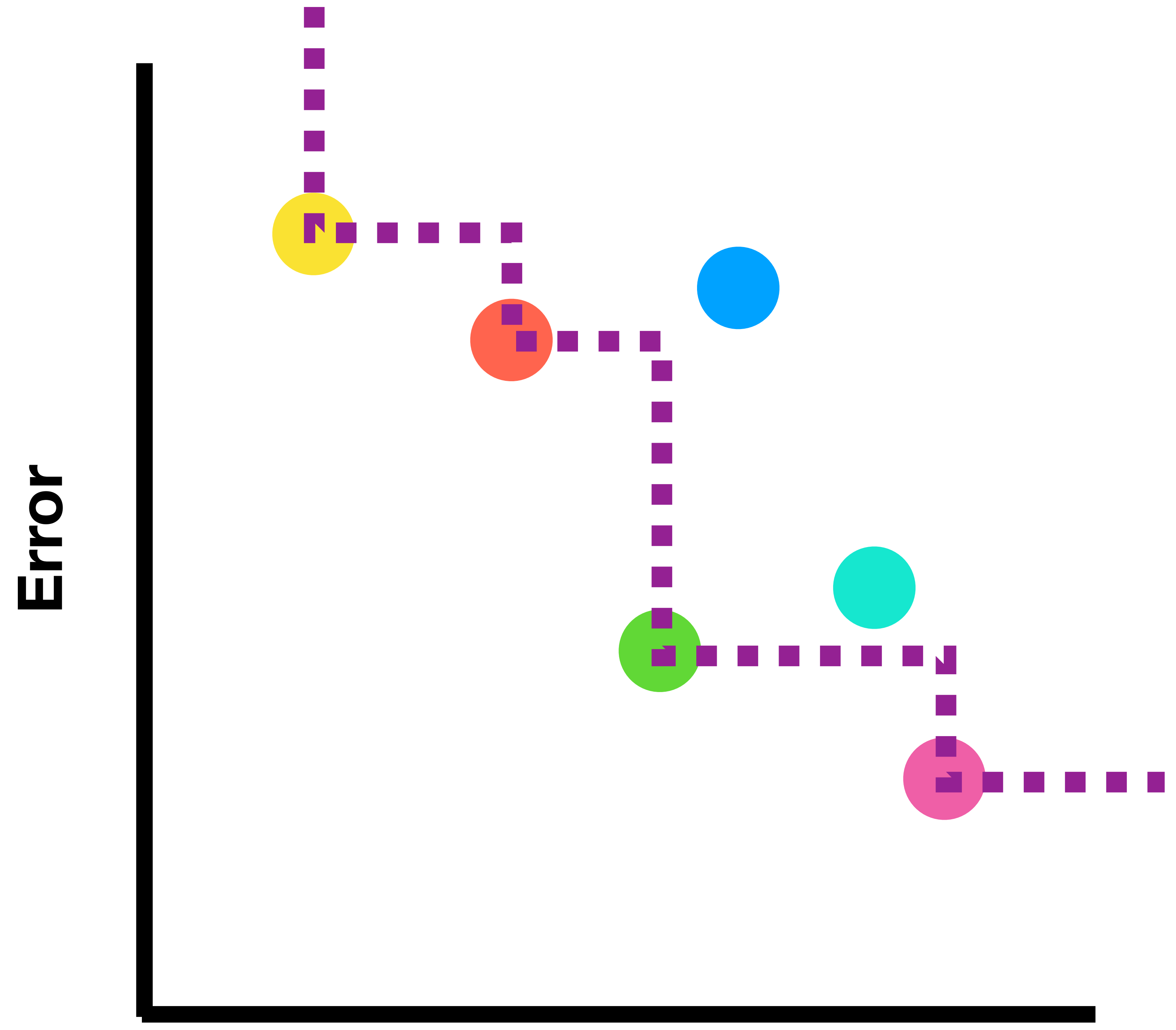
Pareto-Optimal Points





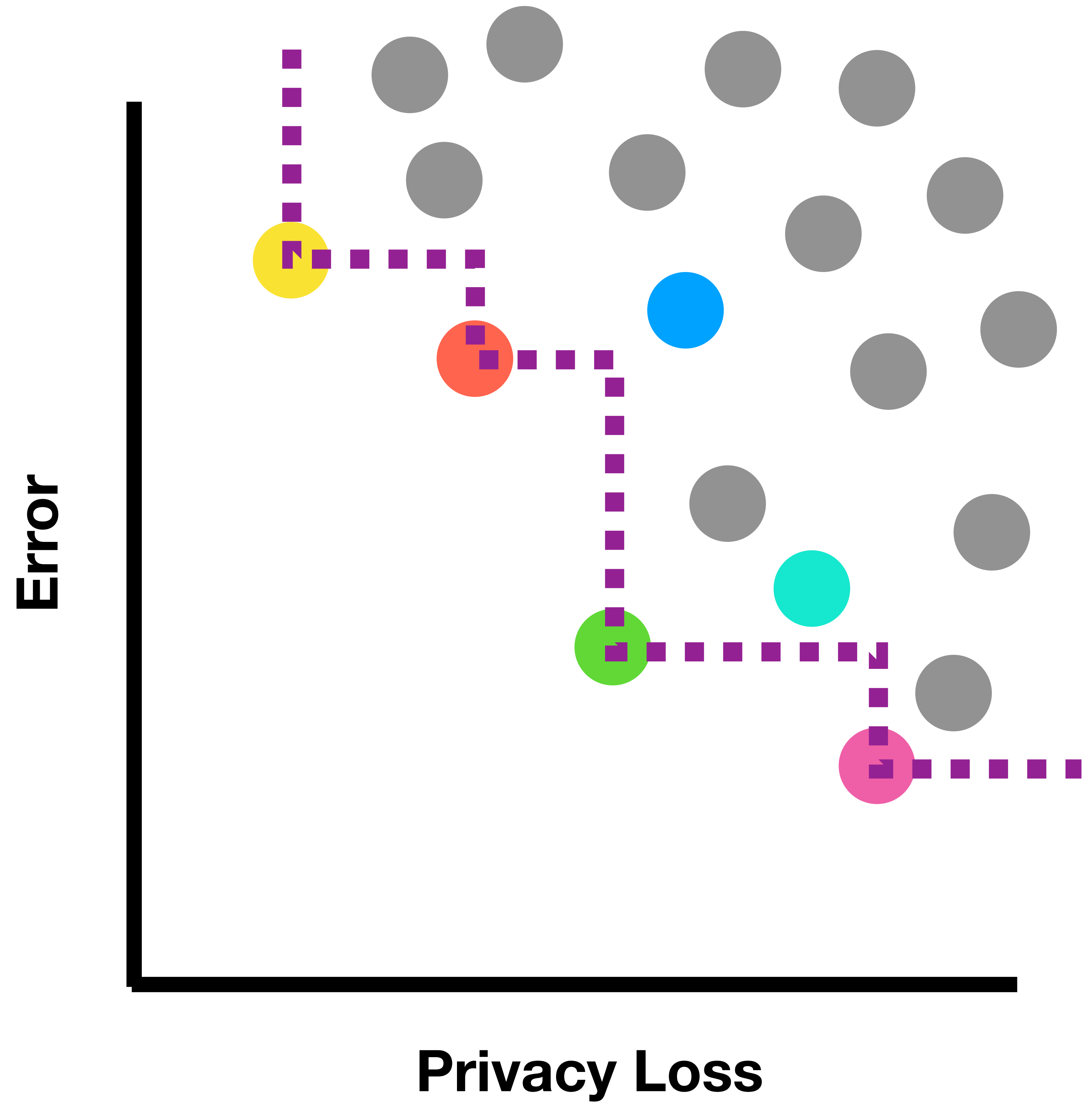
Pareto-Optimal Points





Pareto-Optimal Points

Hyper-parameter Space



Bayesian Optimization (BO)

- Gradient-free optimization for black-box functions
- Widely used in applications (HPO in ML, scheduling & planning, experimental design, etc)

Bayesian Optimization (BO)

- Gradient-free optimization for black-box functions

- Widely used in applications (HPO in ML, scheduling & planning, experimental design, etc)

Input: $F : \Lambda \subset \mathbb{R}^p \rightarrow \mathbb{R}$

Expensive, non-convex, smooth

Goal: $\lambda^\star = \underset{\lambda \in \Lambda}{\operatorname{argmin}}\, F(\lambda)$

Bayesian Optimization (BO)

- Gradient-free optimization for black-box functions

- Widely used in applications (HPO in ML, scheduling & planning, experimental design, etc)

Input: $F : \Lambda \subset \mathbb{R}^p \to \mathbb{R}$

Expensive, non-convex, smooth

Goal: $\lambda^\star = \underset{\lambda \in \Lambda}{\operatorname{argmin}} F(\lambda)$

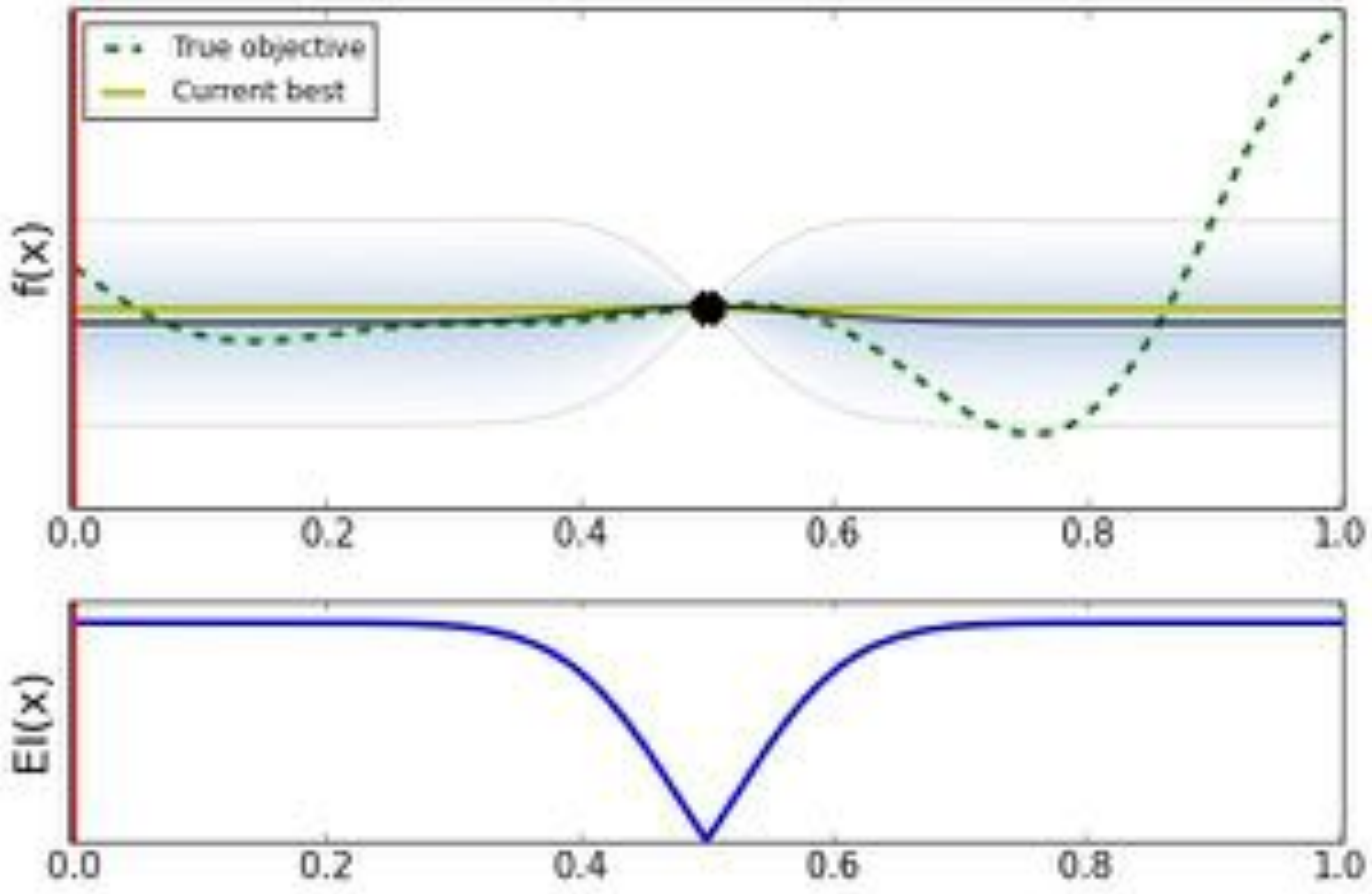
Bayesian Optimization Loop:

Given k evaluations $(\lambda_1, F(\lambda_1)), \ldots, (\lambda_k, F(\lambda_k))$

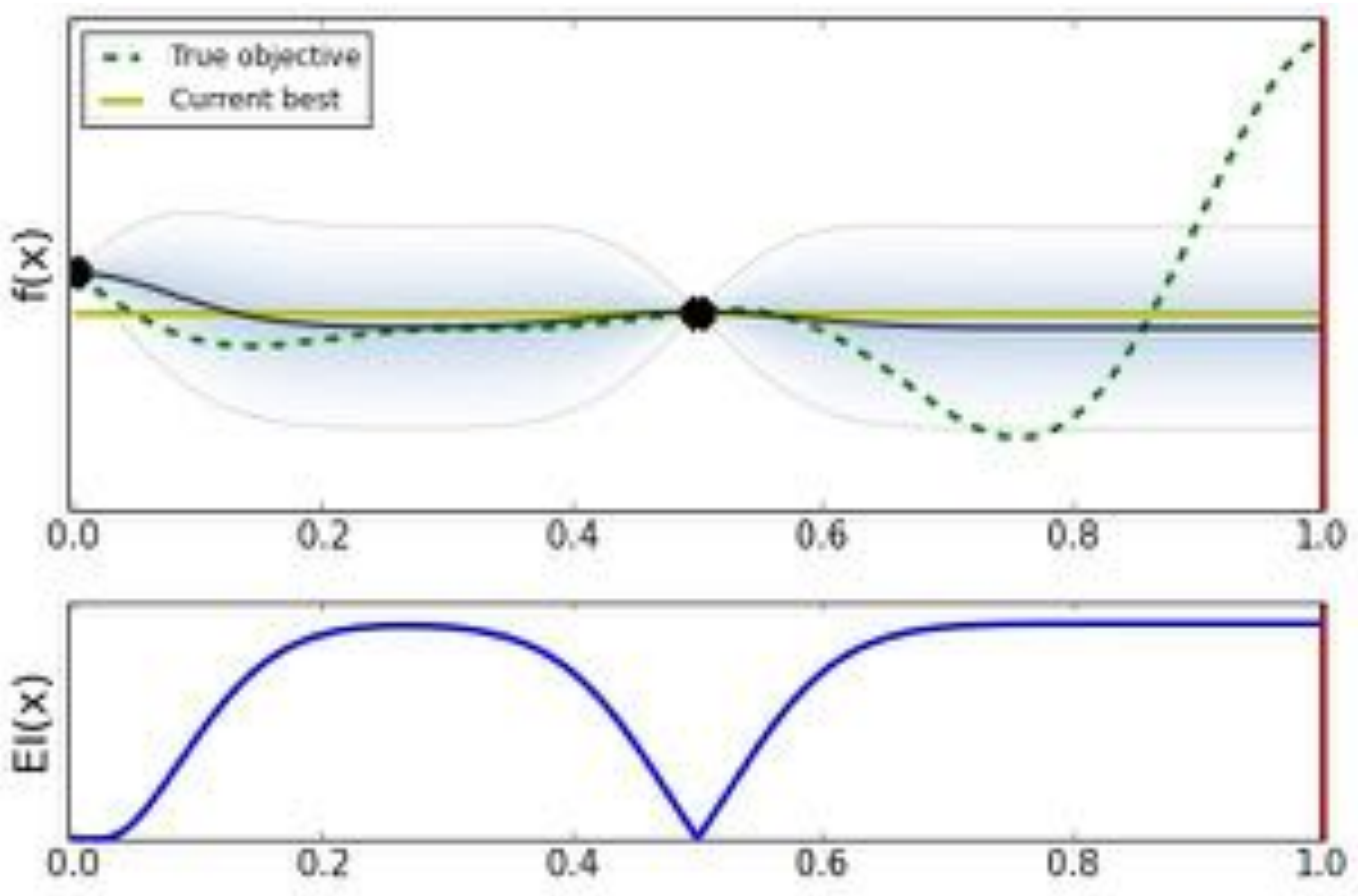
1. Build a surrogate model for F (eg. Gaussian process)

2. Find most promising next evaluation

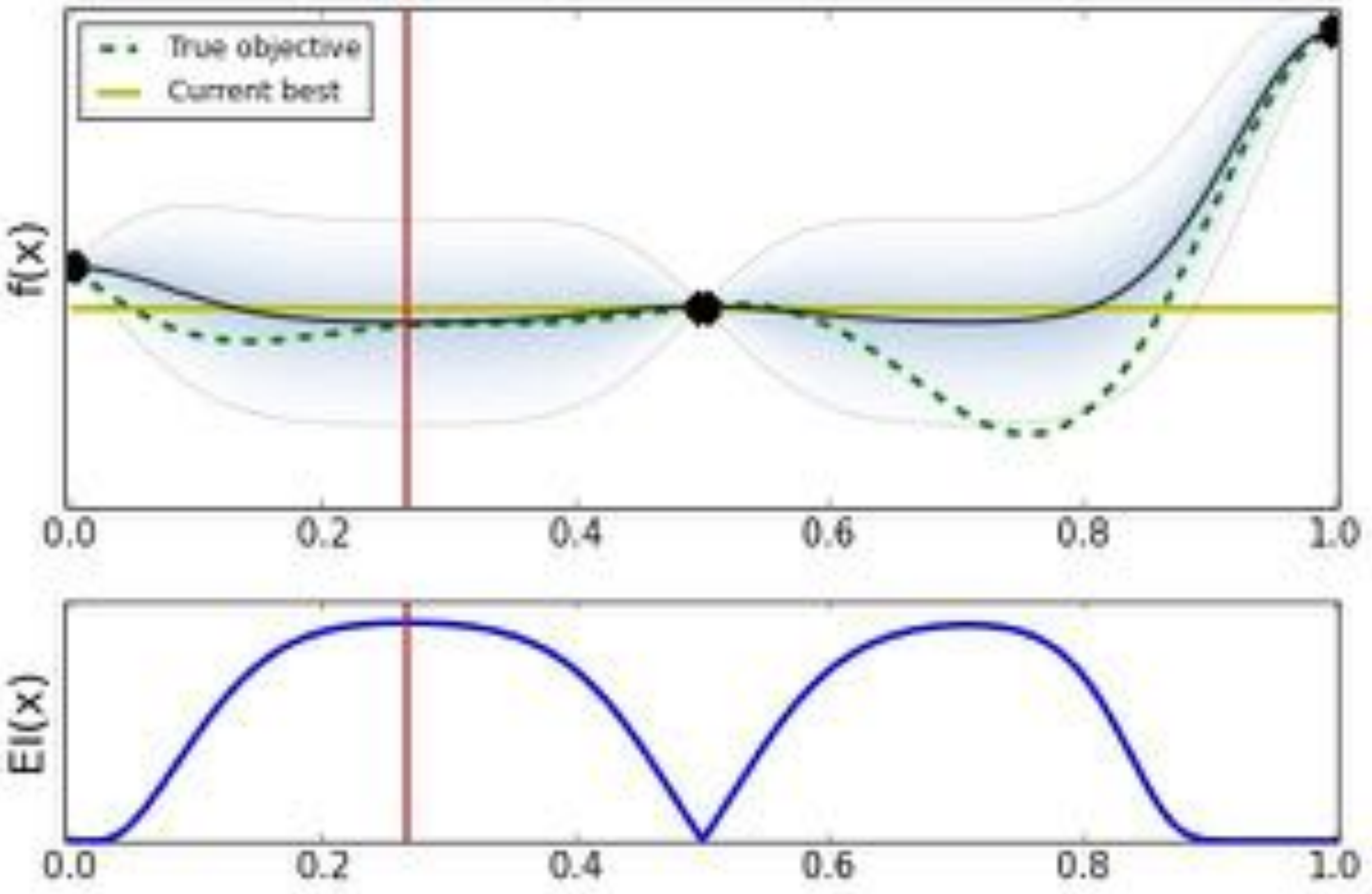
BO: 1-Dimensional Example



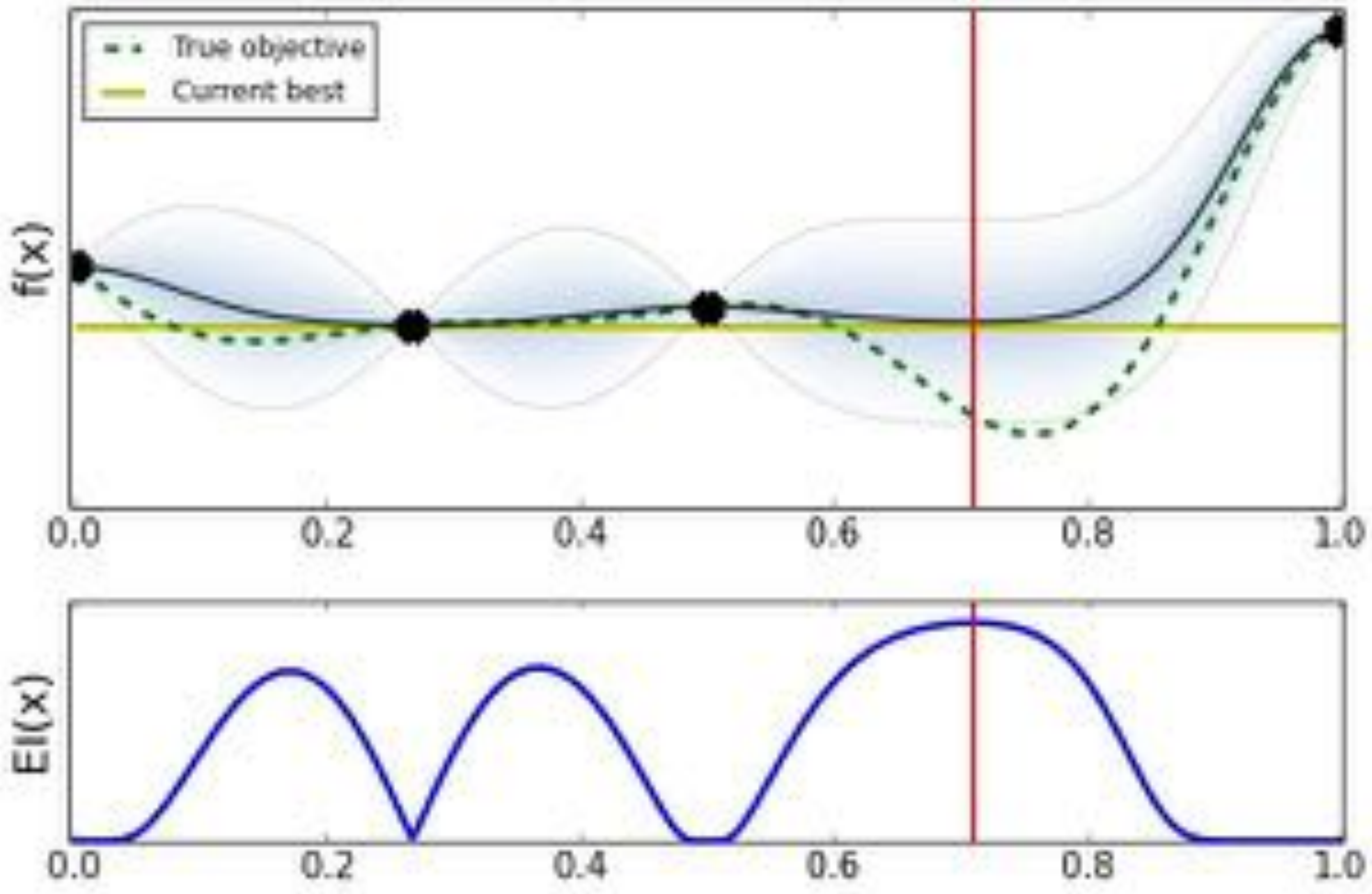
BO: 1-Dimensional Example



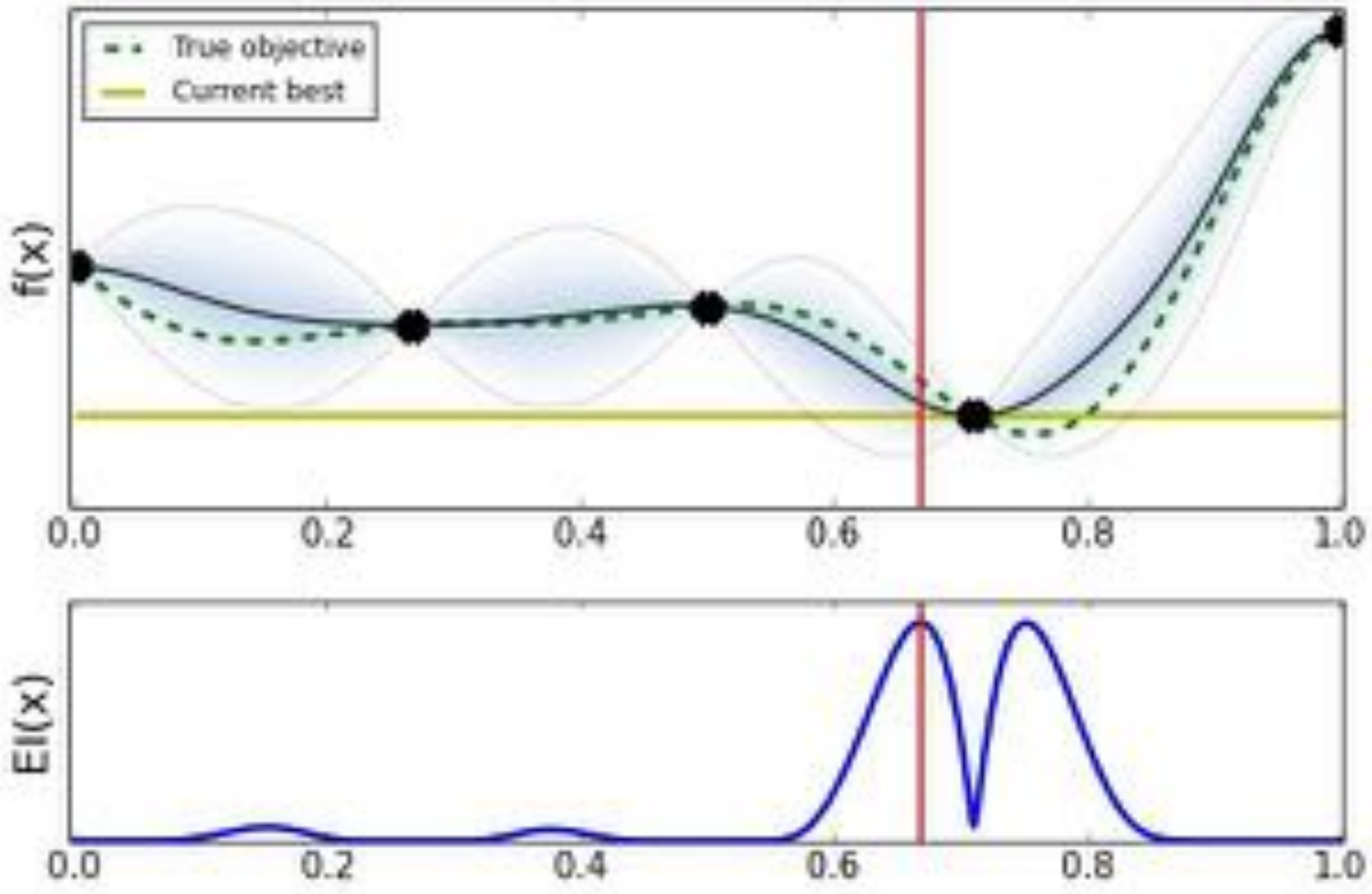
BO: 1-Dimensional Example



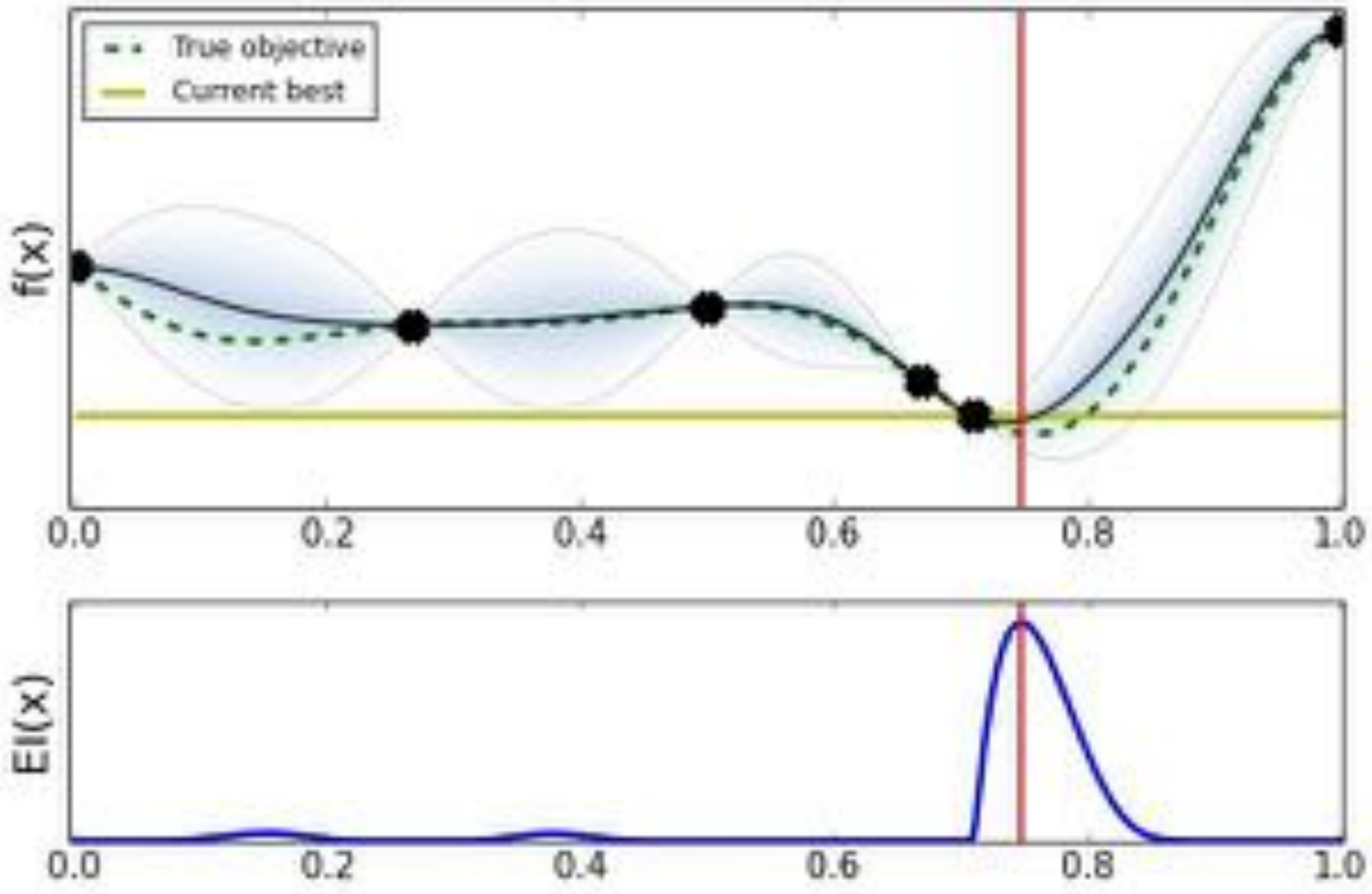
BO: 1-Dimensional Example



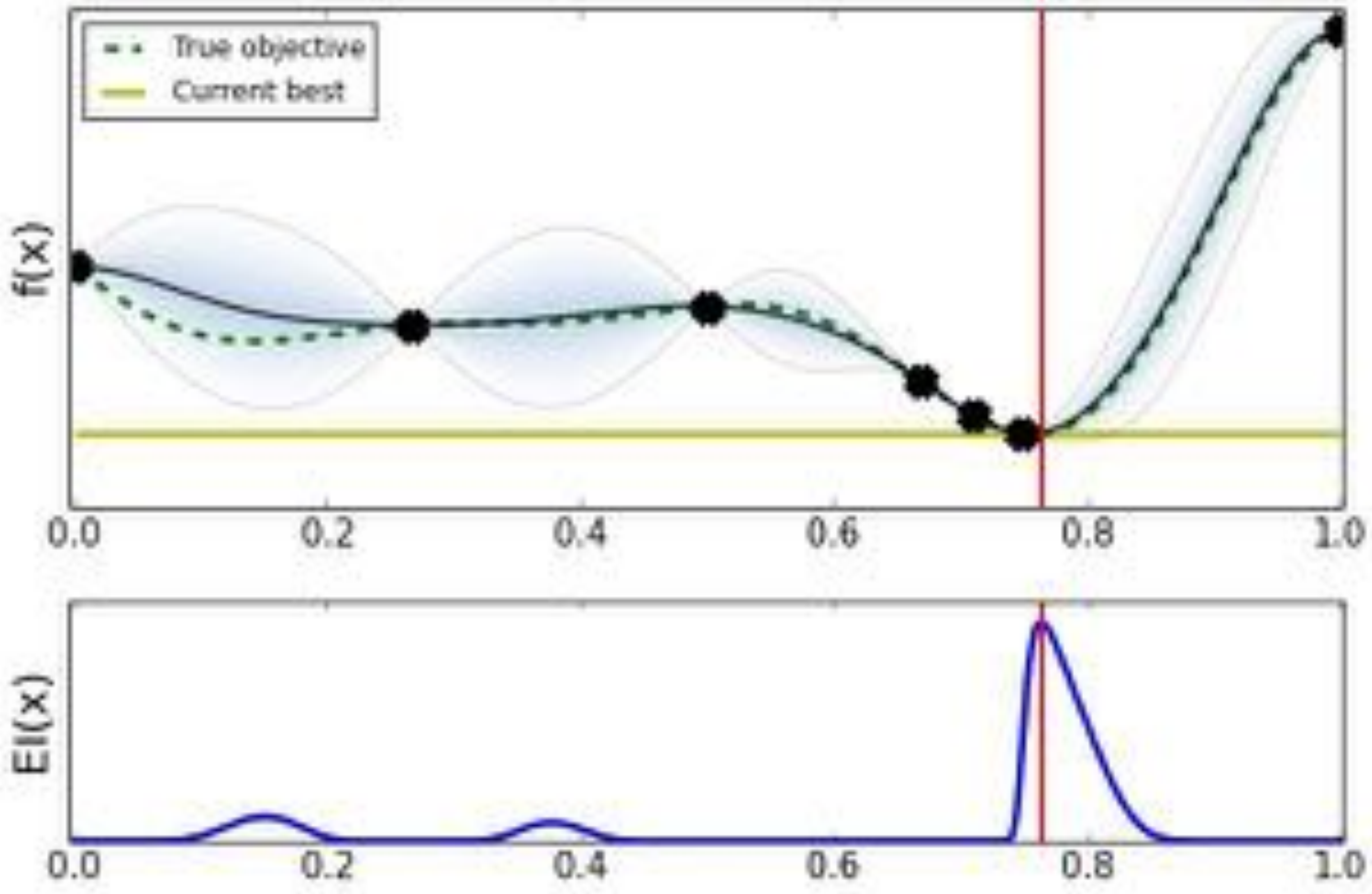
BO: 1-Dimensional Example



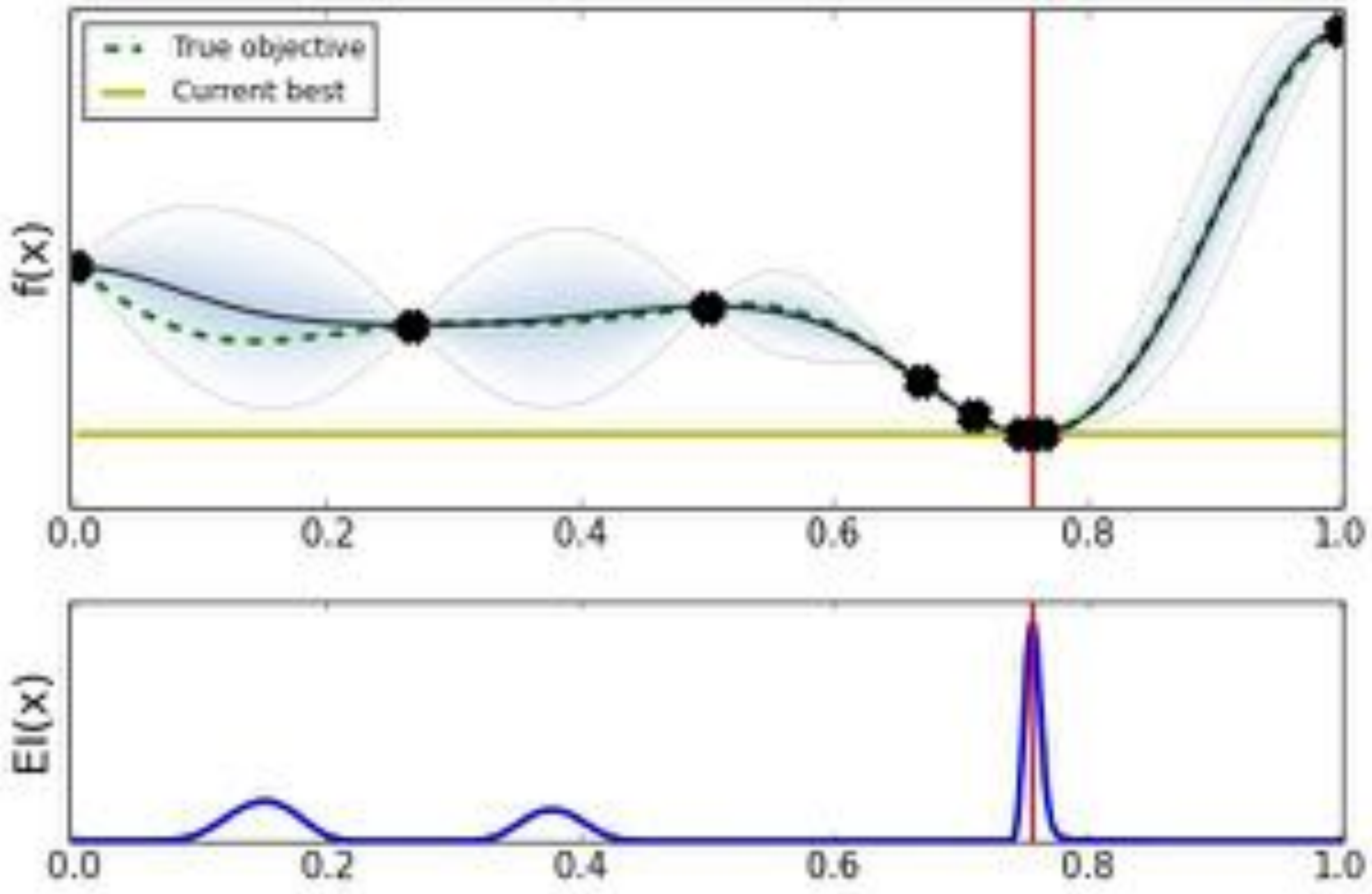
BO: 1-Dimensional Example



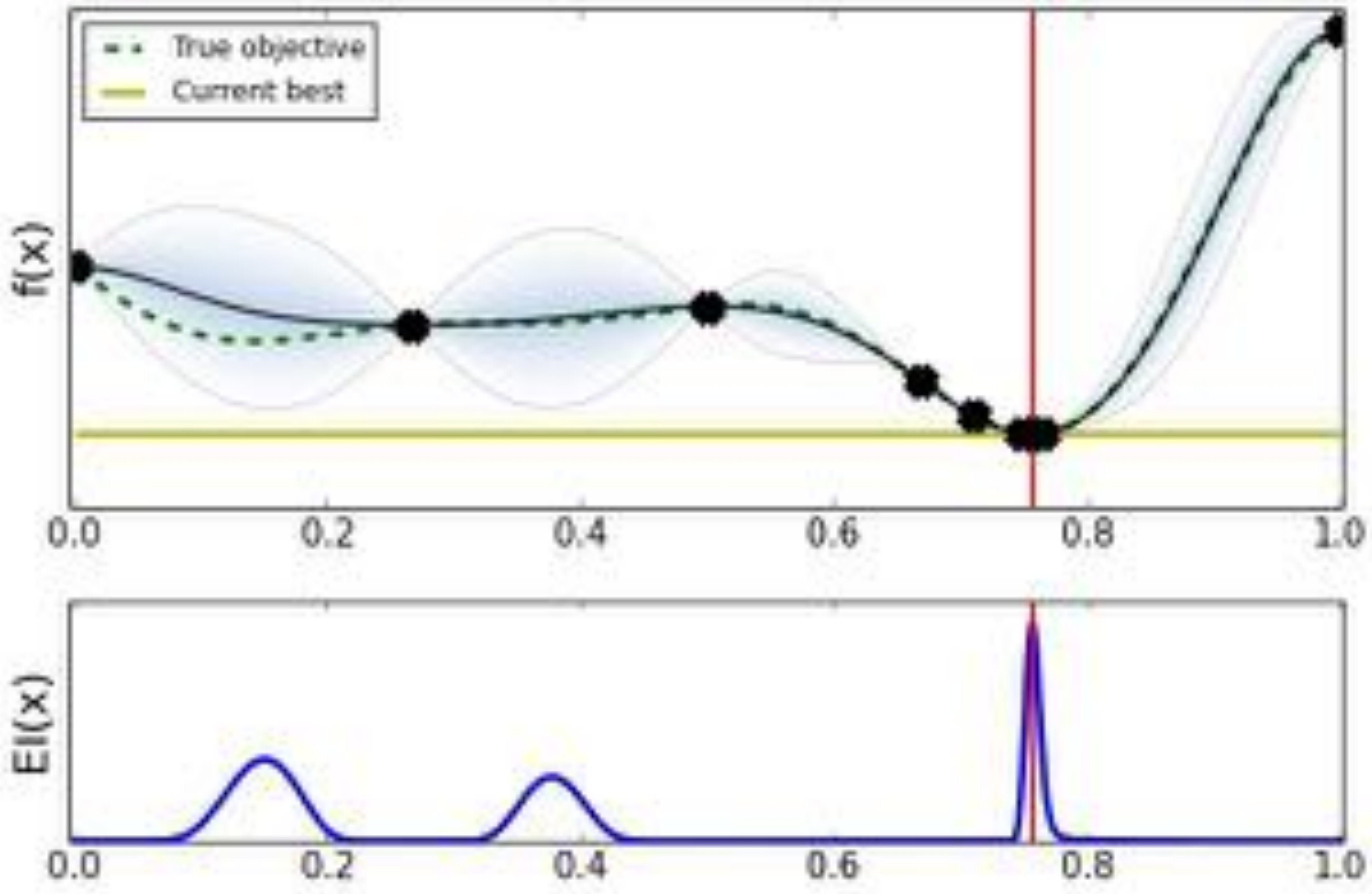
BO: 1-Dimensional Example



BO: 1-Dimensional Example



BO: 1-Dimensional Example



The DPareto Algorithm

- Find privacy-utility Pareto front using *multi-objective* Bayesian optimization
- Use transformed Gaussian processes to model privacy and error oracles
- Acquisition function optimizes *hyper-volume based probability of improvement* [Couckuyt et al. 2014]

Input: hyperparameter set Λ, privacy oracle P, error oracle E, anti-ideal point $v^\dagger$, number of initial points k_0, number of iterations k, prior GP

Initialize dataset $\mathcal{D} \leftarrow \emptyset$

for $i \in [k_0]$ **do**

  Sample random point $\lambda \in \Lambda$

  Evaluate oracles $v \leftarrow (\mathsf{P}(\lambda), \mathsf{E}(\lambda))$

  Augment dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\lambda, v)\}$

for $i \in [k]$ **do**

  Fit a GP to the transformed privacy using $\mathcal{D}$

  Fit a GP to the transformed utility using $\mathcal{D}$

  Optimize the HVPoI acquisition function in Eq. (2) using anti-ideal point $v^\dagger$ and obtain a new query point λ

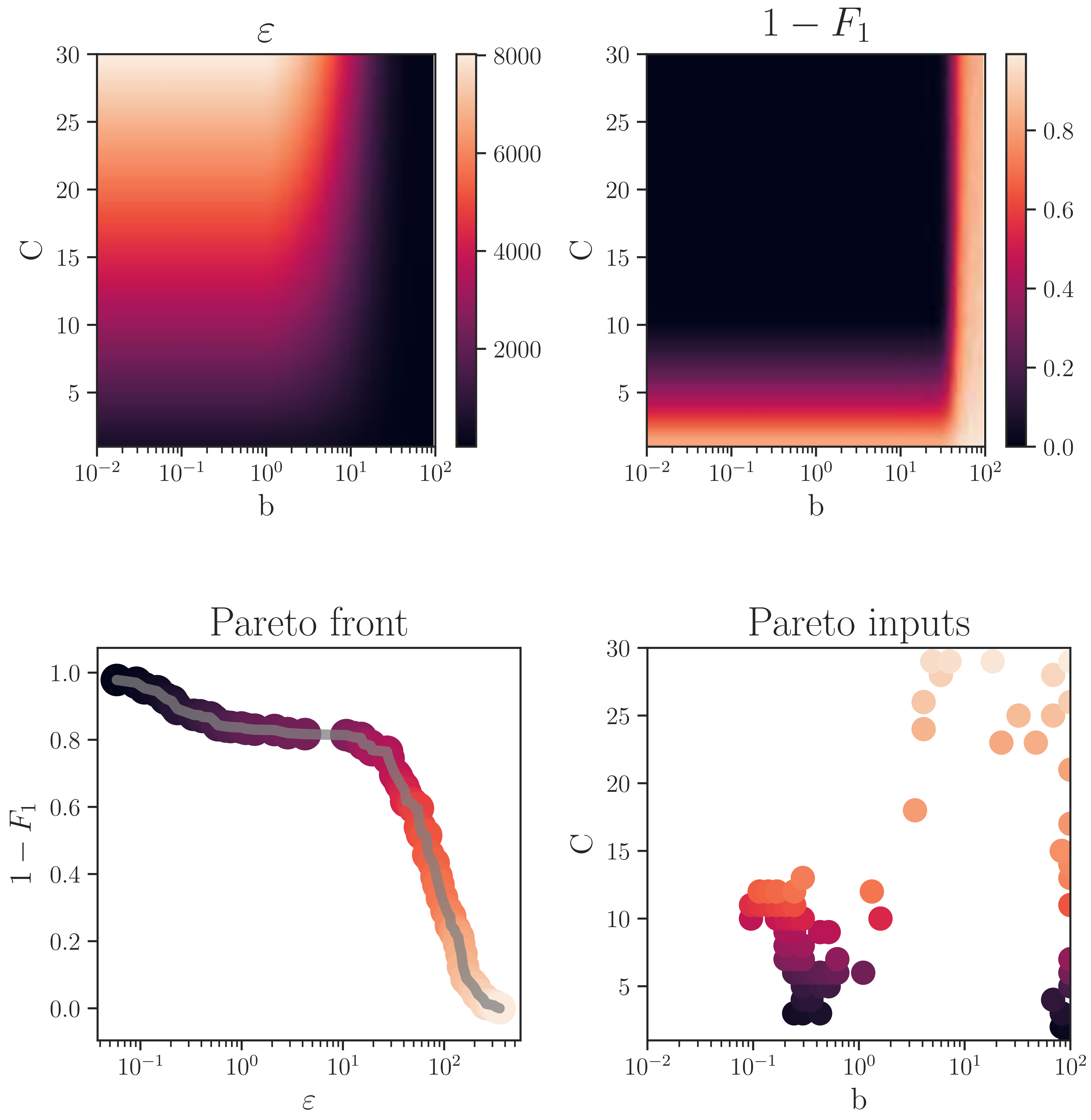
  Evaluate oracles $v \leftarrow (\mathsf{P}(\lambda), \mathsf{E}(\lambda))$

  Augment dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\lambda, v)\}$

return *Pareto front* $\mathcal{PF}(\{v \mid (\lambda, v) \in \mathcal{D}\})$

Example: Sparse Vector Technique

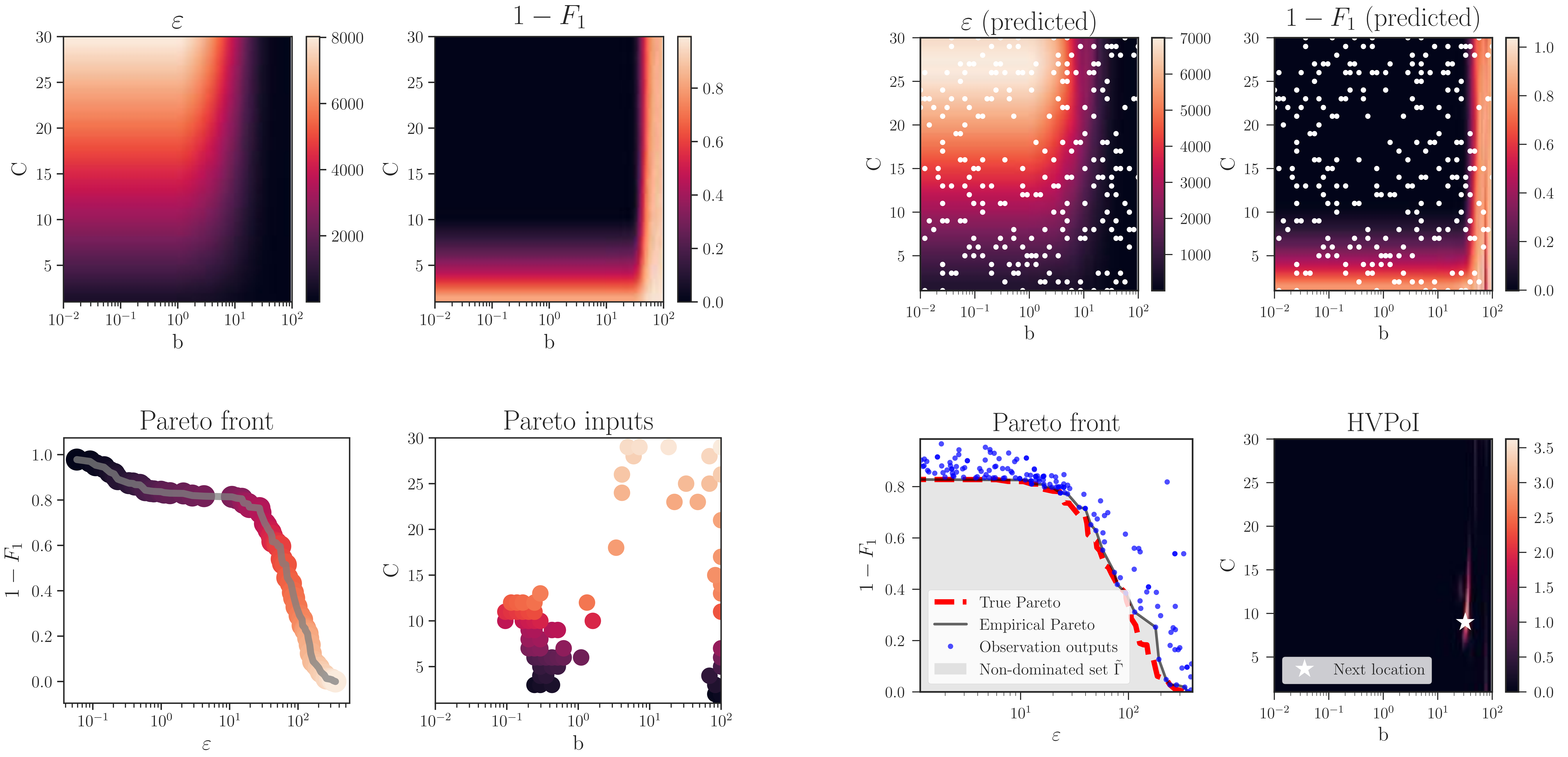
Input: dataset z, queries $q_1, \ldots, q_m$
Hyperparameters: noise b, bound C
$c \leftarrow 0$, $w \leftarrow (0, \ldots, 0) \in \{0,1\}^m$
$b_1 \leftarrow b/(1 + (2C)^{1/3})$, $b_2 \leftarrow b - b_1$, $\rho \leftarrow \mathsf{Lap}(b_1)$
for $i \in [m]$ **do**
    $\nu \leftarrow \mathsf{Lap}(b_2)$
    if $q_i(z) + \nu \geq \frac{1}{2} + \rho$ **then**
        $w_i \leftarrow 1$, $c \leftarrow c + 1$
        if $c \geq C$ **then return** w
return w *[Lyu et al. 2017]*



Setup

- 100 queries with 0/1 output, sensitivity 1
- 10% queries return 1 (randomly selected)
- Privacy: SVT analysis
- Error: 1 - F-score (avg. over 50 runs)

Example: Sparse Vector Technique



Implementing the Oracles

Privacy Oracle

- Epsilon for fixed delta / Others DP variants / Attacks success metrics
- Closed-form expression / Numerical calculation (eg. moments accountant)

Error Oracle

- Fixed input / Distribution over inputs / Worst-case (over a set of) inputs
- On expectation / With high probability
- Exact expression / Empirical evaluation

Implementing the Oracles

Privacy Oracle

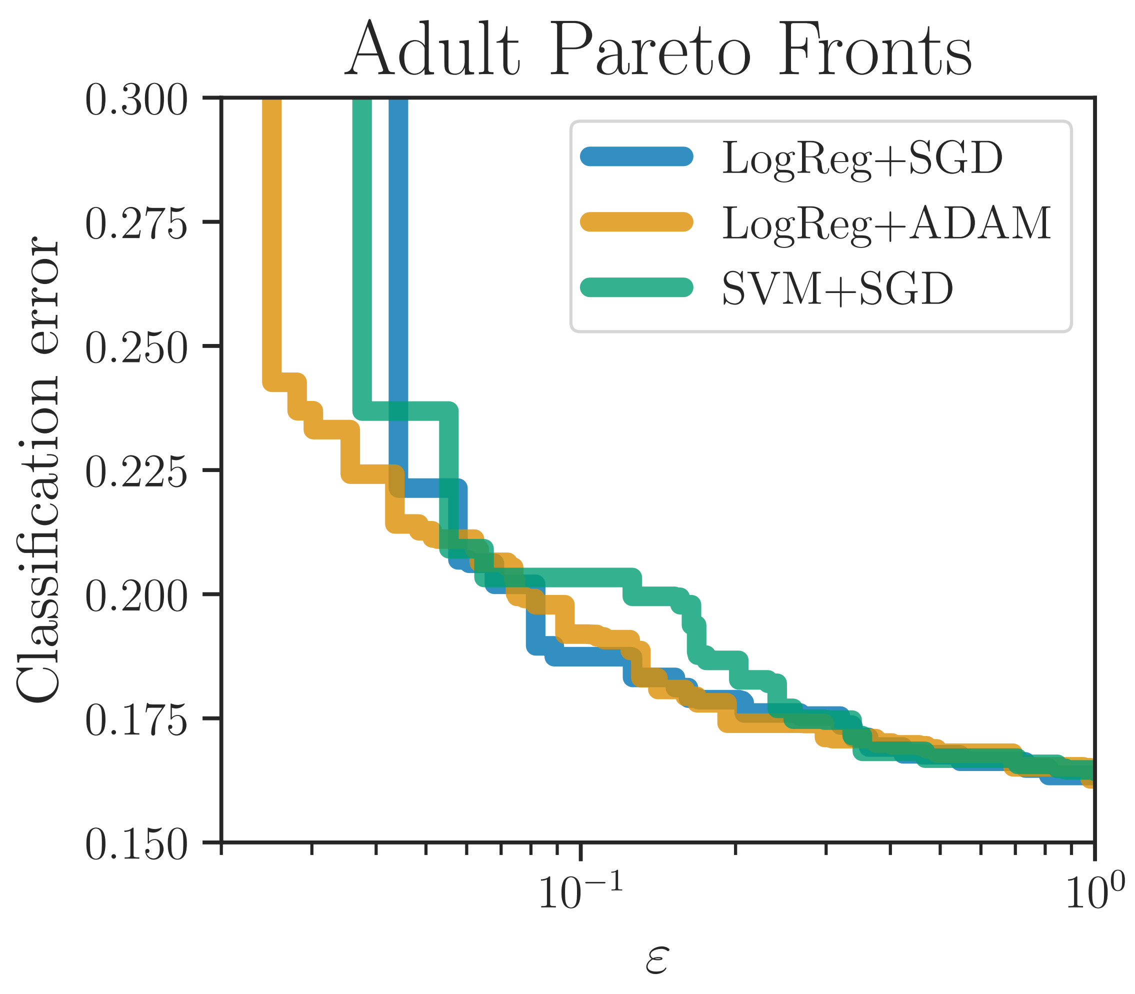
- Epsilon for fixed delta / Others DP variants / Attacks success metrics
- Closed-form expression / Numerical calculation (eg. moments accountant)

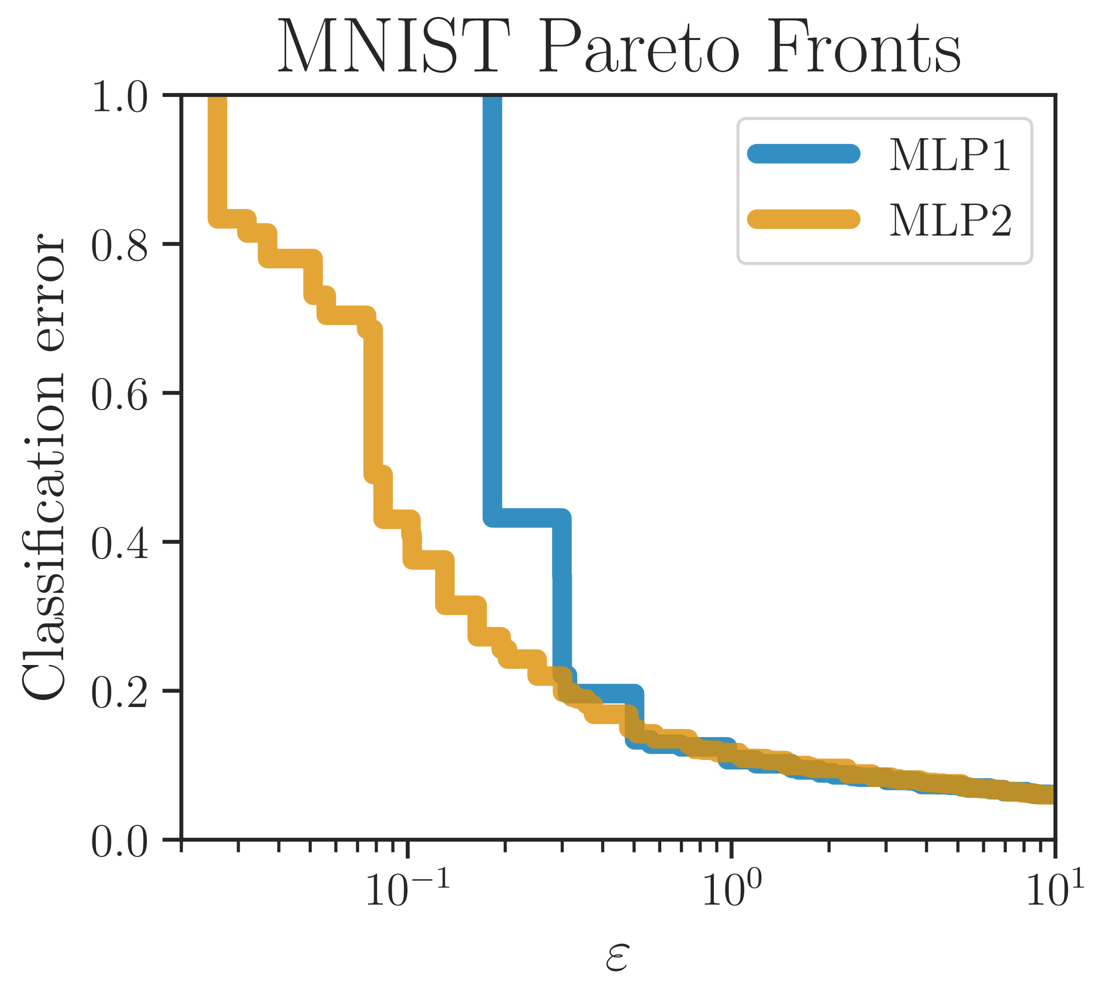
Error Oracle

- Fixed input / Distribution over inputs / Worst-case (over a set of) inputs
- On expectation / With high probability
- Exact expression / Empirical evaluation

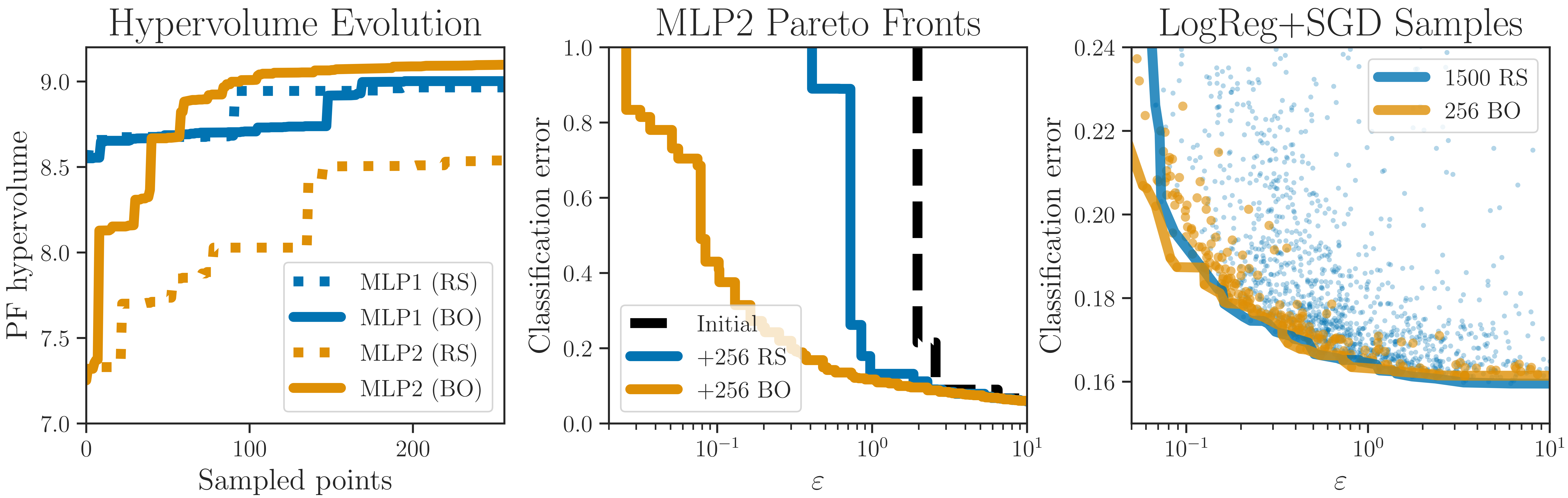
Machine Learning Experiments

- Adult dataset (n=40K, d=123)
  - Logistic regression (SGD and ADAM)
  - Linear SVM (SGD)
- MNIST dataset (n=60K, d=784)
  - MLP1 (1000 hidden)
  - MLP2 (128-64 hidden)





DPareto *vs* Random Sampling



Conclusion

- Empirical privacy-utility trade-off evaluation enables application-specific decisions

- Bayesian optimization provides computationally efficient method to recover the Pareto front (esp. with large number of hyper-parameters)

Future work:

- Address leakage in Pareto front (when error oracle is input-specific)

- Include further criteria (eg. running time of parametrized algorithm)