AI 신약개발, 어떤 후보물질을 선택할까?
Multi-objective decision making under uncertainty

박지원

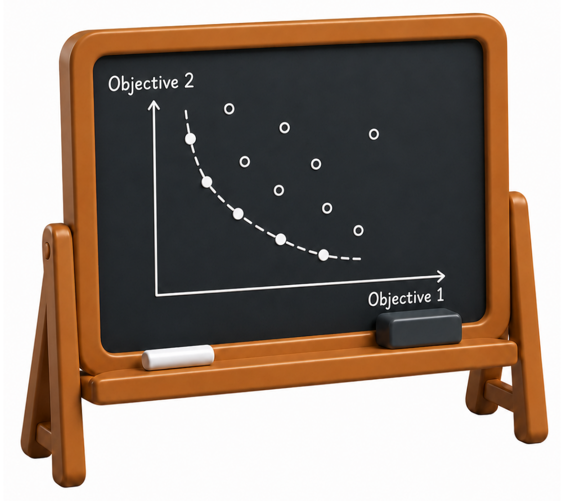
Principal Scientist & Group Lead

Prescient Design, Genentech

2026/07/07

첨단융합학부 SNUTI On Site Workshop

Lecture



I. Multi-objective optimization: the basics

- Motivation: lead optimization in drug discovery
- Pareto front
- Quality indicator

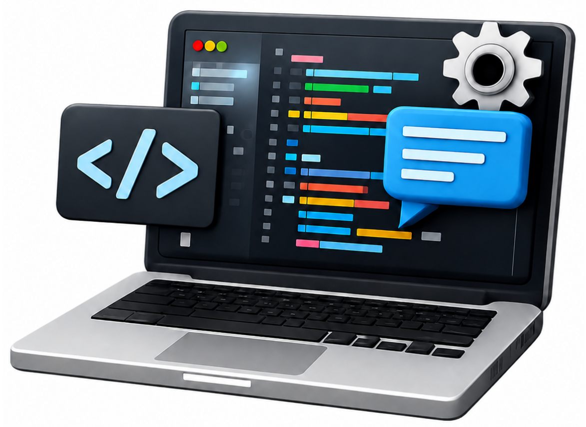
II. Bayesian optimization

- Surrogate modeling
- Utility function → acquisition function

III. Multi-objective reasoning

- Common multi-objective acquisition strategies
- Batched/noisy variants

Practicum



I. Multi-objective optimization: the basics

- Motivation: lead optimization in drug discovery
- Pareto front
- Quality indicator

Notebook 0: evaluating HV

II. Bayesian optimization

- Surrogate modeling
- Utility function → acquisition function

Notebook 1: full MOBO loop

III. Multi-objective reasoning

- Common multi-objective acquisition strategies
- Batched/noisy variants

Notebooks 2, 3: team competition on simulated antibody optimization

Practicum goals

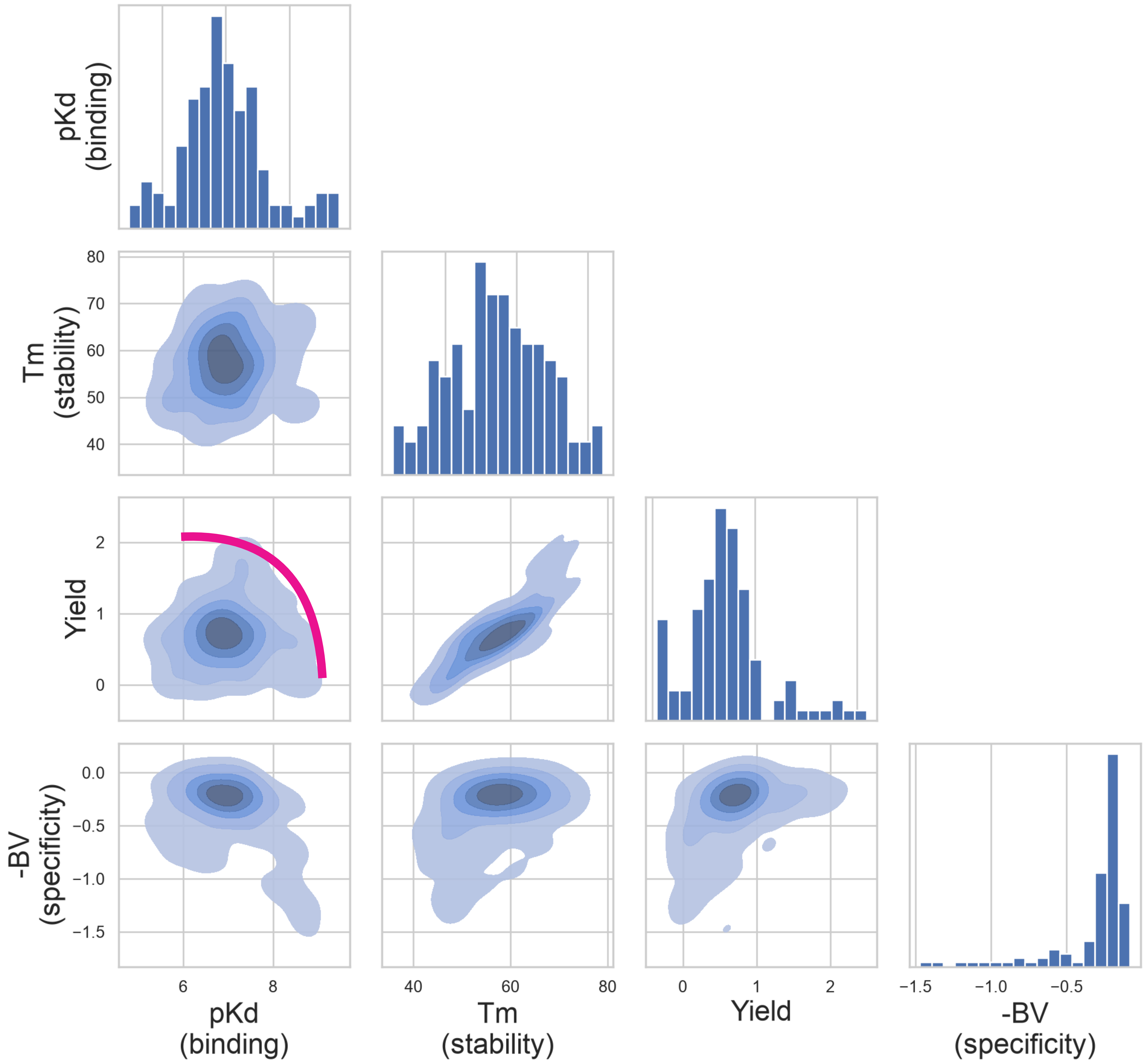
- Learn the **BoTorch** syntax for concepts covered in the lecture
- Be able to **verify** the implementation of specific configurations for your own application
- Gain **intuition** for the pros/cons of each acquisition strategy by visualizing the acquired designs and metrics

Use of AI coding agents is strongly encouraged for visualization of data and results.

Ask questions throughout the lecture, as each concept builds on the next and all the concepts will reappear in the lab!

Motivation: multiple real-world objectives

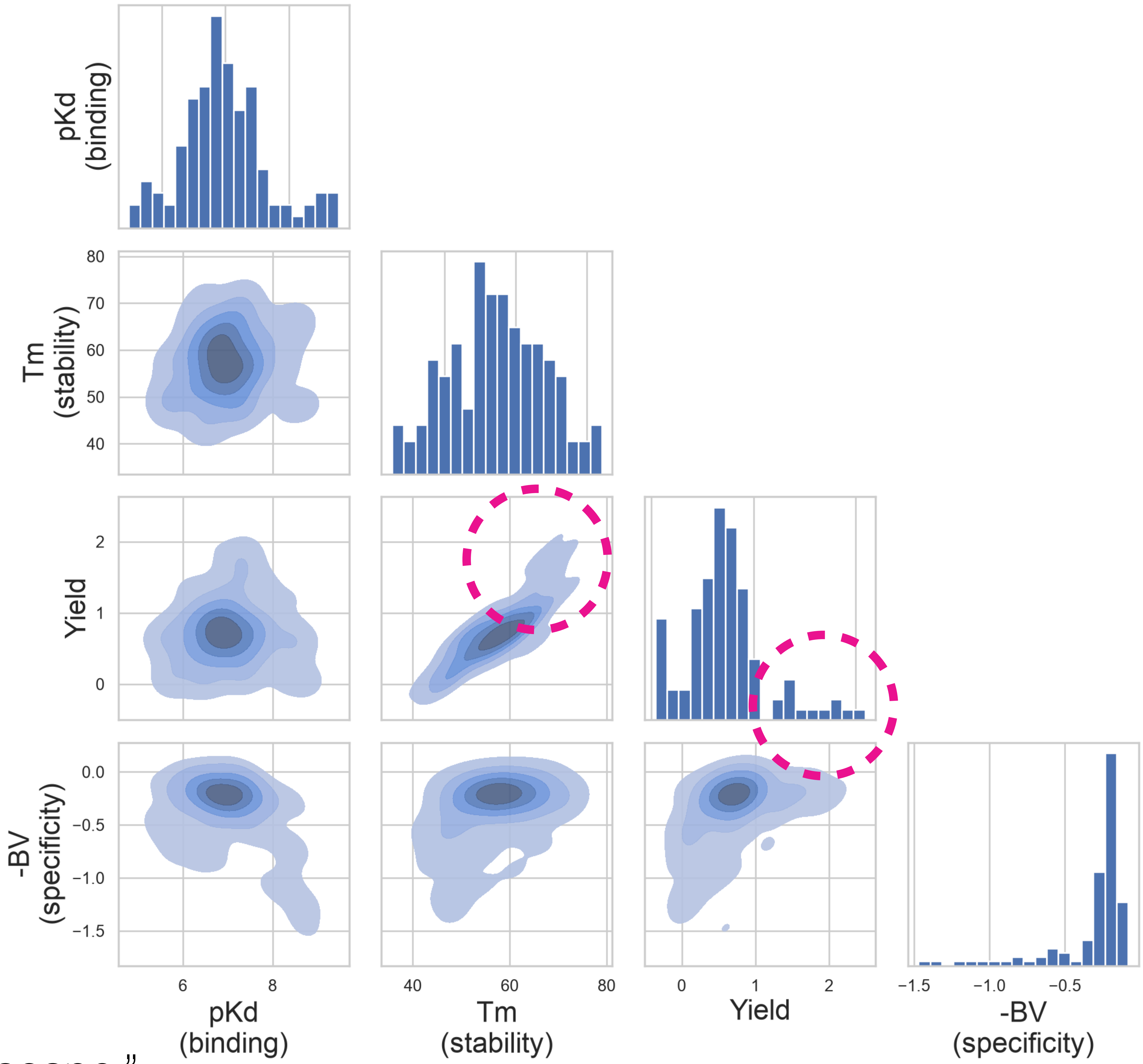
- In practical design problems, we often want to maximize/minimize multiple properties jointly.
  - Antibody design: binding strength, yield, thermostability, specificity
- Trade-offs are common



[1] Jain et al., "Biophysical properties of the clinical-stage antibody landscape." Proc Natl Acad Sci. (2017).

Motivation: multiple real-world objectives

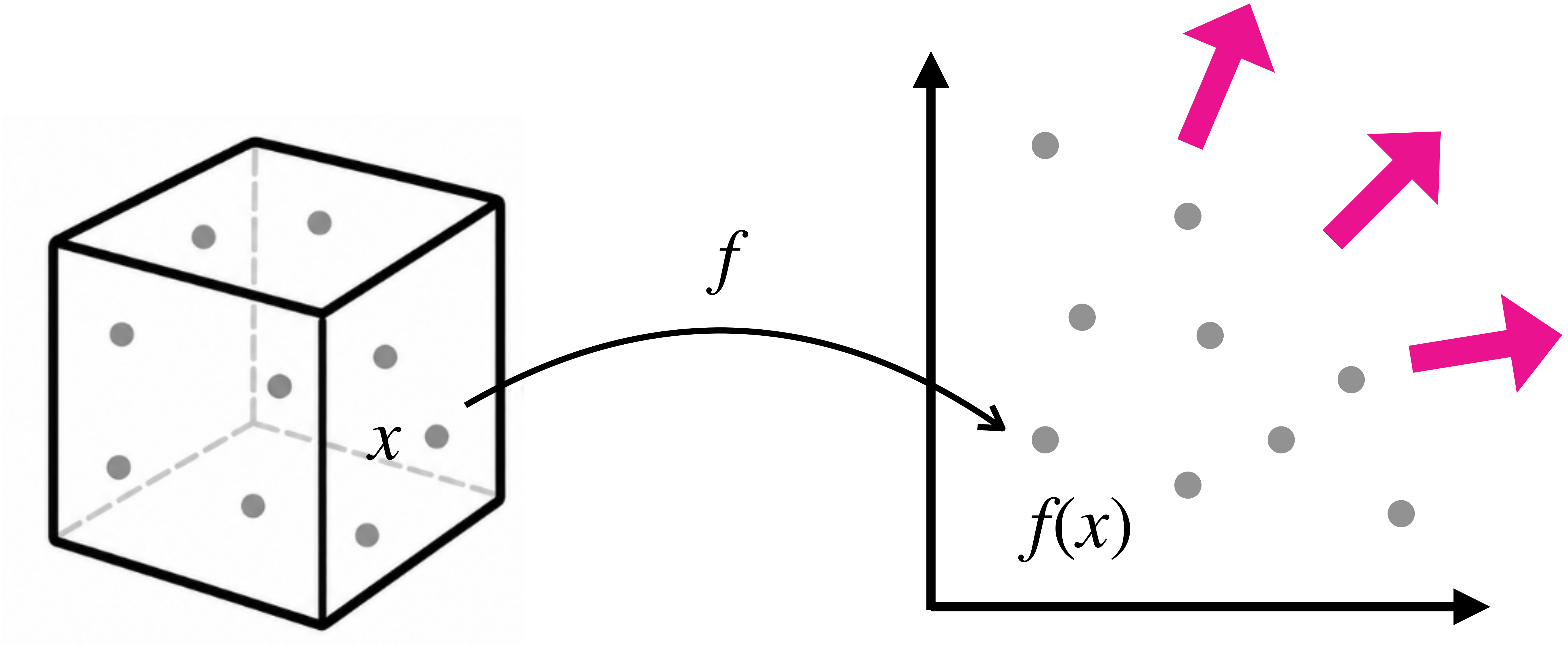
- In practical design problems, we often want to maximize/minimize multiple properties jointly.
  - Antibody design: binding strength, yield, thermostability, specificity
- Trade-offs are common
- Molecular properties tend to have complex correlation structures.
  - long tails and tail correlations



Jain et al., "Biophysical properties of the clinical-stage antibody landscape." Proc Natl Acad Sci. (2017).

Multi-objective optimization

Maximize a vector-valued **objective function** $f: \mathscr{X} \to \mathbb{R}^M$ over $\mathscr{X}$



Design space $\mathscr{X}$

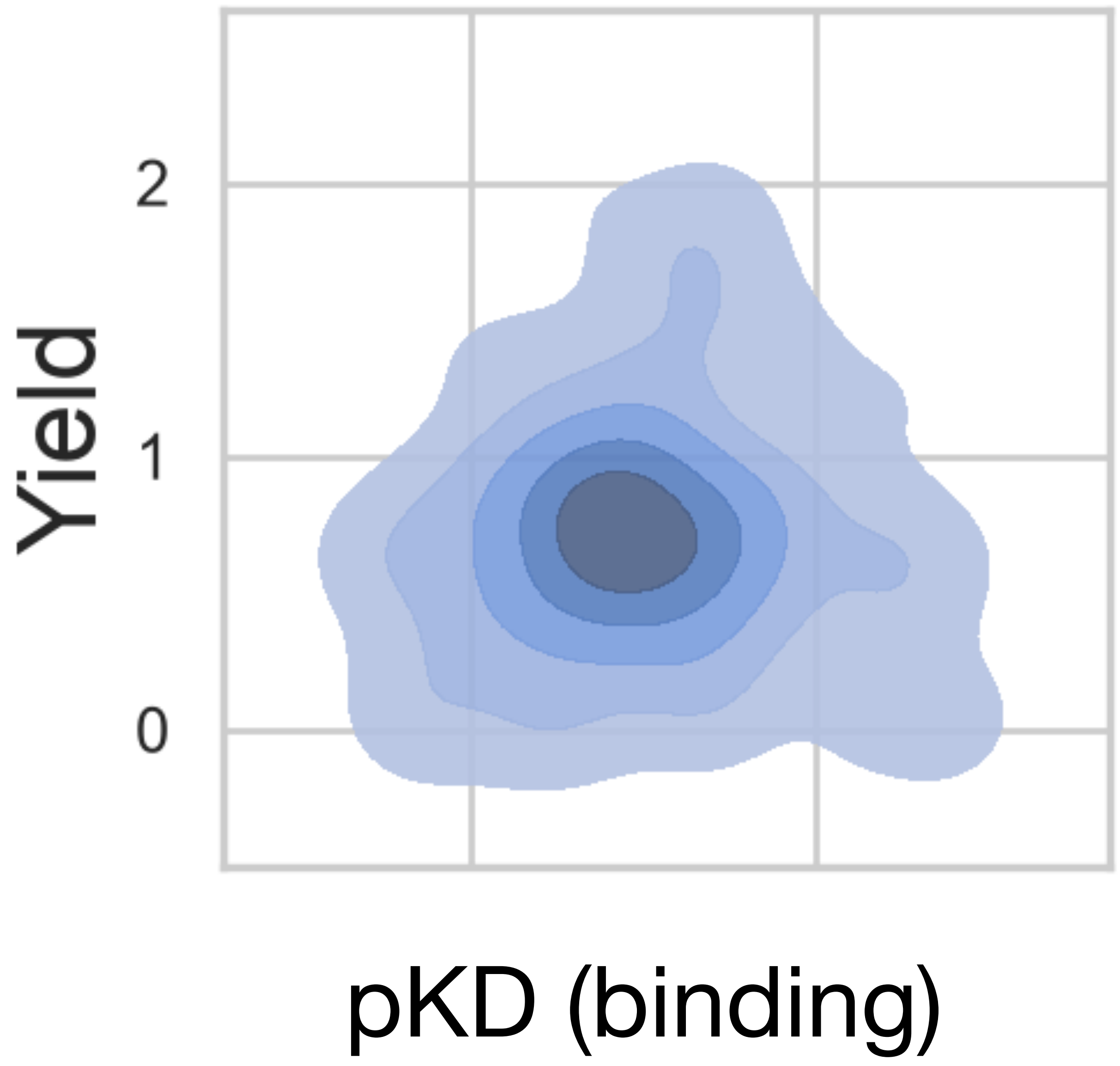
(e.g., molecules)

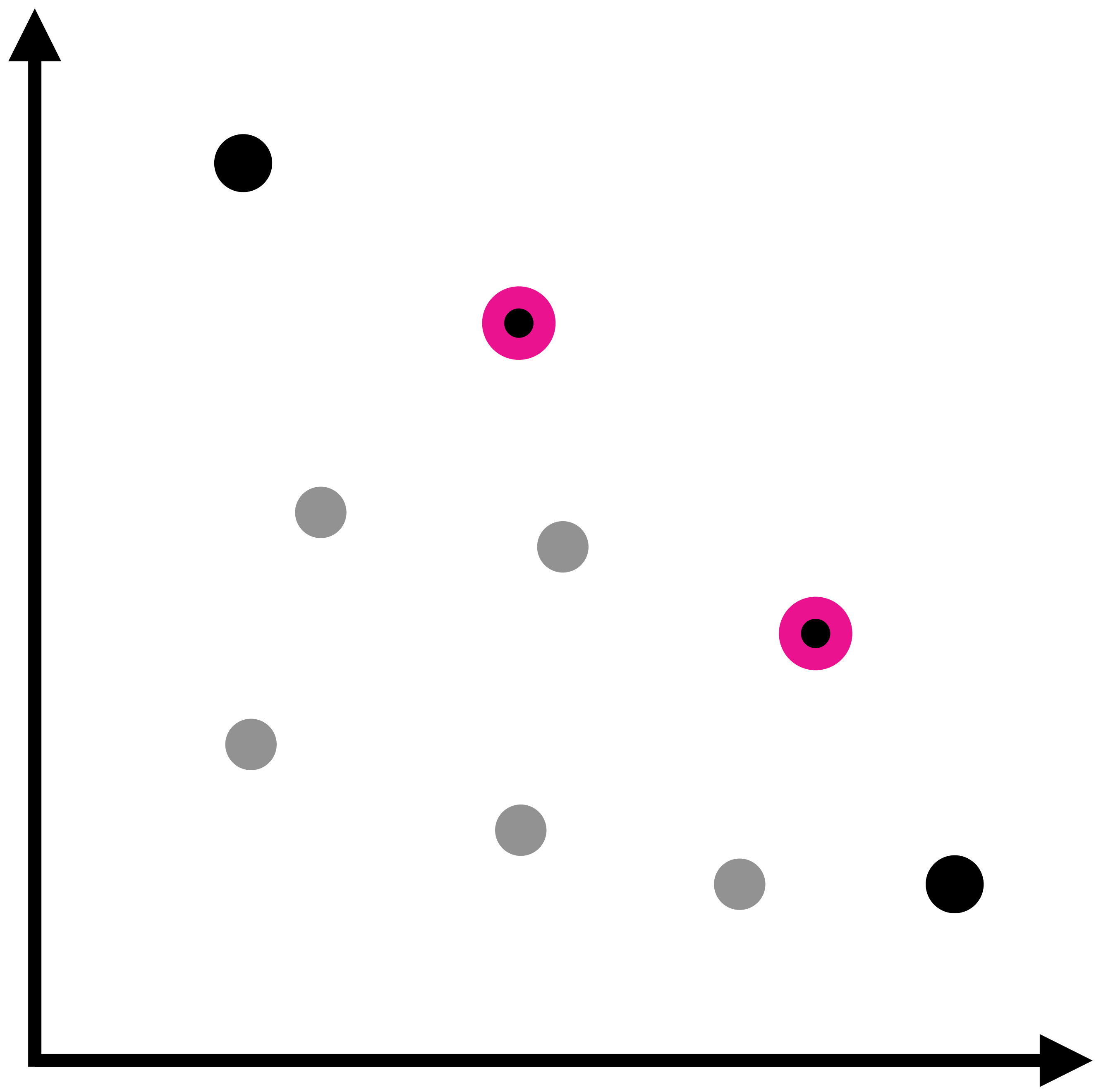
Objective space $\mathbb{R}^M$

(e.g., M molecular properties)

Pareto optimality

For $M > 1$ objectives, a single optimal design may not exist.





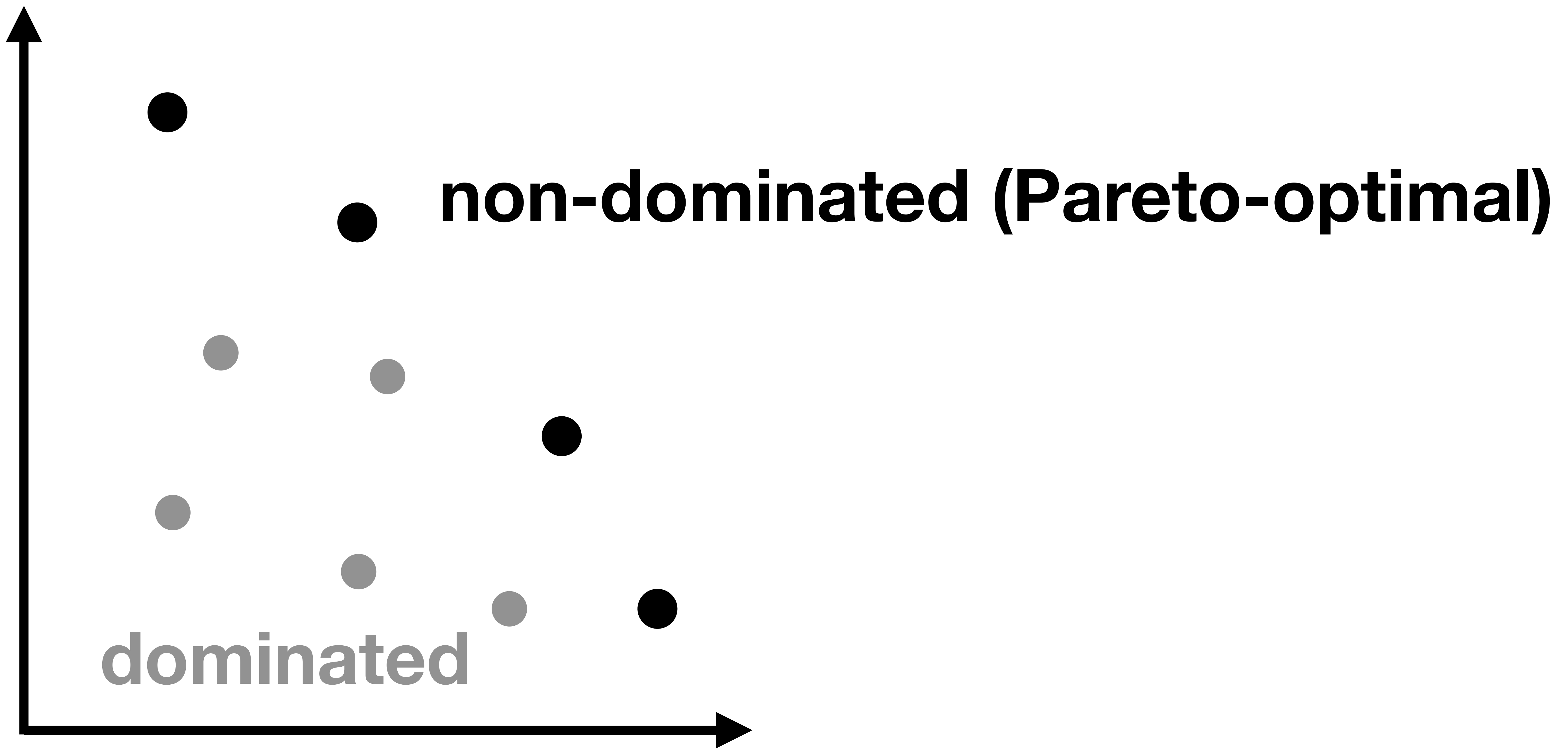
Pareto optimality

For $M > 1$ objectives, a single optimal design may not exist.

A design x' dominates x if x' is at least as good in every objective and better in at least one:

$$f_i(x') \geq f_i(x) \ \forall i, \quad f_j(x') > f_j(x) \text{ for some } j$$

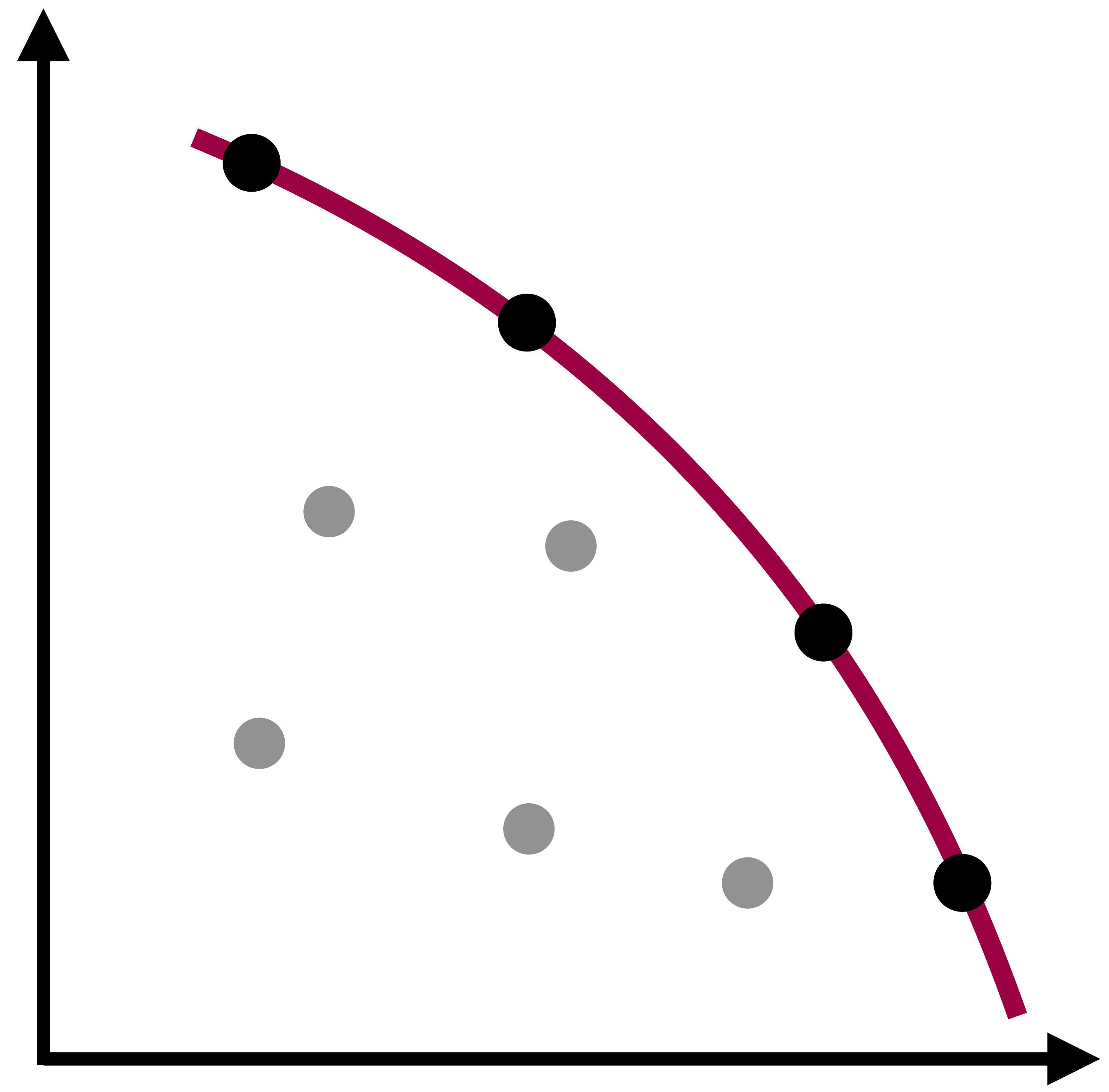
A design is Pareto-optimal if it's not dominated by any other design.



Pareto front

- The set of Pareto-optimal designs is called the Pareto set $\mathscr{X}^*$.
- The Pareto front $\mathscr{P}^* \equiv f(\mathscr{X}^*) \subset \mathbb{R}^M$ is its image in the objective space.

Goal of MOO: find designs $\hat{\mathscr{X}}$ whose objective values $\hat{\mathscr{P}}$ approximate the **true Pareto front** $\mathscr{P}^*$



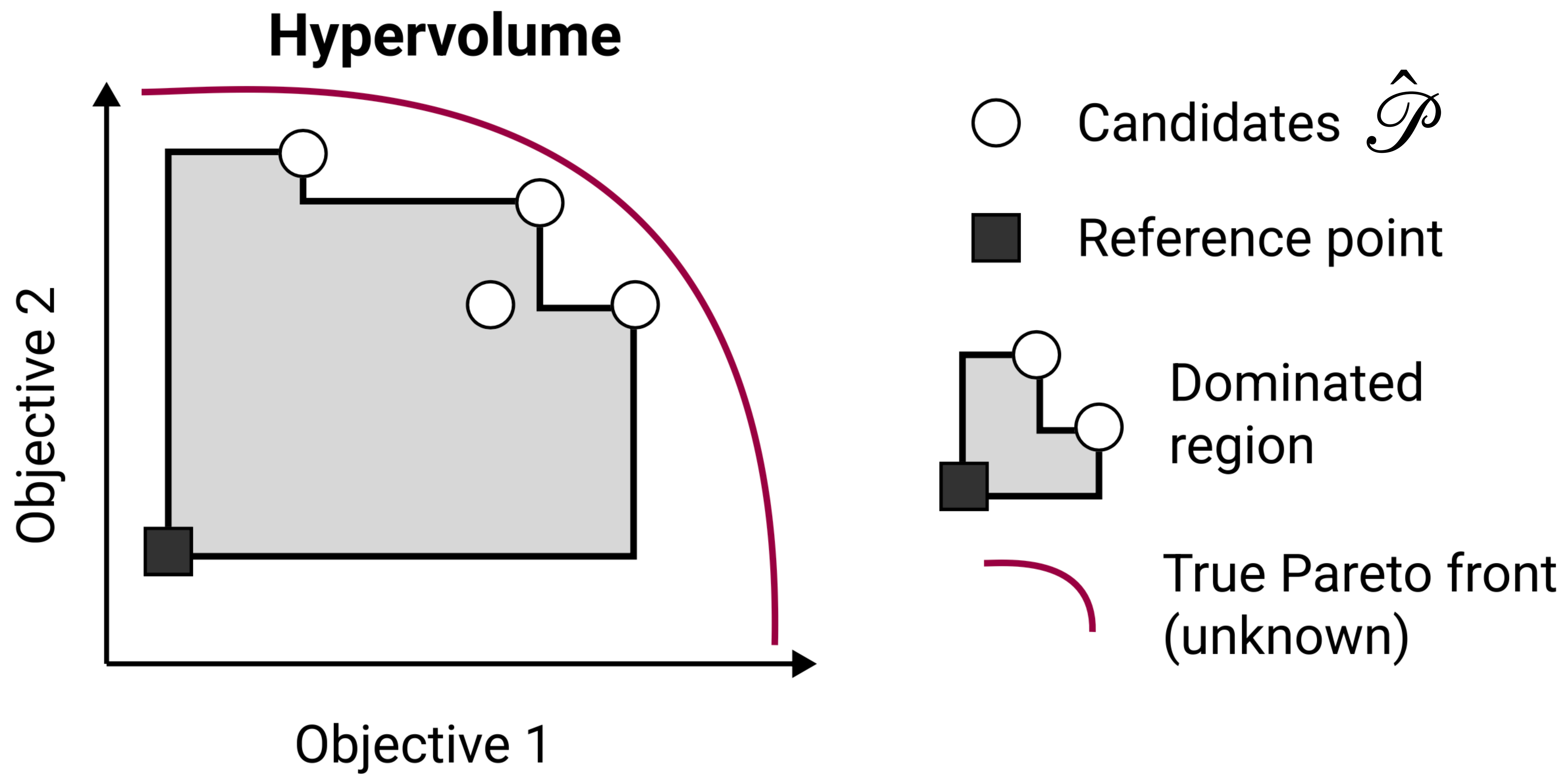
Quality indicators

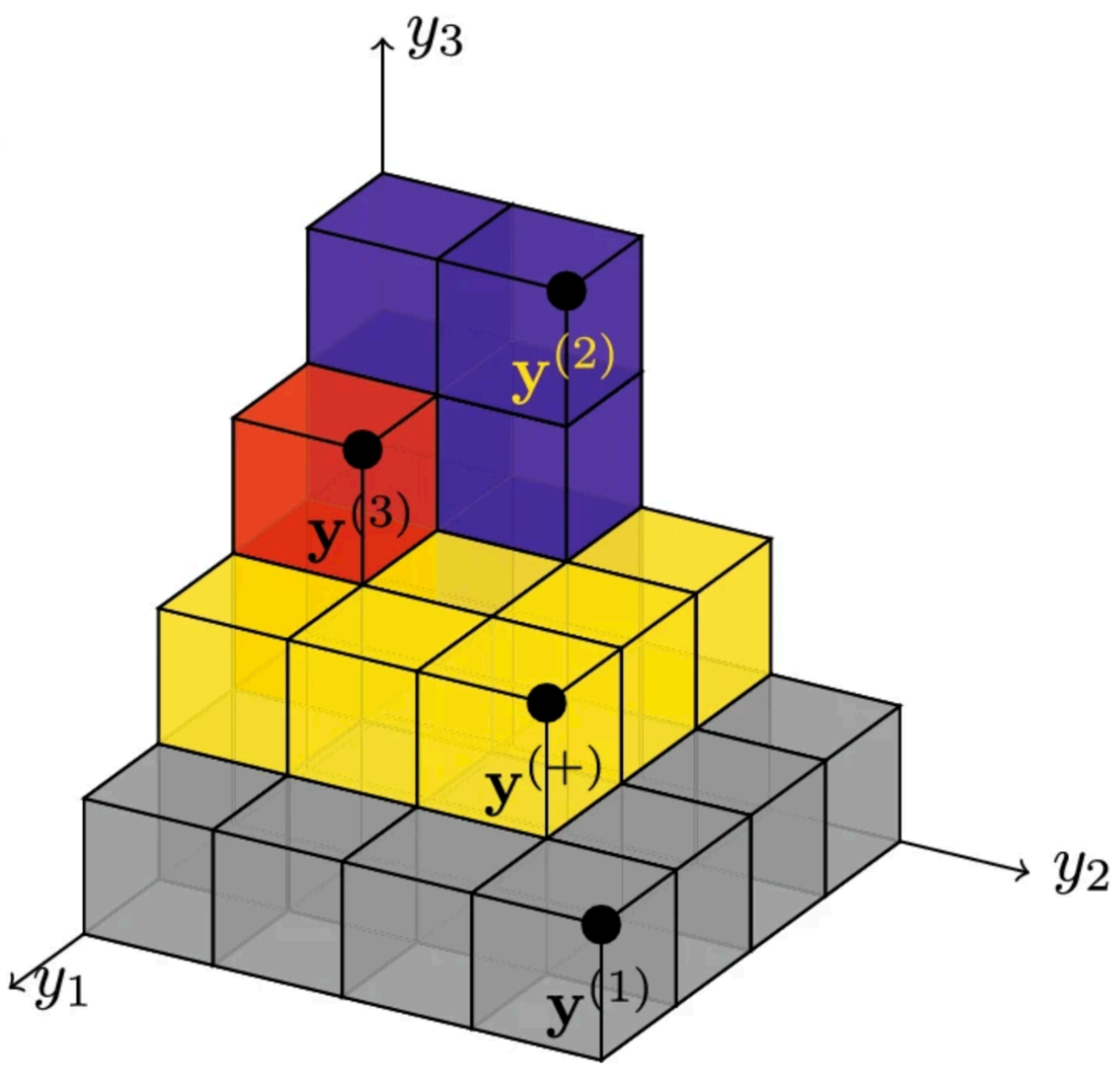
Quality indicator $I : 2^{\mathscr{Y}} \to \mathbb{R}$
evaluates the quality of the approximate Pareto front $\hat{\mathscr{P}}$

Can be used to compare MOO algorithms

Example of QI: hypervolume (HV) indicator

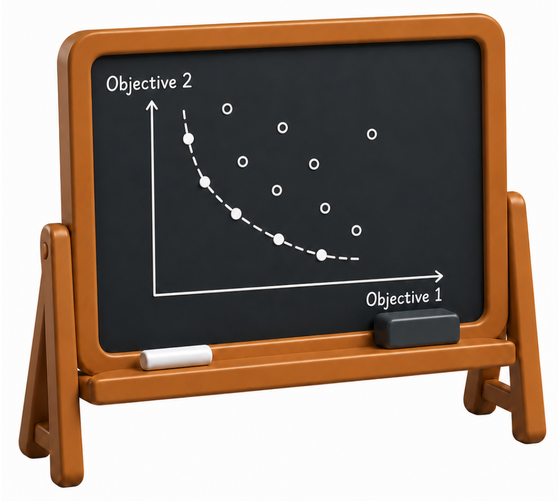
Hypervolume of the polytope[5] dominated by $\hat{\mathscr{P}}$ and bounded from below by a reference point





[5] Emmerich, Deutz, and Klinkenberg, "Hypervolume-based expected improvement." IEEE CEC (2011).

Lecture



I. Multi-objective optimization: the basics

- Motivation: lead optimization in drug discovery
- Pareto front
- Quality indicator

II. Bayesian optimization

- Surrogate modeling
- Utility function → acquisition function

III. Multi-objective reasoning

- Common multi-objective acquisition strategies
- Batched/noisy variants

First, a crash course on Bayesian statistics

Given a labeled dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, we want to predict y for a new input x.

$$y^{(i)} = f(x^{(i)}) + \text{noise}$$

- Conditional probability: after receiving D and a test input x, we treat them as fixed and ask what is still uncertain about y $\rightarrow p(y \mid x, D)$

First, a crash course on Bayesian statistics

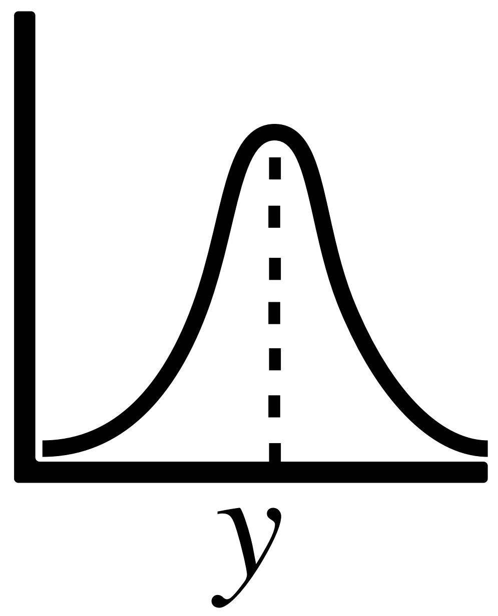
Given a labeled dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, we want to predict y for a new input x.

$$y^{(i)} = f(x^{(i)}) + \text{noise}$$

- Conditional probability: after receiving D and a test input x, we treat them as fixed and ask what is still uncertain about y $\rightarrow p(y \mid x, D)$
- Predictive distribution $p(y \mid x, D)$

  Sometimes I write this as the predictive function being random.

  I can sample a function $\hat{f}^{(s)} \sim p(f \mid D)$ so each $\hat{f}^{(s)}(x)$ is one plausible prediction.



First, a crash course on Bayesian statistics

Given a labeled dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, we want to predict y for a new input x.

$$y^{(i)} = f(x^{(i)}) + \text{noise}$$

- Conditional probability: after receiving D and a test input x, we treat them as fixed and ask what is still uncertain about y $\rightarrow p(y \mid x, D)$
- Predictive distribution $p(y \mid x, D)$

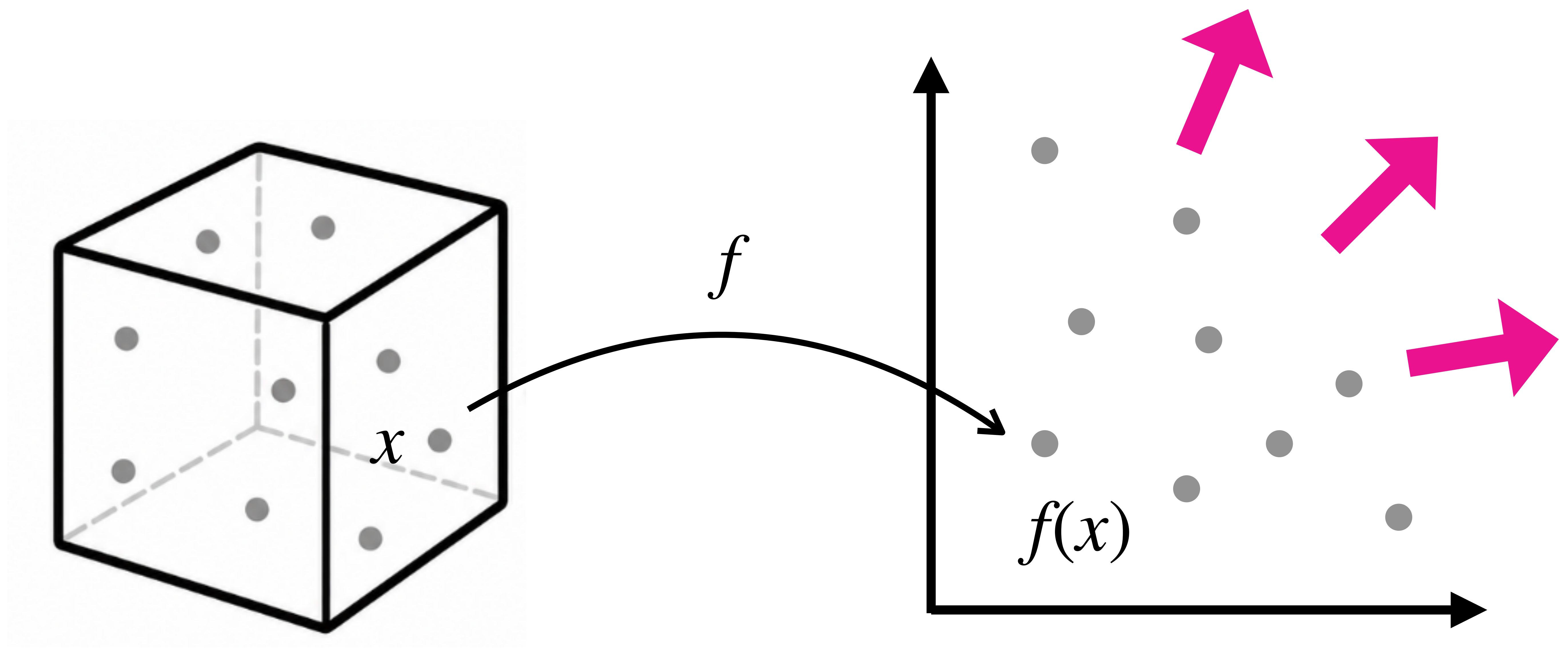
  Sometimes I write this as the predictive function being random.

  I can sample a function $\hat{f}^{(s)} \sim p(f \mid D)$ so each $\hat{f}^{(s)}(x)$ is one plausible prediction.
- Expectation means averaging over possible predictive functions. You can draw S samples $\hat{f}^{(1)}, \ldots, \hat{f}^{(S)}$ and estimate the expectation as:

$$\mathbb{E}_{\hat{f}}\left[g(\hat{f}(x))\right] \approx \frac{1}{S}\sum_{s=1}^{S} g\left(\hat{f}^{(s)}(x)\right)$$

Multi-objective Bayesian optimization

Maximize a vector-valued objective function $f : \mathscr{X} \to \mathbb{R}^M$ over $\mathscr{X}$



Design space $\mathscr{X}$

Objective space $\mathbb{R}^M$

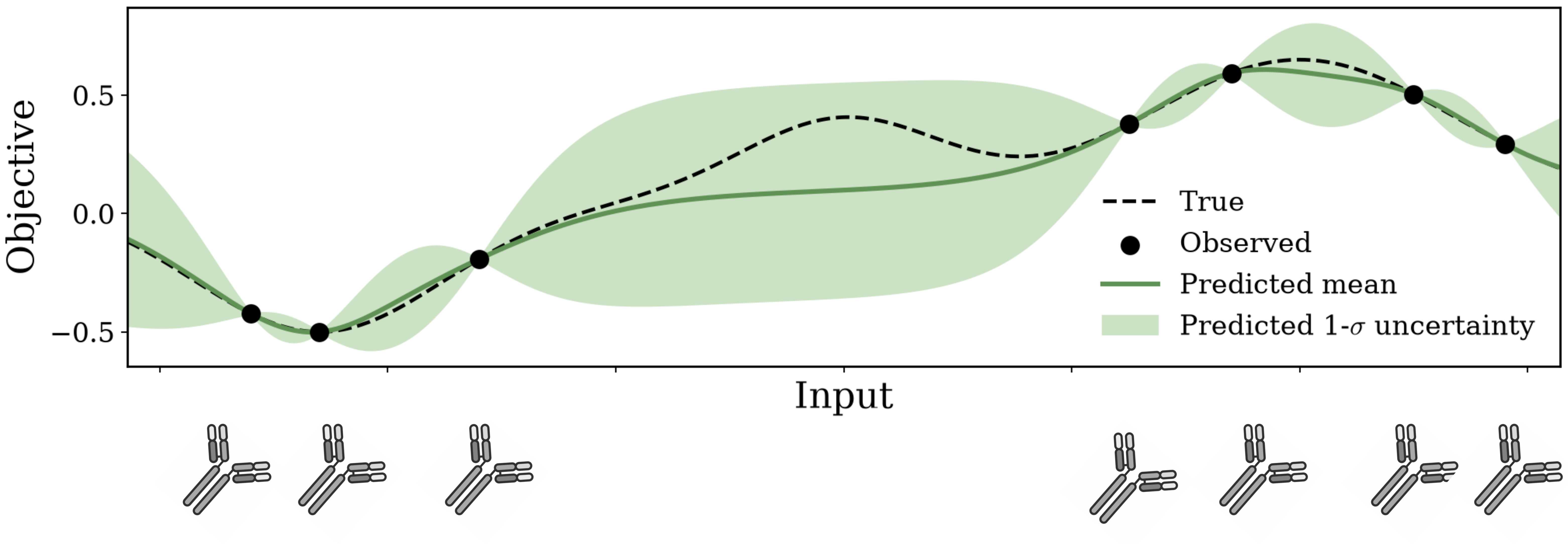
When f is an expensive black-box function (e.g., wet lab),
Bayesian optimization offers a sample-efficient method.

(Single-objective) Bayesian optimization

Specify a probabilistic surrogate model $\hat{f}$ approximating f, often a Gaussian process

Spread of $p(\hat{f} \mid D)$ captures the model uncertainty

Binding strength



Acquisition function = decision-making engine

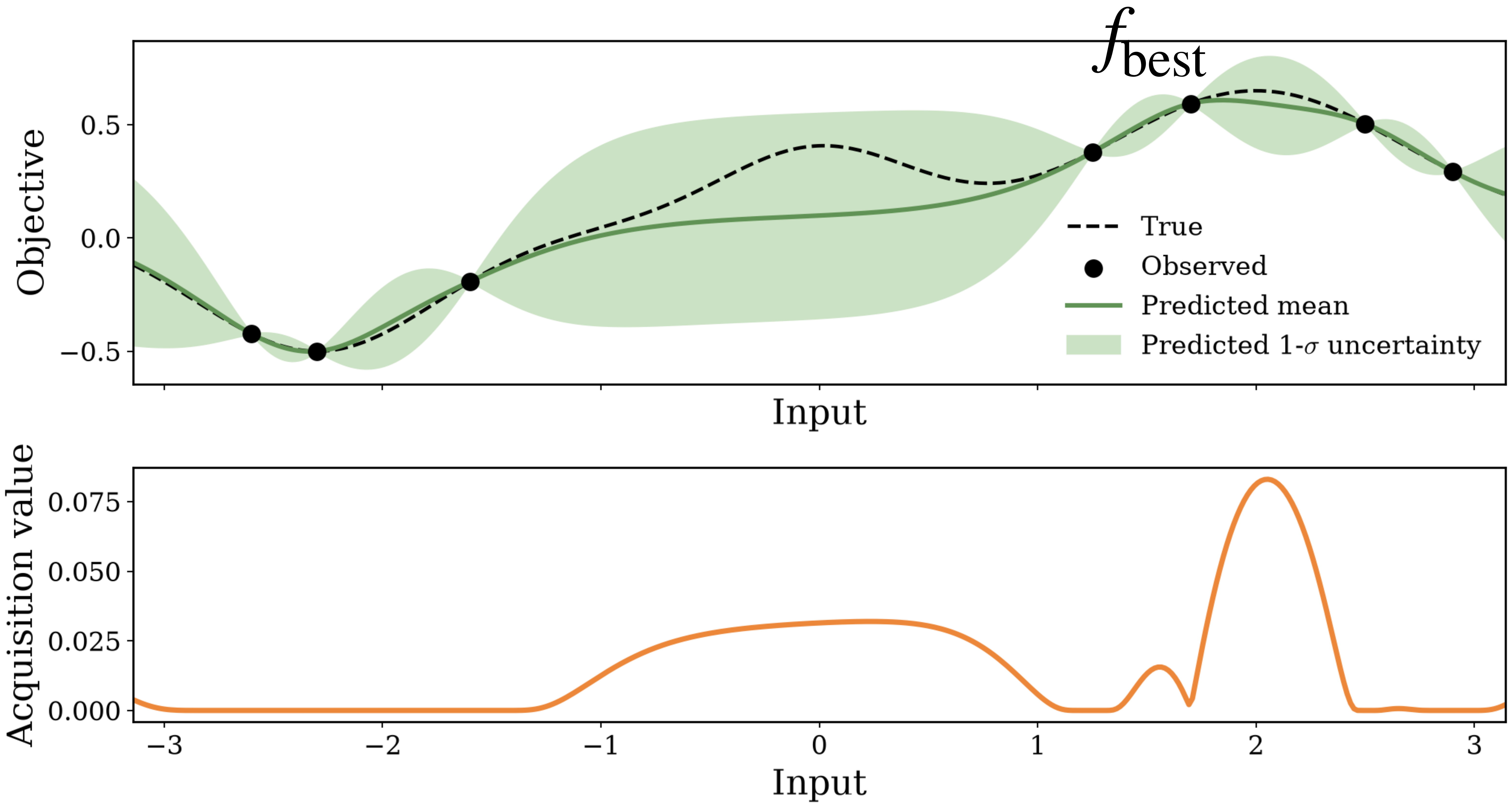
Acquisition function $a^{\hat{f}} : \mathscr{X} \to \mathbb{R}$ scores each design with predicted usefulness, to determine which design to evaluate next.

- **exploration** (of highly uncertain designs)
- **exploitation** (of designs believed to be optimal)

Binding strength

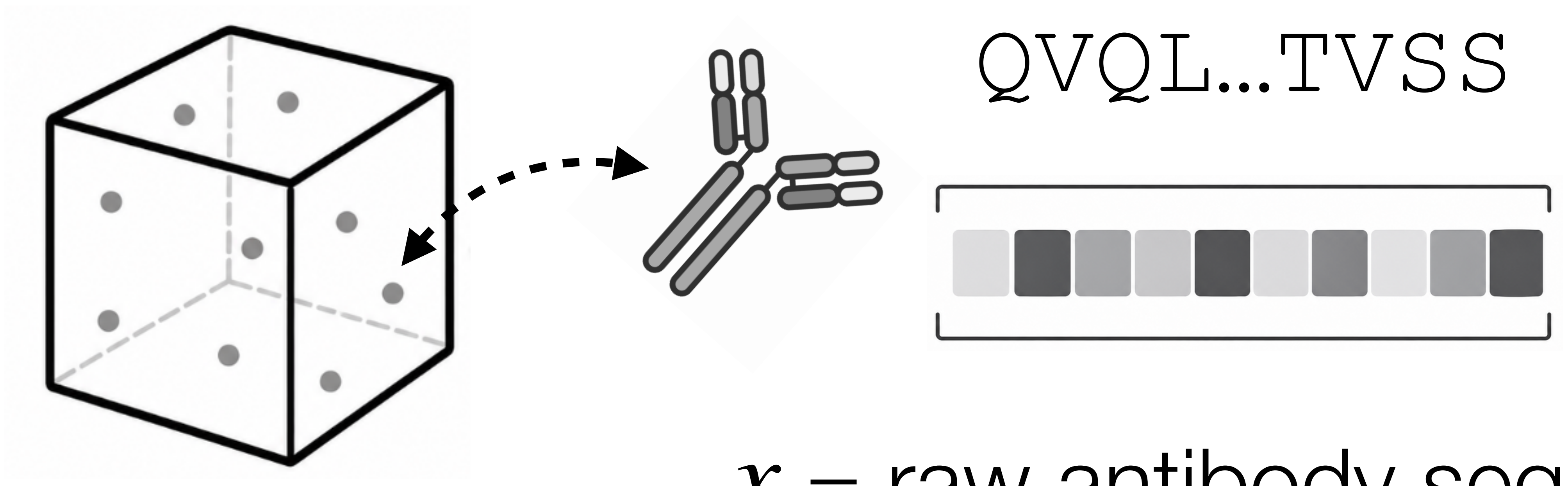
Expected improvement

$$a^{\hat{f}}(x) = \mathbb{E}_{\hat{f}}\left[\hat{f}(x) - f_{\text{best}}\right]_+$$



BayesOpt algorithm

Propose design candidates in $\mathcal{X}$

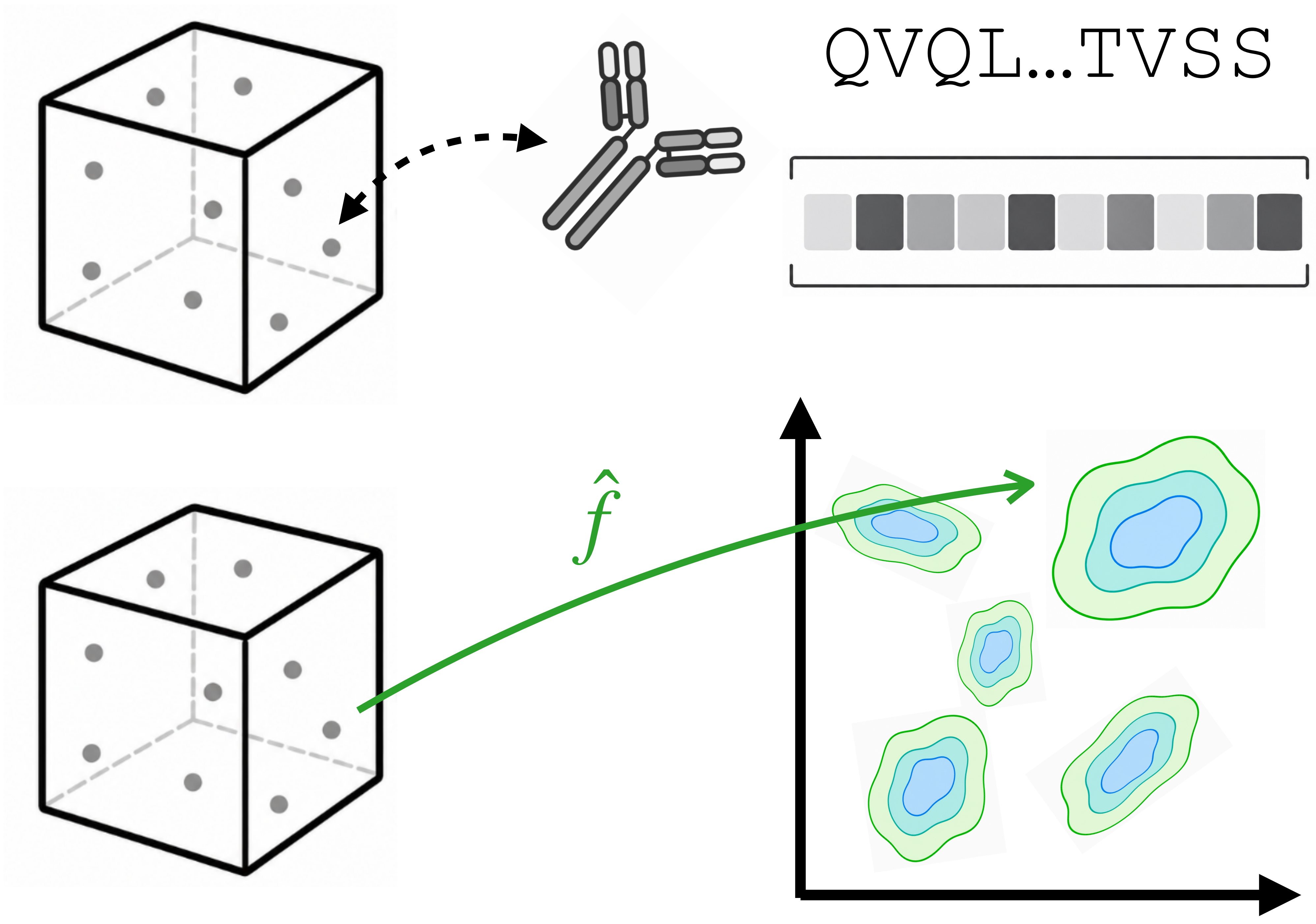


x = raw antibody sequence or low-dimensional representation from a pretrained model (e.g., ESM, AntiBERTy)

BayesOpt algorithm

Propose design candidates in $\mathscr{X}$

Predict properties with surrogate model $\hat{f}$



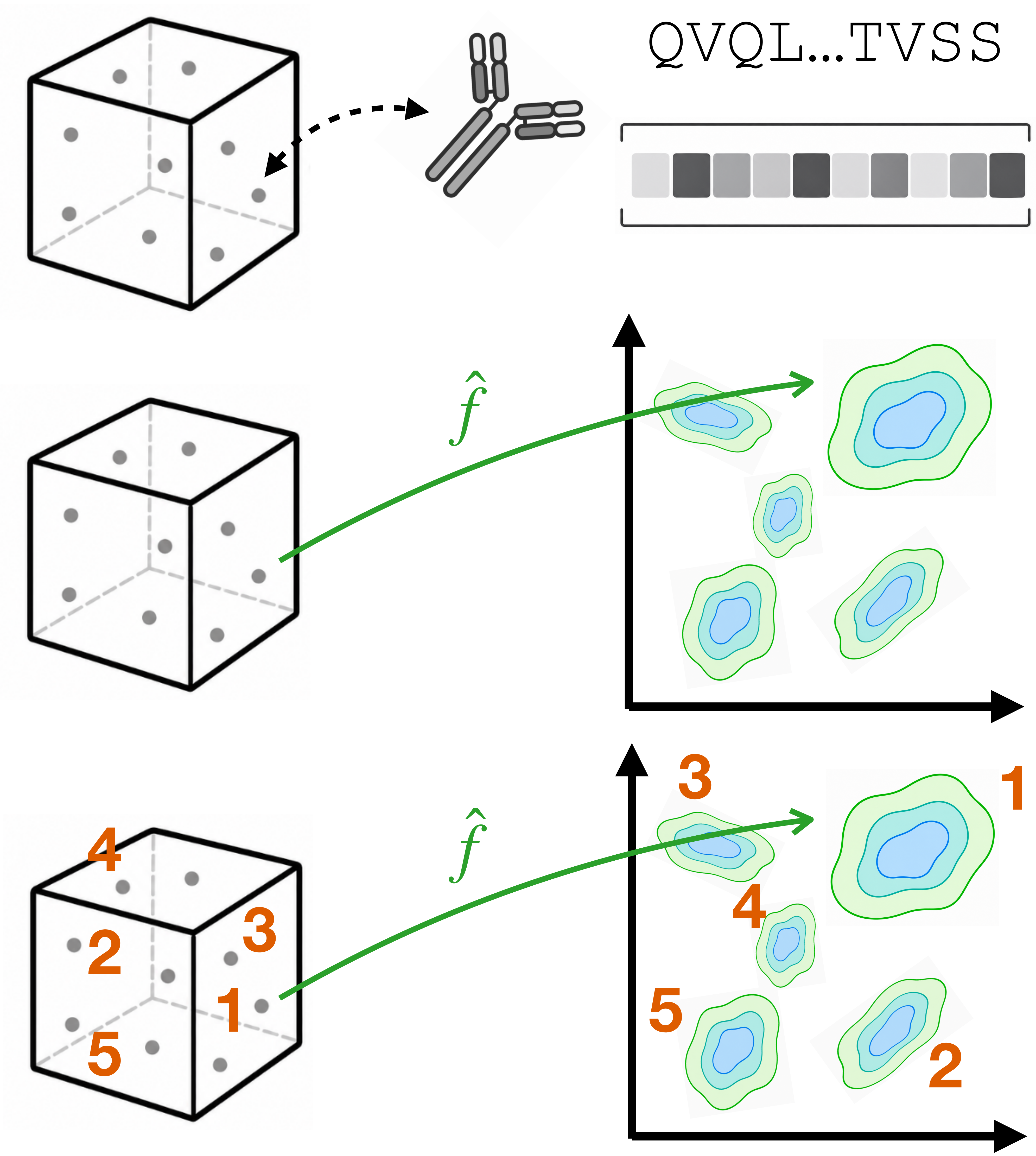
showing $M = 2$

Multi-objective BayesOpt algorithm

Propose design candidates in $\mathscr{X}$

Predict properties with surrogate model $\hat{f}$

Rank candidates with acquisition function

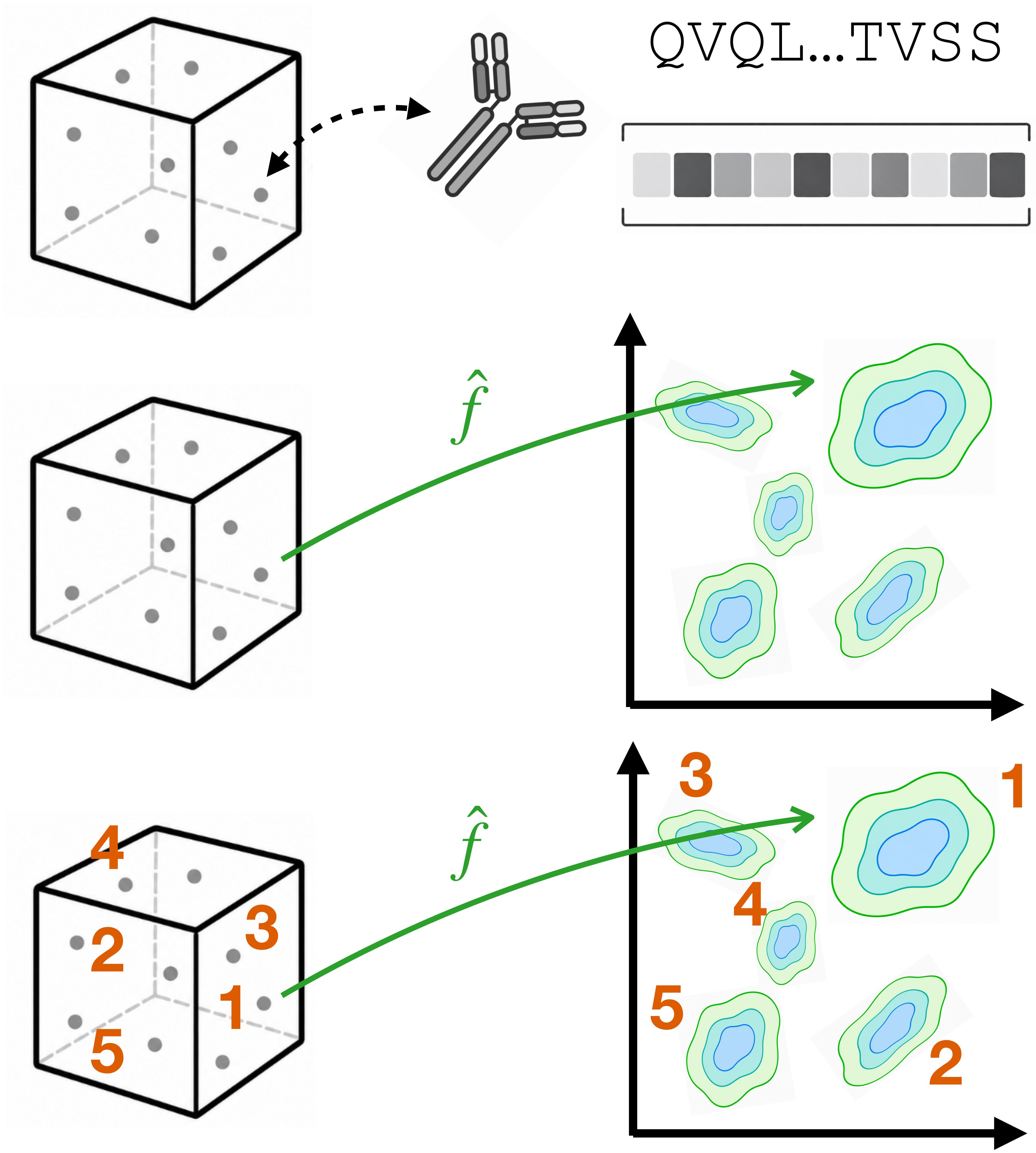


BayesOpt algorithm

Propose design candidates in $\mathscr{X}$

Predict properties with surrogate model $\hat{f}$

Rank candidates with acquisition function



Choose the candidate to evaluate

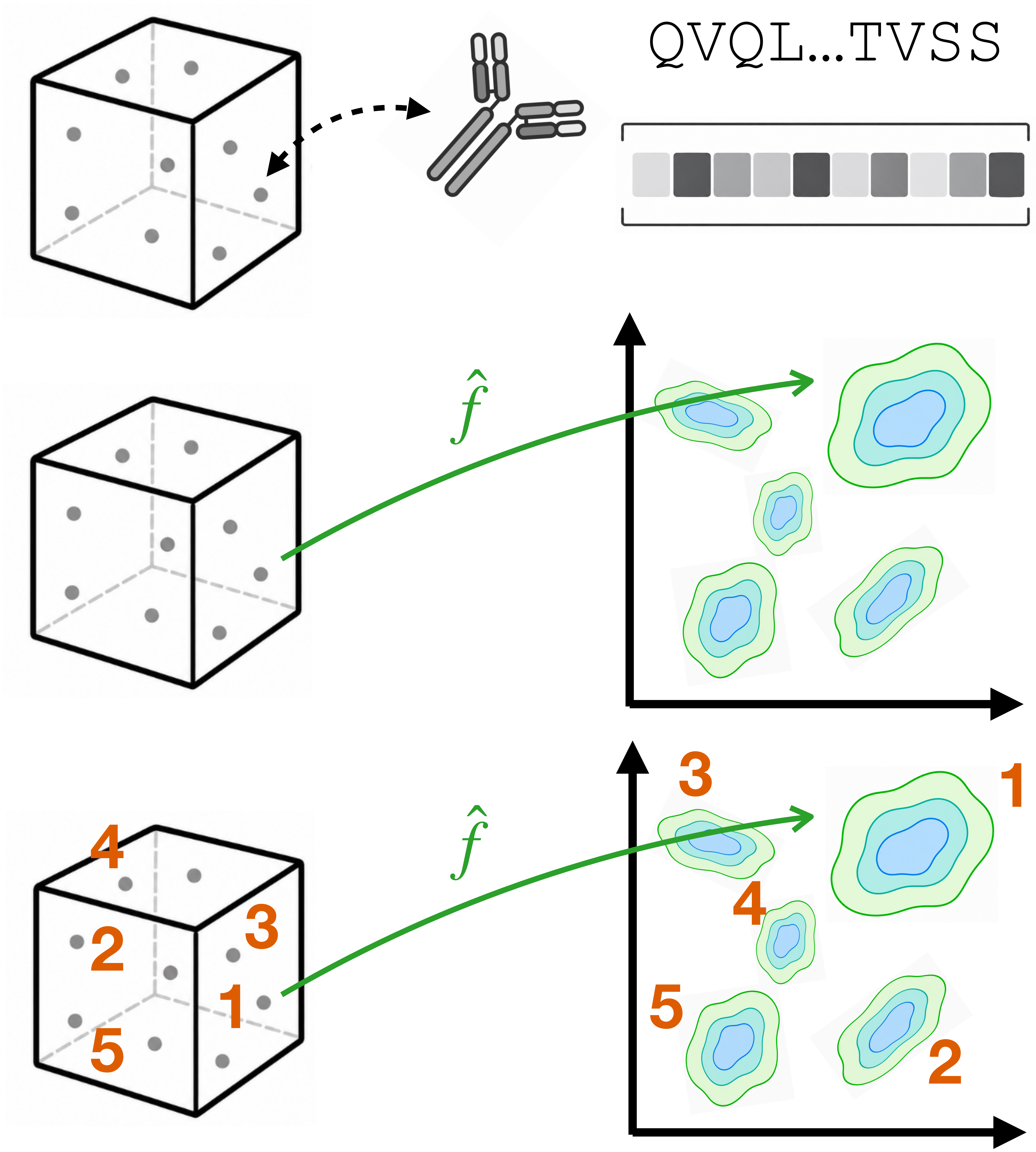
$$x^\star = \arg\max_{x \in \mathscr{X}} \alpha^{\hat{f}}(x)$$

BayesOpt algorithm

Propose design candidates in $\mathscr{X}$

Predict properties with surrogate model $\hat{f}$

Rank candidates with acquisition function



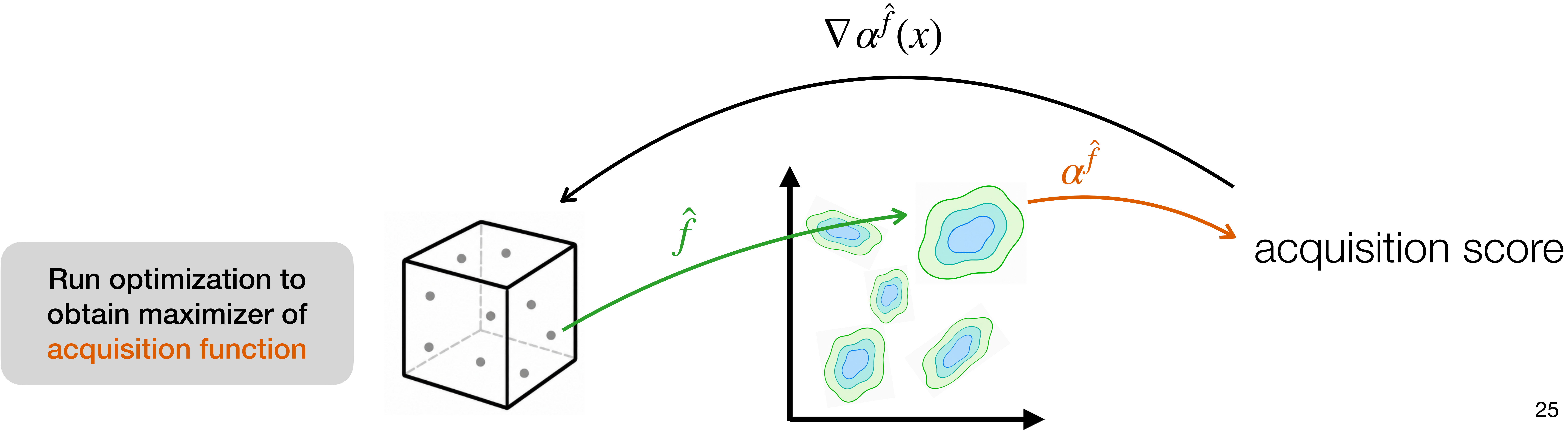
Append measurement to dataset $D = \{x^{(i)}, f(x^{(i)})\}_{i=1}^{n}$ and refit surrogate

Choose the candidate to evaluate

$$\boxed{x^\star} = \arg\max_{x \in \mathscr{X}} \alpha^{\hat{f}}(x)$$

Gradient-based optimization can be more efficient.

- If $\alpha^{\hat{f}}(x)$ is differentiable onto the continuous input x, we can directly run gradient-based optimization to obtain $x^\star = \arg\max_{x \in \mathcal{X}} \alpha^{\hat{f}}(x)$ instead of sampling candidates and ranking.
- More efficient for high-dimensional $\mathcal{X}$
- We'll show how to implement this in BoTorch.



Related concepts

- Quality indicator ✅

  How good is my Pareto front? Example: HV

- Utility function

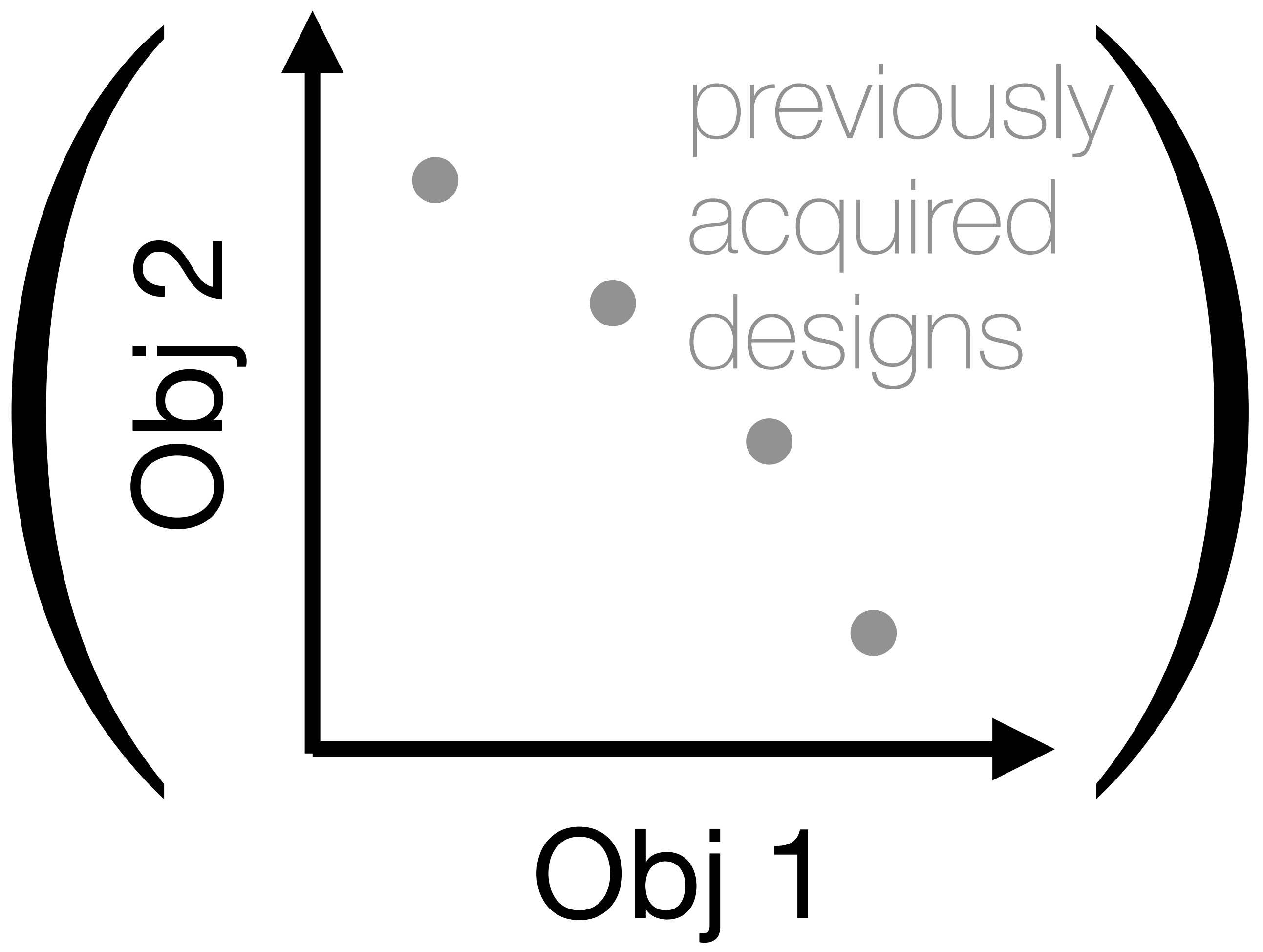
  How much did a design improve my Pareto front, once measured?

- Acquisition function

  How much is a design expected to improve my Pareto front, given my surrogate belief?

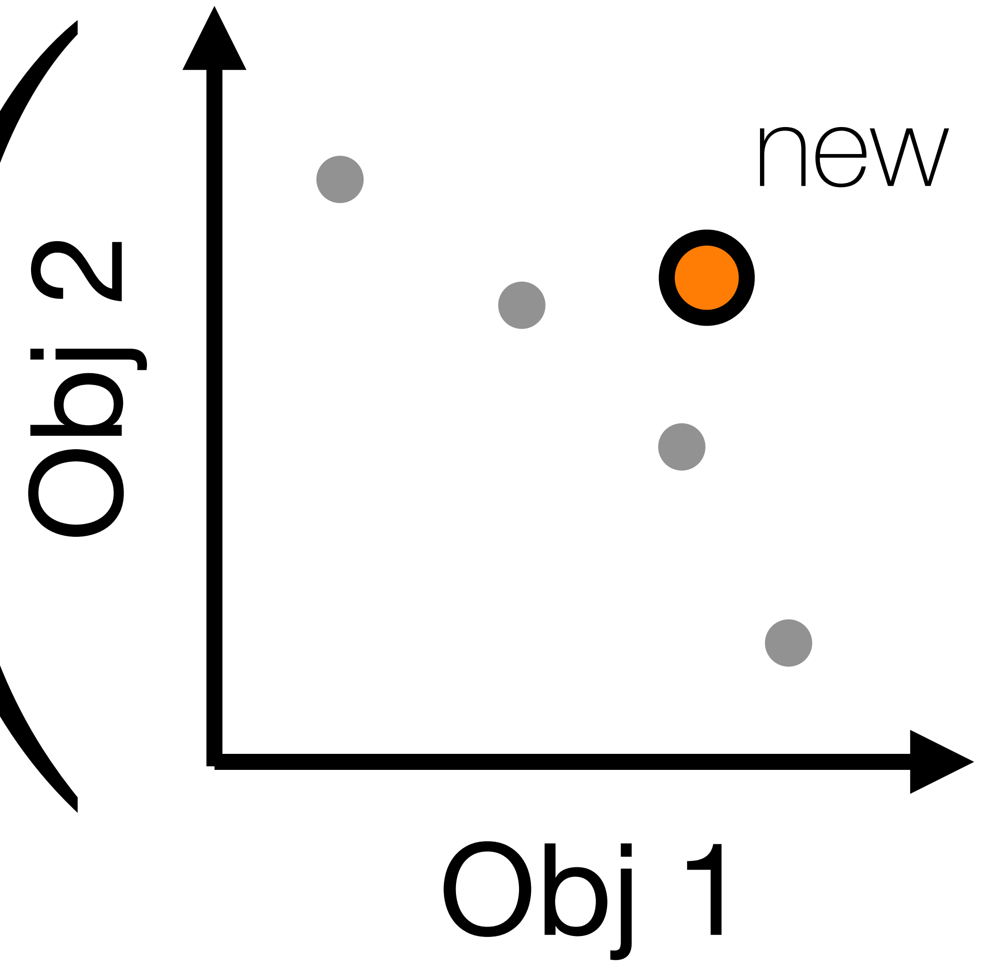
Quality indicator

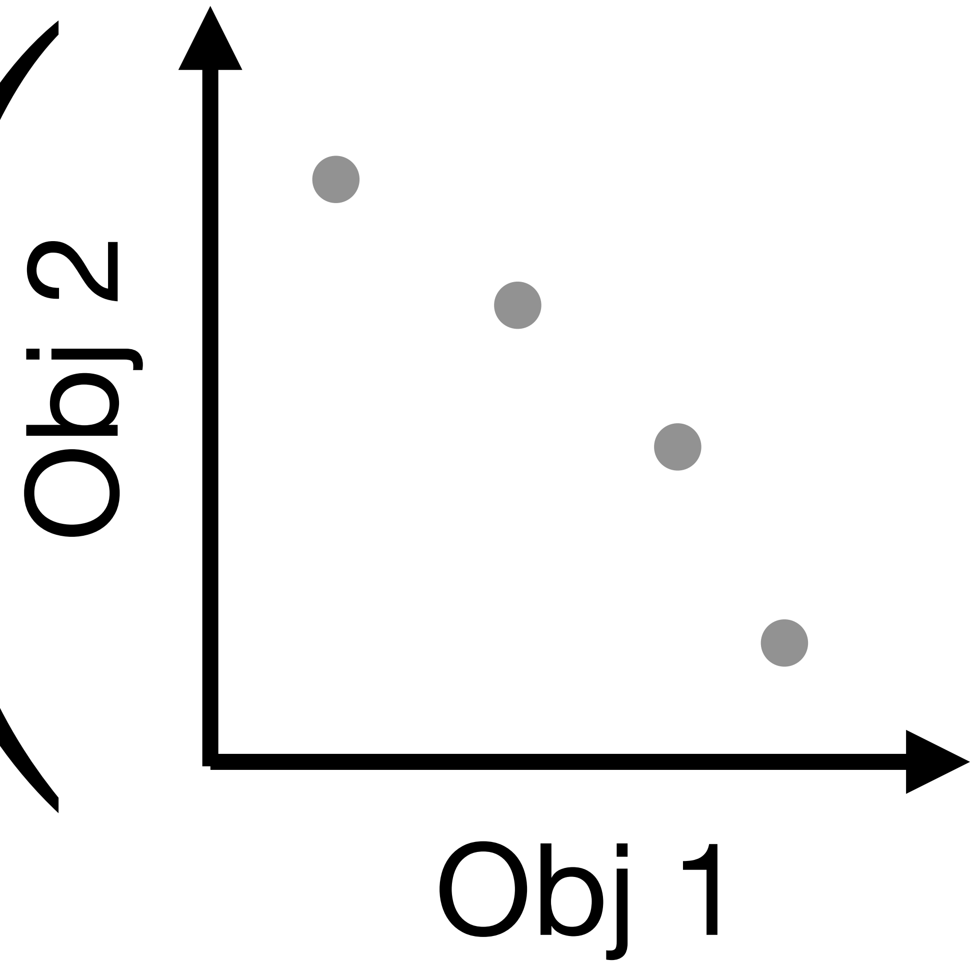
QI



Quality indicator → utility function

How useful was a design?

$\text{utility}(x_{\text{cand}}) = \text{QI}\Big($ 

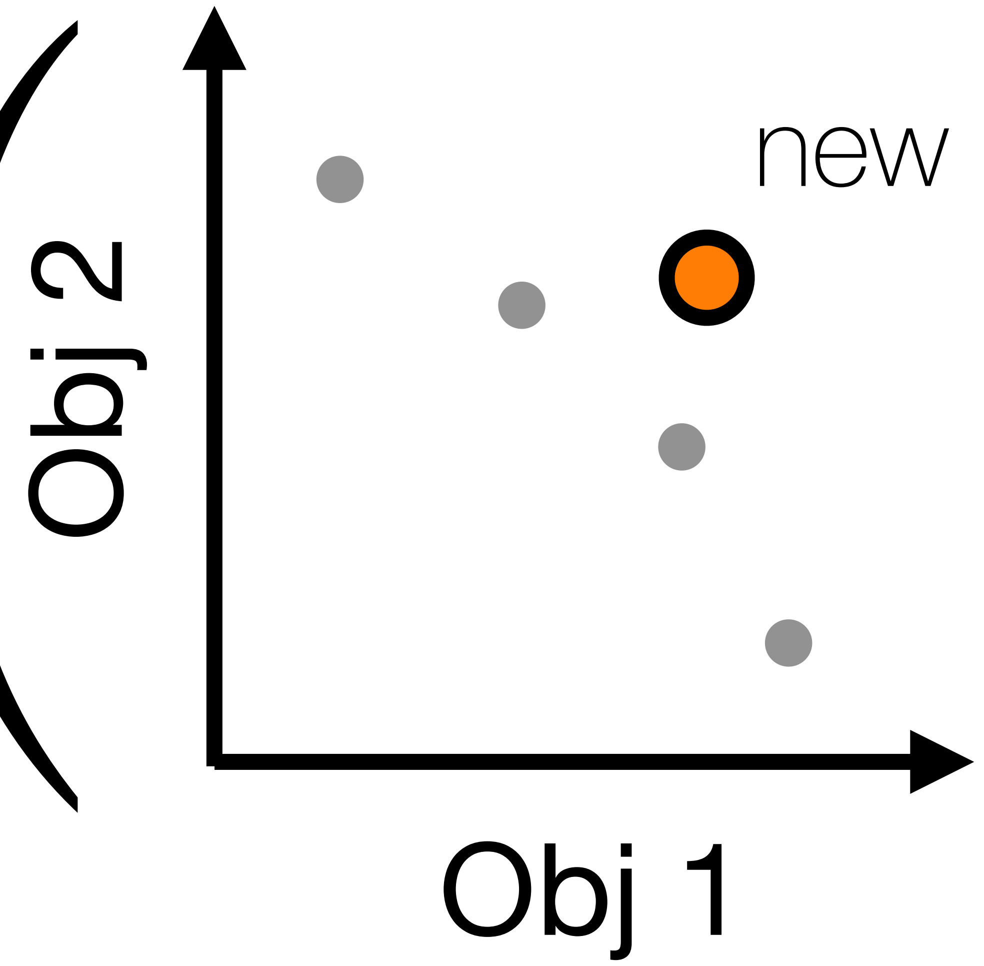
$\Big) - \text{QI}\Big($ 

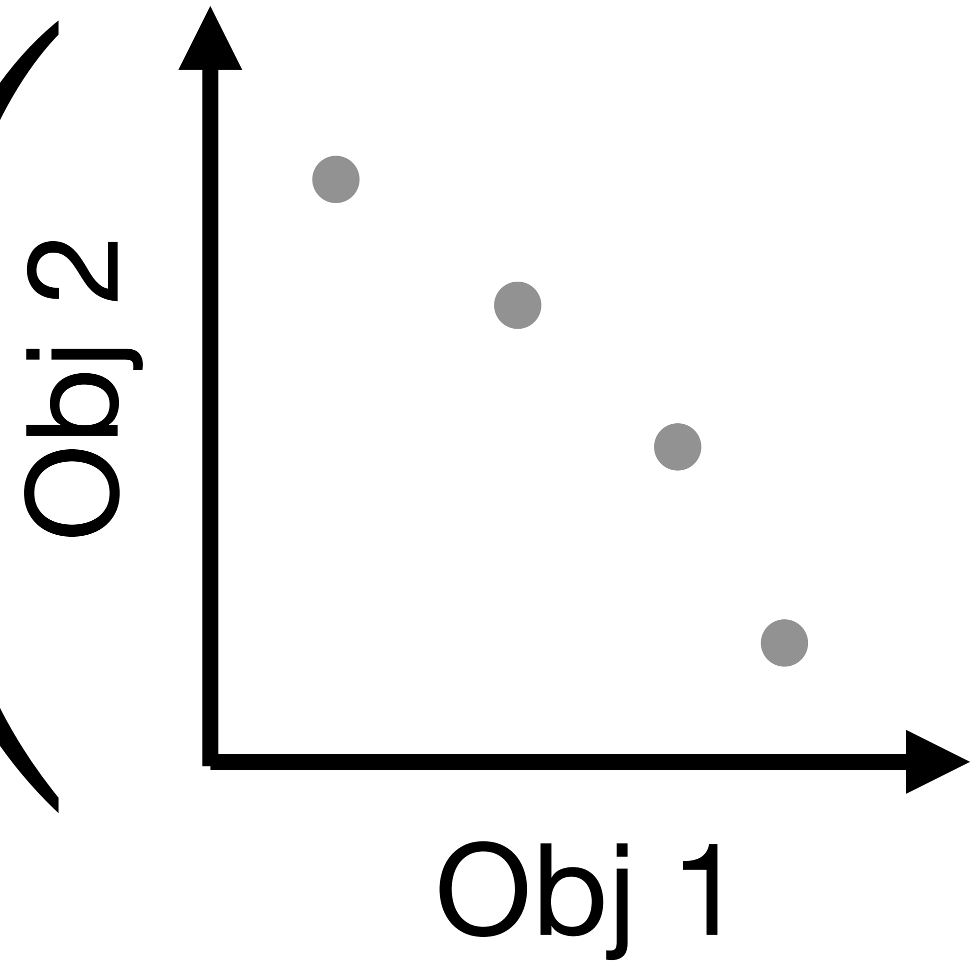
$\Big)$

Quality indicator → utility function → acquisition function

How useful was a design?

$$\text{utility}(x_{\text{cand}}) \;=\; \mathrm{QI}\left(\quad \right) \;-\; \mathrm{QI}\left(\quad \right)$$

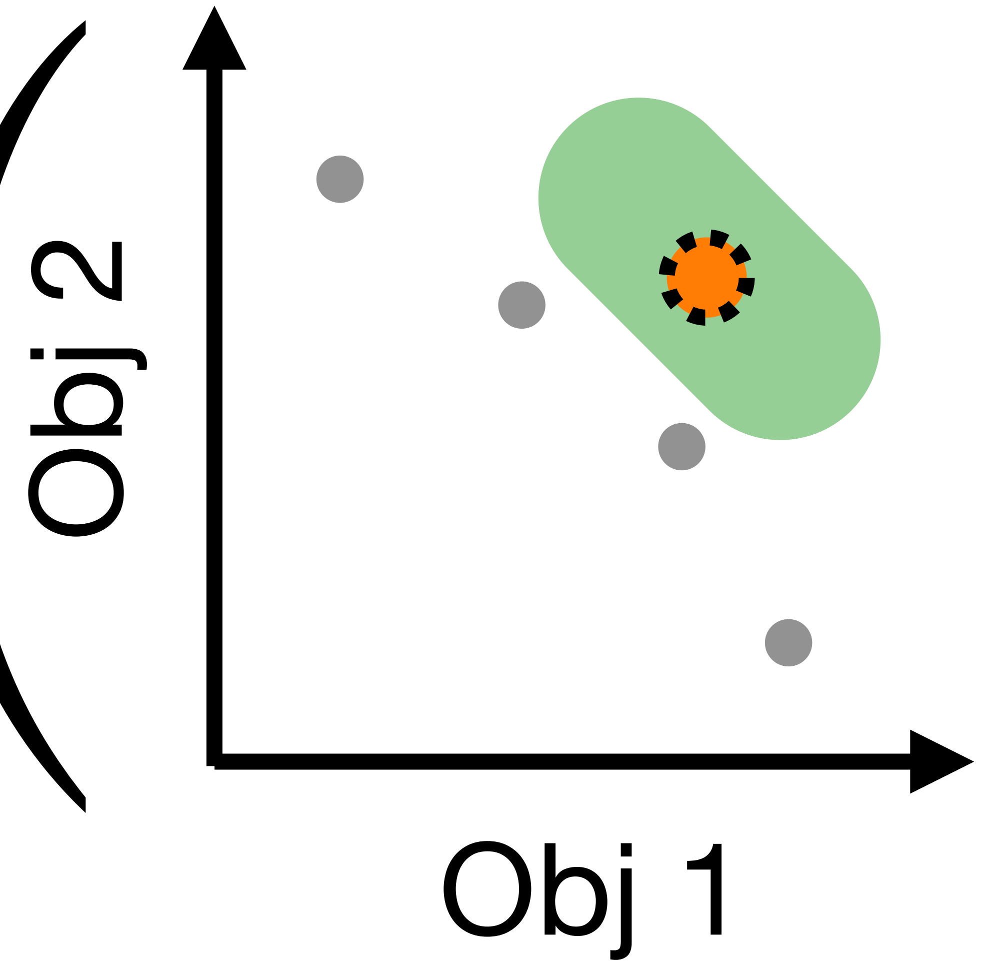


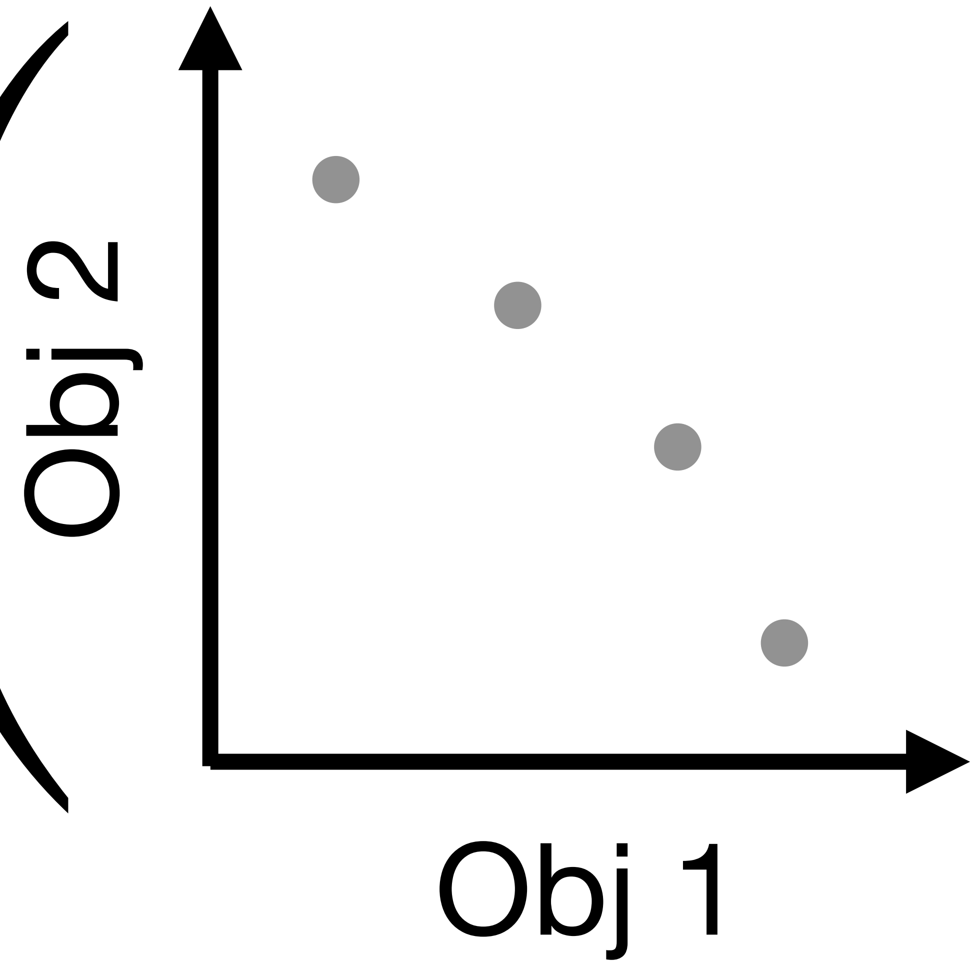


How useful is a design expected to be?

$$\alpha(x_{\text{cand}}) \;=\; \mathbb{E}_{\,\sim \hat{f}}\left[\mathrm{QI}\left(\quad \right) \;-\; \mathrm{QI}\left(\quad \right) \right]$$

posterior predictive sample





Related concepts

- Quality indicator ✅

  How good is my Pareto front?

- Utility function ✅

  How much did a design improve my Pareto front, once measured?

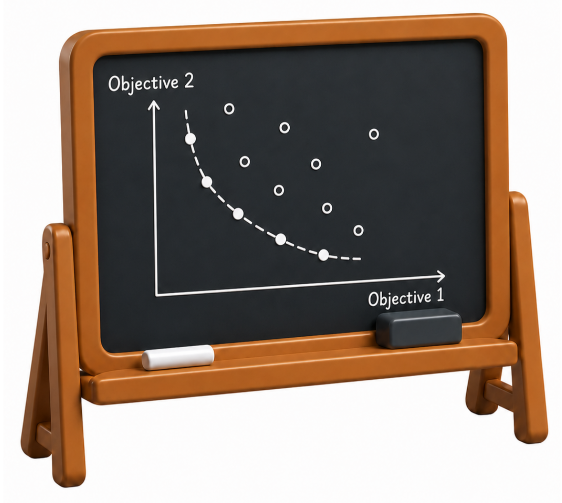
- Acquisition function ✅

  How much is a design expected to improve my Pareto front, given my surrogate belief?

Closely related!

Proposing an acquisition function ~ proposing utility

Lecture



I. Multi-objective optimization: the basics

- Motivation: lead optimization in drug discovery
- Pareto front
- Quality indicator

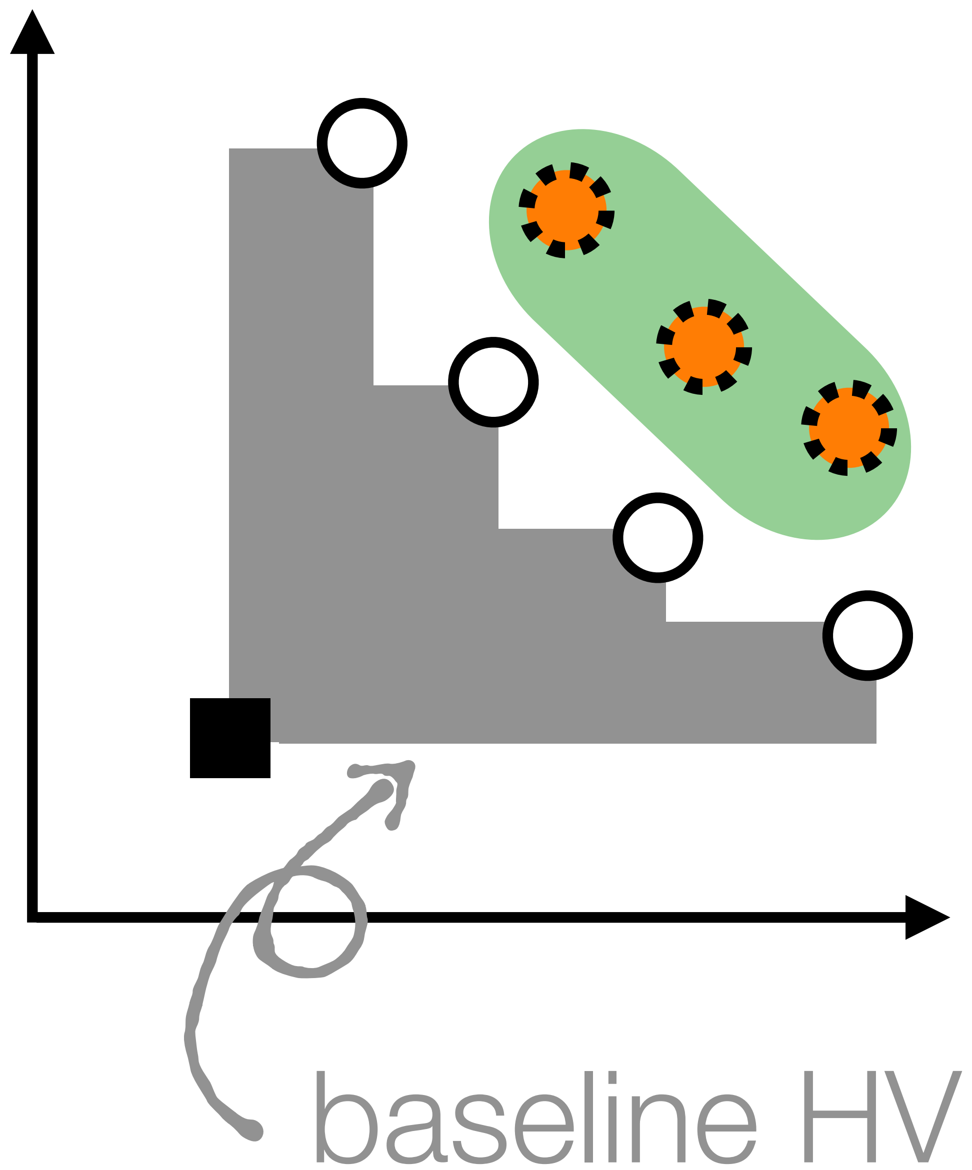
II. Bayesian optimization

- Surrogate modeling
- Utility function → acquisition function

III. Multi-objective reasoning

- Common multi-objective acquisition strategies
- Batched/noisy variants

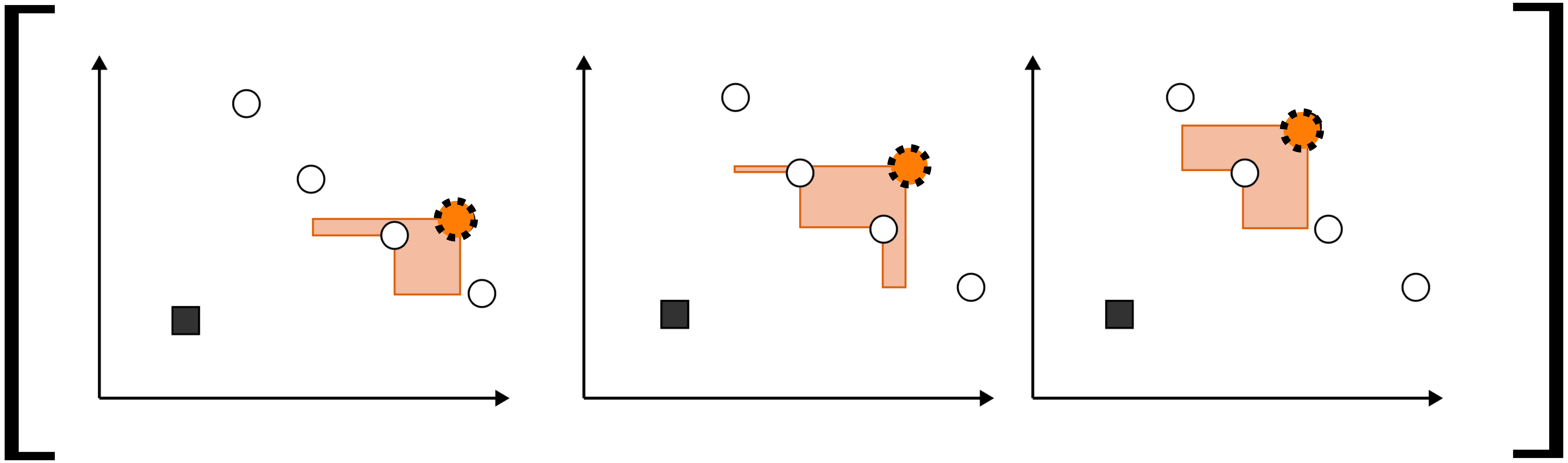
Expected hypervolume improvement (EHVI)

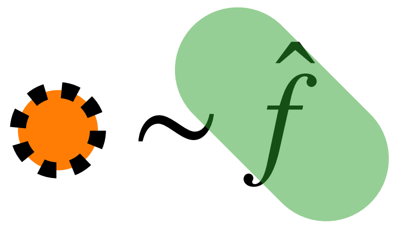


Quality indicator = HV

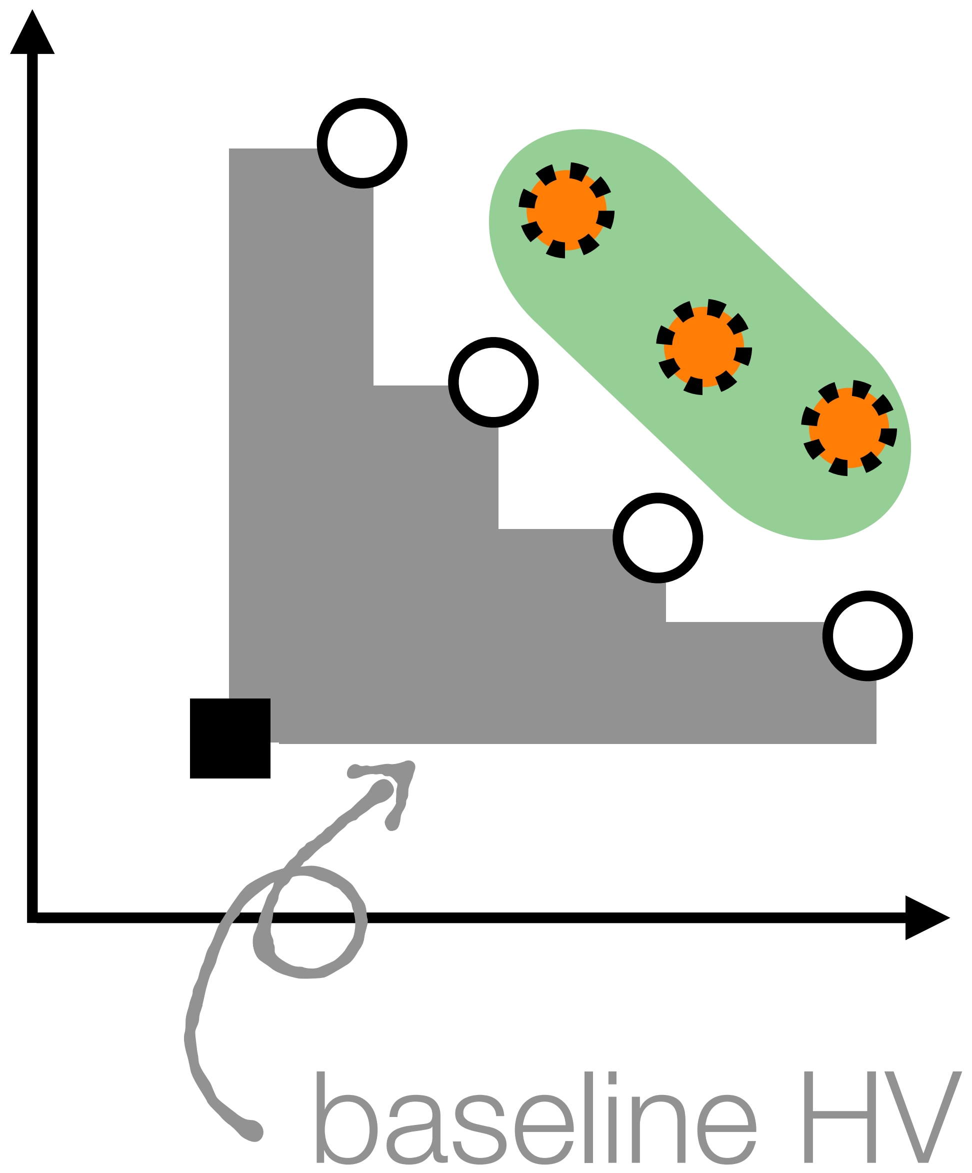
Utility function = HVI = HV $\left(f(X_{\text{base}}) \cup \{\hat{f}(x)\}\right)$ − HV $\left(f(X_{\text{base}})\right)$

Acquisition function = EHVI = $\mathbb{E}_{\hat{f}}$ [HVI]

$a_{\text{EHVI}}(x_{\text{cand}}) = \mathbb{E}$ 



Log Expected hypervolume improvement (EHVI)



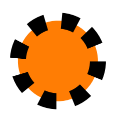
Quality indicator = HV

Utility function = $\mathrm{HVI} = \mathrm{HV}\left(f(X_{\mathrm{base}}) \cup \{\hat{f}(x)\}\right) - \mathrm{HV}\left(f(X_{\mathrm{base}})\right)$

Acquisition function = $\mathrm{LogEHVI} = \log \mathbb{E}_{\hat{f}}\,[\mathrm{HVI}]$

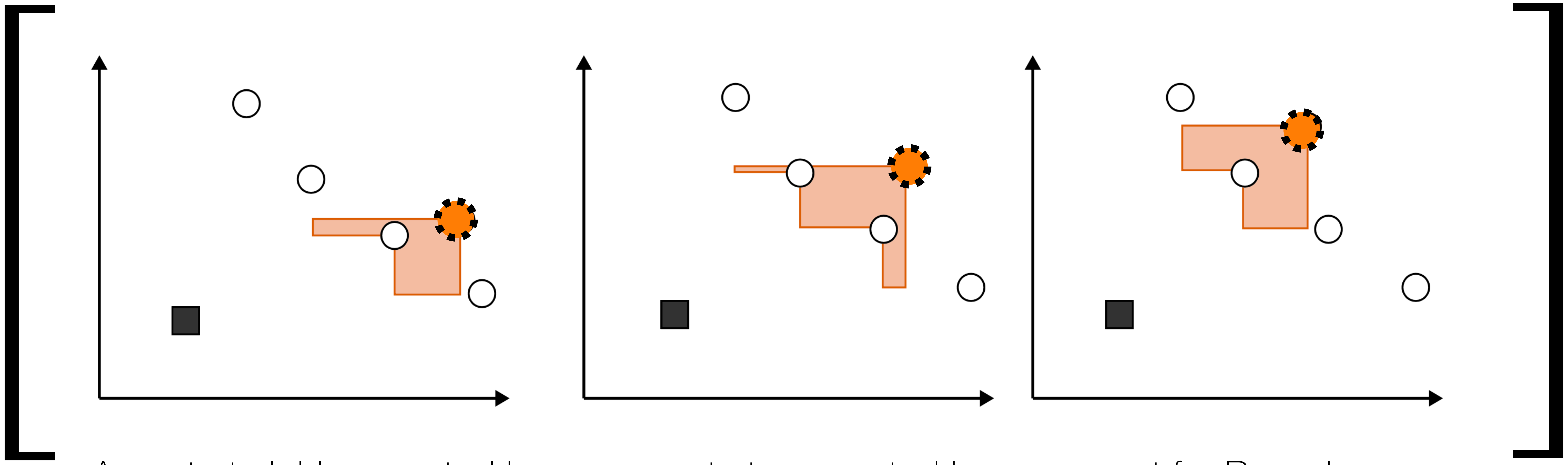
more numerically stable

$a_{\mathrm{EHVI}}(x_{\mathrm{cand}}) = \mathbb{E}$



$\sim \hat{f}$

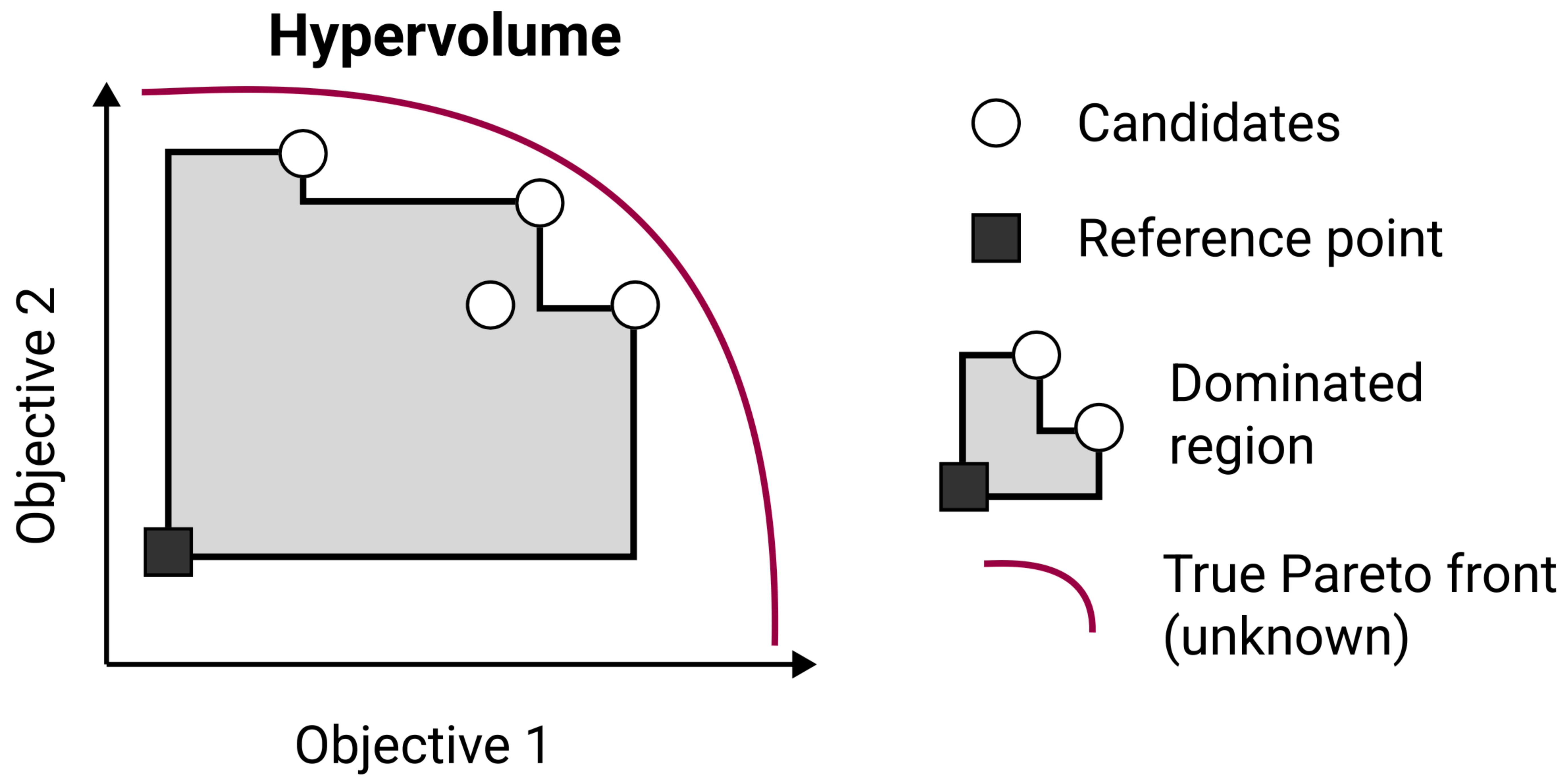
posterior predictive sample

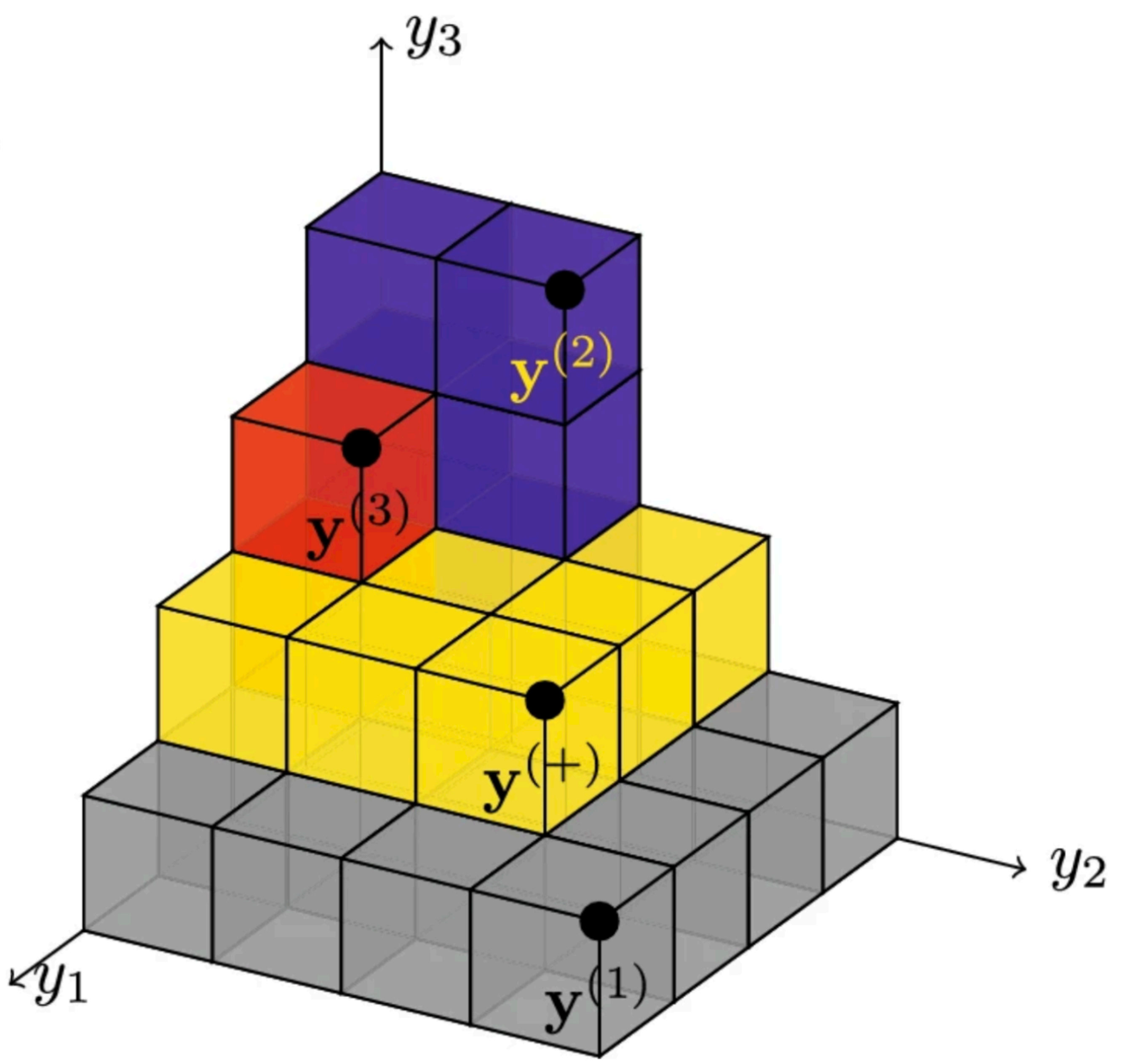


Ament et al. Unexpected improvements to expected improvement for Bayesian optimization. NeurIPS (2023).

EHVI: disadvantages

- $\mathrm{HV} \sim \mathcal{O}(n^{\lfloor \frac{M}{2} \rfloor}) \rightarrow$ impractical for $M > 4$ despite box decomposition[4]
- Sensitive to rescaling of objectives, with different natural units





[5] Emmerich, Deutz, and Klinkenberg, "Hypervolume-based expected improvement." IEEE CEC (2011).
[6] Yang, Kaifeng et al. "A multi-point mechanism of expected hypervolume improvement for parallel multi-objective Bayesian global optimization". GECCO (2019).

Scalarization strategies: linear sum

1. Simplify original problem

$$\max_{x \in \mathcal{X}} \quad f(x) = \big(f_1(x), \ldots, f_M(x)\big)$$

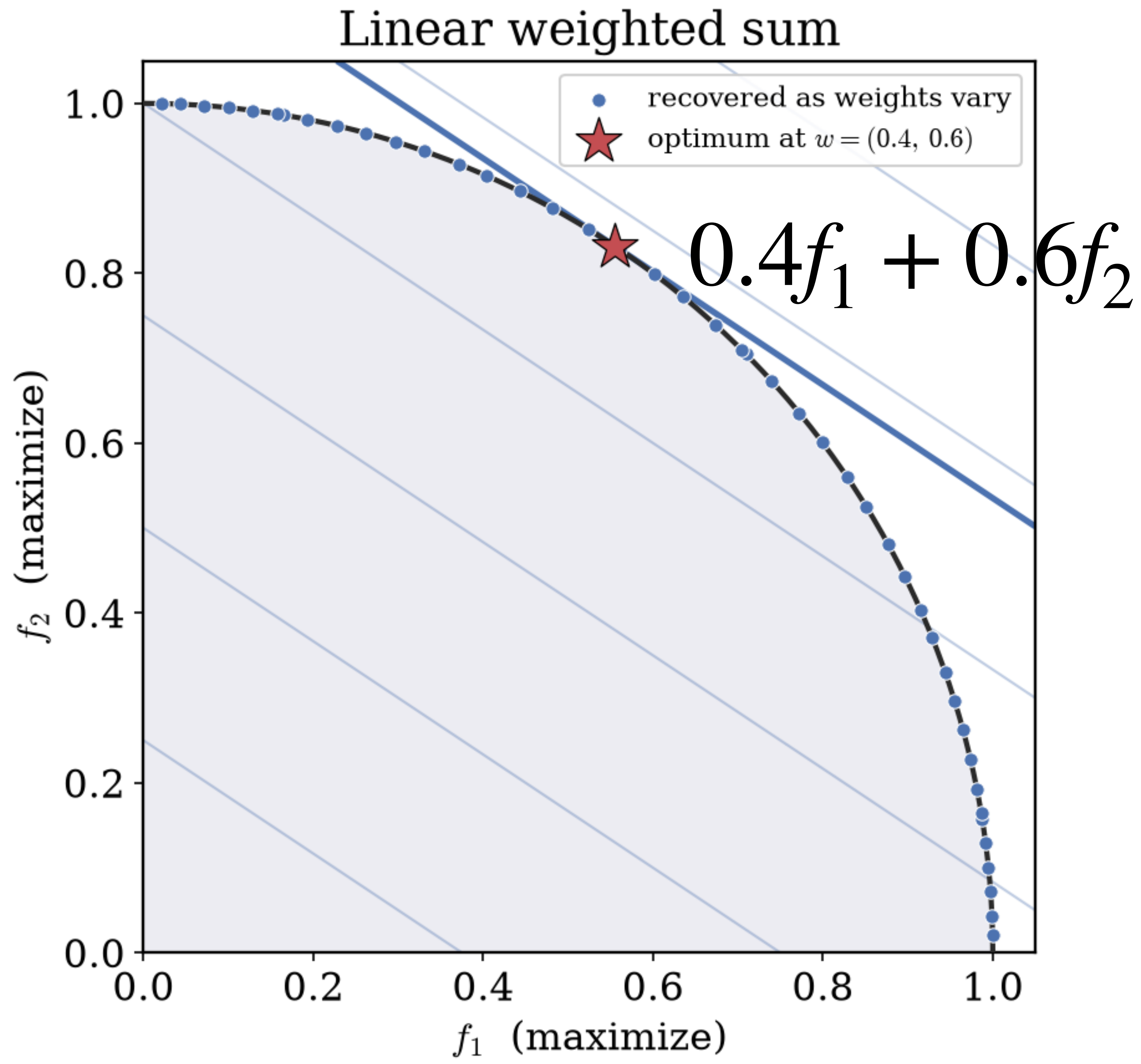
into $\max_{x \in \mathcal{X}} \quad g(x) = w_1 f_1(x) + \ldots + w_M f_M,$

with chosen weights $w_1, \ldots, w_M$ summing to 1.

2. Use a single-objective acquisition function, like EI.

$$a^{\hat{f}}(x) = \mathbb{E}_{\hat{f}}\left[\hat{g}(x) - g_{\text{best}}\right]_+$$

3. Resample different weights.



Zooming out: taxonomy of multi-objective optimization

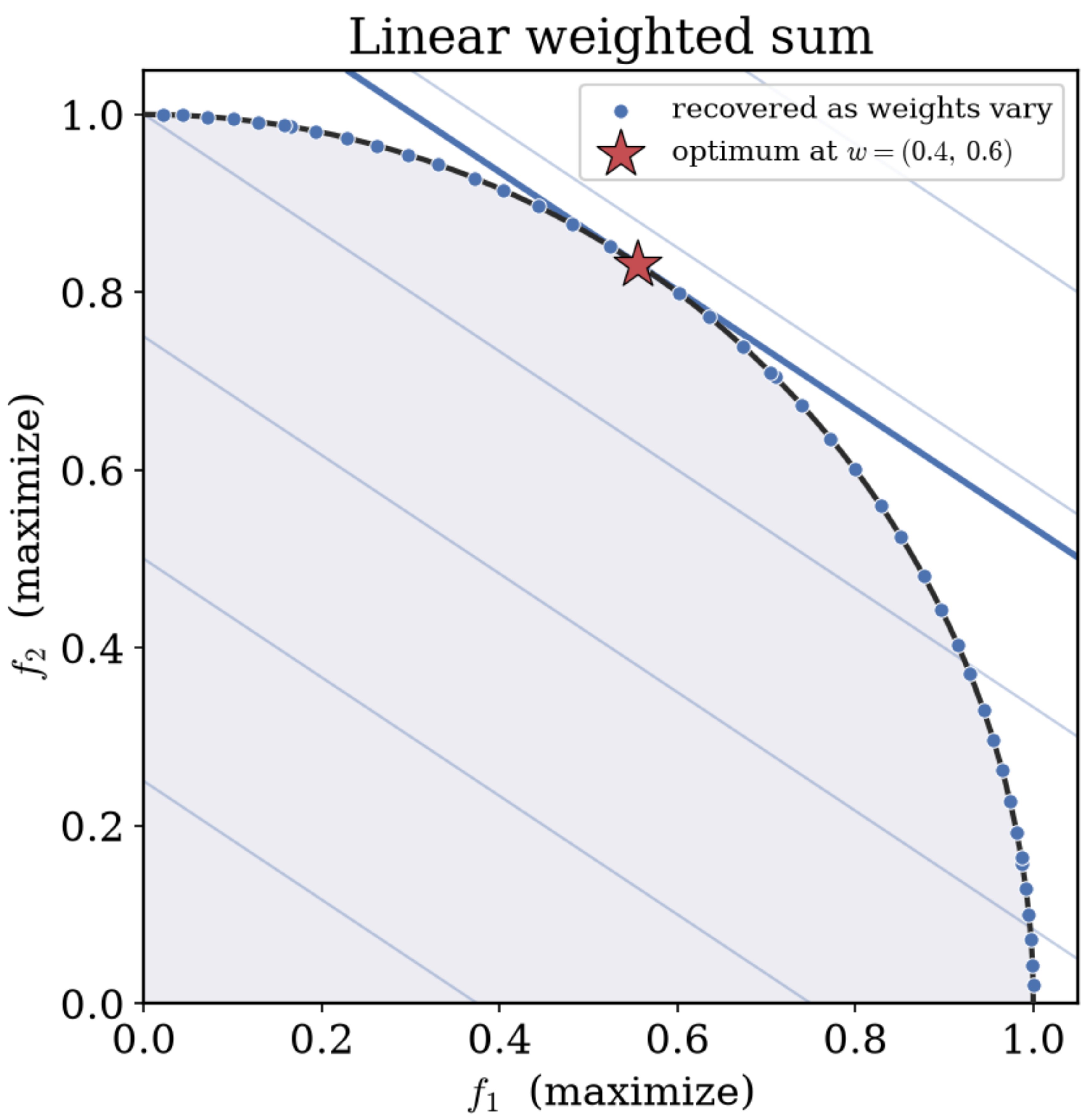
- ***a priori*** : deliver specific solutions (e.g., targeting a region of $\mathscr{P}$) already capturing decision maker's preferences

  rarer scenario

- ***a posteriori*** : approximate the entire $\mathscr{P}$ and decision maker chooses afterwards

  focus today

- interactive : alternate proposing solutions + eliciting preference information from decision maker



Scalarization strategies: linear sum

1. Simplify original problem

$$\max_{x\in\mathcal{X}} \quad f(x) = \big(f_1(x), \ldots, f_M(x)\big)$$

into $\max_{x\in\mathcal{X}} \quad w_1 f_1(x) + \ldots f_M$,

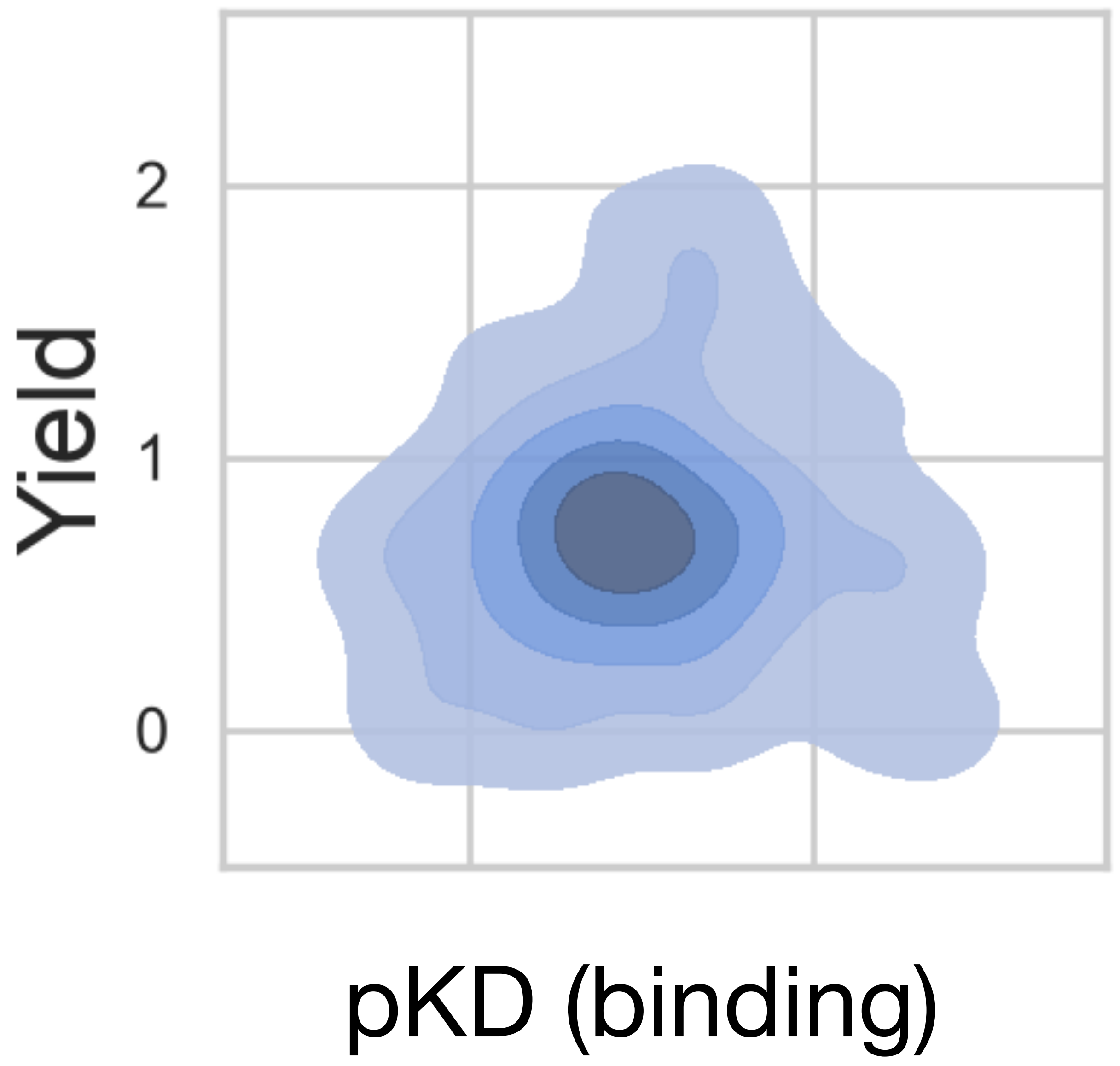
with ... weights $w_1, \ldots, w_M$ summing to 1.

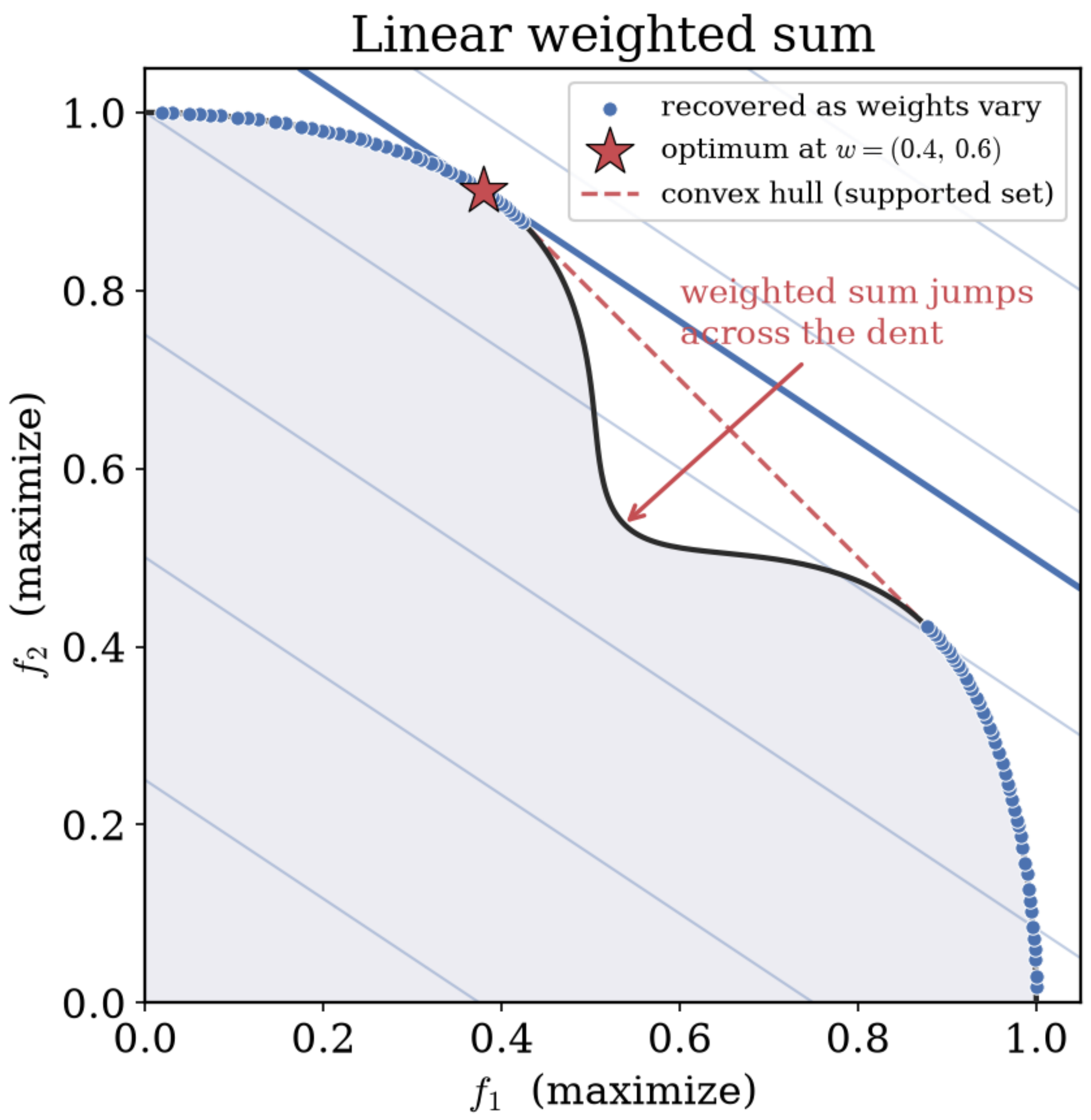
Can't recover unsupported points of non-convex PF

2. Use a single-objective a

$$a^{\hat{f}}(x) = \mathbb{E}$$

3. Resample different weig



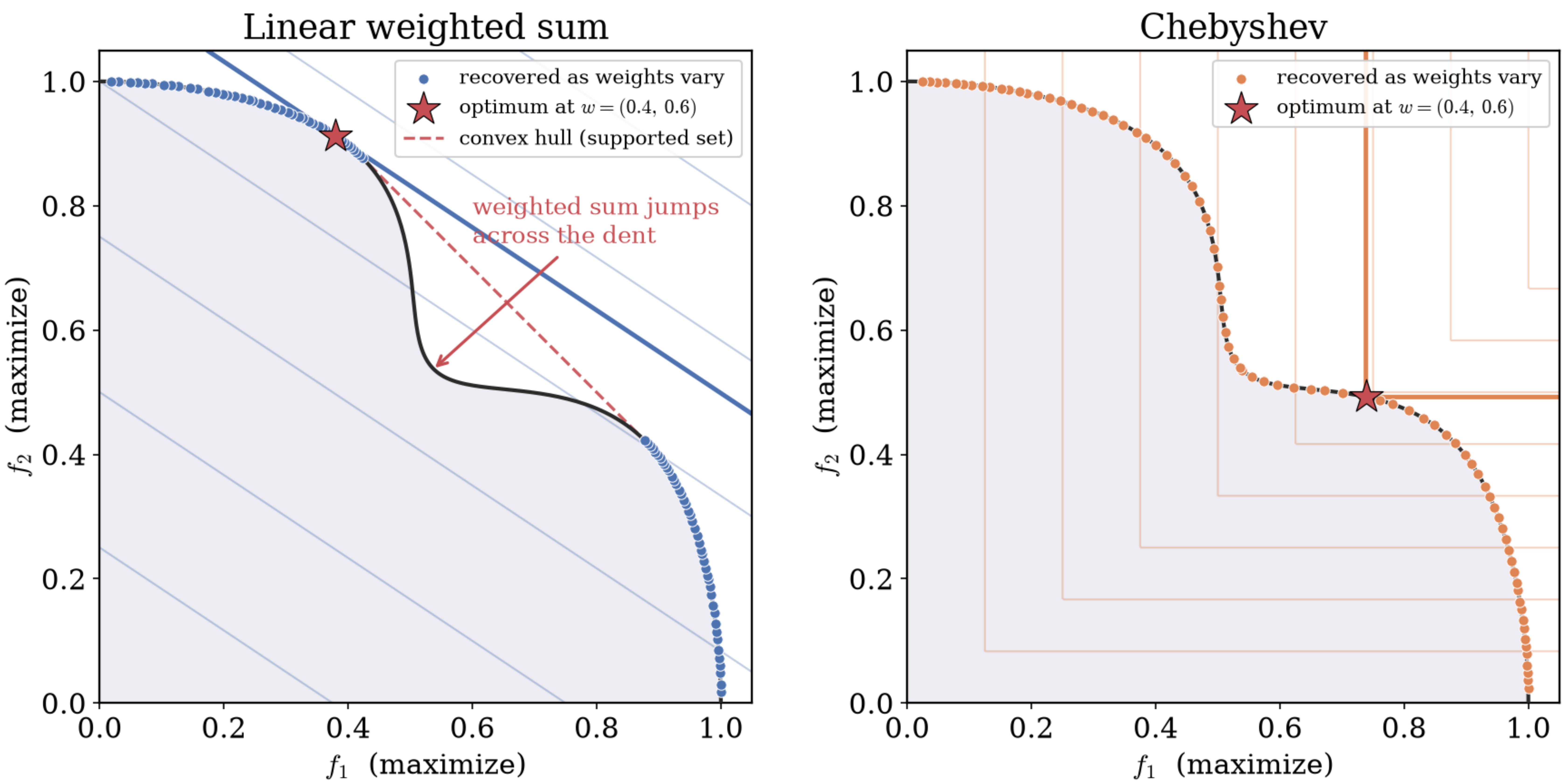


Scalarization strategies: Chebyshev

$$\max_{x \in \mathcal{X}} \min_{i} w_i (f_i - r_i^{\text{nadir}}),$$

opposite intuition from linear sum

with weights $w_1, \ldots, w_M$ summing to 1, where w_i is <u>smaller</u> for preferred objectives

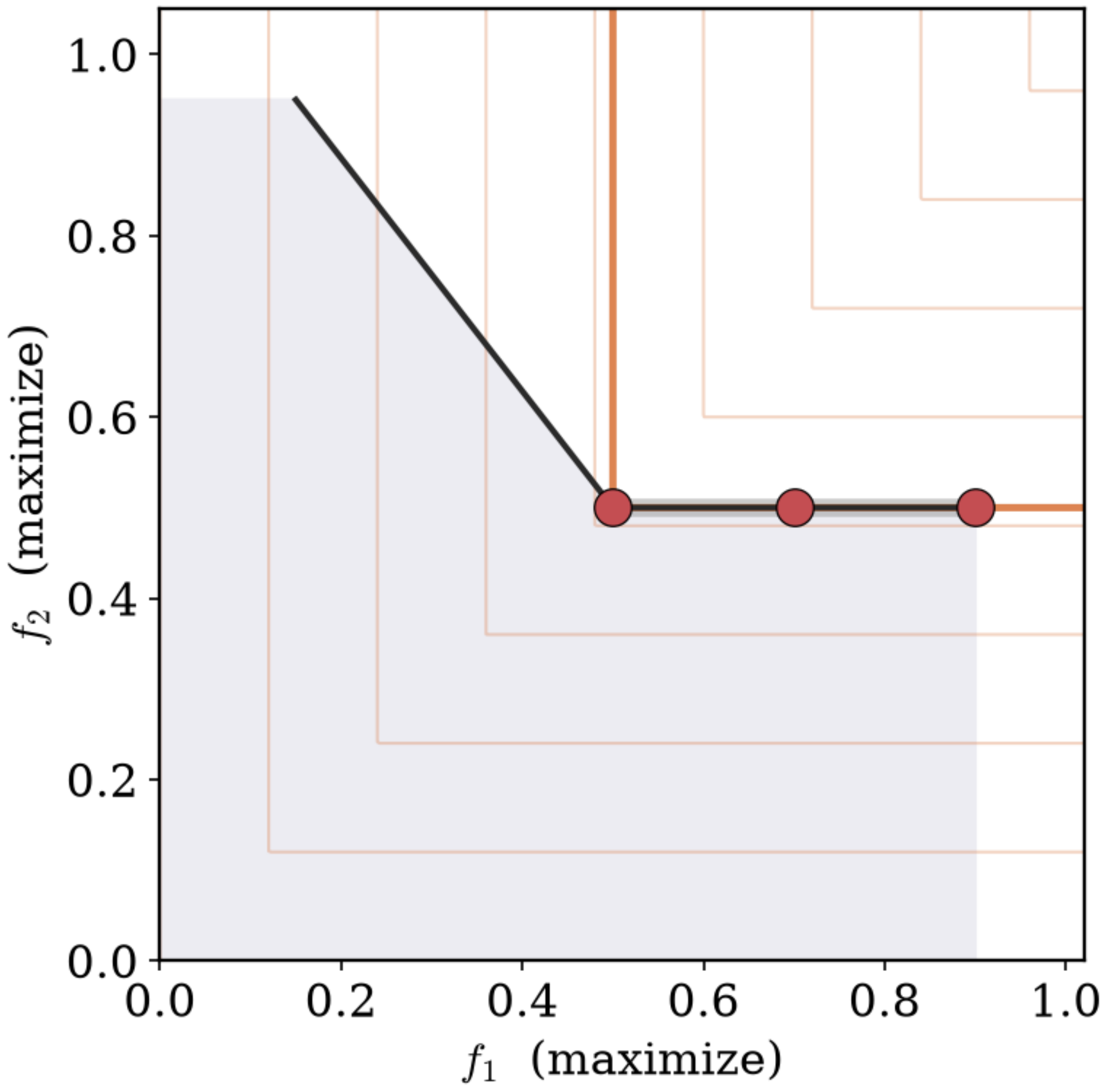


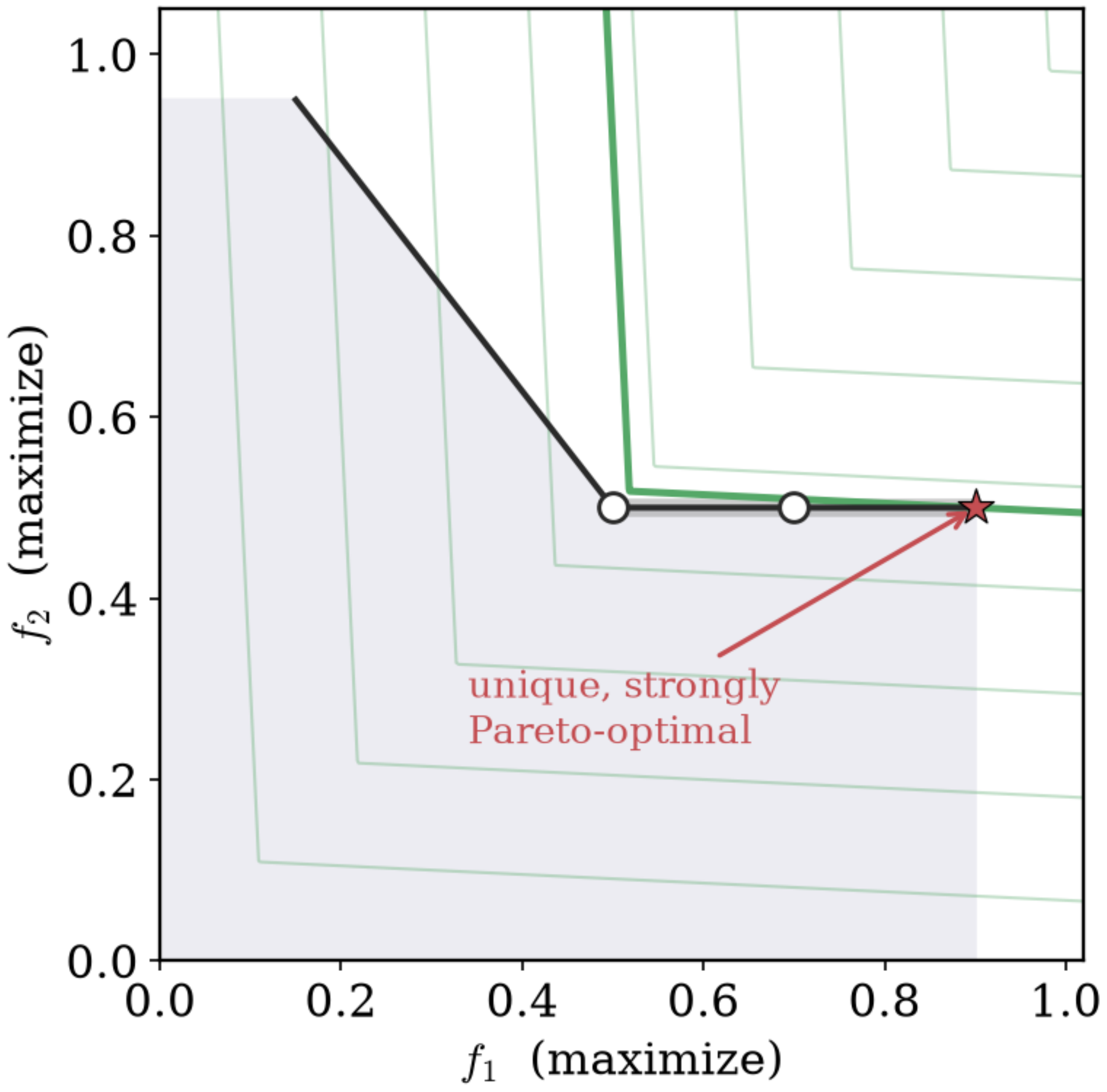
Objectives must be normalized

Scalarization strategies: augmented Chebyshev

$$\max_{x \in \mathcal{X}} \left[\min_i \; w_i (f_i - r_i^{\text{nadir}}) + \rho \sum_i w_i (f_i - r_i^{\text{nadir}}) \right],$$

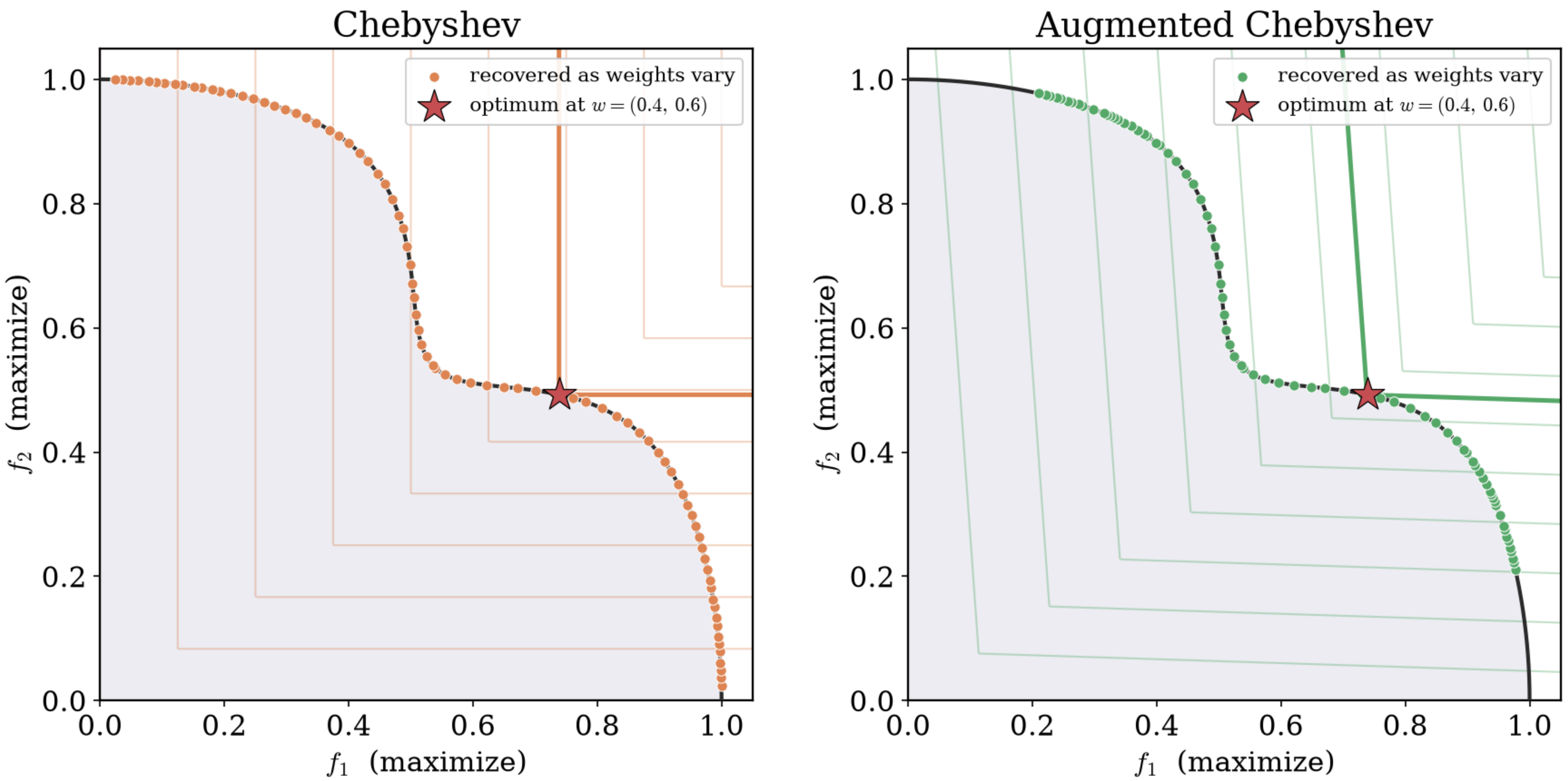
with weights $w_1, \ldots, w_M$ summing to 1, where w_i is <u>smaller</u> for preferred objectives





ParEGO = (augmented) Chebyshev + resampling weights

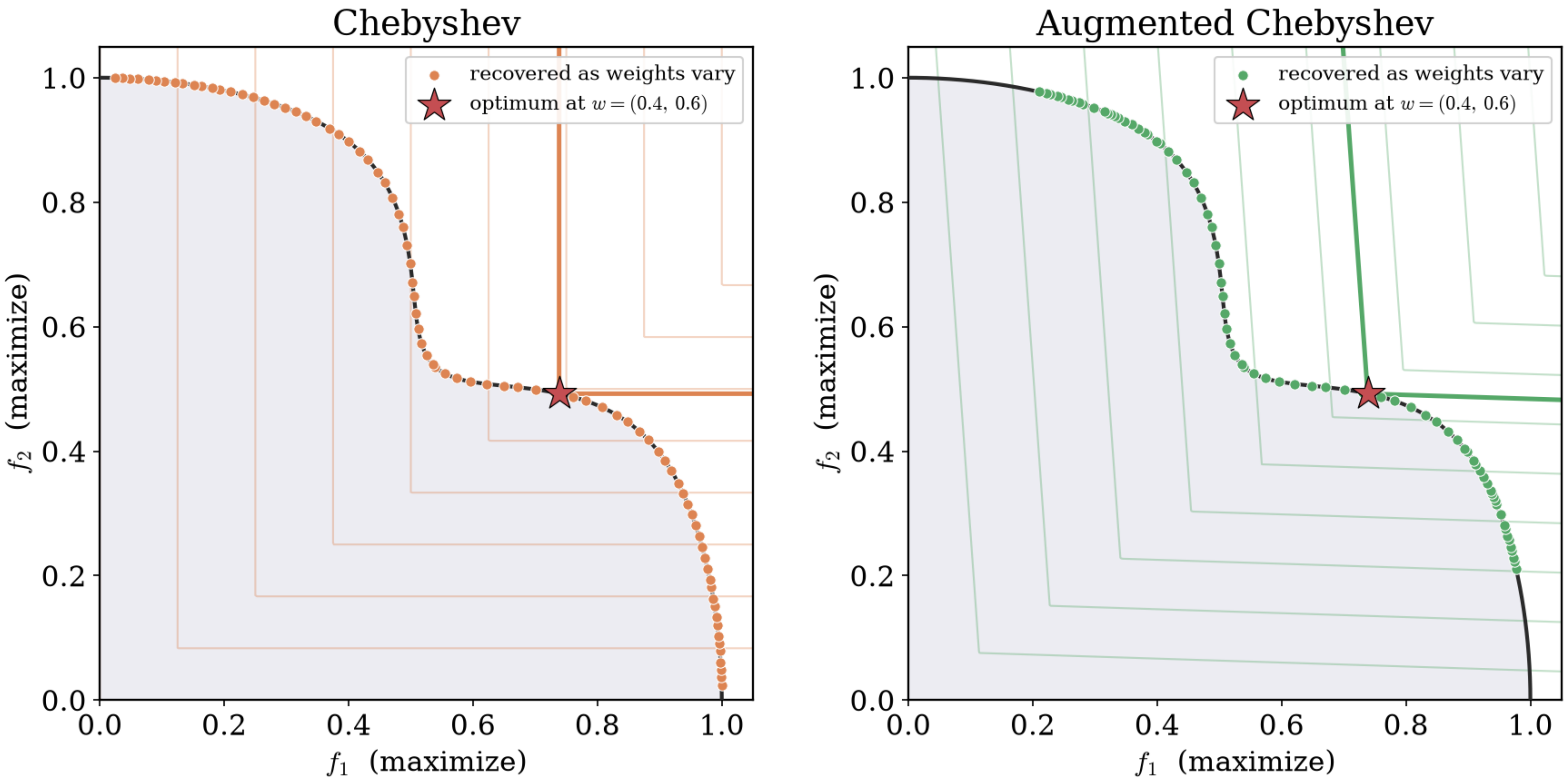
$$\max_{x \in \mathcal{X}} \left[\min_i \; w_i (f_i - r_i^{\text{nadir}}) + \rho \sum_i w_i (f_i - r_i^{\text{nadir}}) \right]$$, with weights $w_1, \ldots, w_M$ resampled every MOBO iteration.



Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE transactions on evolutionary computation* (2006).

Scalarization: disadvantages

- One weight vector is only a temporary utility; you use your entire batch on it
- Need to sample many weight vectors, and coverage difficult for complex + high-dim PF
- Still sensitive to rescaling of objectives



[Advanced, see optional notebook]
Information-theoretic acquisition functions

Predictive entropy search (PES): learn which designs are Pareto-optimal

$$\alpha_{\text{PES}}(x) = I(y_x; \mathcal{X}^\star \,|\, \mathcal{D}) = H[p(y_x \,|\, \mathcal{D})] - \mathbb{E}_{\mathcal{X}^\star}[H[p(y_x \,|\, D, \mathcal{X}^\star)]]$$

Max-value entropy search (MES): learn what the PF trade-off looks like in objective space

$$\alpha_{\text{MES}}(x) = I(y_x; \mathcal{Y}^\star \,|\, \mathcal{D}) = H[p(y_x \,|\, \mathcal{D})] - \mathbb{E}_{\mathcal{Y}^\star}[H[p(y_x \,|\, D, \mathcal{Y}^\star)]]$$

Joint entropy search (JES): learn both

$$\alpha_{\text{JES}}(x) = I(y_x; \mathcal{X}^\star, \mathcal{Y}^\star \,|\, \mathcal{D}) = H[p(y_x \,|\, \mathcal{D})] - \mathbb{E}_{\mathcal{X}^\star, \mathcal{Y}^\star}[H[p(y_x \,|\, D, \mathcal{X}^\star, \mathcal{Y}^\star)]]$$

PES: Hernandez-Lobato et al, "Predictive entropy search for multi-objective Bayesian optimization." ICML (2016).
MES: Belakaria et al, "Max-value entropy search for multi-objective Bayesian optimization." NeurIPS (2019).
JES: Tu et al, "Joint entropy search for multi-objective Bayesian optimization." NeurIPS (2022).

Recap: multi-objective acquisition strategies

- Common acquisition strategies
  - EHVI
  - scalarization with random weights + EI
    - linear weighted sum, (augmented) Chebyshev
  - entropy search
- Easy to reason about acquisition functions in terms of the underlying **utility function**
  - Often, acquisition function = posterior expectation of utility function

Exercise: work out what the underlying utility function is for each.

Recap: multi-objective acquisition strategies

- Common acquisition strategies
  - EHVI ← HVI
  - scalarization with random weights + EI ← improvement of scalarized objective
    - linear weighted sum, (augmented) Chebyshev
  - entropy search ← information gain
- Easy to reason about acquisition functions in terms of the underlying **utility function**
  - Often, acquisition function = posterior expectation of utility function

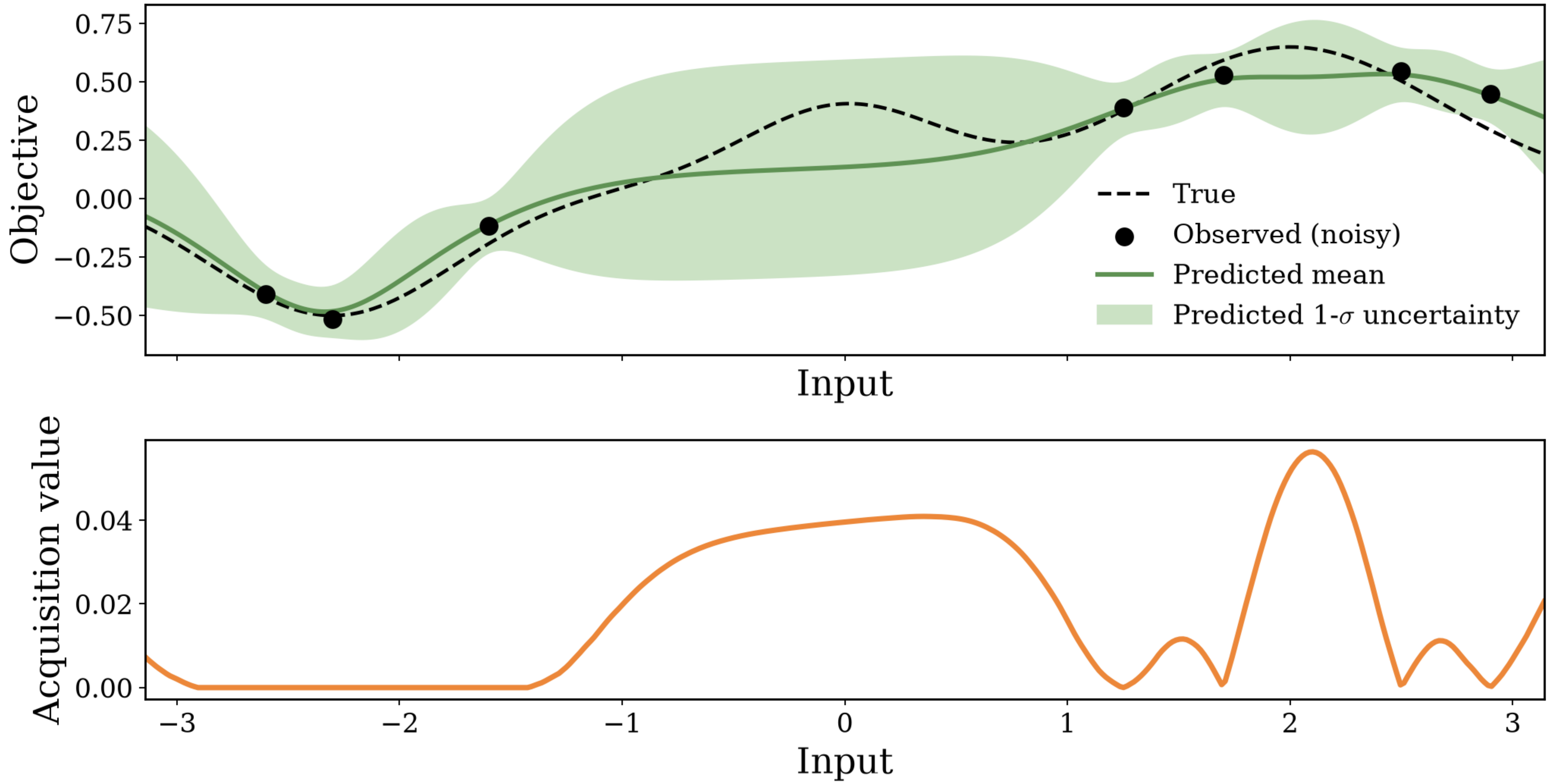
Exercise: work out what the underlying utility function is for each.

Noisy variants of acquisition functions

$$a_{\mathrm{EI}}^{\hat{f}}(x) = \mathbb{E}_{\hat{f}}\,[\hat{f}(x) - f_{\mathrm{best}}]_+$$

- Real-world observations carry measurement noise.
- If the surrogate models the noise, use noisy EI instead of EI:

$$\alpha_{\mathrm{NEI}}^{\hat{f}}(x) = \mathbb{E}_{\hat{f}}\,[\hat{f}(x) - \max_{x' \in X_{\mathrm{base}}} \hat{f}(x')]_+$$



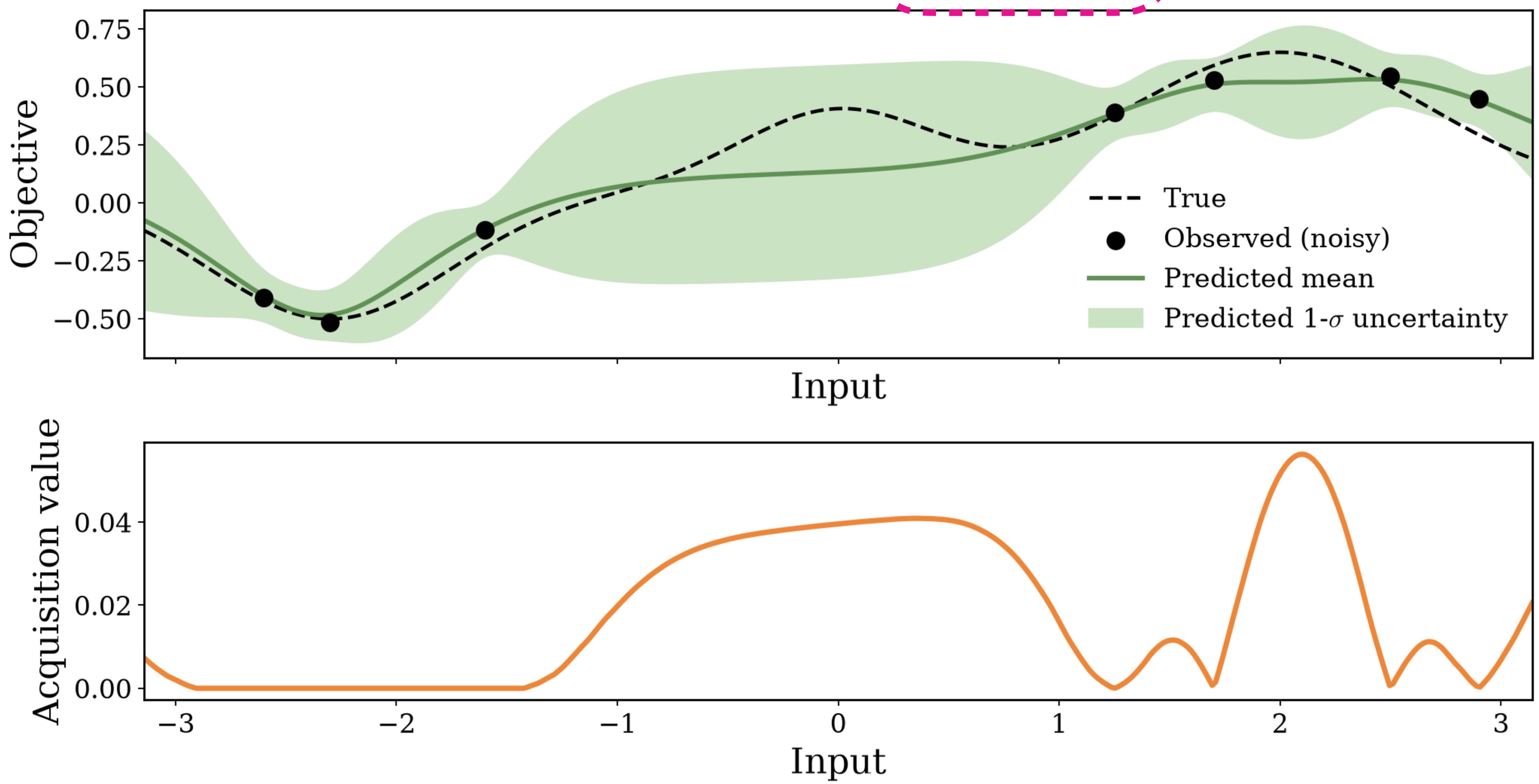
Noisy variants of acquisition functions

$$a_{\mathrm{EI}}^{\hat{f}}(x) = \mathbb{E}_{\hat{f}}\,[\hat{f}(x) - f_{\mathrm{best}}]_+$$

- Real-world observations carry measurement noise.
- If the surrogate models fit the noise, use noisy EI instead of EI:

$$\alpha_{\mathrm{NEI}}^{\hat{f}}(x) = \mathbb{E}_{\hat{f}}\,[\hat{f}(x) - \max_{x' \in X_{\mathrm{base}}} \hat{f}(x')]_+$$

"best observed so far" point now uncertain



Noisy variants of acquisition functions

$$a_{\mathrm{EI}}^{\hat{f}}(x) = \mathbb{E}_{\hat{f}}\,[\hat{f}(x) - f_{\mathrm{best}}]_+$$

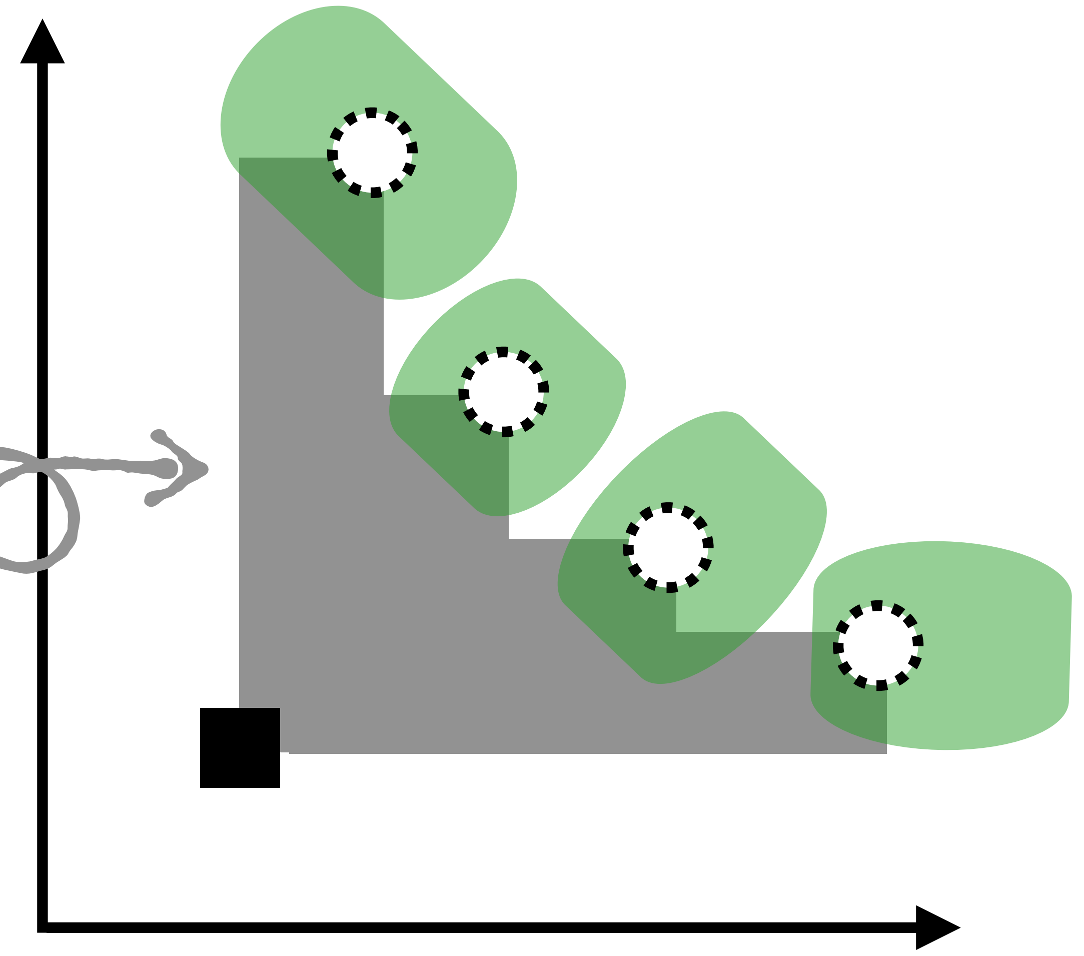
- Real-world observations carry measurement noise.
- If the surrogate models fit the noise, use noisy EI instead of EI:

$$\alpha_{\mathrm{NEI}}^{\hat{f}}(x) = \mathbb{E}_{\hat{f}}\,[\hat{f}(x) - \max_{x' \in X_{\mathrm{base}}} \hat{f}(x')]_+$$

- For $M > 1$, use noisy EHVI instead of EHVI:

$$\alpha_{\mathrm{NEHVI}}^{\hat{f}}(x) = \mathbb{E}_{\hat{f}}\left[\mathrm{HV}\left(\hat{f}(X_{\mathrm{base}}) \cup \{\hat{f}(x)\}\right) - \mathrm{HV}\left(\hat{f}(X_{\mathrm{base}})\right)\right]$$

baseline HV now uncertain



baseline HV

Batched variants of acquisition functions

- So far, we talked about acquiring one design per BO iteration.
- For molecular design, the wet lab runs many molecules at a time (e.g., 96-well plates).
- Need to select **candidate sets** of size q, not singletons. Combinatorial problem:

$$\binom{N}{q} \text{ possibilities with pool size } N$$

- For gradient-based optimization, the search is over $\mathcal{X}^q$

qNEHVI, qNEI, qNParEGO, …

Batched variants of acquisition functions

- **Sequential greedy selection** chooses a batch one point at a time.

  Example: **q**NEHVI

  - Choose the first element x_1 maximizing

    $$\mathbb{E}_{\hat{f}}\left[\mathrm{HV}\left(\hat{f}(X_{\mathrm{base}}) \cup \{\hat{f}(x)\}\right) - \mathrm{HV}\left(\hat{f}(X_{\mathrm{base}})\right)\right]$$

  - Append it to the pending batch, and choose the second maximizing

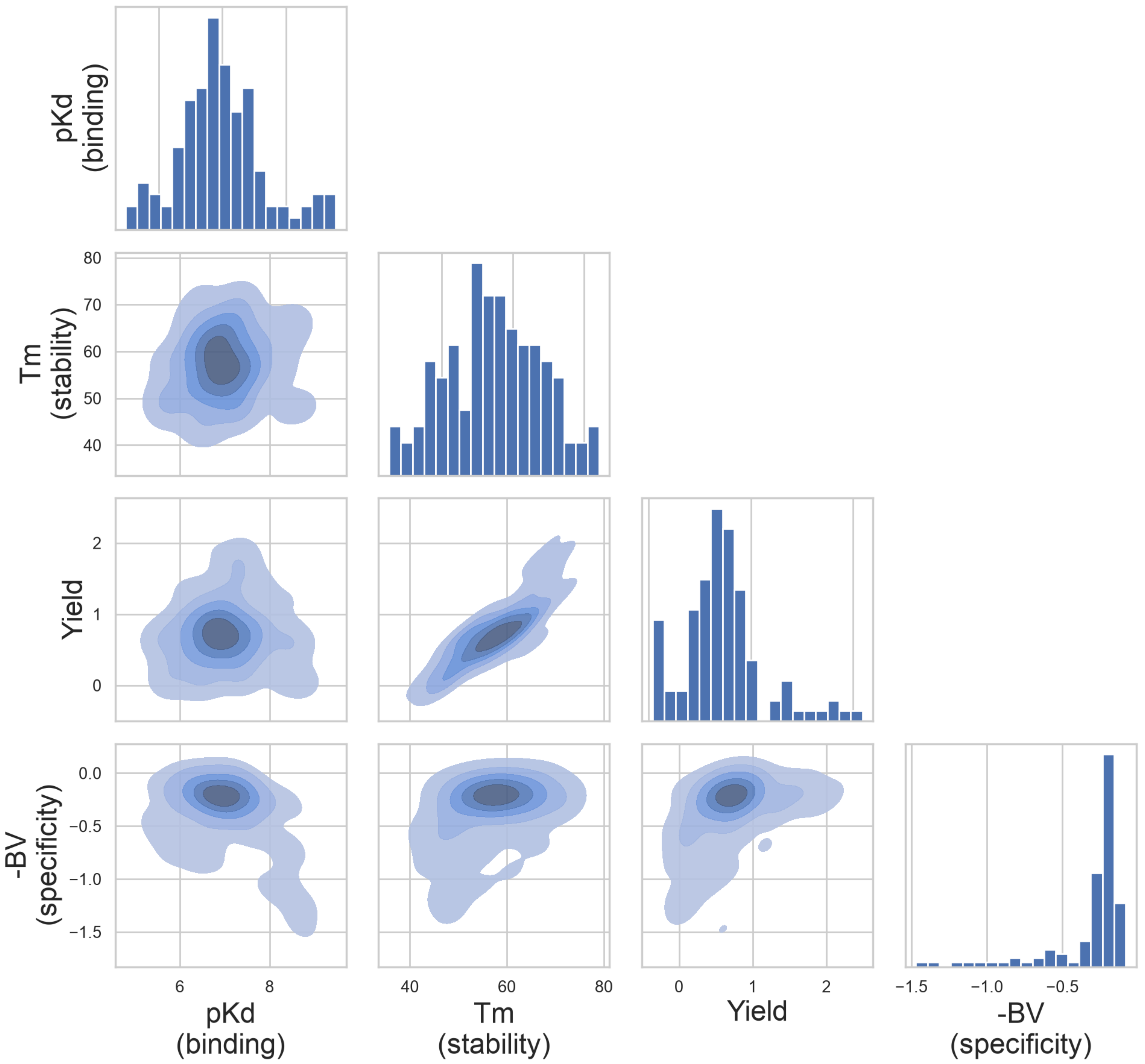
    $$\mathbb{E}_{\hat{f}}\left[\mathrm{HV}\left(\hat{f}(X_{\mathrm{base}}) \cup \{\hat{f}(x), \hat{f}(x_1)\}\right) - \mathrm{HV}\left(\hat{f}(X_{\mathrm{base}}) \cup \{\hat{f}(x_1)\}\right)\right]$$

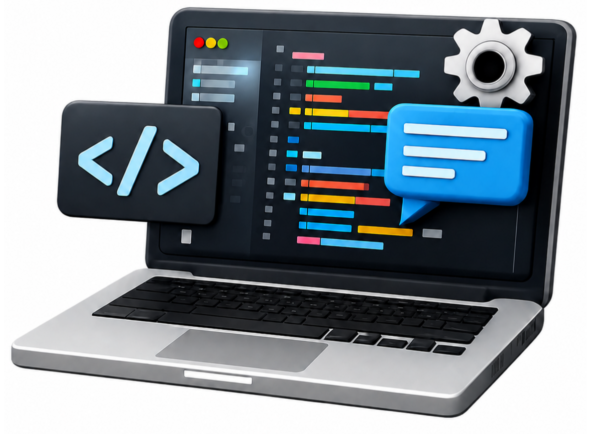
  - ...

Wilson et al. Maximizing acquisition functions for Bayesian optimization. NeurIPS (2018).

Conclusion

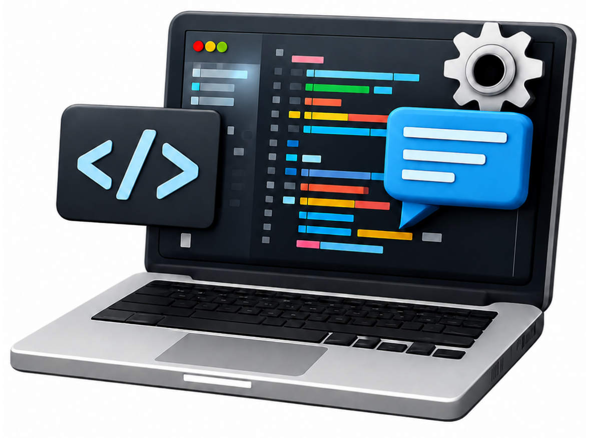
- Bayesian optimization is a sample-efficient method for multi-objective design problems.
- Acquisition function = decision-making engine of MOBO
- We reviewed common acquisition strategies. Choose based on
    - number of objectives
    - shape of Pareto front
    - noise level of measurements
    - batch size
- Multi-objective reasoning applies to many other frameworks, including active learning, experimental design, policy optimization.



Practicum goals 

- Learn the **BoTorch** syntax for concepts covered in the lecture
- Be able to **verify** the implementation of specific configurations for your own application
- Gain **intuition** for the pros/cons of each acquisition strategy by visualizing the acquired designs and metrics

Use of AI coding agents is strongly encouraged for visualization of data and results.

Practicum setup 

- You'll work with your assigned team. Please sit together and come up with a team name!
- Navigate to https://github.com/jiwoncpark/multi-objective-reasoning-lab and follow the instructions in the README to set up your environment.
  - If you aren't familiar with using git/terminal OR you have an Intel Mac, work with someone in your group.

Step 1 — Install uv

Pick the line for your operating system and run it in a terminal:

```
# macOS / Linux
curl -LsSf https://astral.sh/uv/install.sh | sh
```

```
# Windows (PowerShell)
powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
```

Close and reopen your terminal afterward so `uv` is on your `PATH`. Check it:

```
uv --version
```

Step 2 — Clone the repo

```
git clone https://github.com/jiwoncpark/multi-objective-reasoning-lab.git
cd multi-objective-reasoning-lab
```

Step 3 — Build the environment

```
uv sync
```

Schedule

BoTorch

I. Multi-objective optimization: the basics

- Motivation: lead optimization in drug discovery
- Pareto front
- Quality indicator

Setup + Notebook 0: evaluating HV
- 2:30pm
+ 5min review

II. Bayesian optimization

- Surrogate modeling
- Utility function → acquisition function

Notebook 1: full MOBO loop
2:35 - 3:30pm
+ 10min review

III. Multi-objective reasoning

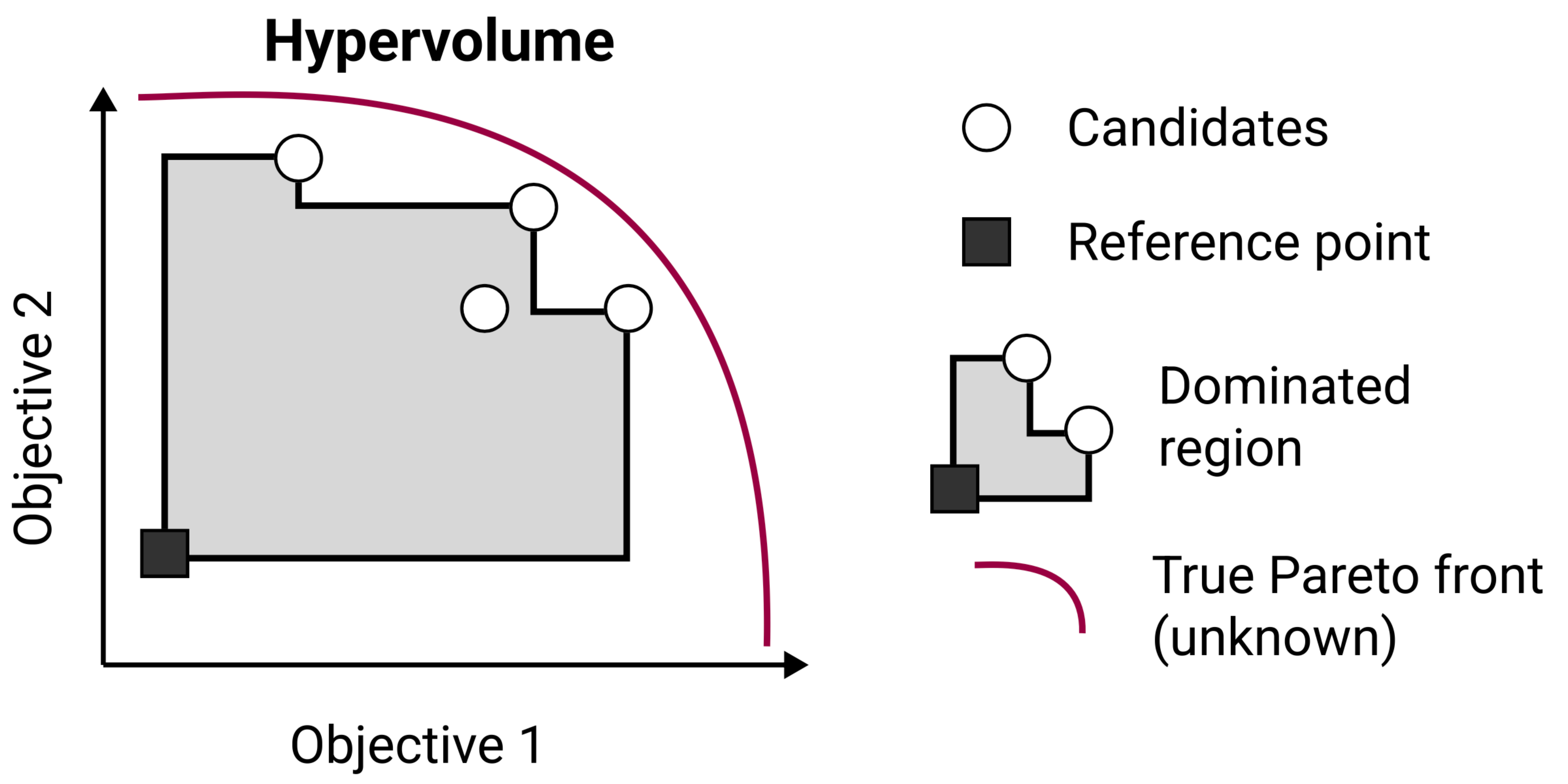
- Common multi-objective acquisition strategies
- Batched/noisy variants

Notebooks 2, 3: team competition on simulated antibody optimization
3:40 - 4:40pm
+ 20min results announcement

Notebook 0 review

BoTorch

```
Y ~ [num_antibodies, num_properties]
hv = metrics.compute_hypervolume(Y, ref_point)
plotting.plot_pareto_front(Y, ref_point=ref_point)
```



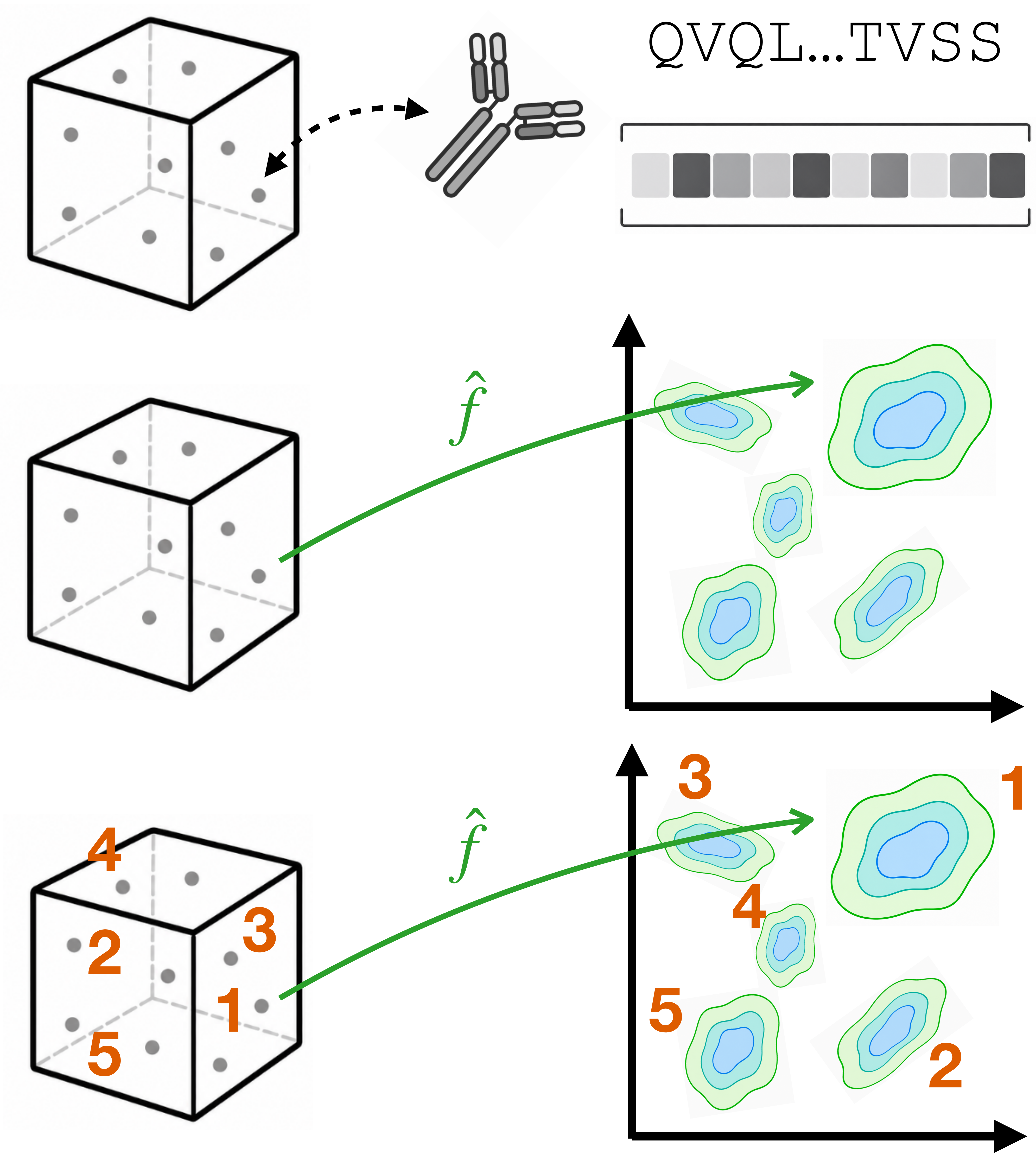
Notebook 1 review

BoTorch

Take the pre-generated pool of 2048 antibody candidates

Predict properties with surrogate model $\hat{f}$

Select batch of size q with noisy acquisition function in a sequential greedy manner



Append measurement to dataset and refit surrogate

Evaluate the batch

Notebooks 2, 3: the team competition

```
LOCKED competition settings
  BATCH_SIZE = 4   (antibodies per round)
  N_ROUNDS   = 6     (rounds)
  N_INITIAL  = 12    (shared starting antibodies)
  budget     = 24 new evaluations total
  pool size  = 2048 candidate sequences
  REF_POINT  = [-0.05, -0.05]
```

Each round uses one strategy card. You may inspect the achieved HV of a previous round to choose the strategy for the next round.

⚠️ Only 6 oracle calls (calls to the objective function) are allowed in total! Think carefully before running each cell of Notebook 3.

By 4:30pm, please email me the results (json file) at