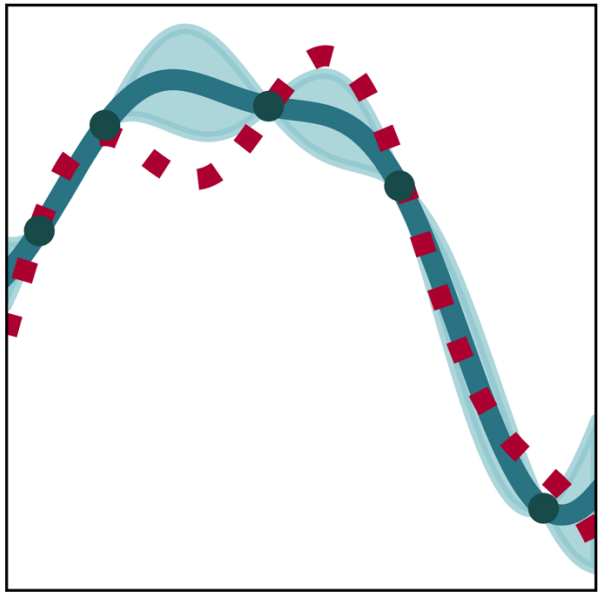
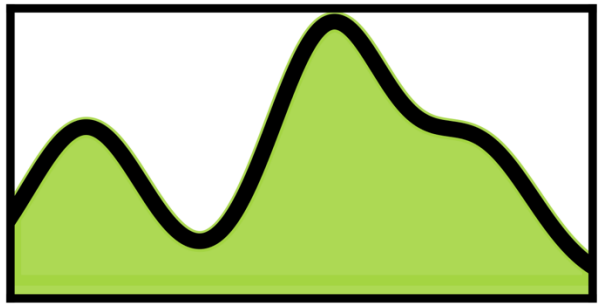
Active Learning and Bayesian Optimization





What is active learning?

Active machine learning refers to a *mode/paradigm* of machine learning in which data is labeled (in iterative fashion) by interactive query

Key Points:

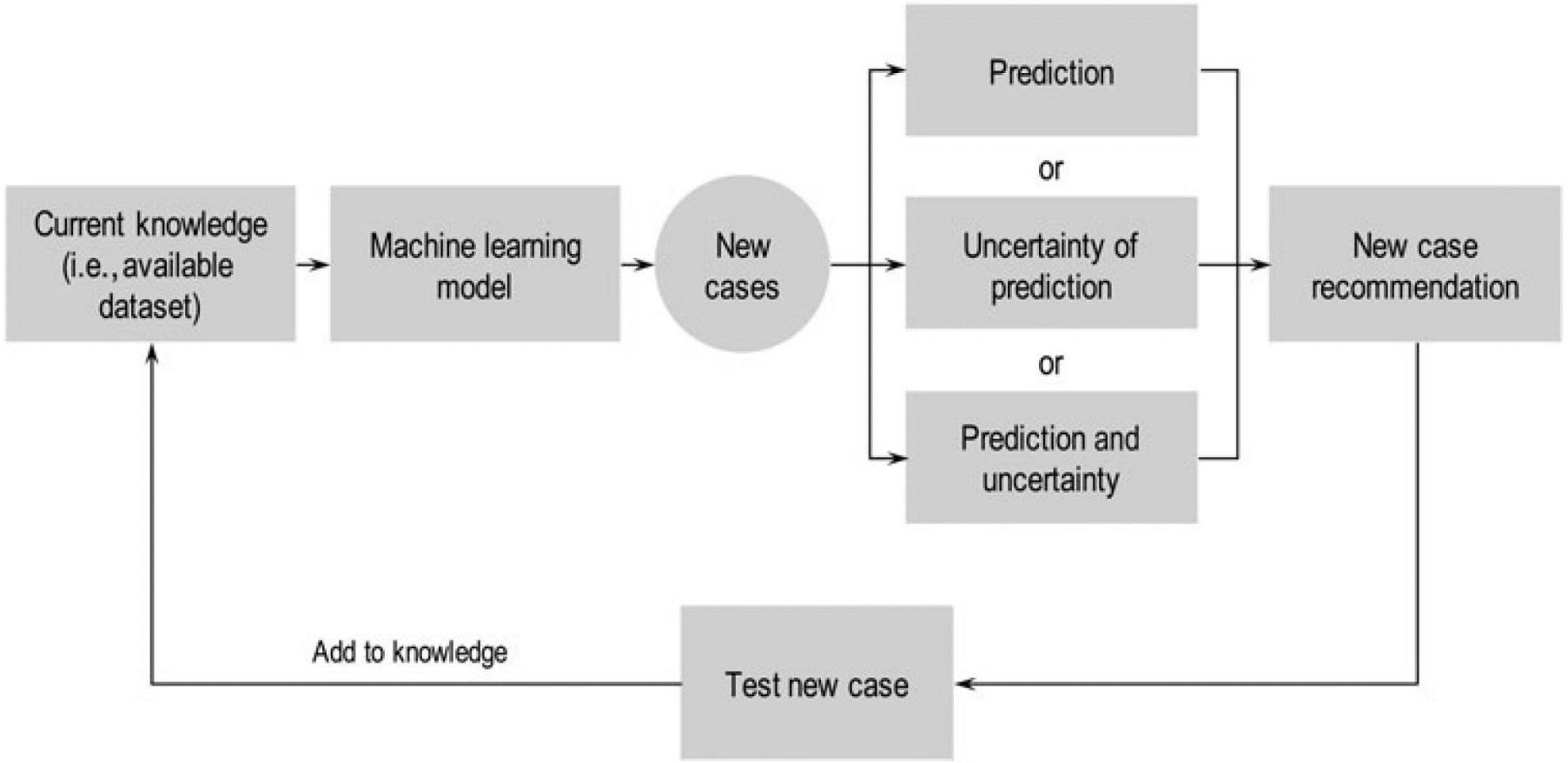
Active learning is commonly considered in situations where labeling data is “expensive” (with respect to some important resource consideration); this usually implies data scarcity

The objective is to be efficient! (create better models with less data, find optimal designs with limited resources)

Active learning may be considered as a conceptual extension of “**Design of Experiments**” method/approach; it is essentially a decision-making formalism that dictates data acquisition

Typical Framework

Active machine learning refers to a *mode/paradigm* of machine learning in which data is labeled (in iterative fashion) by interactive query



Kim et al. *MRS Communications* (2019)

Example Progression with Active Learning Strategy

Active machine learning strategies are roughly defined based on

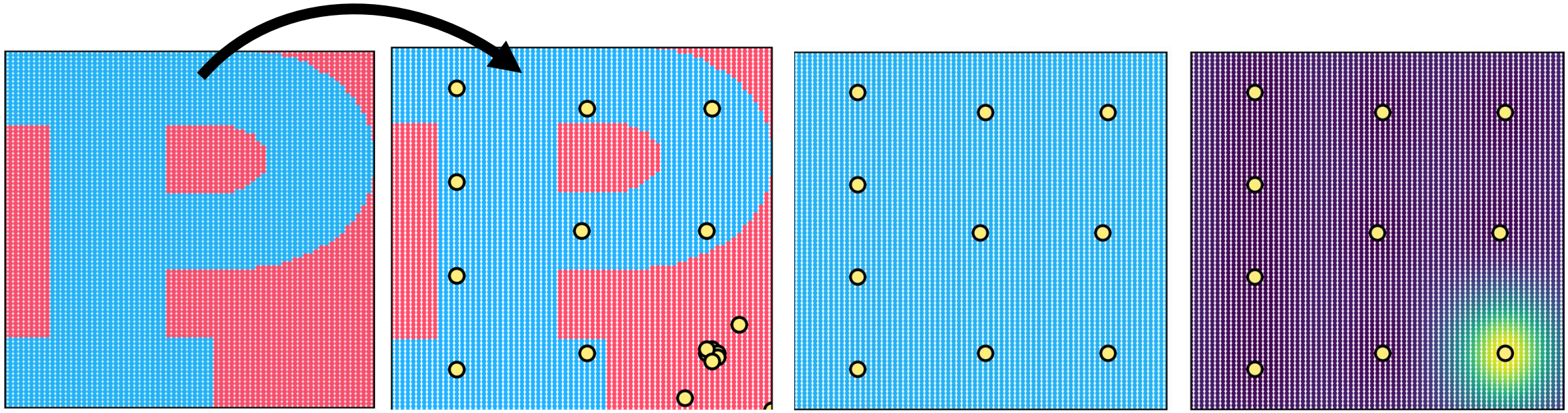
*1) a method of selecting initial points/seed dataset (**sampler**)*

*2) a surrogate model that reflects current knowledge/opinions (**model**)*

*3) a criterion for selecting new samples to query (**acquisition function**)*

The ***sampler*** chooses a "*small*" set of points…

The ***model*** computes *predictions* and *uncertainties* on the domain...



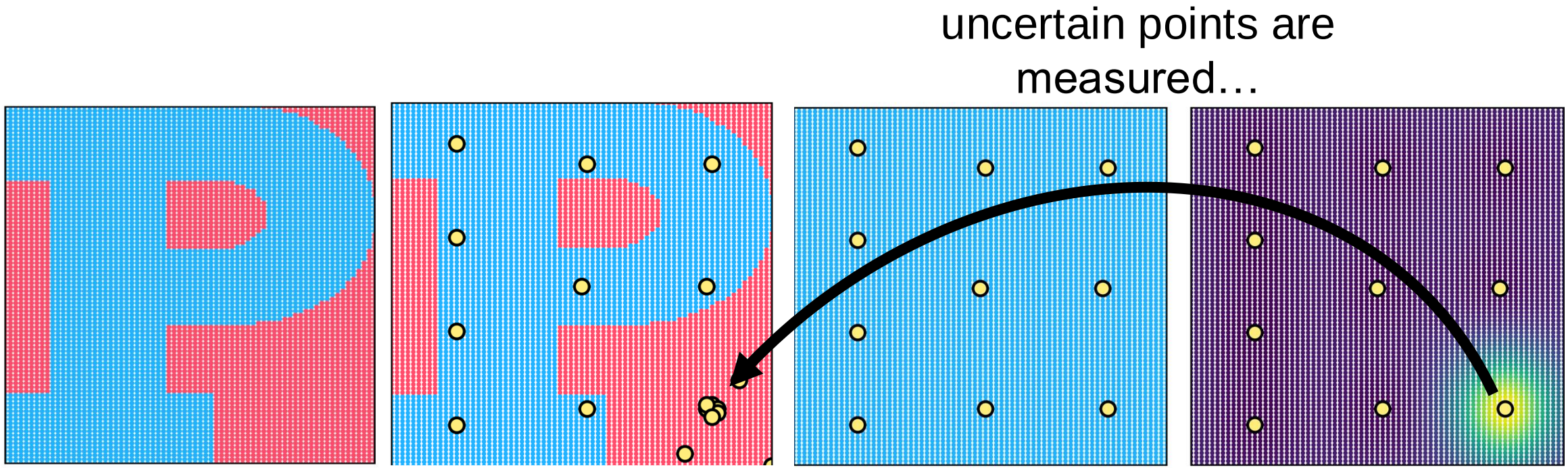
Example Progression with Active Learning Strategy

Active machine learning strategies are roughly defined based on

*1) a method of selecting initial points/seed dataset (**sampler**)*

*2) a surrogate model that reflects current knowledge (**model**)*

*3) a criterion for selecting new samples to query (**acquisition function**)*



Example Progression with Active Learning Strategy

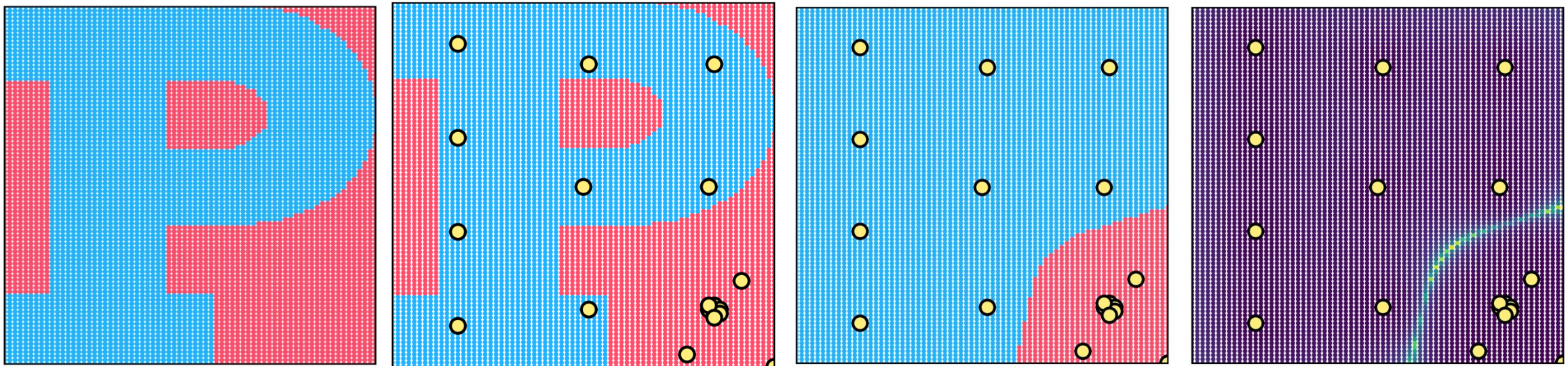
Active machine learning strategies are roughly defined based on

*1) a method of selecting initial points/seed dataset (**sampler**)*

*2) a surrogate model that reflects current knowledge (**model**)*

*3) a criterion for selecting new samples to query (**acquisition function**)*

Resulting in a new set of predictions and uncertainties from the ***model***…



Example Progression with Active Learning Strategy

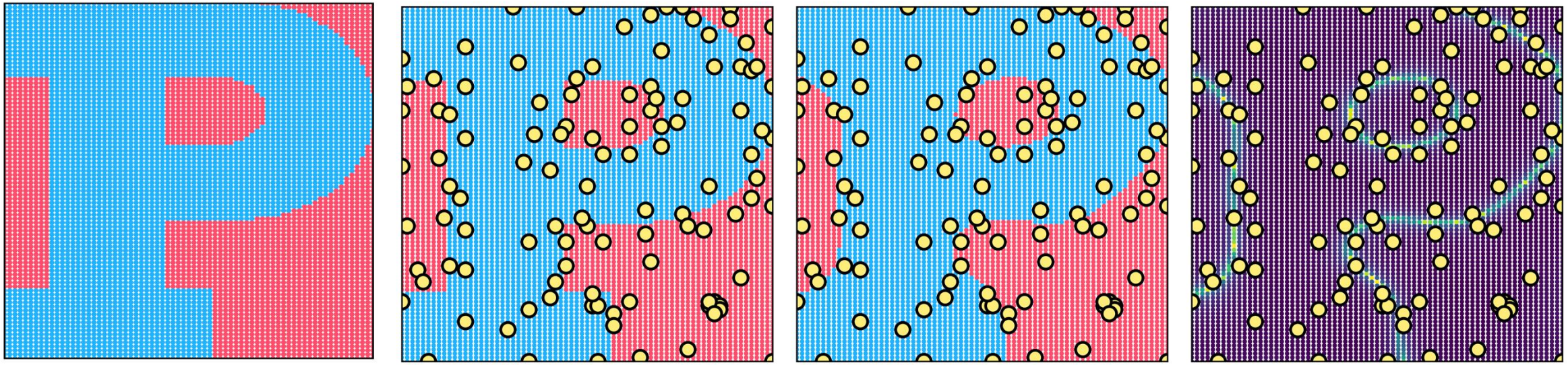
Active machine learning strategies are roughly defined based on

*1) a method of selecting initial points/seed dataset (**sampler**)*

*2) a surrogate model that reflects current knowledge (**model**)*

*3) a criterion for selecting new samples to query (**acquisition function**)*

This process continues until resource budget is exhausted



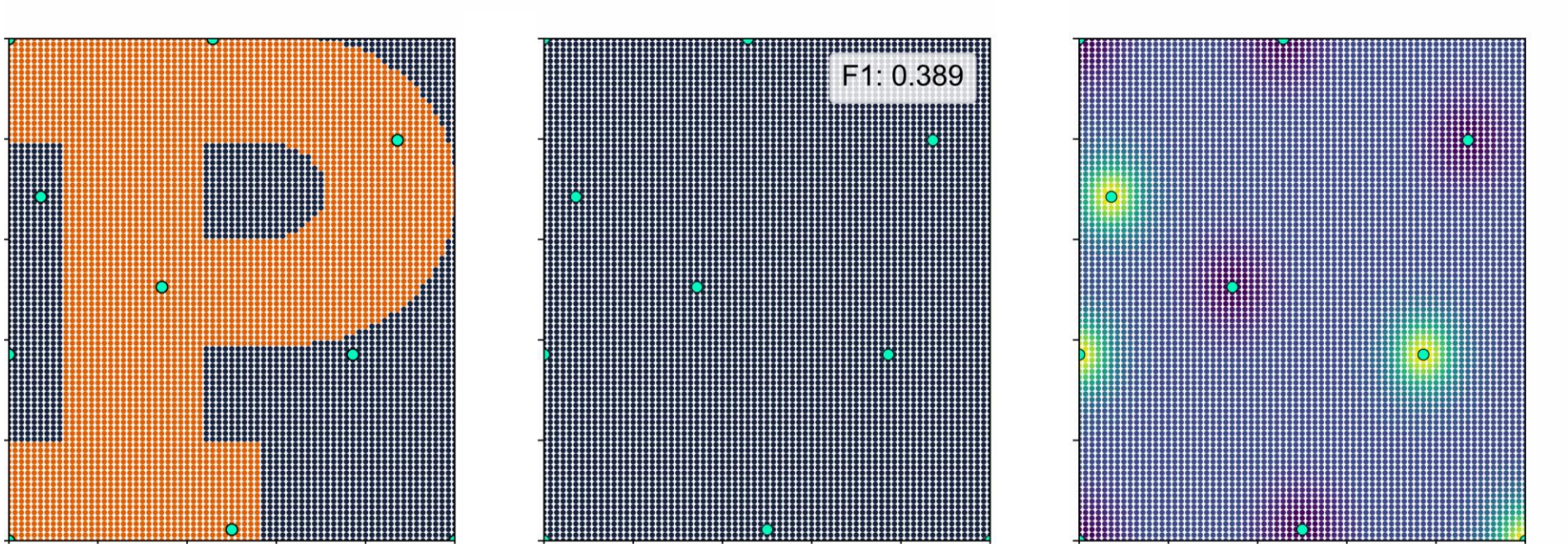
Example Progression with Active Learning Strategy

Active machine learning strategies are roughly defined based on

*1) a method of selecting initial points/seed dataset (**sampler**)*

*2) a surrogate model that reflects current knowledge (**model**)*

*3) a criterion for selecting new samples to query (**acquisition function**)*



Breaking down AL Structure/Considerations

From "Surrogates" Book

Algorithm 6.1: Sequential Design/Active Learning

Assume a flexible surrogate, e.g., a GP model, but with potentially unknown hyperparameterization.

Require a function $f(\cdot)$ providing outputs $y \sim f(x)$ for inputs x, either deterministic or observed with noise; a choice of initial design size n_0 and final size N; and **criterion** $J(x)$ to search for design augmentation.

Then

1. Run a small seed, or bootstrapping experiment
   a. Create an initial seed design X_{n_0} with n_0 runs. Typically X_{n_0} is a model-free choice, e.g., derived from a static LHS or maximin design.
   b. Evaluate $y_i \sim f(x_i)$ under each $x_i^\top$ in the i^{th} row of X_{n_0}, for $i = 1, \ldots, n_0$, obtaining $D_{n_0} = (X_{n_0}, Y_{n_0})$.
   c. Set $n \leftarrow n_0$, indexing iterations of sequential design.
2. Fit the surrogate (and hyperparameters) using D_n, e.g., via MLE.
3. Solve **criterion** $J(x)$ based on the fitted model from Step 2, resulting in a choice of $x_{n+1} \mid D_n$ via $x_{n+1} = \text{argmax}_{x \in \mathcal{X}} J(x) \mid D_n$.
4. Observe the response at the chosen location by running a new simulation, $y_{n+1} \sim f(x_{n+1})$.
5. Update $D_{n+1} = D_n \cup (x_{n+1}, y_{n+1})$; set $n \leftarrow n + 1$ and **repeat** from Step 2 unless $n = N$.

Return the chosen design and function evaluations D_N, along with surrogate fit (i.e., after a final application of Step 2).

How will you choose the next measurement?

This is formalized by defining an acquisition function

How will you prepare the initial set of data for building your first model?

- ***random search***
- ***systematic search***
- ***existing dataset***
- ***Latin hypercube***
- ***Maximin***

How do you represent knowledge or opinions over the domain?

This we generally know how to do

How do you find your next-best measurement?

This is just optimization, but the method and search space can be important

Active Learning vs. Bayesian Optimization

Bayesian optimization is a sequential-based strategy for objective optimization; acquisition of new data is directed by Bayes theorem

$$p(f|\mathcal{D}) \sim p(\mathcal{D}|f)p(f)$$

```
...
# perform the optimization process
for i in range(100):
    # select the next point to sample
    x = opt_acquisition(X, y, model)
    # sample the point
    actual = objective(x)
    # summarize the finding for our own reporting
    est, _ = surrogate(model, [[x]])
    print('>x=%.3f, f()=%3f, actual=%.3f' % (x, est, actual))
    # add the data to the dataset
    X = vstack((X, [[x]]))
    y = vstack((y, [[actual]]))
    # update the model
    model.fit(X, y)
def acquisition(X, Xsamples, model):
    # calculate the best surrogate score found so far
    yhat, _ = surrogate(model, X)
    best = max(yhat)
    # calculate mean and stdev via surrogate function
    mu, std = surrogate(model, Xsamples)
    mu = mu[:, 0]
    # calculate the probability of improvement
    probs = norm.cdf((mu - best) / (std+1E-9))
    return probs
```

Key elements:

- you need a *probabilistic <u>model</u>* of an objective function; this is a surrogate for actual (expensive) function evaluations
- you choose your next query based on this probabilistic model and an acquisition function
- you want to leverage "belief" about the objective function to direct search to areas with a high likelihood of giving useful information

Active Learning vs. Bayesian Optimization

Active Learning

- **Goal:** Improve model accuracy or reduce uncertainty with minimal labelled data.
- **Key Feature:** Selects data points from regions of high uncertainty or disagreement.
- **Use Cases:** Classification, regression, data annotation tasks.
- **Selection Strategy:** Focuses on diverse, representative, or uncertain samples to refine the model.
- **Outcome:** A better-trained model with fewer training samples.

Bayesian Optimization

- **Goal:** Identify the global optimum of an unknown objective function efficiently.
- **Key Feature:** Balances exploration (uncertain regions) and exploitation (high-value regions).
- **Use Cases:** Hyperparameter tuning, process optimization, experimental design.
- **Selection Strategy:** Uses acquisition functions (e.g., Expected Improvement, UCB) to target potential optima.
- **Outcome:** Optimal parameters or conditions with fewer evaluations.

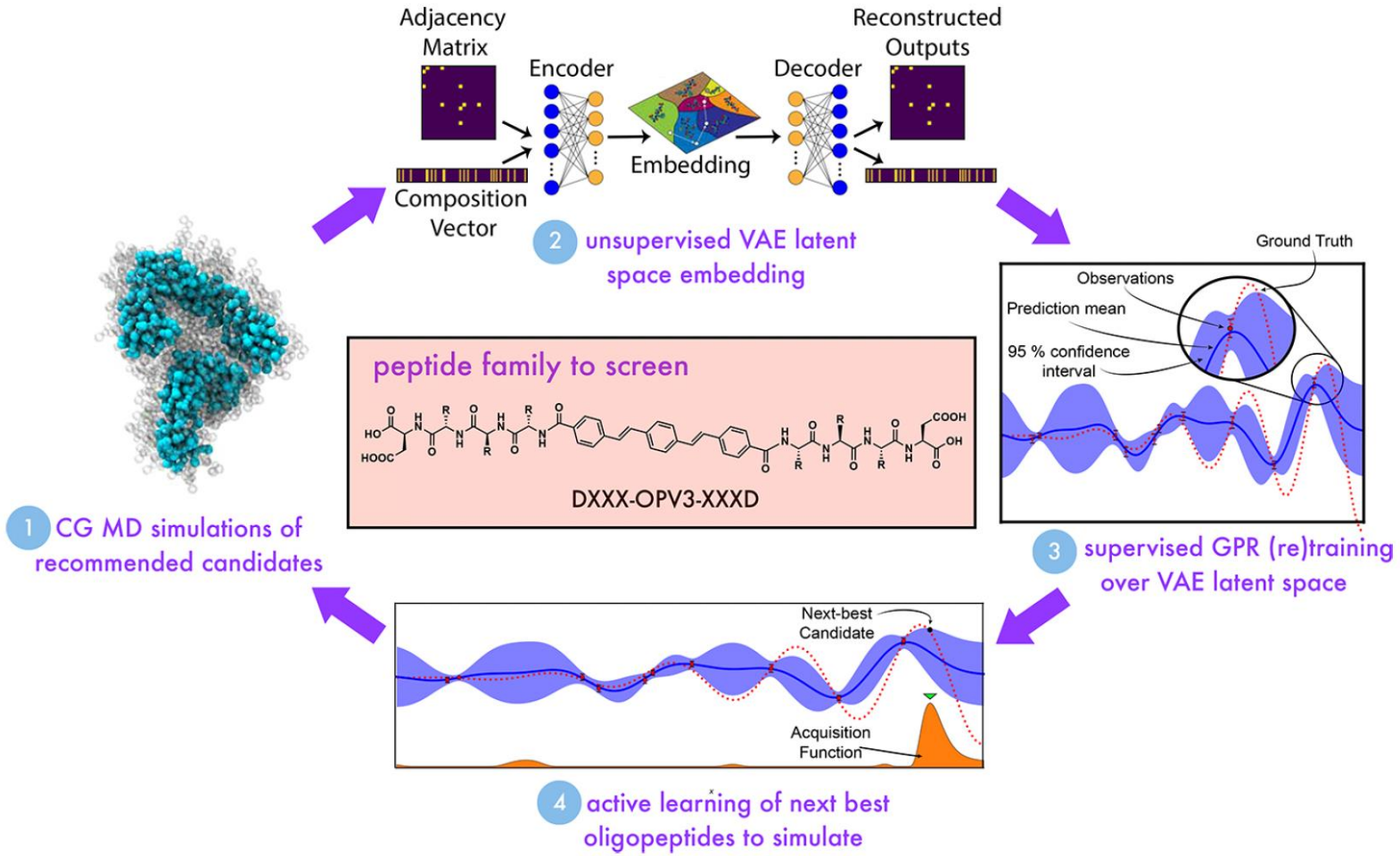
Key Distinctions

- **Objective:** Active learning refines the model; Bayesian optimization seeks the optimal solution.
- **Scope:** Active learning is broader and model-centric; Bayesian optimization is goal-specific.
- **Selection Focus:** Active learning queries uncertain samples; Bayesian optimization targets optimality.

These points provide a concise comparison suitable for a presentation.

Example Framework

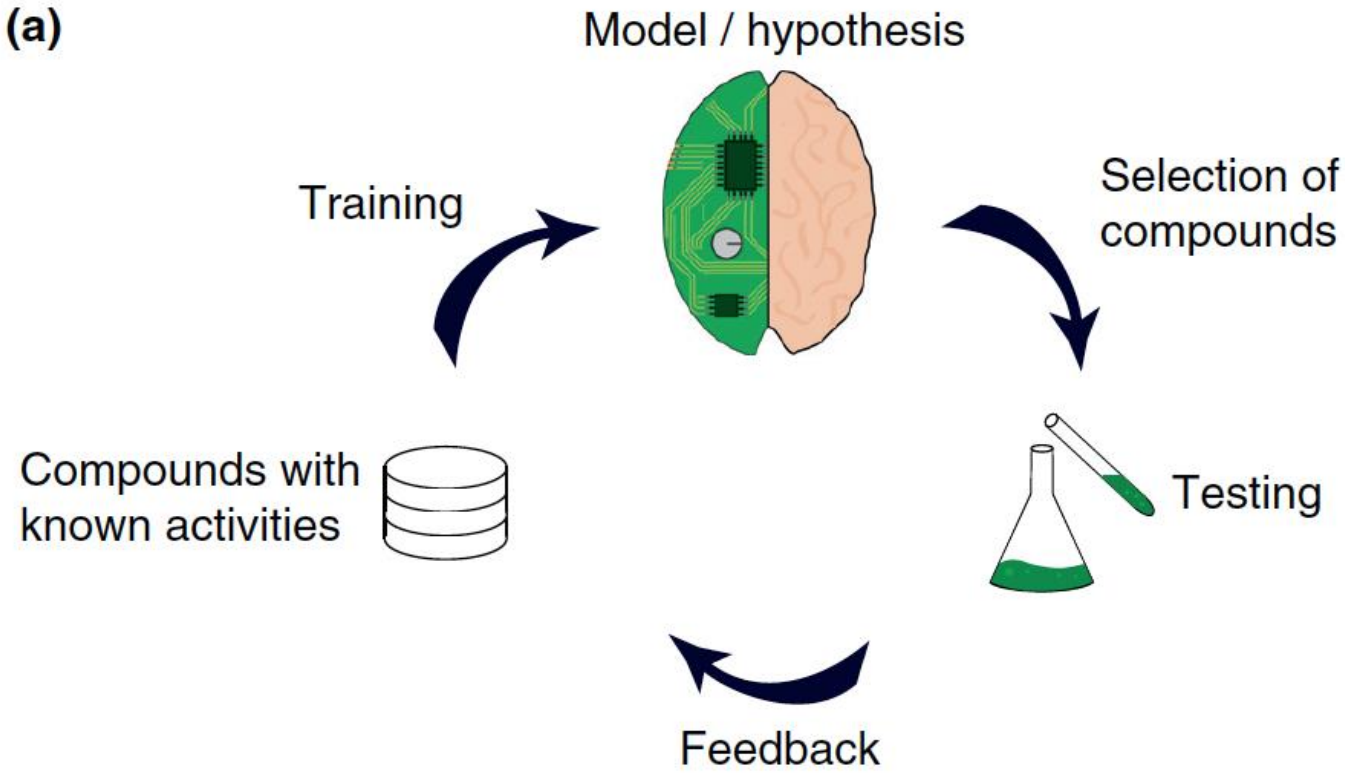
Active machine learning refers to a *mode/paradigm* of machine learning in which data is labeled (in iterative fashion) by interactive query



Shmilovich et al. *JPCB* **2020**

Example Framework

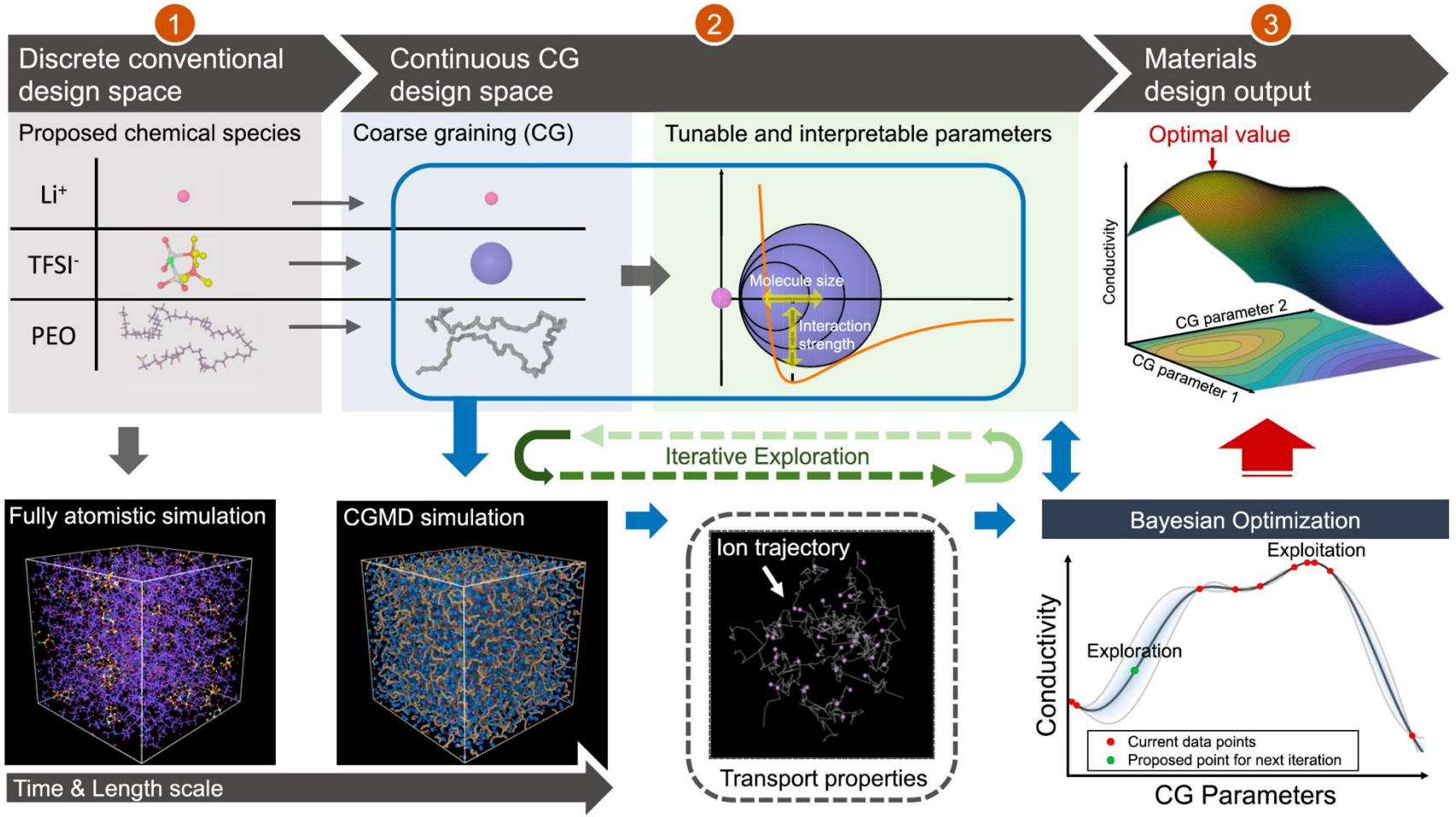
Active machine learning refers to a *mode/paradigm* of machine learning in which data is labeled (in iterative fashion) by interactive query



Reker and Schneider. *Drug Discovery Today* **2015**

Example Framework

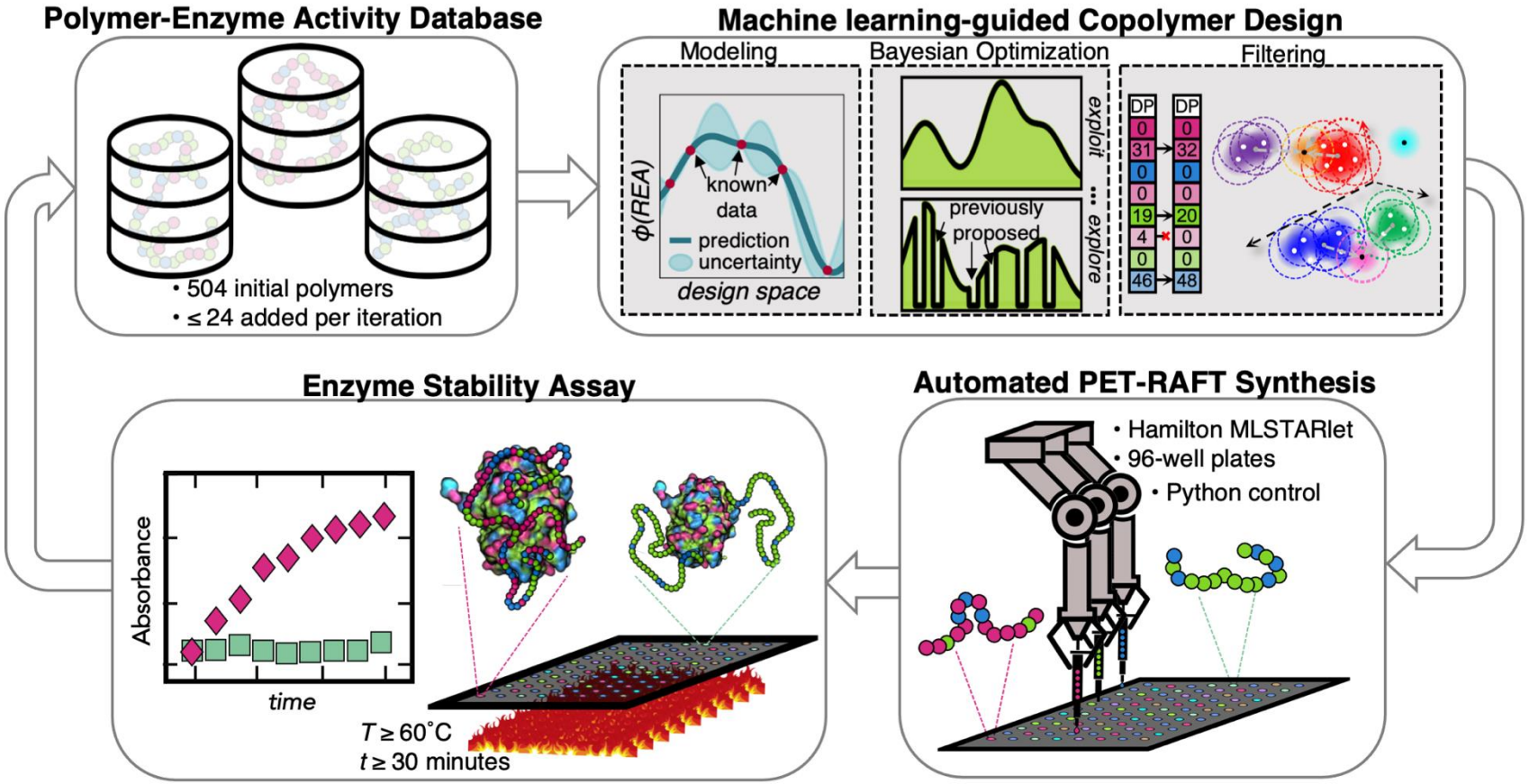
Active machine learning refers to a *mode/paradigm* of machine learning in which data is labeled (in iterative fashion) by interactive query



Wang et al. *Chem. Mater.* **2020**

Example Framework

Active machine learning refers to a *mode/paradigm* of machine learning in which data is labeled (in iterative fashion) by interactive query



Tamasi, Patel, Borca, Kosuri, Mugnier, Upadhya, Murthy, Webb, Gormley. *Adv. Mat. In press.*

Acquisition Functions

A critical component to the active learning paradigm is the acquisition function

Improvement-based acquisition functions

$$f(\boldsymbol{x}^+); \boldsymbol{x}^+ = \underset{\boldsymbol{x}_i \in \mathcal{D}}{\arg\max}\, f(\boldsymbol{x}_i)$$

incumbent

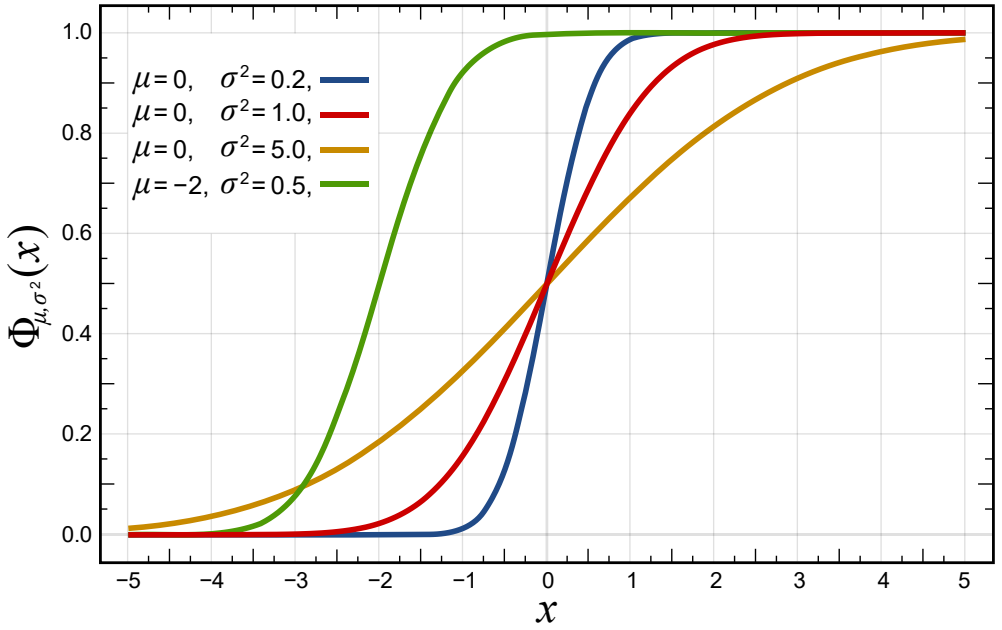
probability of improvement

"maximize the probability of improvement over the ***incumbent****"*

$$\mathrm{PI}(\boldsymbol{x}) = \Phi\left(\frac{\mu(\boldsymbol{x}) - f(\boldsymbol{x}^+) - \xi}{\sigma(\boldsymbol{x})}\right)$$

standard normal cumulative distribution

$$\Phi(z) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{z}{\sqrt{2}}\right)\right]$$



$\xi \geq 0$ *generally and often not used*

Acquisition Functions

A critical component to the active learning paradigm is the acquisition function;

Improvement-based acquisition functions

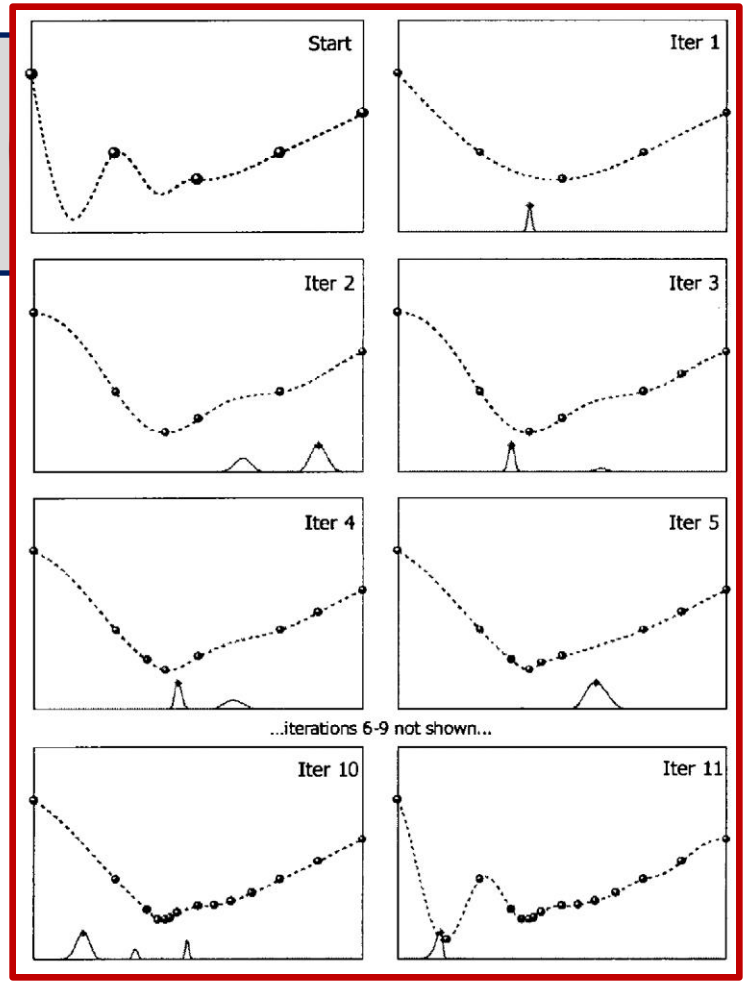
$$f(\boldsymbol{x}^+); \boldsymbol{x}^+ = \underset{\boldsymbol{x}_i \in \mathcal{D}}{\arg\max}\, f(\boldsymbol{x}_i)$$

incumbent

probability of improvement

*"maximize the probability of improvement over the **incumbent**"*

$$\mathrm{PI}(\boldsymbol{x}) = \Phi$$



A Taxonomy of Global Optimization Methods Based on Response Surfaces

DONALD R. JONES
General Motors Corporation, Mail Code 480-305-200, 6440 East 12 Mile Road, Warren, MI 48090, USA (e-mail: don.jones@gm.com)

The performance of Method 4 in Figure 15 is truly impressive. It would be quite natural if the reader, like so many others, became enthusiastic about this approach. But if there is a single lesson to be taken away from this paper, it is that nothing in this response-surface area is so simple. There always seems to be a counterexample. In this case, the difficulty is that Method 4 is extremely sensitive to the choice of the target T. If the desired improvement is too small, the search will be highly local and will only move on to search globally after searching nearly exhaustively around the current best point. On the other hand, if T is set too high, the search will be excessively global, and the algorithm will be slow to fine-tune any promising solutions. This sensitivity to the setting of the target is illustrated in

Acquisition Functions

A critical component to the active learning paradigm is the acquisition function

Improvement-based acquisition functions

$$f(\boldsymbol{x}^+); \boldsymbol{x}^+ = \arg\max_{\boldsymbol{x}_i \in \mathcal{D}} f(\boldsymbol{x}_i)$$

incumbent

expected improvement

"maximize the probability of improvement but also consider the magnitude of that improvement"

standard normal probability density distribution

$$\mathrm{EI}(\boldsymbol{x}) = \sigma(\boldsymbol{x}) \left[Z\Phi(Z) + \phi(Z) \right]$$

$$Z = \frac{\mu(\boldsymbol{x}) - f(\boldsymbol{x}^+) - \xi}{\sigma(\boldsymbol{x})}$$

ξ *roughly controls the exploration-exploitation tradeoff*

- →0, more towards exploitation
- →large, more towards exploration

used by Jones *et al.* [2001]. Lizotte's experiments suggest that setting $\xi = 0.01$ (scaled by the signal variance if necessary) works well in almost all cases, and interestingly, setting a cooling schedule for ξ to encourage exploration early and exploitation later does *not* work well empirically, contrary to intuition (though Lizotte did find that a cooling schedule for ξ might slightly improve performance on short runs ($t < 30$) of PI optimization).

Acquisition Functions

A critical component to the active learning paradigm is the acquisition function

Confidence-bound acquisition functions

$$f(\boldsymbol{x}^+); \boldsymbol{x}^+ = \underset{\boldsymbol{x}_i \in \mathcal{D}}{\arg\max}\, f(\boldsymbol{x}_i)$$

incumbent

lower confidence bound
(typically chosen when minimizing)

$$\mathrm{LCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) - \xi\sigma(\boldsymbol{x})$$

upper confidence bound
(typically chosen when maximizing)

$$\mathrm{UCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \xi\sigma(\boldsymbol{x})$$

variation for fixed number of iterations

$$r(\boldsymbol{x}) = f(\boldsymbol{x}^*) - f(\boldsymbol{x})$$

$$\min \sum_t^T r(\boldsymbol{x}_t)$$

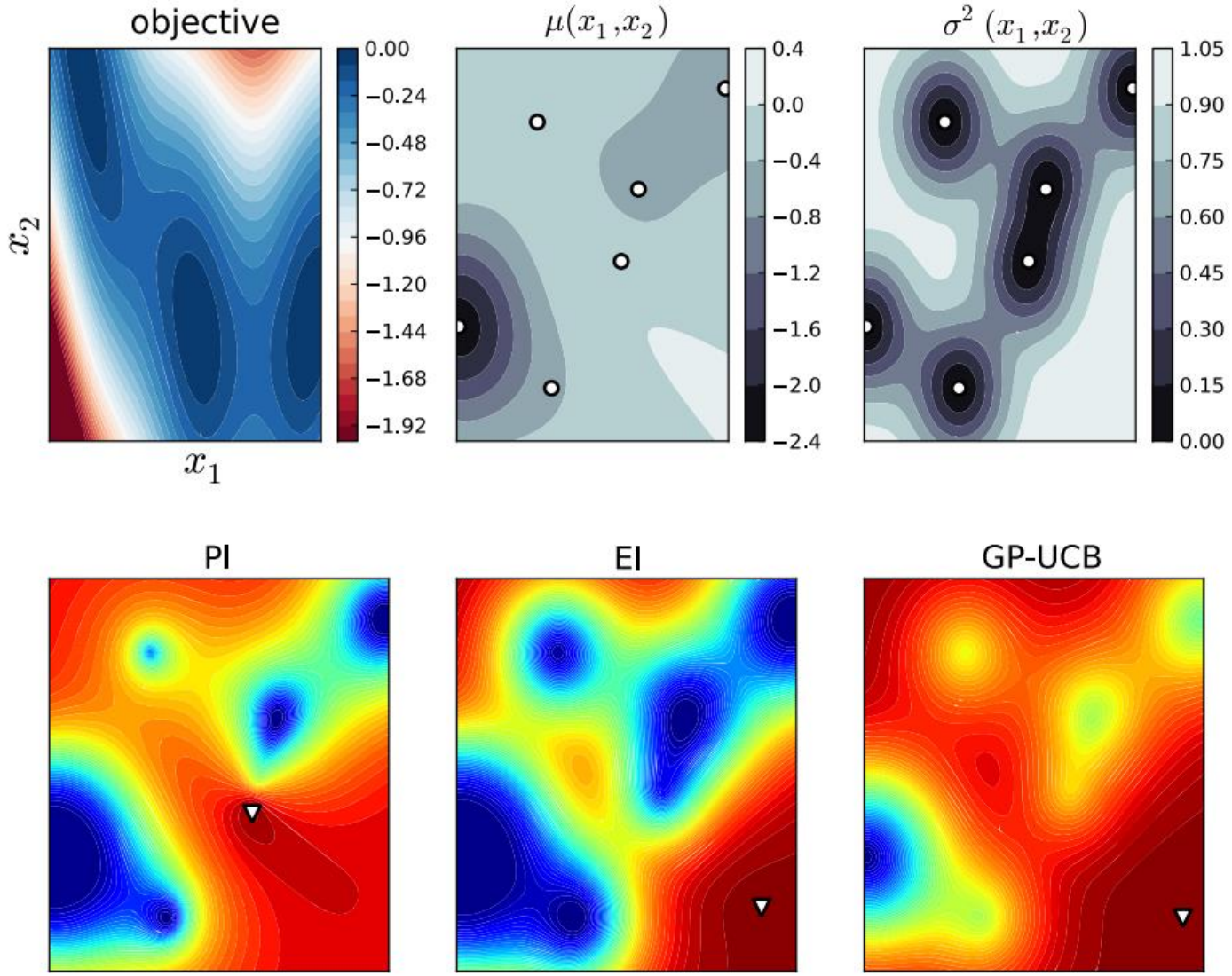
related to ***multi-armed bandit*** problem

$$\text{GP-UCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \sqrt{\nu\tau_t}\sigma(\boldsymbol{x})$$

select schedule on constants that approximate limiting case of ***no regret*** (guaranteed solution/lower bound on convergence rate)

Acquisition Functions

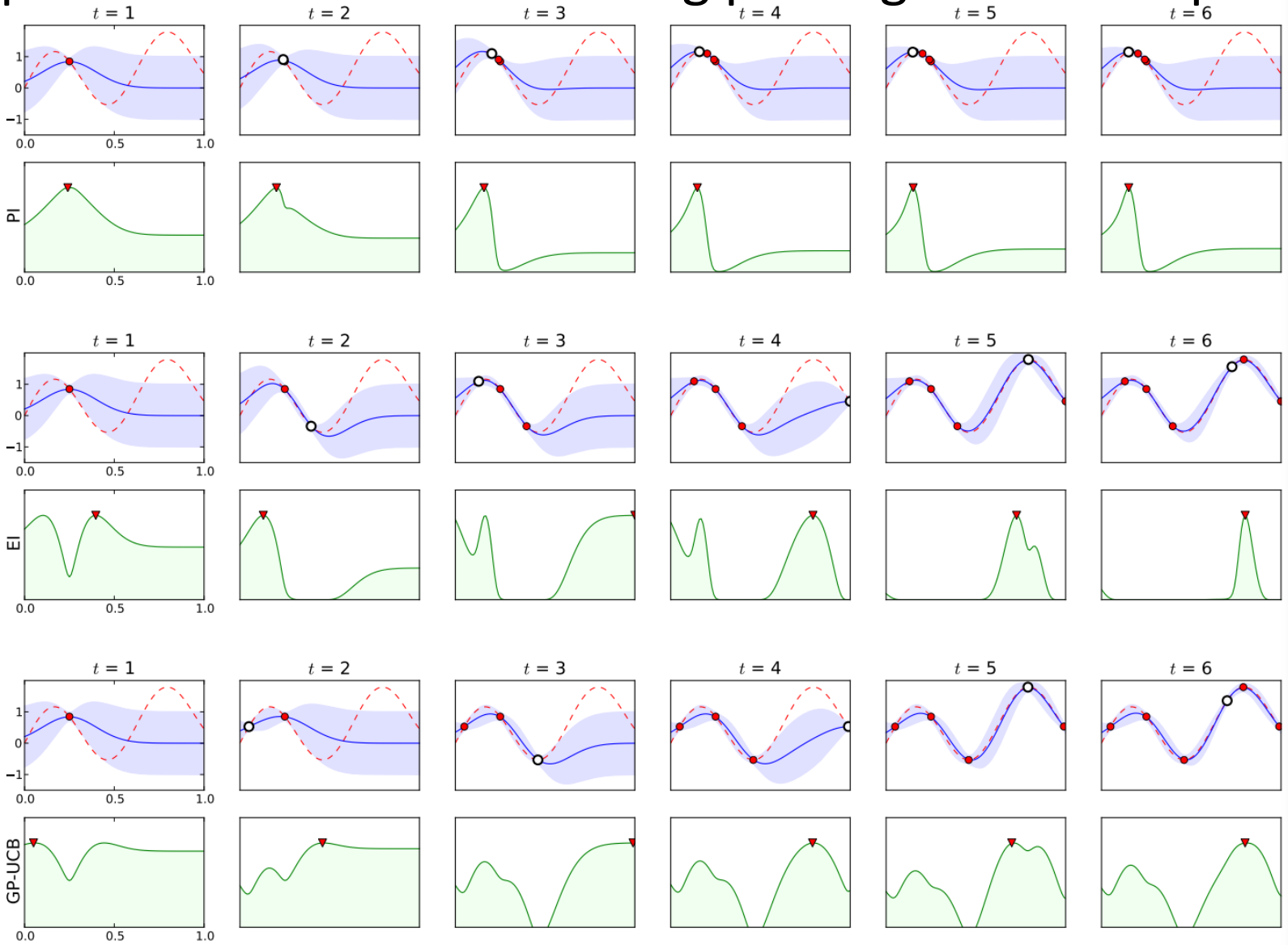
A critical component to the active learning paradigm is the acquisition function



Brochu et al.,
ArXiv:1012.2599

Acquisition Functions

A critical component to the active learning paradigm is the acquisition function

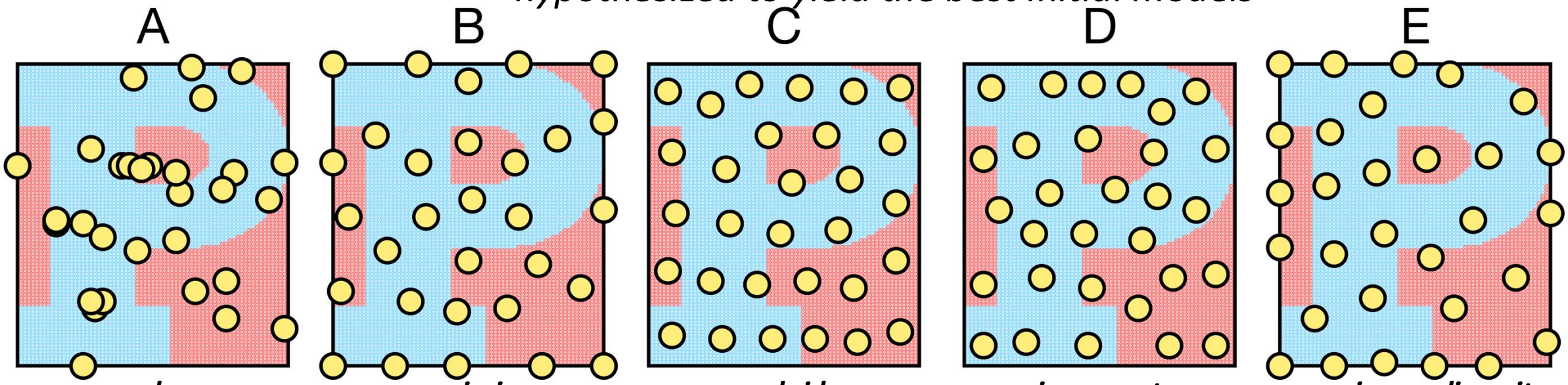


Brochu et al.,
ArXiv:1012.2599

Data Seeding: comparisons

In situations where you are starting *ex nihilo*, there are several options for seeding your active learning process.

Space-filling methods *are geared towards generating dataset diversity, which is hypothesized to yield the best initial models*



random

maximin
"sequential furthest point sampling"

medoids
"k-medoids clustering"

maximum entropy

$$\arg\min_{\mathbf{x}^*}\left[\frac{1}{m+1}\sum_{j=1}^{m}\kappa(\mathbf{z}_j-\mathbf{x}^*)-\frac{1}{N}\sum_{i=1}^{N}\kappa(\mathbf{x}_i-\mathbf{x}^*)\right]$$

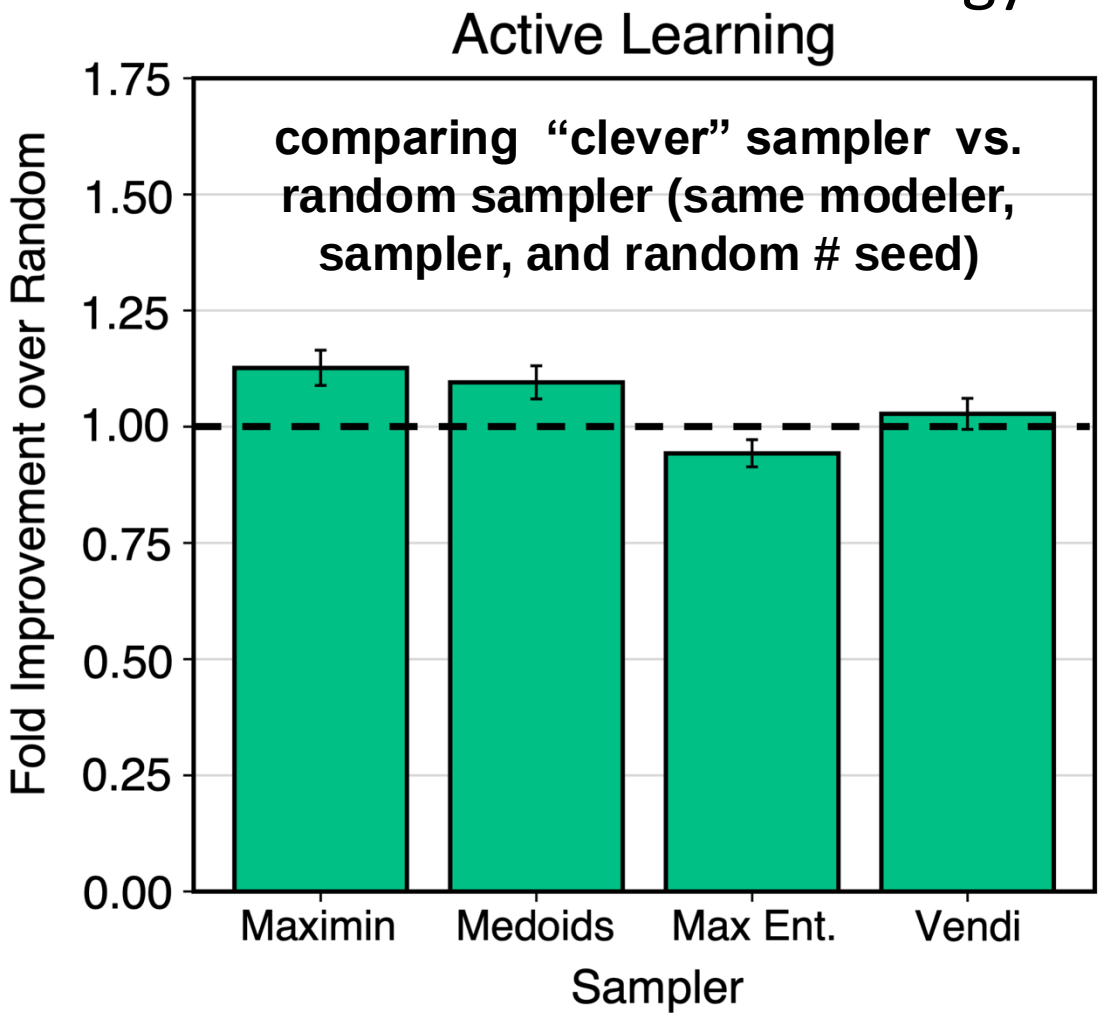
$$\kappa(\mathbf{x}_i,\mathbf{x}_j)=\frac{1}{(2\pi\sigma^2)^{d/2}}\exp\left(-\frac{1}{2\sigma^2}||\mathbf{x}_i-\mathbf{x}_j||^2\right)$$

maximum diversity
(Vendi score: "entropy of norm. Eigenvalues of a Gram matrix"

Gallagher, Webb. "Data Efficiency of Classification Strategies for Chemical and Materials Design" *ChemRxiv.* **2024.**

Data Seeding: comparisons

The relative performance of seeding/sampler strategies depends on the rest of the strategy and data investment



Gallagher, Webb. "Data Efficiency of Classification Strategies for Chemical and Materials Design" *ChemRxiv.* **2024.**

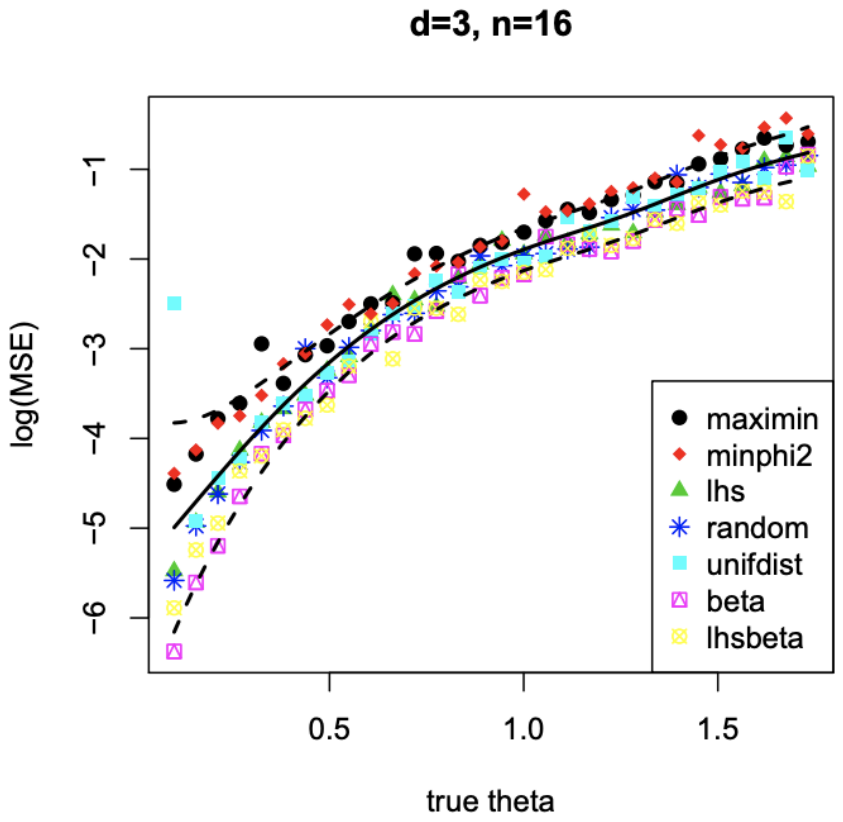
Data Seeding: comparisons

Distance-distributed design for Gaussian process surrogates

Boya Zhang* D. Austin Cole† Robert B. Gramacy†

signs, via maximin distance, Latin hypercube, etc. However, it is easy to demonstrate empirically that such designs disappoint when the model hyperparameterization is unknown, and must be estimated from data observed at the chosen design sites. This is true even when the performance metric is prediction-based, or when the target of interest is inherently or eventually sequential in nature, such as in blackbox (Bayesian) optimization. Here we expose such inefficiencies, showing that in many cases a purely random design is superior to higher-powered alternatives. We then propose a fam-

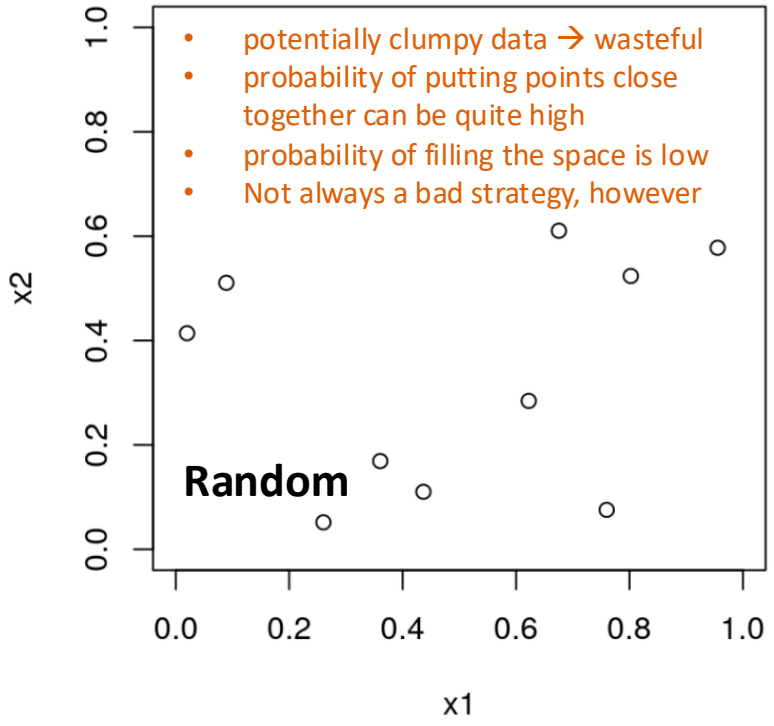
ArXiv1812.02794

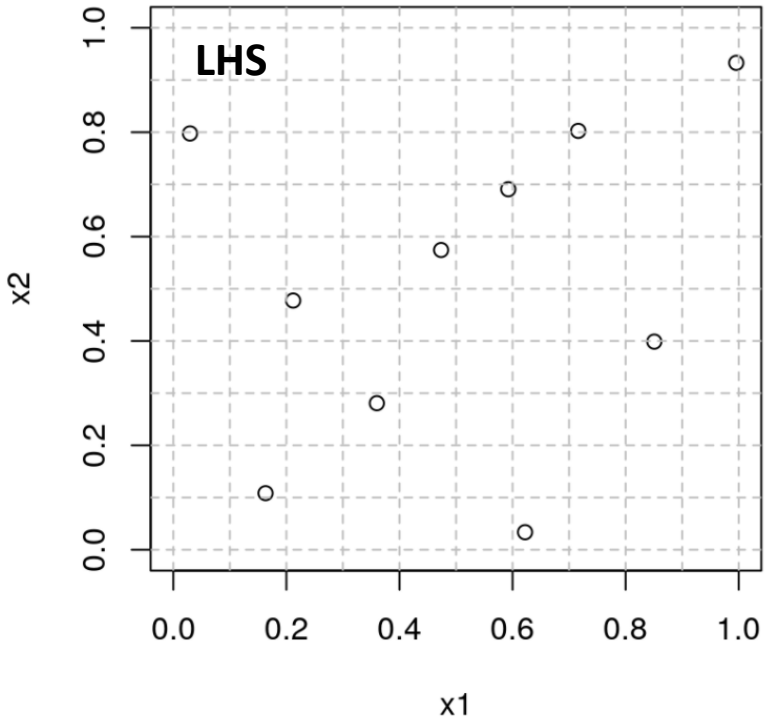


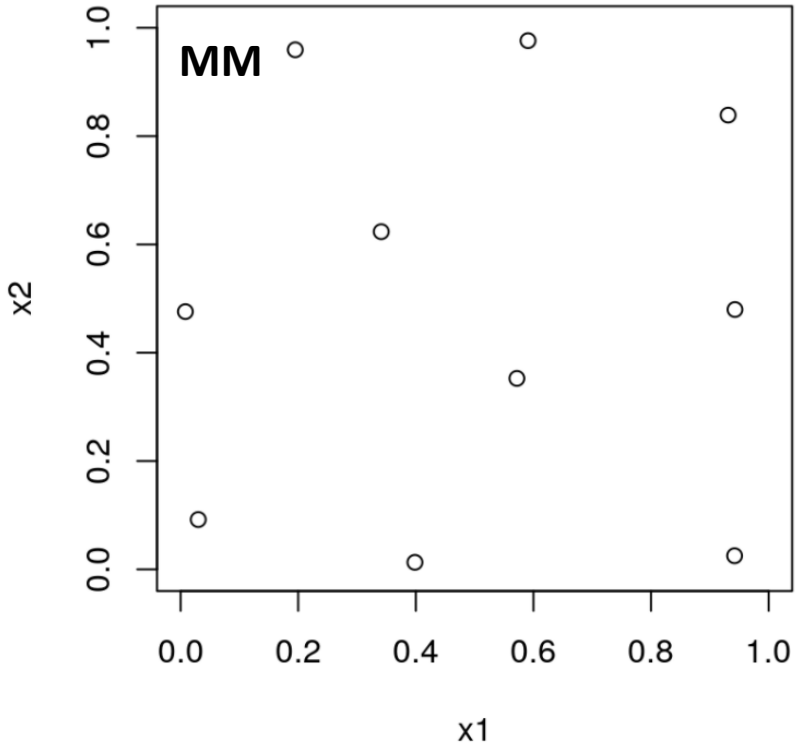
Data Seeding: comparisons

Distance-distributed design
for Gaussian process surrogates

Boya Zhang* D. Austin Cole† Robert B. Gramacy†

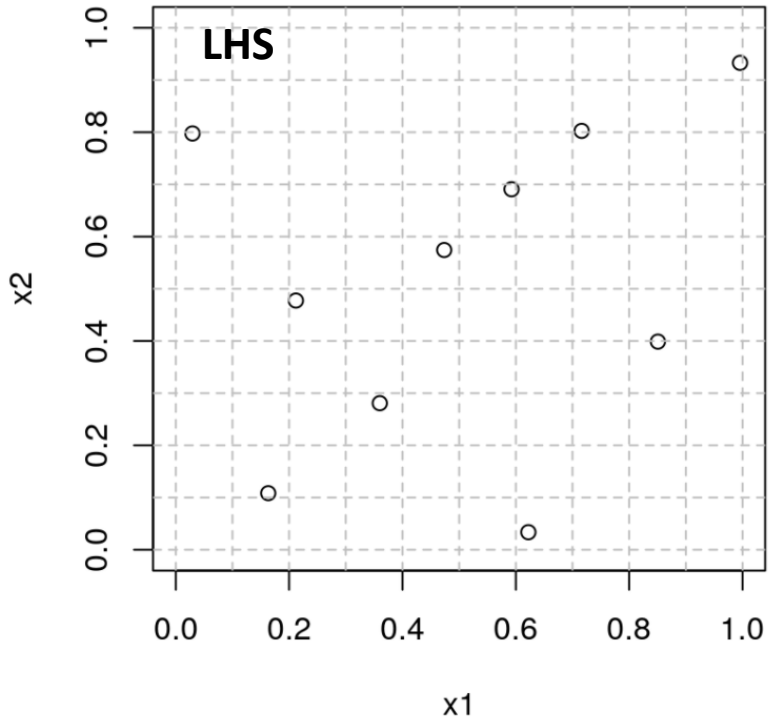


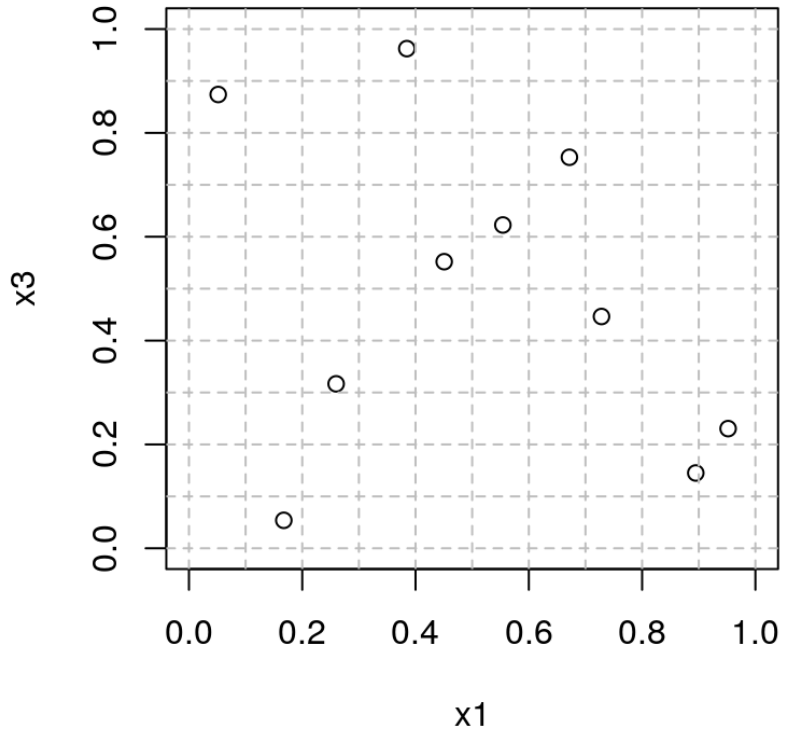


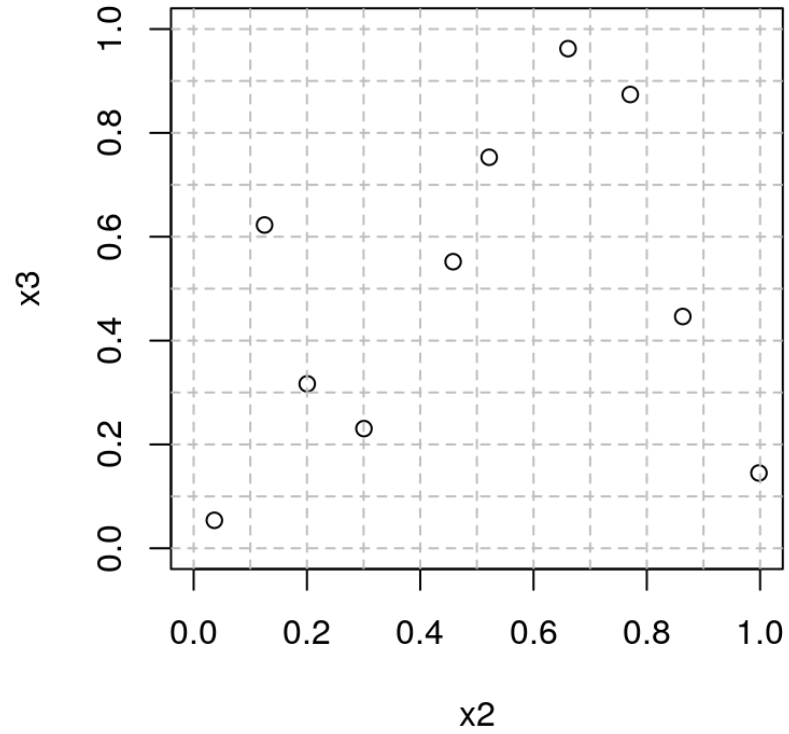


Data Seeding: Latin Hypercube Sampling

Latin Hypercube Sampling **-** a popular method that targets uniformity in the distribution of points

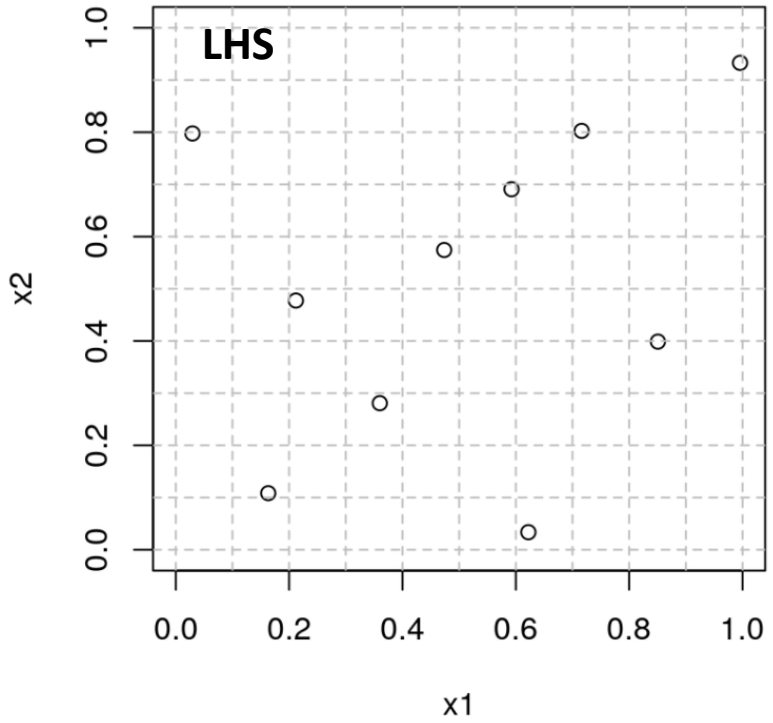






Data Seeding: Latin Hypercube Sampling

Latin Hypercube Sampling **-** a popular method that targets uniformity in the distribution of points

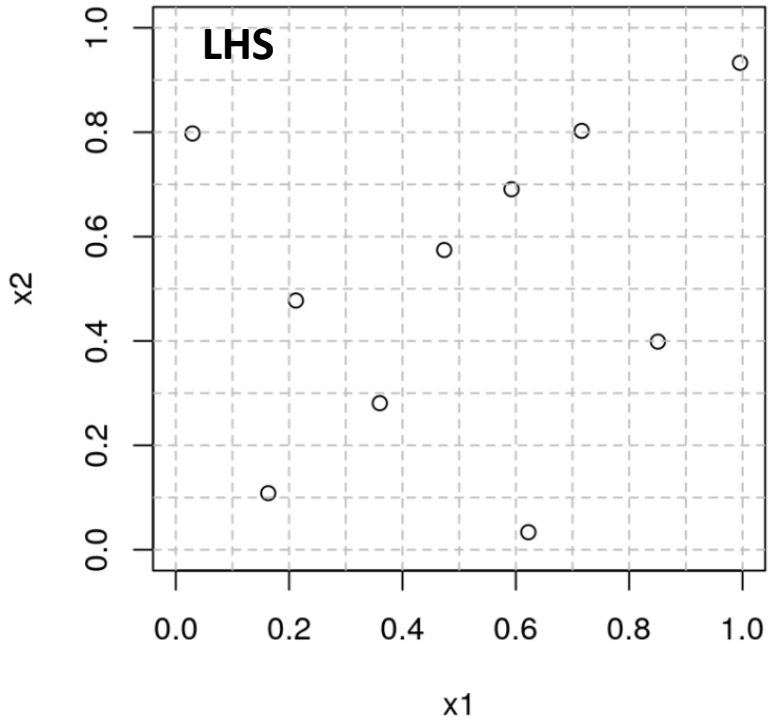


- *enjoys benefits of being random*
- *promotes space-filling by "guaranteeing" a certain amount of spread*
- *grid up space and divvy up points such that there is just one point in each segment per dimension*

By User:Schutz. The stained glass was designed by Maria McClafferty and installed in 1989. - Self-photographed, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=779504

Data Seeding: Latin Hypercube Sampling

Latin Hypercube Sampling **-** a popular method that targets uniformity in the distribution of points



- *enjoys benefits of being random*
- *promotes space-filling by "guaranteeing" a certain amount of spread*
- *grid up space and divvy up points such that there is just one point in each segment per dimension*
- *LHS promotes uniformity of a certain kind but may not be robust to other pattern formation*
- *Bad code execution can lead to aliasing*
- *There are pernicious designs that satisfy LHS but are overall undesirable*

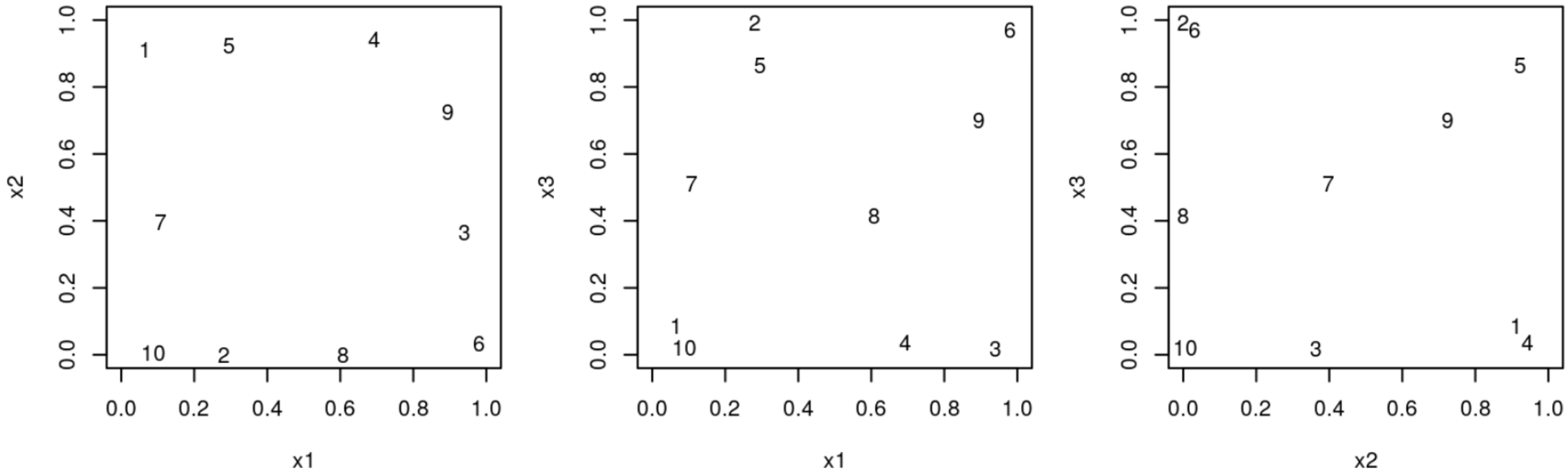
By User:Schutz. The stained glass was designed by Maria McClafferty and installed in 1989. - Self-photographed, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=779504

Data Seeding: Maximin Sampling

$$\mathcal{X}_n = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$$

$$\mathcal{X}_n = \arg\max_{\mathcal{X}_n} \min\{d(\boldsymbol{x}_i, \boldsymbol{x}_k) : i \neq k = 1, \ldots, n\}$$

maximizes minimum distance between pairs



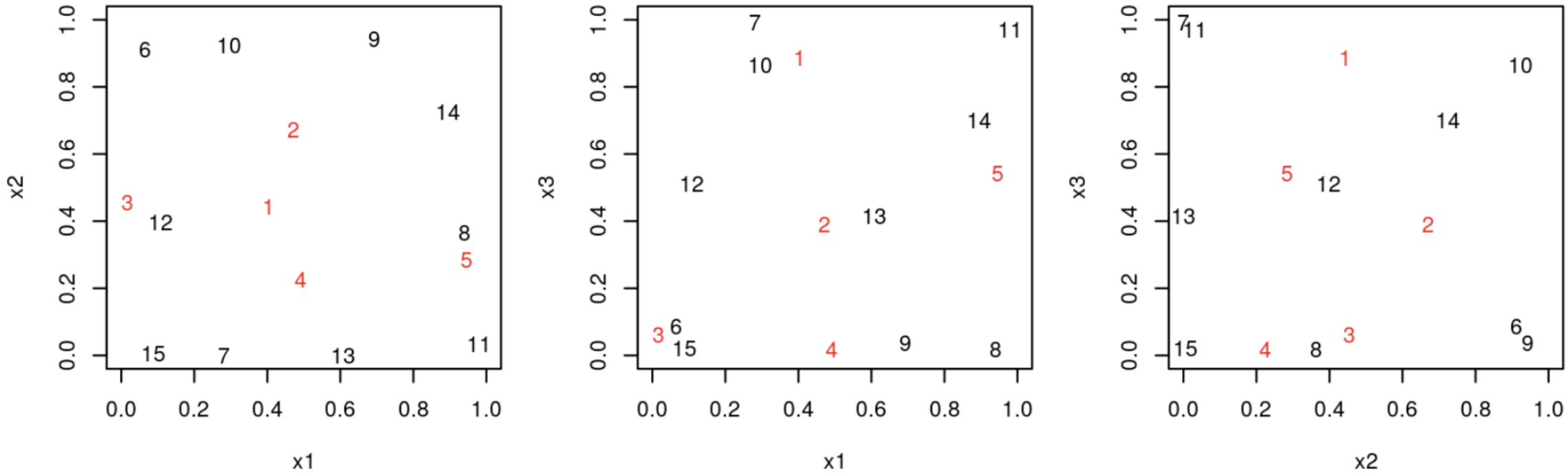
- *Every "Design" features typically O(n²) operations*
- *Requires fast distance calculations*
- *Tends to push points out to boundaries*
- *Limits variability in inputs*

Data Seeding: Maximin Sampling

$$\mathcal{X}_n = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$$

$$\mathcal{X}_n = \arg\max_{\mathcal{X}_n} \min\{d(\boldsymbol{x}_i, \boldsymbol{x}_k) : i \neq k = 1, \ldots, n\}$$

maximizes minimum distance between pairs



- *Every "Design" features typically O(n²) operations*
- *Requires fast distance calculations*
- *Tends to push points out to boundaries*
- *Limits variability in inputs*
- *Permits successive application (can be useful)*