Towards autonomous chemistry with Bayesian optimization

Riley J. Hickman (he/him/his)

Digital Molecular Design Studio (CHM 09-860)
Carnegie Mellon University
04/12/2022 - 4:40-7:30pm ET

the
matter lab

VECTOR
INSTITUTE

My background

the matter lab

Riley J. Hickman

Education

BSc in chemistry and physics (2018)
Carleton University (Ottawa, ON, Canada)
Supervisor: Prof. Tao Zeng

PhD in chemistry (2023*)
University of Toronto (Toronto, ON, Canada)
Supervisor: Prof. Alán Aspuru-Guzik

Lecture outline

the matter lab

Lecture component

- Overview of self-driving labs/autonomous science
- Gaussian processes (surrogate models)
- Bayesian optimization
  - Parameter spaces
  - Acquisition functions
  - Popular software packages
- Advanced topics in Bayesian optimization
  - Multi-objective optimization
  - Robust optimization
  - Multi-fidelity optimization
- Prototypes of self-driving labs
- Outlook

Programming component

Code-along tutorial session in Google colaboratory

1. Build and test a vanilla Bayesian optimizer using *BoTorch* + *Olympus* (continuous, categorical and mixed parameter spaces)

2. Extend our optimizer to multi-task setting

Link

Link

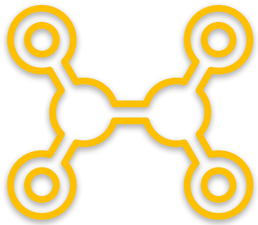
Automating the scientific method

the matter lab

Machine-learnt estimate of the questions, given observations

Refined hypothesis about some scientific question

Prepare a new system for testing



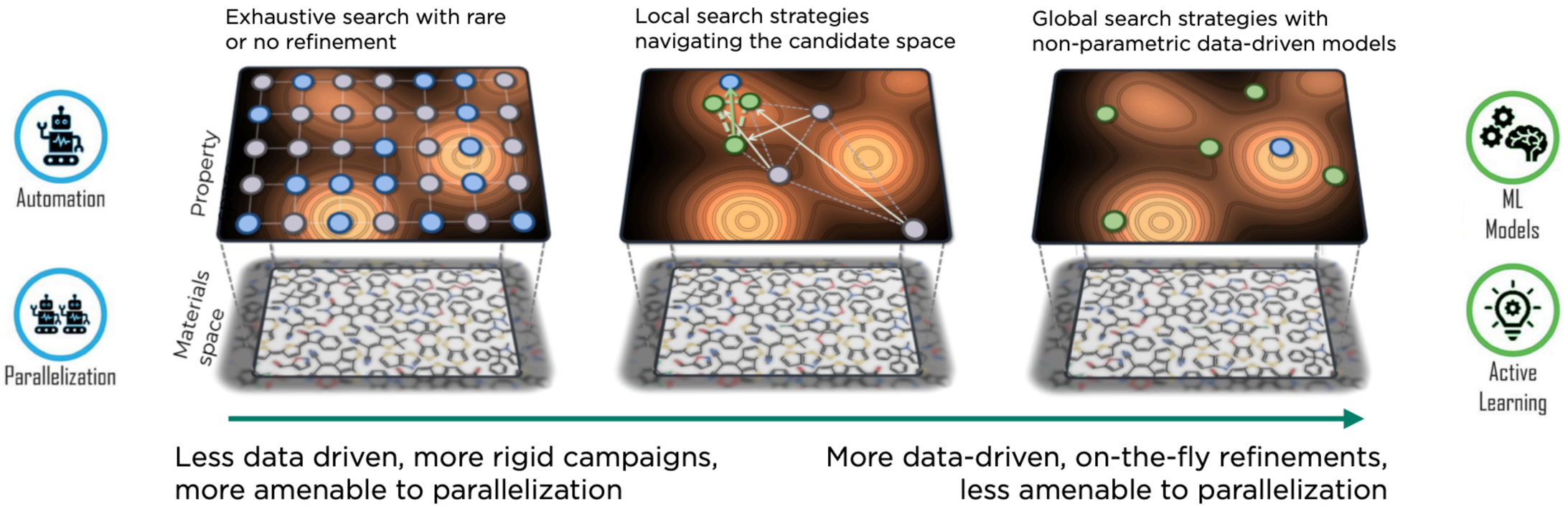
Automated synthesis, device fabrication, simulation preparation

Automated characterisation of system, data post-processing

Measurement, recording, and analysis

Automating the scientific method

the matter lab



Factorial grid search, DOE, random search, low discrepancy sequences

Gradient-based, evolutionary strategies, heuristic strategies

Model-based optimization, reinforcement learning, active learning

Automating the scientific method

the matter lab

Bayes Theorm

$$P(B|A) = \frac{P(A|B)\,P(A)}{P(B)}$$

$P(B\,|\,A)$: Posterior
$P(A\,|\,B)$: Likelihood
$P(A)$: Prior
$P(B)$: Marginal

Useful for understanding probability, statistical modelling and inference, machine learning, uncertainty quantification

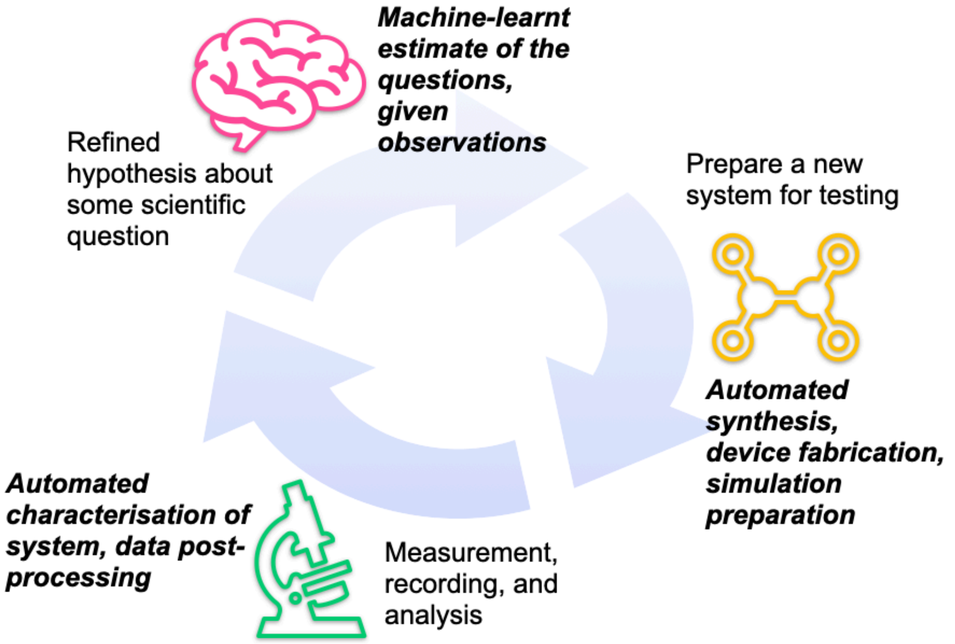
Reformulating in terms of the scientific method …

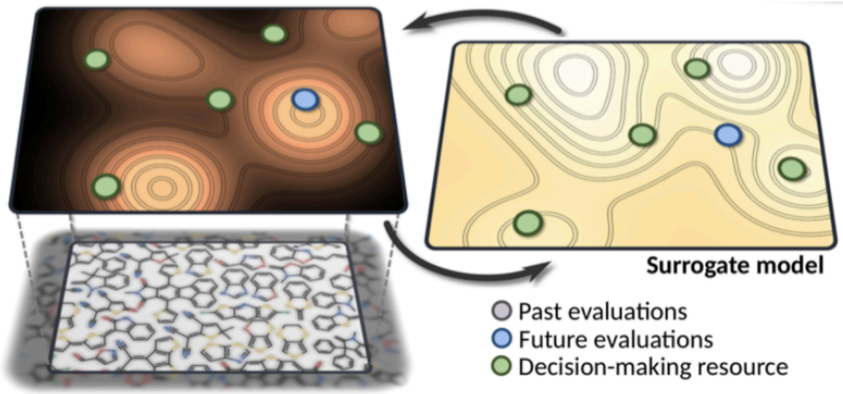
$$P(\text{Hypothesis}|\text{Evidence}) = \frac{P(\text{Evidence}|\text{Hypothesis})\,P(\text{Hypothesis})}{P(\text{Evidence})}$$

Posterior inferred by our ML "*surrogate model*"

So, what type of probabilistic ML models can we use for our surrogate model?
Many!

- **Gaussian processes**
- Tree-based models
- Neural networks
- Kernel smoothing





What is a Gaussian process?

"A GP is any distribution over functions such that any finite set of function values has a joint Gaussian distribution"

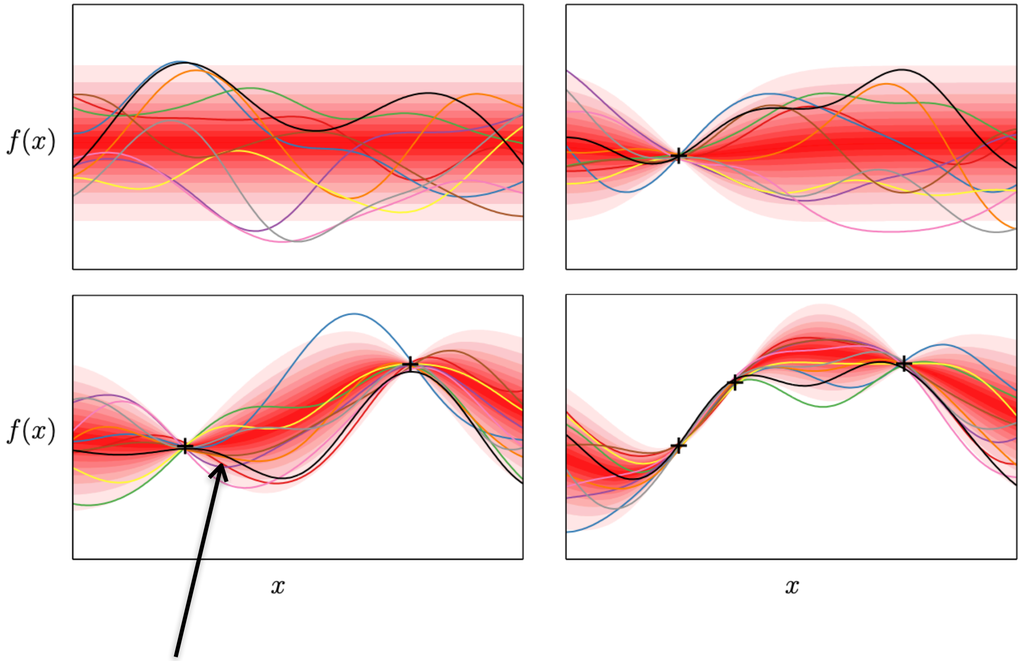
- Before conditioning on data, GP model is specified by a **mean** and **covariance** function (or *kernel*)

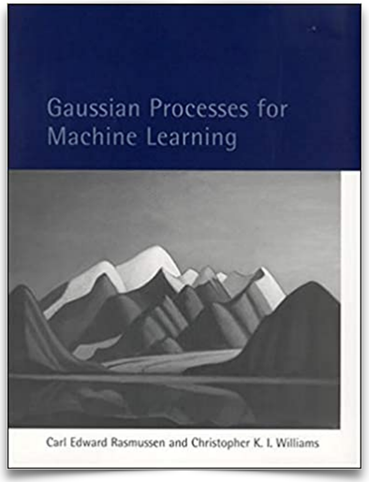
$$\mathbb{E}\left[f(\boldsymbol{x})\right] = \mu(\boldsymbol{x})$$

Mean function commonly set to zero everywhere

$$\mathrm{Cov}\left[f(\boldsymbol{x}), f(\boldsymbol{x}')\right] = k\left(\boldsymbol{x}, \boldsymbol{x}'\right)$$

The form of the kernel determines which *classes* of functions the GP can capture





At some *query* parameter location, $\boldsymbol{x}_*$ one can ask the model for likely function values given past observations $\boldsymbol{y}$

$$p\left(f(\boldsymbol{x}_*)|\boldsymbol{f}(\boldsymbol{X}), \boldsymbol{X}, \mu(\cdot), k(\cdot,\cdot)\right) = \mathcal{N}\left(f(\boldsymbol{x}_*)|\hat{\mu}(\boldsymbol{x}_*), \hat{\sigma}^2(\boldsymbol{x}_*)\right)$$

$$\hat{\mu}\left(\boldsymbol{x}_*\right) = \boldsymbol{k}\left(\boldsymbol{x}_*, \boldsymbol{X}\right)\left(\boldsymbol{K}\left(\boldsymbol{X}, \boldsymbol{X}\right) + \alpha_0 \boldsymbol{I}\right)^{-1} \boldsymbol{y}$$

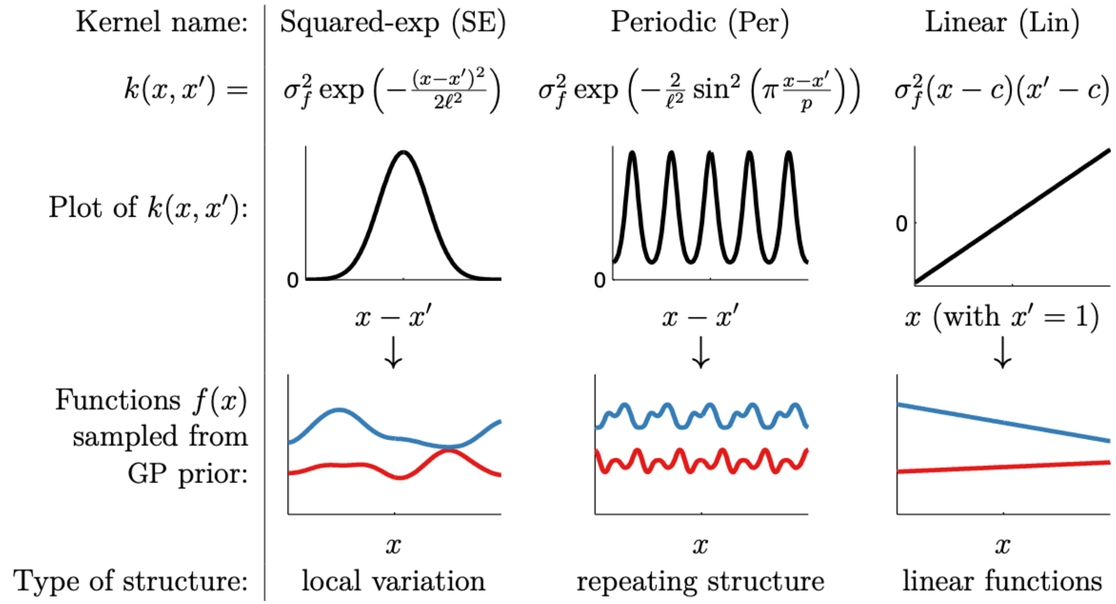
$$\hat{\sigma}^2\left(\boldsymbol{x}_*\right) = k\left(\boldsymbol{x}_*, \boldsymbol{x}_*\right) - \boldsymbol{k}\left(\boldsymbol{x}_*, \boldsymbol{X}\right)\left(\boldsymbol{K}\left(\boldsymbol{X}, \boldsymbol{X}\right) + \alpha_0 \boldsymbol{I}\right)^{-1} \boldsymbol{k}\left(\boldsymbol{X}, \boldsymbol{x}_*\right)$$

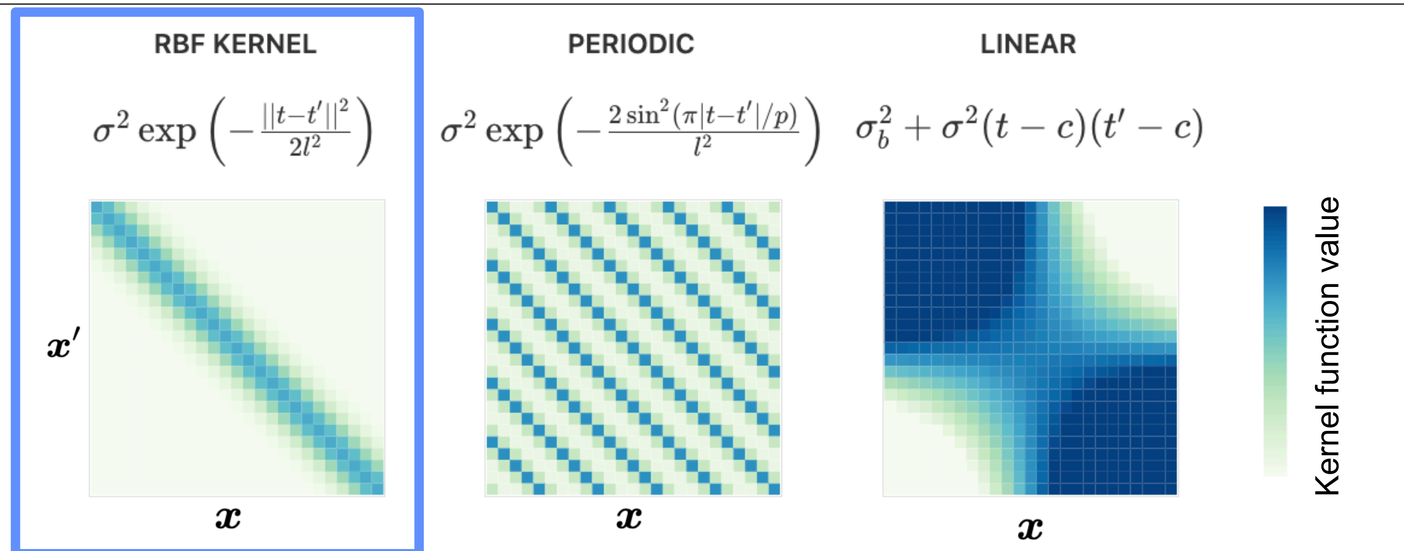
The kernel function

$$\mathrm{Cov}\left[f(\boldsymbol{x}), f(\boldsymbol{x}')\right] = k\left(\boldsymbol{x}, \boldsymbol{x}'\right)$$

The form of the kernel determines which *classes* of functions the GP can capture

It measures the *similarity* of two function values , which specifies which type of functions are likely under the prior
Popular choices are linear, square exponential (RBF), Matérn and periodic





Parameters in the kernel function are called *hyperparameters*

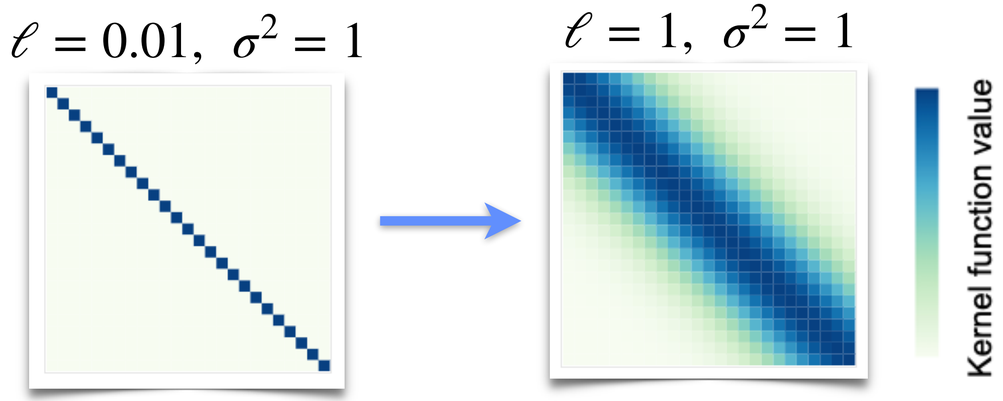
For example, the RBF kernel has a kernel variance, σ^2, and lengthscale ℓ

Hyperparameters can have great effect on GP performance, we need to optimize them, but how?

GP model selection

- Can compute *marginal likelihood* or *evidence* of the model given some data

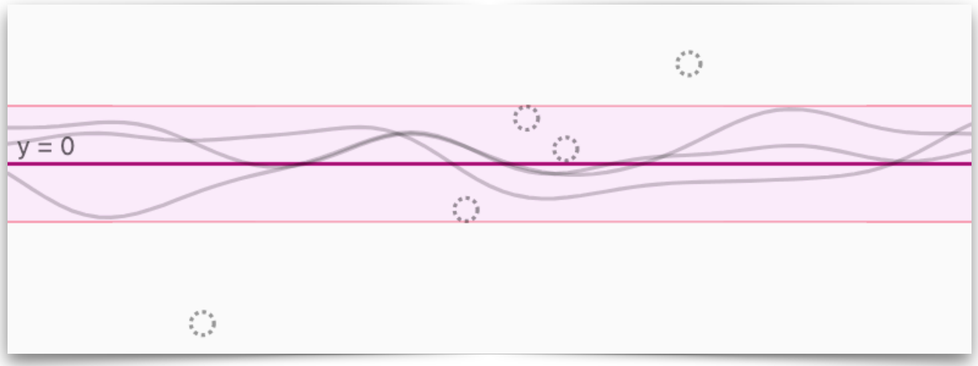
- This will let us construct models automatically by maximizing (or minimizing its negative) this value using a (usually gradient-based optimizer)

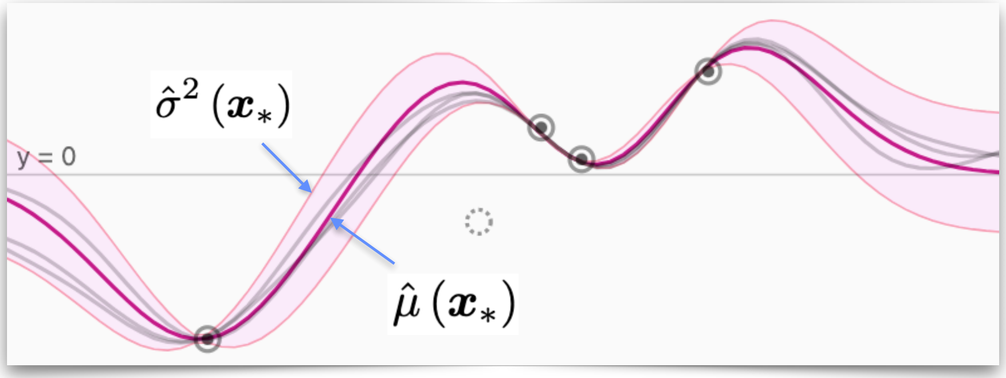


$$p\left(\boldsymbol{f}(\boldsymbol{X})|\boldsymbol{X}, \mu(\cdot), k(\cdot,\cdot)\right) = \mathcal{N}\left(\mu(\boldsymbol{X}), k(\boldsymbol{X},\boldsymbol{X})\right)$$
$$= \frac{1}{\sqrt{2\pi^N |k(\boldsymbol{X},\boldsymbol{X})|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{\mu}(\boldsymbol{X})\right)^T k(\boldsymbol{X},\boldsymbol{X})^{-1}\left(\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{\mu}(\boldsymbol{X})\right)\right\}$$

controls model capacity

encourages fit with data





Strengths and weaknesses of Gaussian processes

the matter lab

Strengths

- *Analytic inference:* predictive posterior distribution can be written in closed form
- *Expressivity*: can model a rich class of functions
- *Closed-form predictive distribution*: simply a multivariate Gaussian distribution!
- *Easy to analyze*: simple models with intuitive parameters

Weaknesses

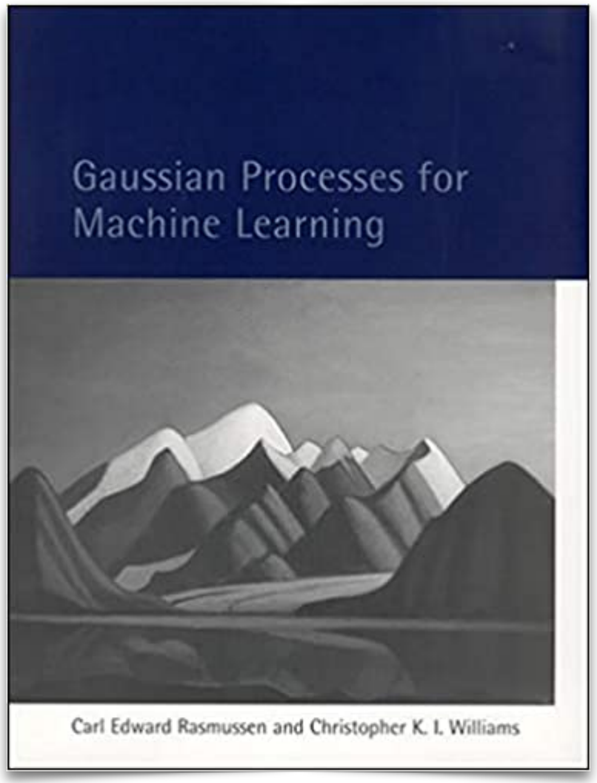
- *Costly inference,* $\mathcal{O}\left(N^3\right)$: prohibitively slow for more than a few thousand points
- *Need to choose an expressive kernel*: which kernel to use for a given problem?
- *Approximate inference needed for non-Gaussian likelihoods*: *e.g.* Bernoulli likelihood for binary classification

Resources and software for GPs

the matter lab

Resources

- Visual exploration Distill article
- GP book by Rasmussen [1]
- New BayesOpt book by Roman Garnett [2]



Software

- GPFlow (Google, Tensorflow)
- GPyTorch (Meta, PyTorch)
- PyMC3
- GPy
- scikit-learn
- Wikipedia article comparing GP softwares

[1] C.E. Rasmussen *et al.*, “Gaussian Processes for Machine Learning”, *MIT Press* (2006) [2] R. Garnett, “Bayesian Optimization”, *In prep* (2022)

Bayesian optimization

- Gold-standard in sample efficiency for global optimization
- Usually targeting noisy, expensive-to-evaluate, black-box functions (*i.e.* do not have gradient information)
- Train ML **"surrogate" model** on all available observations at each iteration
- Compute ***acquisition function*** from surrogate model (trade off between exploitation and exploration)
- Many surrogate models have been considered (GP, RF, NN, kernel smoothing)

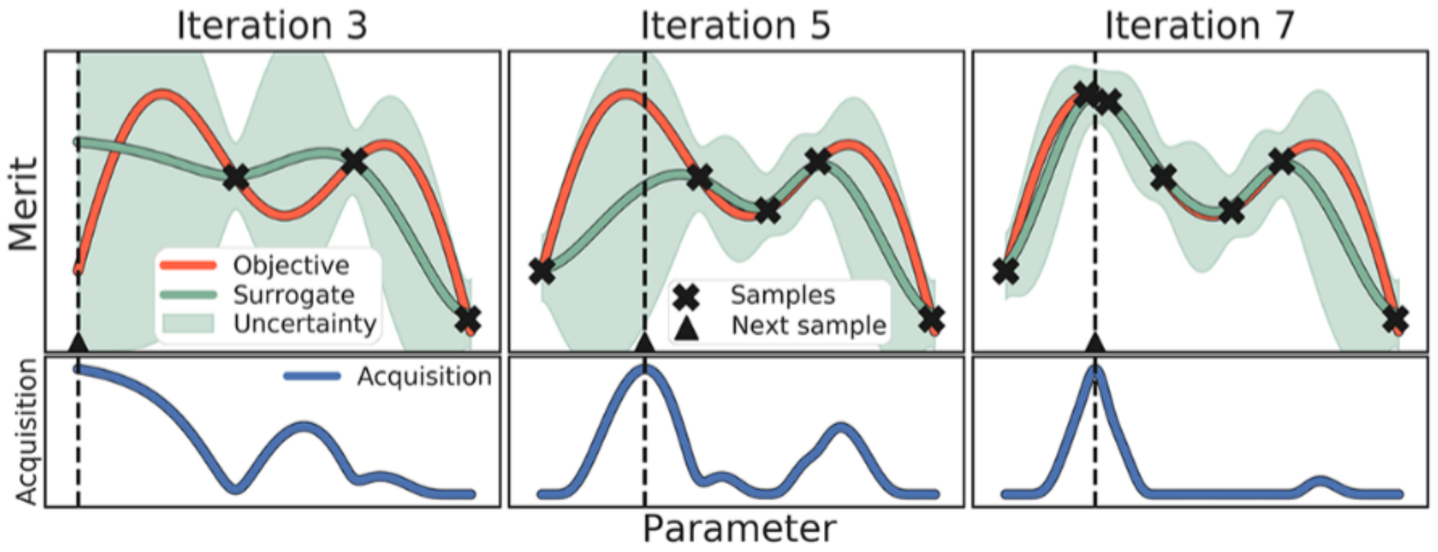
Algorithm: Bayesian optimization

Result: Optimize objective function, $f(x)$

while $i <$ *budget* **do**

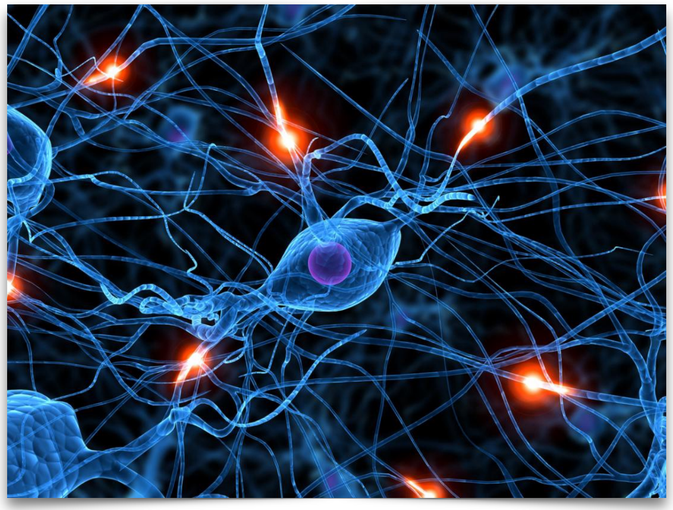
- build surrogate model ;
- compute acquisition function, $a(x)$;
- $x_{next} \leftarrow$ maximize $a(x)$;
- evaluate $f(x_{next})$;
- $i \leftarrow i + 1$;

end



What is Bayesian optimization used for?

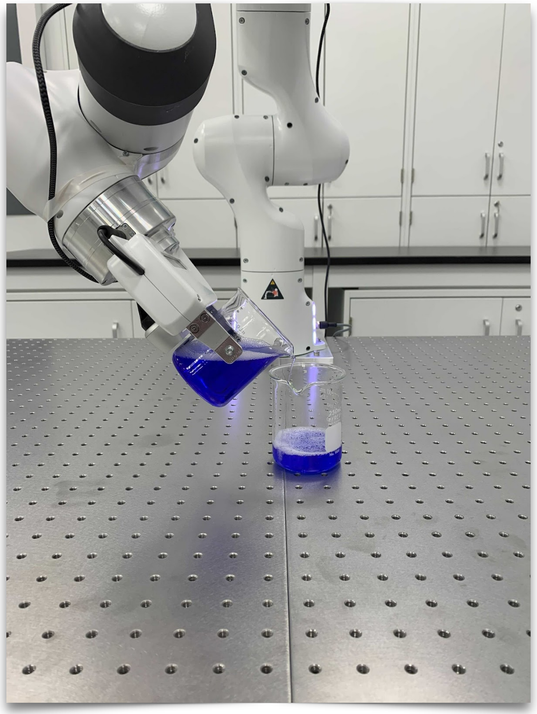
the matter lab



Hyperparameter optimization of machine learning models

Inverse design of parametric systems or models in physics, chemistry, electrical engineering, and biology

Control parameter tuning, policy optimization and structure design in robotics



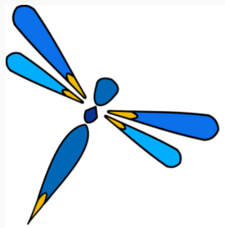
Optimization of matter in self-driving labs or materials acceleration platforms

Popular software packages for Bayesian optimization

the matter lab

- There are many nice existing BayesOpt software packages (focusing exclusively on *Python*)
- APIs vary from high-level (application focused) and low-level (developer focused)
- Some provide “custom” components (acq funds, surrogate models), and some focus on providing wrappers for multiple
  - Ax/BoTorch (Meta)
  - GPyOpt
  - Dragonfly (CMU)
  - Gryffin (Aspuru-Guzik group)
  - SMAC3 (Freiburg-Hannover)
  - Hyperopt
  - HEBO (Huawei)

Ax



Dragonfly

Scalable Bayesian Optimisation

HYPEROPT

HEBO 河伯

Heteroscedastic Evolutionary Bayesian Optimisation

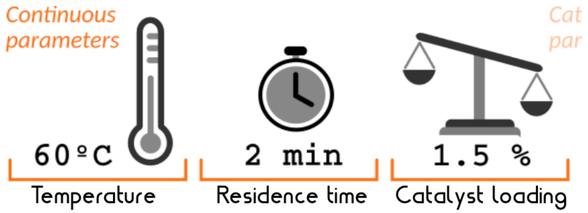
Parameter spaces

- The parameter space or domain $\mathcal{X}$ is the space which the optimizer can traverse
- There are several different flavours of parameters, but we will consider only three main types

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$$

Continuous parameters

Continuously varying parameters,
specified by an *upper* and *lower* bound on their range



Discrete parameters

Discretized continuous space, specified by an *upper* and *lower* bound on their range and a *stride* or *interval*

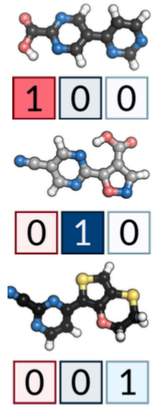
Categorical parameters

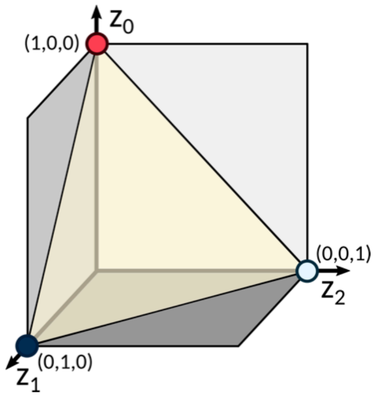
Parameters *without* natural ordering between their options

Most commonly (in chemistry) molecules, fragments, R-groups

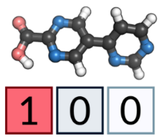
Usually represented as one-hot-encoded vectors

Possible to use *descriptors* of each option to introduce notion of similarity

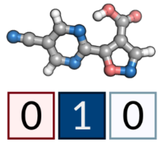




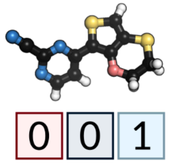
$$\zeta \in \partial\Delta^{n-1} = \left\{ \zeta \in \mathbb{R}^n | \zeta_i \in \{0,1\} \text{ and } \sum_{i=1}^{n} \zeta_i = 1 \right\}$$



HOMO: -5.48 eV
LUMO: -3.84 eV



HOMO: -5.18 eV
LUMO: -3.24 eV



HOMO: -6.40 eV
LUMO: -4.10 eV

Naïve categorical encodings, we will use kernel based on Hamming distances

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\text{dist}(\boldsymbol{x}, \boldsymbol{x}')/\ell\right)$$

$$\text{dist}(\boldsymbol{x}, \boldsymbol{x}') = \begin{cases} 0 & \boldsymbol{x} = \boldsymbol{x}' \\ 1 & \boldsymbol{x} \neq \boldsymbol{x}' \end{cases}$$

Parameter spaces

- The parameter space $\mathcal{X}$ can also have constraints!
- A constraint function $c(x)$ determines for which parameter settings the objective function $f(x)$ can be measured
- $c(x)$ could be *a priori* known, or *a priori* unknown, and can be interdependent, non-linear and can lead to non-compact optimization domains

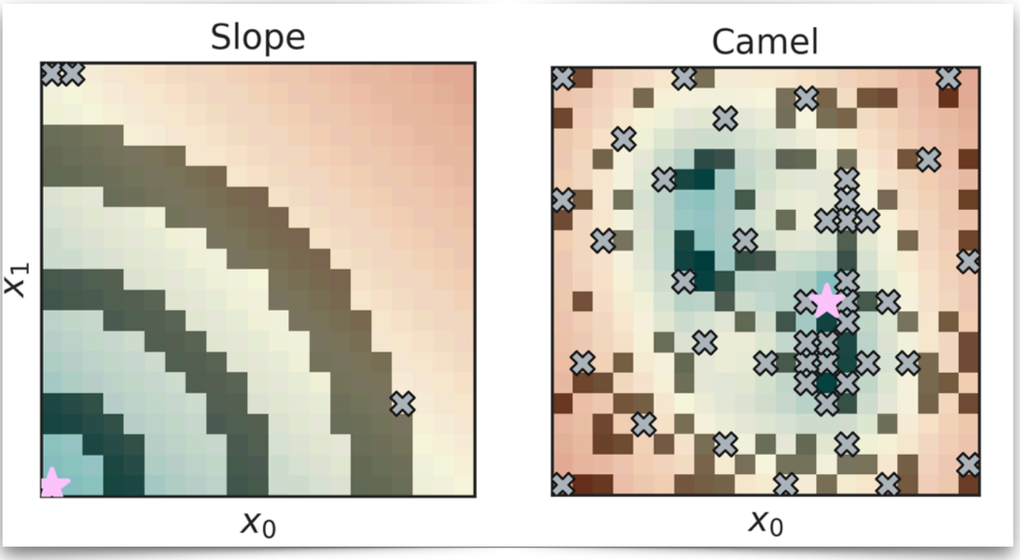
$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) \longrightarrow \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}), \quad \text{s.t. } c(\boldsymbol{x}) \mapsto \text{feasible}.$$

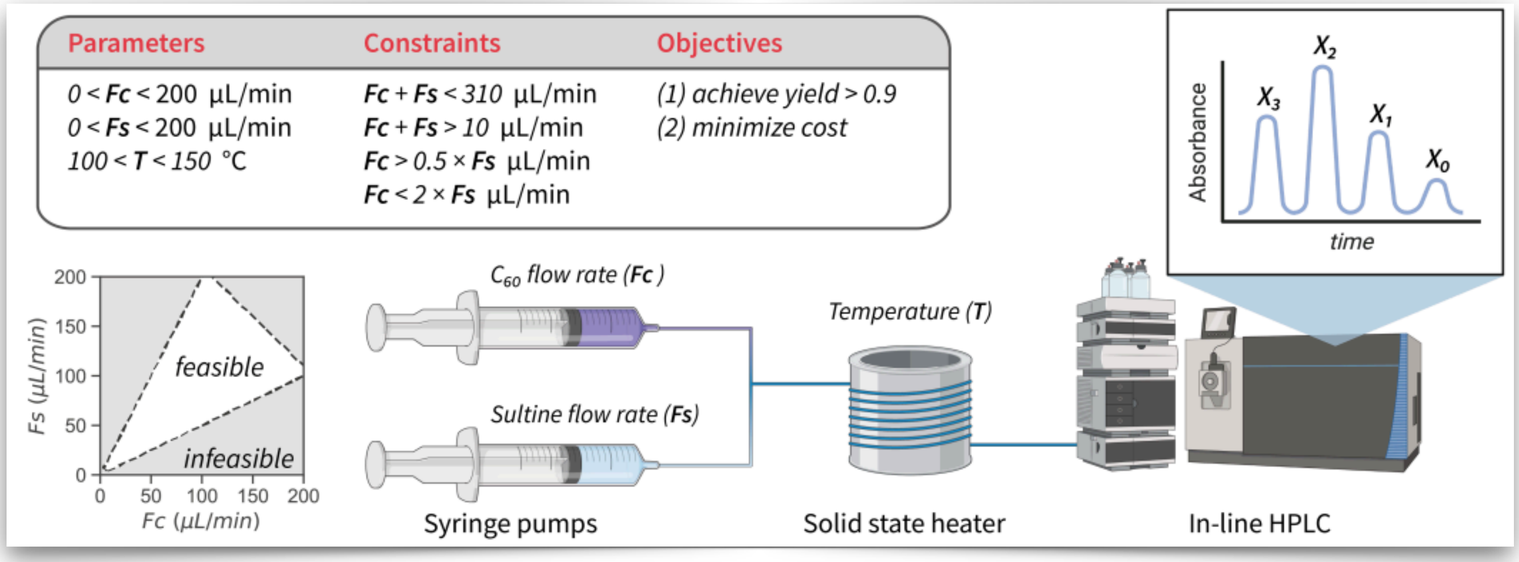
Known constraints

e.g. volume of mixed liquids cannot exceed volume of reaction vessel, solvent dependent temperature ranges

Unknown constraints

e.g. failed/abandoned syntheses, unstable molecules/materials, insufficient conditions to make property measurement





Acquisition functions

the matter lab

- An acquisition function, $\alpha(x)$, evaluates candidate parameter points based on their expected informativeness and performance

- Most acquisition functions maintain a “tradeoff” between *exploitation* and *exploration*

- An important subroutine of Bayesian optimization is the optimization of the acquisition function. Its argmin/ argmax corresponds to the parameter point to measure next

$$\boldsymbol{x}_{\text{next}} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \alpha(\boldsymbol{x})$$

- Some common acquisition functions include *probability of improvement* (PI), *upper confidence bound* (UCB) and *expected improvement* (EI)

Upper confidence bound (UCB)

UCB computes utility by trading off the models mean and variance using a parameter $\beta > 0$

$$\alpha(\boldsymbol{x}) = \hat{\mu}(\boldsymbol{x}) - \beta\hat{\sigma}(\boldsymbol{x})$$

Expected improvement (EI)

EI proposes to evaluate the objective where we expect the greatest improvement over current best point f'

$$\alpha(\boldsymbol{x}) = (\hat{\mu}(\boldsymbol{x}) - f' - \xi)\,\Phi(Z) + \hat{\sigma}(\boldsymbol{x})\phi(Z)$$

$$Z = \frac{\hat{\mu}(\boldsymbol{x}) - f' - \xi}{\hat{\sigma}(\boldsymbol{x})}.$$

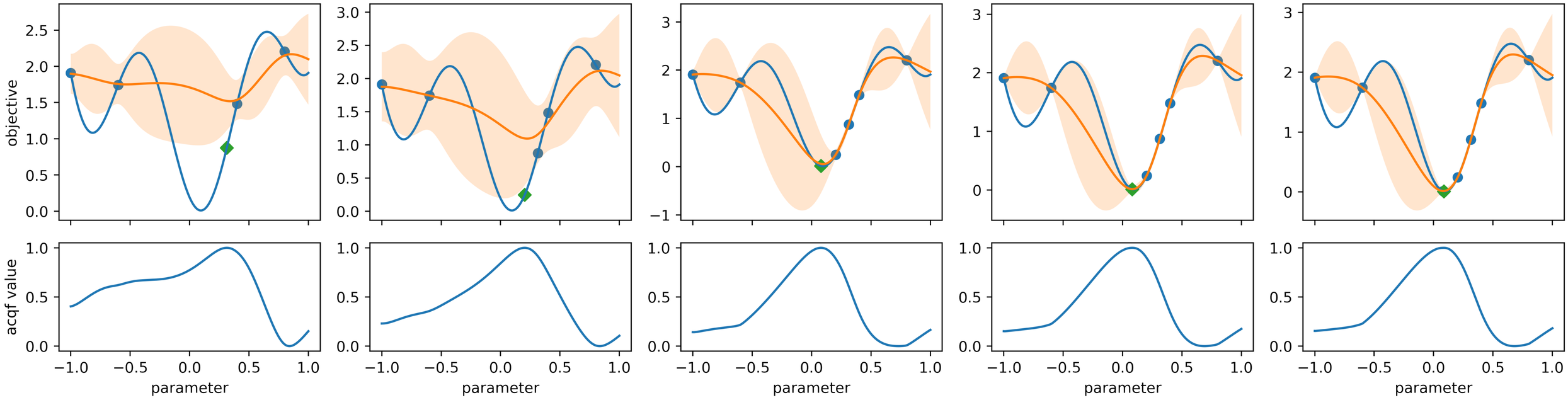
Acquisition functions

the matter lab

Upper confidence bound (UCB)

UCB computes utility by trading off the models mean and variance using a parameter $\beta > 0$

$$\alpha(\boldsymbol{x}) = \hat{\mu}(\boldsymbol{x}) - \beta\hat{\sigma}(\boldsymbol{x})$$



Metrics for assessing the performance of Bayesian optimization

the matter lab

- Bayesian optimizers seek to identify promising target property values more efficiently (i.e. with fewer evaluations) than other strategies
- One can use several metrics to compare optimization performance, depending on parameter space and objective

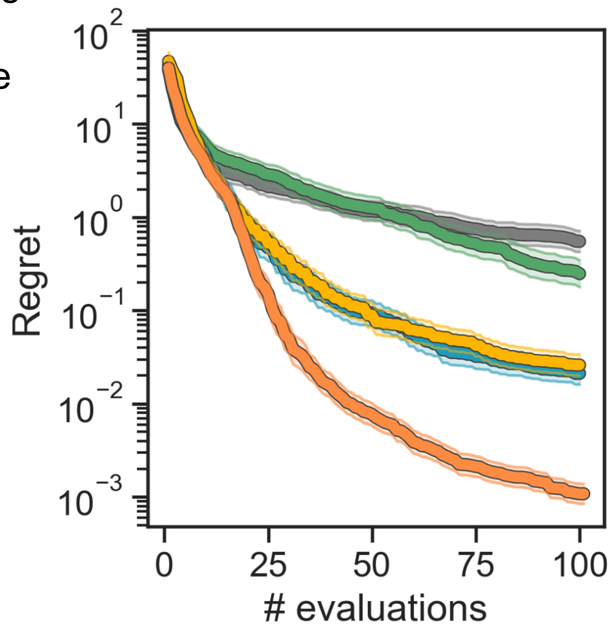
Continuous-valued parameters

Regret measures the instantaneous distance between best observed objective and global optimum value

$$r_k = |f(\boldsymbol{x}^*) - f(\boldsymbol{x}_k^+)|$$

Cumulative regret sums all regret values over the entire optimization campaign (with budget K)

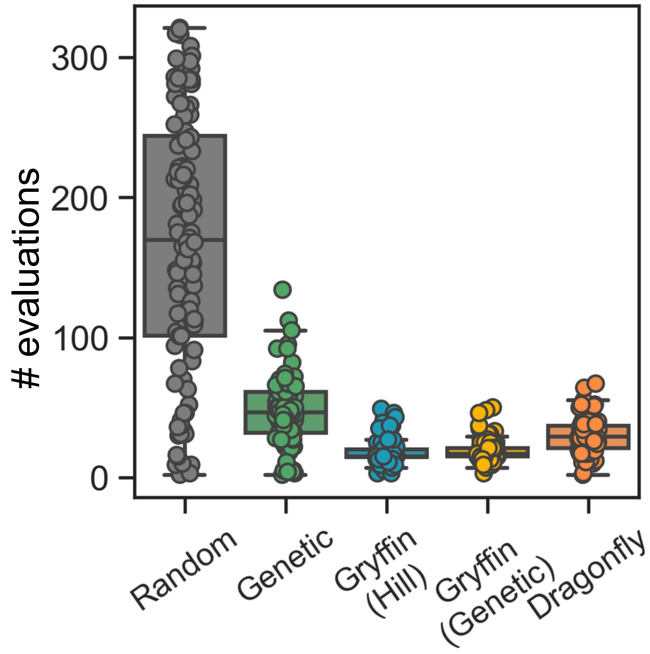
$$C_K = \sum_{k=1}^{K} r_k$$



Categorical-valued parameters

Could also use regret metrics, but now we have a discrete option that is the global optimum. Here I plot number of evaluations needed to identify that option

If you like traces, you can also plot the cumulative best rank achieved at each iteration to highlight differences in small objective function differences



Putting it all together

Now that we have seen the machinery behind Bayesian optimization, let's code up a Bayesian optimizer wrapper using *BoTorch* and *Olympus* and test it on some chemistry/materials science problems!

Link

Link

What will our optimizer look like?

- Will build it using the Olympus API, as a *CustomPlanner*
- Will have an ask-tell interface (ask planner for parameters, tell planner about measurements)
- Will handle continuous, categorical, and mixed parameter spaces
- Will use a Gaussian process surrogate model
- Will use expected improvement acquisition function

Problems for testing

1. Optimization of the yield of a Suzuki reaction (fully continuous, 4 parameters, 1 objective) [1]
2. Design of hybrid organic-inorganic perovskite materials (fully categorical, with/without descriptors, 3 parameters, 1 objective) [2]

[1] M. Christensen *et al. Commun. Chem.* **4**, 112 (2021)

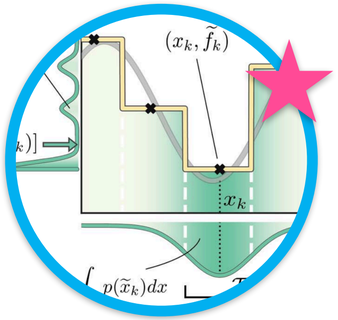
[2] C. Kim *et al.* Sci. *Data.* **4**, 170057 (2017)

First coding session

Advanced Bayesian optimization topics

the matter lab

Golem Robust optimization to input parameter noise

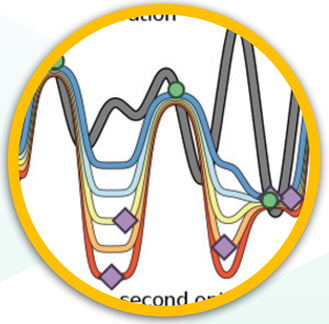


M. Aldeghi, F. Häse, **RJH**, *et al.*
Chem. Sci. **12**, 14792-14807 (2021)

Olympus Community accessible benchmarking of experiment planning strategies

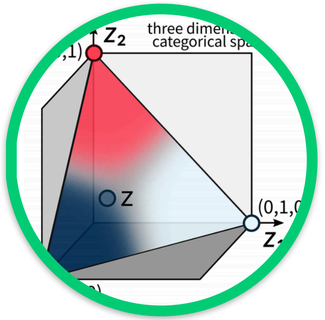
F. Häse*, M. Aldeghi*, **RJH**, *et al.*
Mach. Learn. Sci. Technol. **2**, 035021 (2021)

Phoenics KDE BayesOpt for continuous parameters



F. Häse, *et al. ACS Cent. Sci.* ***4****,* 1134–1145 (2018)

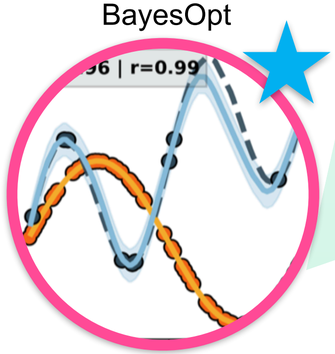
Gryffin KDE BayesOpt for categorical parameters



F. Häse, M. Aldeghi, **RJH**, *et al.*
Appl. Phys. Rev. **8**, 031406 (2021)

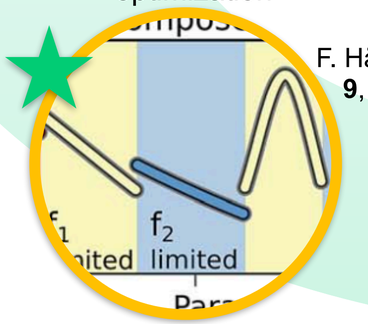
the matter lab

Gemini Scalable multi-fidelity prediction and BayesOpt



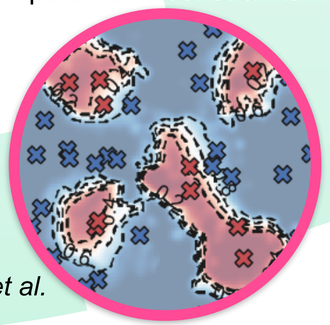
RJH, *et al.* arXiv:2103.03391 **[stat.ML]** (2021)

Chimera Hierarchy-based multi-objective optimization



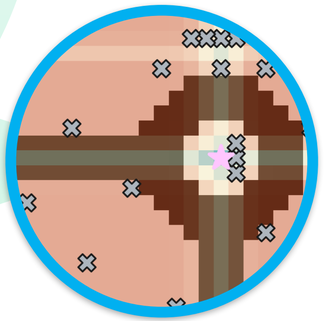
F. Häse, *et al. Chem. Sci.* **9**, 7642–7655 (2018)

Unknown constraints *A priori* unknown parameter constraints



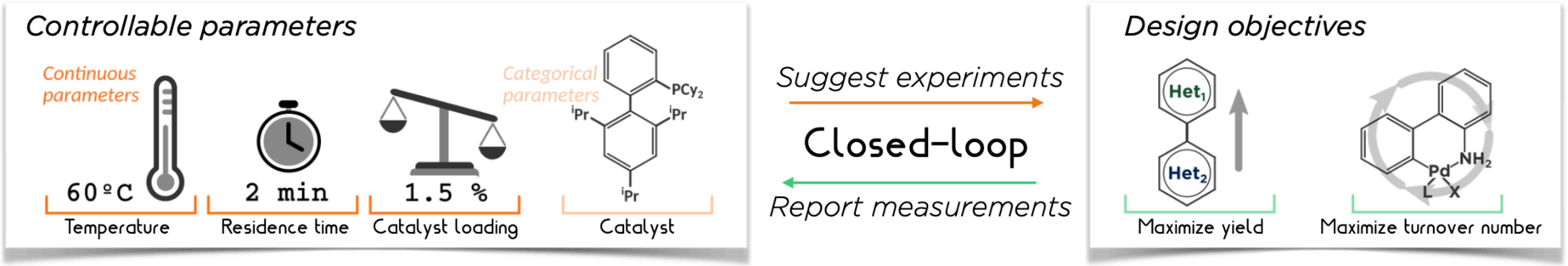
M. Aldeghi*, **RJH***, *et al.*
In preparation

Known constraints *A priori* known parameter constraints

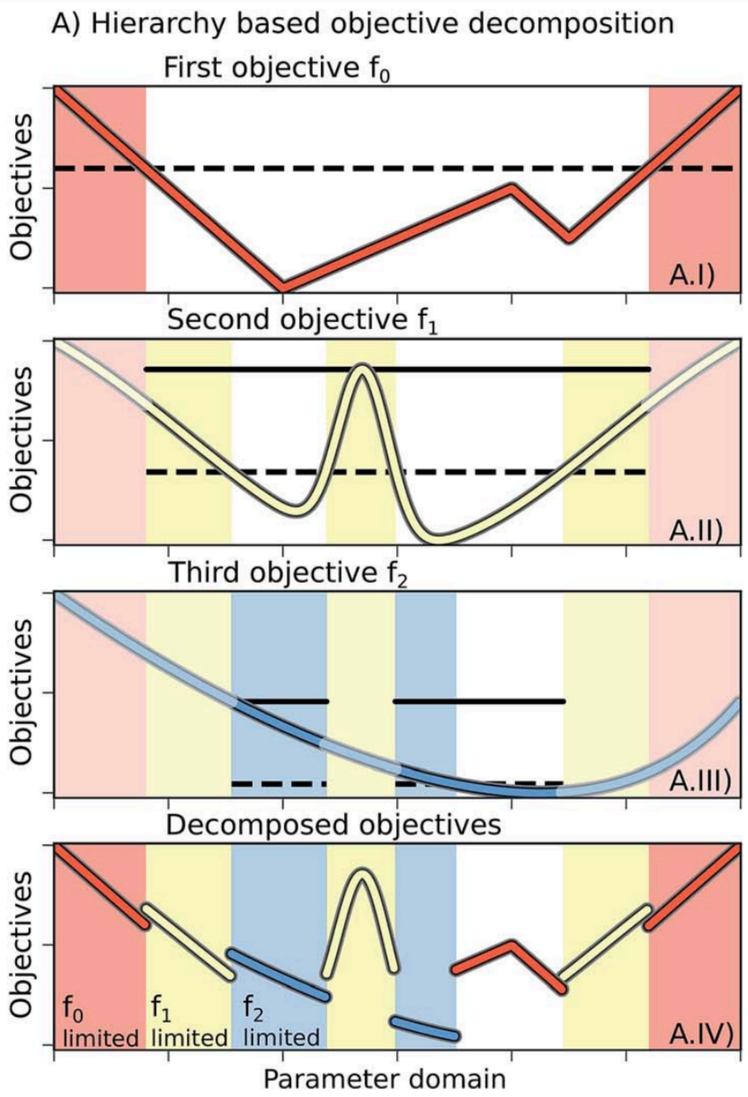


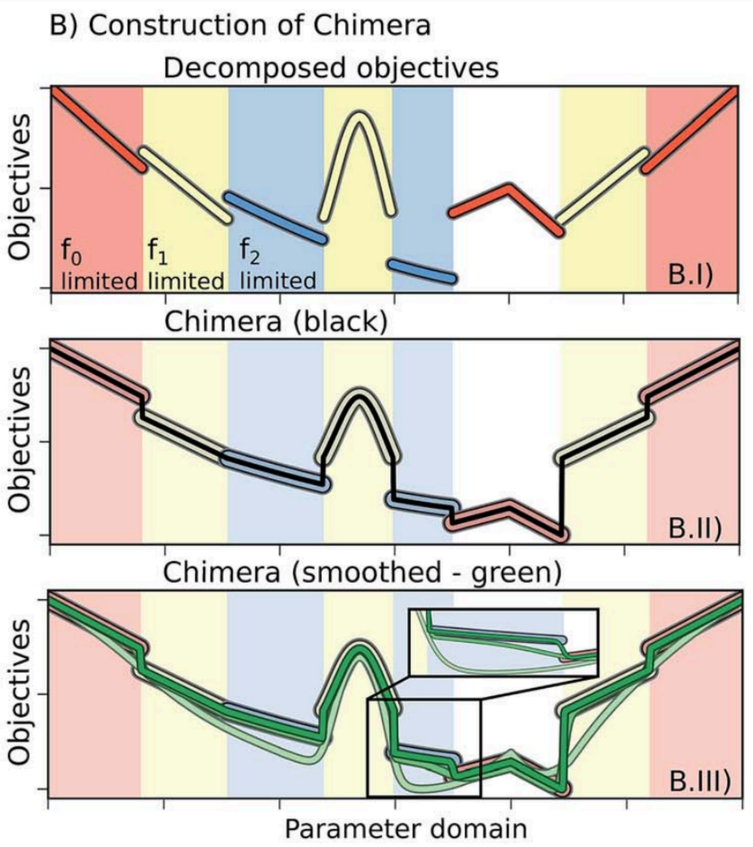
RJH*, M. Aldeghi*, *et al.*
arXiv:2203.17241 [math.OC] (2022)

Multiobjective optimization



Multiobjective optimization





C) Pseudo code representation

function $\chi(\mathbf{x}, \mathcal{Y})$

$f_0^{\min} \leftarrow \min_{\mathbf{x}_i \in \mathcal{Y}} f_0(\mathbf{x}_i)$

$f_1^{\min} \leftarrow \min_{\mathbf{x}_i \in \mathcal{Y}_0} f_1(\mathbf{x}_i)$

$f_2^{\min} \leftarrow \min_{\mathbf{x}_i \in \mathcal{Y}_1} f_2(\mathbf{x}_i)$

if $f_0(\mathbf{x}) \leq f_0^{\text{tol}}$ **then**
  $y_1 \leftarrow f_1(\mathbf{x}) - f_0^{\min}$
  if $f_1(\mathbf{x}) \leq f_1^{\text{tol}}$ **then**
    $y_2 \leftarrow f_2(\mathbf{x}) - f_1^{\min}$
    if $f_2(\mathbf{x}) \leq f_2^{\text{tol}}$ **then**
      $y_0 \leftarrow f_0(\mathbf{x}) - f_2^{\min}$
      return y_0
    else
      return y_2
  else
    return y_1
else
  return $f_0(\mathbf{x})$

D) Analytic form of Chimera (B.II) for three objectives (A.I to A.III)

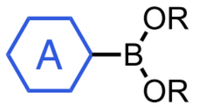
$$\chi(\mathbf{x}) = \Theta_0^+(\mathbf{x}) f_0(\mathbf{x}) + \Theta_0^-(\mathbf{x})\Theta_1^+(\mathbf{x}) \left(f_1(\mathbf{x}) - f_0^{\min}\right) + \Theta_0^-(\mathbf{x})\Theta_1^-(\mathbf{x})\Theta_2^+(\mathbf{x}) \left(f_2(\mathbf{x}) - f_1^{\min}\right) + \Theta_0^-(\mathbf{x})\Theta_1^-(\mathbf{x})\Theta_2^-(\mathbf{x}) \left(f_0(\mathbf{x}) - f_2^{\min}\right)$$

Design of organic laser molecules with targeted photophysical properties

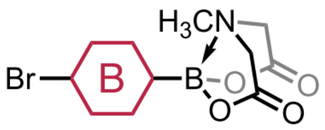
the matter lab

- *RouteScore* metric: mixed automated/manual synthetic efficiency score
- Multi-objective optimization of organic laser molecules for targeted photophysical properties with synthetic efficiency considered

14 “caps”

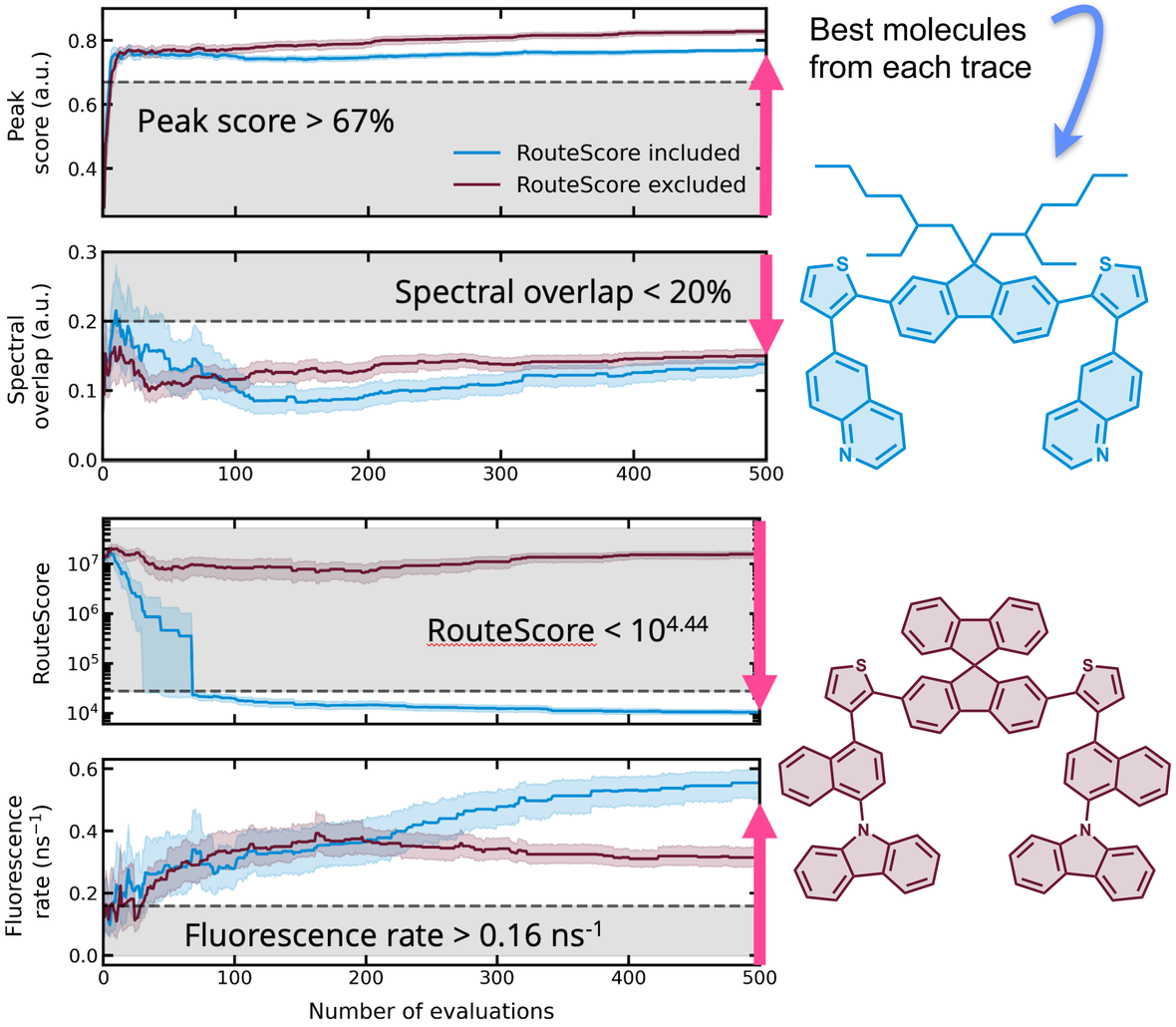


13 “bridges”



19 “cores”

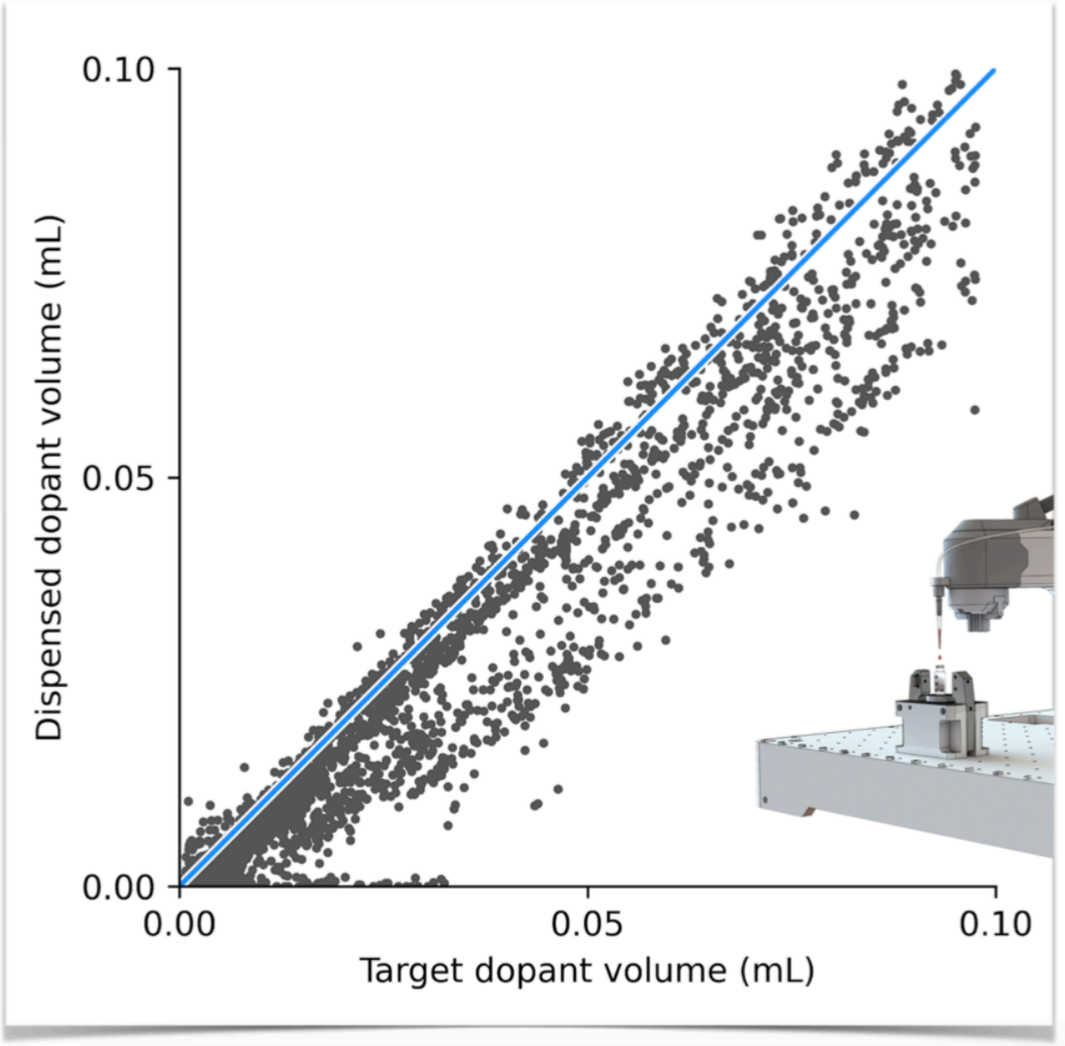
3458 symmetric pentamers



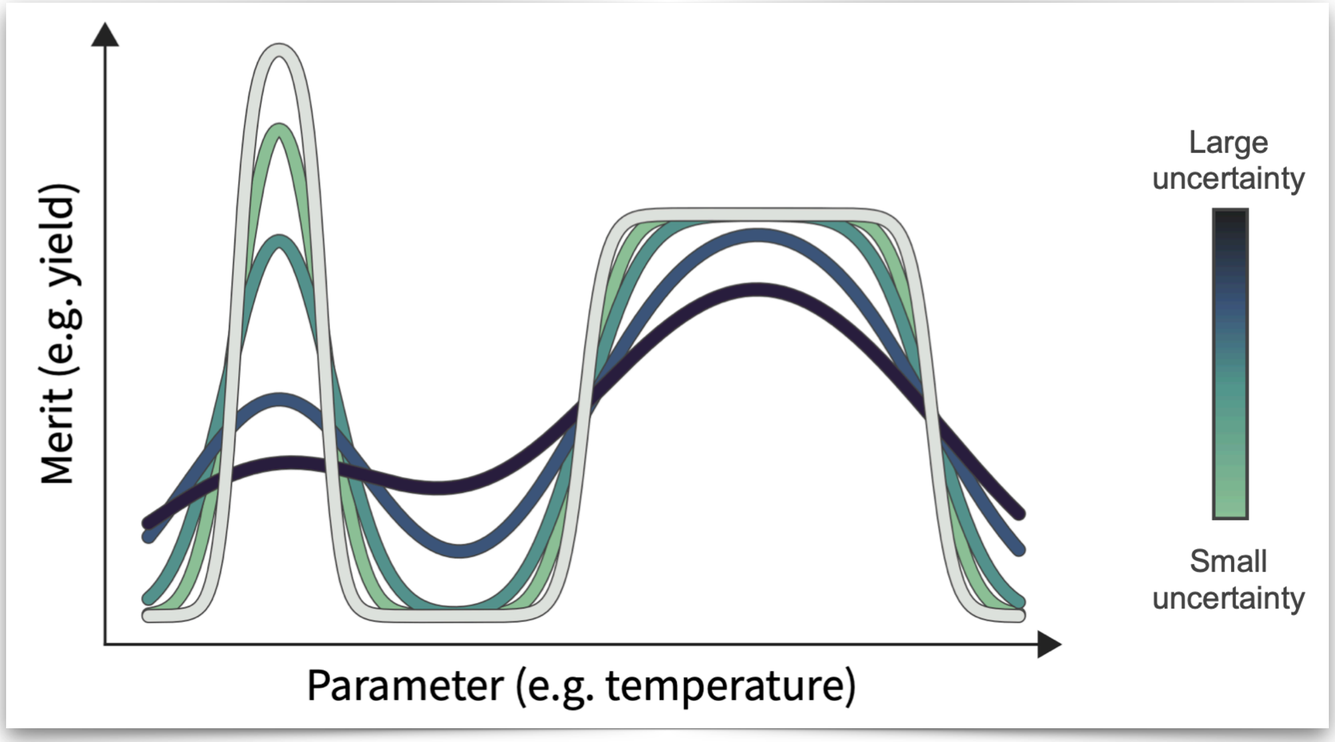
Robust optimization with Golem

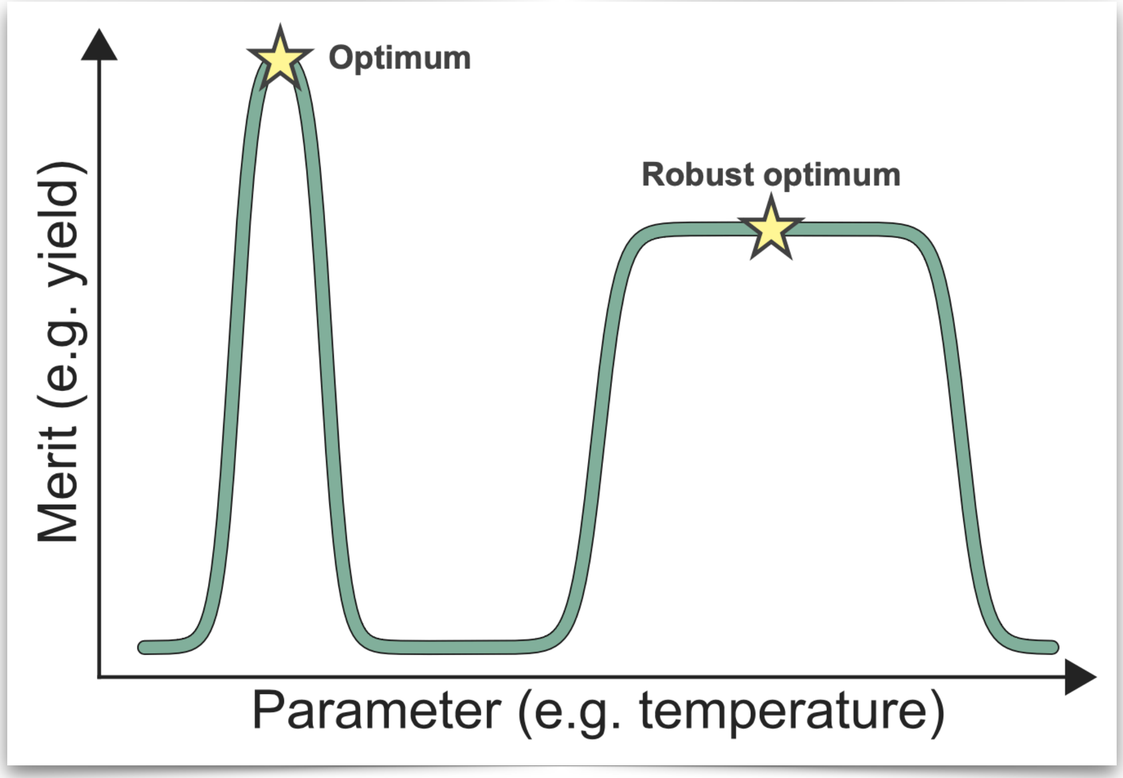
the matter lab

- Most experiment planning algorithms assume exact control over input parameters
- In reality, this is often not the case
- Noisy experimental conditions
- Operational variability in manufacturing
- “Hidden” input parameters varying between laboratories (ambient pressure, temperature)



Robust optimization with Golem





$$g(\boldsymbol{x}) \equiv \mathbb{E}[f(\boldsymbol{x})] = \int f(\boldsymbol{x}+\boldsymbol{\delta})p(\boldsymbol{\delta})d\boldsymbol{\delta}$$

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmin}}\, g(\boldsymbol{x})$$

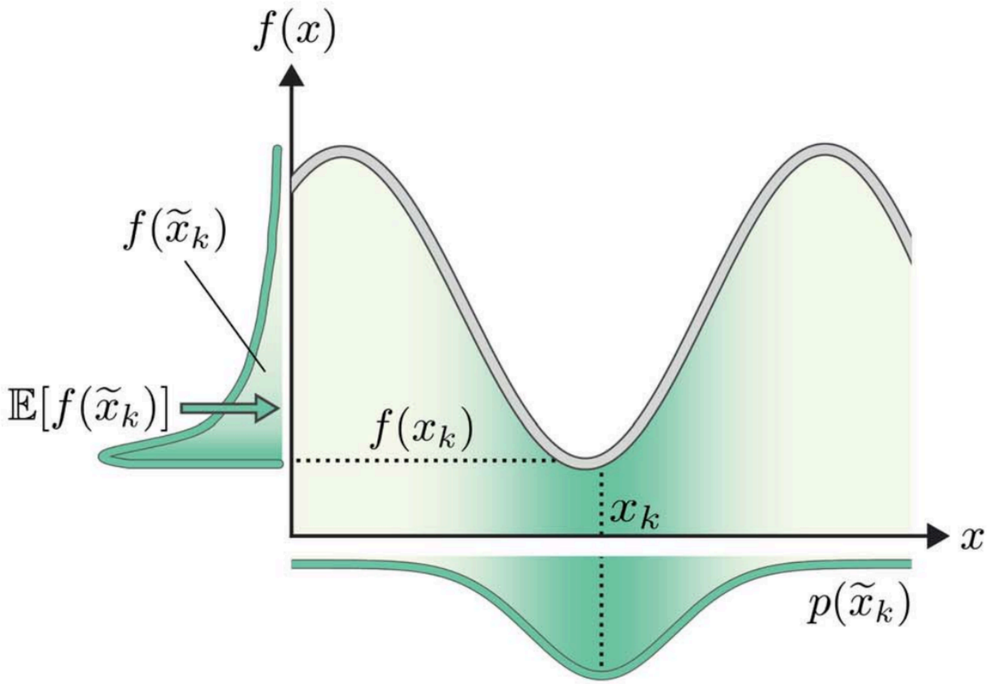
Robust optimization with Golem

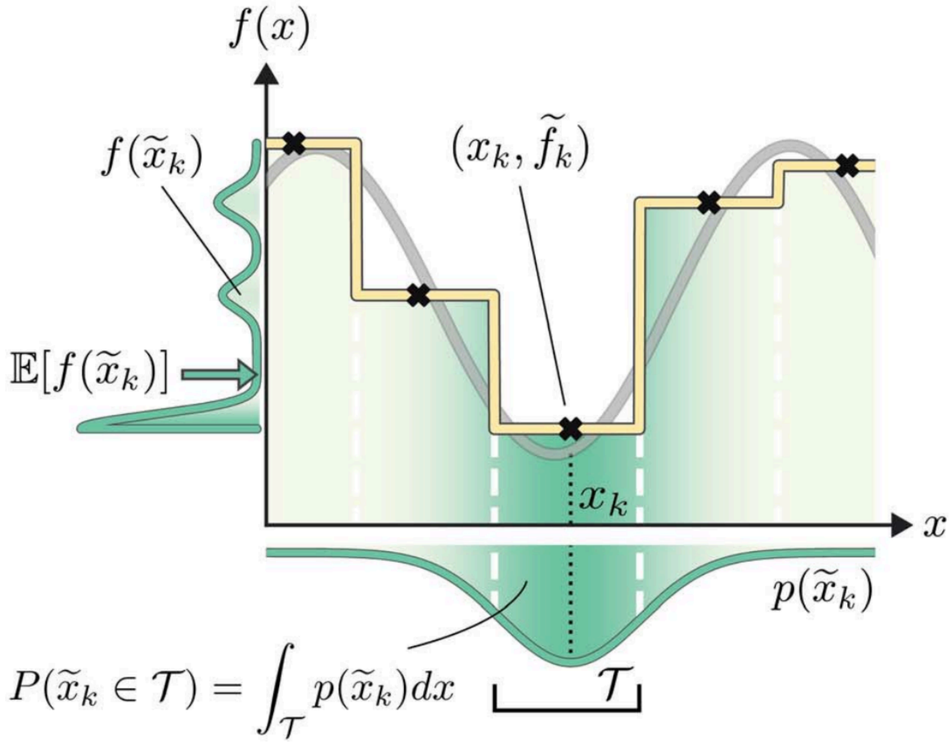
the matter lab

- use tree-based ML method to model $f(x)$ so integral $\int f(x)p(x)dx$ can de discretized for *any* $p(x)$

$$g(\boldsymbol{x}) = \mathbb{E}\left[f(\boldsymbol{x})\right] = \sum_{m=1}^{M} f_m P\left(\boldsymbol{x} \in \mathcal{T}_m\right)$$

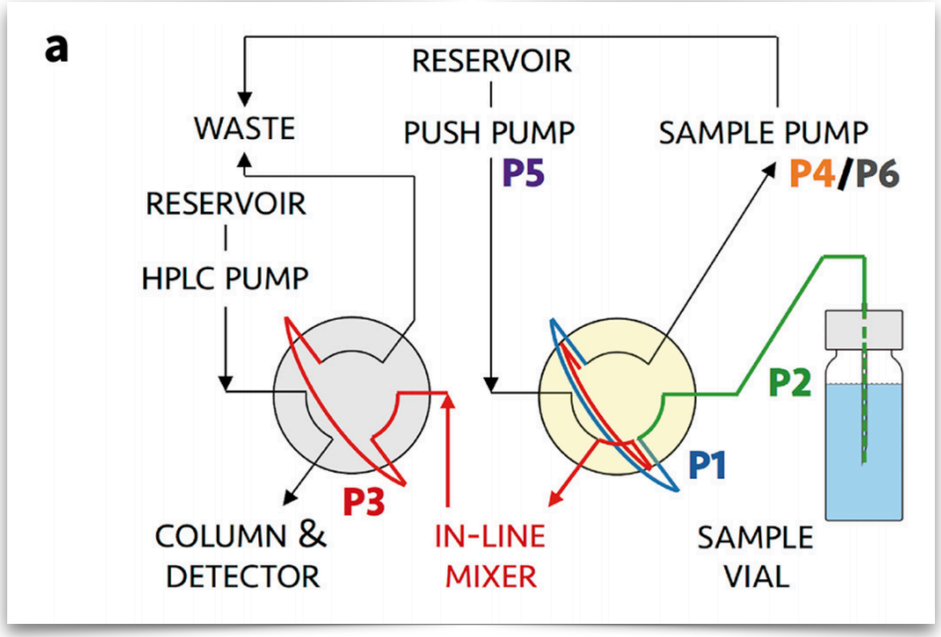
$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmin}}\, g(\boldsymbol{x})$$

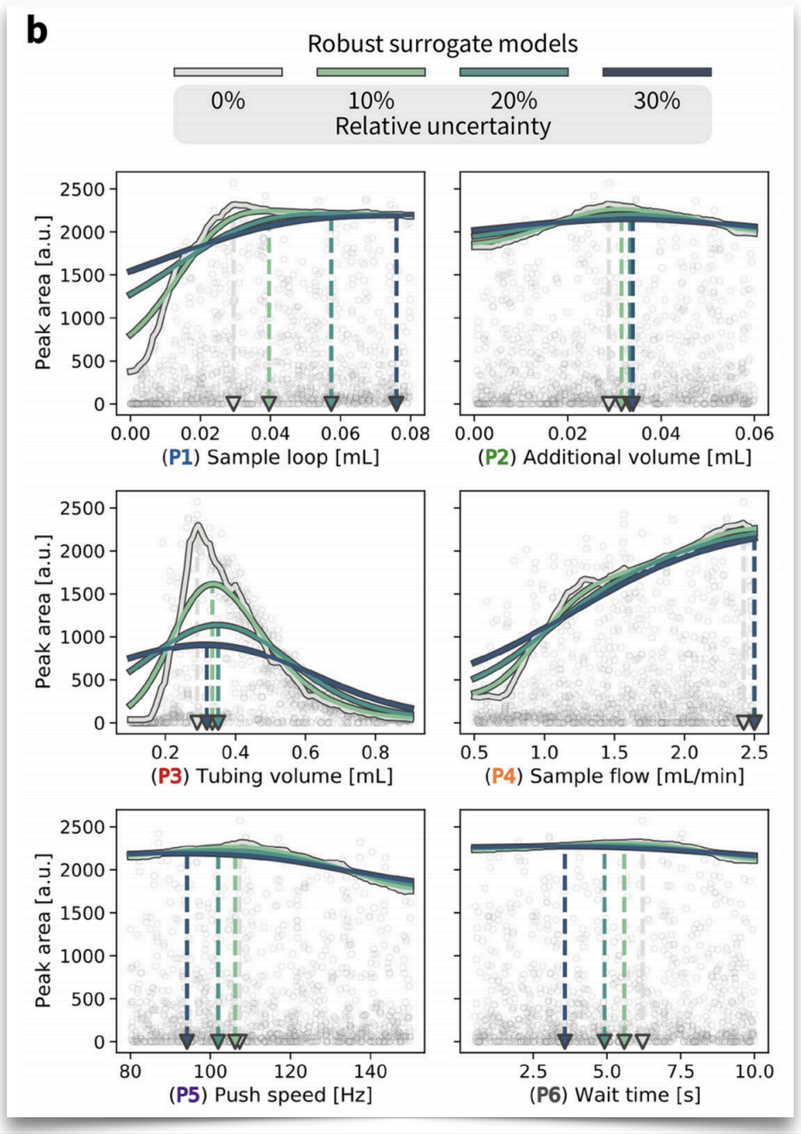




HPLC calibration application

- 6 parameters
- Objective is peak area maximization
- 1386 randomly collected samples



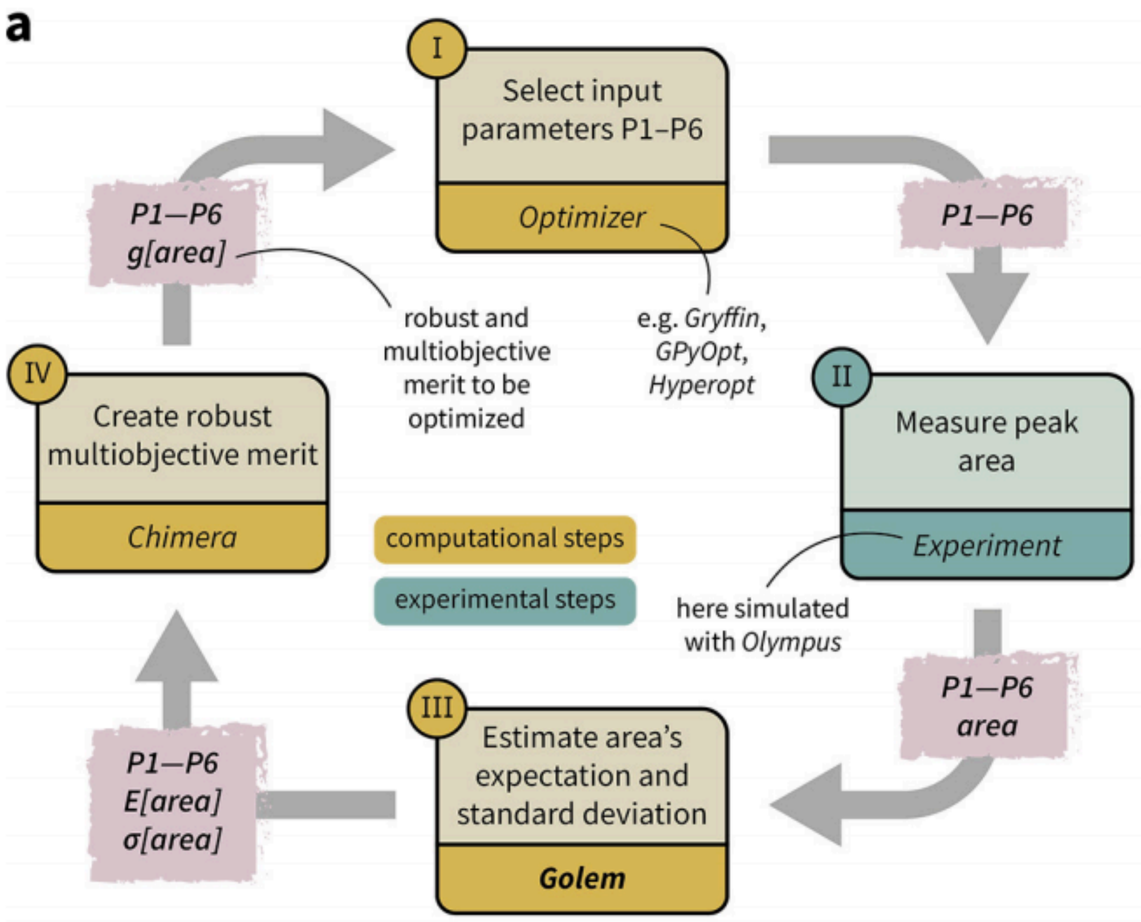


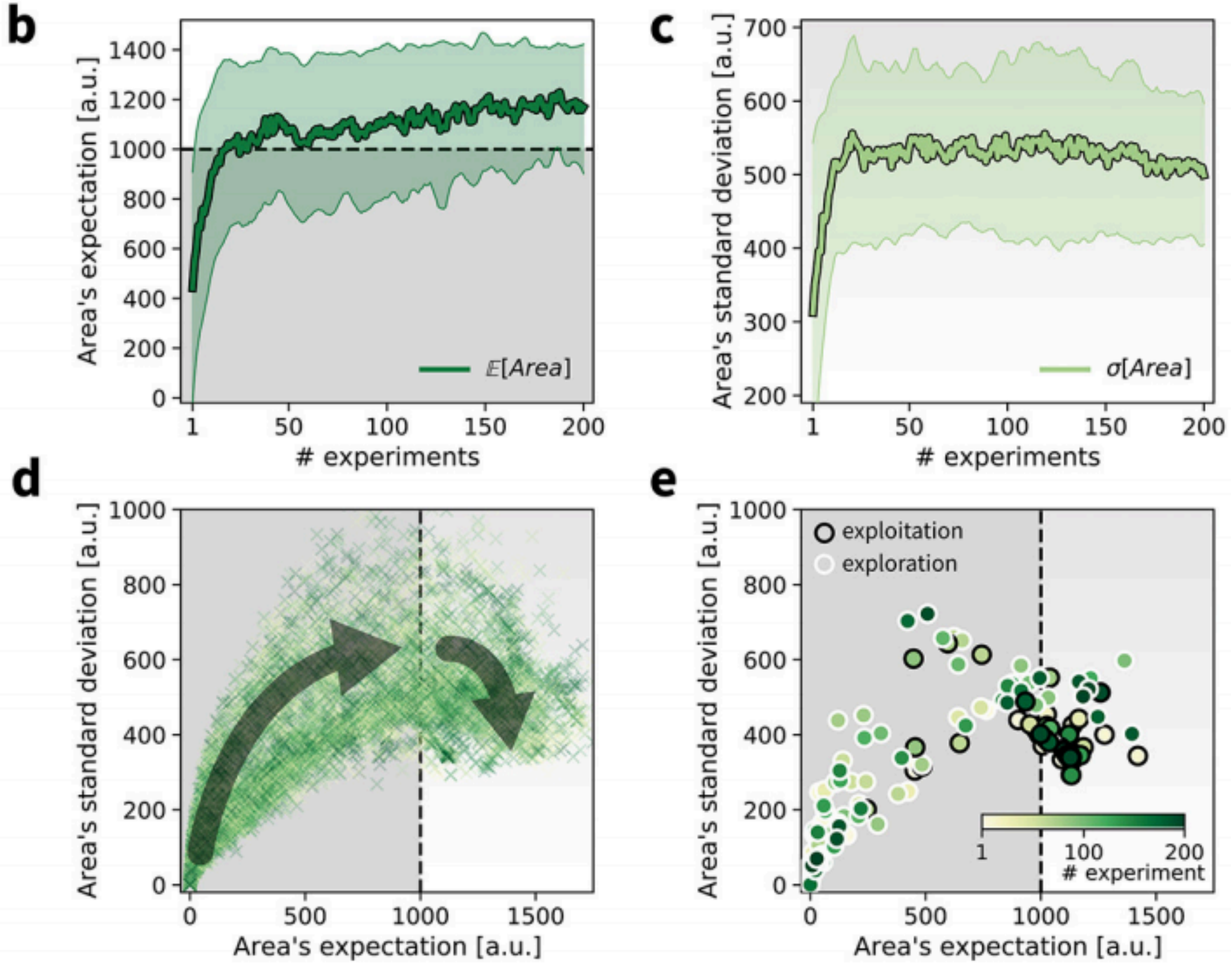
A posteriori noise analysis

HPLC calibration application

the matter lab

Maximize $\mathbb{E}[Area]$, minimize $\sigma[Area]$ —> GOLEM + CHIMERA —> scalar-valued robust objective $g[Area]$





Multi-fidelity prediction and optimization

the matter lab

What is “heterogenous” data?

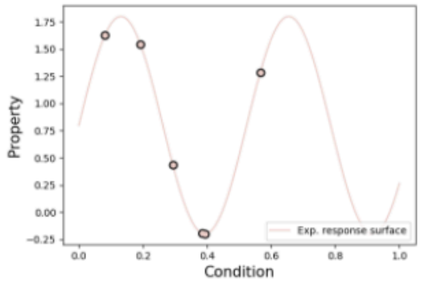
Great question! It can be defined in an application specific manner

i.e. monetary cheap vs expensive, fast vs slow, high vs low fidelity,
toxic vs non-toxic, environmentally friendly vs harmful, rare vs abundant, …

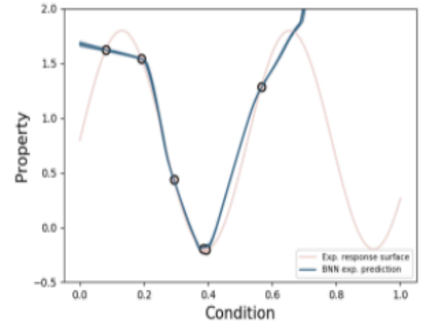
Can we use **cheap, less accurate** measurements to enhance the prediction of the outcomes of **slow, more accurate** measurements?

Expensive measurement

Single response surface

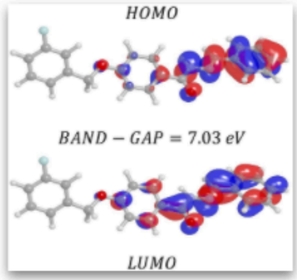


Poor prediction using ML

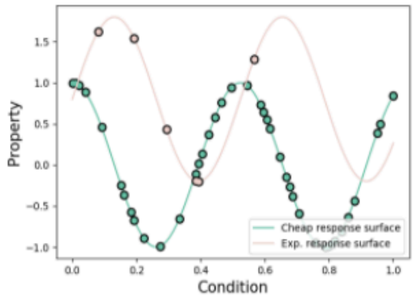


Expensive measurement

Cheap measurement



Two response surfaces



More accurate prediction?

Multi-task Gaussian processes

- Lets start from the beginning, with the multi-task Gaussian process model for BayesOpt, as implemented by Swersky *et* al. in 2013
- There have been many multi-fidelity strategies introduced since, including deep GPs, NN-based models, autoregressive GPs, to name a few

Assume we have T tasks, each with N_t observations, we then have the datasets

$$\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^{T} \qquad \mathcal{D}_t = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^{N_t}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') \longrightarrow k_{\text{multi}}((\boldsymbol{x}, t), (\boldsymbol{x}', t')) = k_t(t, t') \otimes k_x(\boldsymbol{x}, \boldsymbol{x}')$$

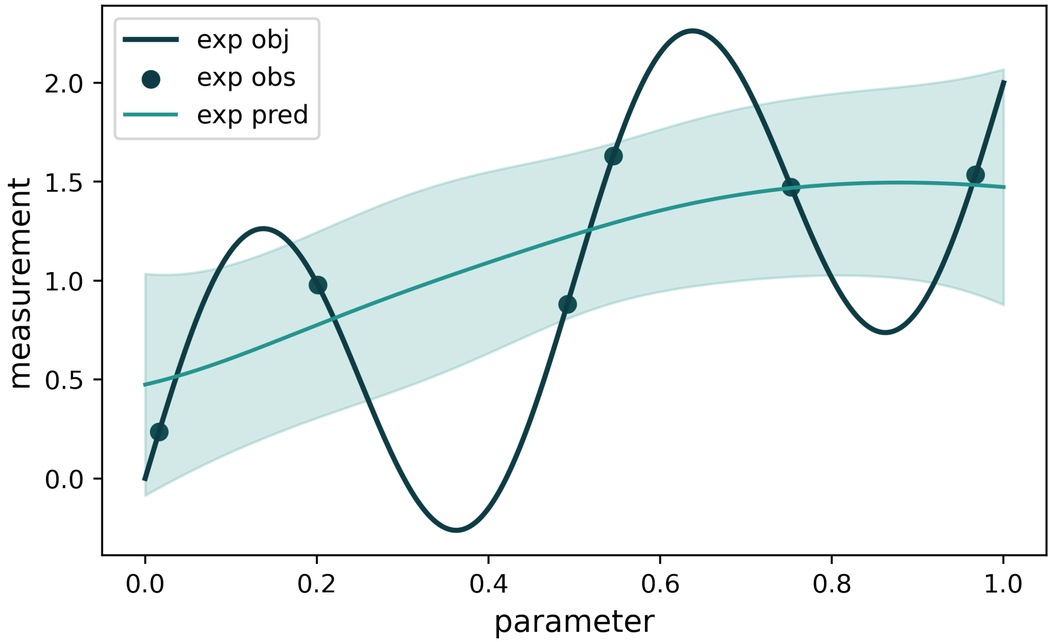
Single task Multi-task

Synthetic example

$$y_{\text{exp}} = \sin(4\pi x_{\text{exp}}) + 2x_{\text{exp}}$$

$$y_{\text{cheap}} = \frac{1}{2}\sin(4\pi x_{\text{cheap}})$$

Single task prediction



Multi-task Gaussian processes

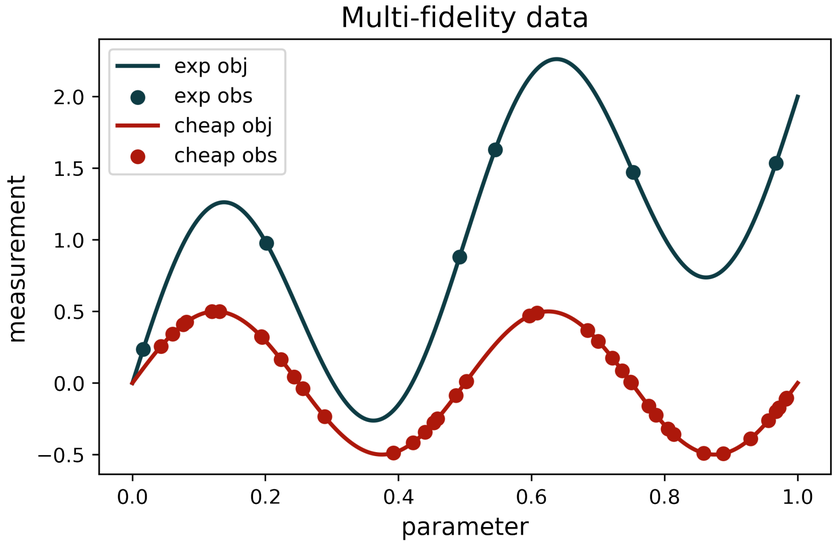
- Lets start from the beginning, with the multi-task Gaussian process model for BayesOpt, as implemented by Swersky *et* al. in 2013
- There have been many multi-fidelity strategies introduced since, including deep GPs, NN-based models, autoregressive GPs, to name a few

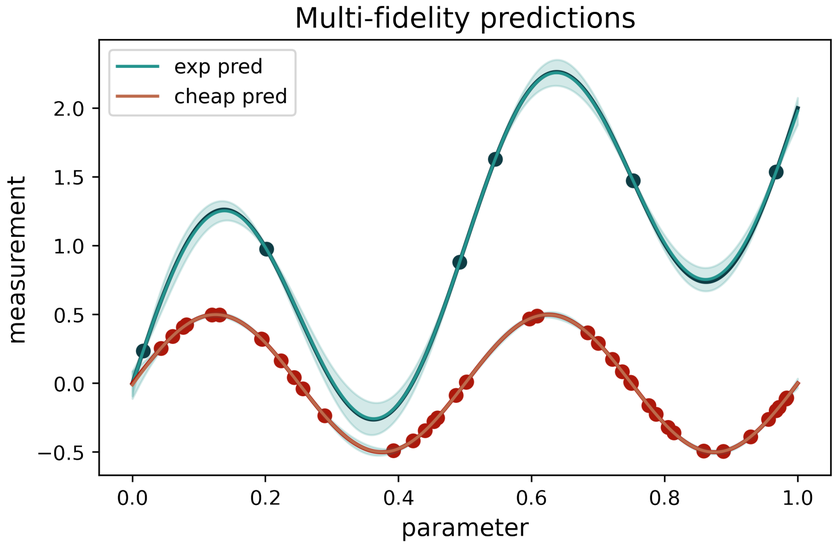
Assume we have T tasks, each with N_t observations, we then have the datasets

$$\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^{T} \qquad \mathcal{D}_t = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^{N_t}$$

$$k\left(\boldsymbol{x}, \boldsymbol{x}'\right) \longrightarrow k_{\text{multi}}\left((\boldsymbol{x}, t), (\boldsymbol{x}', t')\right) = k_t(t, t') \otimes k_x(\boldsymbol{x}, \boldsymbol{x}')$$

Single task Multi-task

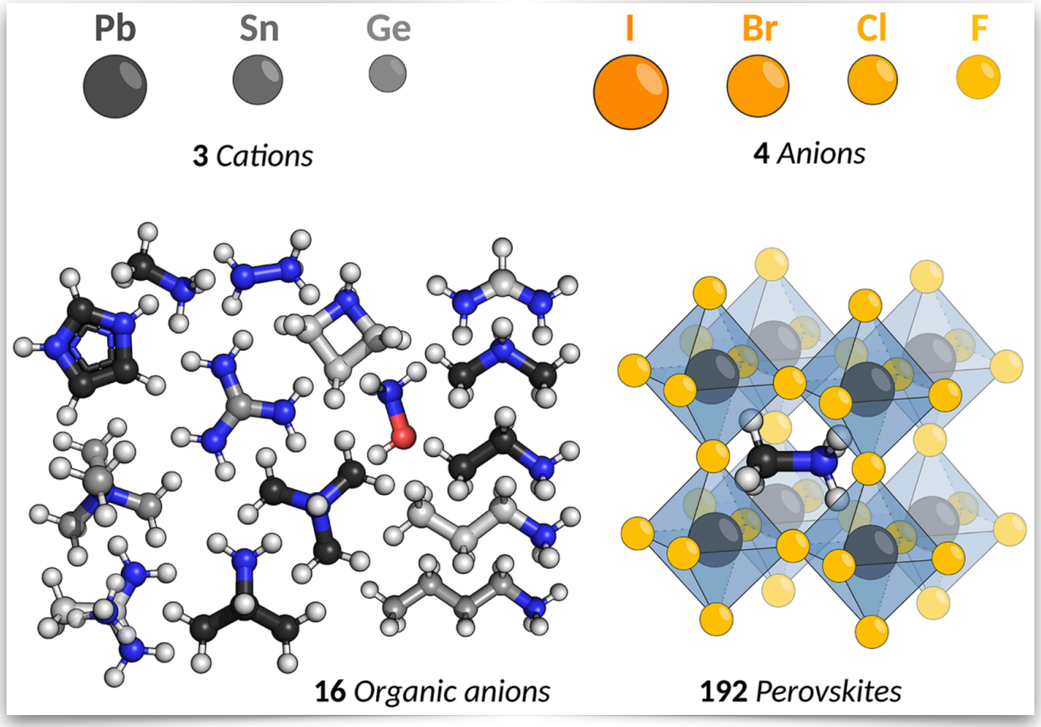




Multi-fidelity prediction/optimization of bandgaps of hybrid organic-inorganic perovskites

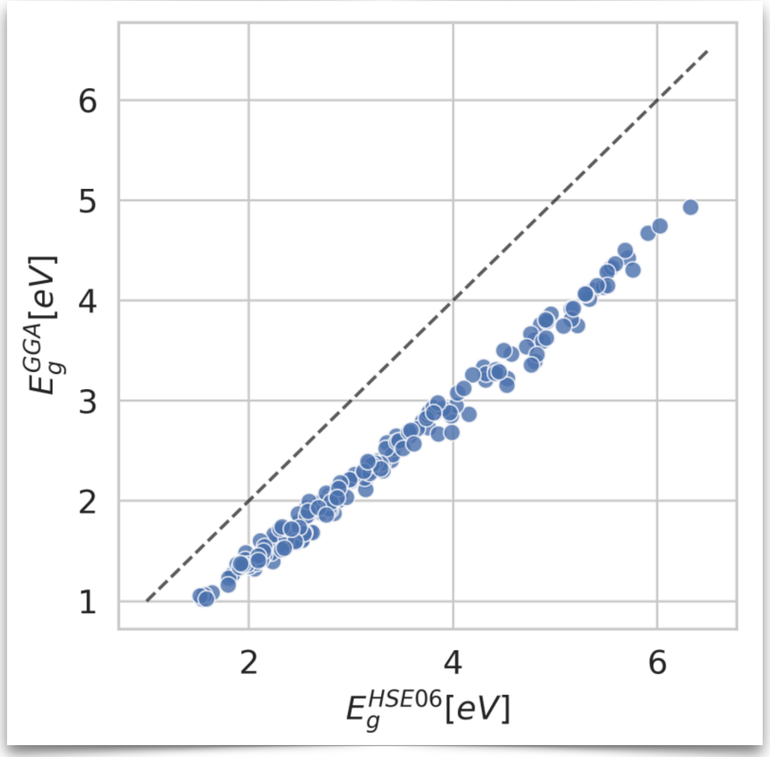
the matter lab

- Lets put together everything we learned so far and extend our Bayesian optimization code to handle multi-fidelity optimization
- We'll re-use the perovskites data, only this time, we'll include a second dataset obtained using a much less expensive computational method



Measurement fidelity levels

- low fidelity: GGA bandgaps (E_g^{GGA})
- high fidelity: HSE06 bandgaps (E_g^{HSE06})



C. Kim *et al.* Sci. *Data.* ***4***, 170057 (2017)

Second coding session

Seedless photoinitiated nanoparticle synthesis

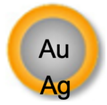
the matter lab

- design of gold nanoparticles with targeted spectroscopic properties
- integration of BayesOpt tools with automated microfluidic reactor and characterization, fully automated self-driving lab (MF-ML platform)

Taylor Wu

Prof. Kumacheva

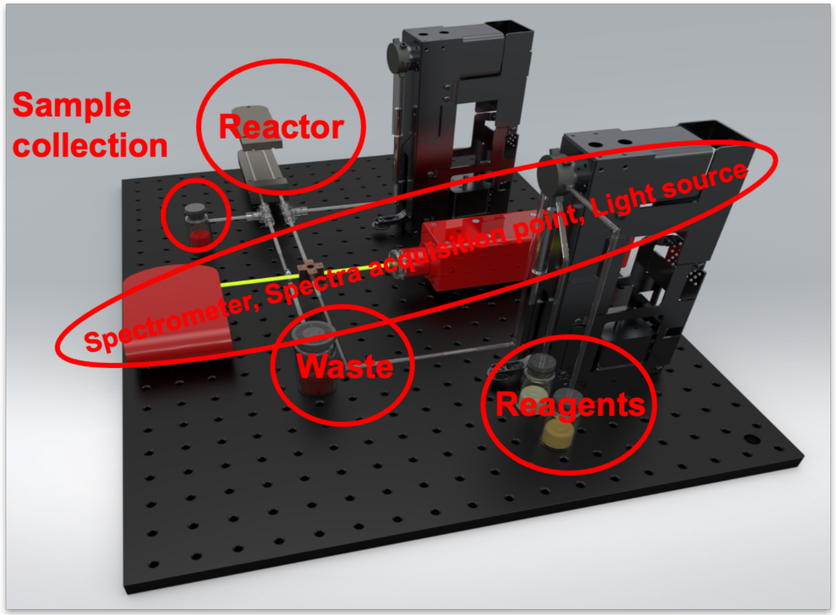
$$HAuCl_4 + AgNO_3 \xrightarrow[CTAB\ /\ H_2O]{I\text{-}2959 + h\nu}$$

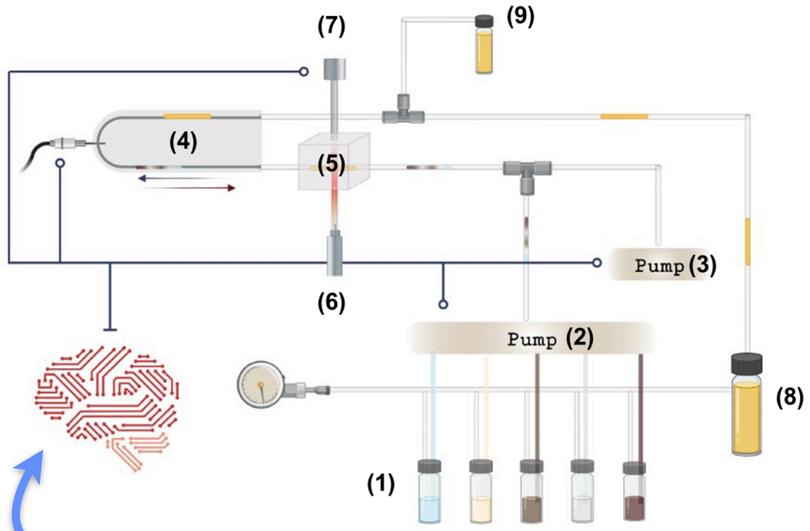


AuAg core-shell

AuAg alloy

Au nanorod





Gryffin, Chimera, known constraints

Parameters (7)

- concentration of reagents
- reaction time
- UV intensity
- mixing rate

Objectives (4)

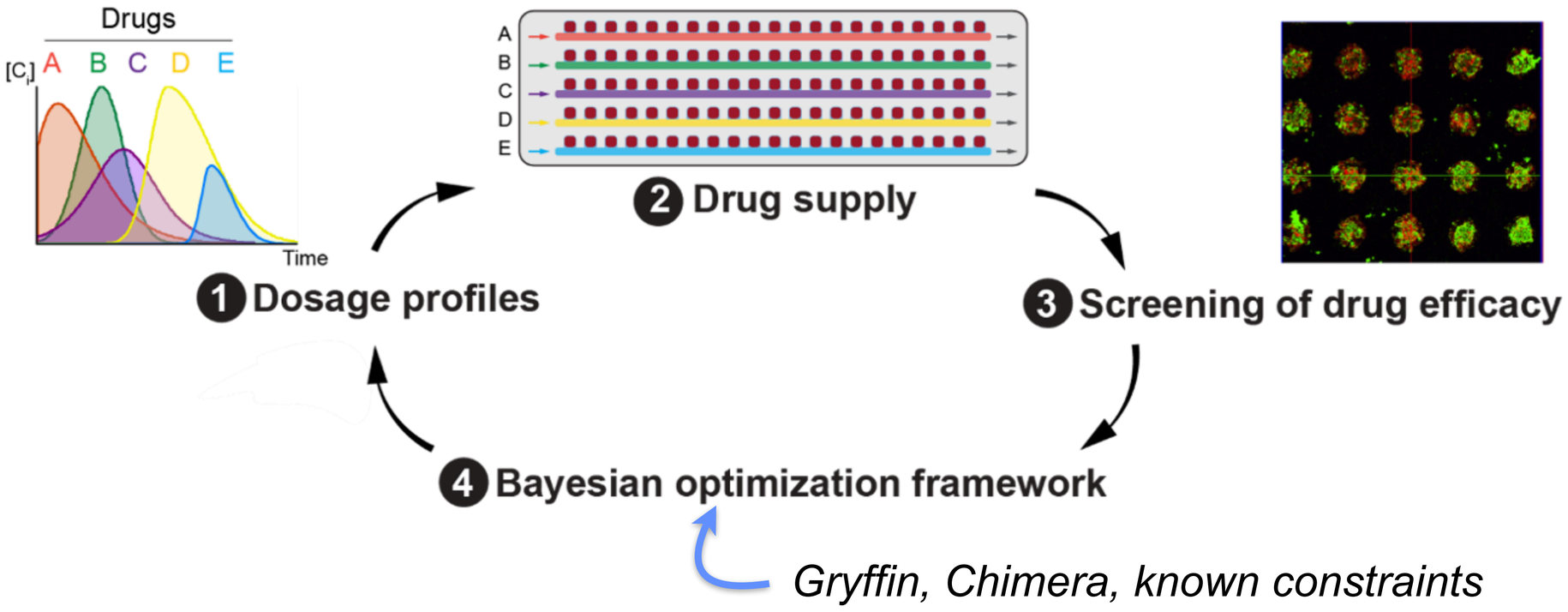
- spectroscopic characteristics of NPs

(1) Reagents (2) Dispenser (3) Oscillator (4) Reactor (5) Flow cell (6) Light source (7) Spectrometer (8) Waste (9) Sample collection outlet

in vitro design of combination cancer therapies

the matter lab

- Optimize for synergistic effects of multi-drug combinations (common breast cancer chemotherapy regime)
- Breast cancer spheroids subject to drugs in microfluidic array
- Treatment efficiency evaluated with live/dead assays based on fluorescence imaging




Dr. Ilya Yakavets


Sina Kheiri


Ella Rajaonson


Prof. Kumacheva

Parameters (5)
- drug doses (concentration)
- dose time
- dose duration

Objectives (2)
- combination index
- cell viability

Optimization of process parameters for accurate liquid dispensing with Chemspeed

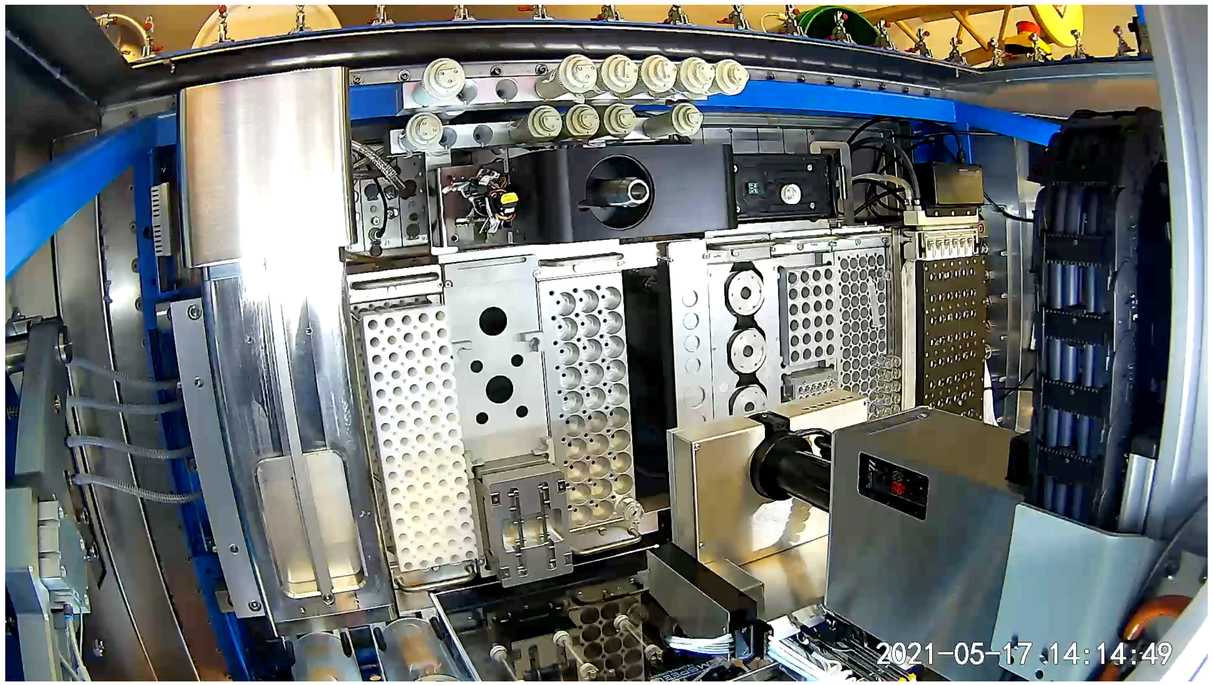
the matter lab

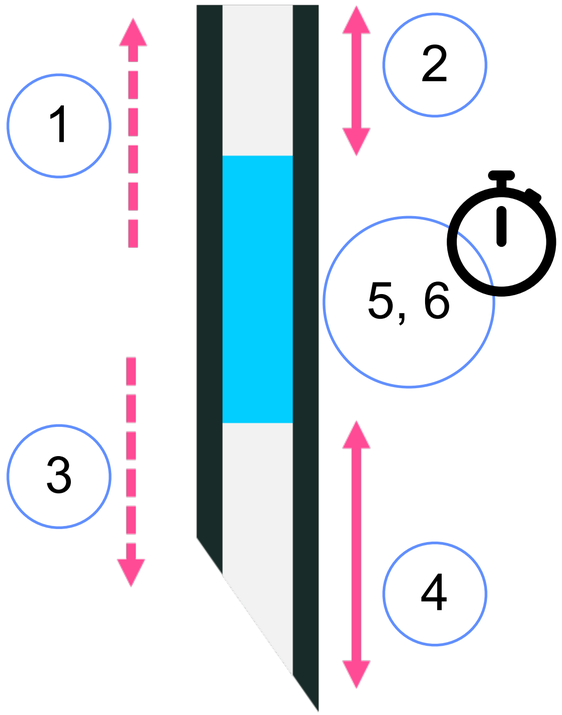
- Determine liquid specific process parameters for accurate and precise dispensing with Chemspeed robot
- BayesOpt framework interfaced with Python Chemspeed control
- Fully automated workflow enabled by balance placed inside Chemspeed


Dr. Martin Seifrid


Ella Rajaonson


Dr. Tony Wu





Parameters (6)

1. Aspiration speed (source)
2. Airgap volume
3. Dispense speed (destination
4. Post-airgap volume
5. Equilibration time (source)
6. Equilibration time (destinatio

Objectives (2)

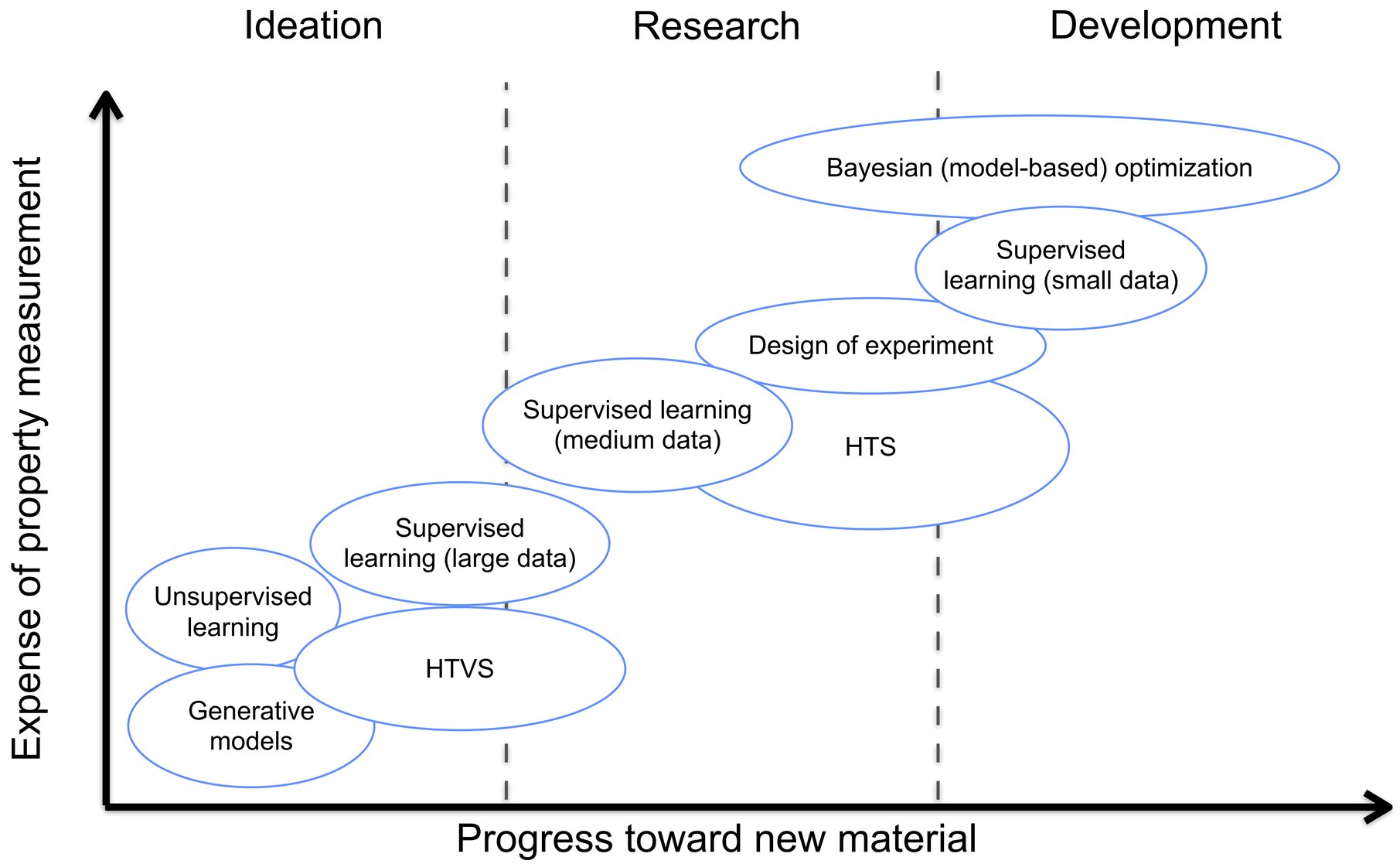
1. Minimize mean relative error
2. Minimize standard deviation


Thank you for your attention!
Questions?

Backup slides

Tools for data-driven scientific innovation

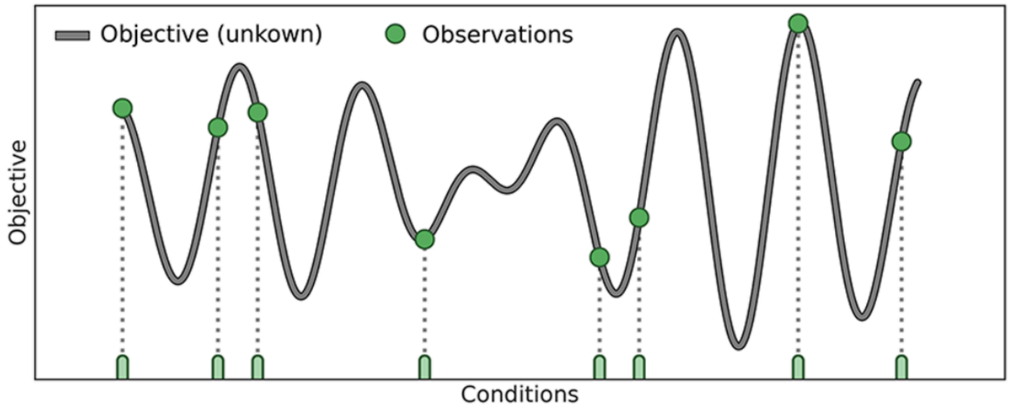
the matter lab



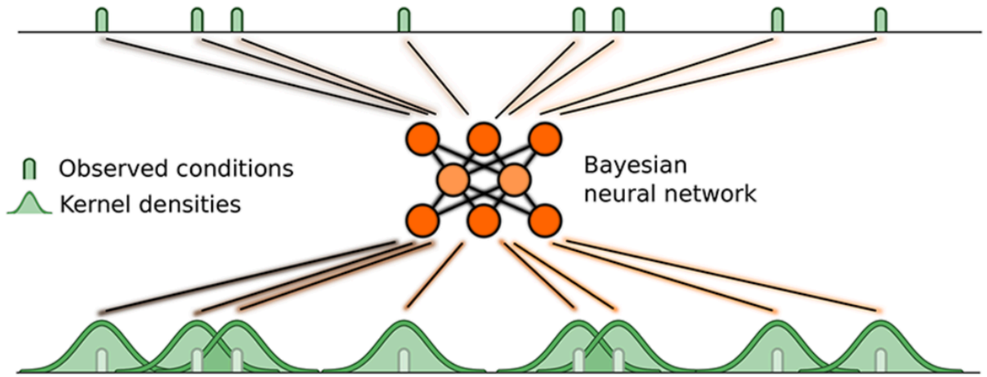
Phoenics: Continuous-valued parameters

the matter lab

A) Available observations and unknown objective

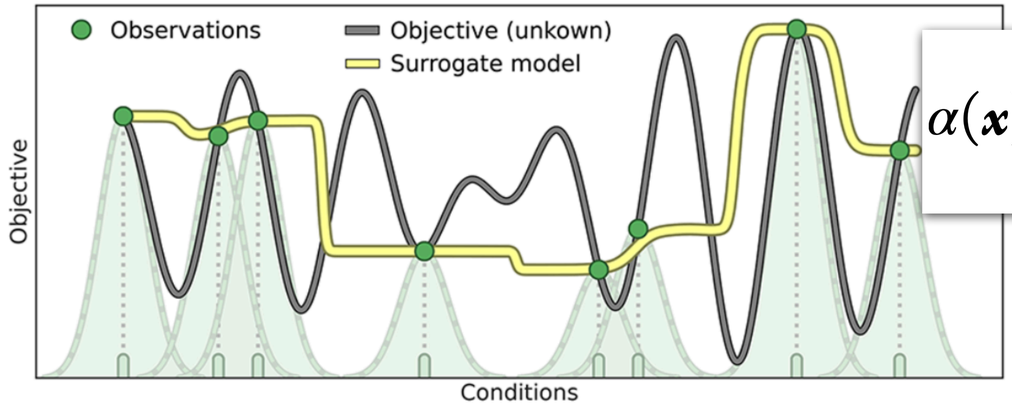


B) Estimation of kernel densities



$$p_k(\boldsymbol{x}) = \left\langle \sqrt{\frac{\tau_n}{2\pi}} \exp\left[-\frac{\tau_n}{2}(\boldsymbol{x} - \boldsymbol{x}_{\text{pred}}(\boldsymbol{\theta};\, \boldsymbol{x}_k))^2\right] \right\rangle_{\text{BNN}}$$

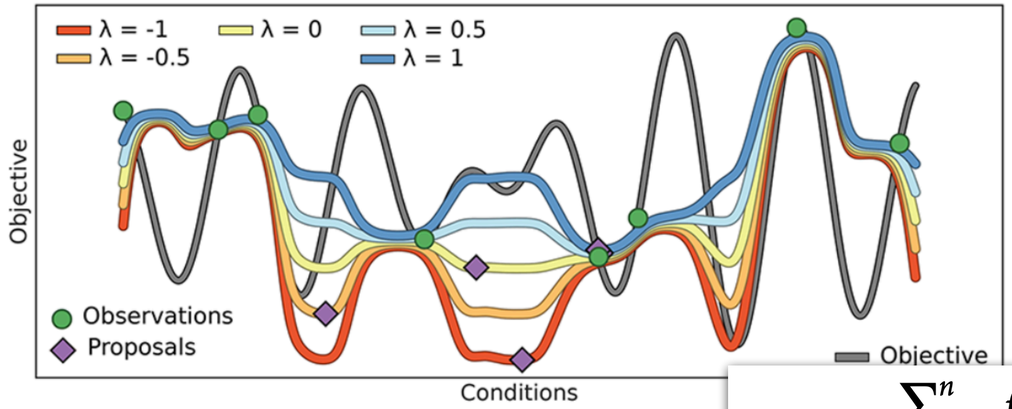
C) Constructing the surrogate model from the kernel densities



Surrogate

$$\alpha(\boldsymbol{x}) = \frac{\sum_{k=1}^{n} f_k p_k(\boldsymbol{x})}{\sum_{k=1}^{n} p_k(\boldsymbol{x})}$$

D) Constructing acquisition functions with different sampling strategies

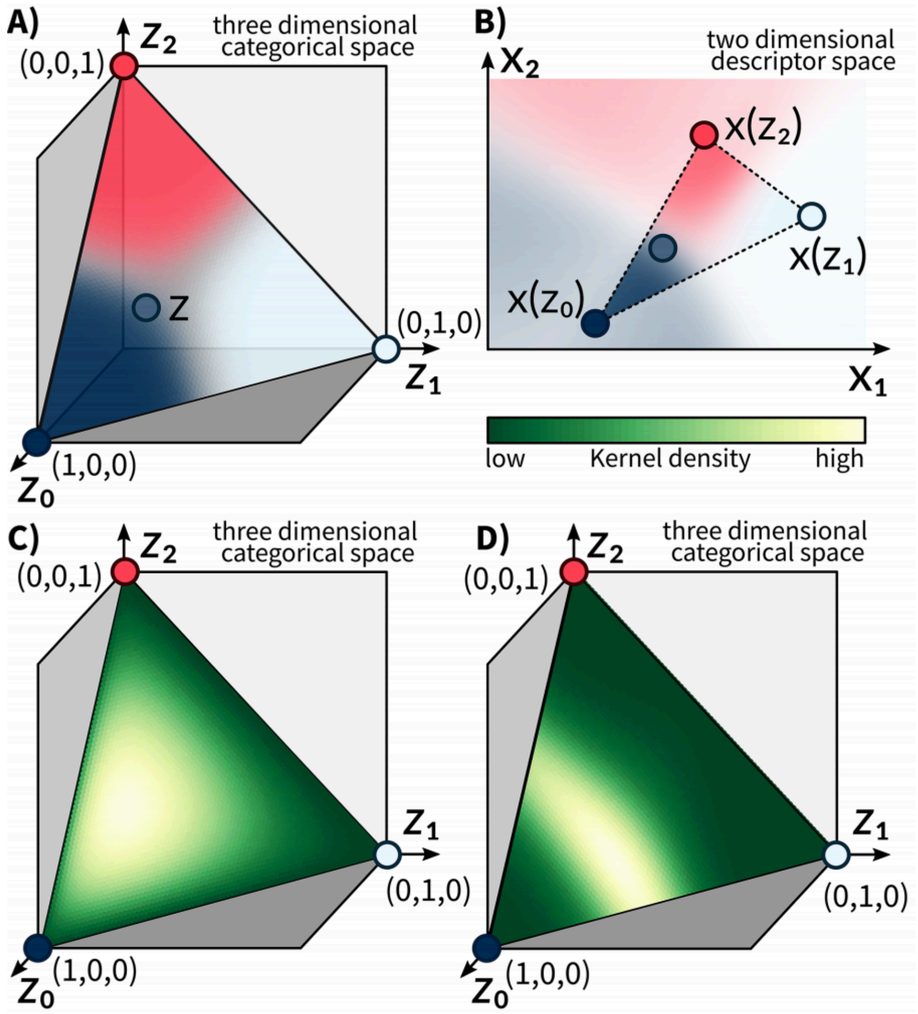


Acquisition

$$\alpha(\boldsymbol{x}) = \frac{\sum_{k=1}^{n} f_k p_k(\boldsymbol{x}) + \lambda p_{\text{uniform}}(\boldsymbol{x})}{\sum_{k=1}^{n} p_k(\boldsymbol{x}) + p_{\text{uniform}}(\boldsymbol{x})}$$

Gryffin: Categorical-valued parameters

the matter lab



Probability density

$$p_{\pi,\tau}(\mathbf{z}) = \Gamma(n)\tau^{n-1}\prod_{k=1}^{n}\left(\frac{\pi z_k^{-\tau-1}}{\sum_{i=1}^{n}\pi z_i^{-\tau}}\right)$$

Sampling

$$z_k = \frac{\exp\left(\left(\log\pi + g_k\right)/\tau\right)}{\sum_{i=1}^{n}\exp\left(\left(\log\pi + g_k\right)/\tau\right)}$$

Acquisition

$$\alpha(\mathbf{z}) = \frac{\sum_{k=1}^{n} f_k p_k(\mathbf{z}) + \lambda p_{\text{uniform}}(\mathbf{z})}{\sum_{k=1}^{n} p_k(\mathbf{z}) + p_{\text{uniform}}(\mathbf{z})}$$

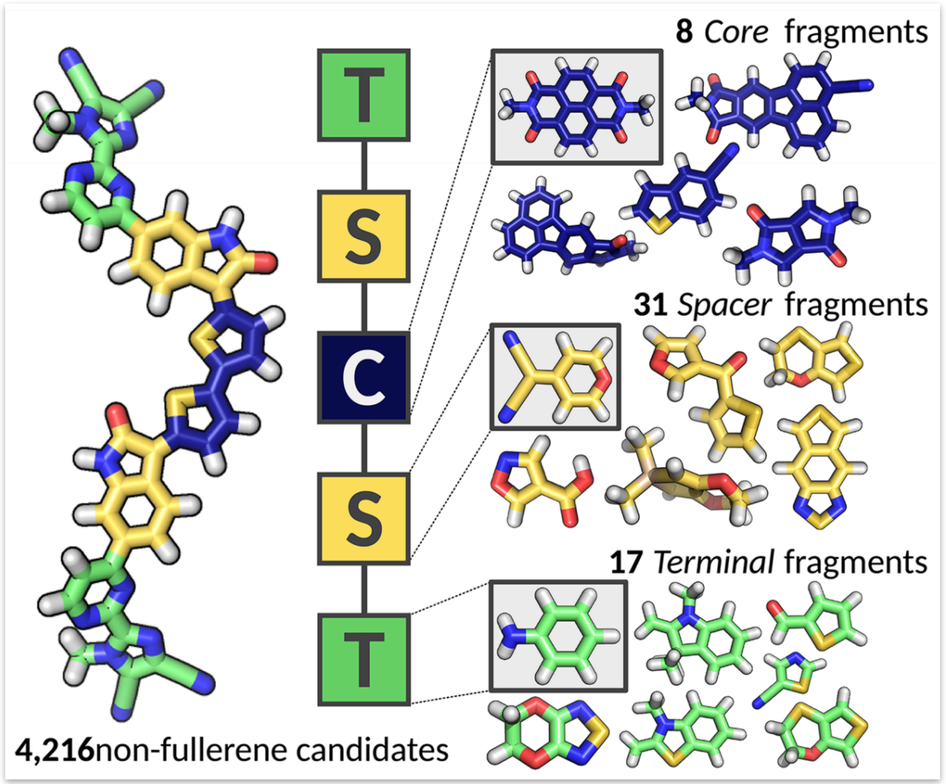
Naive Gryffin: Equal covariances between options

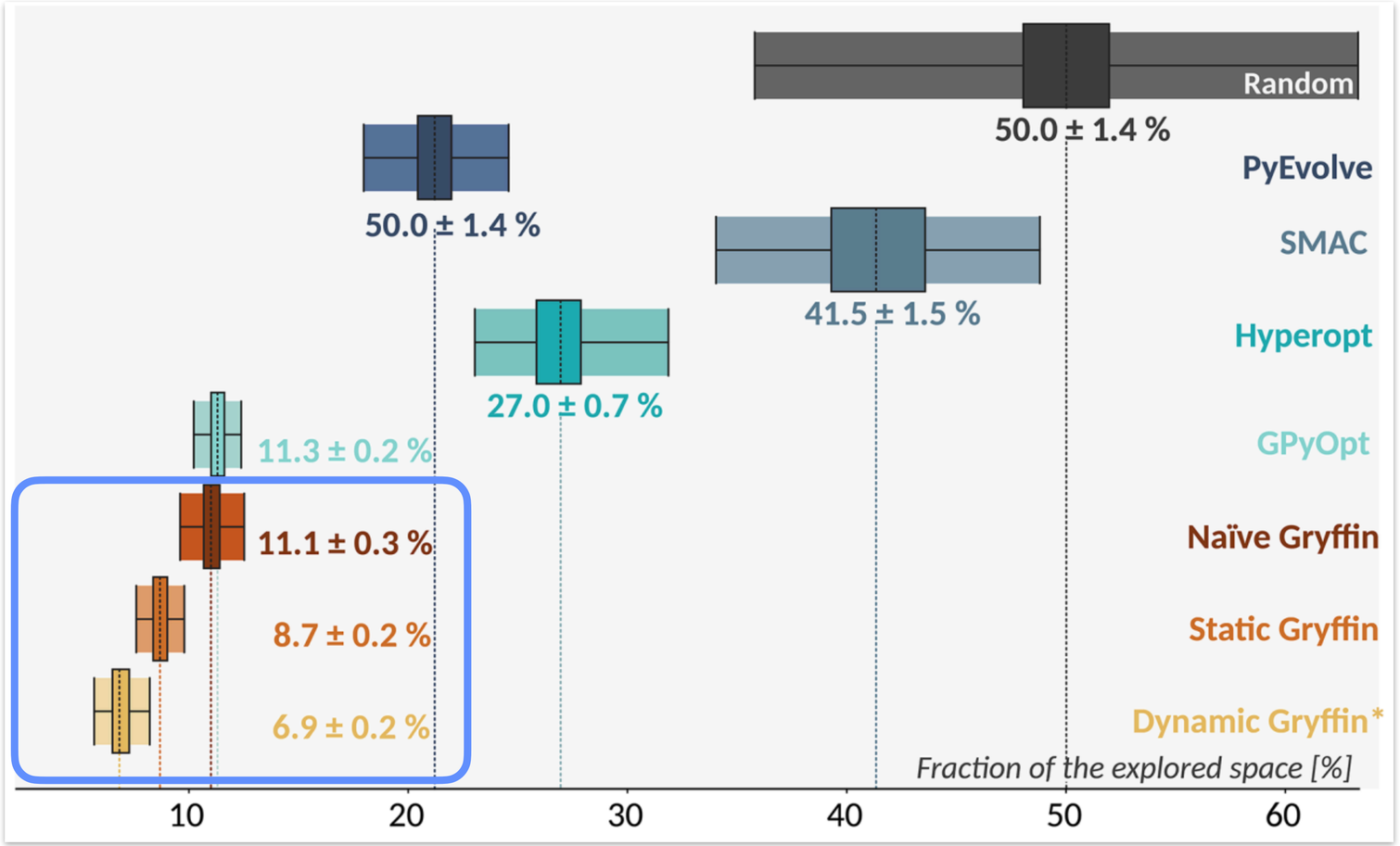
Static Gryffin: Refined metric using descriptors

Dynamic Gryffin: Refined descriptors using perceptron

Gryffin: Opt. of PCEs of non-fullerene acceptors

the matter lab





“Known” and “unknown” constraints

the matter lab

Known constraints

- constraints on parameters known *a priori* by the scientist
- physical constraints on implementation, safety considerations, user preference, prior knowledge, etc.

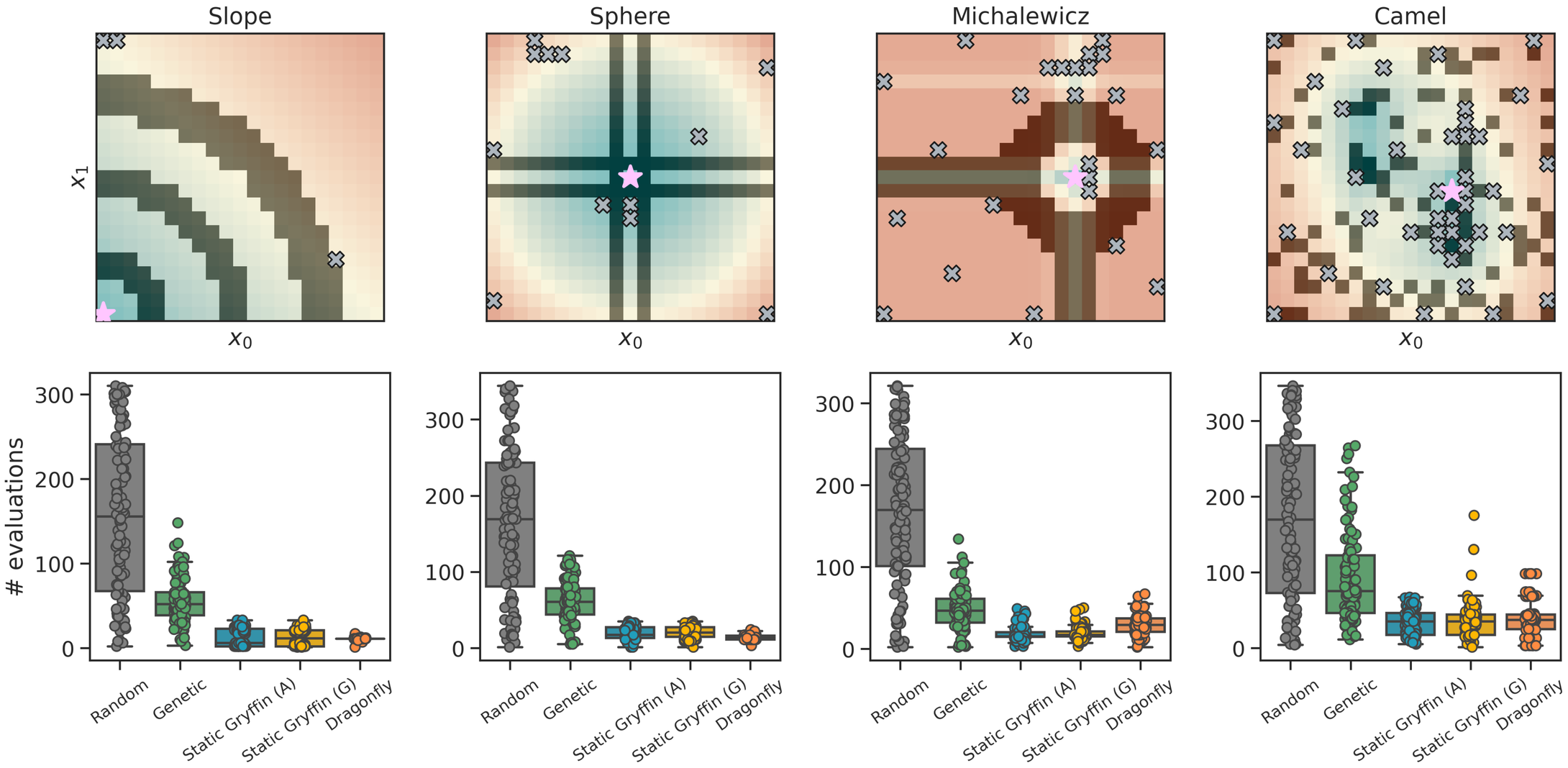
E.g.
- glassware has max volume, s.t. vol(A) + vol(B) < 20 mL
- temperature bounds depend on solvent identity, 10 < T < 100°C if H_2O, 10 < T < 66°C if THF

Unknown constraints

- unexpected parameter constraints that are identified “on-the-fly”
- parameter settings that essentially return “NaN” values

E.g.
- unexpected experimental failures
- “abandoned” syntheses
- unstable molecules or materials
- insufficient conditions for property measurements

Known constraints: Synthetic tests



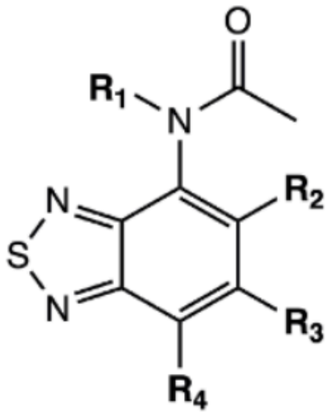
Known constraints: Design of redox-active materials for flow batteries

the matter lab

- Design of redox-active materials for flow batteries with synthetic accessibility constraints

a Design space

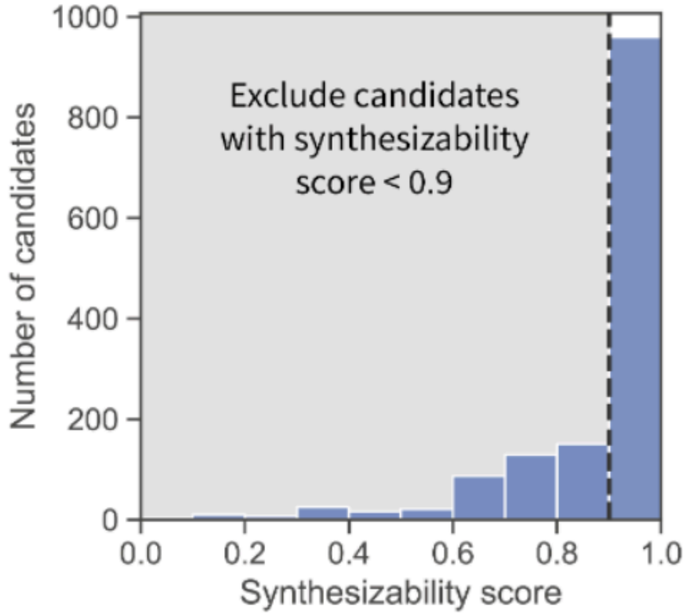
2 R_1 * 8 R_2 * 8 R_3 * 12 R_4 = 1408 candidates



R_1 = -CCOC, $-CF_2CF_3$
R_2, **R_3** = $-CH_3$, $-CF_3$, -CN, $-OCH_3$, $-OCF_3$, $-SCH_3$, $-SCF_3$, $-N(CH_3)_2$
R_4 = $-CH_3$, $-CF_3$, -CN, $-OCH_3$, $-OCF_3$, $-SCH_3$, $-SCF_3$, $-N(CH_3)_2$, $-NO_2$, $-SO_2N(CH_3)_2$

Agarwal *et al. Chem. Mater.* **33**, 8133-8144 (2021)

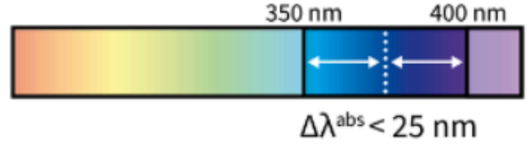
b Constraints



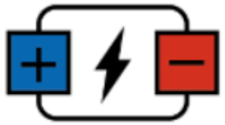
Thakkar *et al. Chem. Sci.* **12**, 3339-3349 (2021)

c Objectives

1 Achieve absorption wavelength (λ^{abs}) in 350–400 nm range



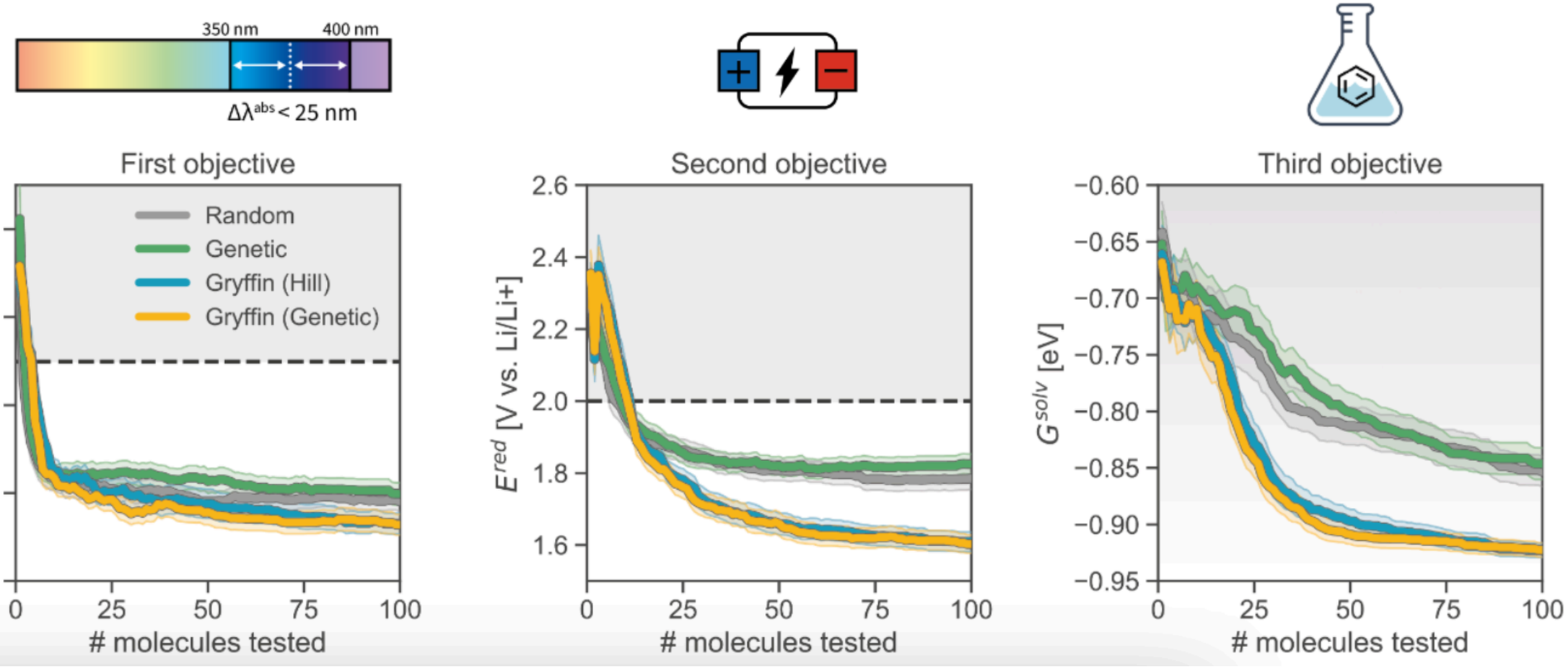
2 Achieve reduction potential (E^{red}) < 2.04 V



3 Minimize solvation free energy (G^{solv})

Known constraints: Design of redox-active materials for flow batteries

the matter lab



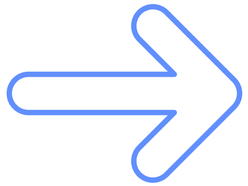
Unknown constraints: Inferring the constraint function on-the-fly

the matter lab

- constraint function, $c(x)$, is now *a priori* unknown
- $c(x)$ must be resolved by sequential measurement, much like objective function $f(x)$
- Approach: construct a Bayes classifier based on *Gryffin* (Gumbel-softmax) *Phoenics* (Gaussian) kernel densities
- Model $f(x)$ and $c(x)$ at the same time, then combine their contributions into a single acquisition function

Bayes' theorem

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$



Bayes classifier

$$P(\text{feasible}|\boldsymbol{x}) \propto P(\boldsymbol{x}|\text{feasible})\,P(\text{feasible})$$

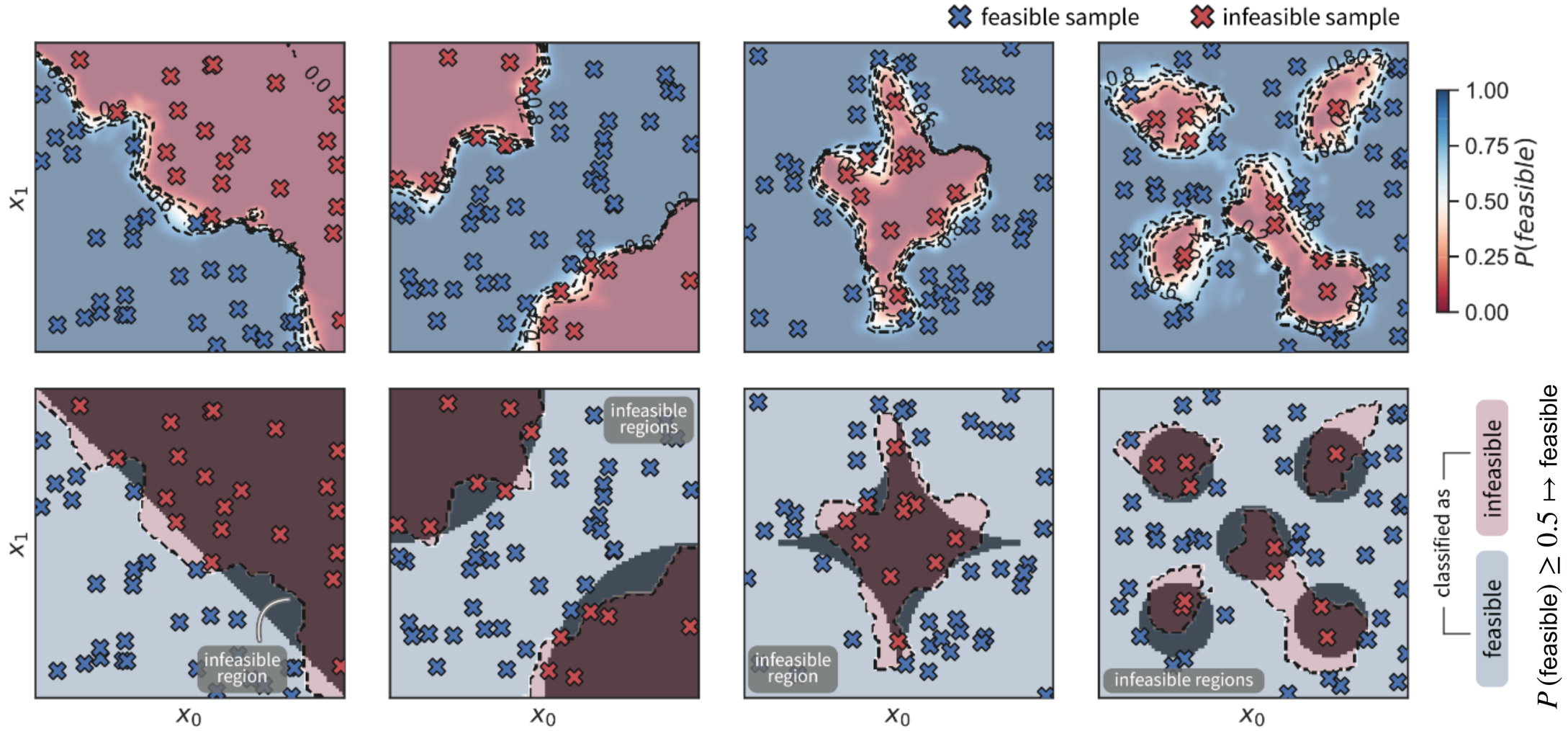
Prior, fraction of feasible samples observed

Likelihood, from kernel density estimate

Unknown constraints: Inferring the constraint function on-the-fly

the matter lab

- Demonstration of Bayes classifier learning constraints on 2-dimensional surfaces



Unknown constraints: Feasibility aware acquisition functions

the matter lab

- Model $f(x)$ and $c(x)$ at the same time, then combine their contributions into a single acquisition function
- Testing 3 distinct approaches for combined acquisition functions

Feasibility-weighted acquisition (FWA)

$$\alpha\left(\boldsymbol{x}\right) \times P\left(\text{feasible}|\boldsymbol{x}\right)$$

R. B. Gramacy, *et al.* arXiv:11004.402 **[stat.ME]** (2010)

M. A. Gelbart, *et al.* arXiv:1403.5607 **[stat.ML]** (2014)

Feasibility-interpolated acquisition (FIA)

$$\left(1 - c^t\right) \times \alpha\left(\boldsymbol{x}\right) + c^t \times P\left(\text{feasible}|\boldsymbol{x}\right)$$

$$c = N_{\text{infeas}}/N_{\text{tot}} \; ; \quad t \in \mathbb{R}_+$$

Feasibility-constrained acquisition (FCA)

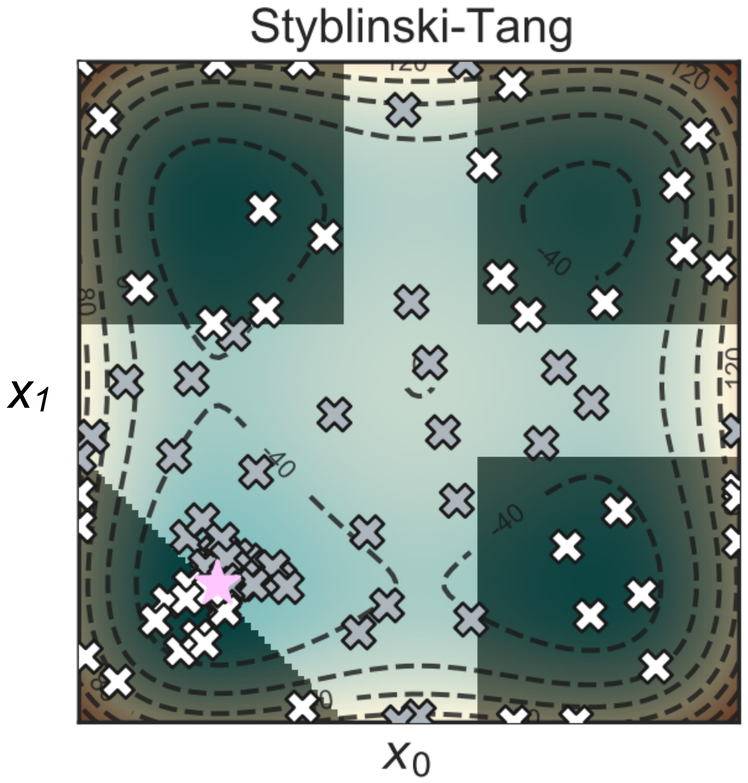
$$\alpha\left(\boldsymbol{x}\right)$$

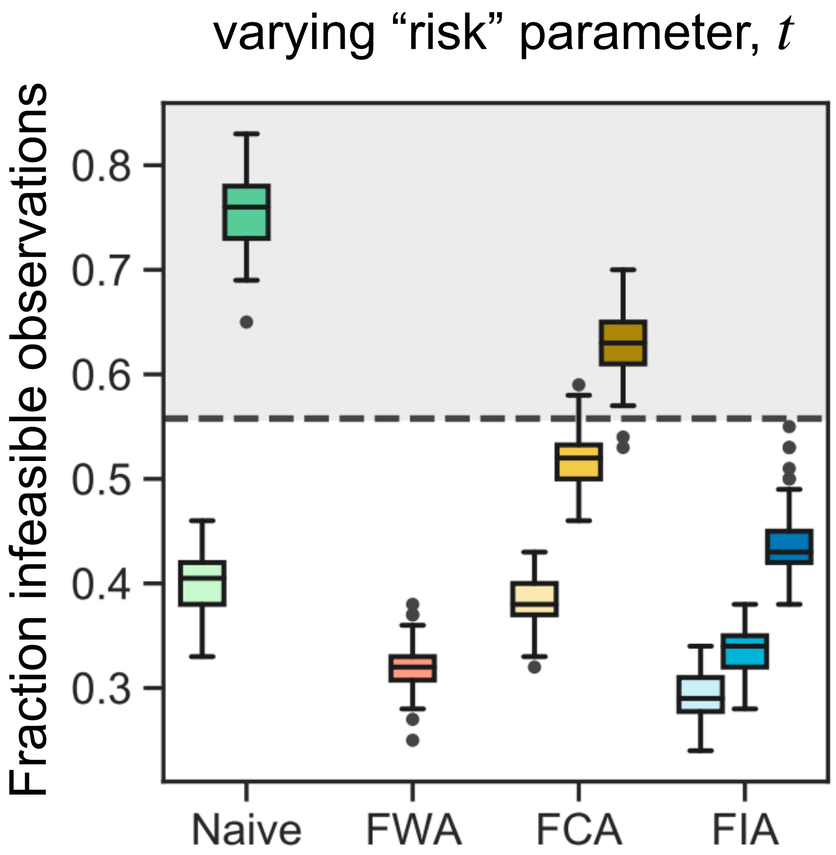
$$\text{s.t.} \;\; P\left(\text{feasible}|\boldsymbol{x}\right) > t \; ; \quad t \in [0, 1]$$

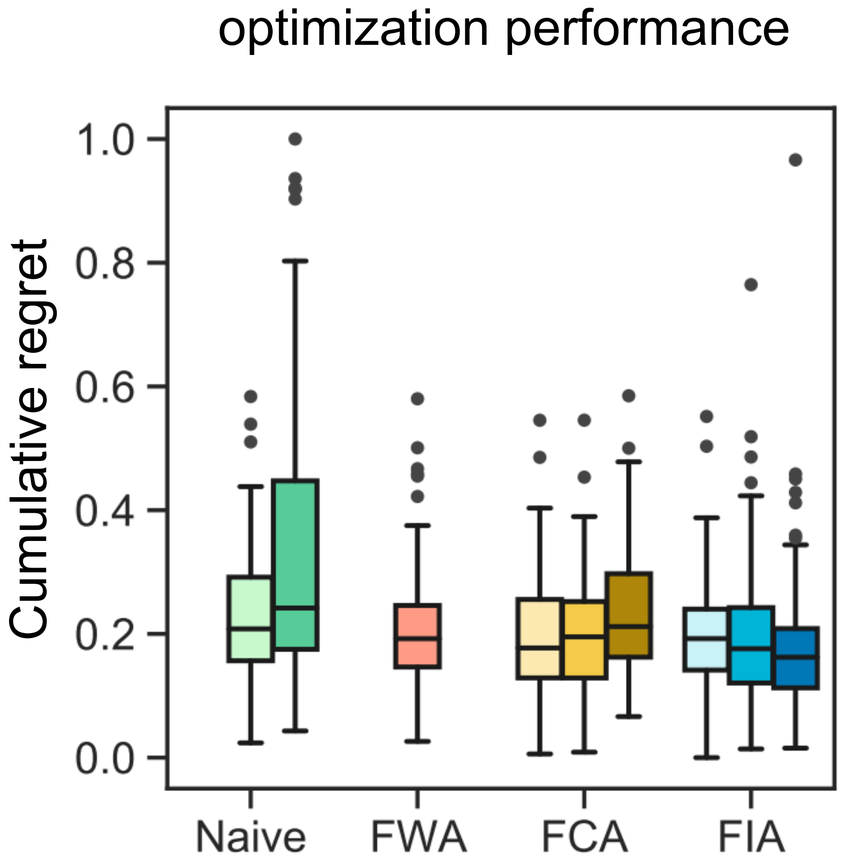
A. Candelieri, *J. Glob. Optim.* **79**, 281-303 (2019)

Unknown constraints: Synthetic tests (continuous parameters)

the matter lab





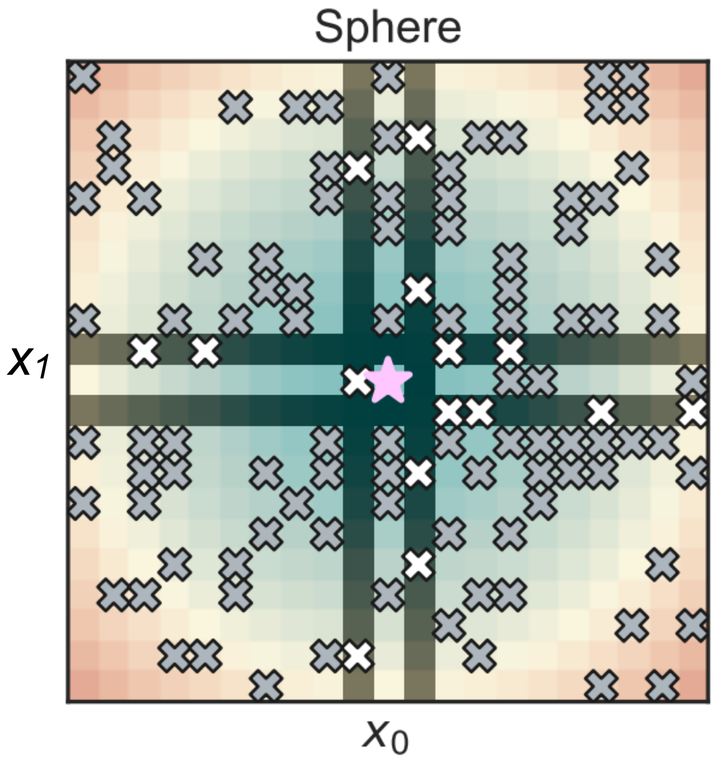


FCA (left to right): $t = \{0.8, 0.5, 0.2\}$

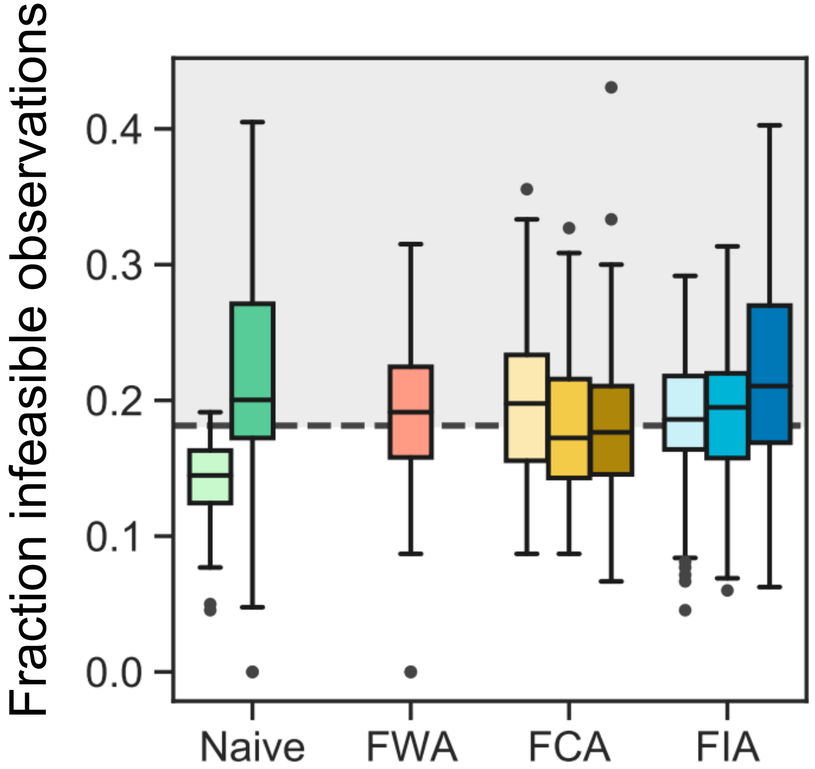
FIA (left to right): $t = \{0.5, 1, 2\}$

Unknown constraints: Synthetic tests (categorical tests)

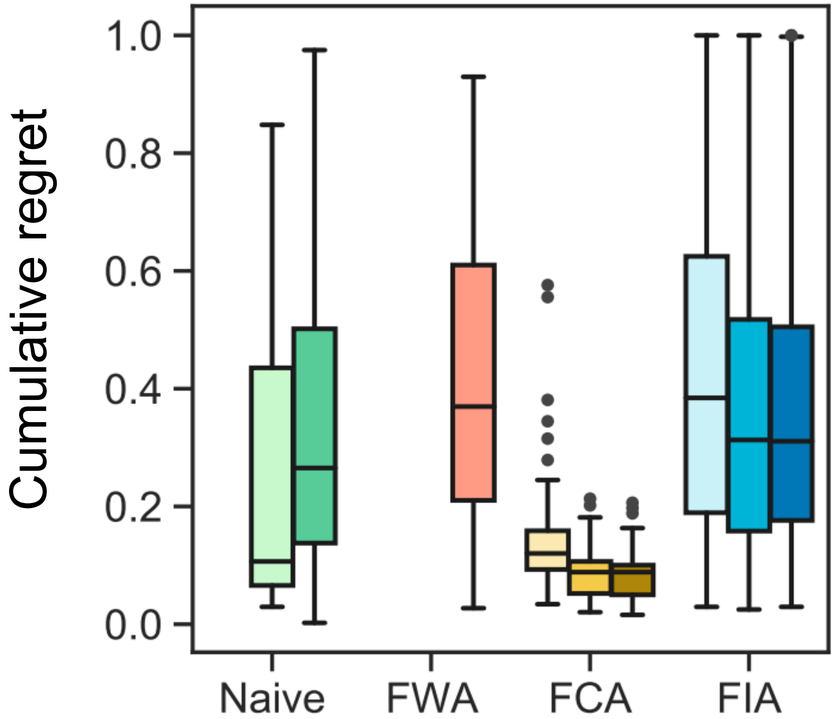
the matter lab



varying "risk" parameter, t



optimization performance



FCA (left to right): $t = \{0.8, 0.5, 0.2\}$

FIA (left to right): $t = \{0.5, 1, 2\}$

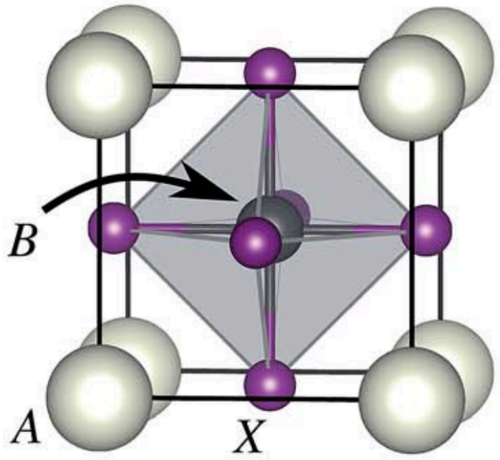
Unknown constraints: Computational design of stable HOIP perovskites

the matter lab

Stable hybrid organic–inorganic halide perovskites for photovoltaics from *ab initio* high-throughput calculations†

Sabine Körbel, ‡[a] Miguel A. L. Marques [b] and Silvana Botti *[ac]

111/1276 stable
$A^+B^{2-}X_3^-$ materials



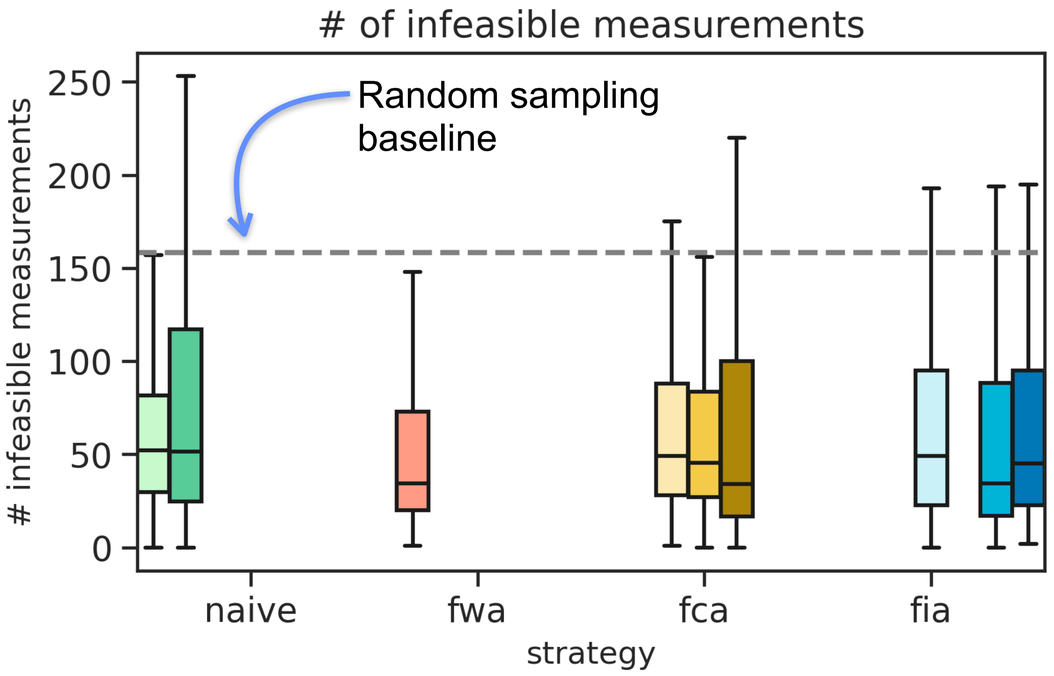
Parameters
- molecular organic cation, A (11 options)
- divalent metal, B (29 options)
- halogen, X (4 options)

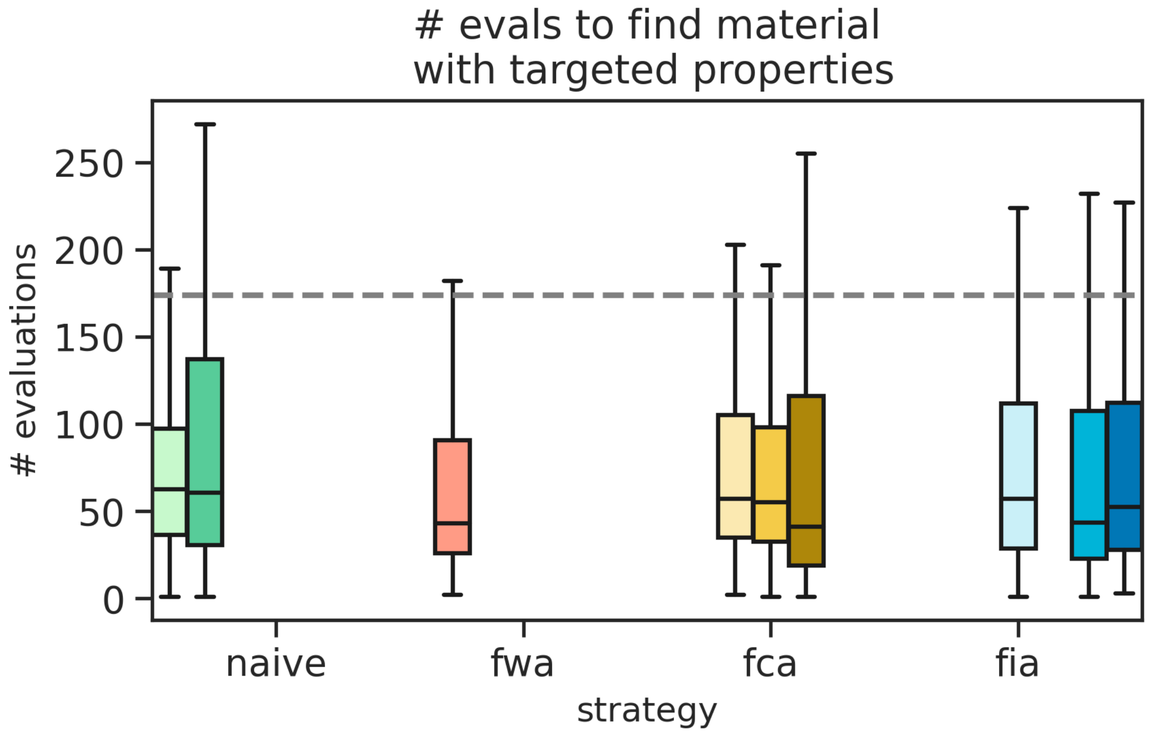
Objectives
- Achieve bandgap of 1.25 ± 0.5 eV
- Minimize effective mass (threshold of 4.0)

- **9/1276** materials that are stable *and* satisfy both objectives

Unknown constraints: Computational design of stable HOIP perovskites

the matter lab

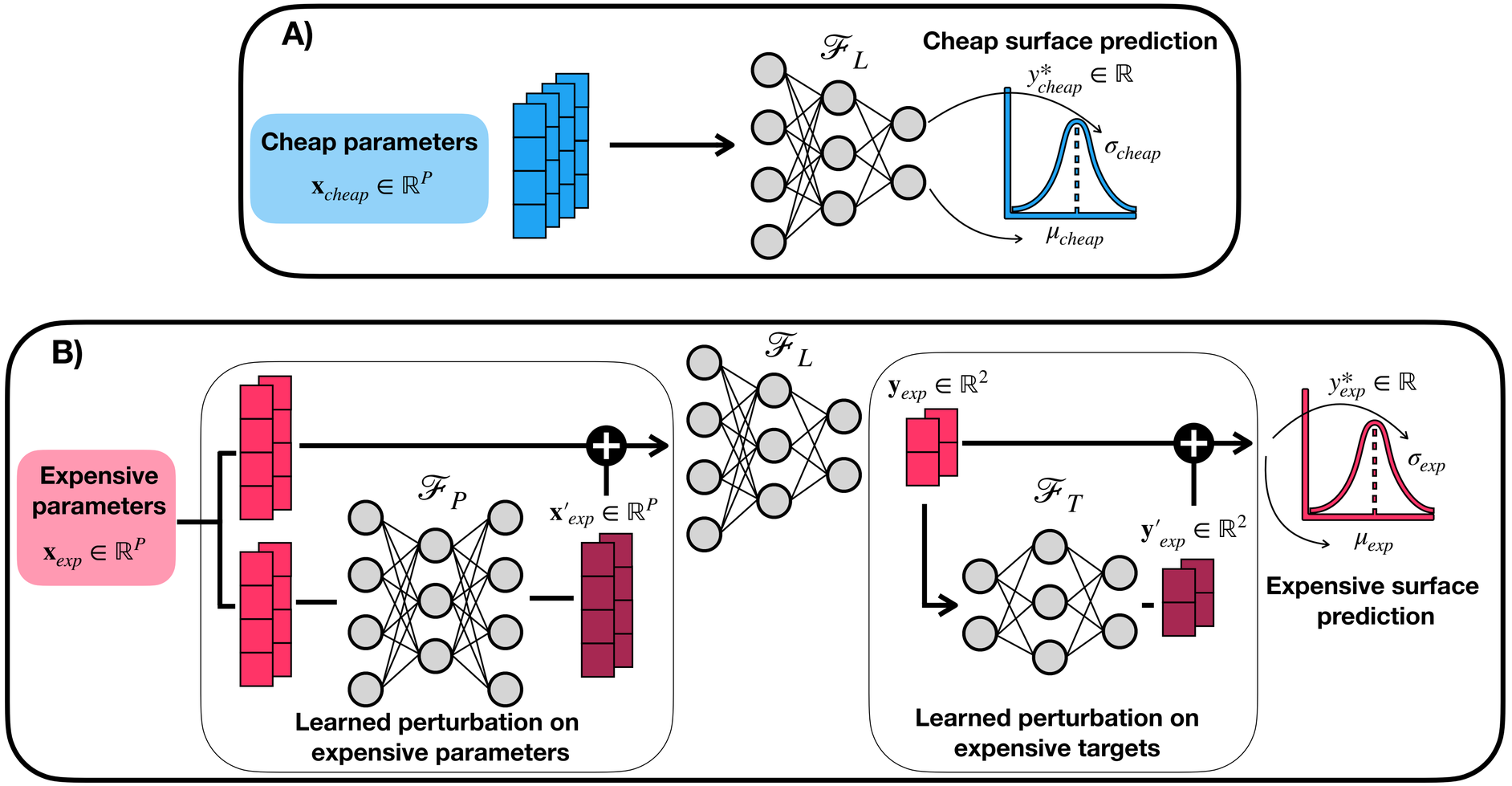




–	Random	N-0	N-FIA-1000	FWA	FCA-0.2	FCA-0.5	FCA-0.8	FIA-0.5	FIA-1	FIA-2
# nans	158.5 ± 0.7	51.5 ± 0.9	52.0 ± 0.5	**34.5 ± 0.8**	**34.0 ± 1.0**	45.5 ± 0.5	49.0 ± 0.6	49.0 ± 0.8	**34.5 ± 0.6**	45.0 ± 0.6
# evals	174.0 ± 0.7	62.5 ± 0.6	60.5 ± 1.0	**43.0 ± 0.8**	**41.0 ± 1.2**	55.0 ± 0.6	57.0 ± 0.6	57.0 ± 0.9	43.5 ± 0.7	52.5 ± 0.7

Gemini: Model architecture

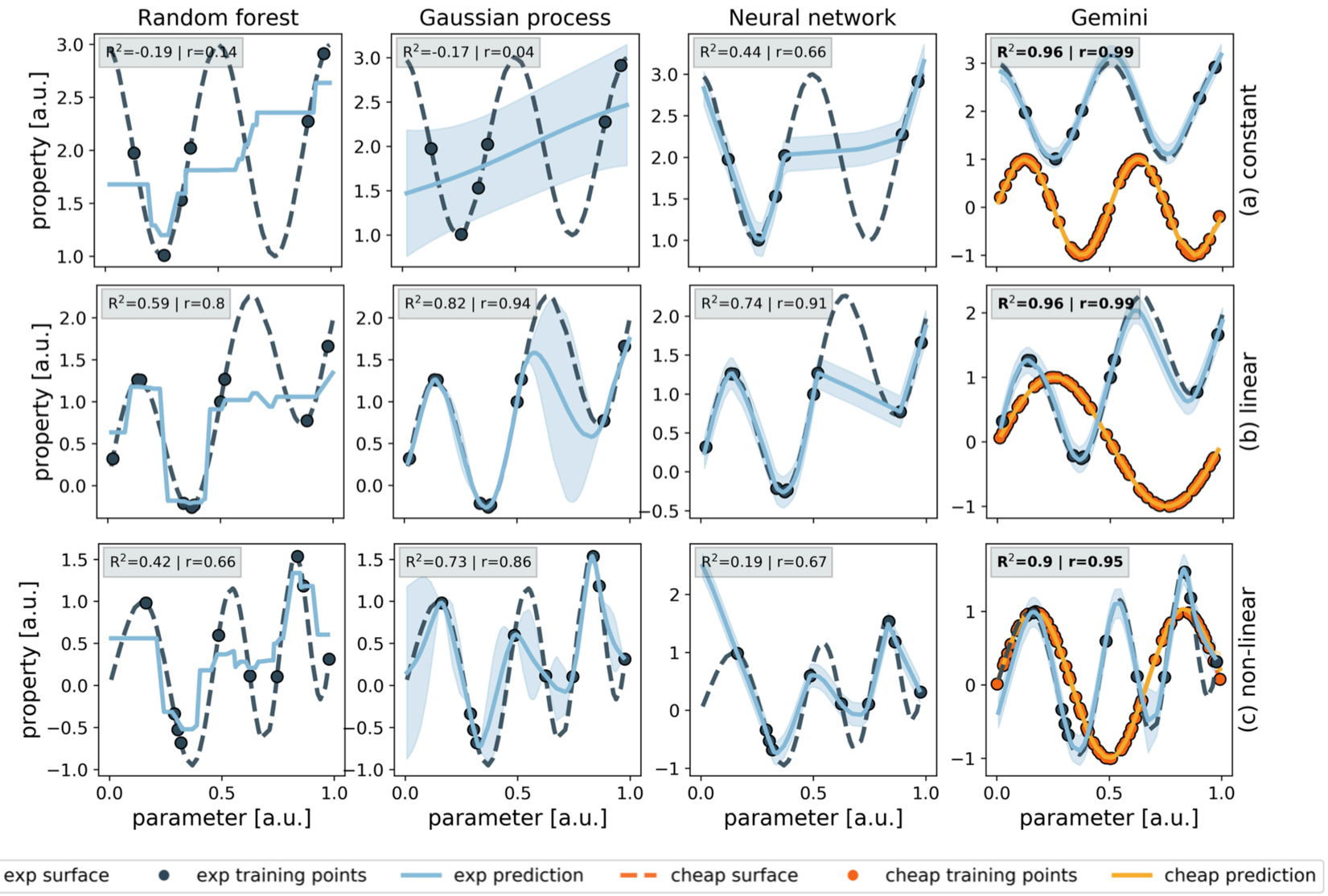
the matter lab



Gemini: 1-dimensional function regression

the matter lab

- “Single task” ML models compared to Gemini with access to “cheap” *and* “expensive” data
- “cheap” and “expensive” surfaces differ by constant, linear and non-linear biases
- Gemini reconstructs “expensive” surface with greater accuracy than the single task models



Gemini: Closed-loop workflow

the matter lab

Gemini can inform the acquisition function of *Gryffin*.

ρ provides a measure of *trust* or *anti-trust* in the predictive model

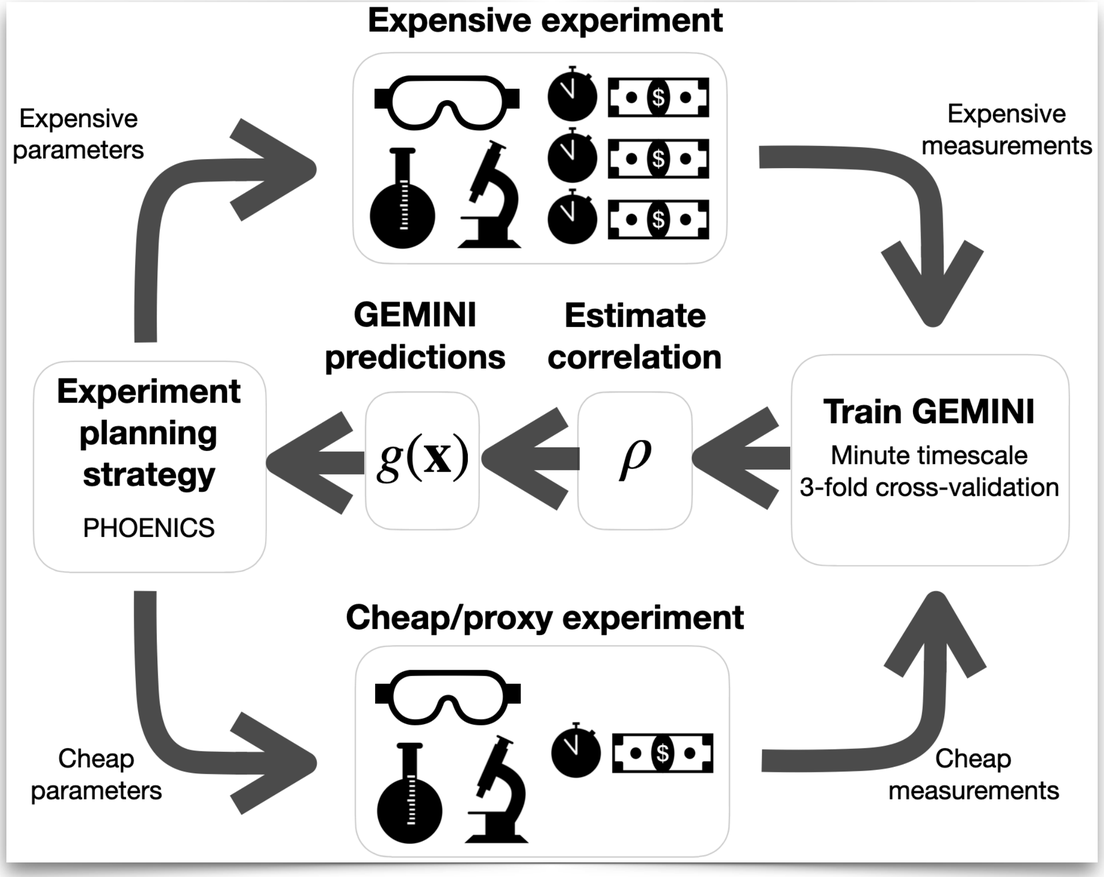
GRYFFIN only

$$\alpha(\boldsymbol{x}) = \frac{\sum_{k=1}^{n} f_k p_k(\boldsymbol{x}) + \lambda p_{\text{uniform}}(\boldsymbol{x})}{\sum_{k=1}^{n} p_k(\boldsymbol{x}) + p_{\text{uniform}}(\boldsymbol{x})}$$

GRYFFIN + GEMINI

$$\alpha_g(\boldsymbol{x}) = \frac{\sum_{k=1}^{n} f_k p_k(\boldsymbol{x}) + \lambda p_{\text{uniform}}(\boldsymbol{x}) + \rho g(\boldsymbol{x})}{\sum_{k=1}^{n} p_k(\boldsymbol{x}) + p_{\text{uniform}}(\boldsymbol{x}) + 1}$$

Combined workflow for GEMINI + GRYFFIN



Augmenting `Gryffin` with `Gemini`

the matter lab

Beginner users

Scalable multi-fidelity Bayesian optimization

Gemini's predictions of expensive-to-evaluate objective functions can be used to reduce the number of expensive black-box evaluations necessary to achieve a desired target value.

The deep Bayesian optimizer Gryffin currently supports Gemini as a built-in predictive model. After installing Gemini and Gryffin,

```
from gryffin import Gryffin

# instantiate Gryffin
gryffin = Gryffin('config_file.json')

# optimization loop
while num_eval < budget:

    samples = gryffin.recommend(observations,
                                proxy_observations)
```

The Gryffin config file must include a section specifying the predictive model, i.e.

```
...
"predictive_model": {
                "model_kind": "gemini"
},
...
```

Expert users

Alternatively, you can train Gemini in an external manner, this gives the user greater flexibility in their expreiment. Gryffin allows for the optional passing of a callable object to its `recommend` method.

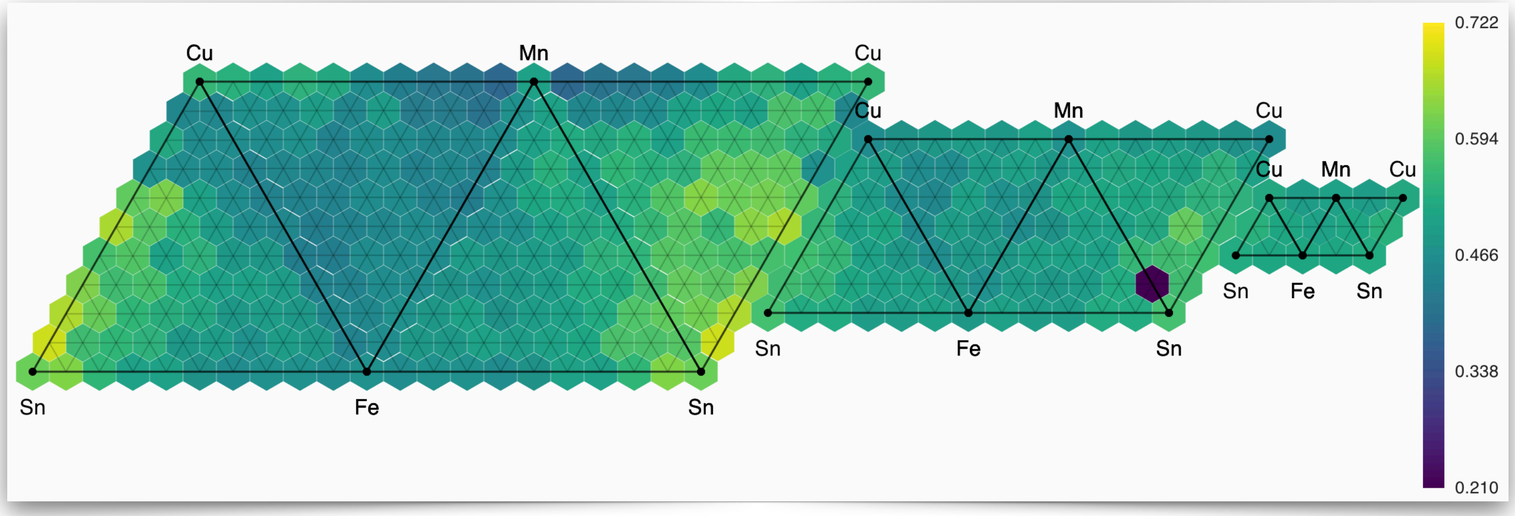
```
from gryffin import Gryffin
from gemini import GeminiOpt as Gemini

# instantiate Gryffin
gryffin = Gryffin('config_file.json')

# instantiate Gemini
gemini = Gemini()

# optimization loop
while num_eval < budget:

    if len(observations >= 2) and len(proxy_observations >= 2):

        # construct training set with current observations
        training_set = gryffin.construct_training_set(observations, proxy_observations)

        # train Gemini
        gemini.train(training_set['train_features'], training_set['train_targets'],
                     training_set['proxy_train_features'], training_set['proxy_train_targets'],
                     num_folds=3)

    # pass callable when asking Gryffin for new samples
    samples = gryffin.recommend(observations,
                                predictive_model=gemini)
```

In this external trianing case, you need only provide a Gryffin config file (i.e. no predictive model entries)

Gemini: Augmenting

the matter lab

Group of John Gregoire report overpotential composition spaces (6 metals total)



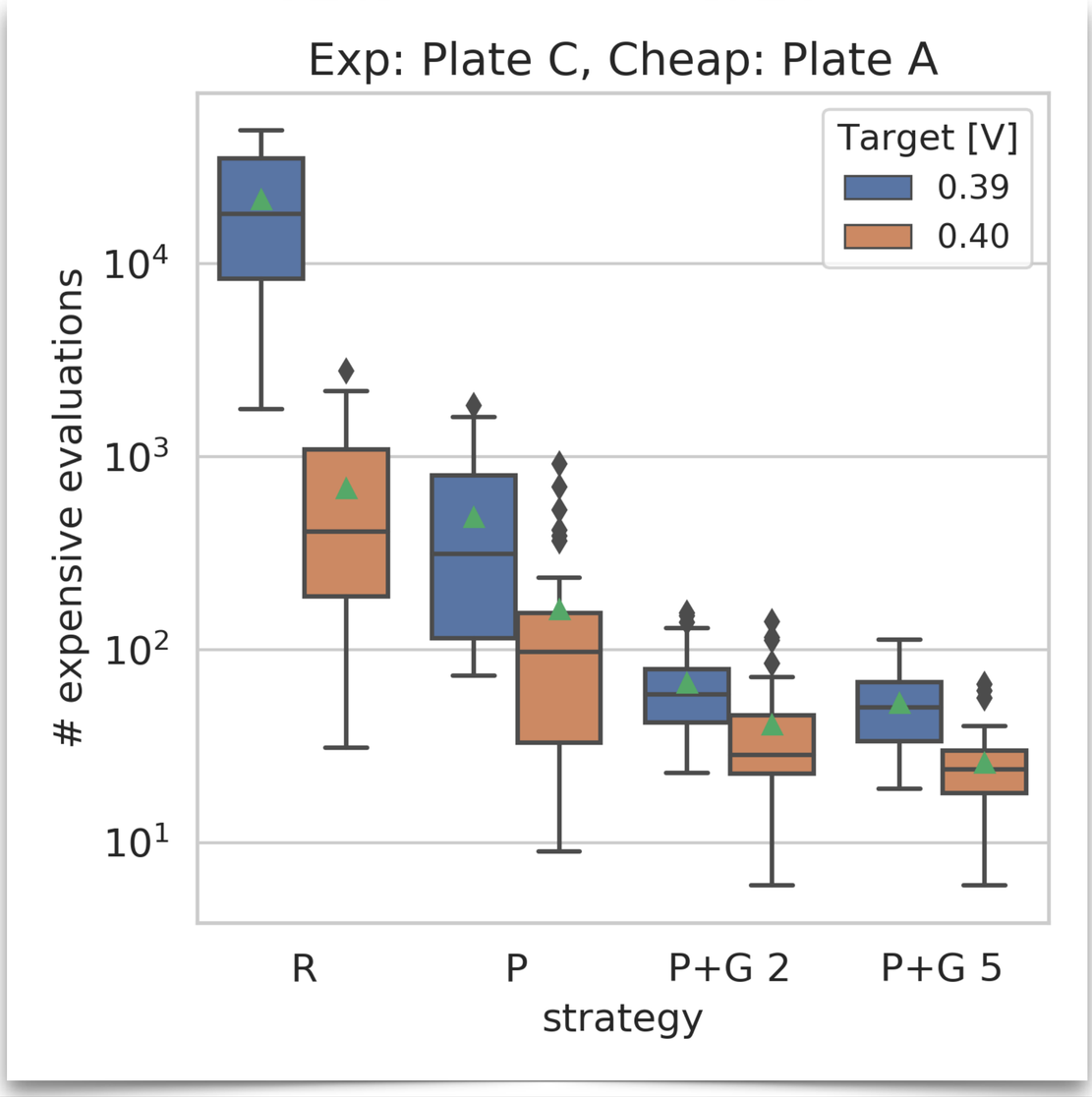
https://data.matr.io/

Each composition system comes with some cost…

Label	Composition system	Price [USD/g]
A	Mn-Fe-Co-Ni-La-Ce	0.06
B	Mn-Fe-Co-Ni-Cu-Ta	0.37
C	Mn-Fe-Co-Cu-Sn-Ta	0.37
D	Ca-Mn-Co-Ni-Sn-Sb	0.08

Our goal: optimize the C compositon space (expensive) with proxy-measurements from the A composition space (cheap)

Gemini: Perovskites application



Strategy	Target overpotential 0.39 V	0.40 V
Random sampling	21157 ± 2456	618 ± 106
	N/A	N/A
PHOENICS only	486 ± 78	161 ± 33
	(1.7×10^{-8})	(8.6×10^{-5})
PHOENICS + GEMINI $r = 2$	67 ± 6	41 ± 5
	(6.7×10^{-7})	(2.2×10^{-4})
PHOENICS + GEMINI $r = 5$	**53** ± 4	**26** ± 2
	(1.8×10^{-2})	(6.9×10^{-3})

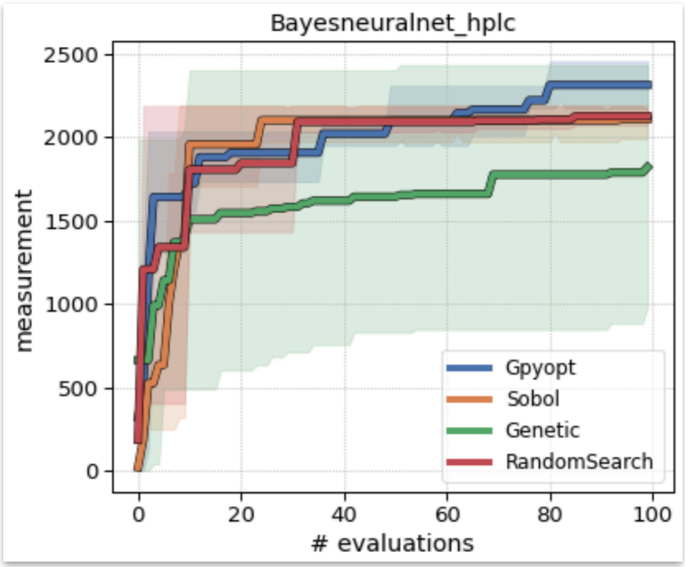
`Olympus`: Benchmarking experiment planning algorithms

the matter lab

https://aspuru-guzik-group.github.io/olympus/

- **18 experiment planning algorithms**
- **10 BNN emulated datasets** from chemistry and materials science
- **23 synthetic surfaces**
- **Standardized** benchmarks
- **Custom** planners, datasets, emulators
- **Plotting**
- **Analysis**

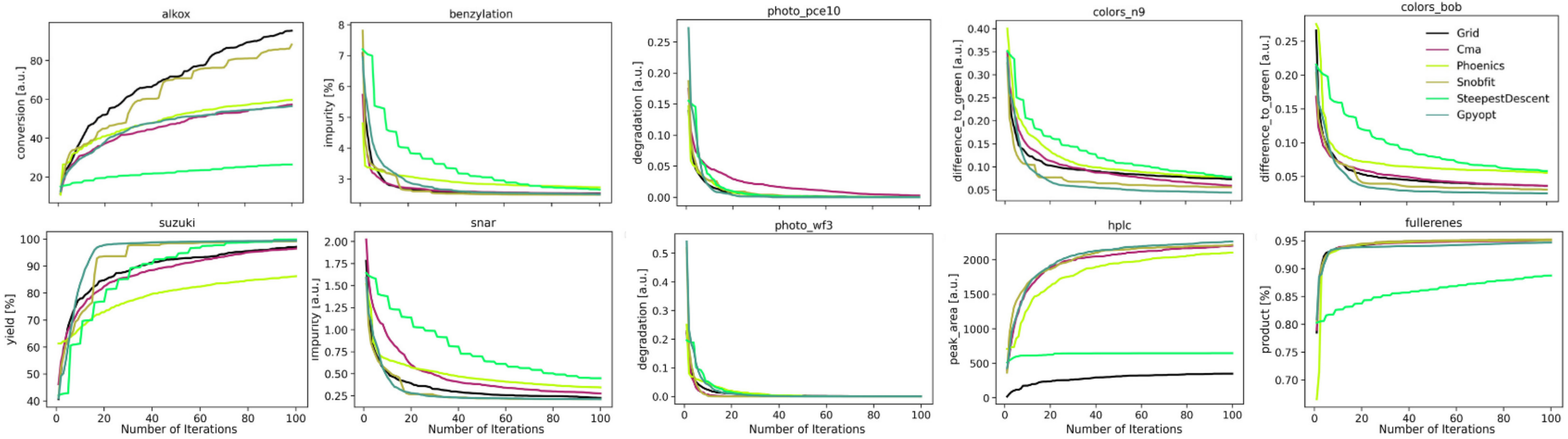
```python
from olympus import Campaign

DATASET = 'hplc'
NUM_REPETITIONS = 3
PLANNERS = ['Gpyopt', 'Sobol', 'Genetic', 'RandomSearch']

for PLANNER in PLANNERS:
  for repetition in range(NUM_REPETITIONS):

    olymp.run(
        planner=PLANNER,       # run simulation with <PLANNER>,
        dataset=DATASET,       # on emulator trained on dataset <DATASET>;
        campaign=Campaign(),   # store results in a new campaign,
        num_iter=100,          # run benchmark for 100 iterations
    )
```



`Olympus`: Benchmarking experiment planning algorithms

the matter lab

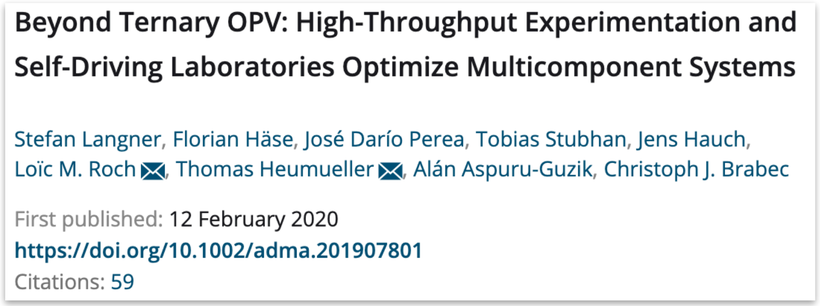
Priyansh Parakh



P. Parakh, **RJH**, *et al. In preparation*

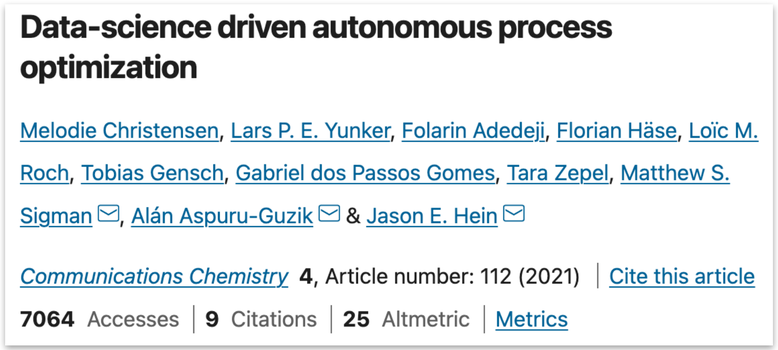
https://github.com/aspuru-guzik-group/olympus

Previous applications



S. Langner *et al. Adv. Mater.* **32**, 1907801 (2020)

B. MacLeod, *et al. Sci. Adv.* **6**, eaaz8867 (2020)



H. Tao *et al. Adv. Funct. Mater.* **31**, 2106725 (2021)

Christensen *et al*. *Commun. Chem*. **4**, 1-12, (2021)

M. Seifrid, **RJH**, *et al. ACS Cent. Sci.* **8**, 122-131 (2022)

Seedless photoinitiated nanoparticle synthesis

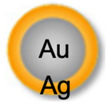
the matter lab

- design of gold nanoparticles with targeted spectroscopic properties
- integration of BayesOpt tools with automated microfluidic reactor and characterization, fully automated self-driving lab (MF-ML platform)

Taylor Wu

Prof. Kumacheva

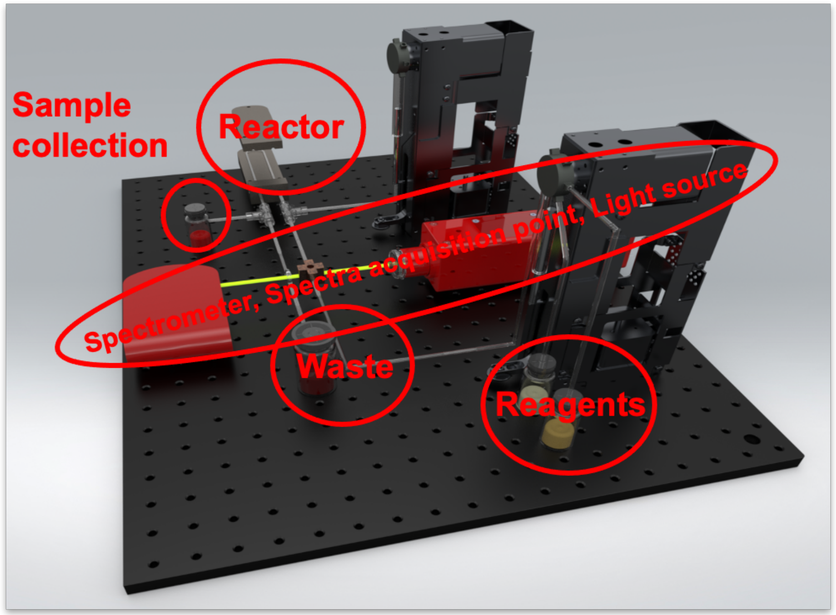
$$HAuCl_4 + AgNO_3 \xrightarrow[CTAB\ /\ H_2O]{\text{I-2959} + h\nu}$$

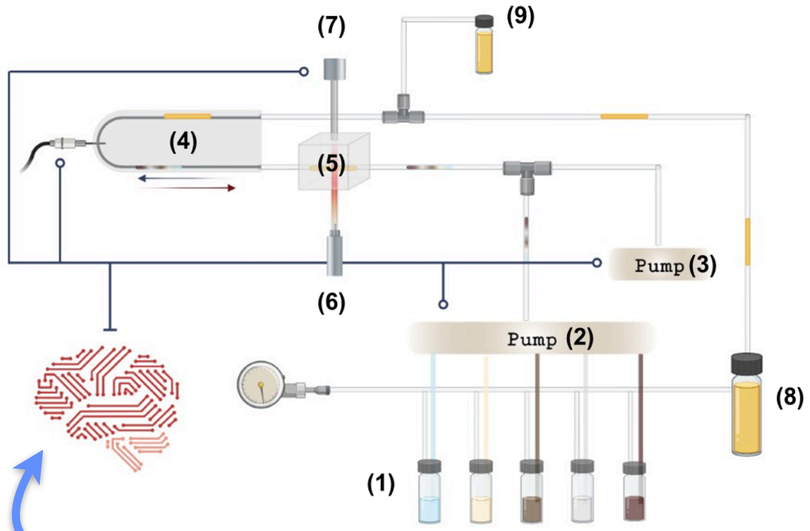


AuAg core-shell

AuAg alloy

Au nanorod





Gryffin, Chimera, known constraints

Parameters (7)

- concentration of reagents
- reaction time
- UV intensity
- mixing rate

Objectives (4)

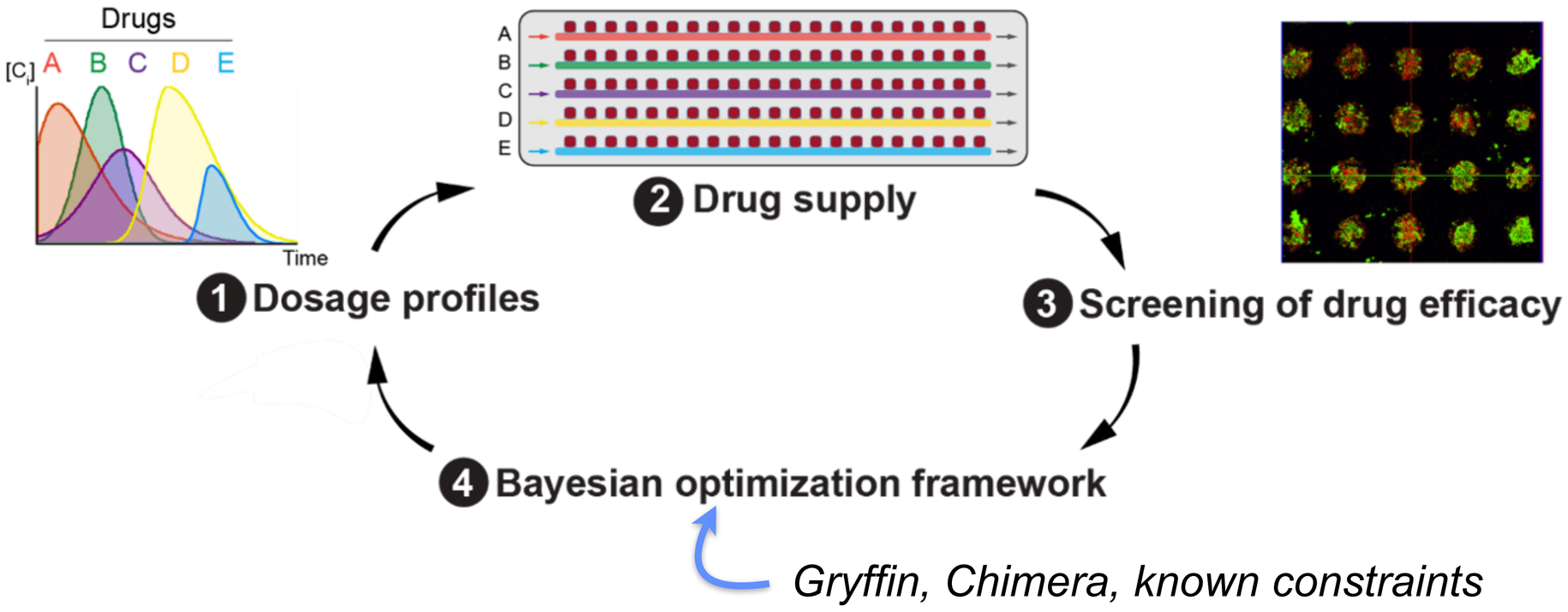
- spectroscopic characteristics of NPs

(1) Reagents (2) Dispenser (3) Oscillator (4) Reactor (5) Flow cell (6) Light source (7) Spectrometer (8) Waste (9) Sample collection outlet

in vitro design of combination cancer therapies

the matter lab

- Optimize for synergistic effects of multi-drug combinations (common breast cancer chemotherapy regime)
- Breast cancer spheroids subject to drugs in microfluidic array
- Treatment efficiency evaluated with live/dead assays based on fluorescence imaging




Dr. Ilya Yakavets


Sina Kheiri


Ella Rajaonson


Prof. Kumacheva

Parameters (5)
- drug doses (concentration)
- dose time
- dose duration

Objectives (2)
- combination index
- cell viability

Optimization of process parameters for accurate liquid dispensing with Chemspeed

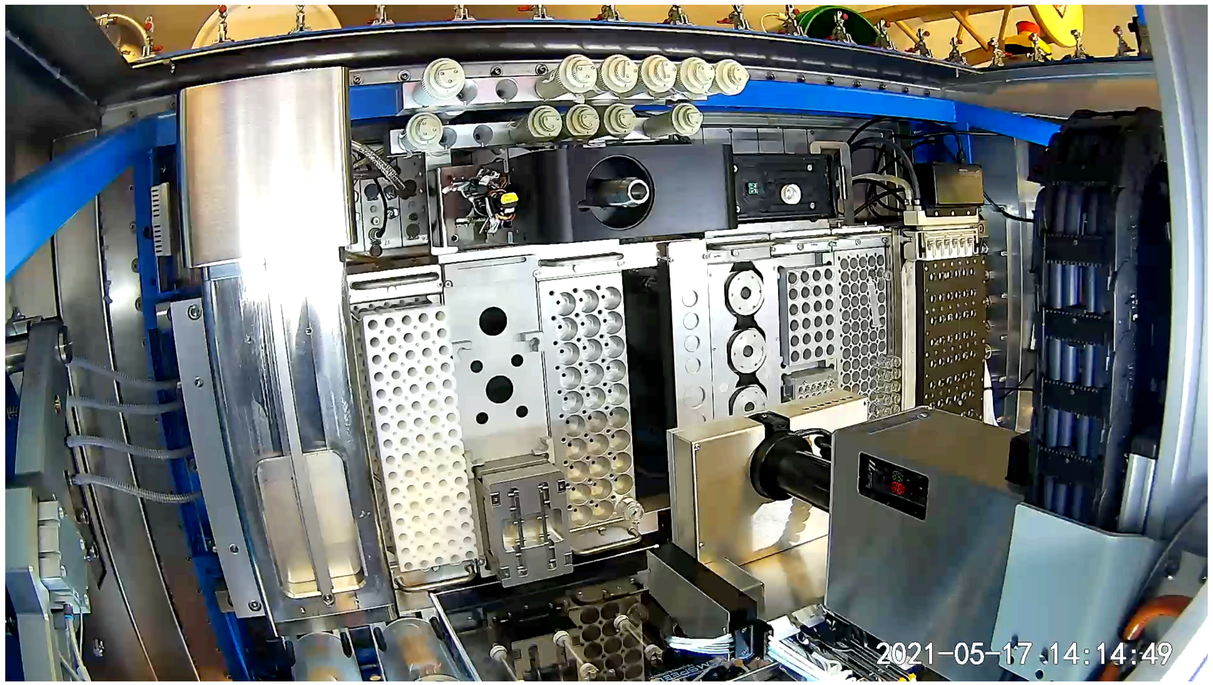
the matter lab

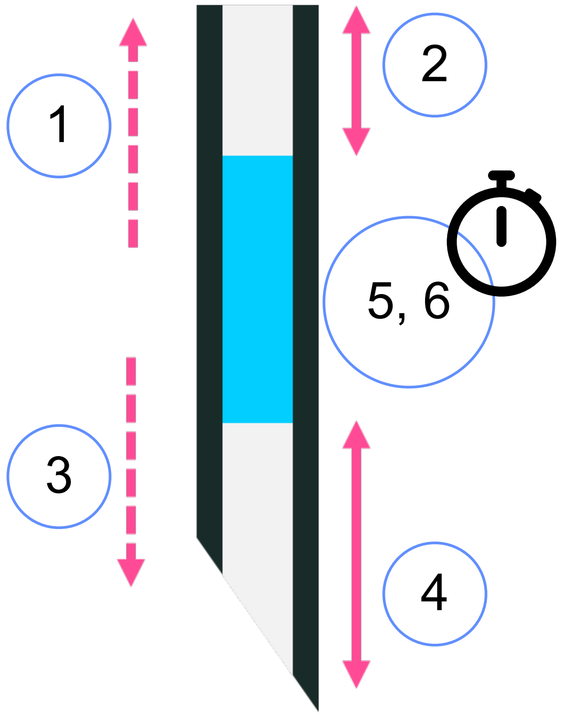
- Determine liquid specific process parameters for accurate and precise dispensing with Chemspeed robot
- BayesOpt framework interfaced with Python Chemspeed control
- Fully automated workflow enabled by balance placed inside Chemspeed


Dr. Martin Seifrid


Ella Rajaonson


Dr. Tony Wu





Parameters (6)

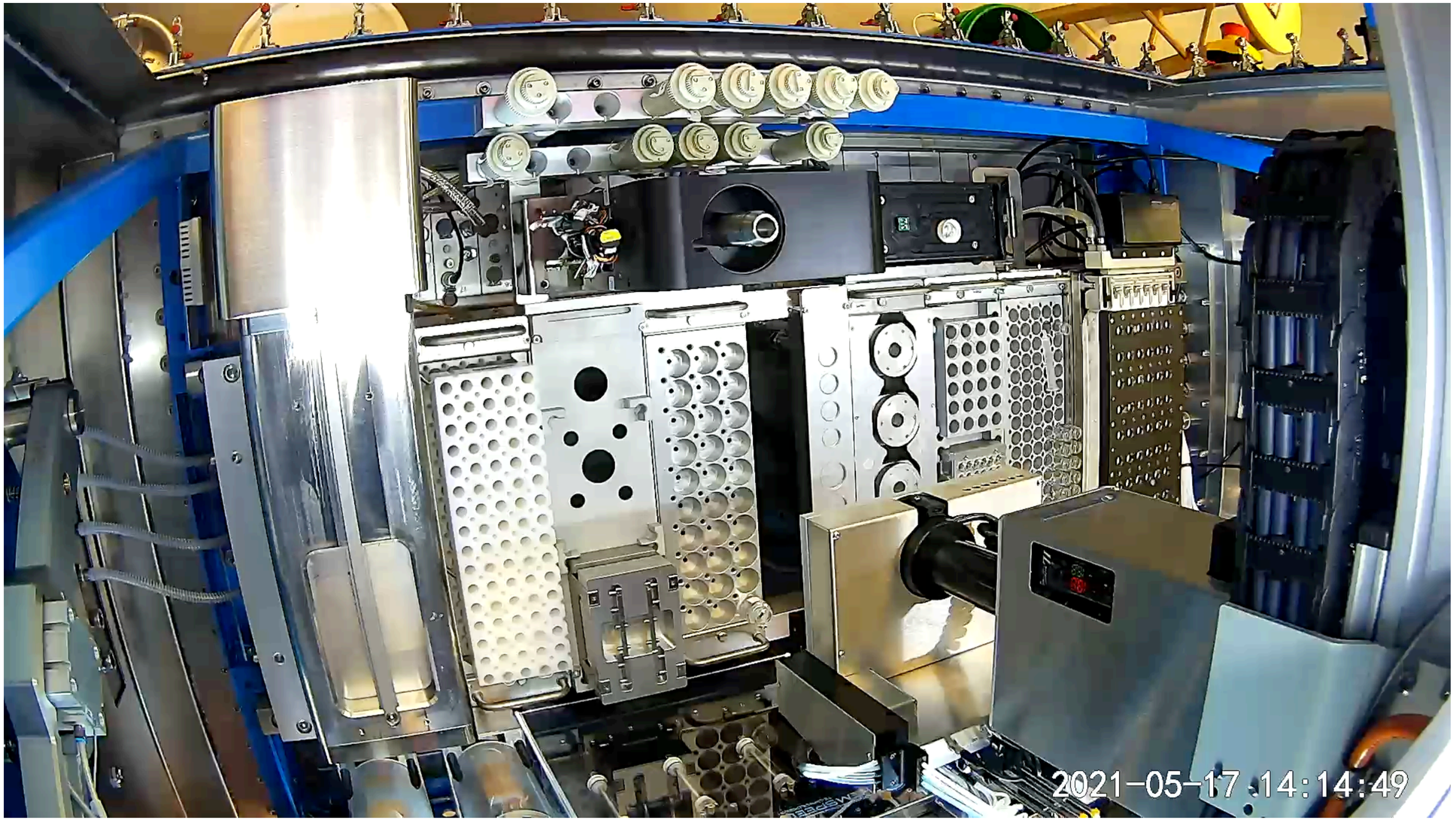
1. Aspiration speed (source)
2. Airgap volume
3. Dispense speed (destination
4. Post-airgap volume
5. Equilibration time (source)
6. Equilibration time (destinatio

Objectives (2)

1. Minimize mean relative error
2. Minimize standard deviation

Optimization of process parameters for liquid dispensing with Chemspeed

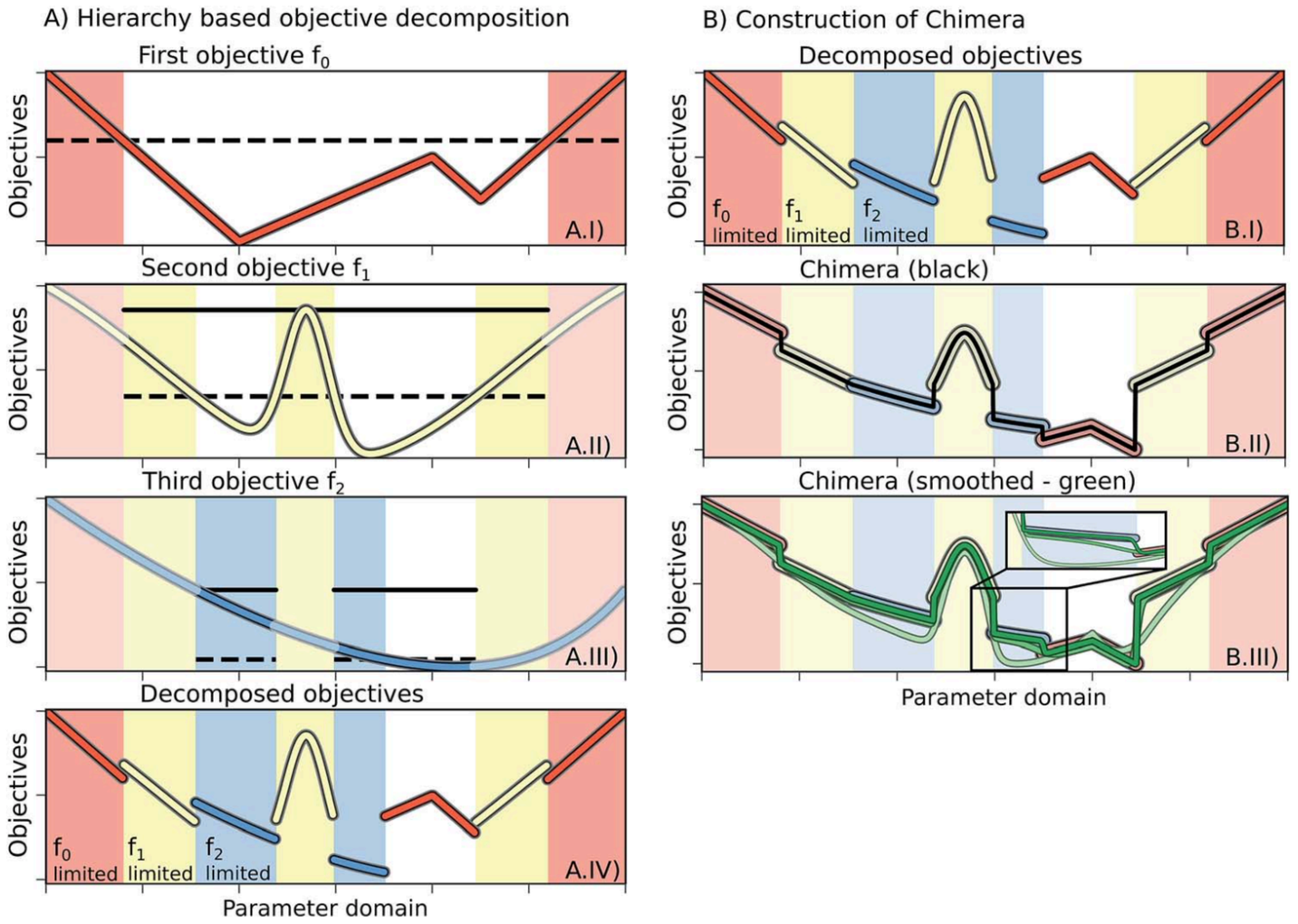
the matter lab

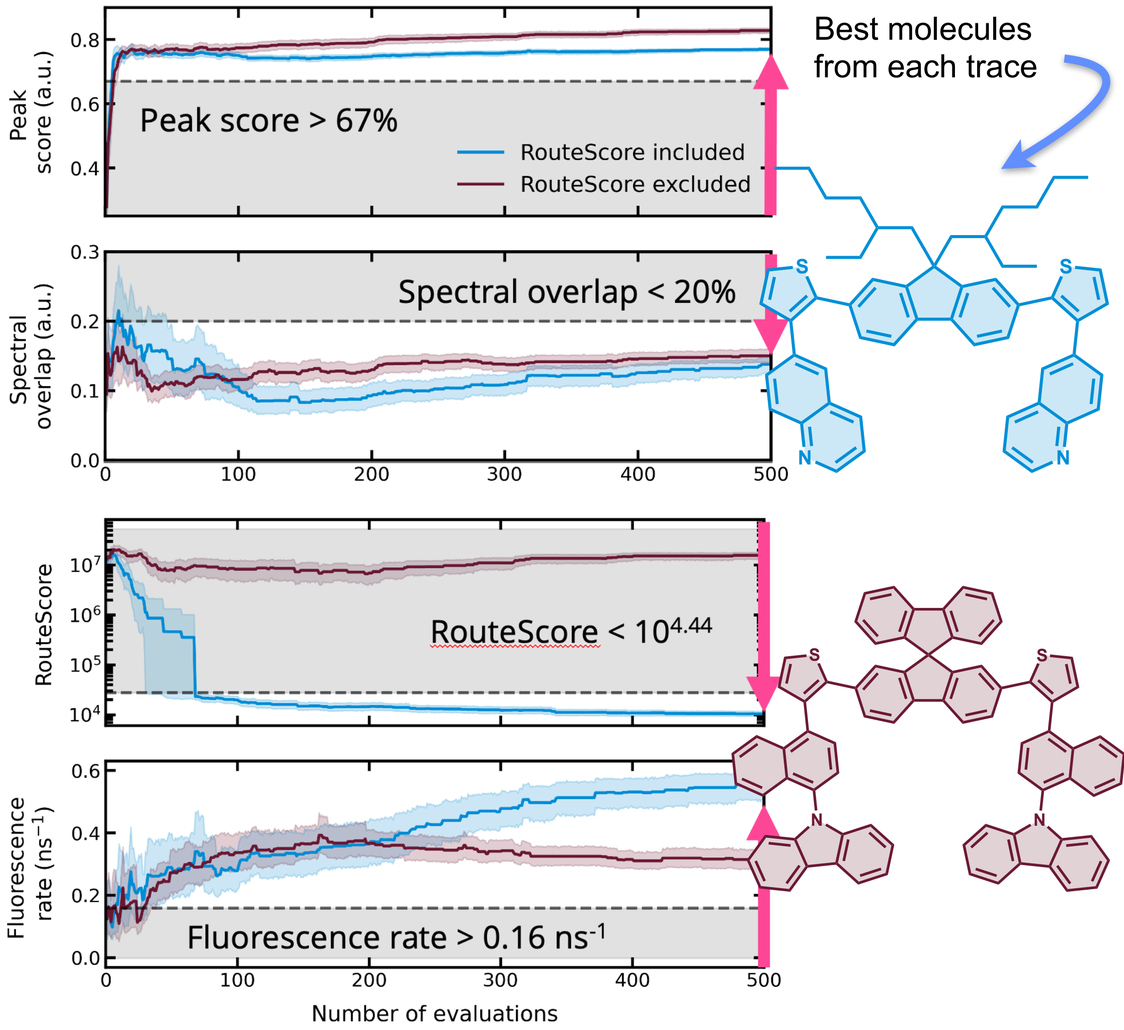


M. Seifrid, E. Alca, E. Rajaonson, **RJH**, *et al. In preparation*

Chimera: Hierarchical scalarizing for MOO

the matter lab

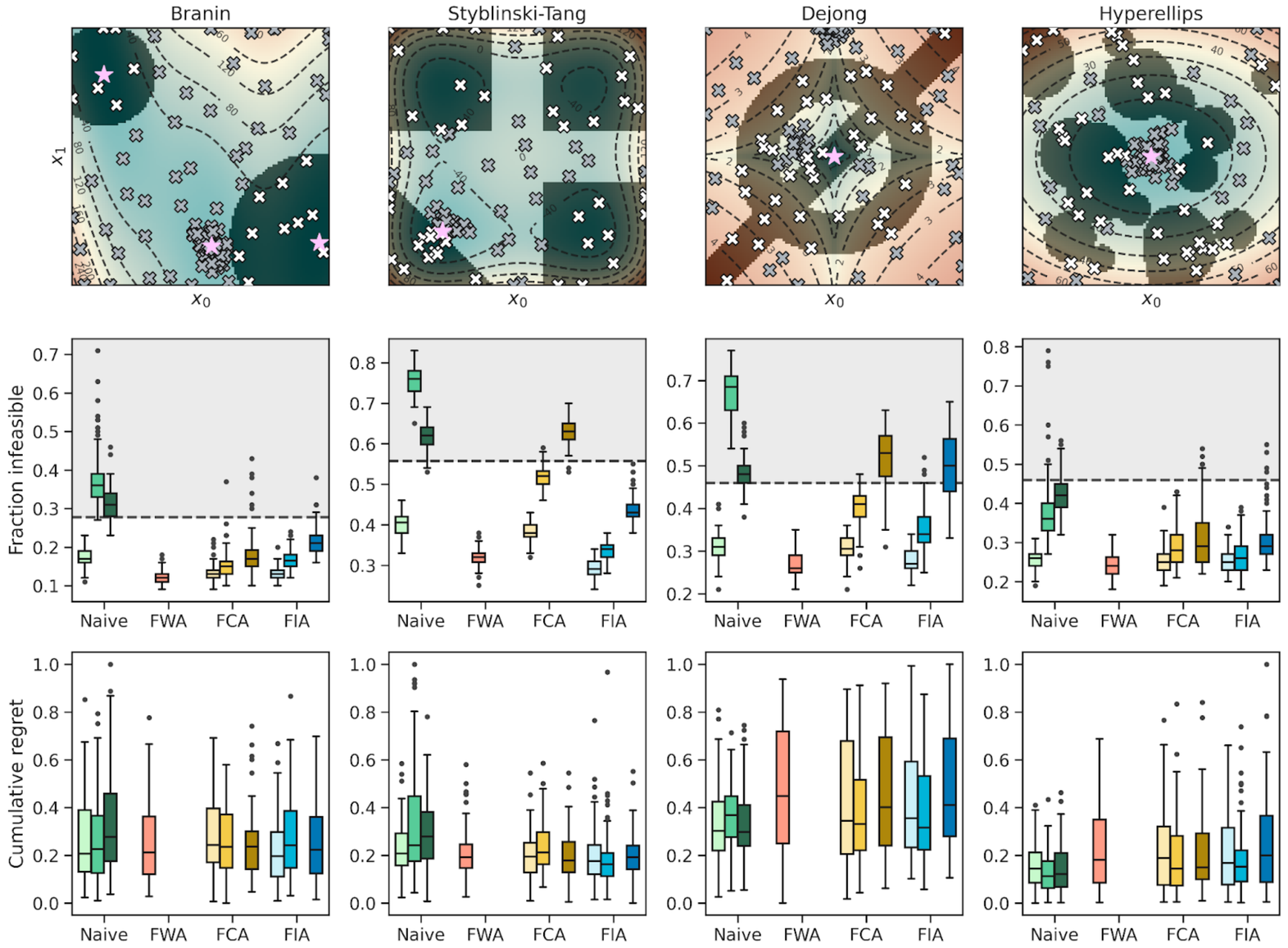




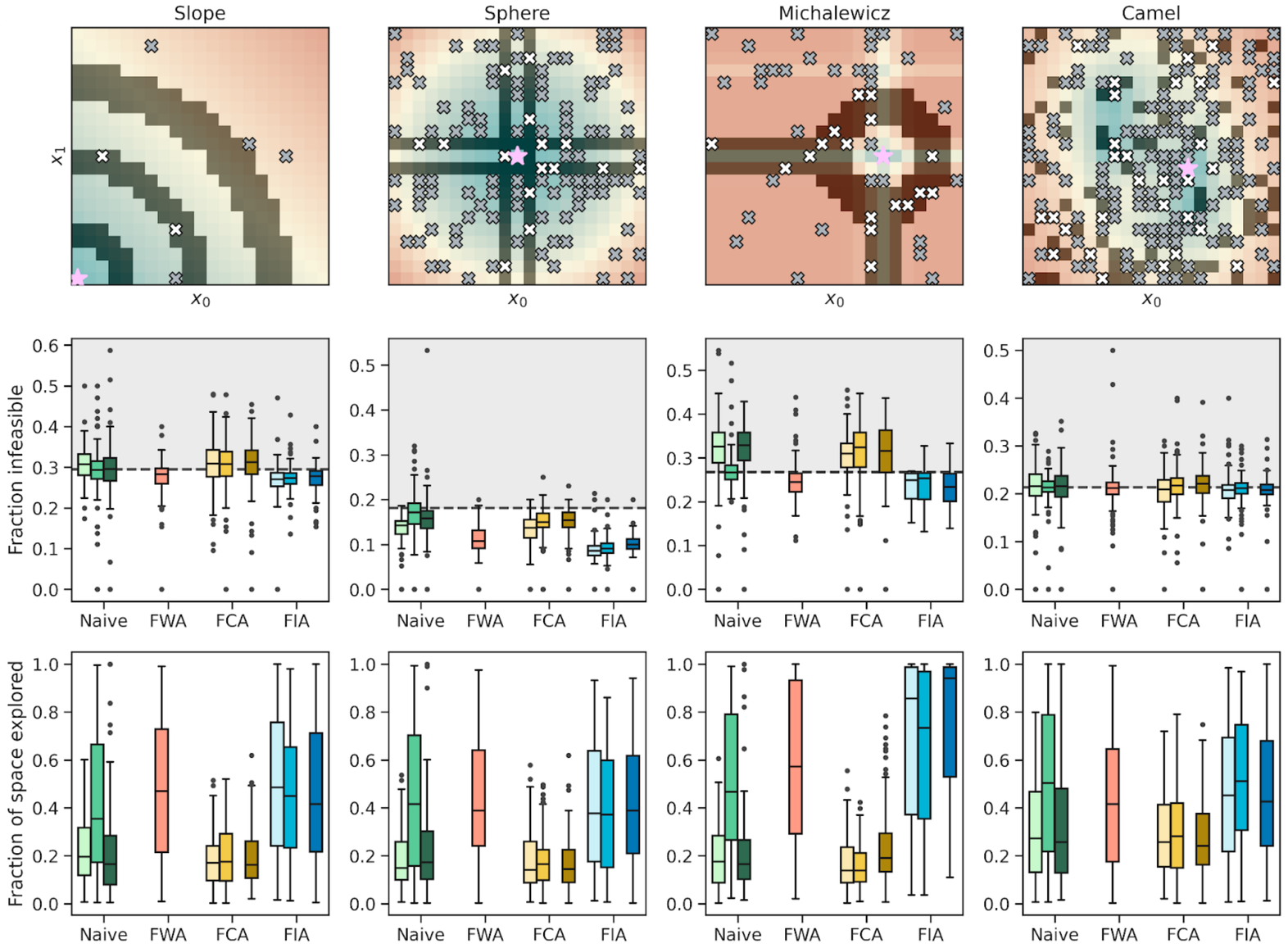
F. Häse, *et al. Chem. Sci.* **9**, 7642–7655 (2018)

M. Seifrid, **RJH**, *et al. ACS Cent. Sci.* **8**, 122-131 (2022)

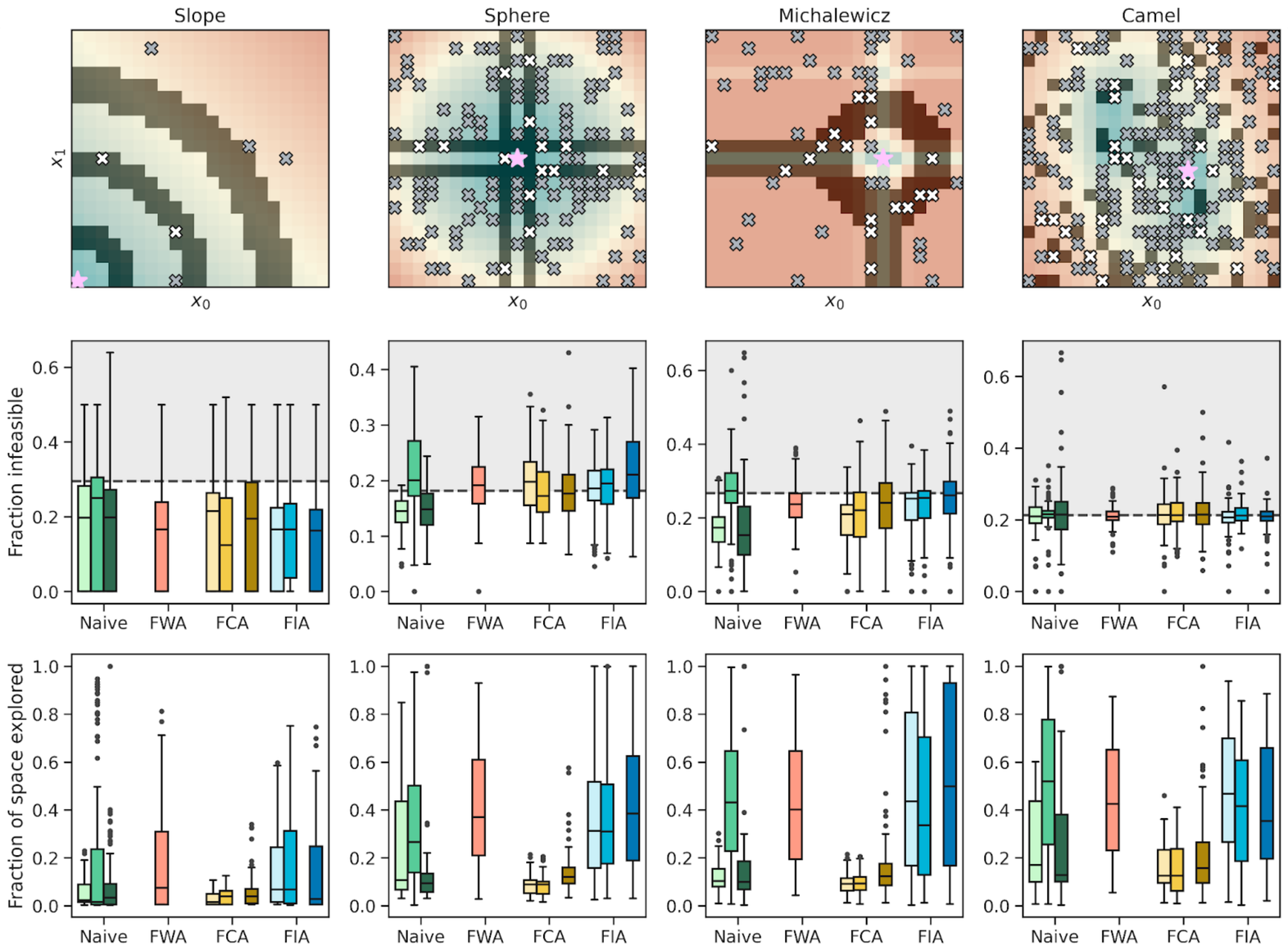
Unknown constraints - Naive Gryffin continuous



Unknown constraints - Naive Gryffin categorical

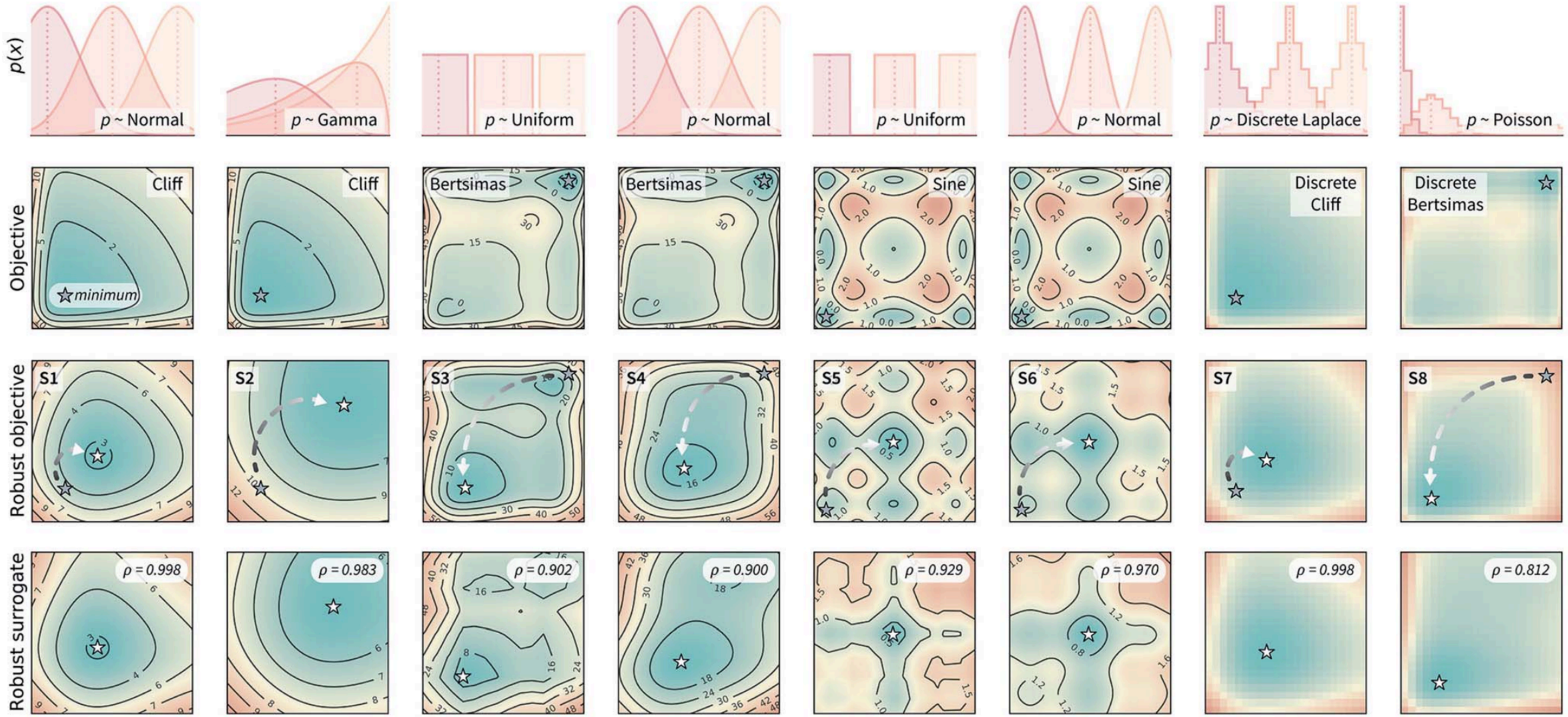


Unknown constraints - Dynamic Gryffin categorical



`Golem`: Synthetic surfaces with input parameter noise

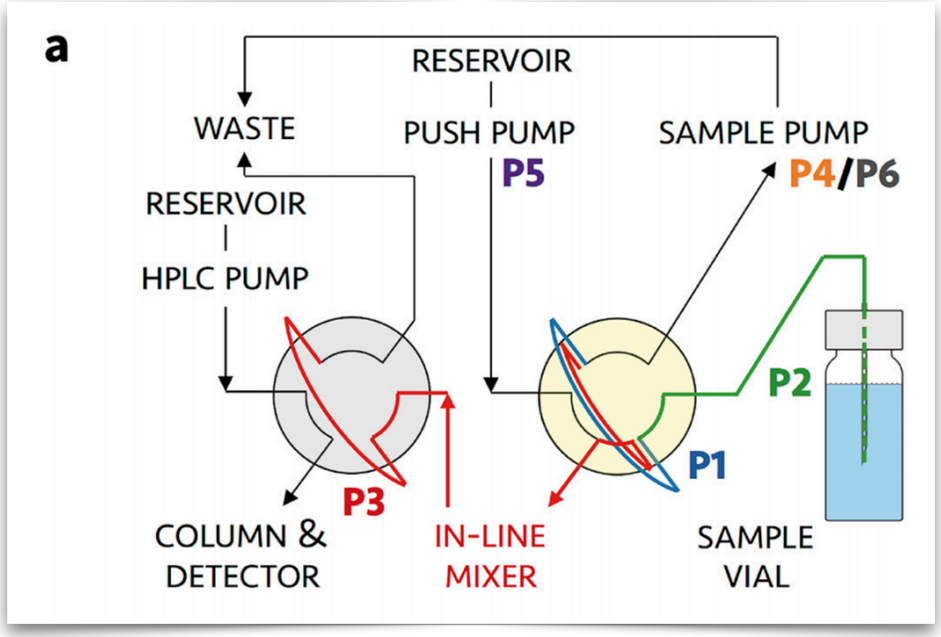
the matter lab

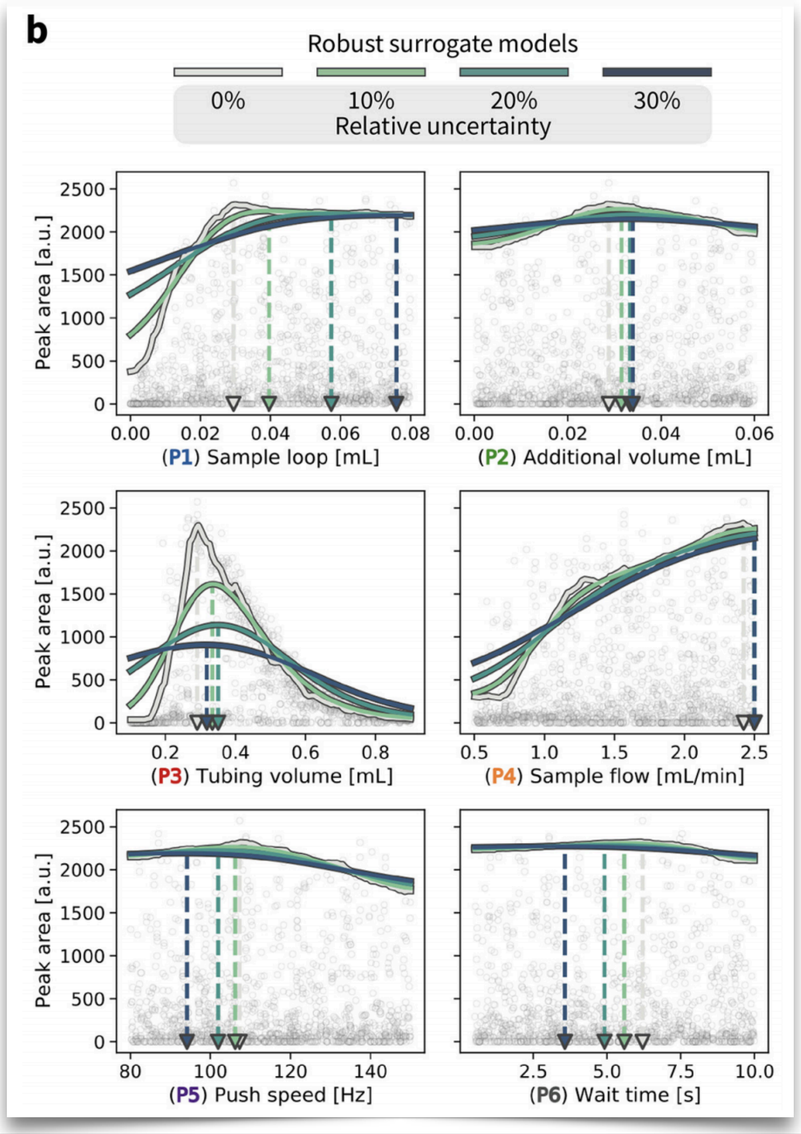


Golem: HPLC calibration application

the matter lab

- 6 parameters
- Objective is peak area maximization
- 1386 randomly collected samples

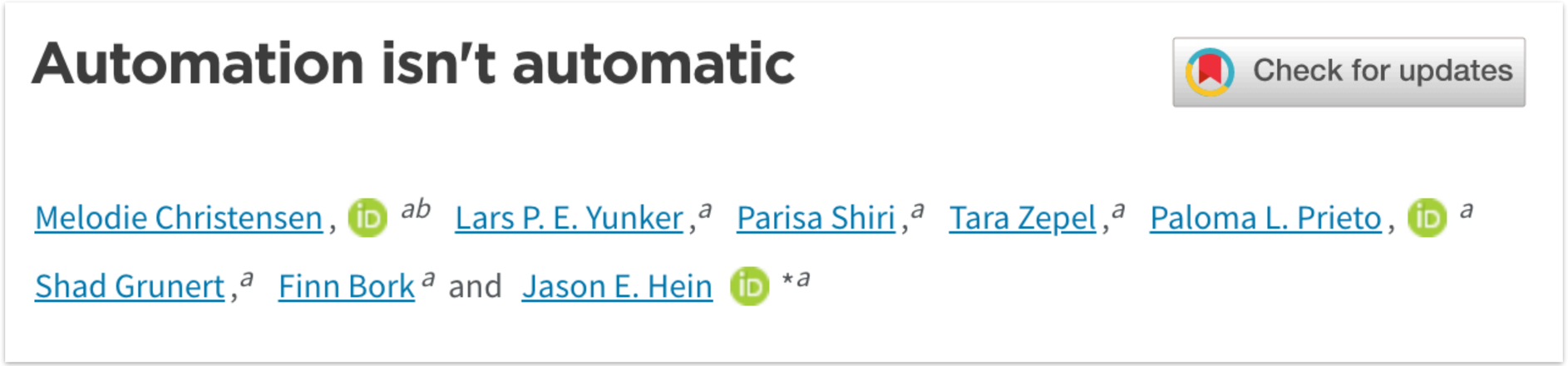




A posteriori noise analysis

Limitations/difficulties

- One size does not fit all on algorithm level and laboratory level
- Parameter space is usually fixed *a priori!* How do chose a good search space is an open (and puzzling) question
- Humans still play an indispensable role in “autonomous” science! (infusing expert intuition into an experiment planning algorithm is incredibly difficult)
- Interpretability: how can deliver understandable messaged to humans?



Acknowledgements

Alán Aspuru-Guzik

NSERC CRSNG

PGSD3-534584-2019