THE UNIVERSITY of TENNESSEE KNOXVILLE

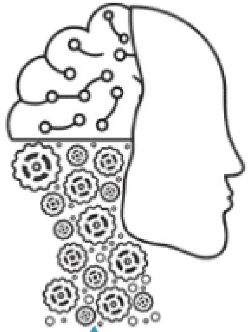
Lecture 5: Decision making algorithms and reward-based workflows

Sergei V. Kalinin

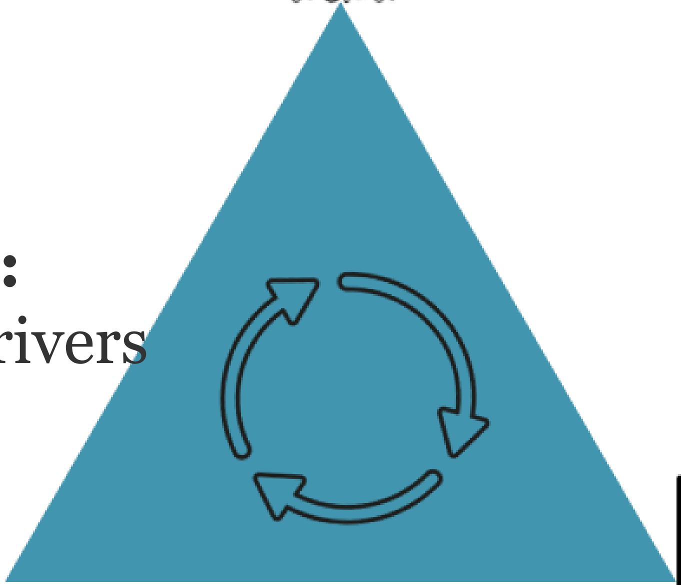
University of Tennessee, Knoxville, and
Pacific Northwest National Laboratory

Pacific Northwest NATIONAL LABORATORY

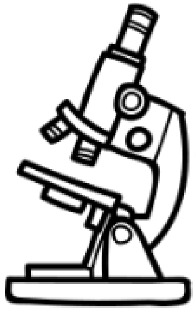
Challenges for the AE Microscopy

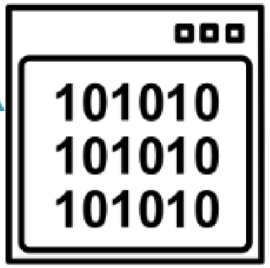


Workflow design:
- Reward and value functions
- Human heuristics
- Monitoring
- Limited experimental budget

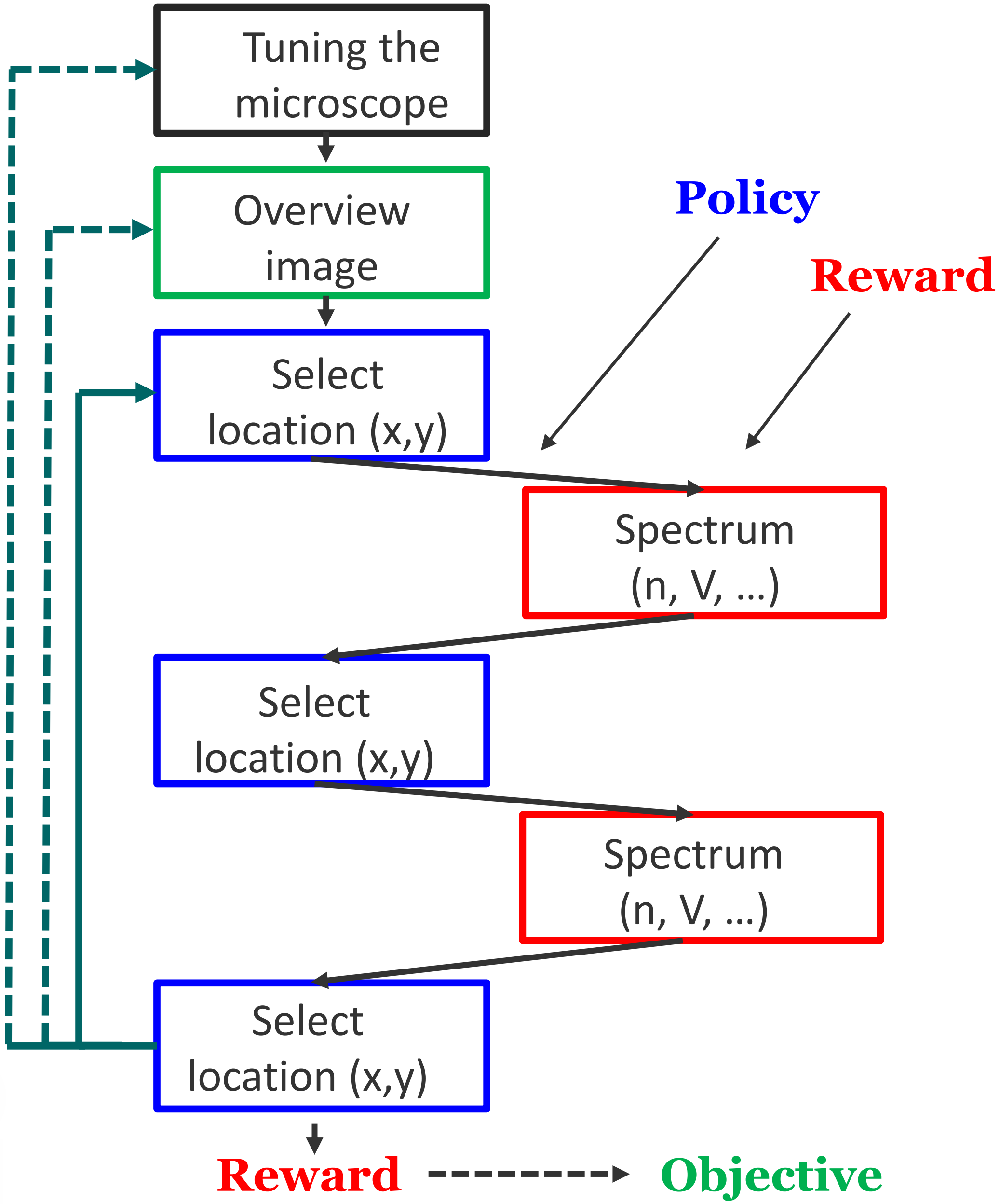


Engineering controls:
- Instrument-specific drivers
- Hyperlanguage
- Python APIs





ML/AI



To implement the ML workflows, we start from emulating the human operations:

- Well defined and explainable commands
- Extensive domain expertise
- Potentially available data from experiments

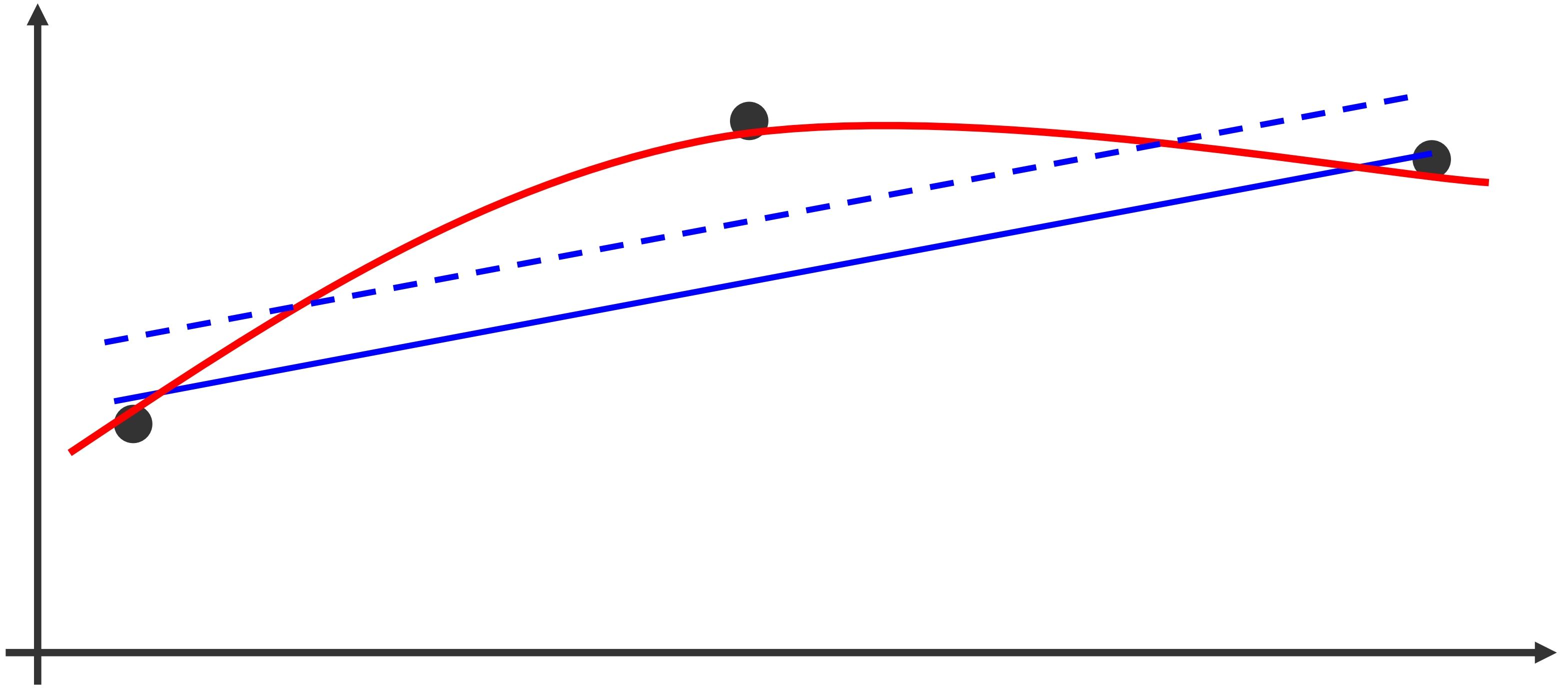
Development of ML workflows can give rise to more complex imaging modalities

- Data volumes and dimensionalities above human level
- More complex modes of sampling
- "Guardian angel" modules

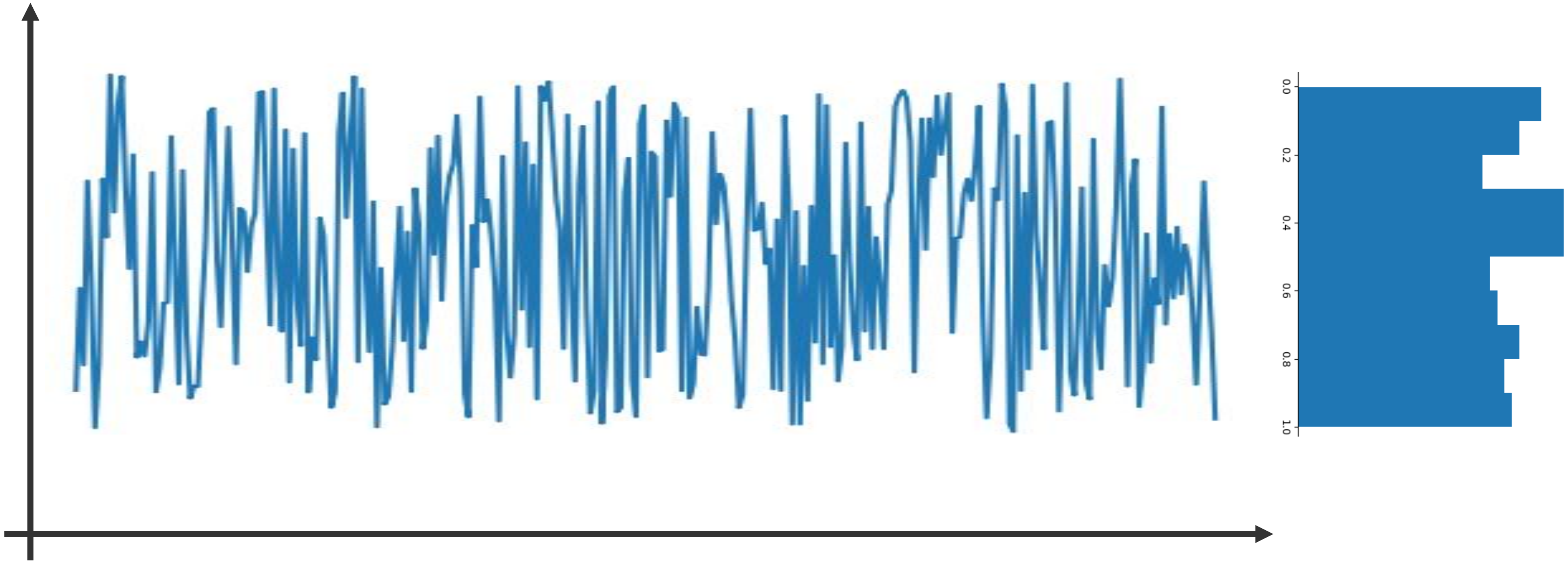
However, we always have to think about

- Reward function(s) for imaging problem
- Reward functions for materials problem
- Overall objective

What do we know if we do not know anything?



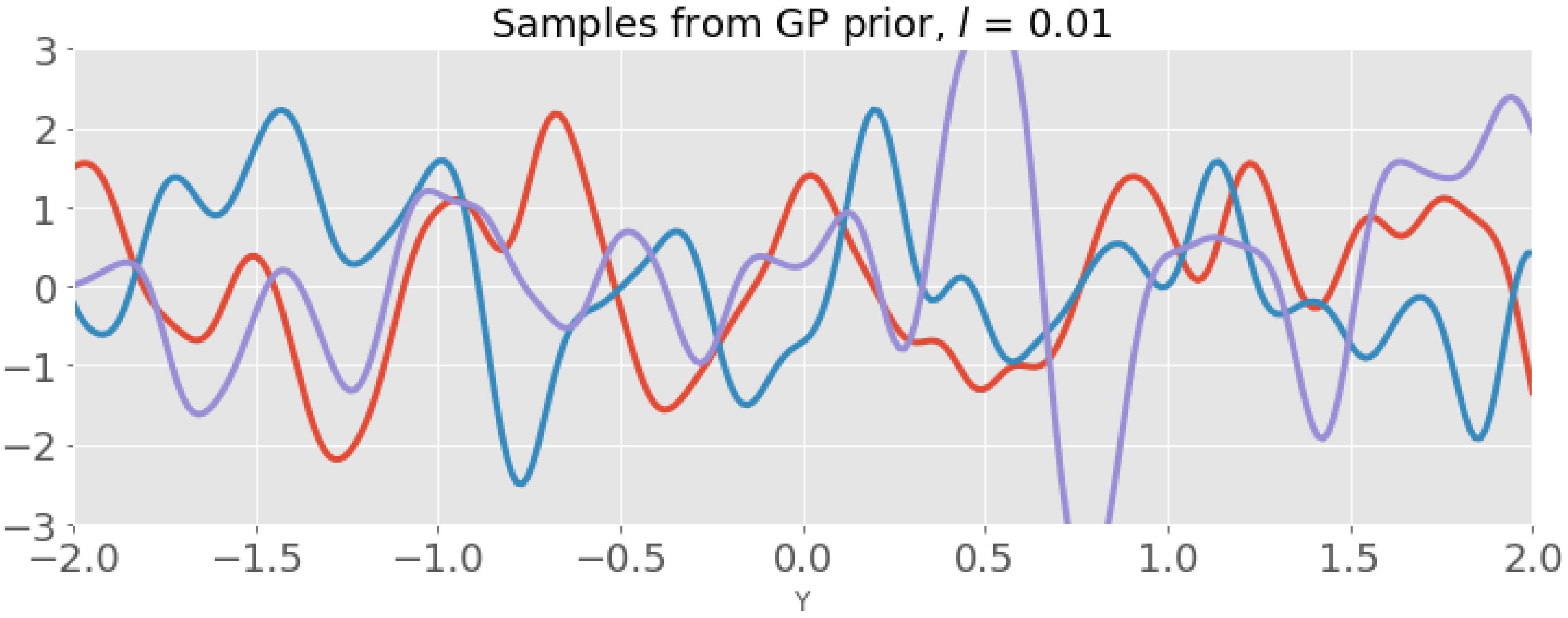
What do we know if we do not know anything?



Gaussian Process Regression

- Covariance matrix determines what type of functions we will allow.

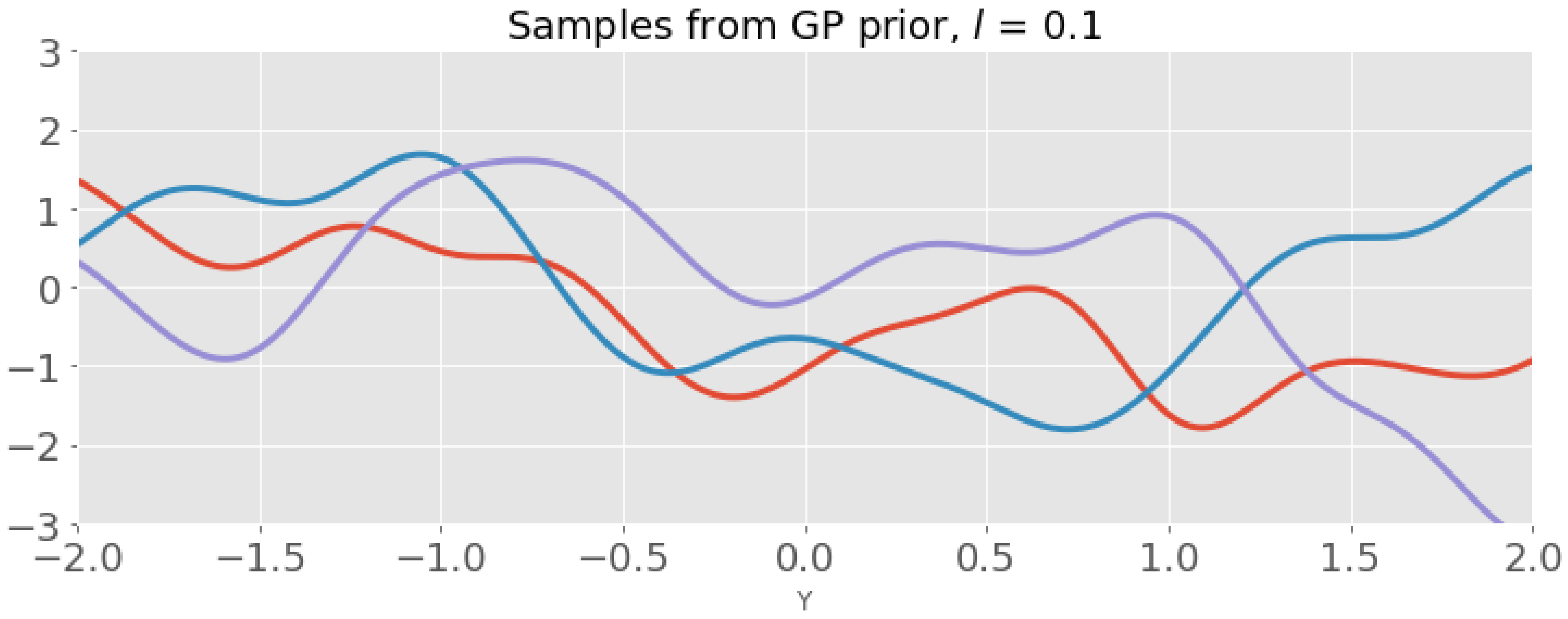
$$\mathrm{k}(\mathrm{x}, \mathrm{x}') = \exp\left(-\frac{1}{2l}(x - x')^2\right)$$



Gaussian Process Regression

- Covariance matrix determines what type of functions we will allow.

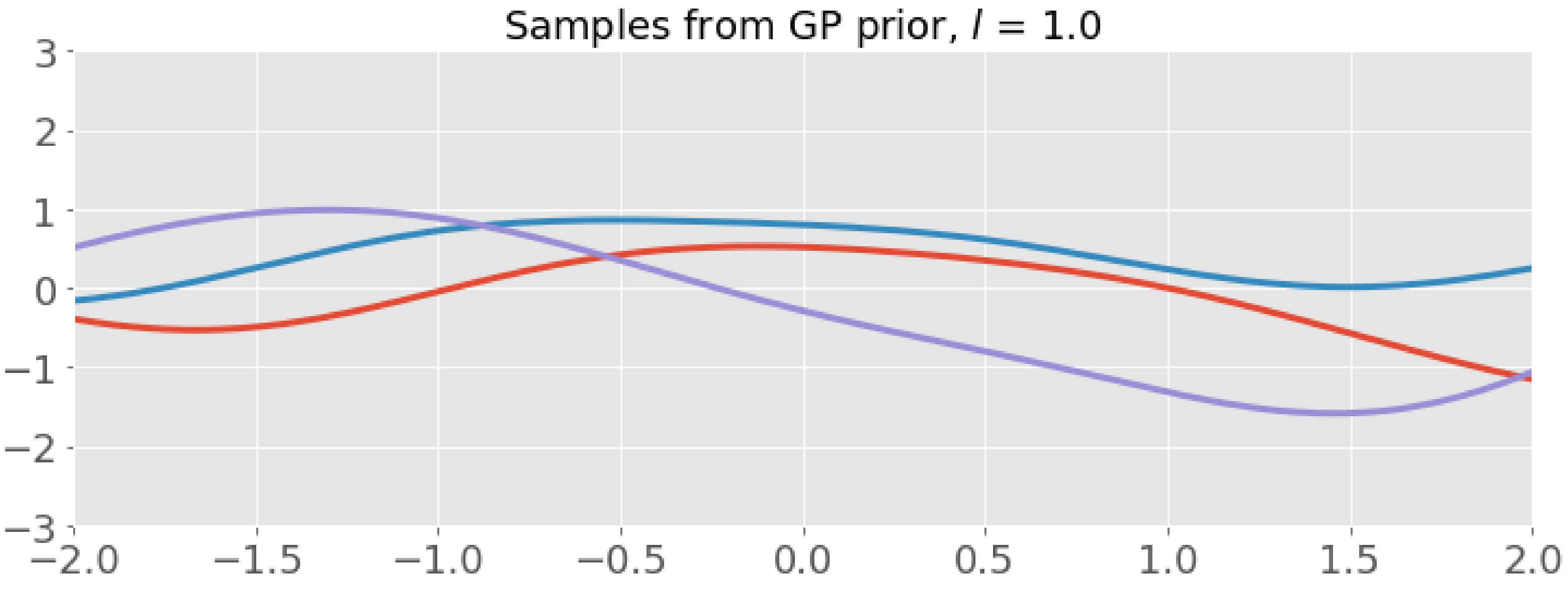
$$k(x, x') = \exp\left(-\frac{1}{2l}(x - x')^2\right)$$



Gaussian Process Regression

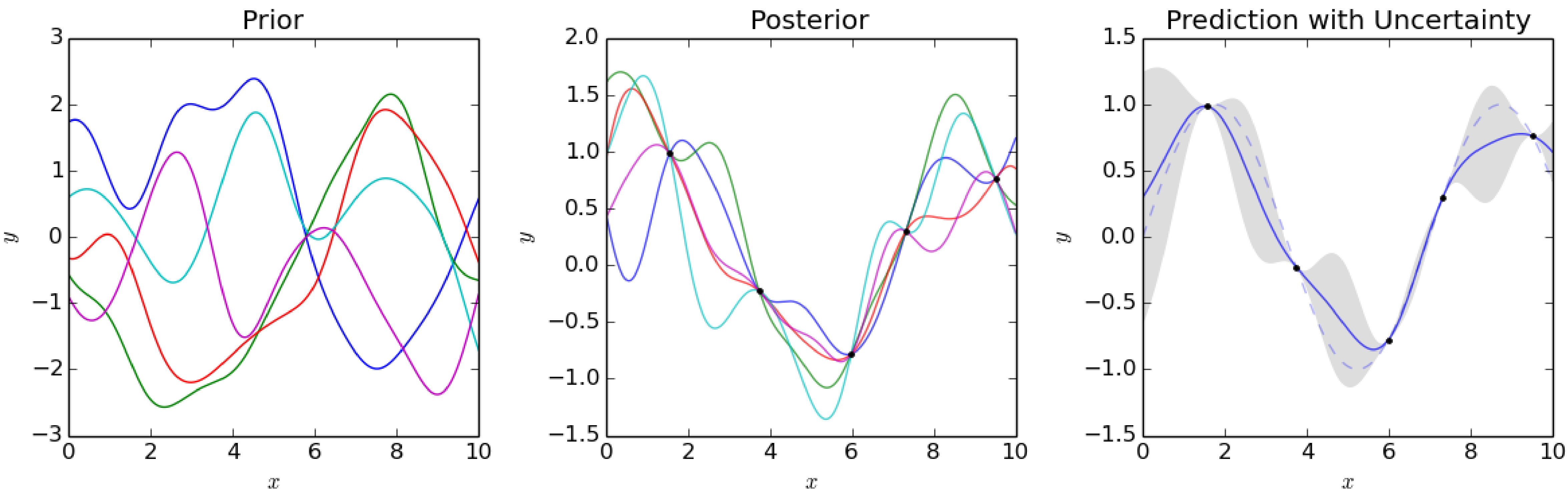
- Covariance matrix (kernel) determines what type of functions we will allow.

$$k(x, x') = \exp\left(-\frac{1}{2l}(x - x')^2\right)$$



L controls the length scale – sort of how far points should be to make them independent of each other.

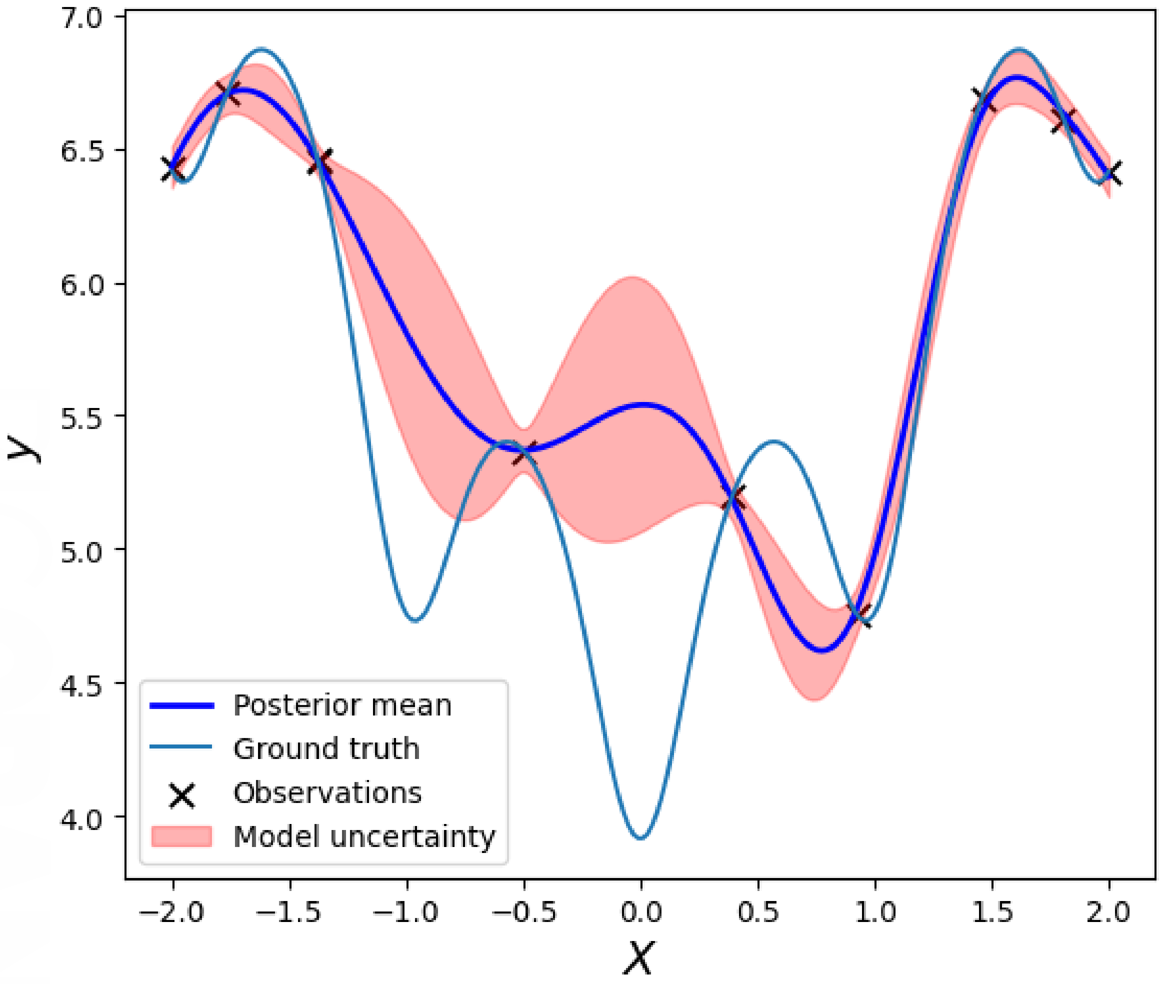
Gaussian Process Regression



Prior: What can the function be before the measurement
Data: Measurement
Posterior: What can the function be after measurement

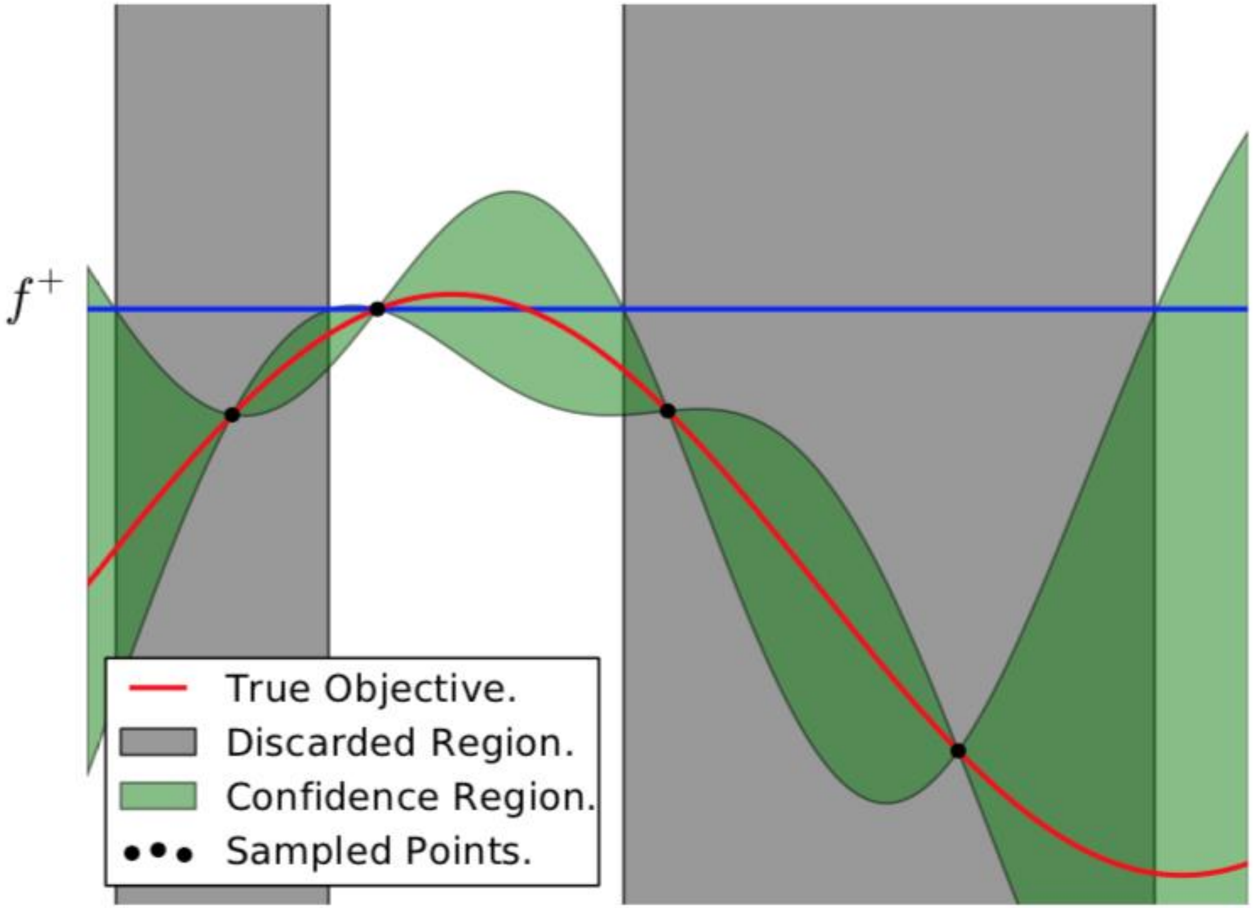
Policy: How do we balance exploration and exploitation (acquisition function)

GP Vocabulary



- Gaussian Process
- Kernel and kernel parameters
- Kernel Priors
- Noise Priors
- Posteriors

Bayesian Optimization

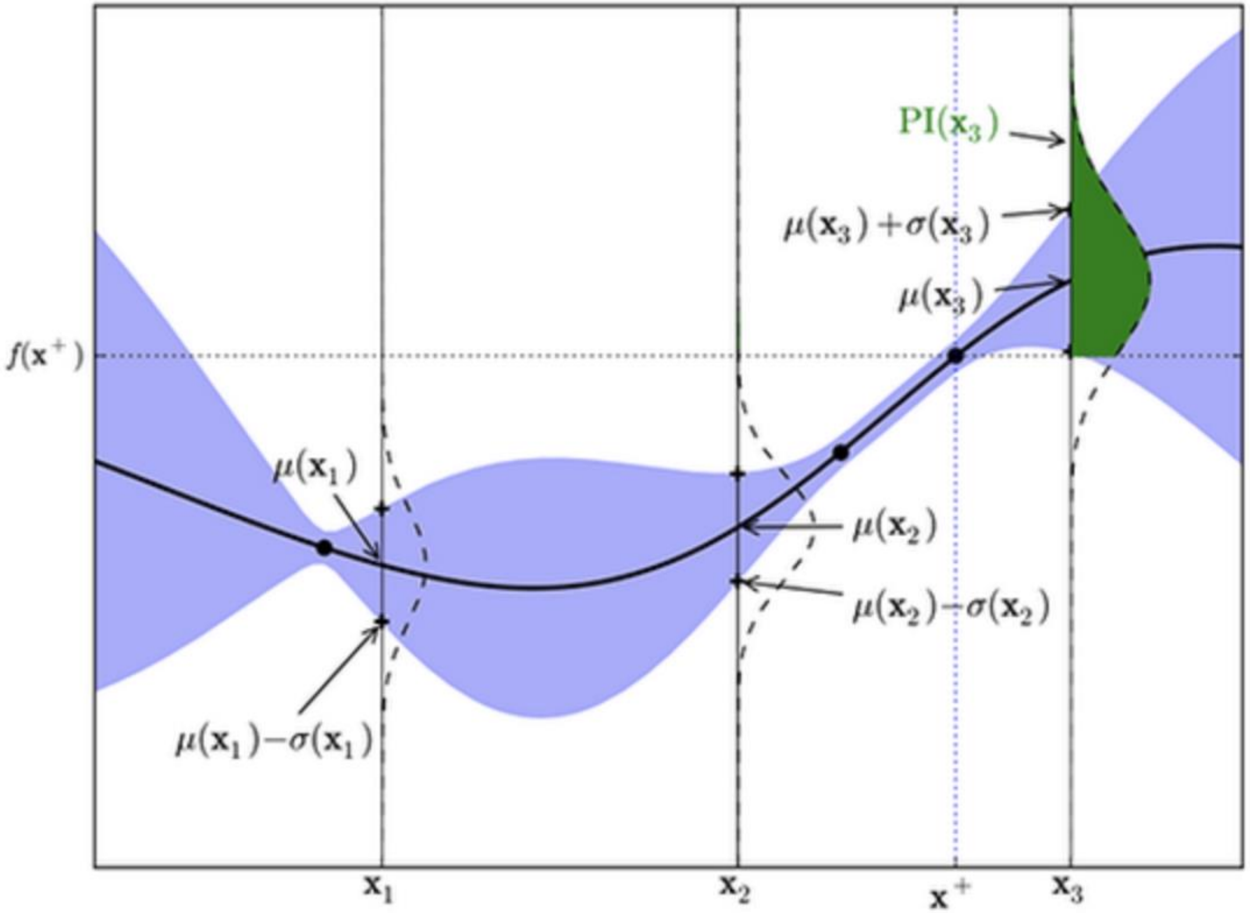


- We have some measurements in space X, and we want to maximize some property f(X).
- How can we decide what point to measure next to best maximize f?
- We need to balance the exploration of the space with exploitation of regions near we have already know

N. de Freitas et al., Taking the Human Out of the Loop: A Review of Bayesian Optimization , *Proceedings of the IEEE* **104**, 148 (2015)

Acquisition Functions

Probability of Improvement
Acquisition Function



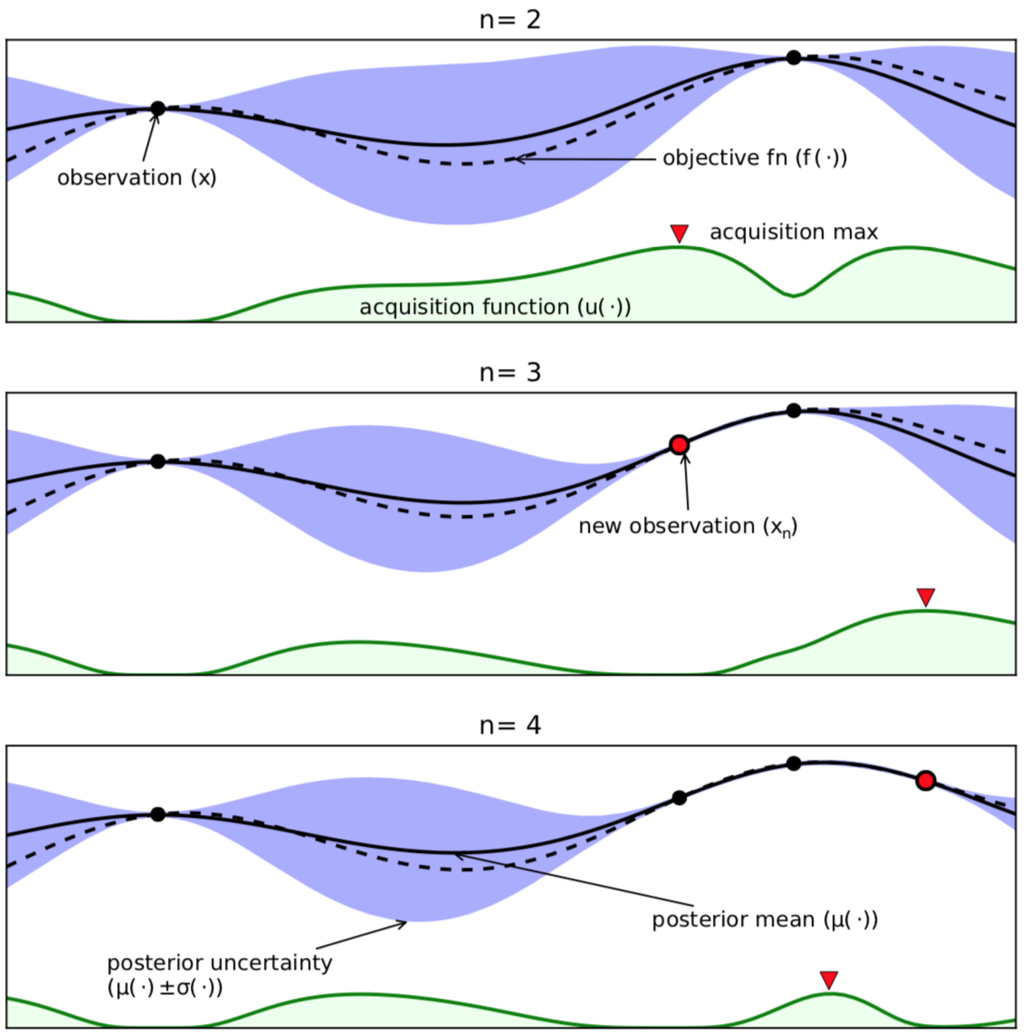
1. **Upper confidence bound:** simplest possible - just take the upper confidence bound from the prediction

2. **Probability of Improvement:** Integral from current functional maximum to upper limit of distribution as test point

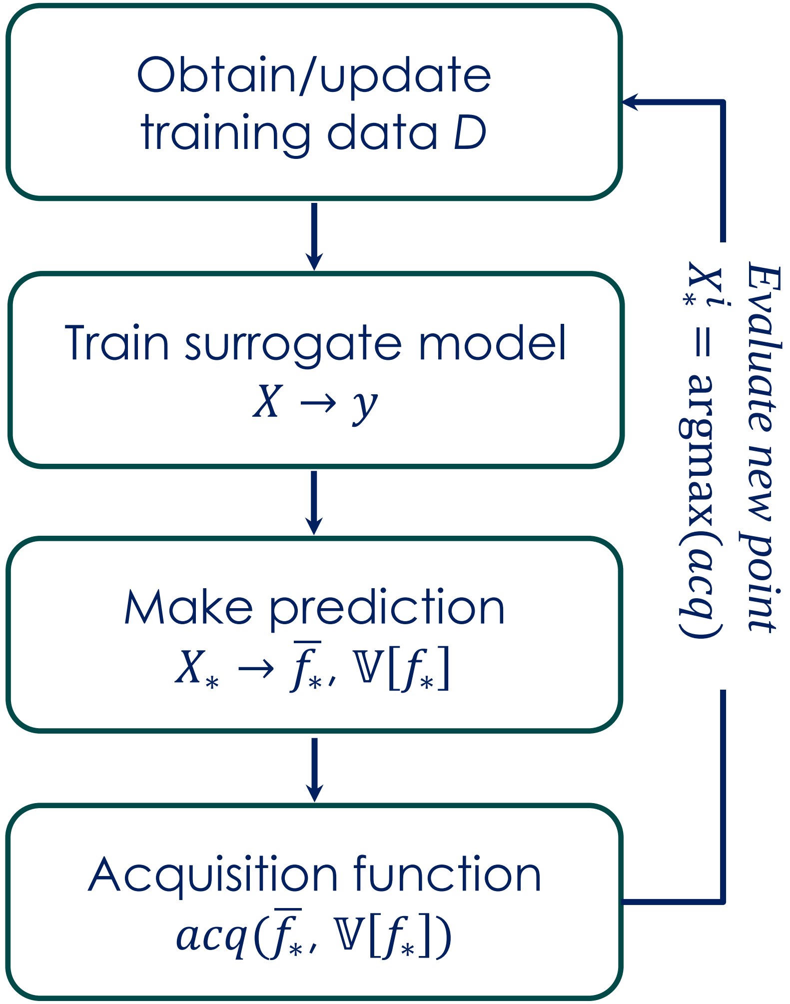
3. **Expected Improvement:** Instead of probability of improvement, we want to maximize the expected increase in the function value

4. **There are (always) more...**

The basics: Bayesian Optimization



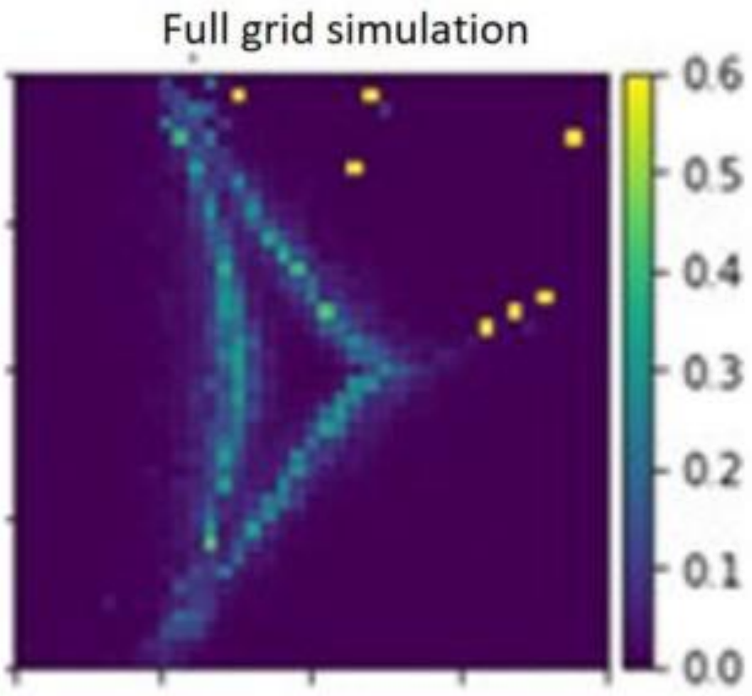
X, y: (sparse) Training data
X_*: New (not yet evaluated) points

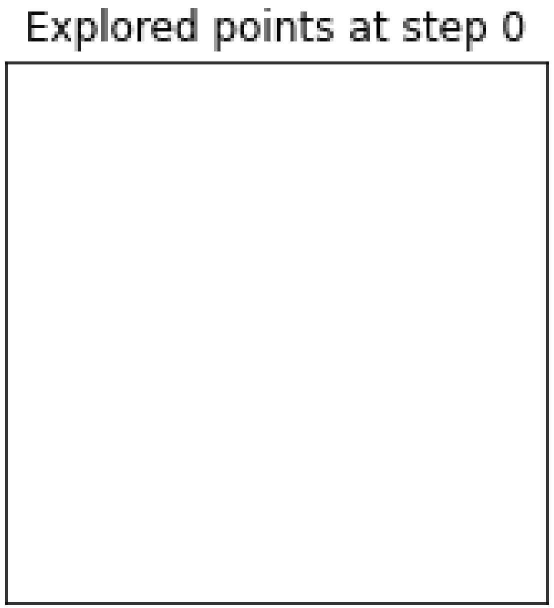


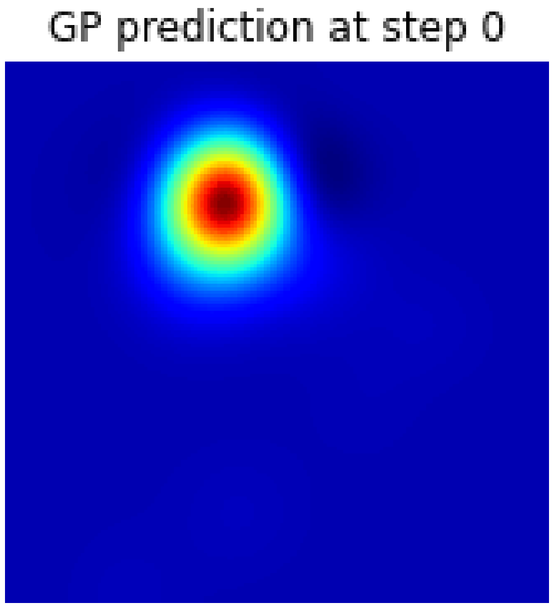
N. de Freitas et al., Taking the Human Out of the Loop: A Review of Bayesian Optimization , *Proceedings of the IEEE* **104**, 148 (2015)

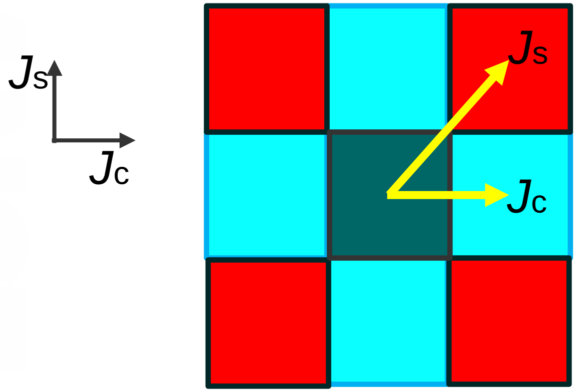
Bayesian Optimization for physical discovery

Discovering regions where heat capacity is maximized in NNN Ising model



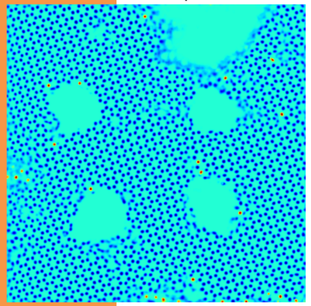




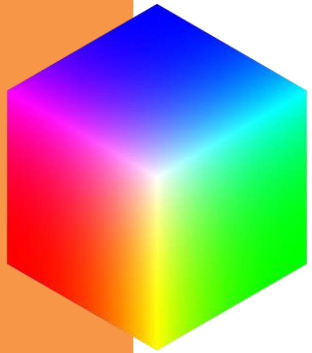


Going real time: automated experiment

SPM or STEM image



EELS or SPM datacube



- **Sliding window/linear transform**
- **Keras DCNN**
- rVAE (rotational invariance)
- rcVAE (plus classification)

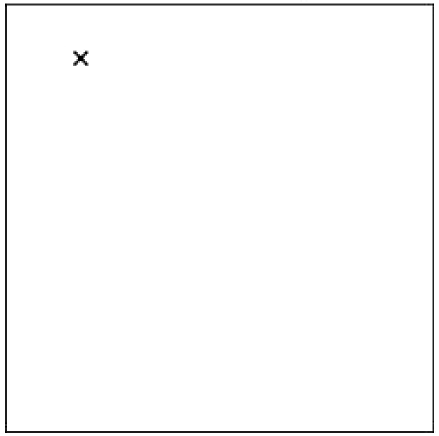
Descriptor

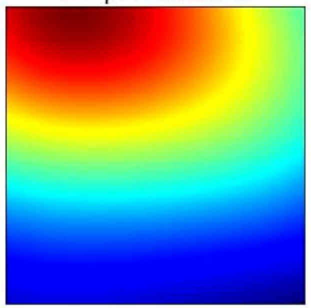
Gaussian processing

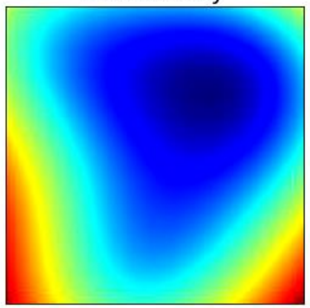
- **Integrated intensity**
- **Keras DCNN**
- Spec2im autoencoder
- (im,spec)2(spec,im)
- CycleGAN

GPim library (M. Ziatdinov)

- **Acquisition functions**
- **Pathfinder functions**
- **Kernel control**

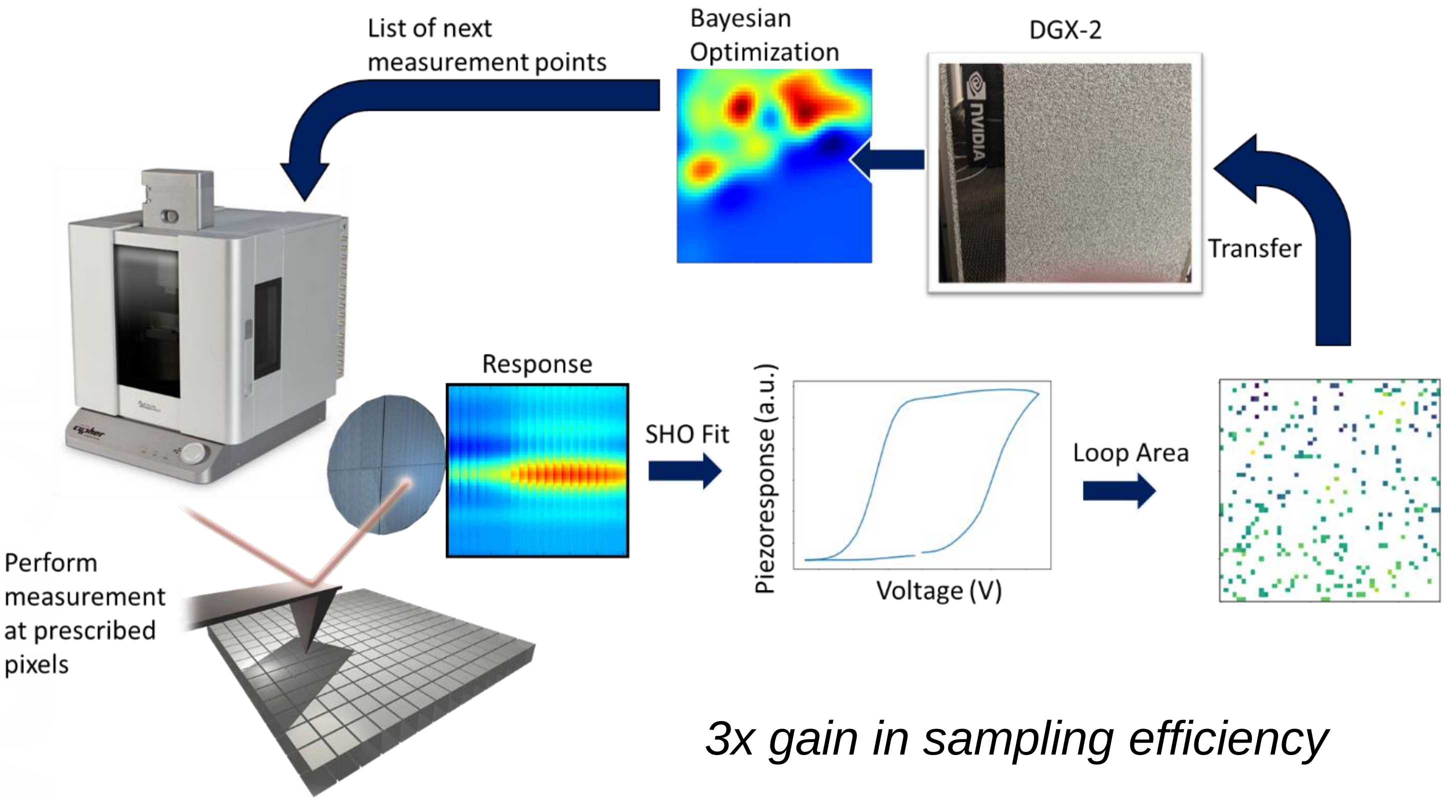






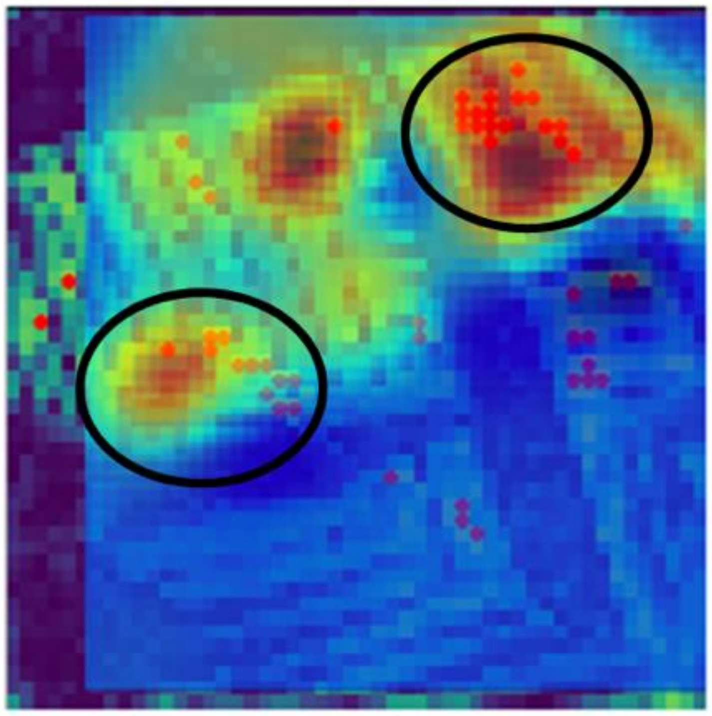
- AE based on structural analysis for STEM data
- AE based on spectral data in PFM
- AE based on DL for EELS data
- Feature of interest finding for mesoscopic images

BO for Self-Driving Microscopy



3x gain in sampling efficiency

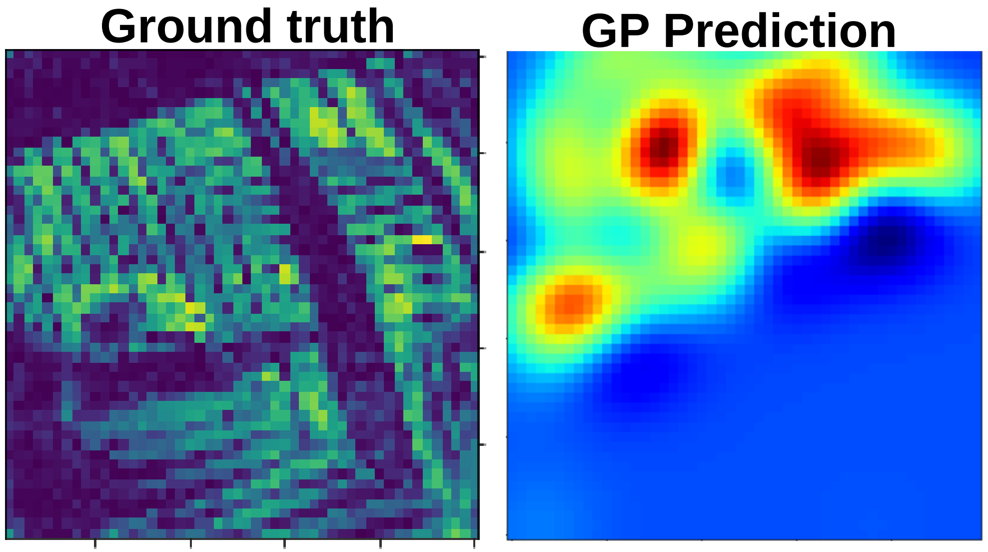
Comparison with "ground truth"



R. K. Vasudevan, K. Kelley, H. Funakubo, S. Jesse, S. V. Kalinin, M. Ziatdinov, ***ACS Nano*** (2021) *https://doi.org/10.1021/acsnano.0c10239*

Classical GP based BO

- Purely data-driven: limited advantage in high dimensional spaces
- Predicts scalar functions
- Typically used assuming equal cost of measurements
- And targeting fully automated process



Vasudevan et al, ACS Nano 2021

These assumptions rarely comport to real world scenarios

- We typically have ample (but partial) physical knowledge
- Multiple proxy signals
- Our observed data is very often high-dimensional (spectra, images)
- Cost and latencies of measurements is determined by physical equipment
- We can co-orchestrate measurements
- Humans are a part of the process (if process is slow)

But what about noises?

Gaussian Process learns the noise and kernel function while exploring parameter space. What if the noise level is not constant?

$$f(x) \sim \mathcal{GP}\left(m(x), k(x, x')\right)$$

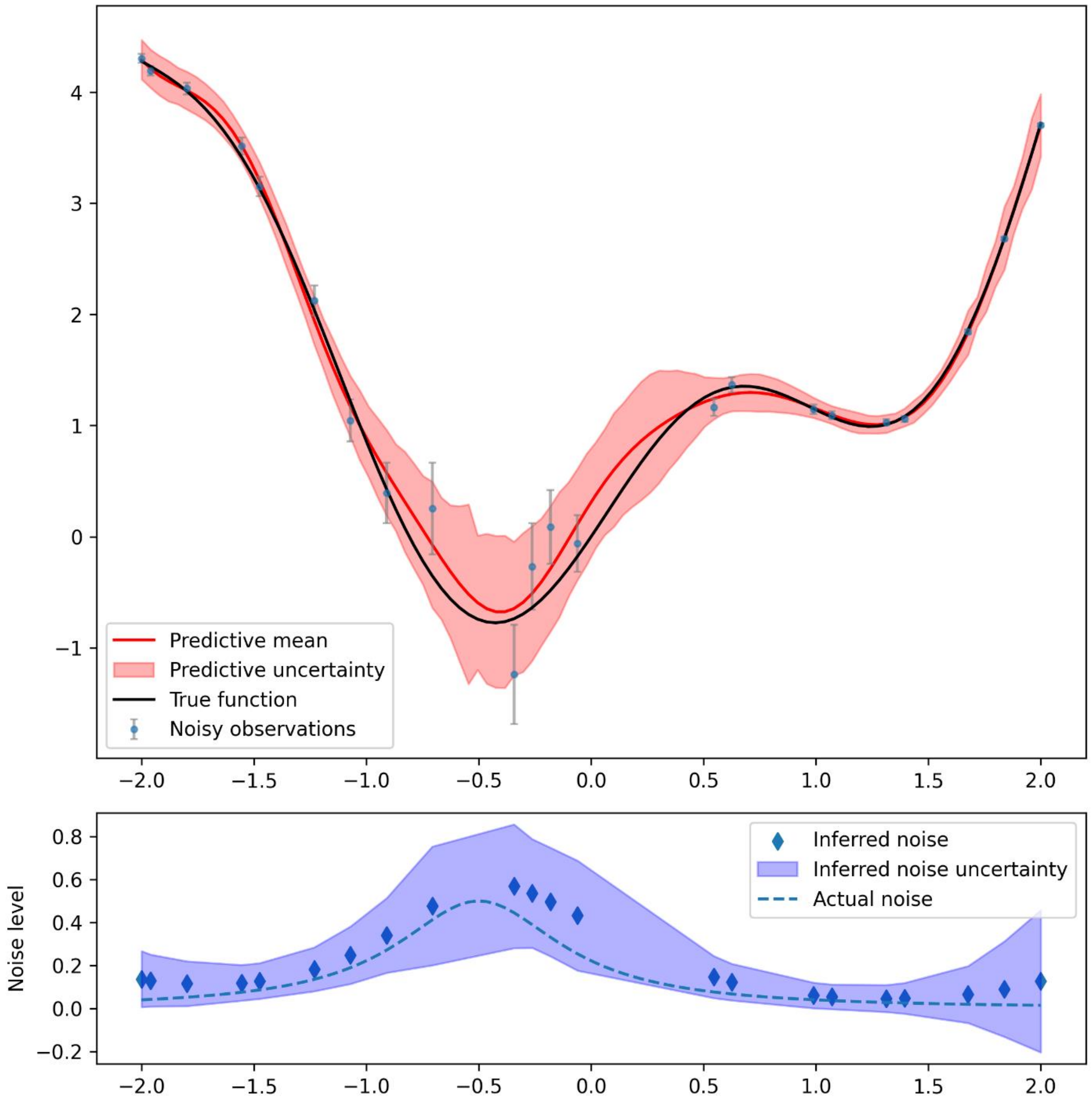
$$y(x) = f(x) + \sigma(x)$$

Solution: heteroscedastic GP uses

- one GP for function, and
- another GP for the noise

Note that we can **create models** for function and noise (structured GP)

Heteroscedastic GP



But what if the noise can be measured?

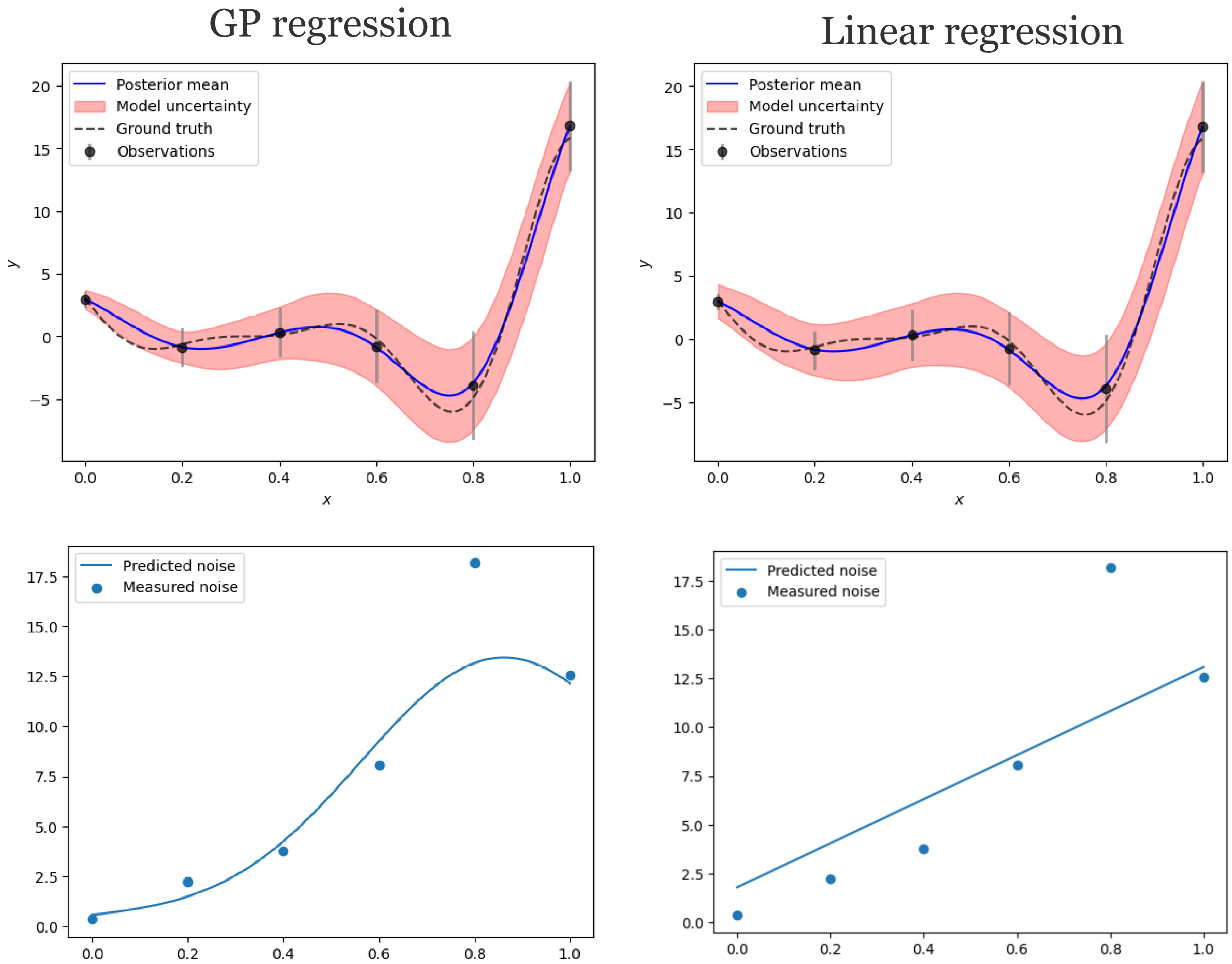
In many experimental scenarios, the experiment can be configured so that the noise can be measured (or estimated).

For example, it is often easier to measure multiple times (indentation curves, spectra, etc) at one location rather then move around

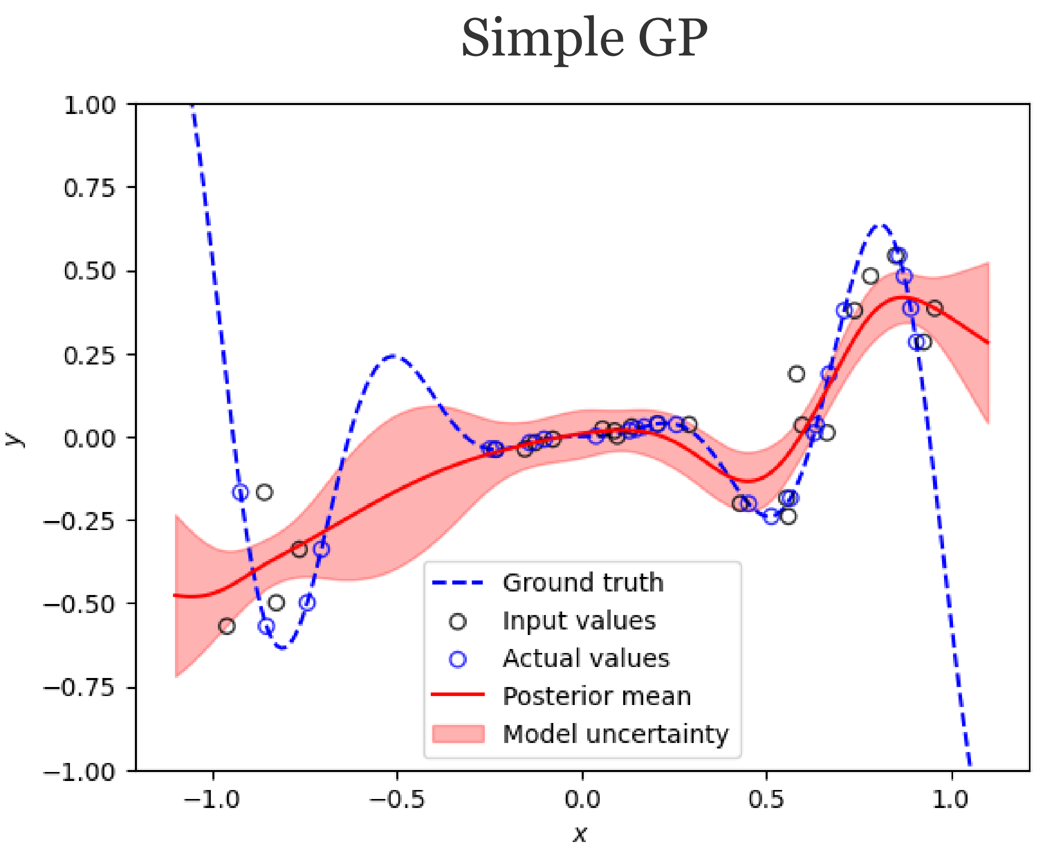
Alternatively, noise can be estimated from single measurement.

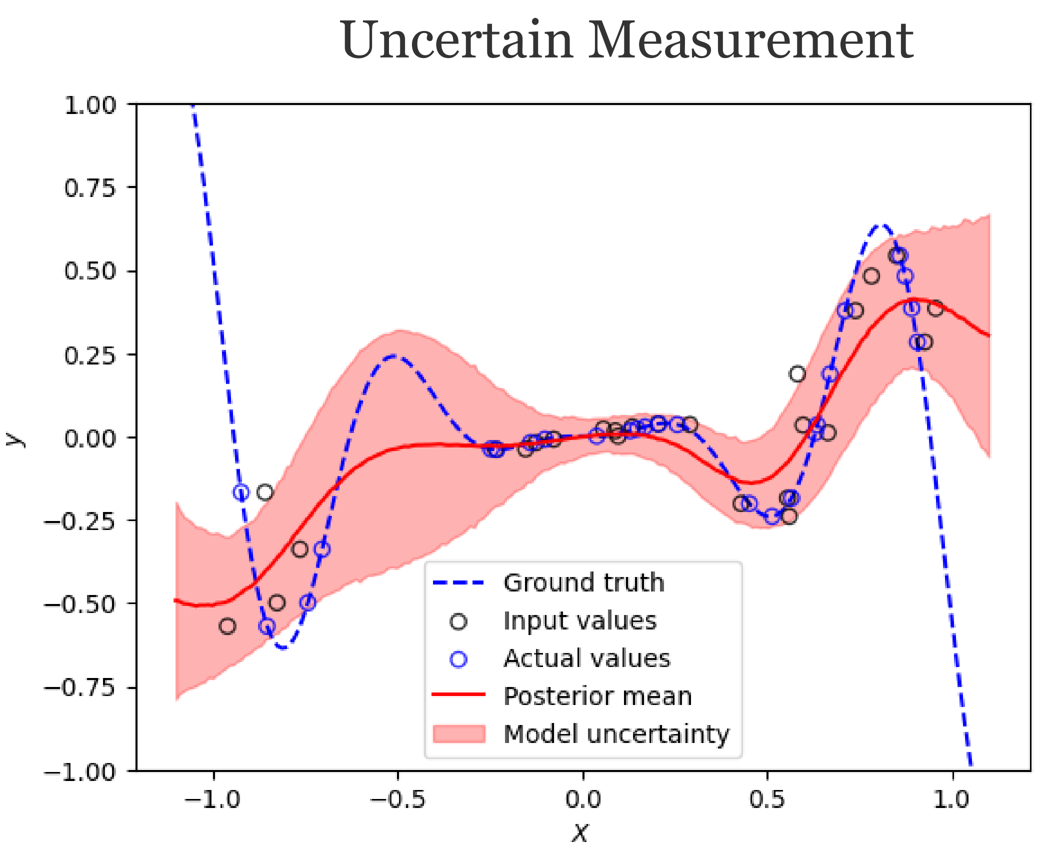
Note that we still need **noise model** even when the noise is measured, since we need to have an estimate of noise at the yet-unmeasured locations

Measured Noise GP

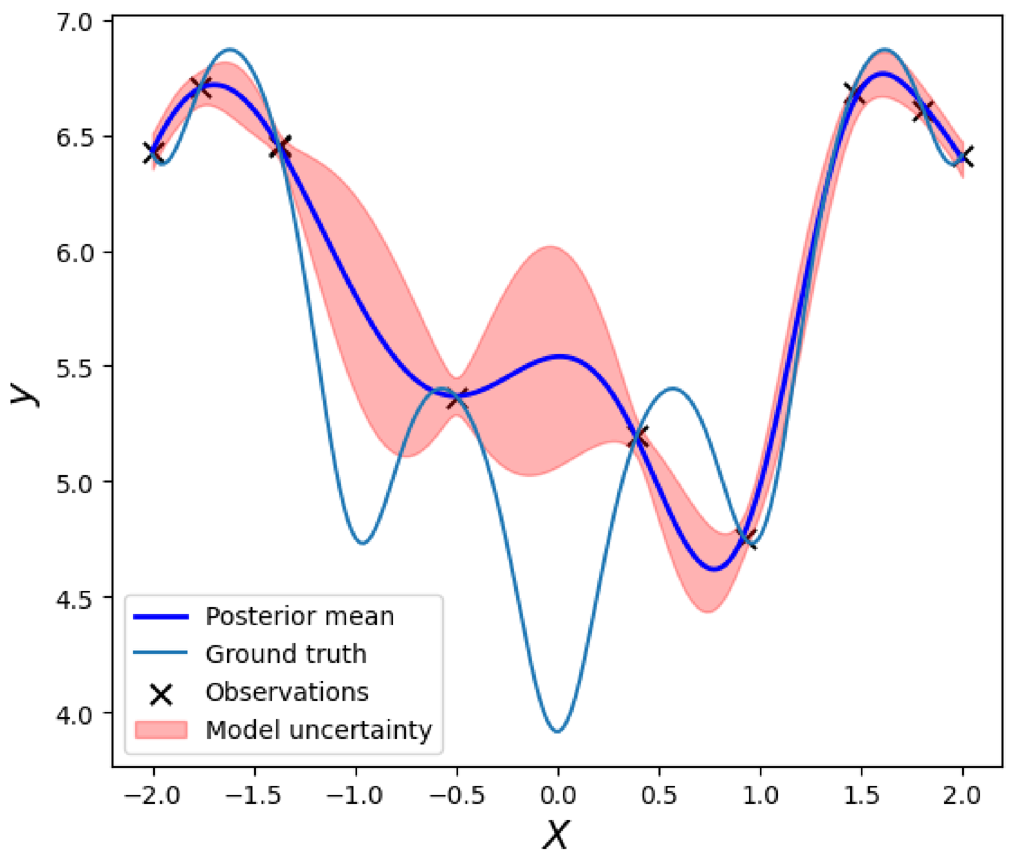


What if the measurement location in uncertain?





- Classical Bayesian Optimization is useful for microscope tuning and imaging optimization, but almost useless for exploration in image plane
- Limited to low D: we need Deep Kernel Learning for Structure-Property relationship discovery
- No physics priors: we need structured Gaussian Processes to learn physics



GP Augmented with Structural model

Define a probabilistic model:

$$\mathbf{y} \sim MVNormal(\mathbf{m}, \mathbf{K})$$

$$K_{ij} = \sigma^2 \exp(0.5(x_i - x_j)^2 / l^2$$

$$\sigma \sim LogNormal(0, s_1)$$

$$l \sim LogNormal(0, s_2)$$

- We substitute a constant GP prior mean function **m** with a structured probabilistic model of the expected behavior.
- This probabilistic model reflects our prior knowledge about the system, but it does not have to be precise.
- The model parameters are inferred together with the kernel parameters via the Hamiltonian Monte Carlo.
- The fully Bayesian treatment of the model allows additional control over the optimization via the selection of priors for the model parameters.

Prediction on new data X_*:

$$\mathbf{f}_*^i \sim MVNormal\left(\mu_{\boldsymbol{\theta}^i}^{\text{post}}, \Sigma_{\boldsymbol{\theta}^i}^{\text{post}}\right)$$

replaced with

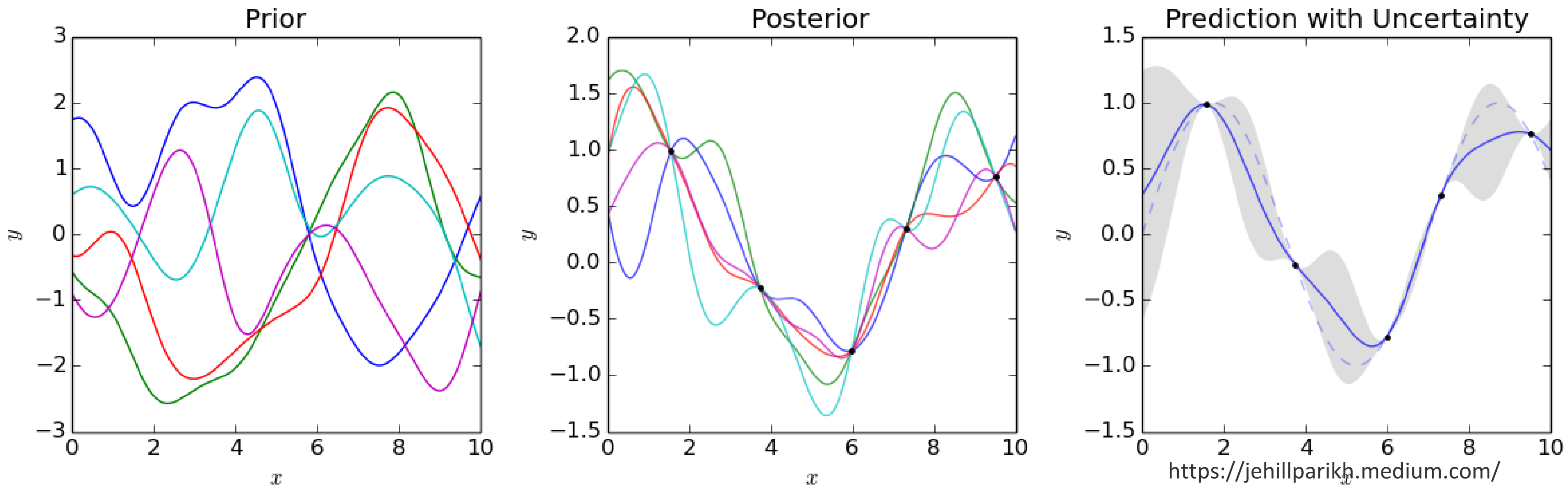
$$\mu_{\boldsymbol{\theta}^i}^{\text{post}} = \mathbf{m}(X_*) + \mathbf{K}(X_*, X|\boldsymbol{\theta}^i)\mathbf{K}(X, X|\boldsymbol{\theta}^i)^{-1}(\mathbf{y} - \mathbf{m}(X)) \rightarrow \mu_{\boldsymbol{\Omega}^i}^{\text{post}} = \mathbf{m}(X_*|\phi^i) + \mathbf{K}(X_*, X|\boldsymbol{\theta}^i)\mathbf{K}(X, X|\boldsymbol{\theta}^i)^{-1}(\mathbf{y} - \mathbf{m}(X|\phi^i))$$

$$\Sigma_{\boldsymbol{\theta}^i}^{\text{post}} = \mathbf{K}(X_*, X_*|\boldsymbol{\theta}^i) - \mathbf{K}(X_*, X|\boldsymbol{\theta}^i)\mathbf{K}(X, X|\boldsymbol{\theta}^i)^{-1}\mathbf{K}(X, X_*|\boldsymbol{\theta}^i)$$

$\boldsymbol{\Omega}^i = \{\phi^i, \boldsymbol{\theta}^i\}$ is a single HMC posterior sample with the kernel and prob model parameters

GP Augmented with Structural Model

Standard Gaussian process aims to discover function based on learned correlations (kernel)



Probabilistic model

$$m = y_0 - \sum_{n=1}^{N} L_n \quad (N{=}2)$$

$$y_0 \sim Uniform(-10, 10)$$

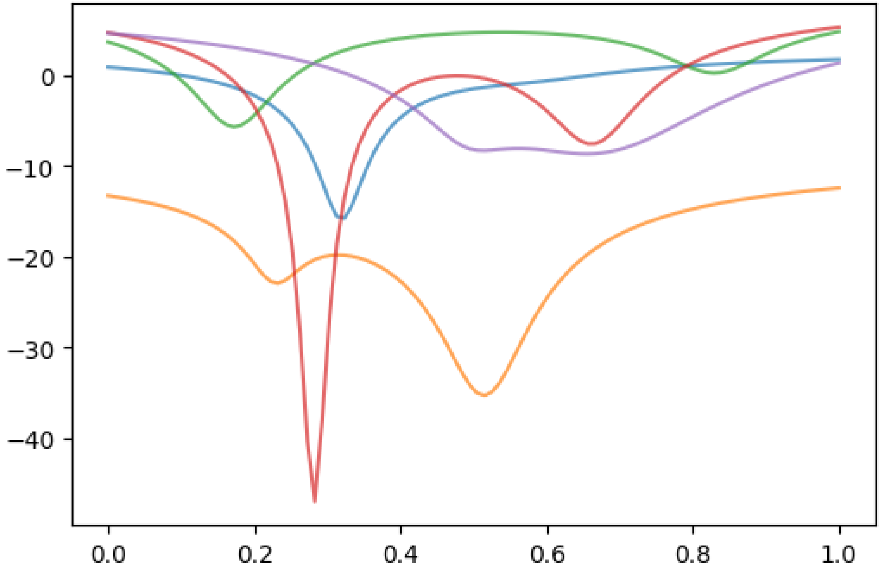
$$L_n \sim \frac{A_n}{\sqrt{(x - x_n^0)^2 + w_n^2}}$$

$$A_n \sim LogNormal(0, 1)$$

$$w_n \sim HalfNormal(.1)$$

$$x_n^0 \sim Uniform(0, 1)$$

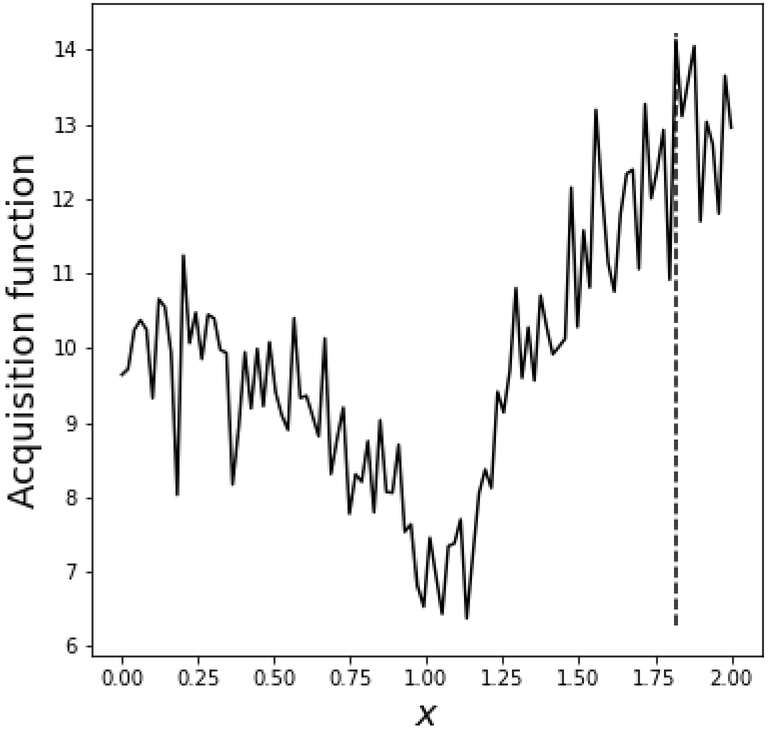
Prior predictive distribution

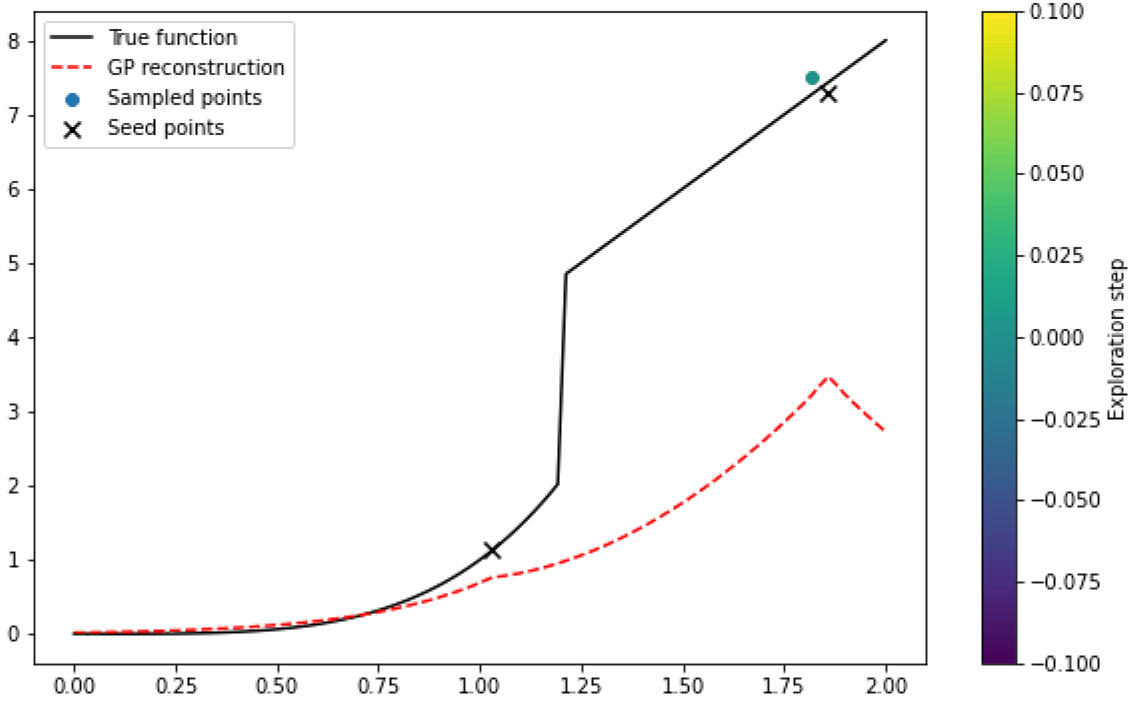


This model simply tells us that there are two minima in our data but does not assume to have any prior knowledge about their relative depth, width, or distance

Simple GP search

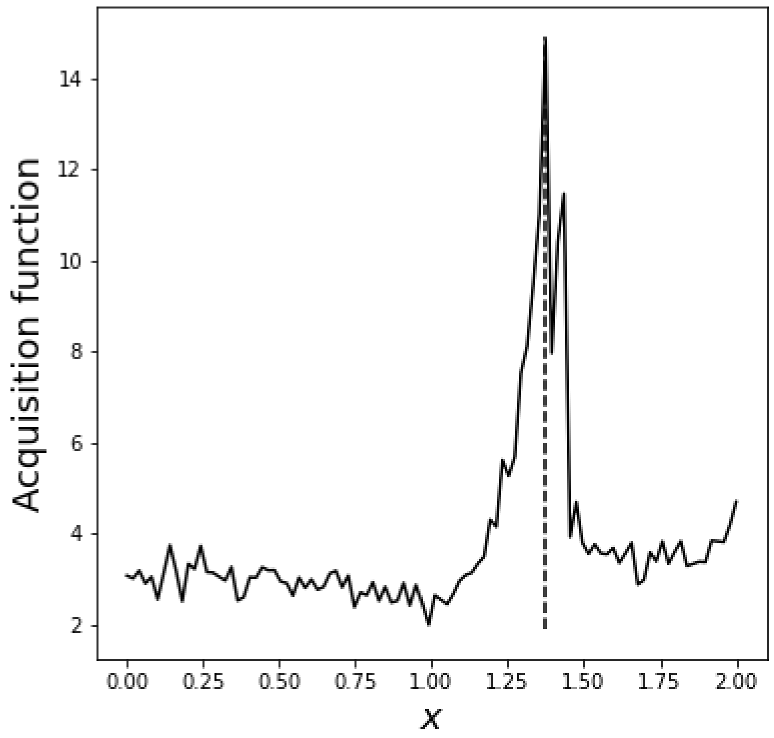
THE UNIVERSITY of TENNESSEE KNOXVILLE

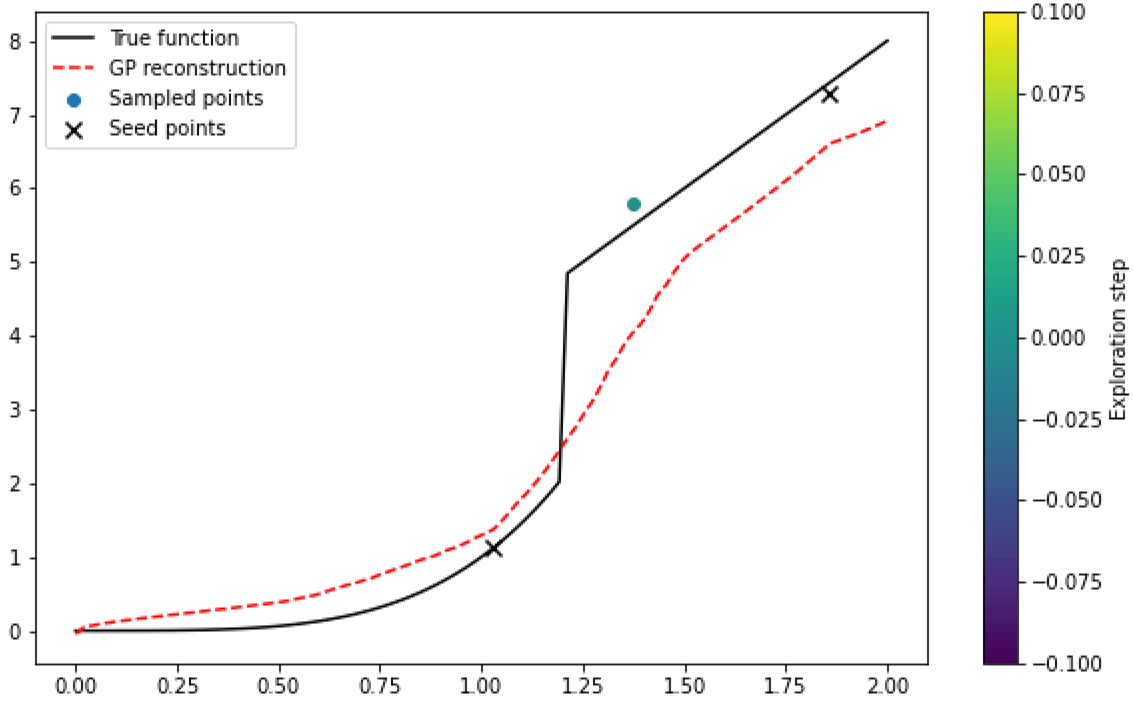




Structured GP search

THE UNIVERSITY of TENNESSEE KNOXVILLE



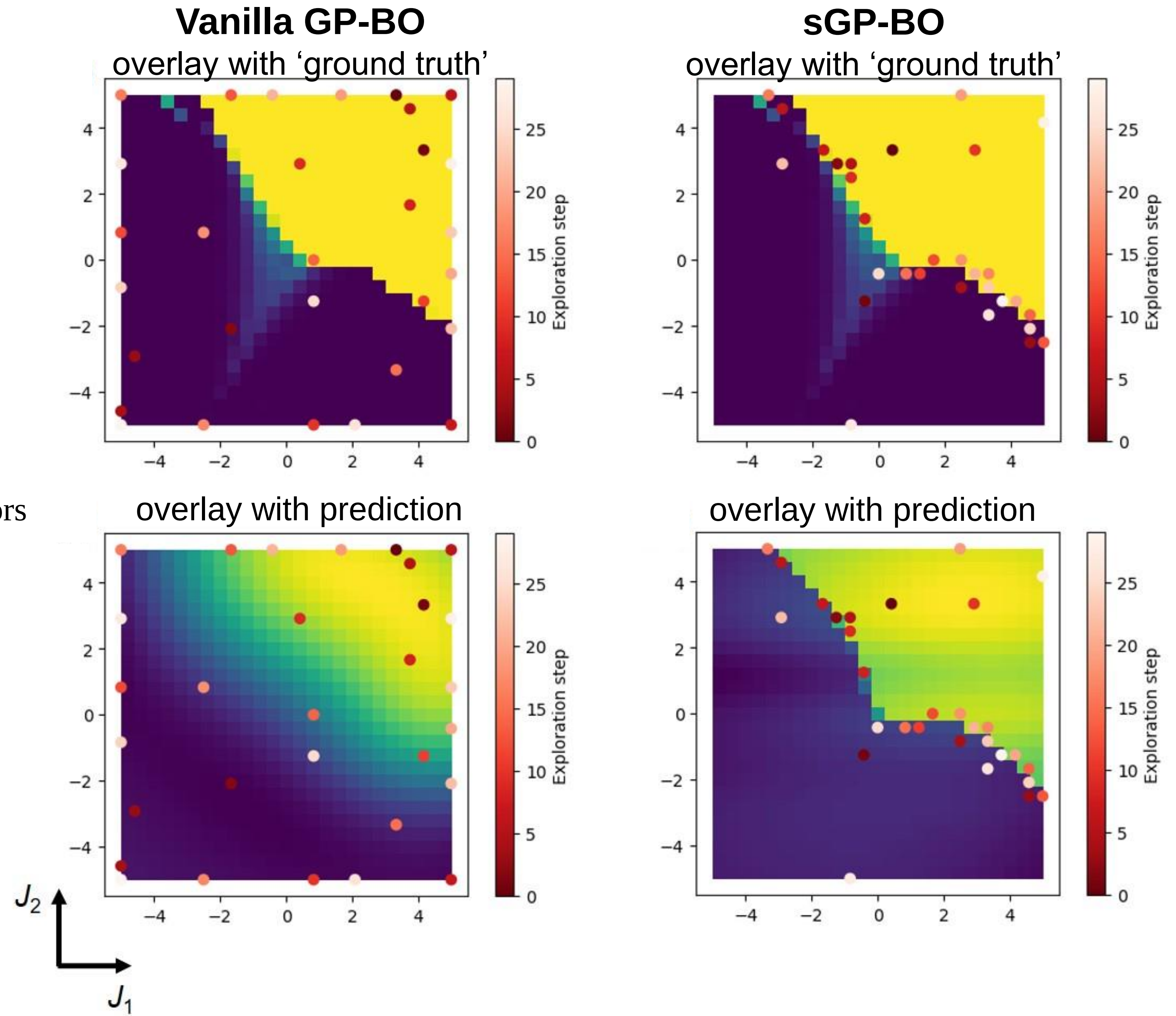


Application to Ising model

Probabilistic model

$A/\tanh(\frac{f(J_1)+f(J_2)}{w})$

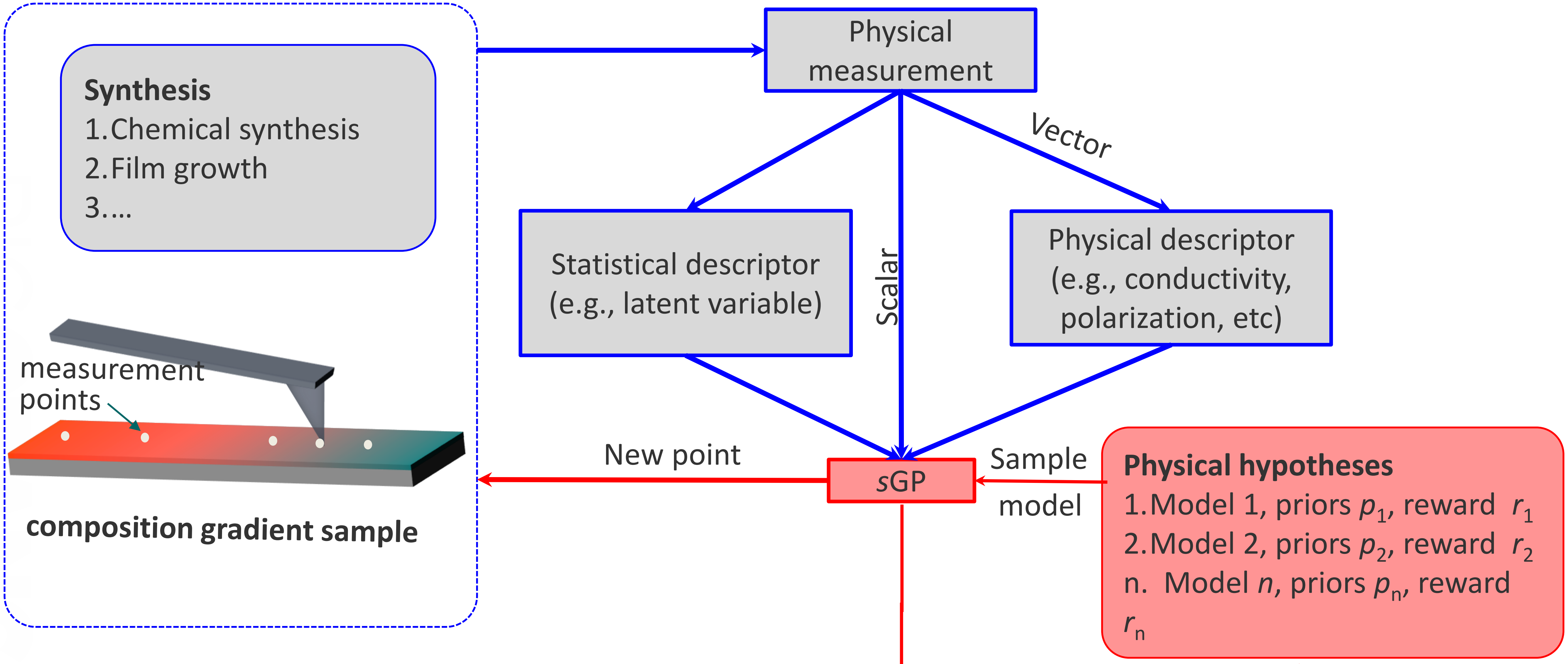
where $f(J)$ is a third-degree polynomial with normal priors on its parameters



Hypothesis Active Learning

Co-navigation of experimental and hypothesis spaces

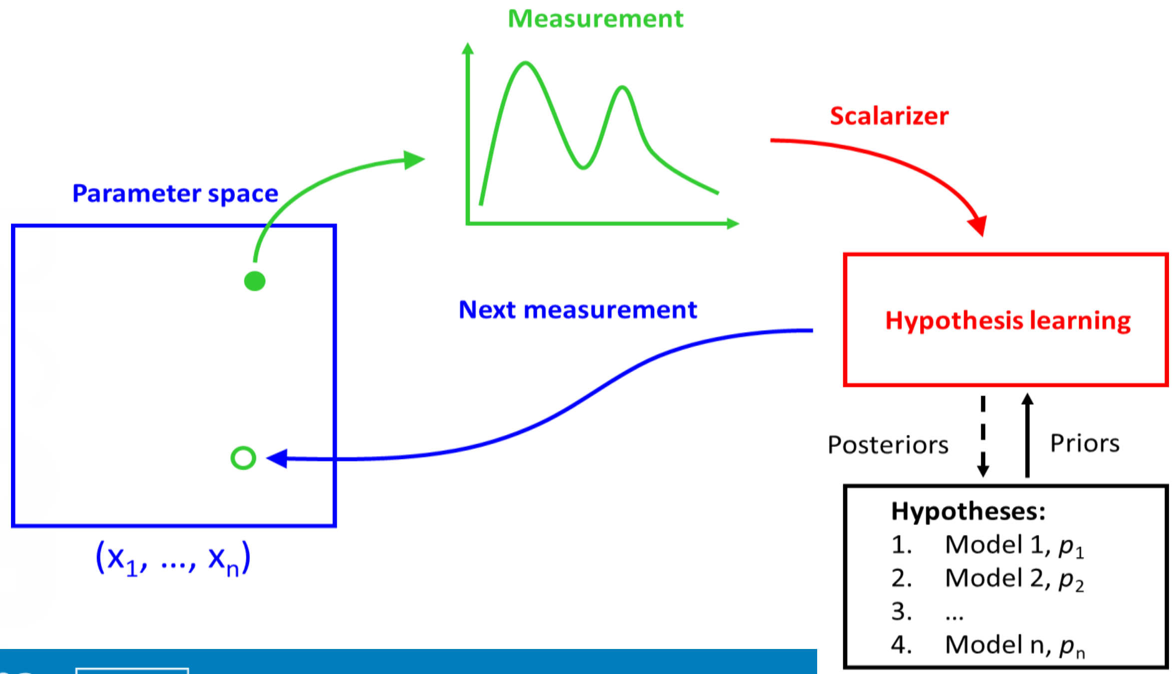
Goal: Learn (1) physical property distribution and (2) a correct model of system's behavior

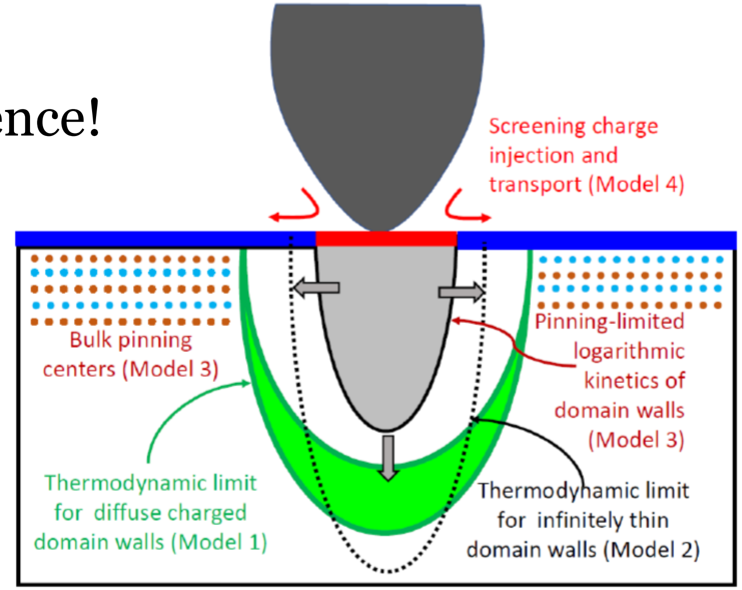


M. Ziatdinov et al. *Advanced Materials* 2201345 (2022)
Y. Liu et al., *Patterns* 4, 100704 (2023)

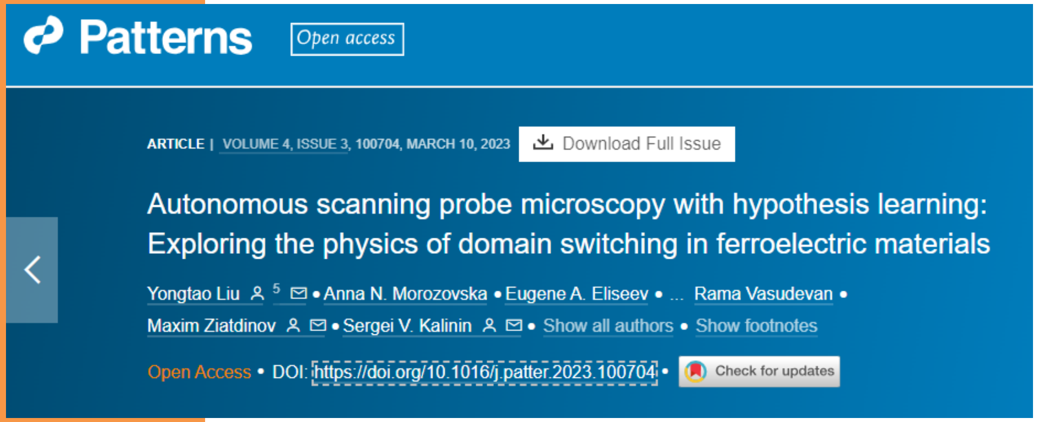
Hypothesis Learning

- Can ML algorithm think like a scientist?
- Yes – automated experiment can pursue hypothesis-driven science!





		Model Equation
Thermodynamic 1	**Model I**	$r(V) = r_{cr} + r_0\sqrt{\left(\frac{V}{V_c}\right)^{2/3} - 1}$
Thermodynamic 2	**Model II**	$r(V) = r_{cr} + r_0\sqrt[3]{\left(\frac{V}{V_c}\right)^{2} - 1}$
Wall pinning	**Model III**	$r(V,t) = V^{\alpha}\log\tau$
Charge injection	**Model IV**	$r(V,t) = V^{\alpha}\tau^{\beta}$

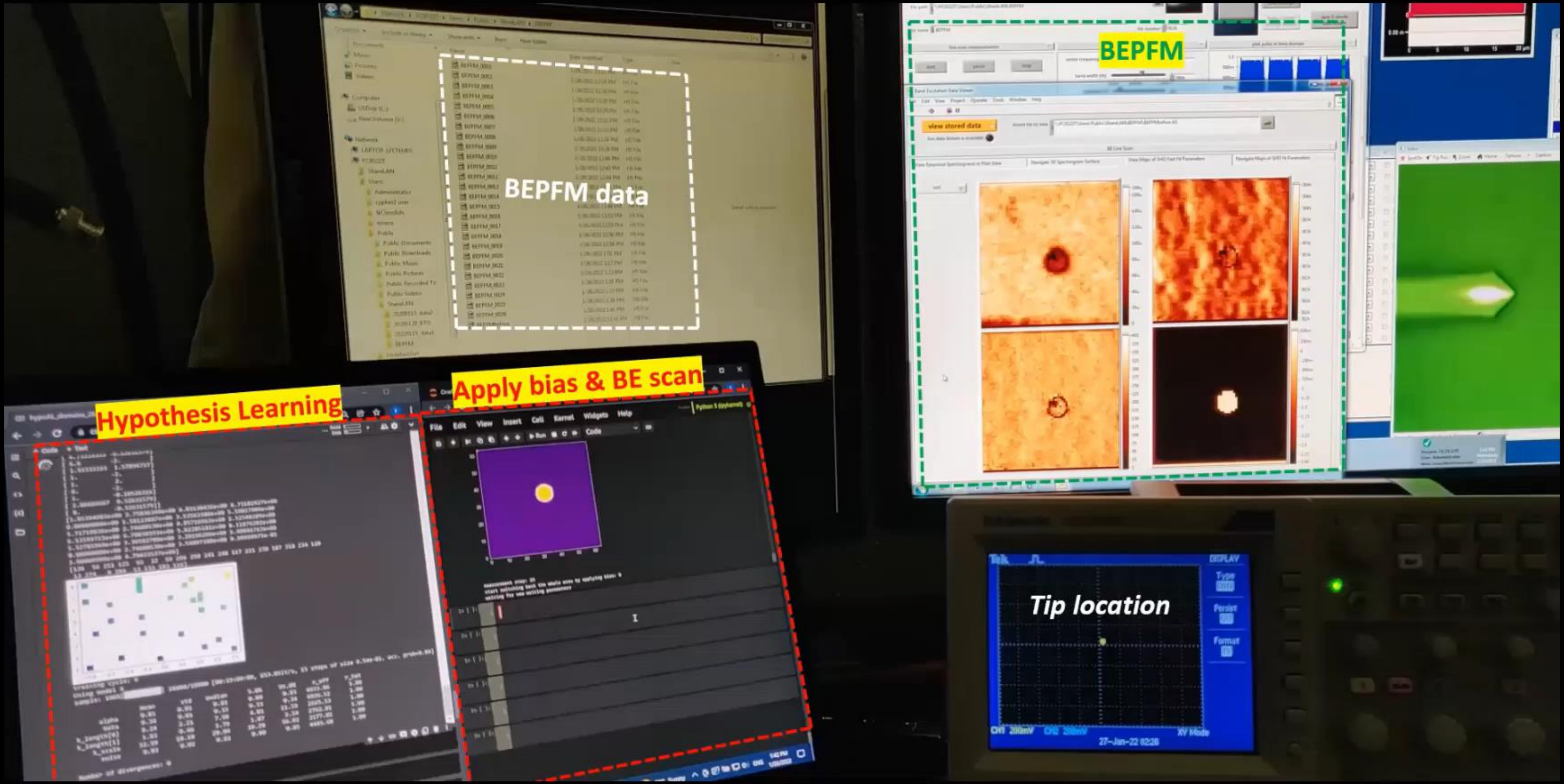

Patterns Open access

ARTICLE | VOLUME 4, ISSUE 3, 100704, MARCH 10, 2023 Download Full Issue

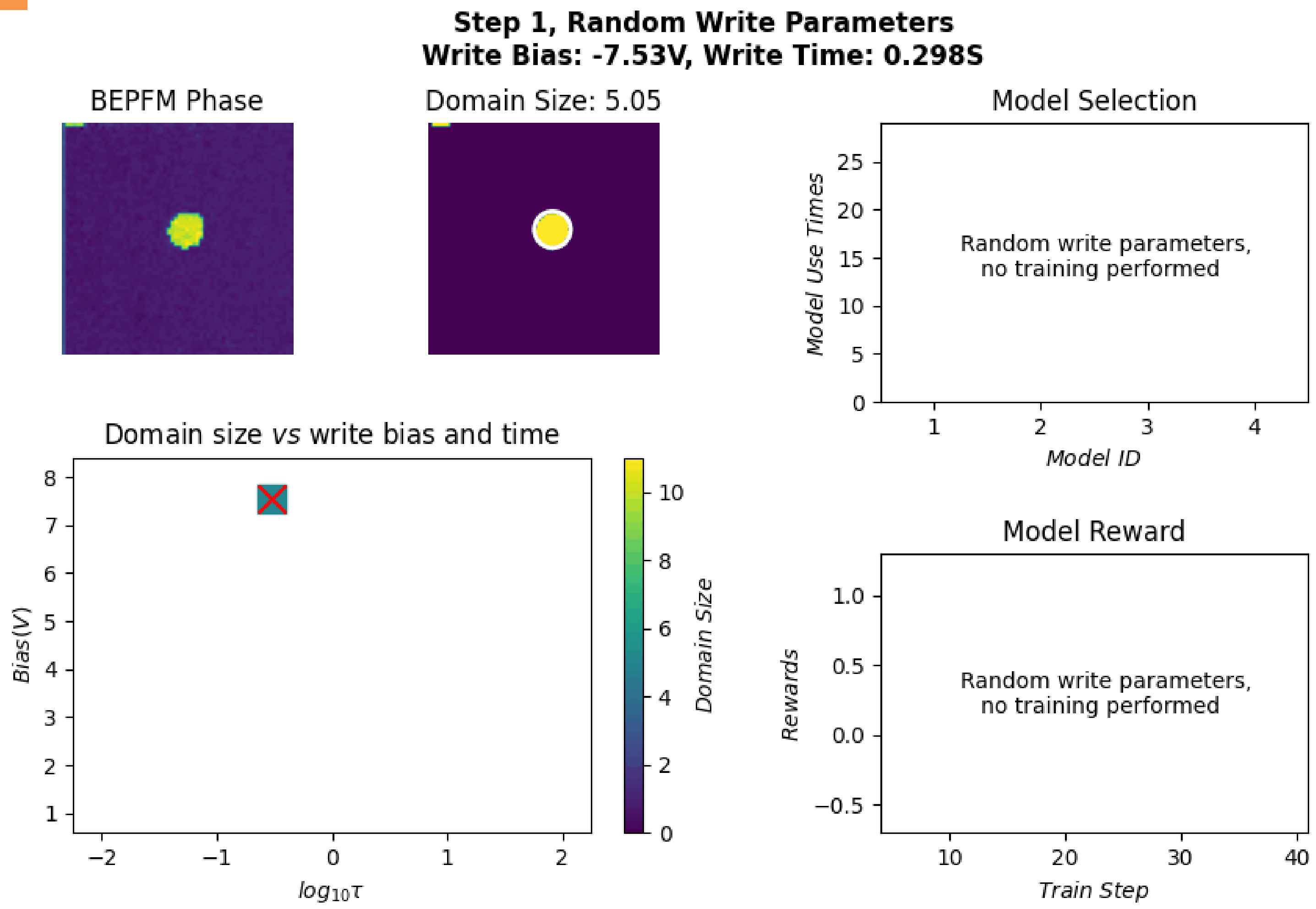
Autonomous scanning probe microscopy with hypothesis learning: Exploring the physics of domain switching in ferroelectric materials

Yongtao Liu [5] • Anna N. Morozovska • Eugene A. Eliseev • … Rama Vasudevan • Maxim Ziatdinov • Sergei V. Kalinin • Show all authors • Show footnotes

Open Access • DOI: https://doi.org/10.1016/j.patter.2023.100704 • Check for updates


BEPFM data
BEPFM
Hypothesis Learning
Apply bias & BE scan
Tip location

Hypothesis learning in action

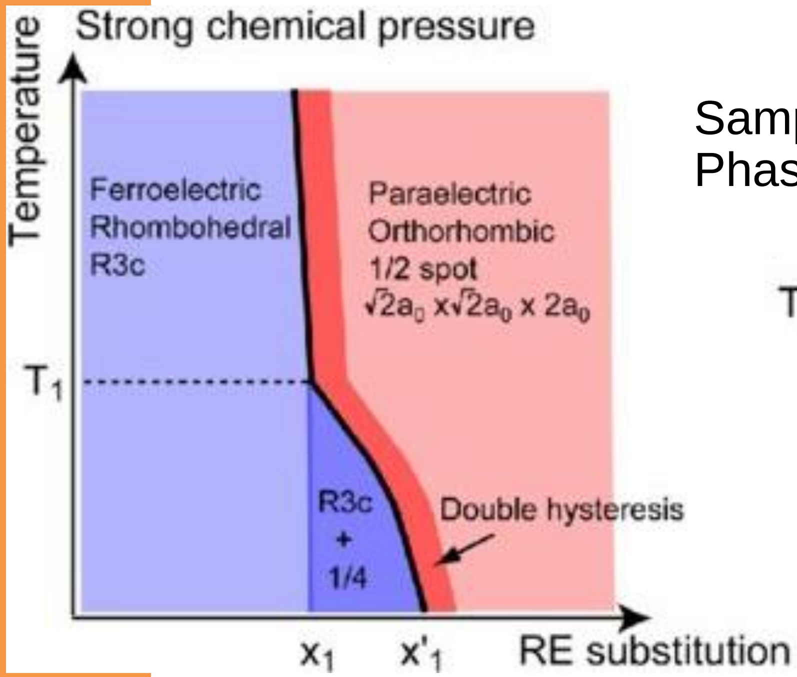


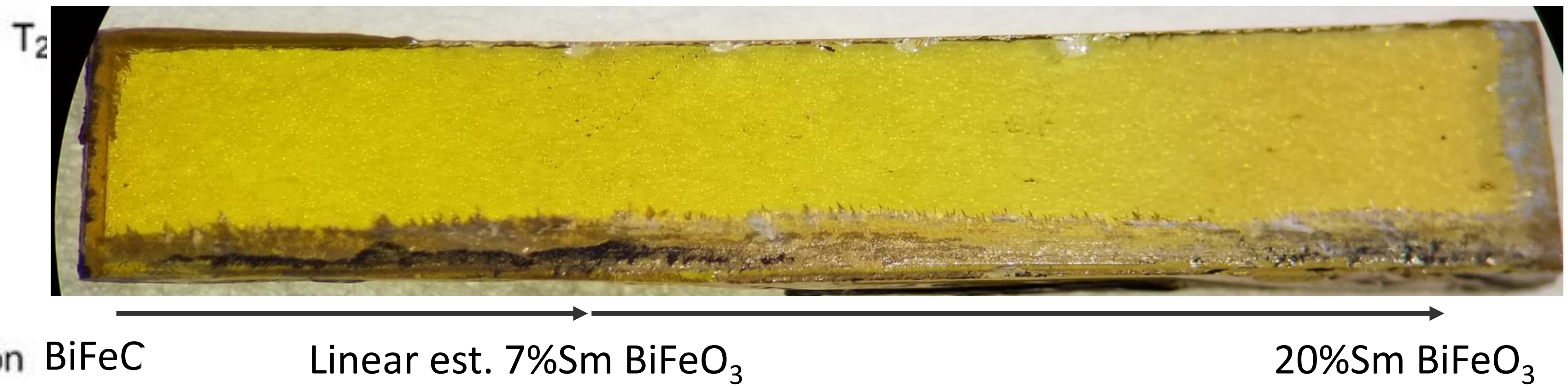
Y. Liu, arxiv 2202.01089
Y. Liu, arxiv 2112.06649

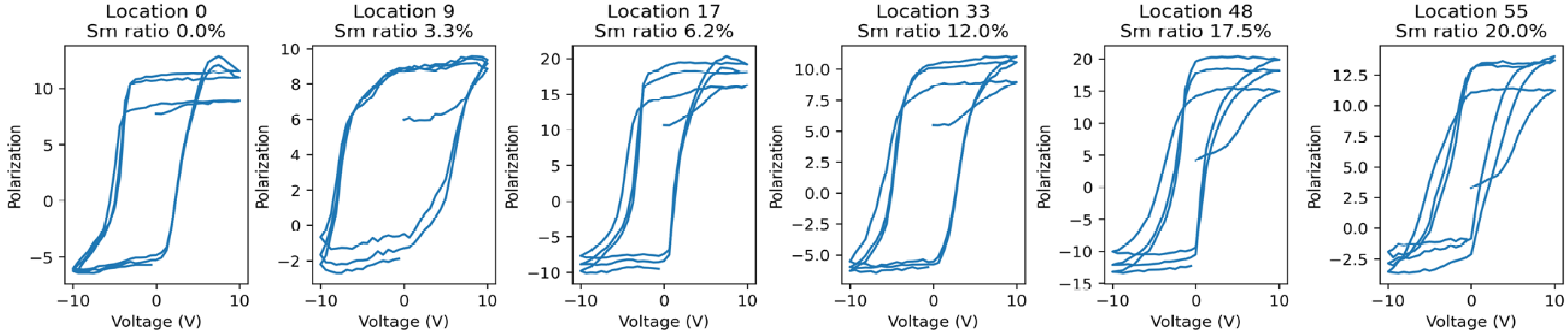
- ML algorithm has 4 competing hypothesis on domain switching mechanisms
- These hypothesis represent full set of possibilities for this system
- The microscope chooses experimental parameters in such a way as to establish which hypothesis is correct fastest
- Important: the same approach can be implemented in synthesis and electrical characterization
- Machine learning meets hypothesis-driven scientific discovery!

Combinatorial Synthesis

Sample by I. Takeuchi, UMD
Phase diagram by N. Valanoor et al.







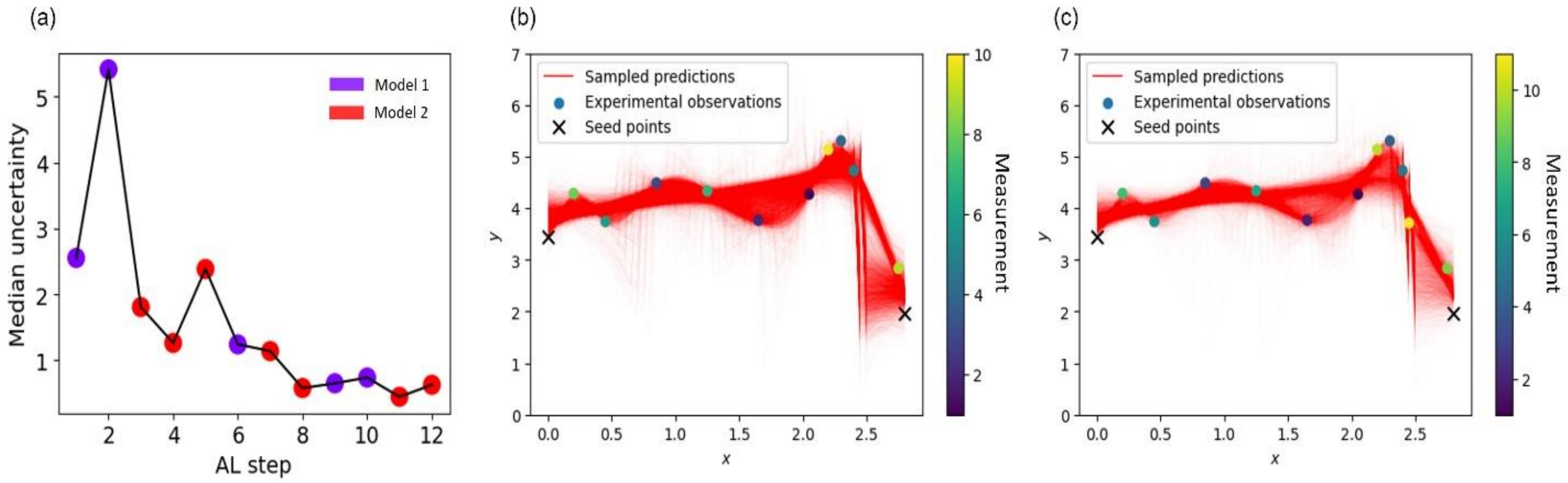
Hypothesis Selection for Ferroelectric

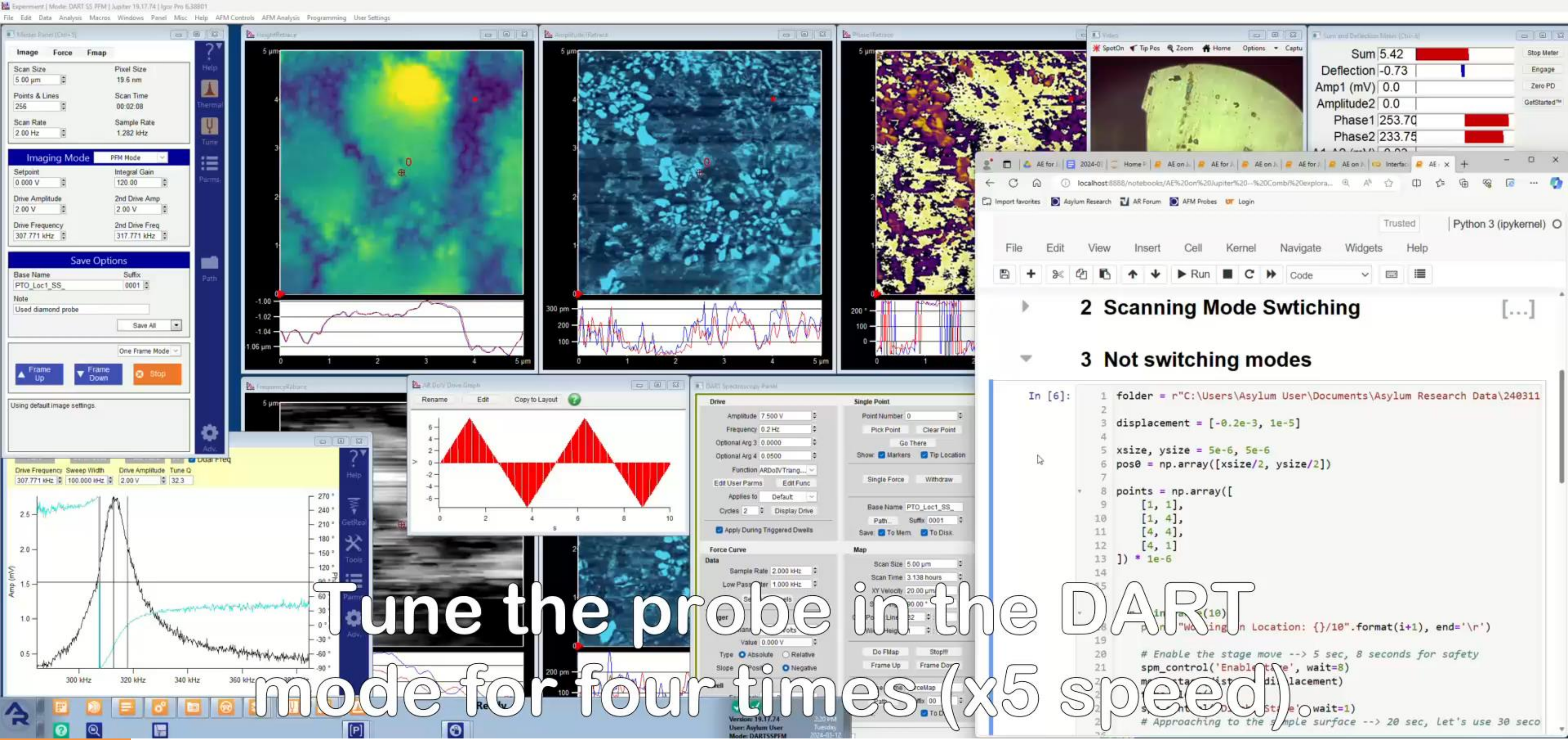
Model 1 (second order phase transition):

$$S = \begin{cases} S_0\left(1 - \dfrac{x}{x_0}\right)^2 + C, & x \le x_c, \\ C, & x > x_c \end{cases}$$

Model 2 (first order phase transition):

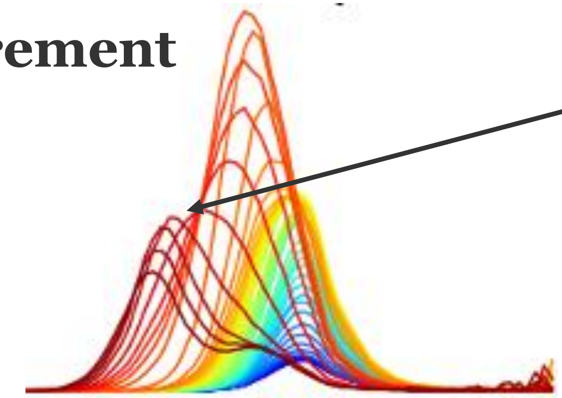
$$S = \begin{cases} S_0\left(1 - \dfrac{x}{x_0}\right)^{\frac{5}{4}} + C_0, & x \le x_c, \\ C_1, & x > x_c \end{cases}$$




2 Scanning Mode Swtiching
3 Not switching modes
In [6]:
folder = r"C:\Users\Asylum User\Documents\Asylum Research Data\240311
displacement = [-0.2e-3, 1e-5]
xsize, ysize = 5e-6, 5e-6
pos0 = np.array([xsize/2, ysize/2])
points = np.array([
[1, 1],
[1, 4],
[4, 4],
[4, 1]
]) * 1e-6
Enable the stage move --> 5 sec, 8 seconds for safety
Approaching to the sample surface --> 20 sec, let's use 30 seco
Tune the probe in the DART mode for four times (x5 speed)

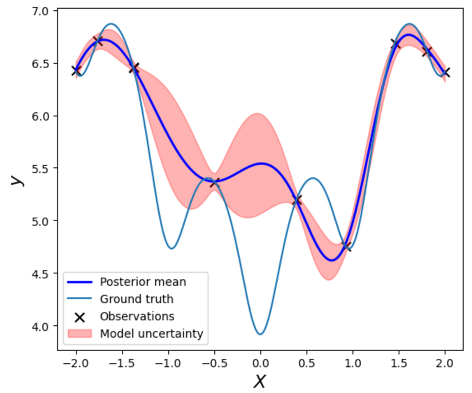
Classical Bayesian Optimization

Measurement

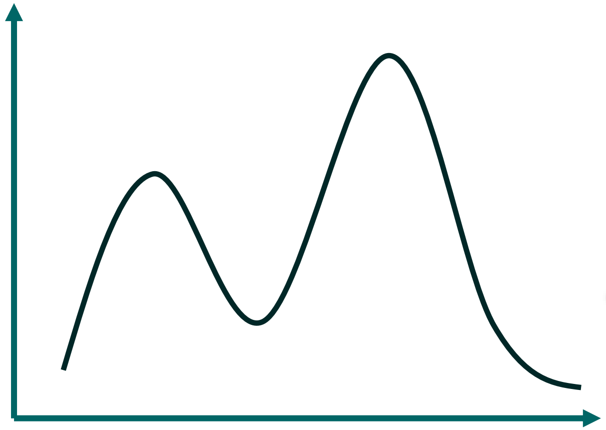


Scalarizer!

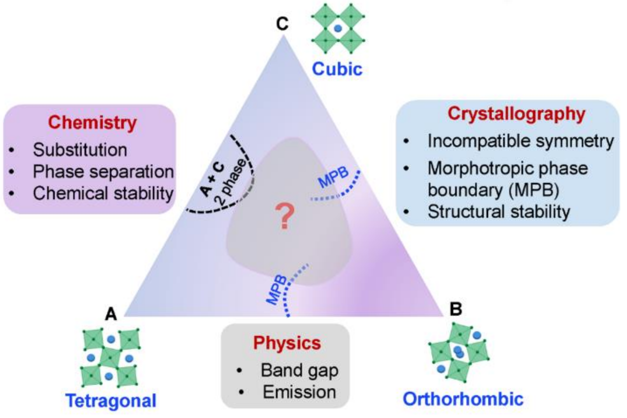
Surrogate model



Acquisition function

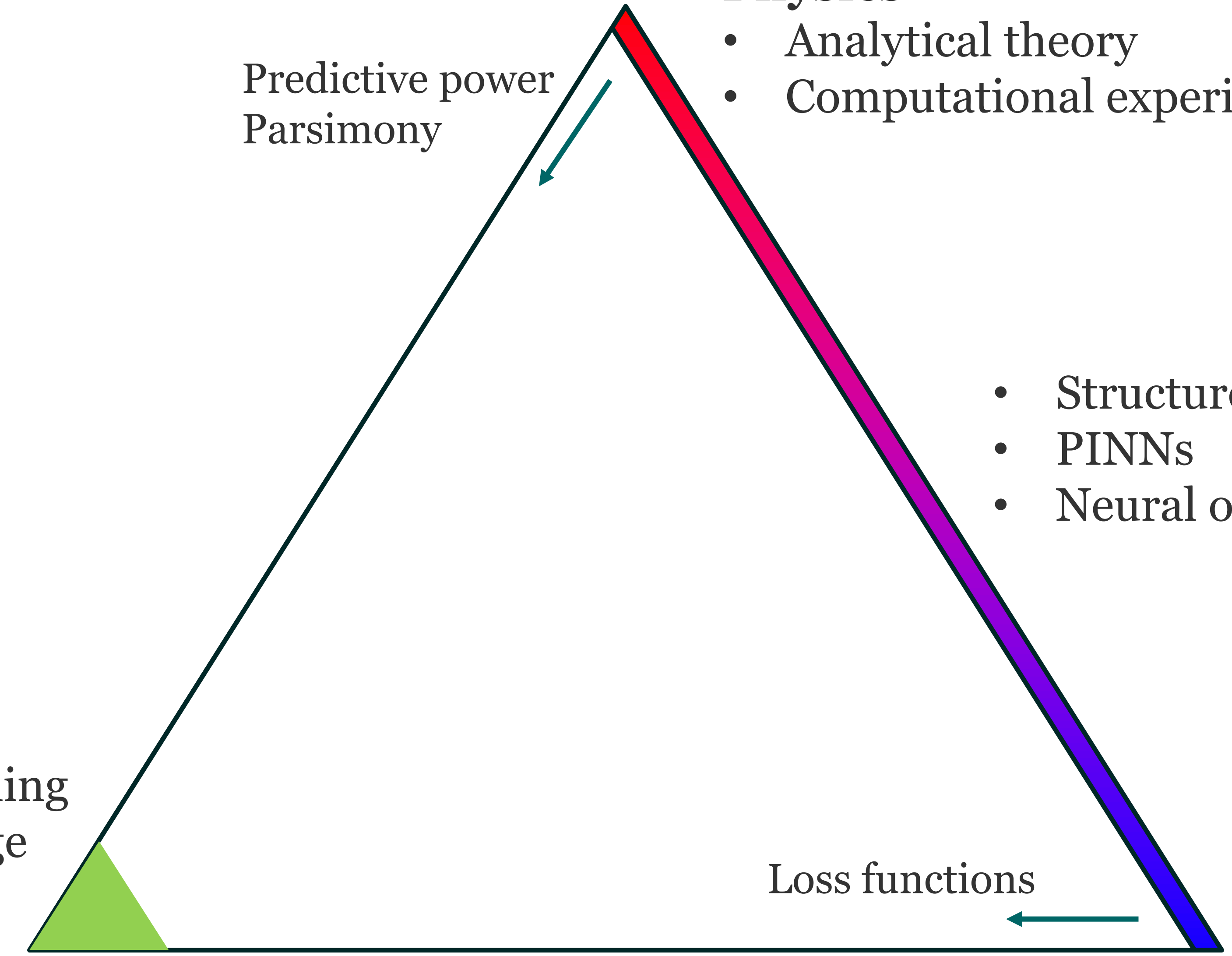


Parameter space

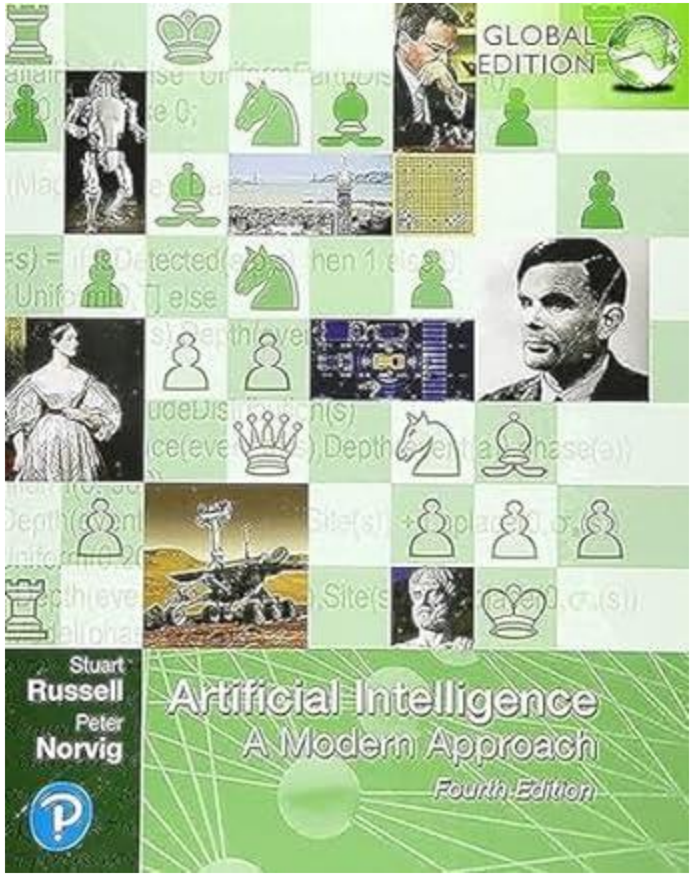


Implicitly, we postulate that cost and gain are constant along the iterative cycle

Is Machine Learning and Physics Enough?



Somewhat remarkably, almost all AI research until very recently has assumed that the performance measure can be exactly and correctly specified in the form of utility or reward function



Reward functions in imaging

Imaging Optimization

Physical laws discovery

Image-based reward functions
- Human selected objects (DCNNs)
- Equal sampling of feature space
- Equal sampling of parameter space (combi library)

Structure property relationship discovery
- Reward definition (with cost)
- Tuning curiosity
- Human in the loop DKL

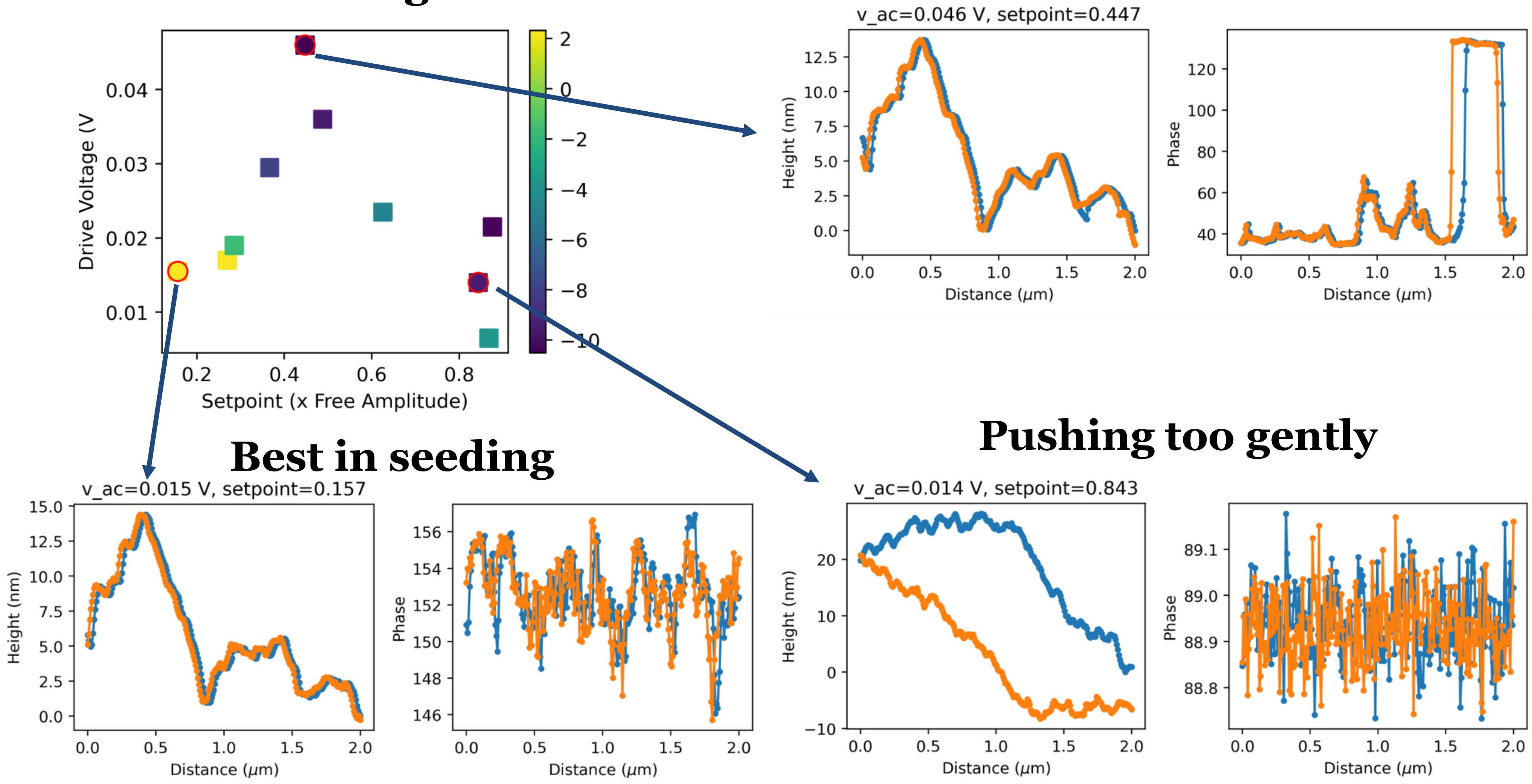
Co-orchestration multiple tools
Co-navigation between theory and experiment

Reward functions in imaging

- There are ~100,000 AFMs in the world
- Each day of AFM operator starts with tuning

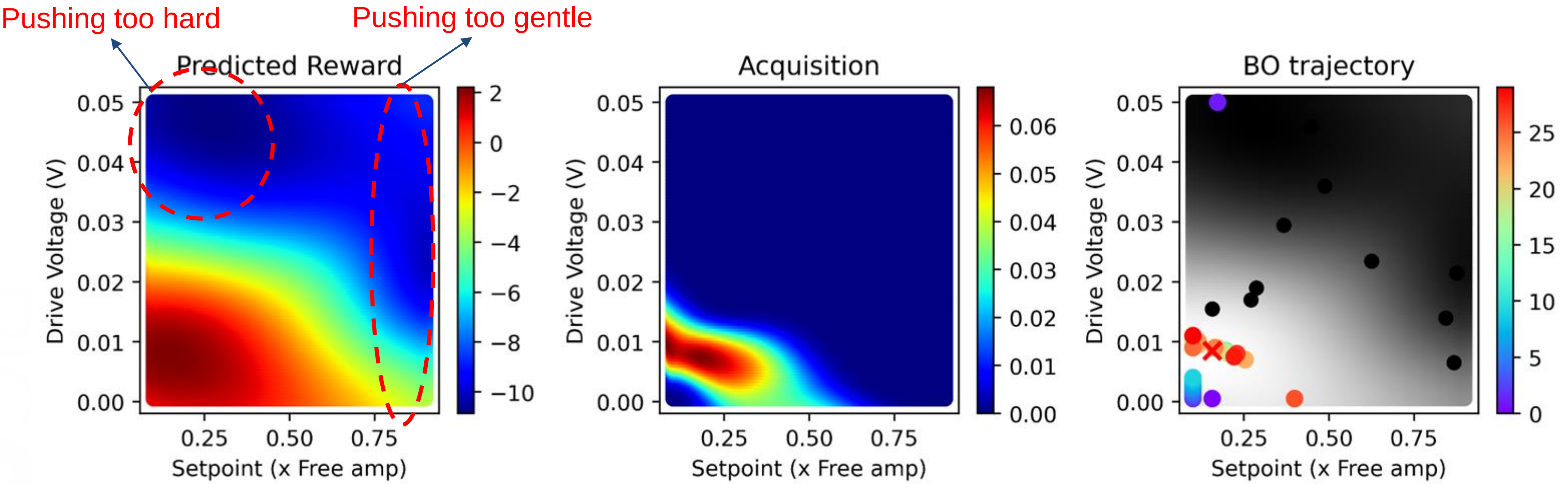
Reward of 10 seeding measurements

Pushing too hard



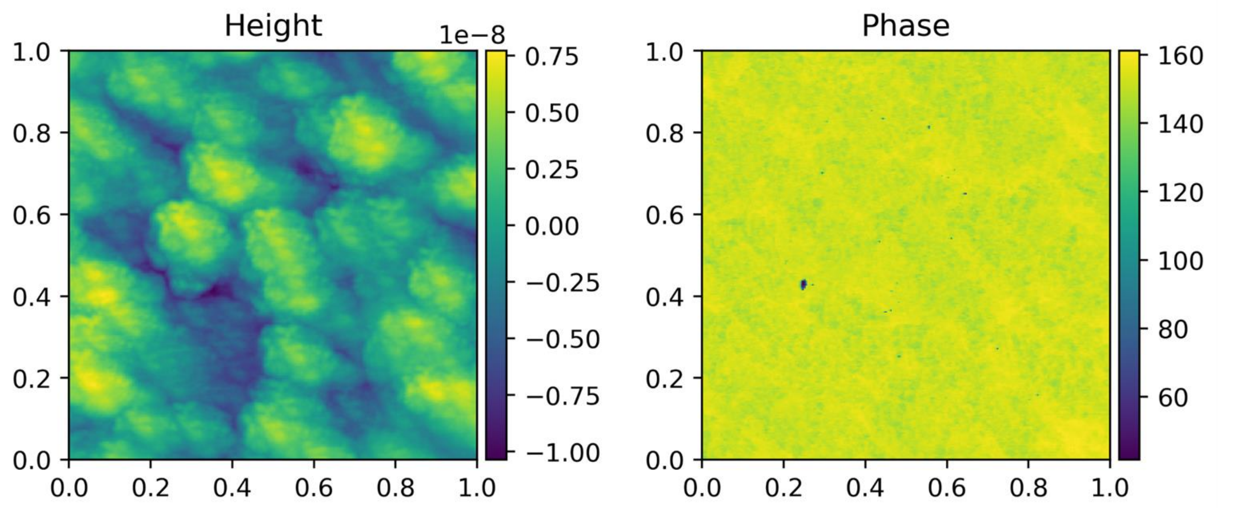
Secret sauce – reward function!

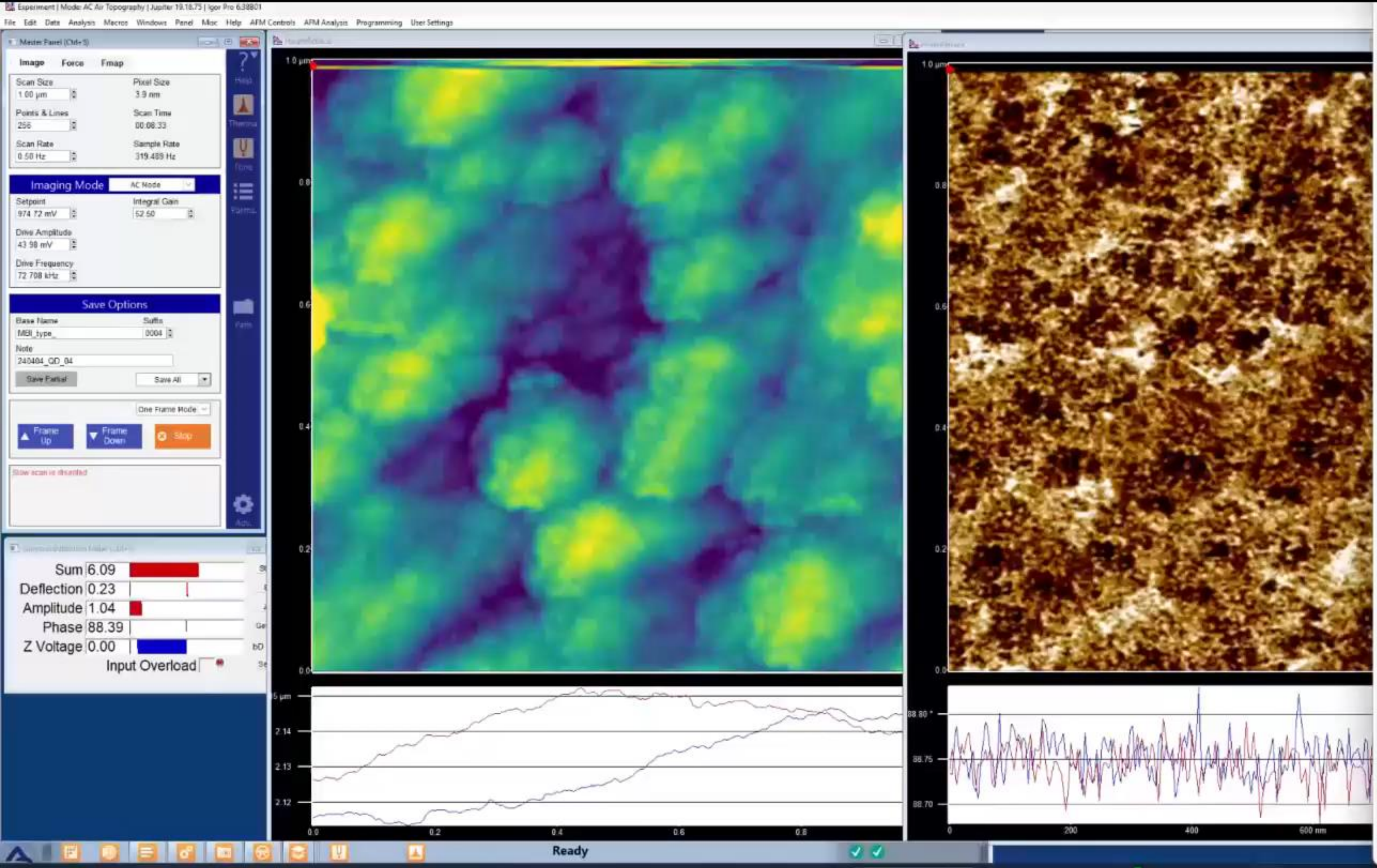
Reward functions in imaging



- BO works for any probe or sample – as long as the reward function catches main features of a good scan, BO will find the parameters to achieve that

- We're working on safe seeding now – so the probe is not damaged in the random seeding in the parameter space





BO functions

+3 cells hidden

Run the BO

[97]:

```python
# Make the grid for exploration
save_name = 'AR_Height/240404_QD_04_'

x = []

# v_ac
x1_min = 0.3e-3
x1_max = 0.05

# setpoint
x2_min = 0.1
x2_max = 0.9

factor = 248 / 10.07

offset = read_igor(key=['FreeAirPhase'], connection=connection)[0] - 90
print(offset)

x1 = np.linspace(x1_min, x1_max, num=100)
x2 = np.linspace(x2_min, x2_max, num=100)

d1, d2 = len(x1), len(x2)

for i in range(len(x1)):
    for j in range(len(x2)):
        x.append([x1[i], x2[j]])

x = np.asarray(x, dtype=np.float32)
X = torch.from_numpy(x)
```

```
-0.9607010000000002
```

[98]:

```python
X_measured, X_unmeasured, y_measured, global_min = generate_seed(num=10, factor=factor,
```

Python notebook is running on supercomputer -- ISSACs at UTK (x3 speed).

Reward functions in imaging

Imaging Optimization

Physical laws discovery

Image-based reward functions

- Human selected objects (DCNNs)
- Equal sampling of feature space
- Equal sampling of parameter space (combi library)
- Reward driven segmentation for atomic systems and beyond

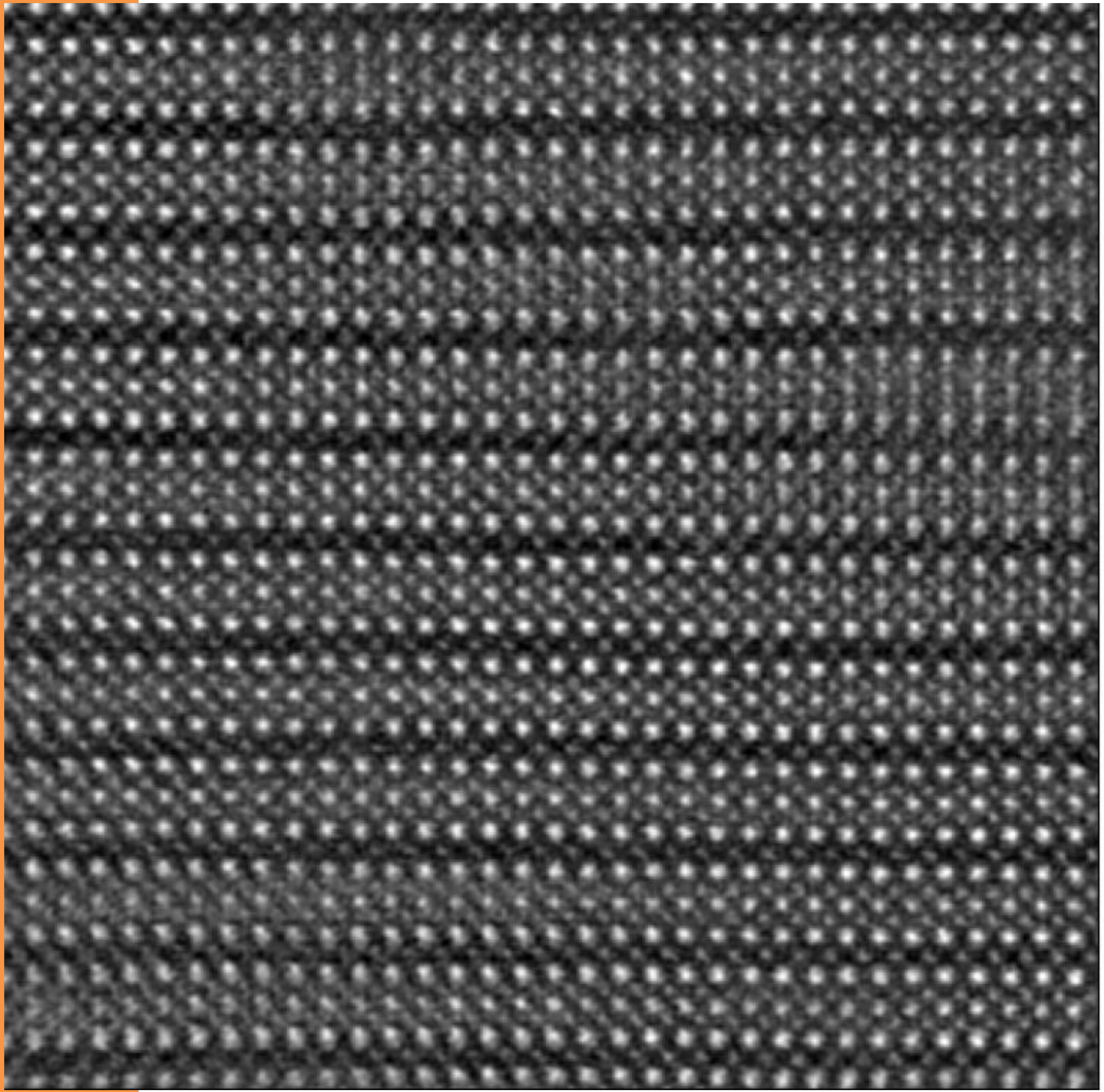
Structure property relationship discovery

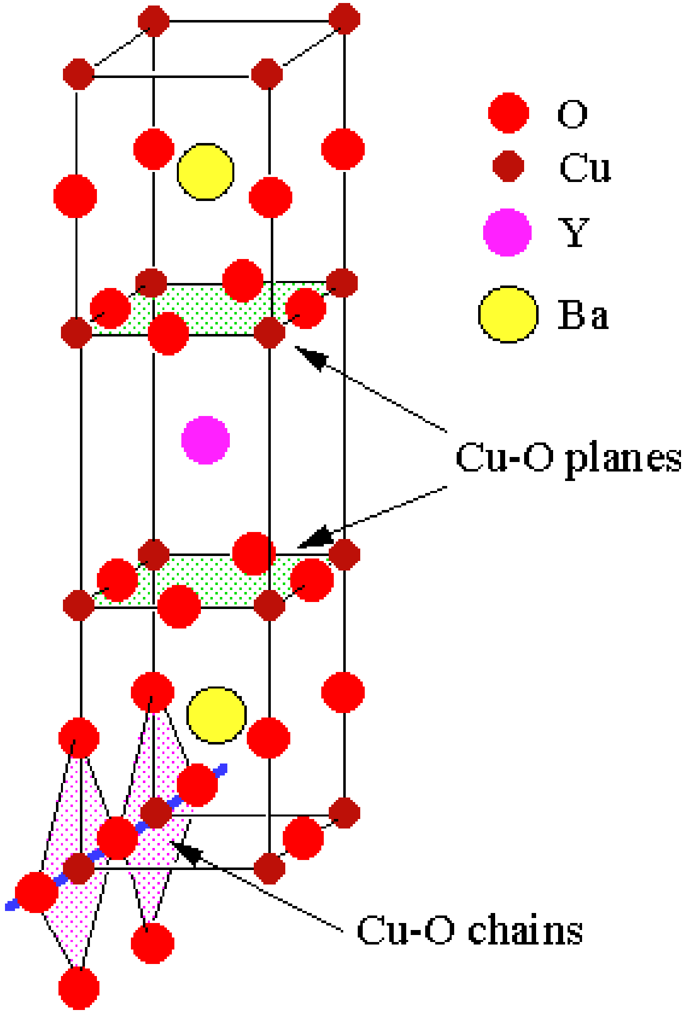
- Reward definition (with cost)
- Tuning curiosity
- Human in the loop DKL

Co-orchestration multiple tools

Image-based reward functions

… are defined only in the context of the subsequent experiments

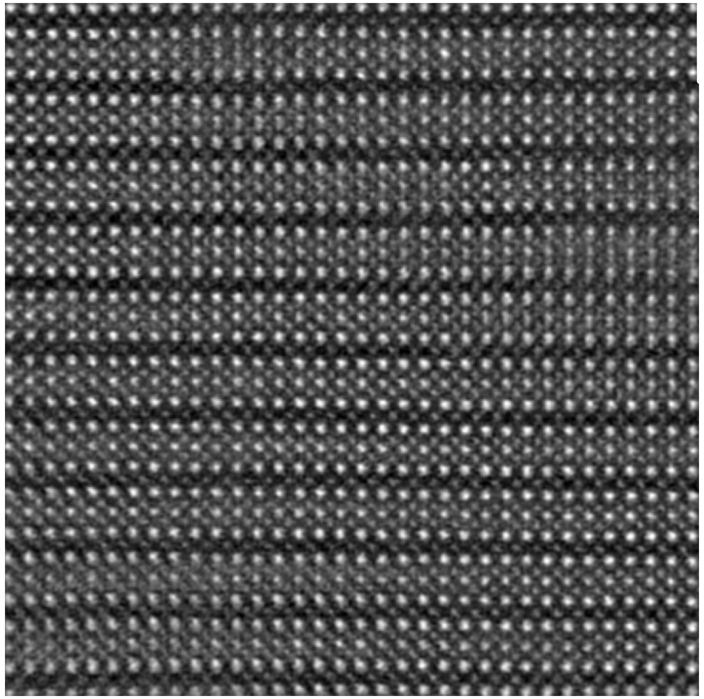


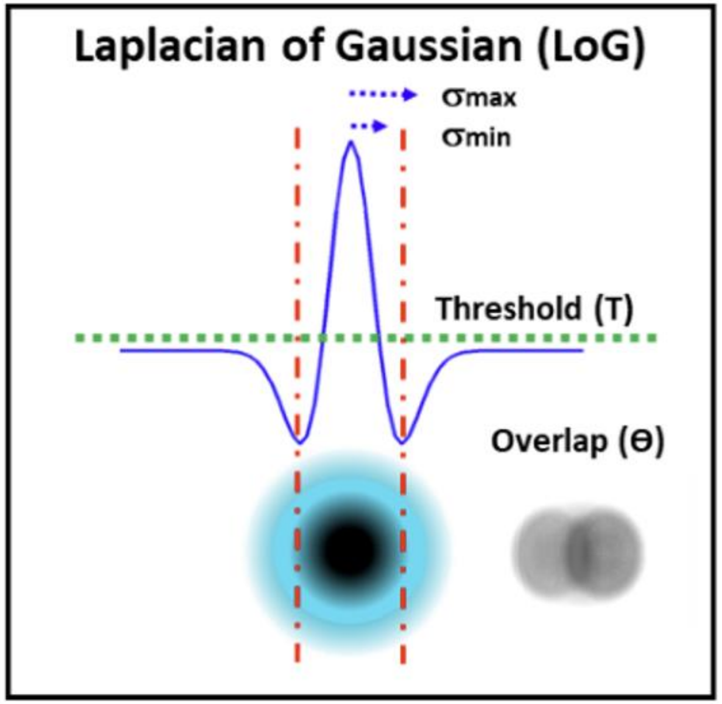


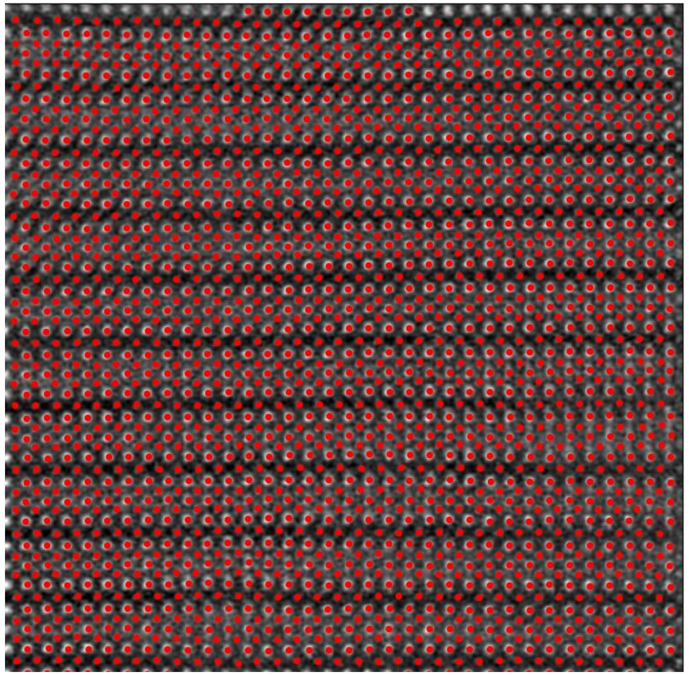
- Radiation damage on oxygen sublattice
- Formation of antiphase defects
- Bending of Cu-O layers

Data by A. Goyal and H. Zhang

From LoG to LoG*







$YBa_2Cu_3O_{7\text{-}\delta}$ thick film, doped with Dy_2O_3 nanoparticles

Zhang, Y.; Rupich, M.; Solovyov, V.; Li, Q.; Goyal, A. Dynamic behavior of reversible oxygen migration in irradiated-annealed high temperature superconducting wires. *Scientific Reports* **2020**, *10* (1), 14848.

σ_{min} = (0.1, 3.0),

σ_{max} = (1.0, 6.0)

Threshold = (0.01, 0.4)

Overlap = (0.01, 0.5)

Physics-based Rewards

Quality Function: Objective 1

Quality Count (QC) defined as the normalized difference between the number of atoms found by LoG and the number of atoms found defined by Oracle:

$$QC = \frac{LoG\ blobs - Oracle}{Oracle}$$

Error Function: Objective 2

Any atom with a cumulative interaction value less than DS (summation of distances) with its four nearest neighbors will be regarded as not having physical significance, and thus, categorized as an error within this context.

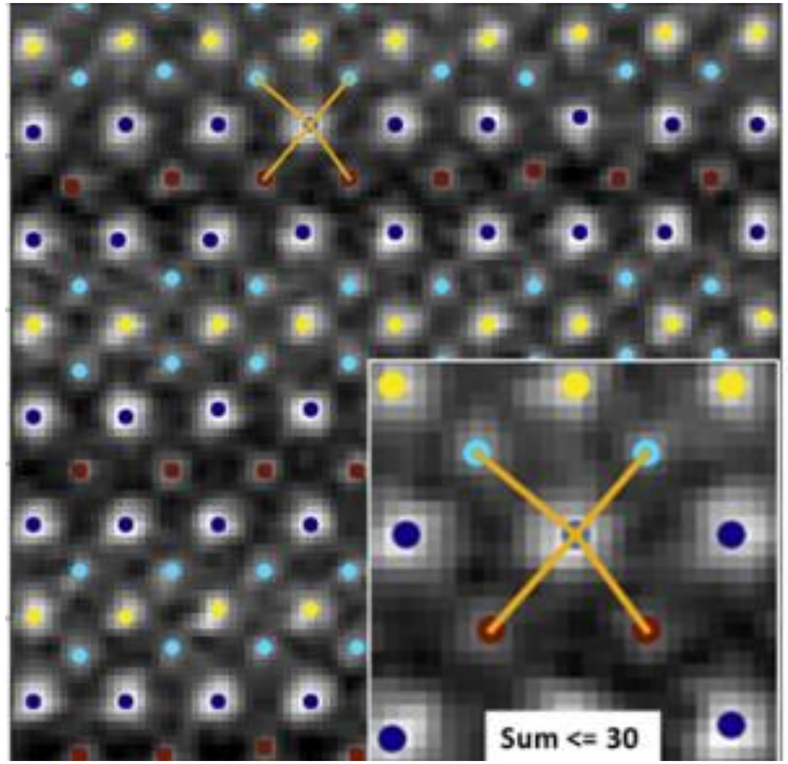
$$ER = \frac{\#\ atoms\ with\ \text{cumulative interaction value less than}\ DS}{Oracle}$$

Oracle can be:

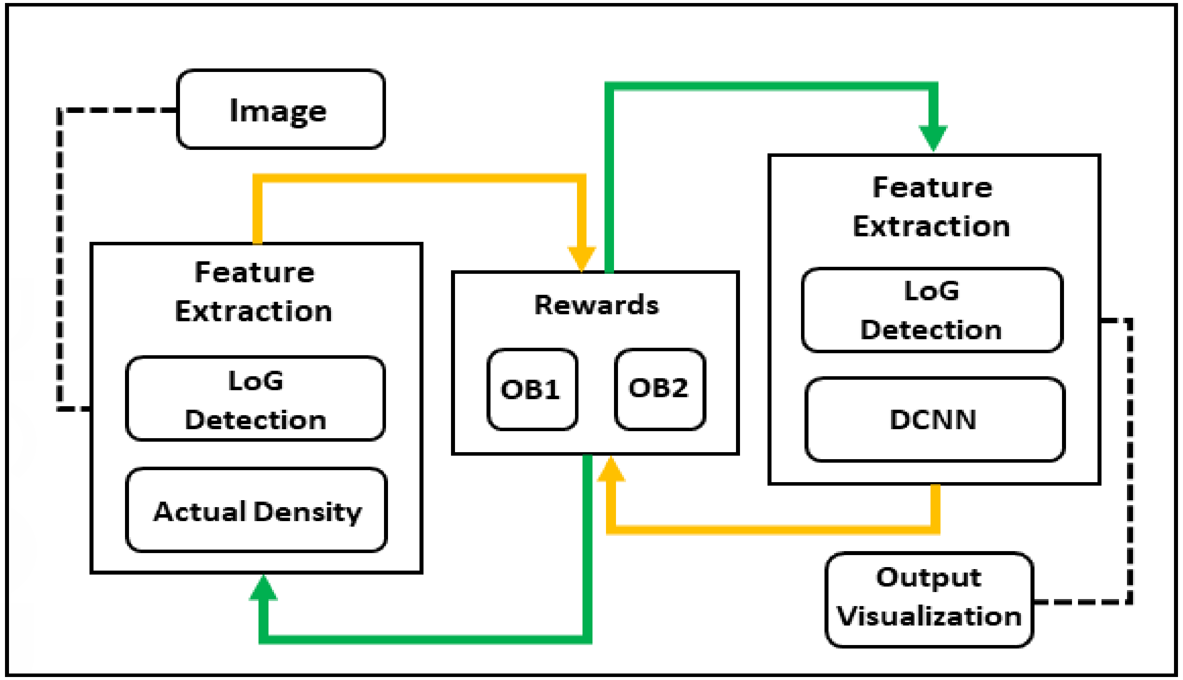
- Number of atoms via **physical criterion** – i.e. number of atoms in the structure with respect to stoichiometry

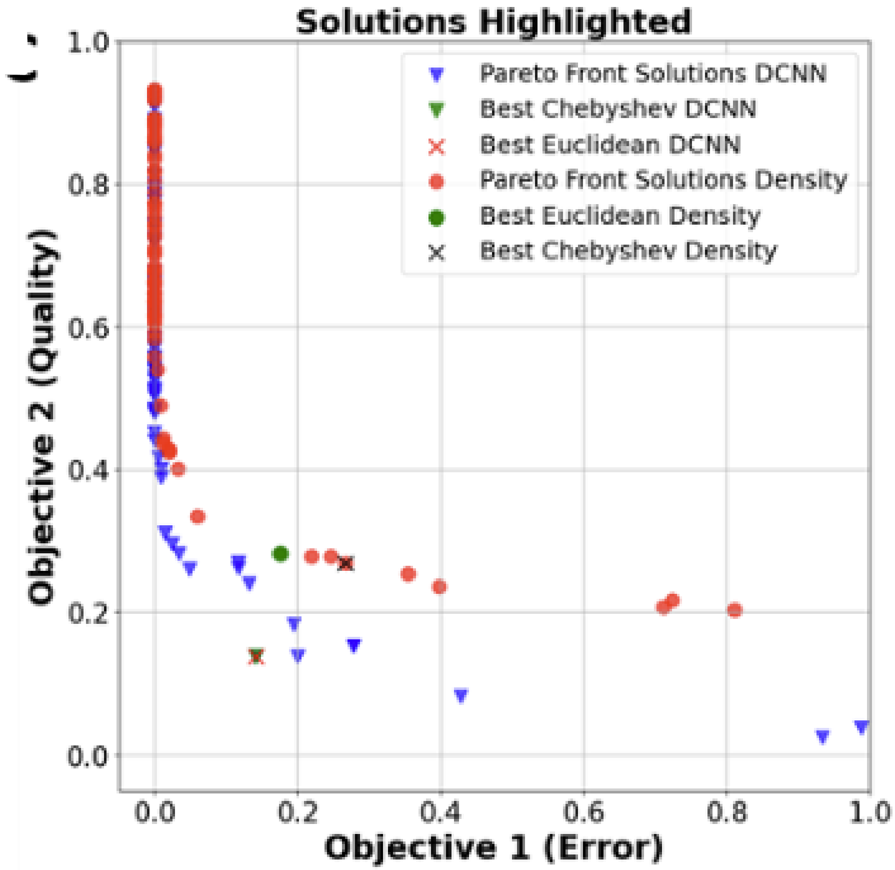
Benchmark:

- Number of atoms via **DCNN**



It’s optimization problem!

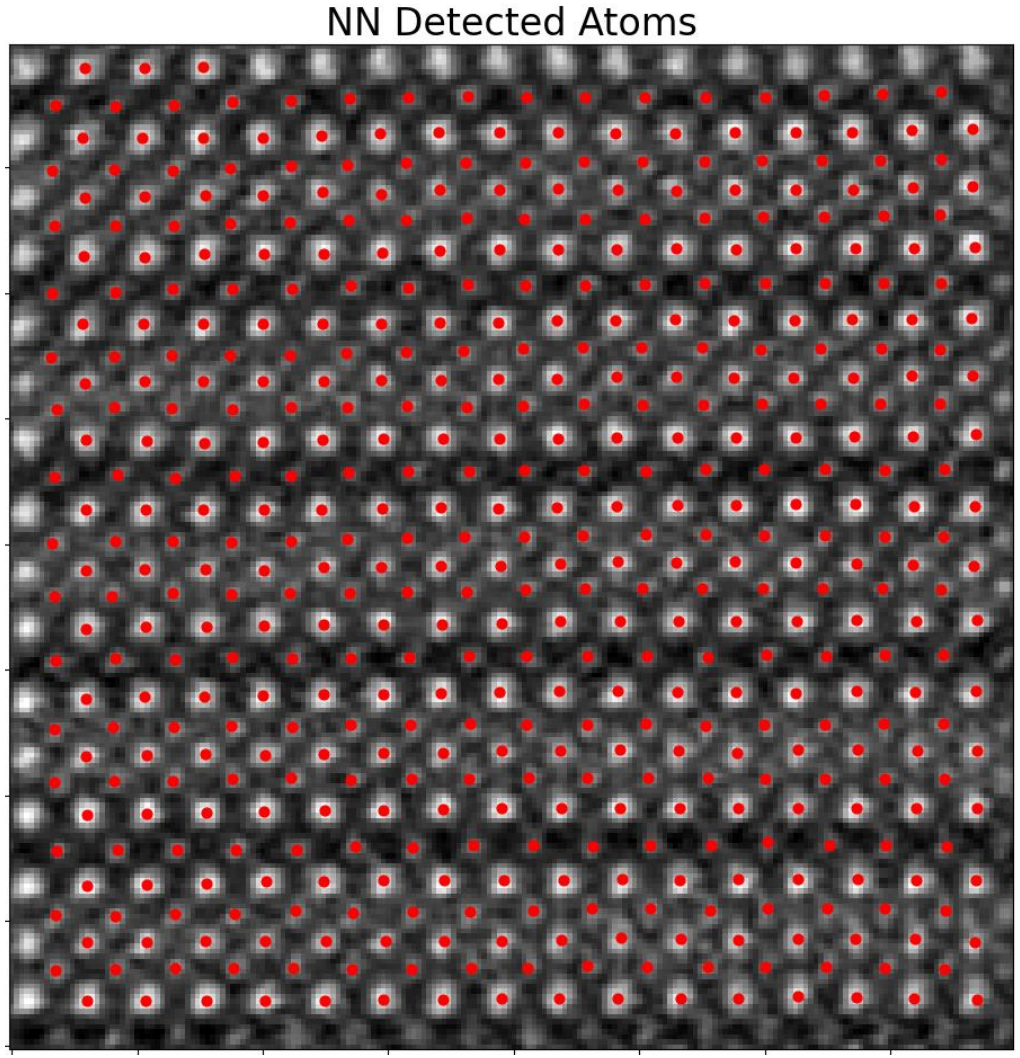


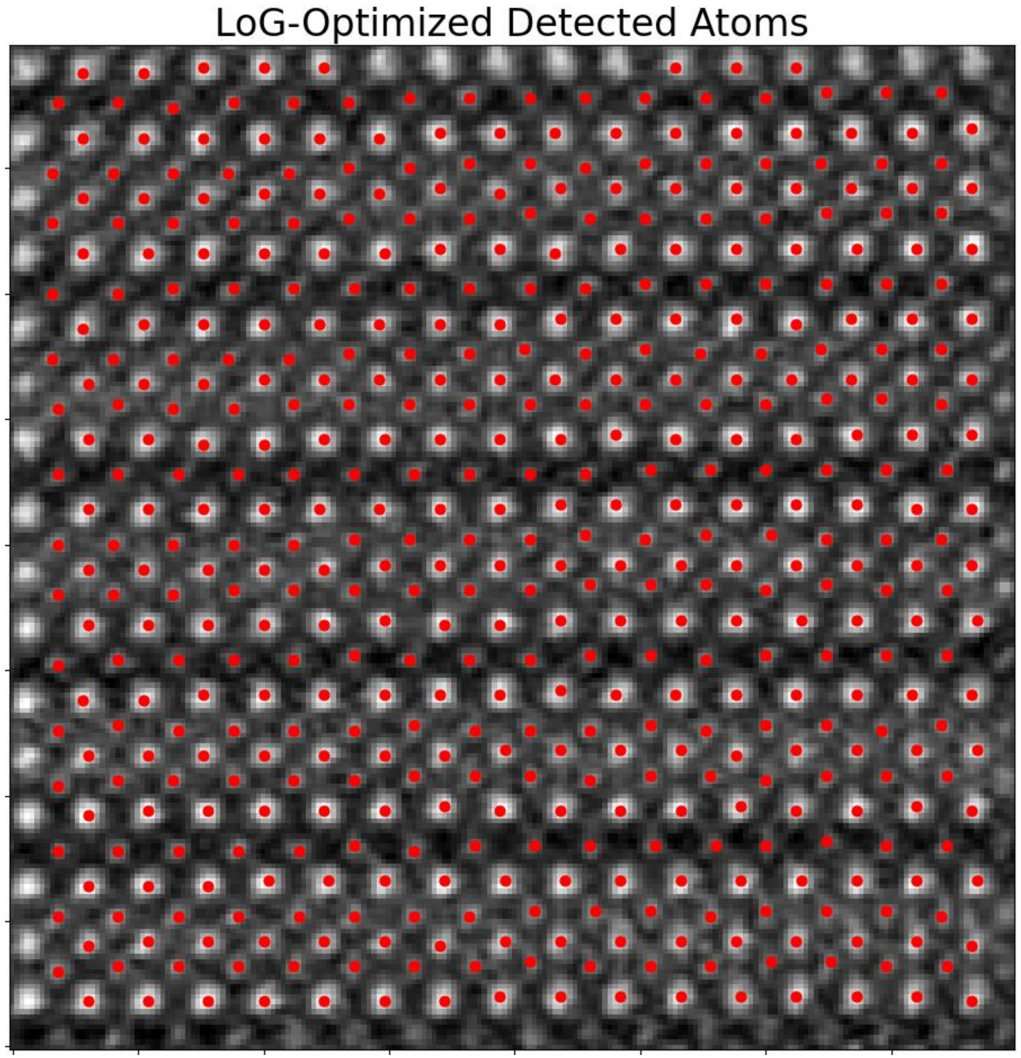


Both side of the reward should be minimized

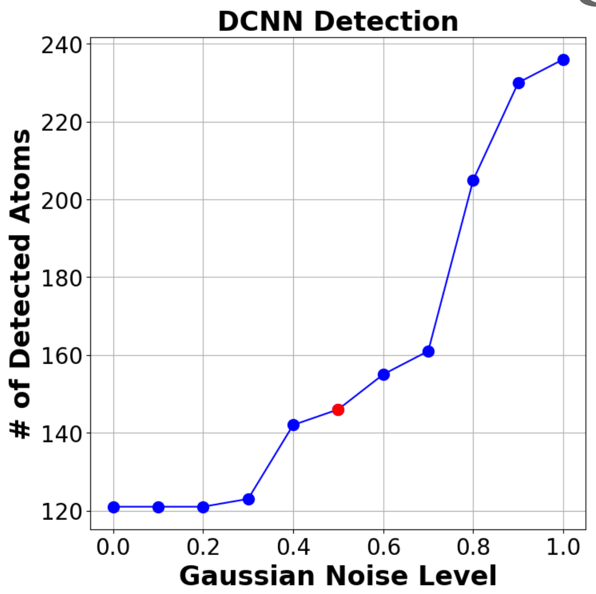
Quality Function: Objective 1
Error Function: Objective 2

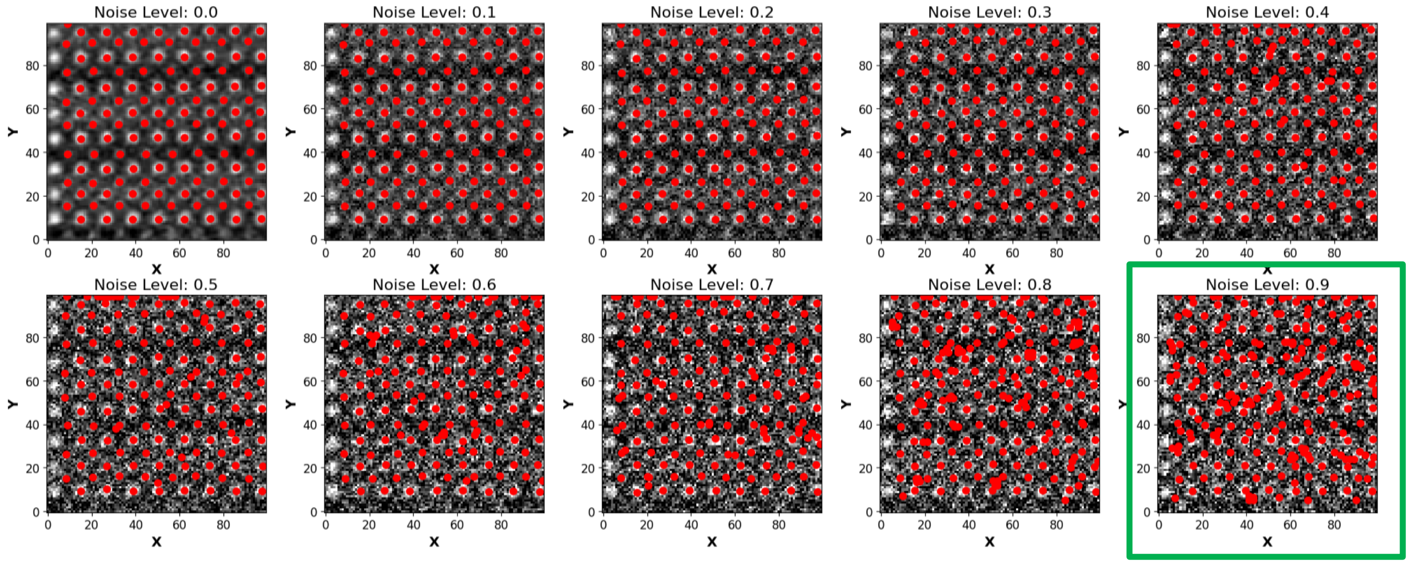
Benchmarking LoG* vs. DCNN

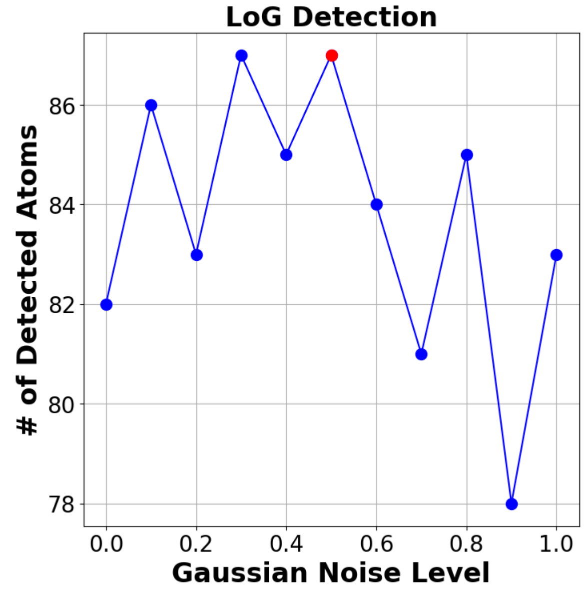


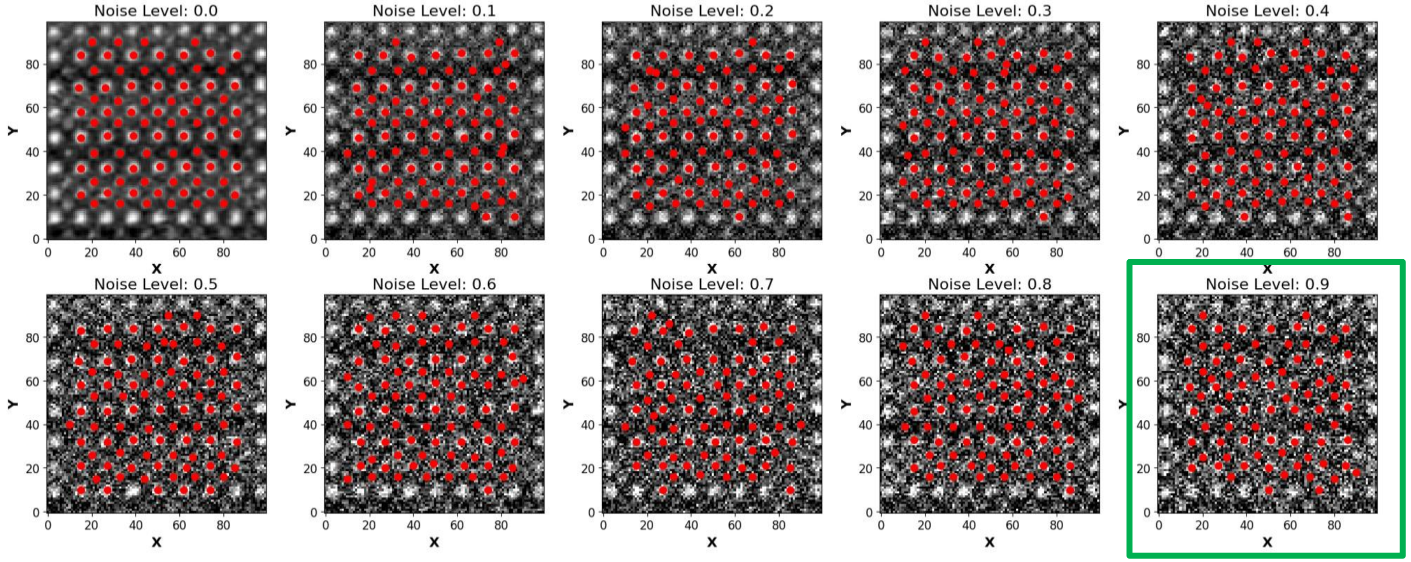


Benchmarking LoG* vs. DCNN



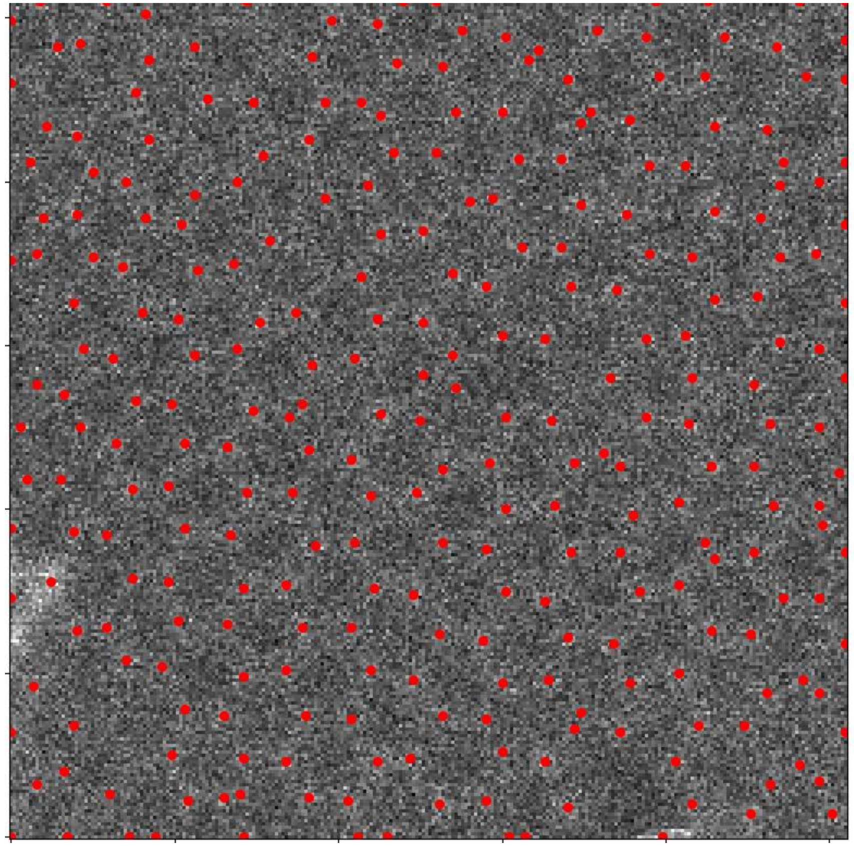


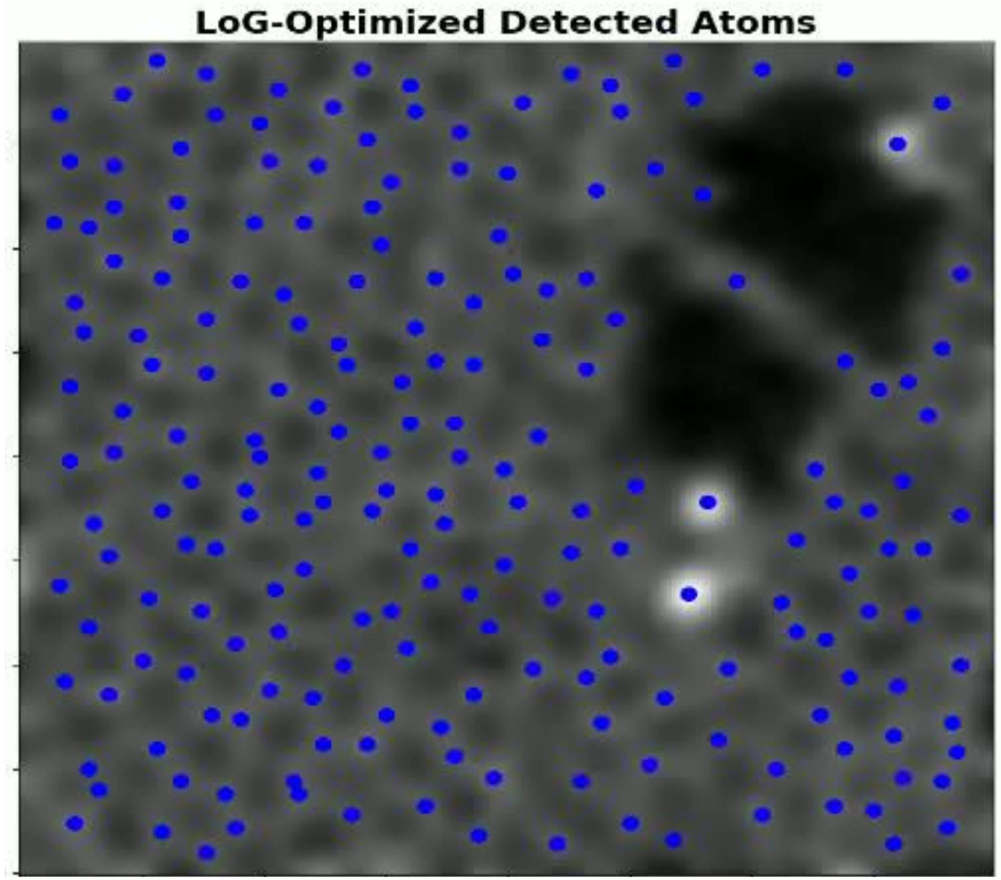




In contrast to the DCNN, the LoG detection exhibits a much lower variability in the number of detected atoms across noise levels, maintaining a relatively consistent count.

LoG* for Dynamic Image Analysis





Reward function approach reduces the supervised learning problem to optimization problem
(human labeling, sensitive to distribution shift, black box) -> (unsupervised, robust, explainable)

Example of analysis pipeline

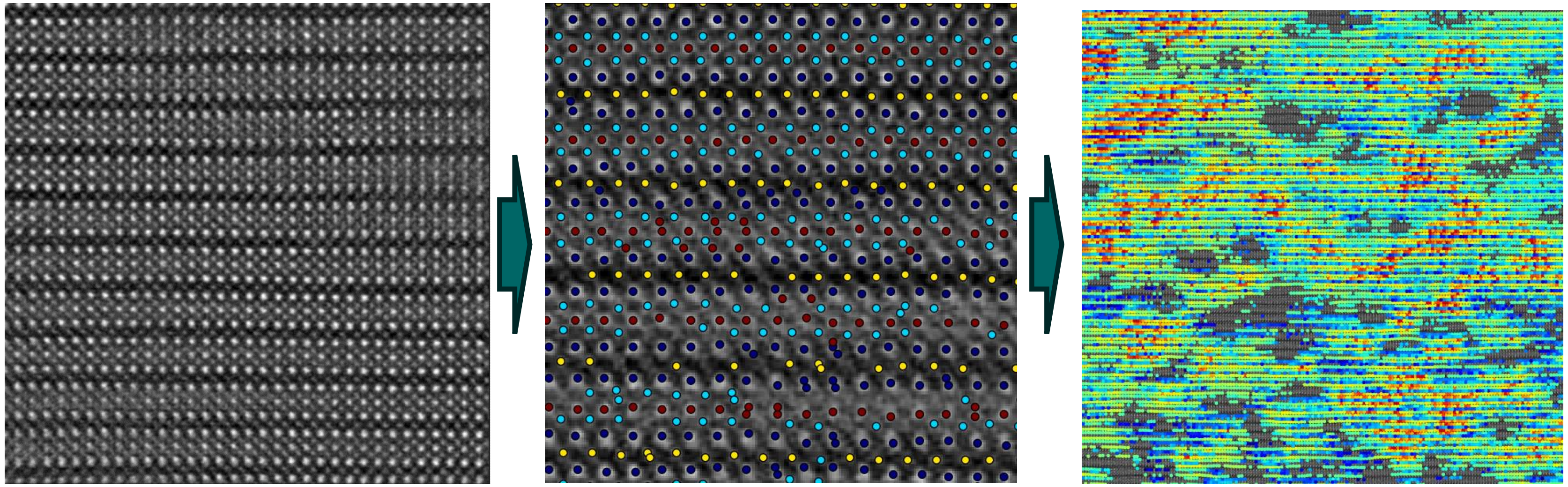
Image or video

Creating descriptors
(size, shape)

Identity transform
Radon transform
Fourier transform

Dimensionality reduction of sub-images
GMM
PCA
NMF

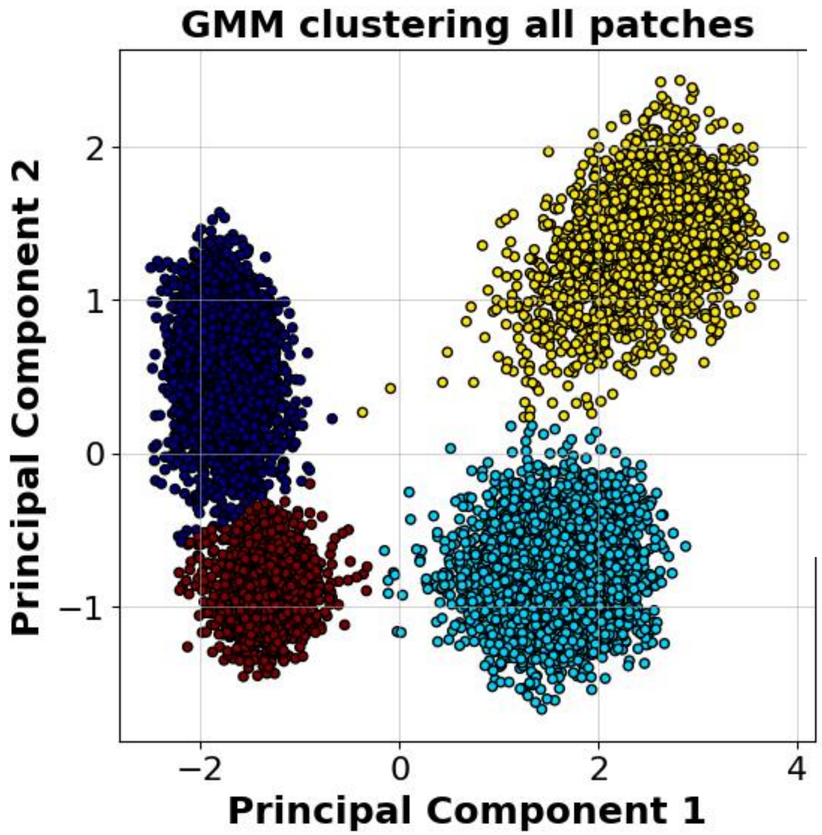
Visualization and descriptive statistics:
Loading maps and components

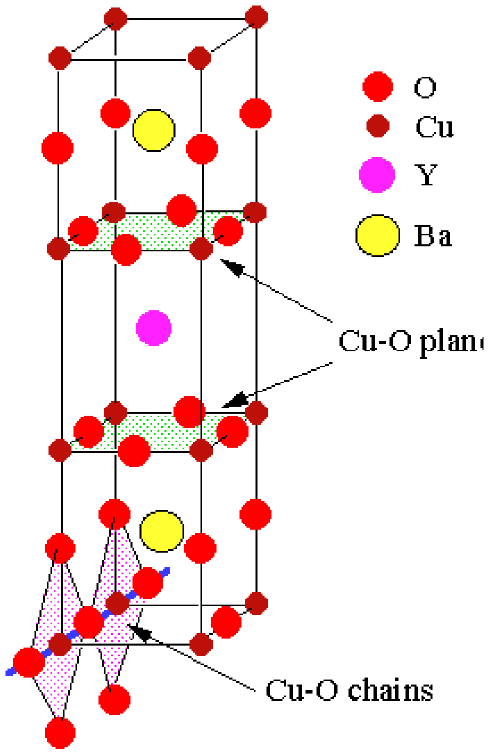


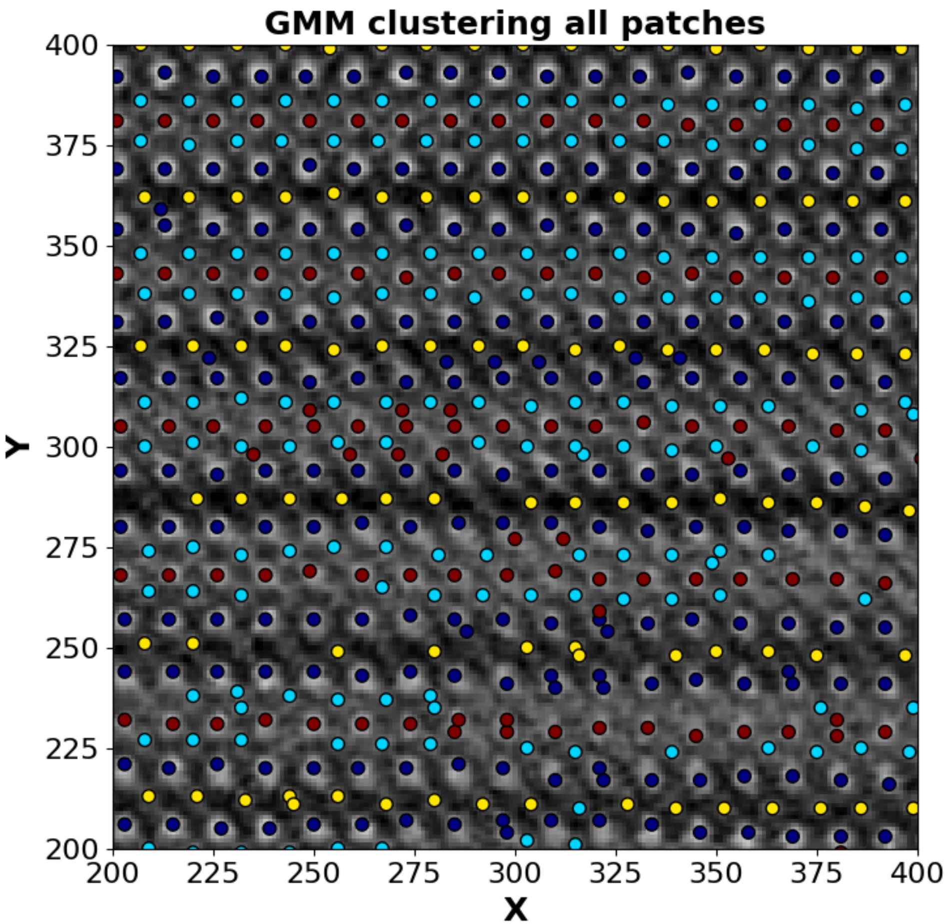
Object discovery
- Large number of iterative steps
- Workflows with constant human oversight
- Non-myopic reward

Discovering damage

1: Perform GMM clustering on all patches (LoG-optimized)



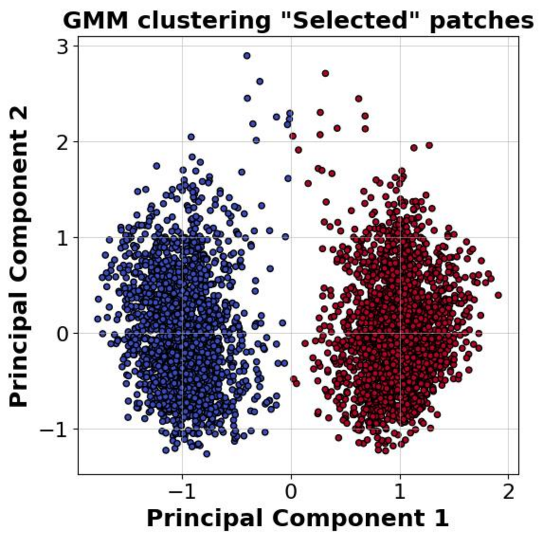


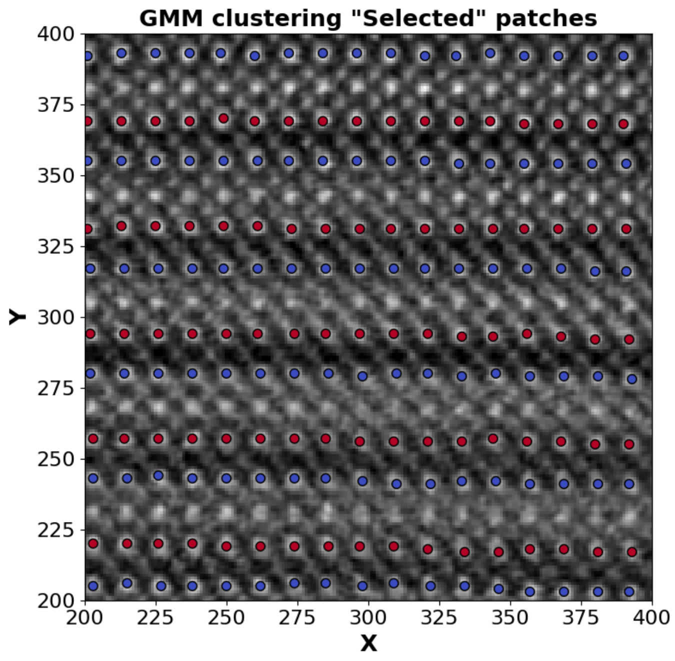


Well-defined clustered scattered on the image
CuO plane : yellow
CuO chain : cyan
Ba: Dark blue
Y: Red

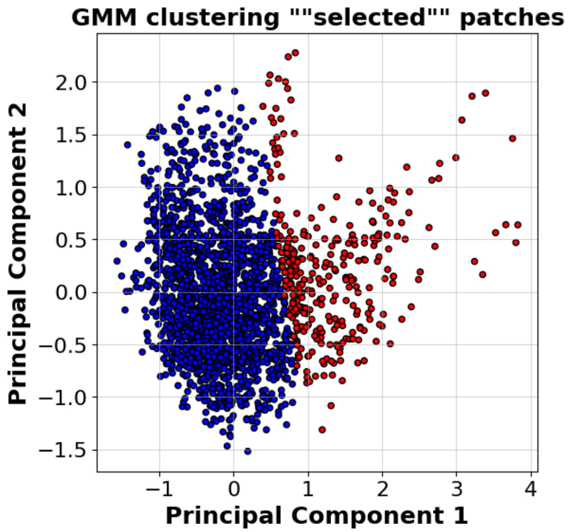
Discovering damage

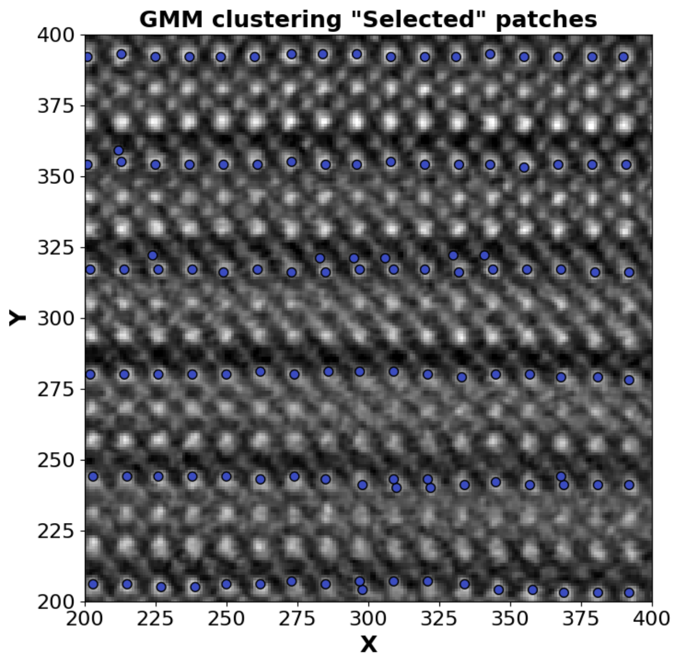
2: Perform GMM clustering on patches (LoG-optimized) **centered on all Ba atoms**





3: Perform GMM clustering on patches (LoG-optimized) **centered on one type of Ba atoms**





Here we are seeking the accurate values of Threshold and covariance type to observe if there is a variation in just one type of Ba?

Discovering damage

Threshold = (0.0, 1.0)

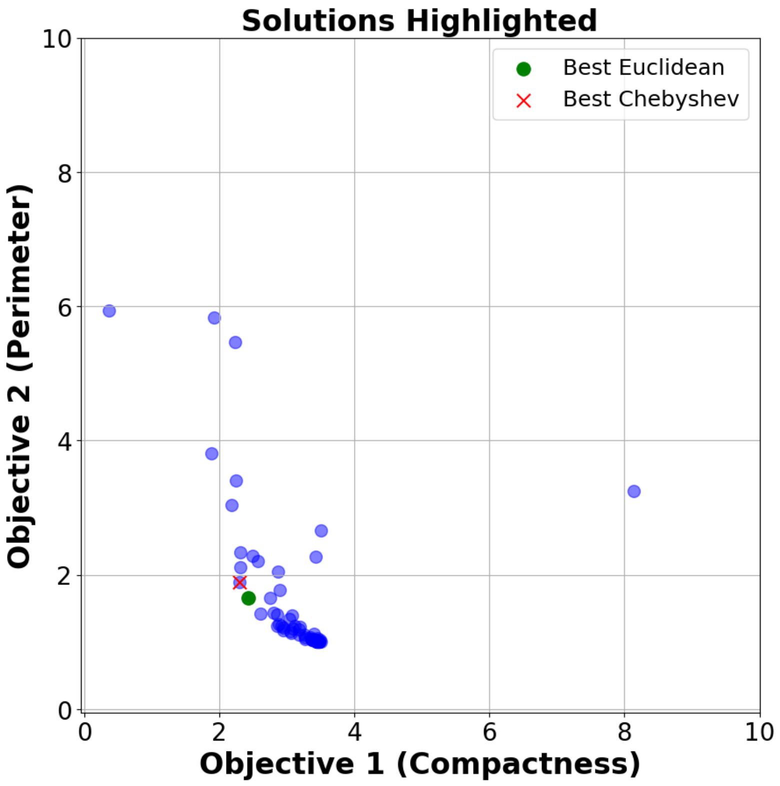
Covariance Type: tied, full, diag, spherical

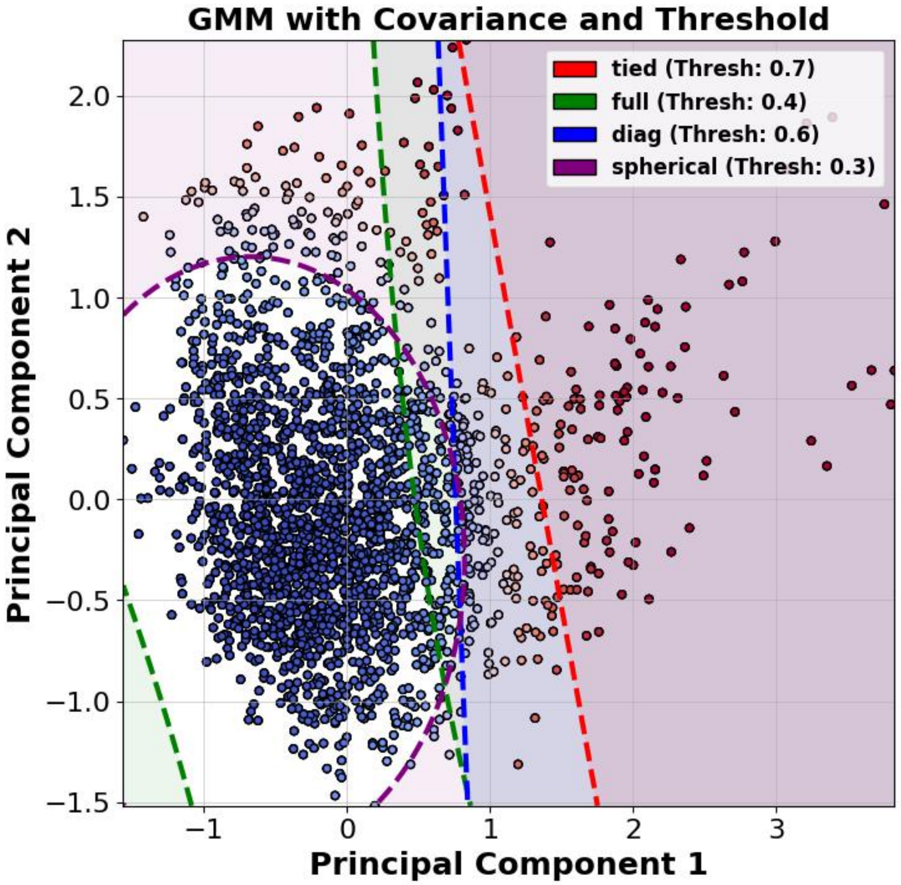
GMM clusters based on only one type of Ba atoms, introducing some variety, which it can be differentiated by different values of threshold and covariance type in GMM clustering

Both side of the reward should be minimized

Objective 1: **Compactness** of the amorphous area

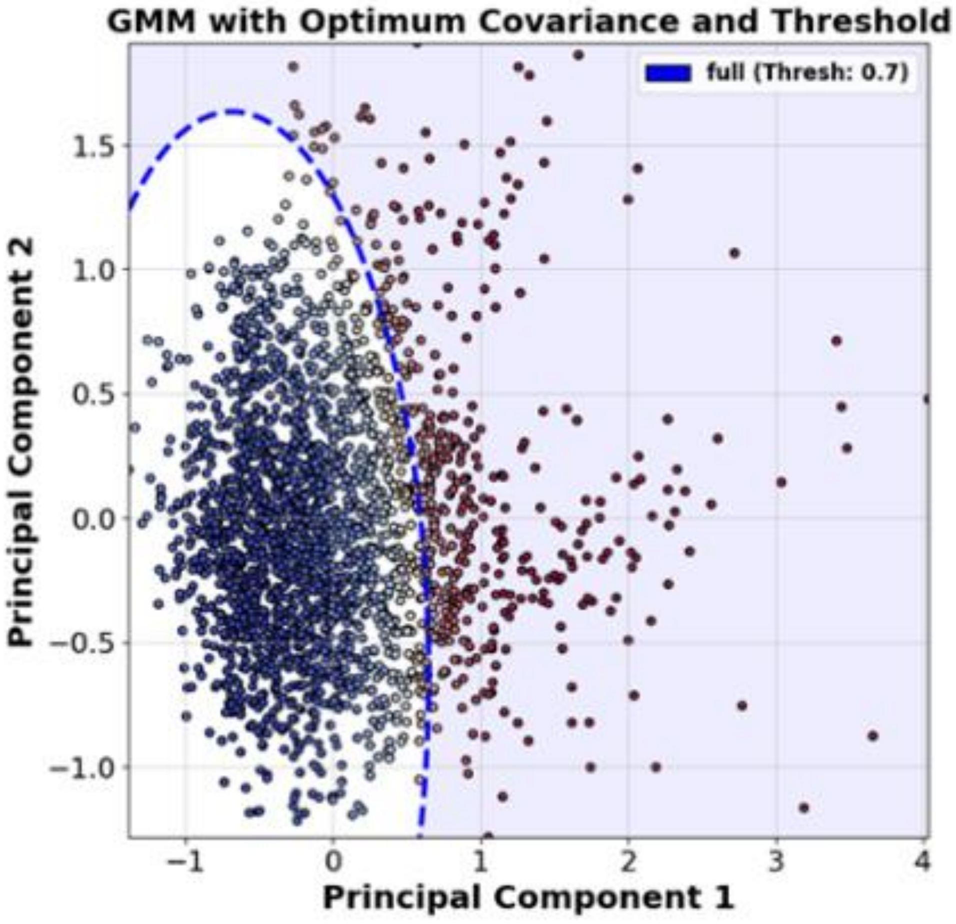
Objective 2: **Perimeter** of amorphous area





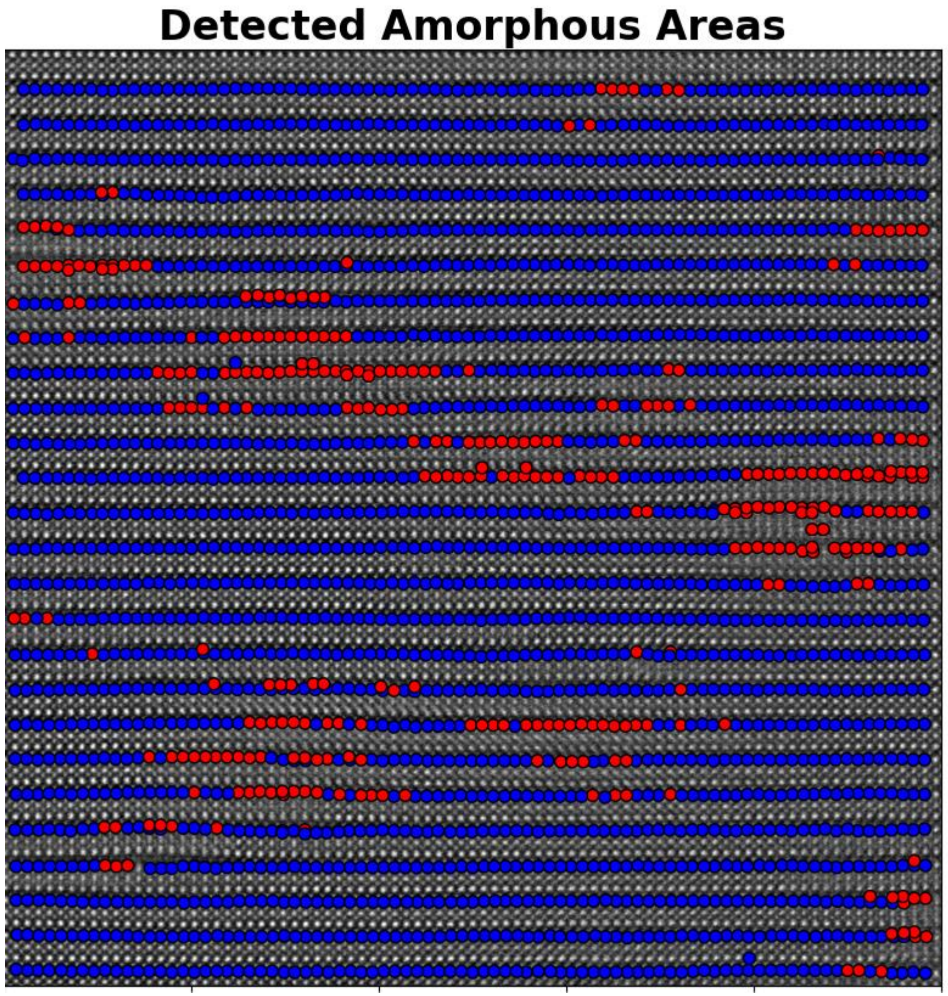
Discovering damage

Objective 1: Compactness of the amorphous area
Objective 2: Perimeter of amorphous area



Optimal threshold and covariance type achieved by MOBO for GMM clustering,

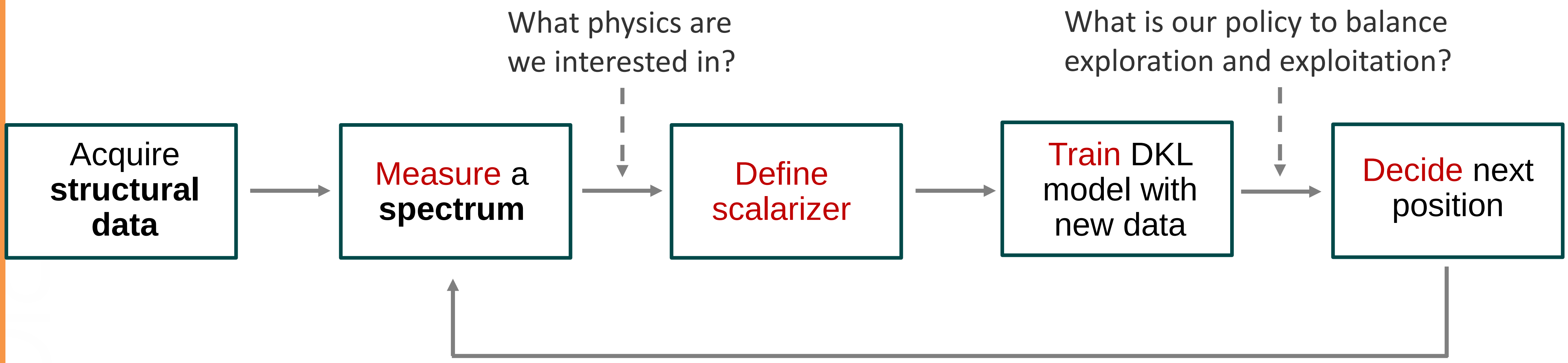
1000*1000 pixels



Uncovered amorphous areas in the substrate

Automated Microscopy with Deep Kernel Learning

Deep Kernel Learning based BO



Key concepts:

- **Scalarizer:** (any) function that transforms spectrum into measure of interest. Can be integration over interval, parameters of a peak fit, ration of peaks, or more complex analysis

- **Experimental trace:** collection of image patches and associated spectra acquired during experiment. Note that we collect spectra, not only scalarizers