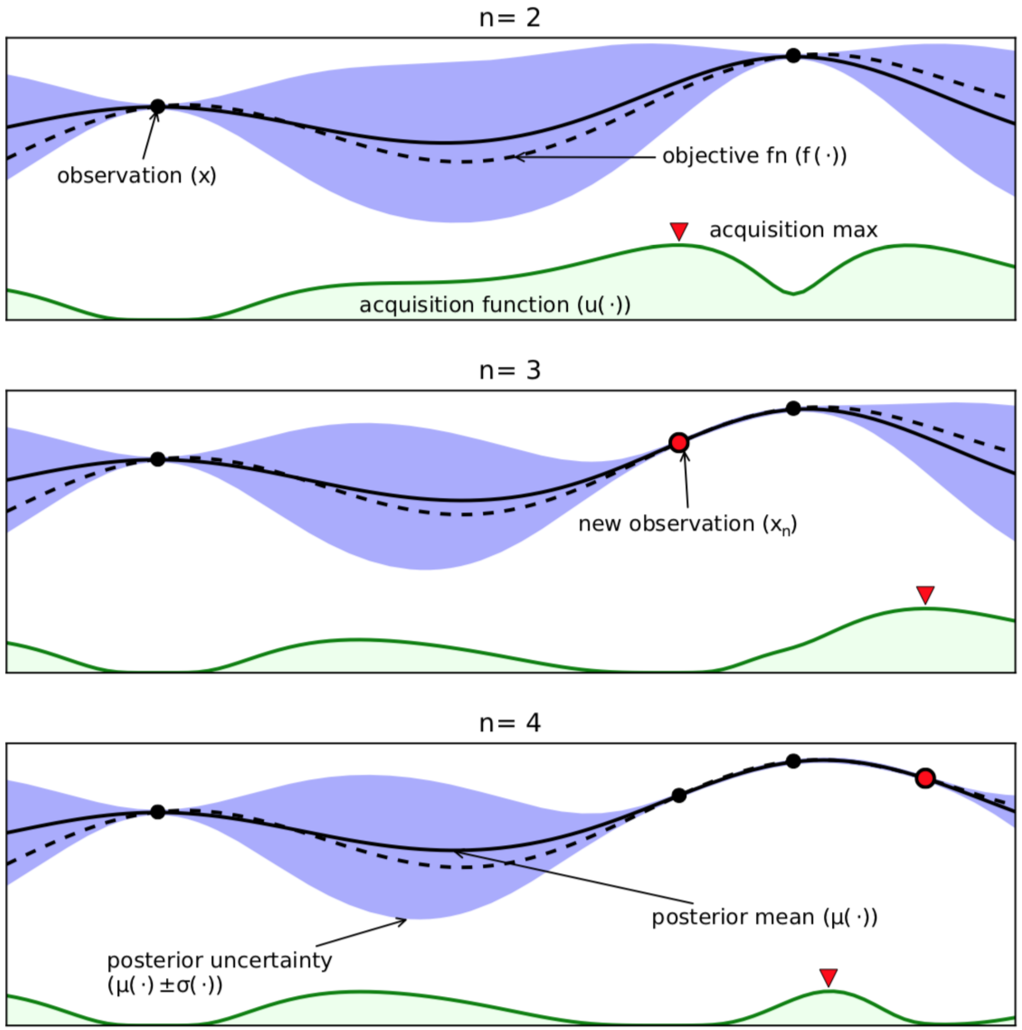
THE UNIVERSITY of TENNESSEE KNOXVILLE

Deep Kernel Learning – II: Rewards, Explainability, and Human in the loop Intervention

Sergei V. Kalinin

Bayesian Optimization



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

Obtain/update training data *D*

↓

Train surrogate model
$X \rightarrow y$

↓

Make prediction
$X_* \rightarrow \bar{f}_*, \mathbb{V}[f_*]$

↓

Acquisition function
$acq(\bar{f}_*, \mathbb{V}[f_*])$

Evaluate new point
$X_*^i = \text{argmax}(acq)$

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

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)

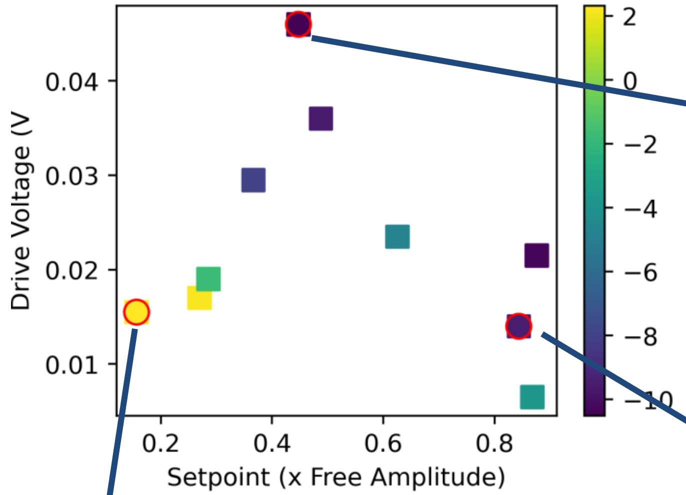
Structure property relationship discovery
- Reward definition (with cost)
- Tuning curiosity
- Human in the loop DKL

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

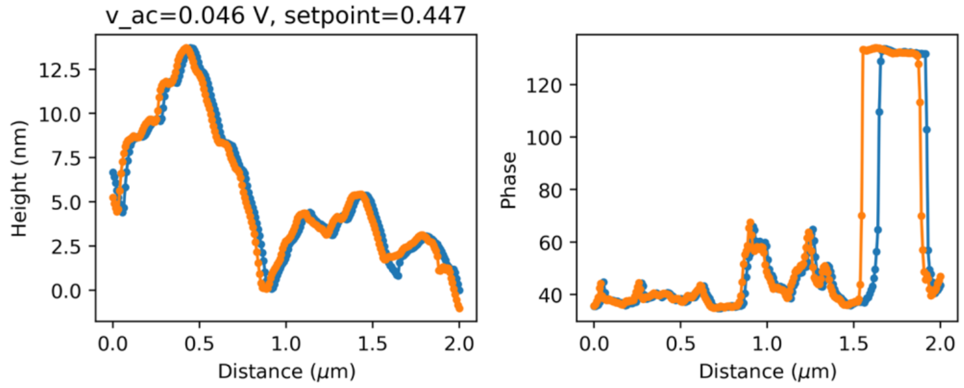
Reward functions in imaging

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

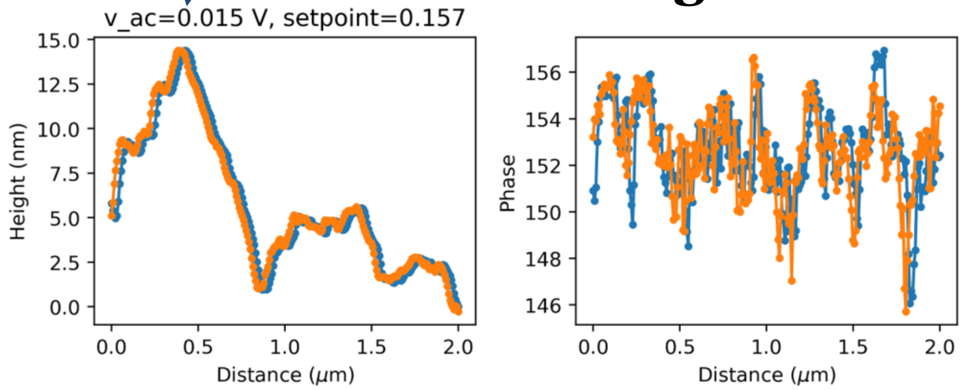
Reward of 10 seeding measurements



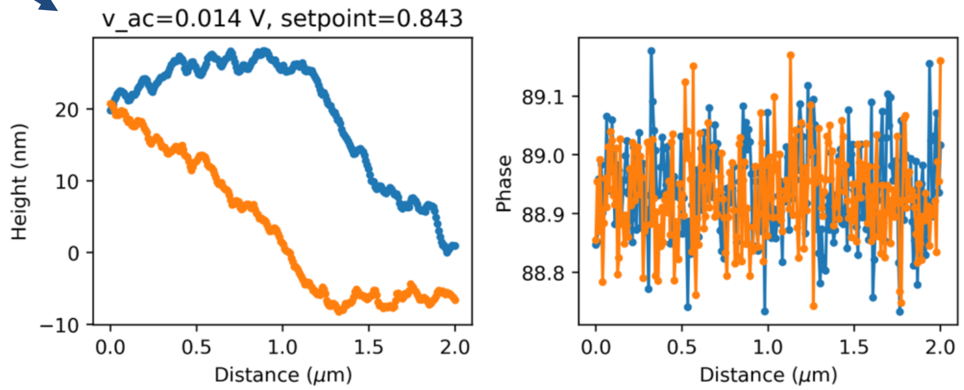
Pushing too hard



Best in seeding

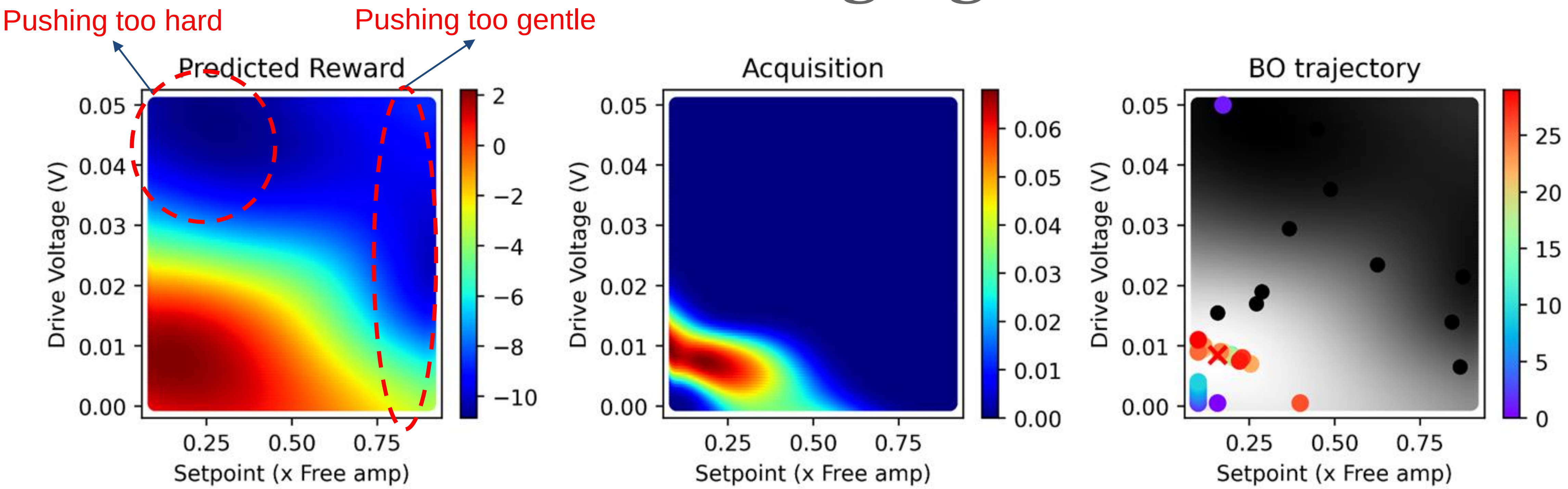


Pushing too gently



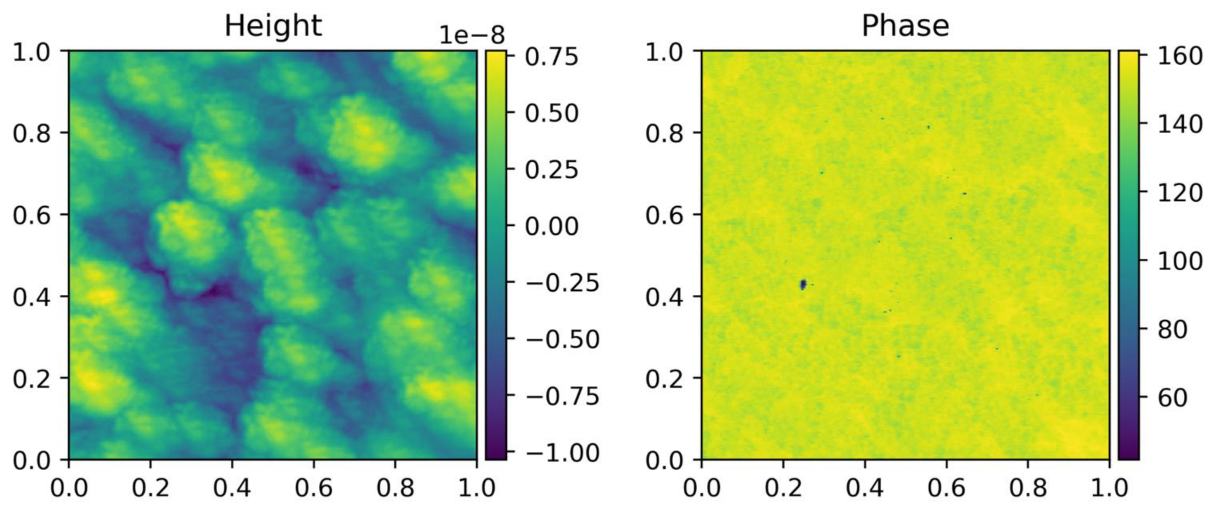
Secret sauce – reward function!

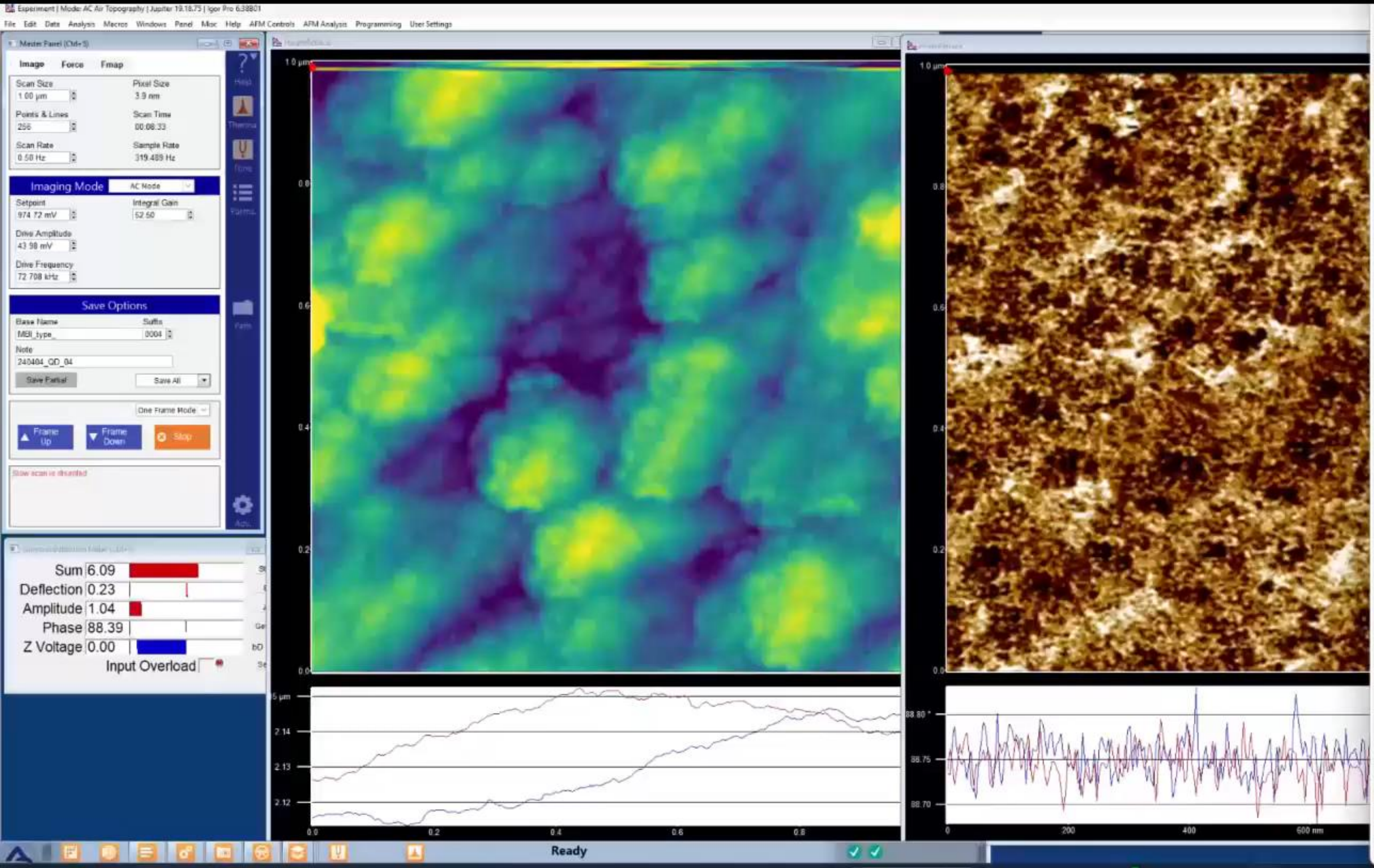
Reward functions in imaging



- 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)

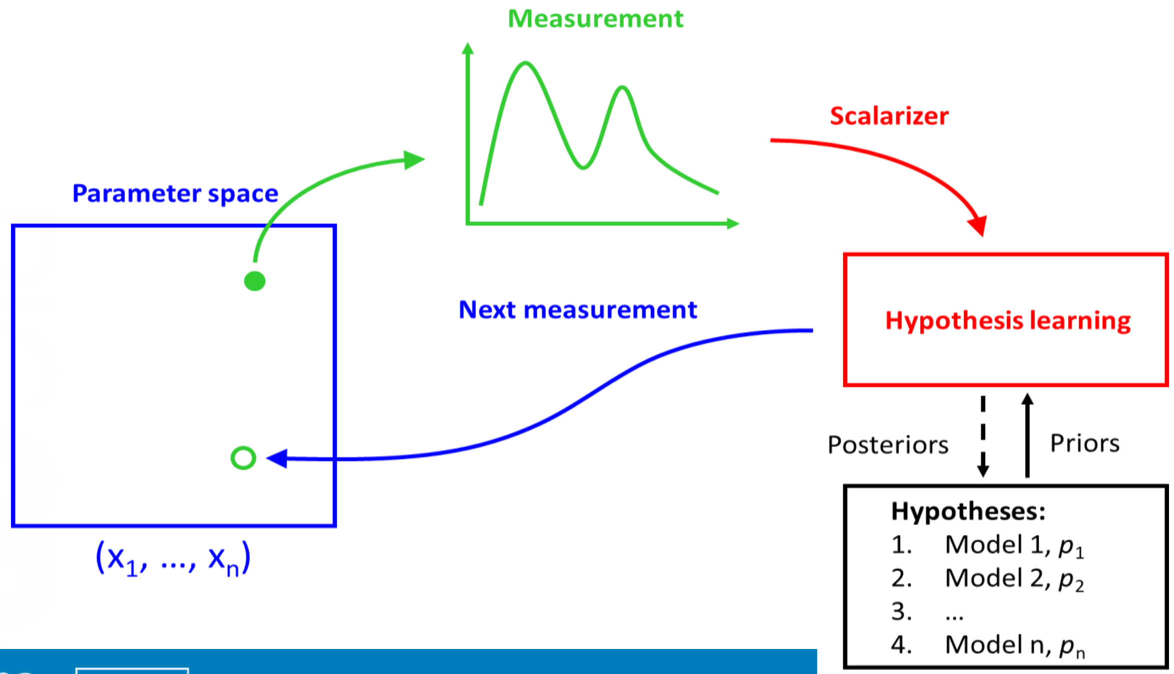
Structure property relationship discovery

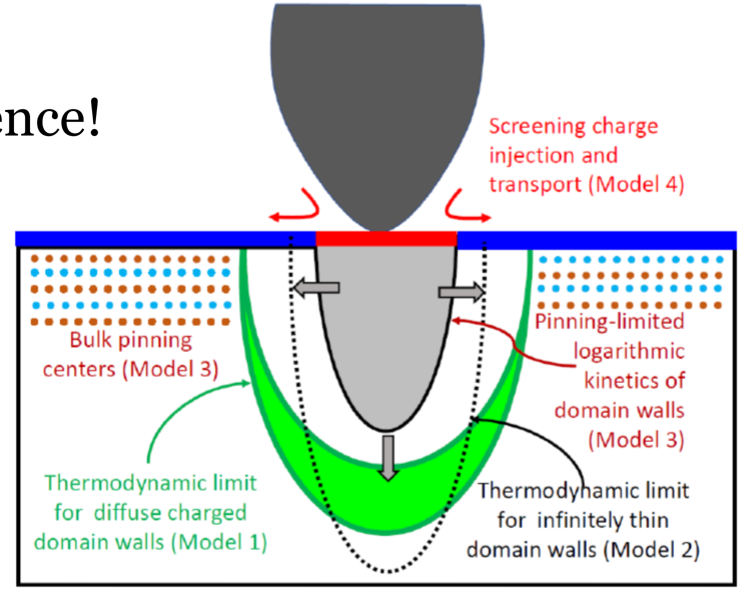
- Reward definition (with cost)
- Tuning curiosity
- Human in the loop DKL

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

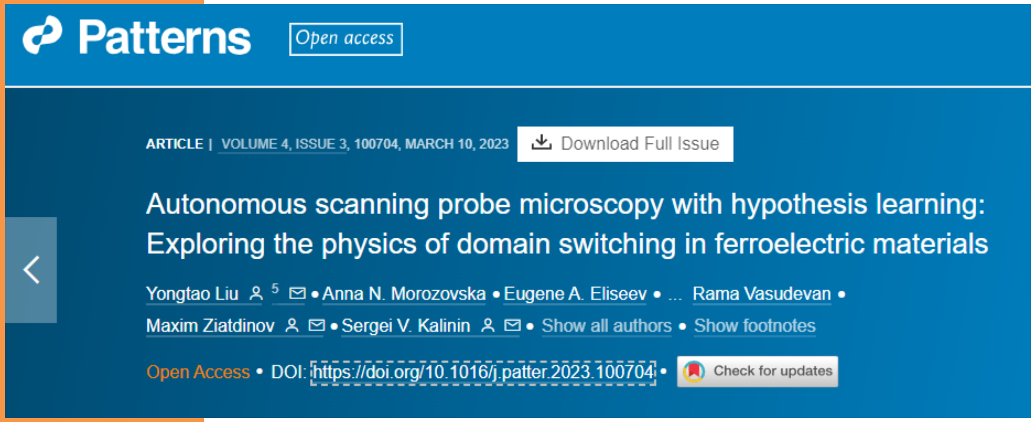
Hypothesis Learning

- 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

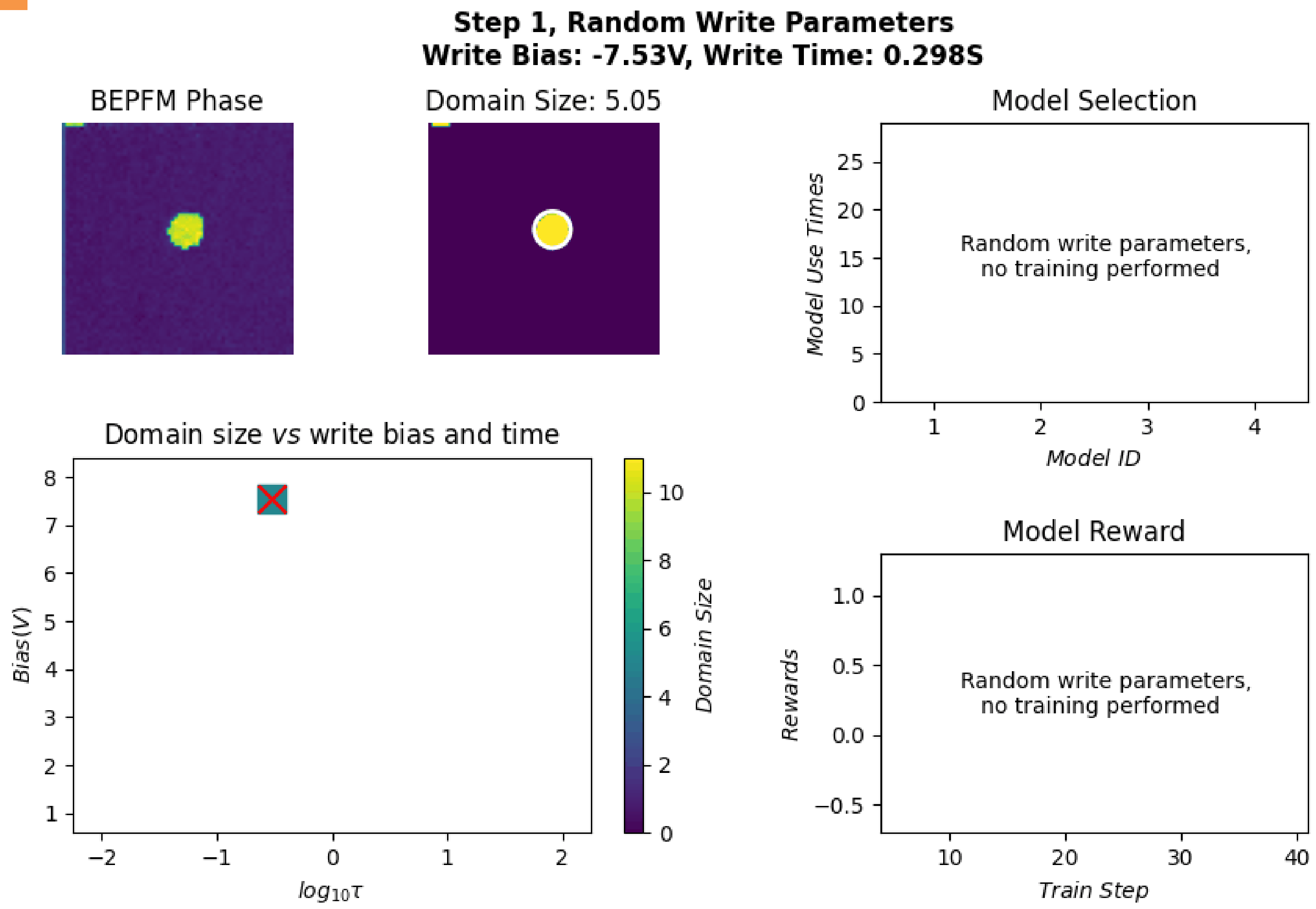
ARTICLE | VOLUME 4, ISSUE 3, 100704, MARCH 10, 2023 Download Full Issue

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

Hypothesis learning in action

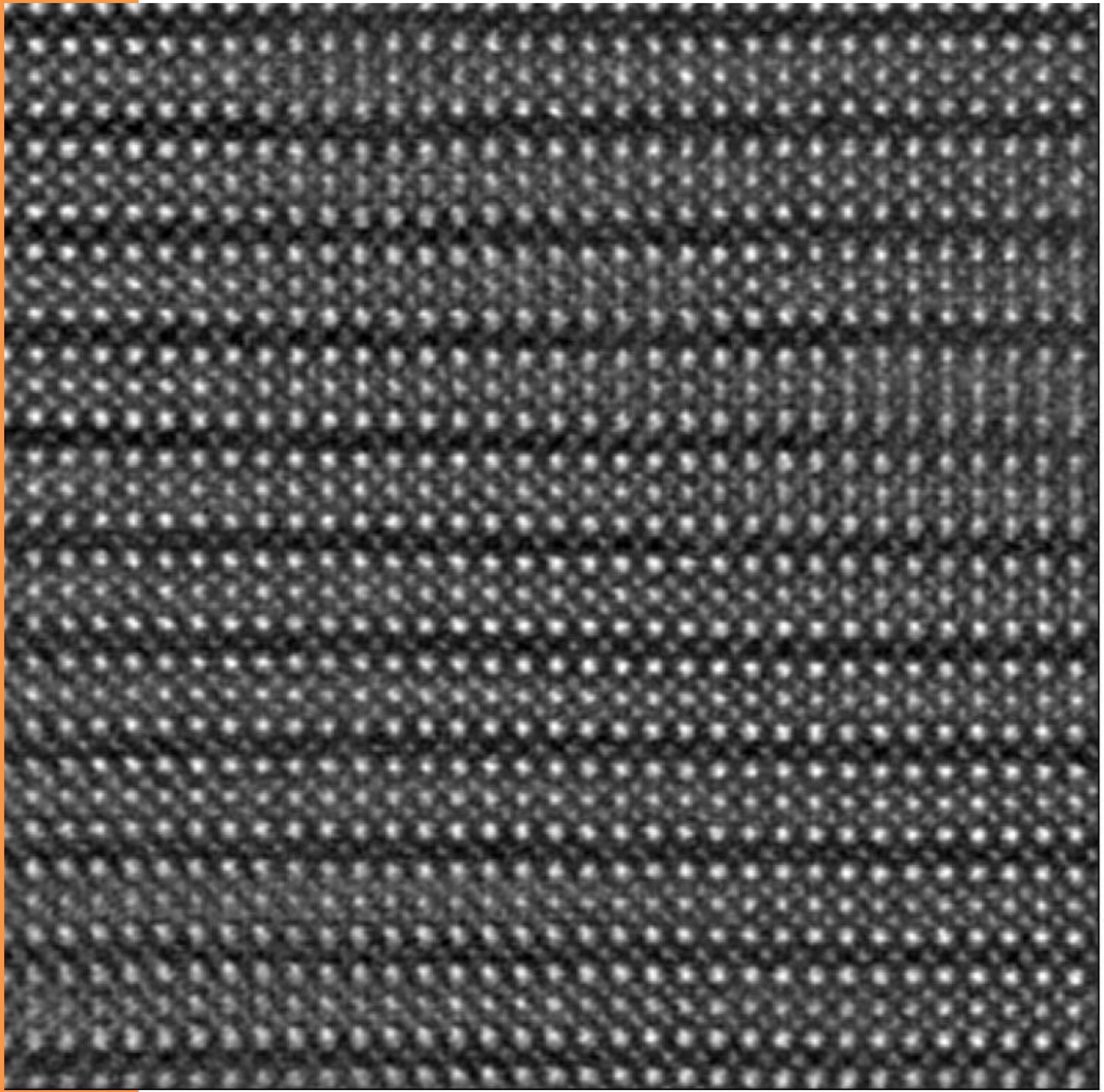


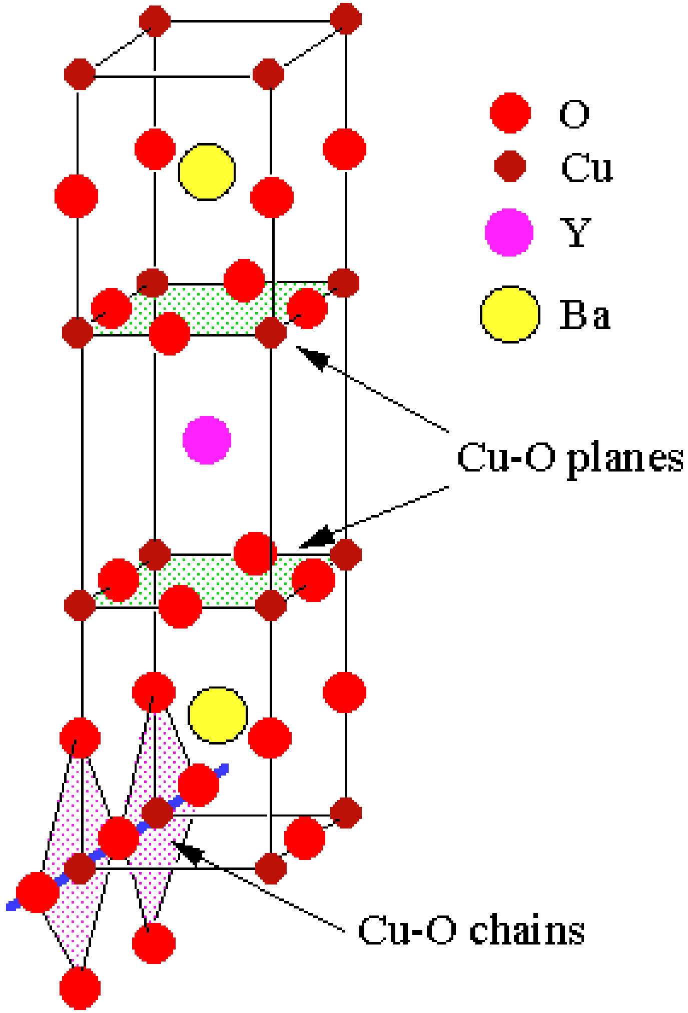
- ML algorithm has 4 competing hypothesis on domain switching mechanisms
- These hypothesis represent full set of possibilities for this system
- The microscope chooses experimental parameters in such a way as to establish which hypothesis is correct fastest
- Important: the same approach can be implemented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

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

Image-based reward functions

… are defined only in the context of the subsequent experiments

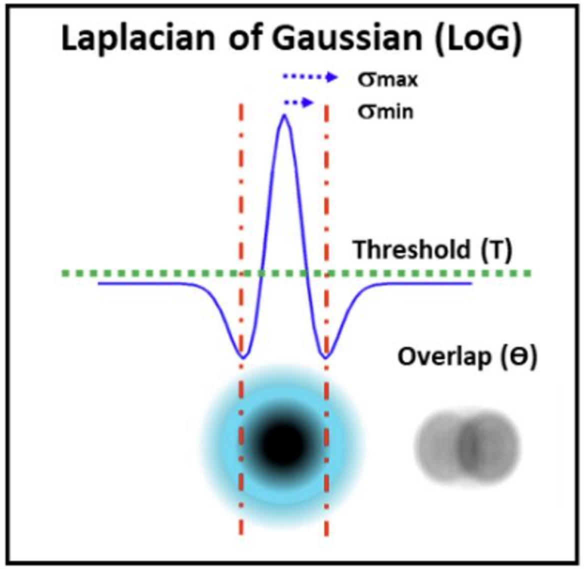




- Radiation damage on oxygen sublattice
- Formation of antiphase defects
- Bending of Cu-O layers

Data by A. Goyal and H. Zhang

DCNN vs LoG*

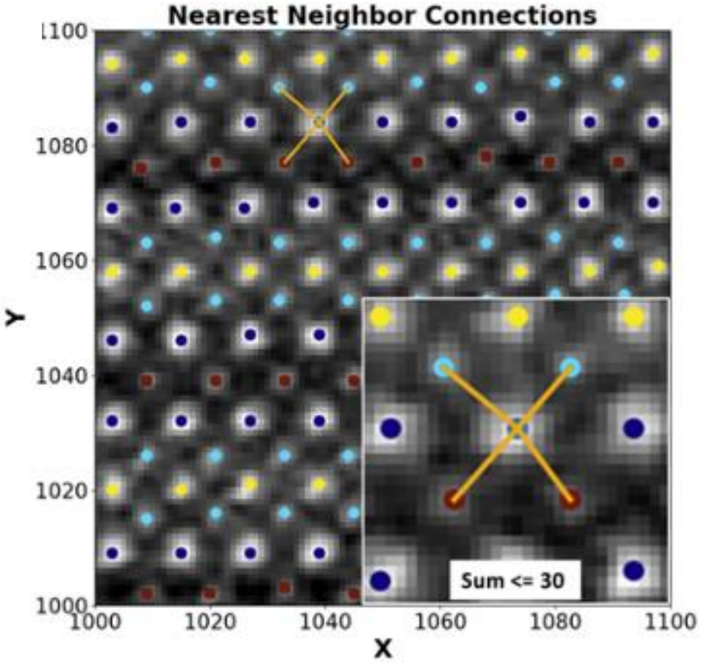


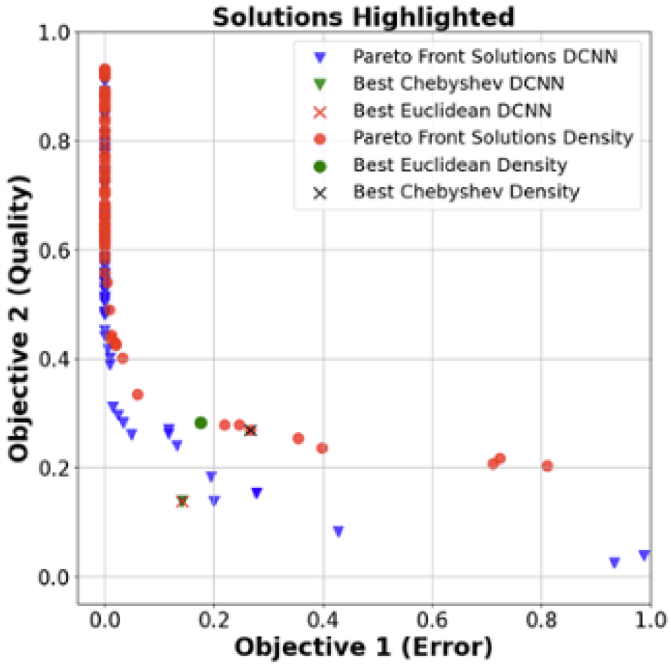
Multiobjective optimization of LoG

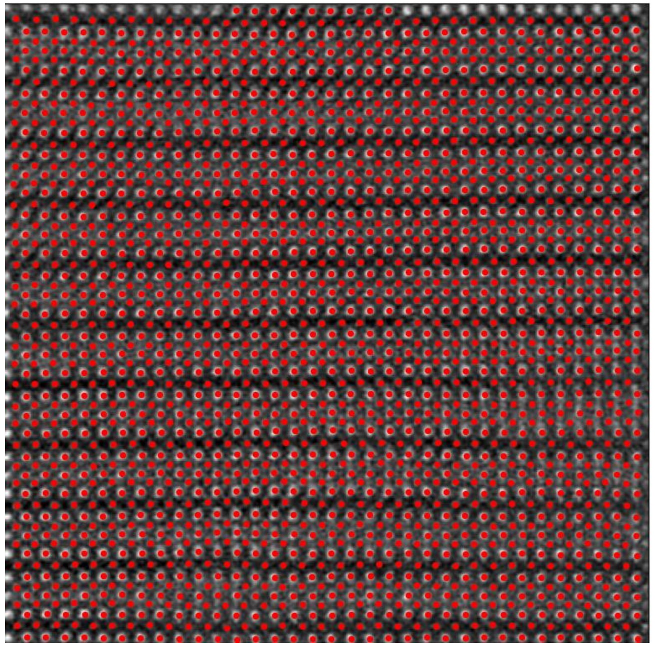
- Number of atoms via **physical criterion** – like number of atoms in the structure with respect to stoichiometry
- False positives (atoms too close)

Benchmarking

- Number of atoms via **DCNN**







Example of analysis pipeline

Image or video

Creating descriptors
(size, shape)

Identity transform

Radon transform

Fourier transform

Dimensionality reduction of sub-images

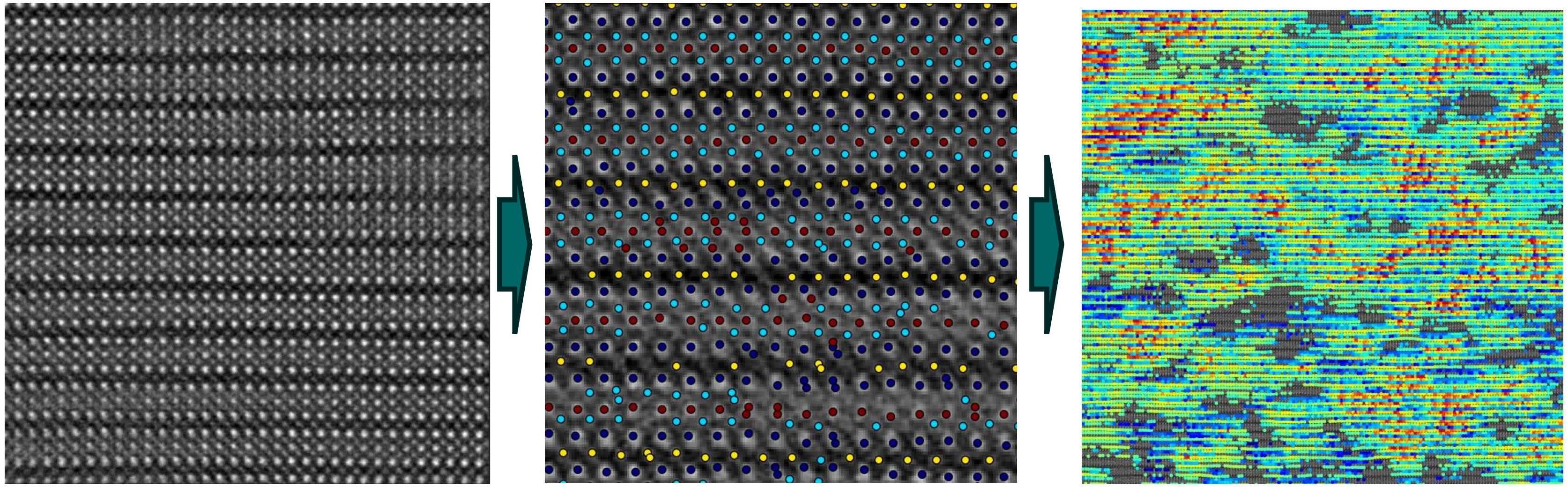
GMM

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

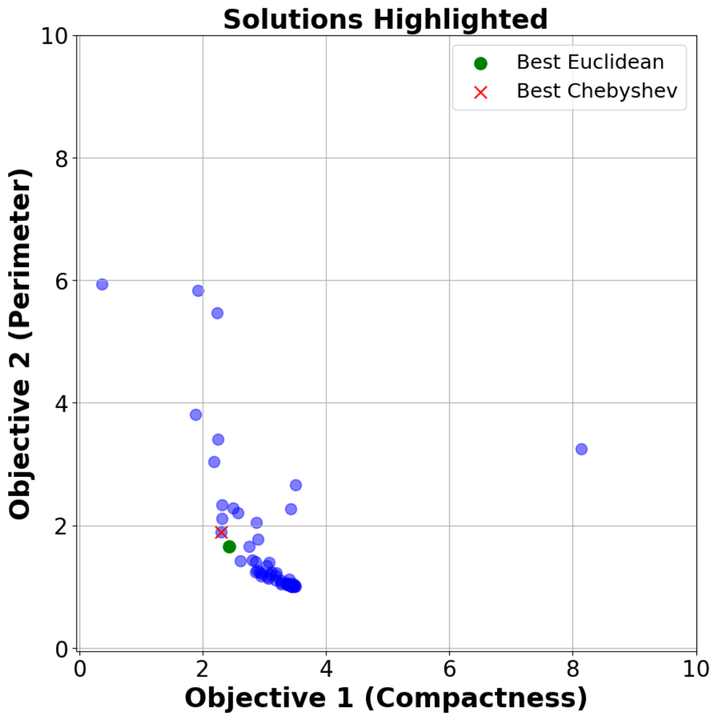
Example of optimization pipeline

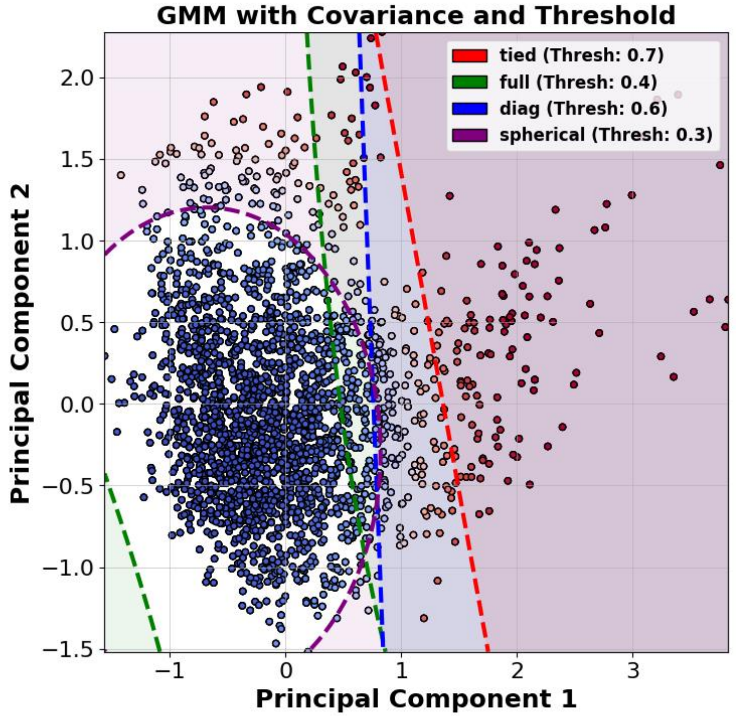
Objectives:

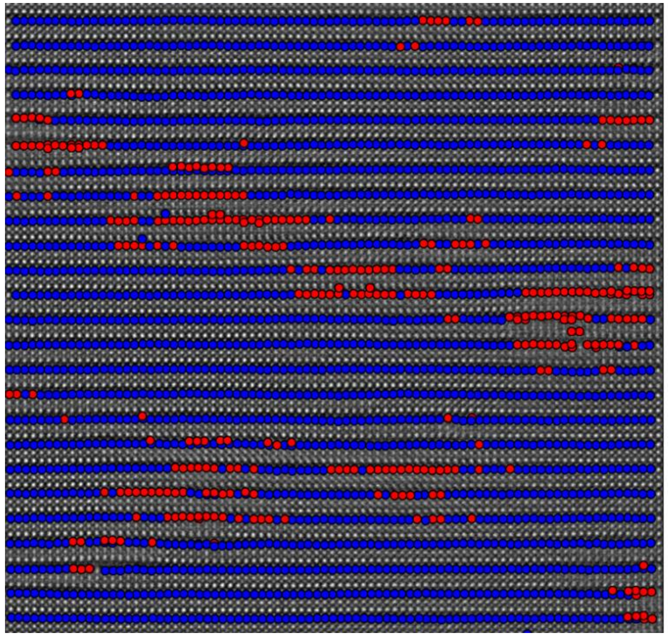
- **Compactness** of the amorphous area
- **Perimeter** of amorphous area

Parameter space:

- Window size = (2, 30)
- Threshold = (0.0, 1.0)
- Covariance Type: tied, full, diag, spherical





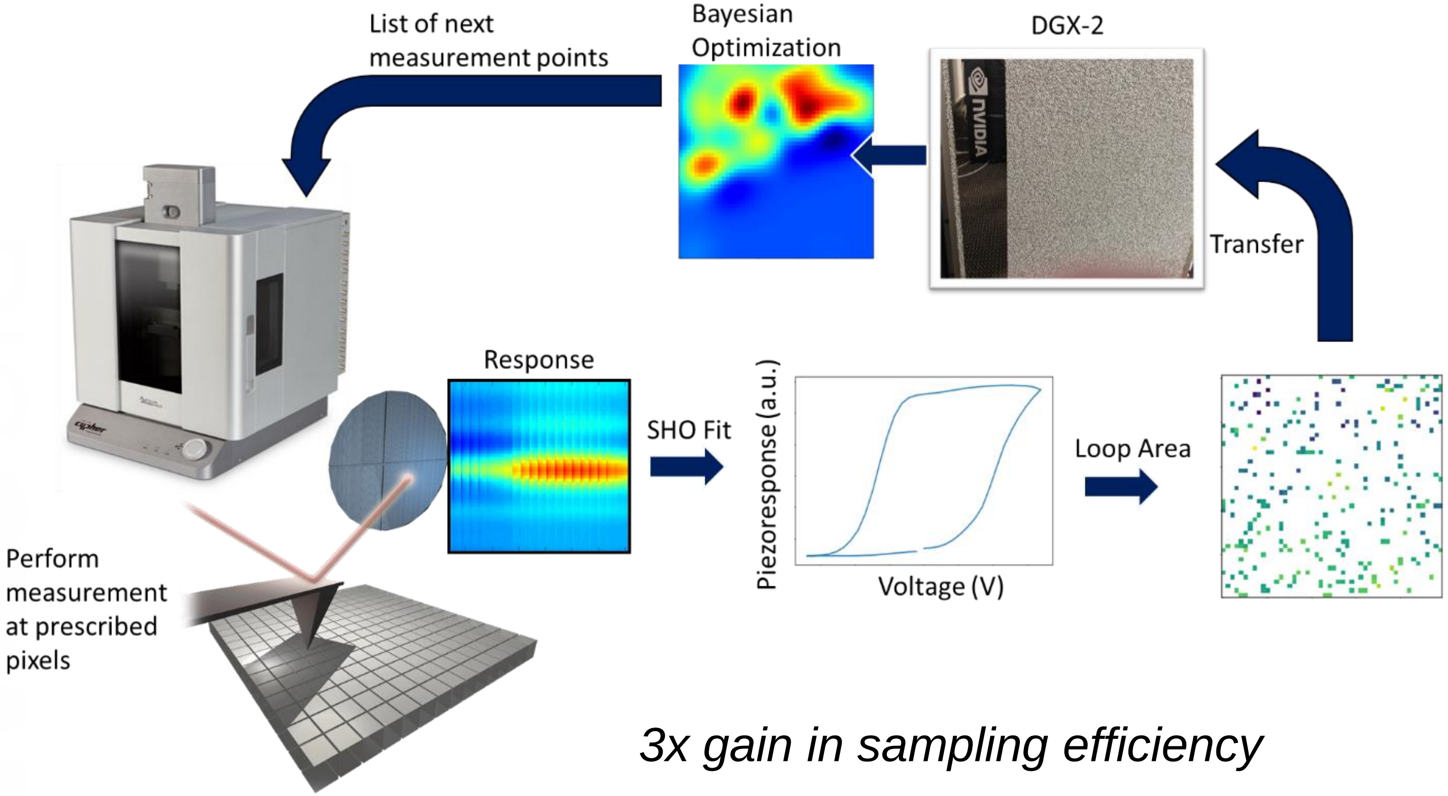


We can complement computationally heavy and out-of-distribution drift prone DCNNs with the light optimization workflows

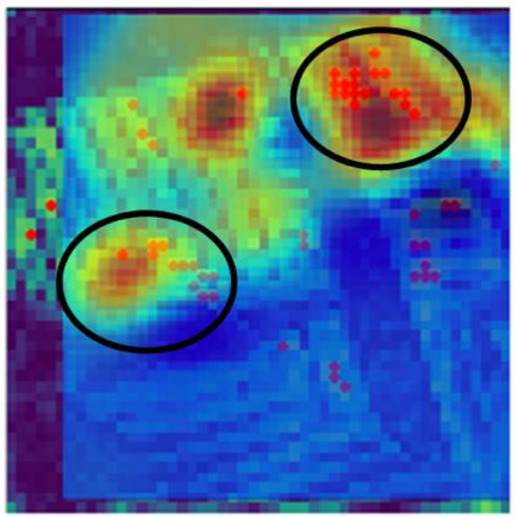
BO in Imaging

BO for Self-Driving Microscope

First implementation of self-driving microscope: 2020

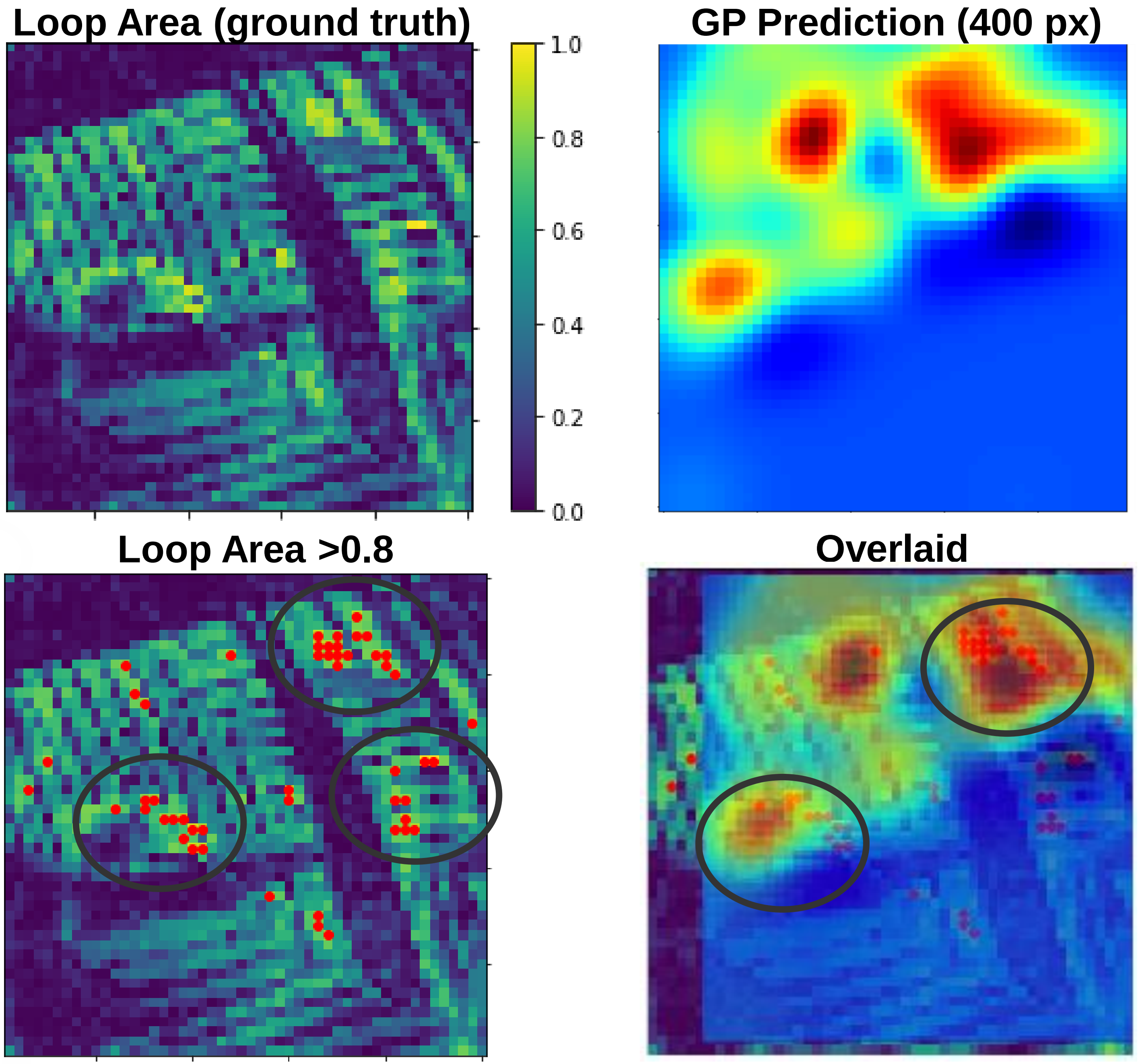


Comparison with "ground truth"



3x gain in sampling efficiency

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



arXiv:2103.12165

arXiv:2011.13050

The application of simple data-driven GP for real world scenarios did not work particularly well.

What is the limitation of the GP/BO?

1. Works only in low-dimensional spaces

2. The correlations are defined by the kernel function (very limiting)

3. We do not use any knowledge about physics of the system

4. We do not use cheap information available during the experiment (proxies)

Can we somehow make high dimensional space low-D?

Two modes of operations

Structural imaging (Cheap)

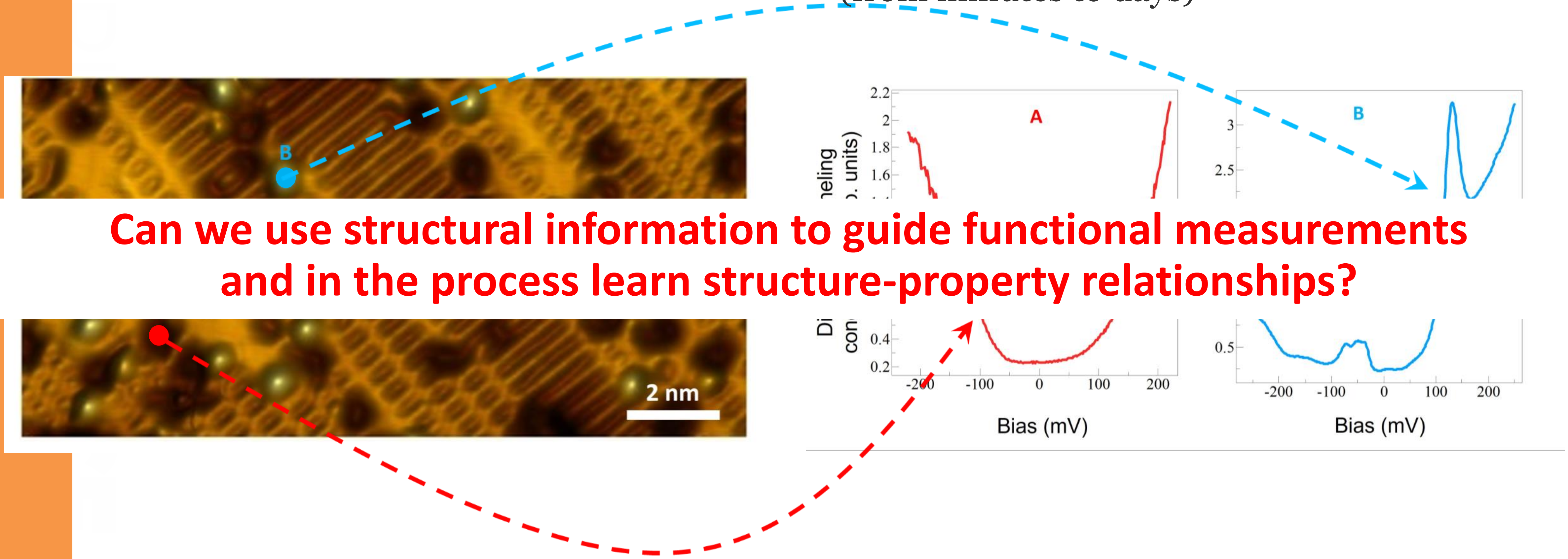
Topography in STM, amplitude/phase in SPM, (HA)ADF-image in STEM, etc.

These are FAST measurements
(from seconds to minutes)

Functional imaging (Costly)

Scanning tunneling spectroscopy (STS), polarization loops in SPM, EELS in STEM, etc.

These are SLOW and/or DESTRUCTIVE measurements
(from minutes to days)

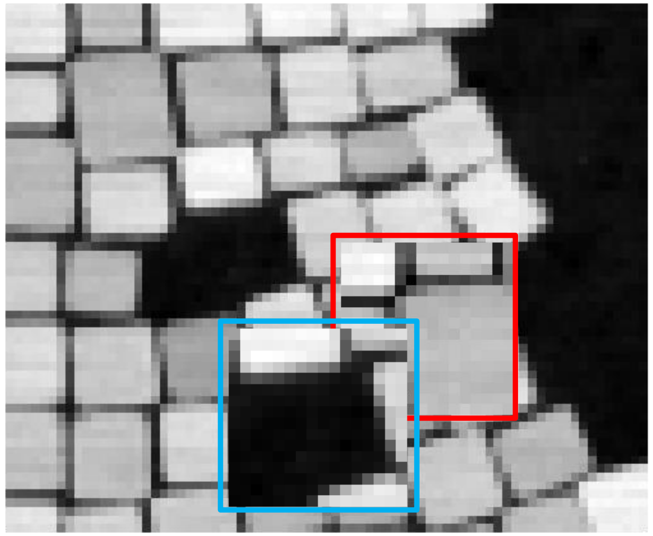


Can we use structural information to guide functional measurements and in the process learn structure-property relationships?

Physics discovery in active experiments

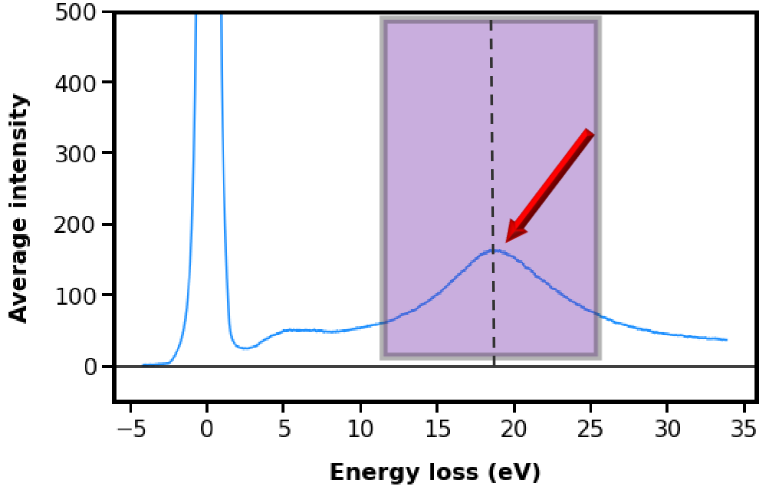
- Suppose we know what physical behavior/property we are interested in (superconductivity, ferroelectric switching, plasmonic modes, etc.)
- This behavior is encoded in spectra that we can measure everywhere in the sample (size of superconducting gap, polarization loop area, peak intensity, etc.)
- We want to identify (local) structural features where this behavior is maximized/minimized
- We want to achieve this with as few measurements as possible (**< 5% of the entire grid**)

Structure

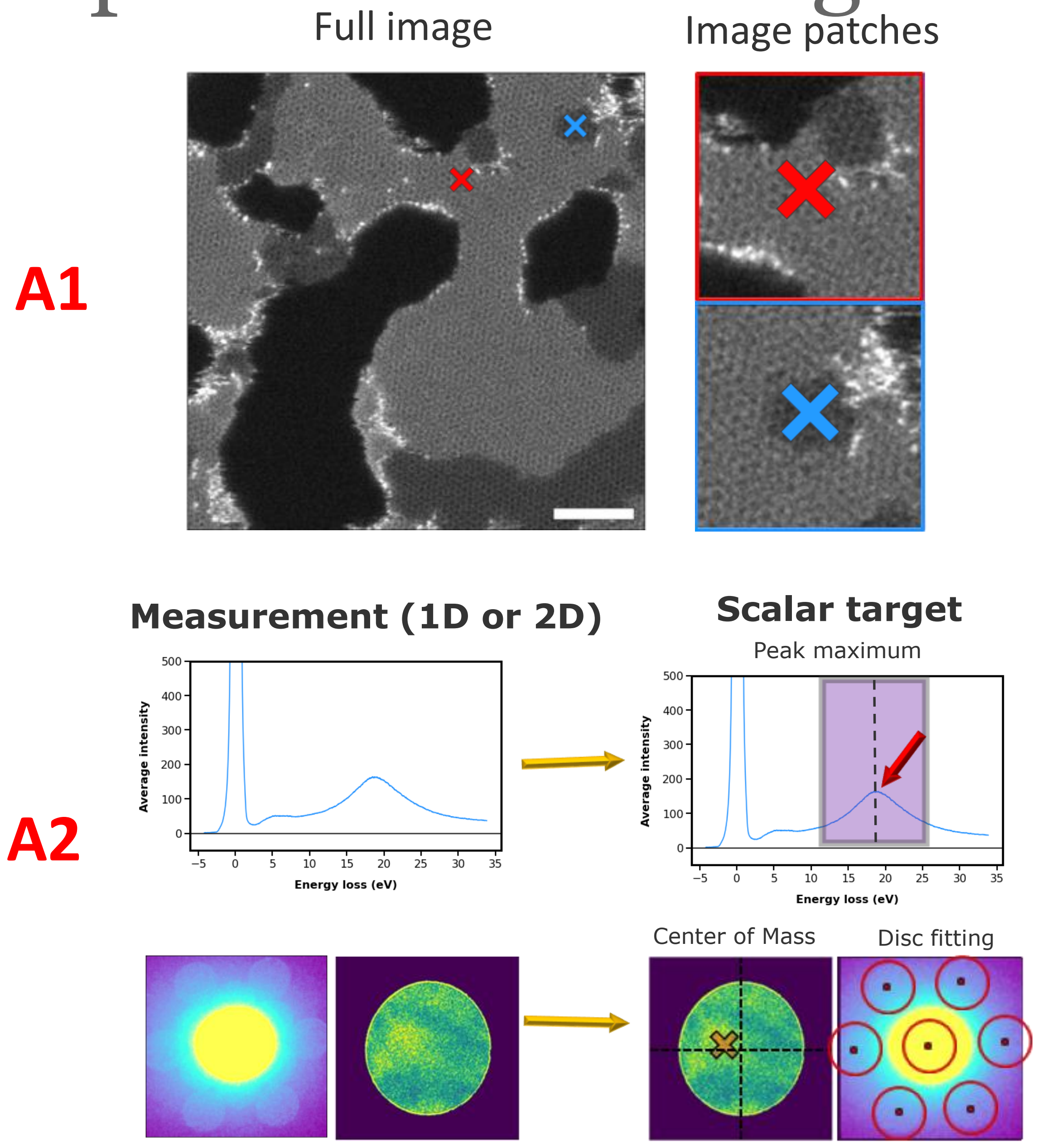


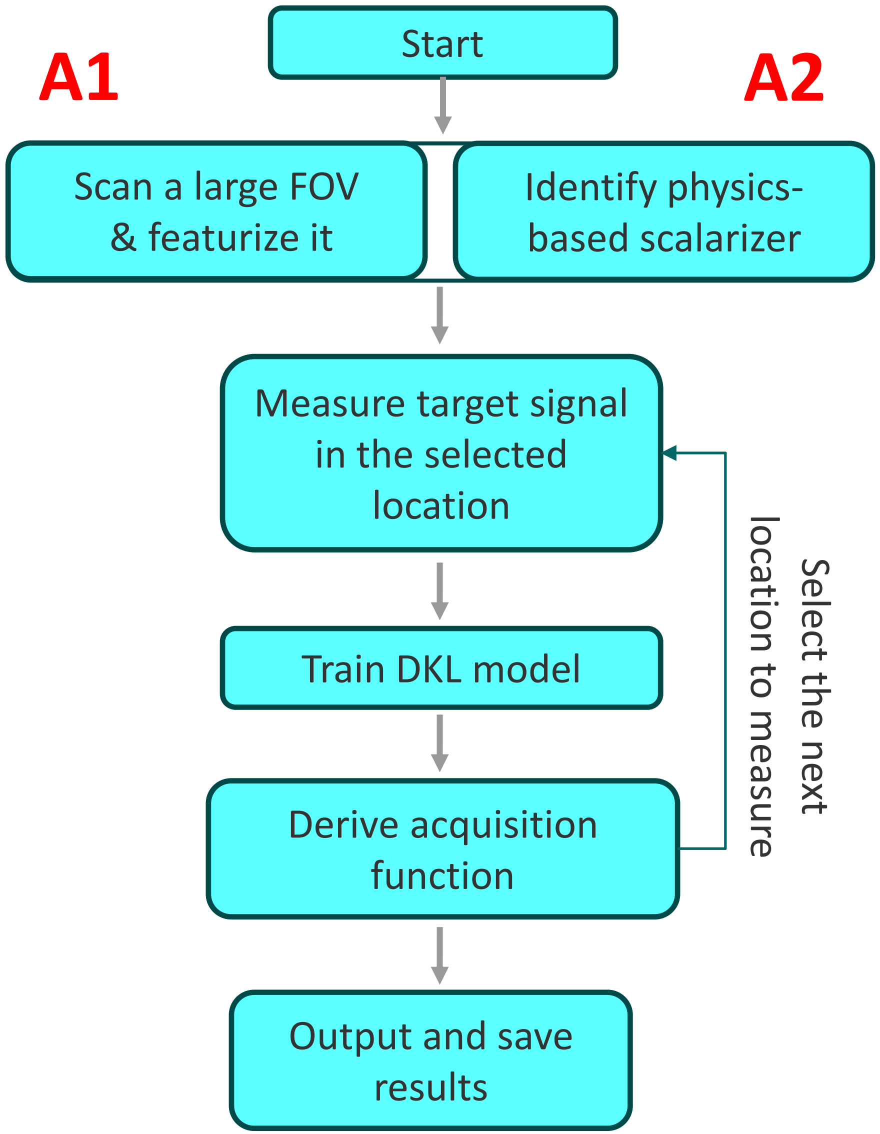
?

Physical behavior of interest

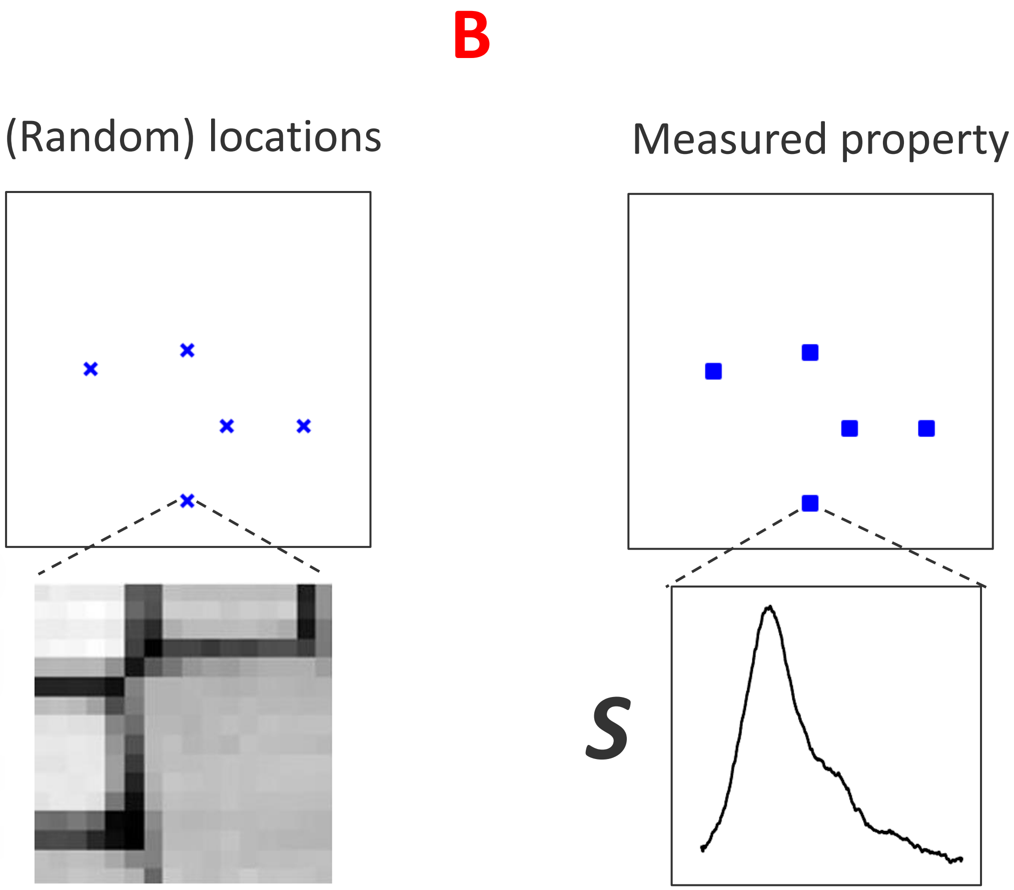


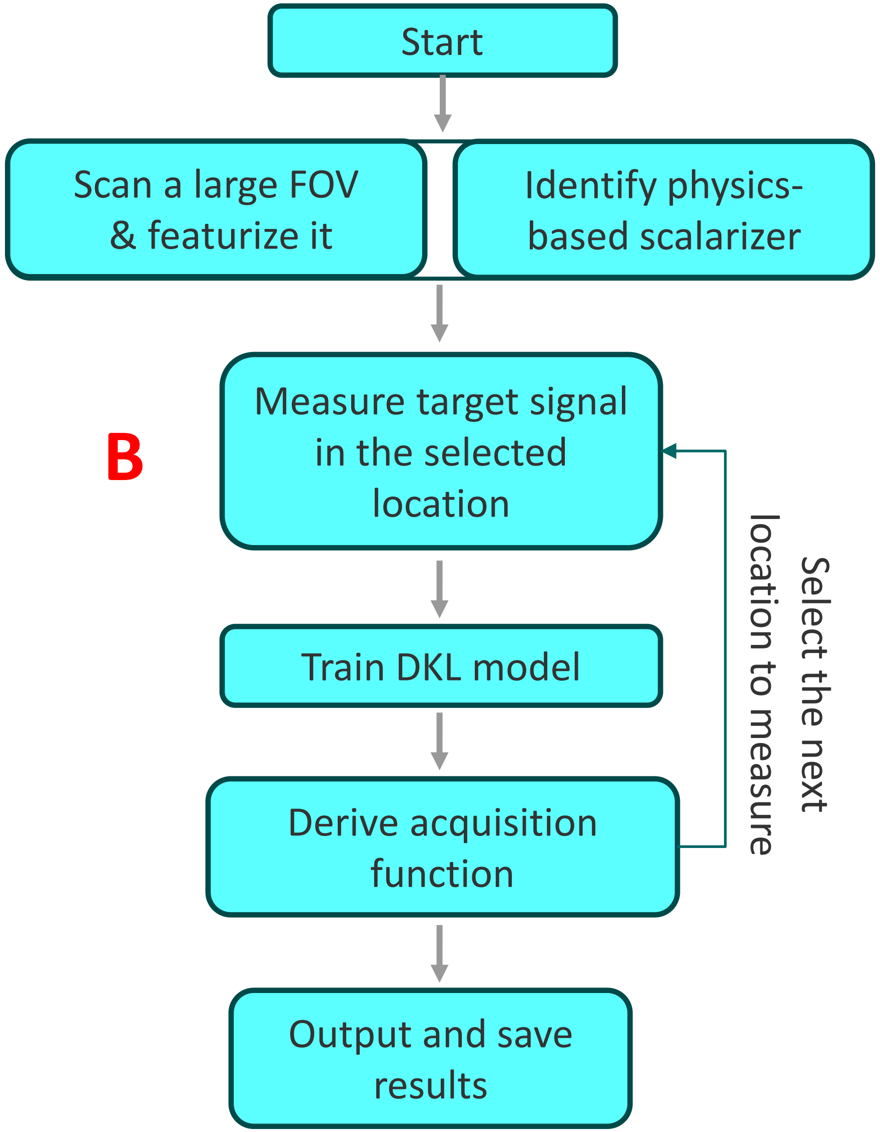
Deep Kernel Learning: Step 1



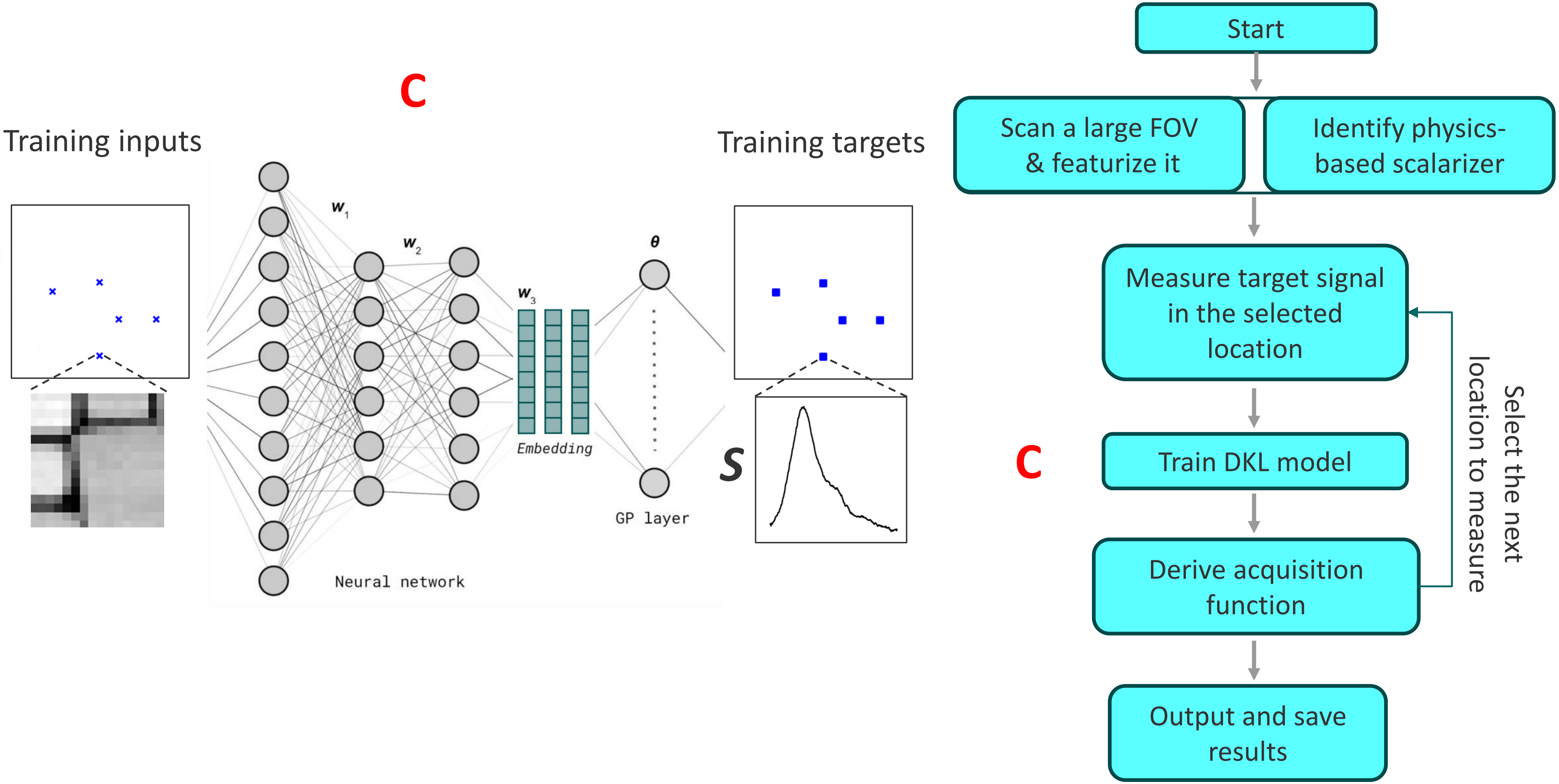


Deep Kernel Learning: Step 2



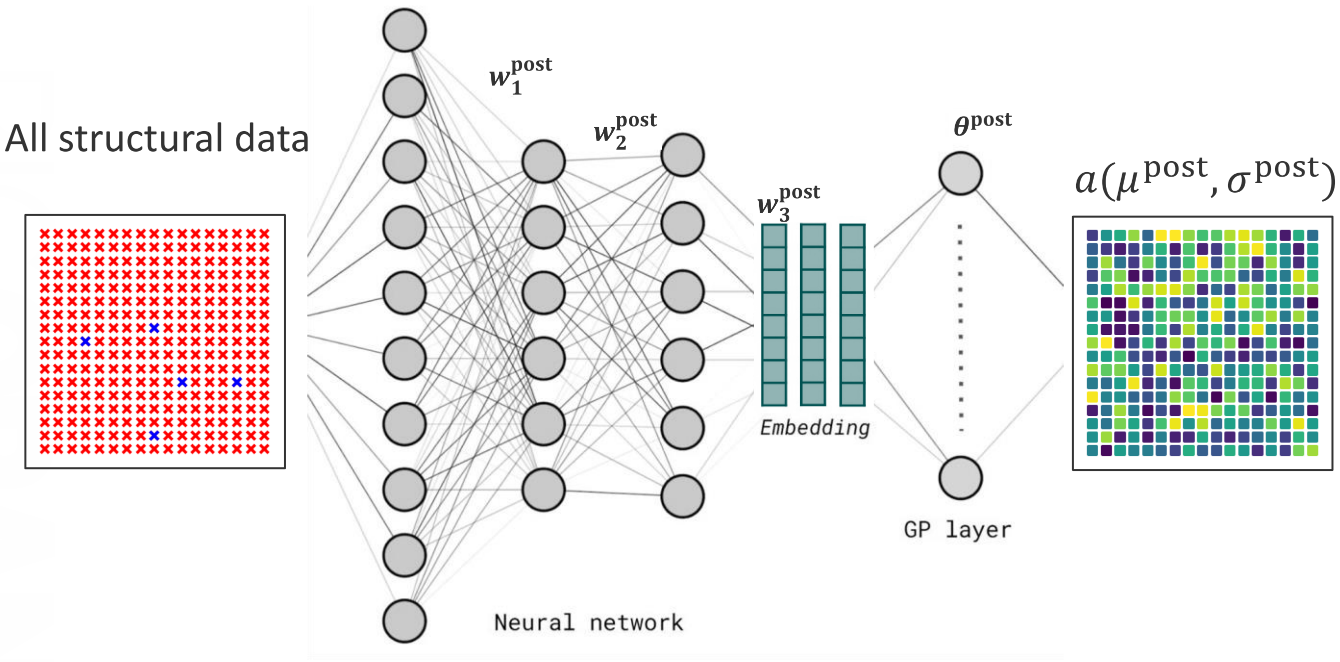


Deep Kernel Learning: Step 3



Deep Kernel Learning: Going Active

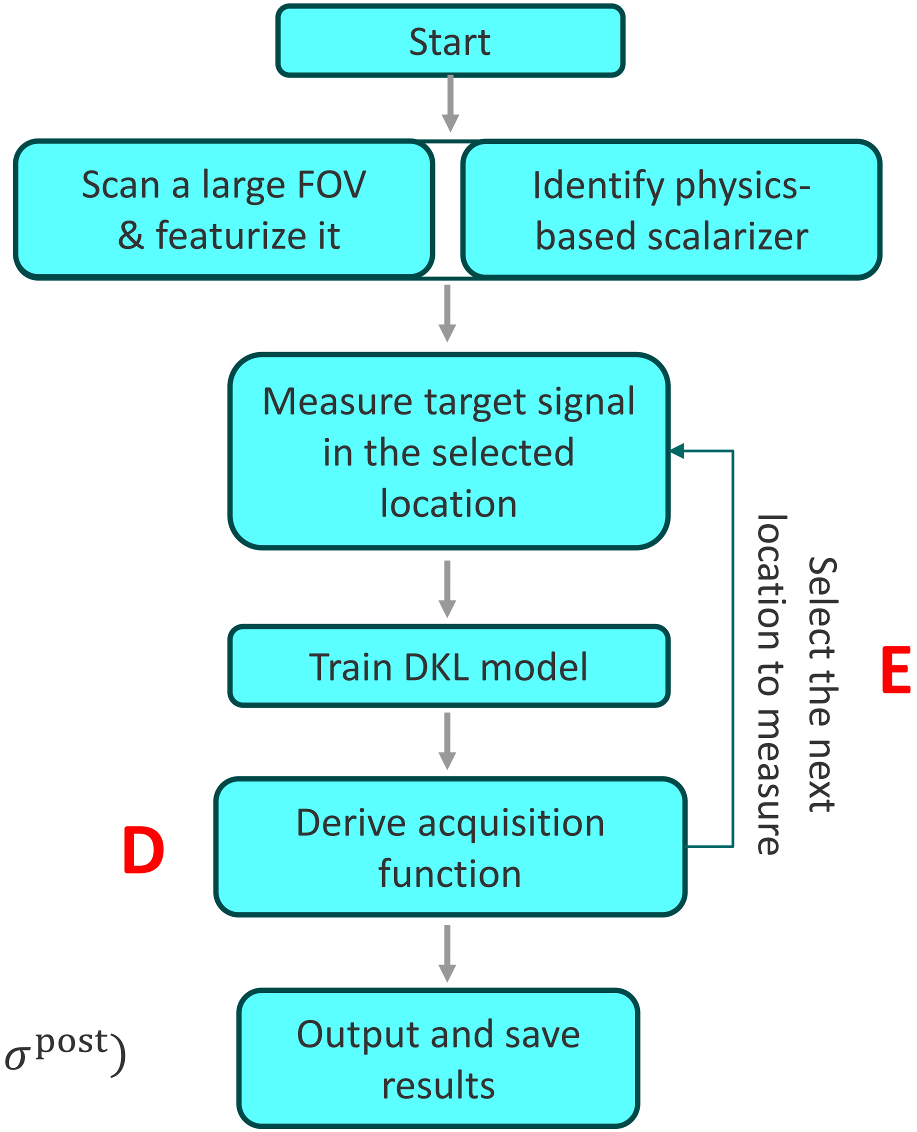
D



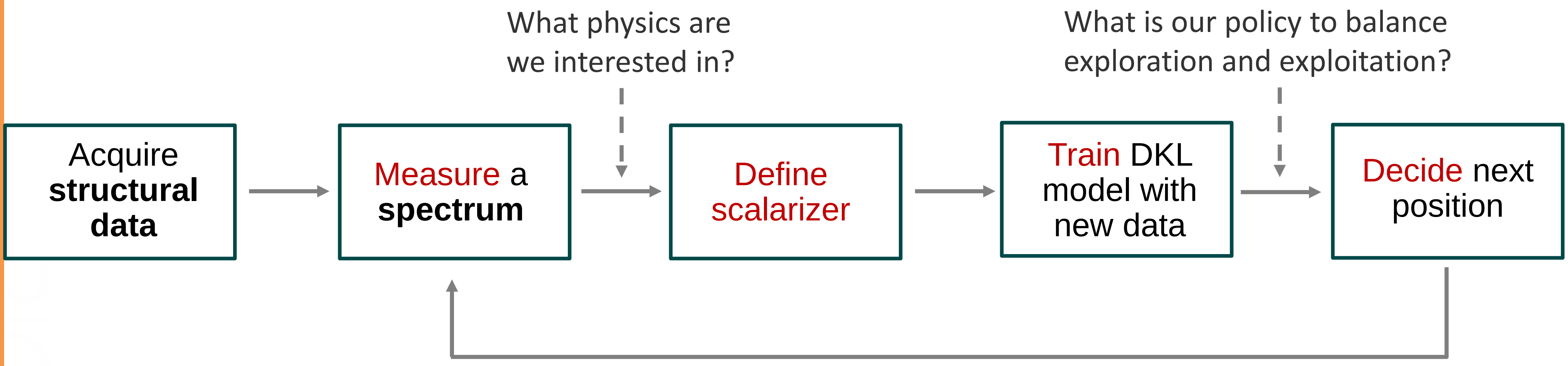
E

Next point to measure:

$$x_{next} = \arg\max_{\mathbf{x}} a(\mu^{post}, \sigma^{post})$$



Deep Kernel Learning based BO

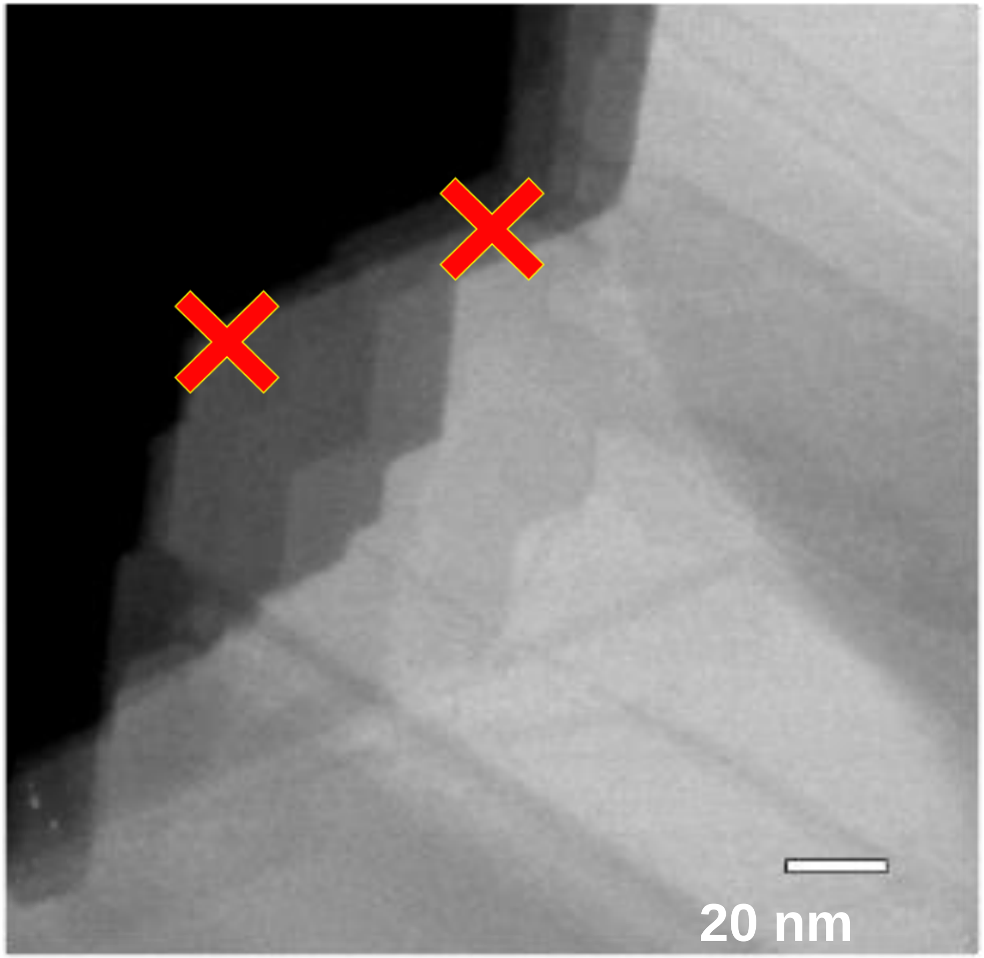


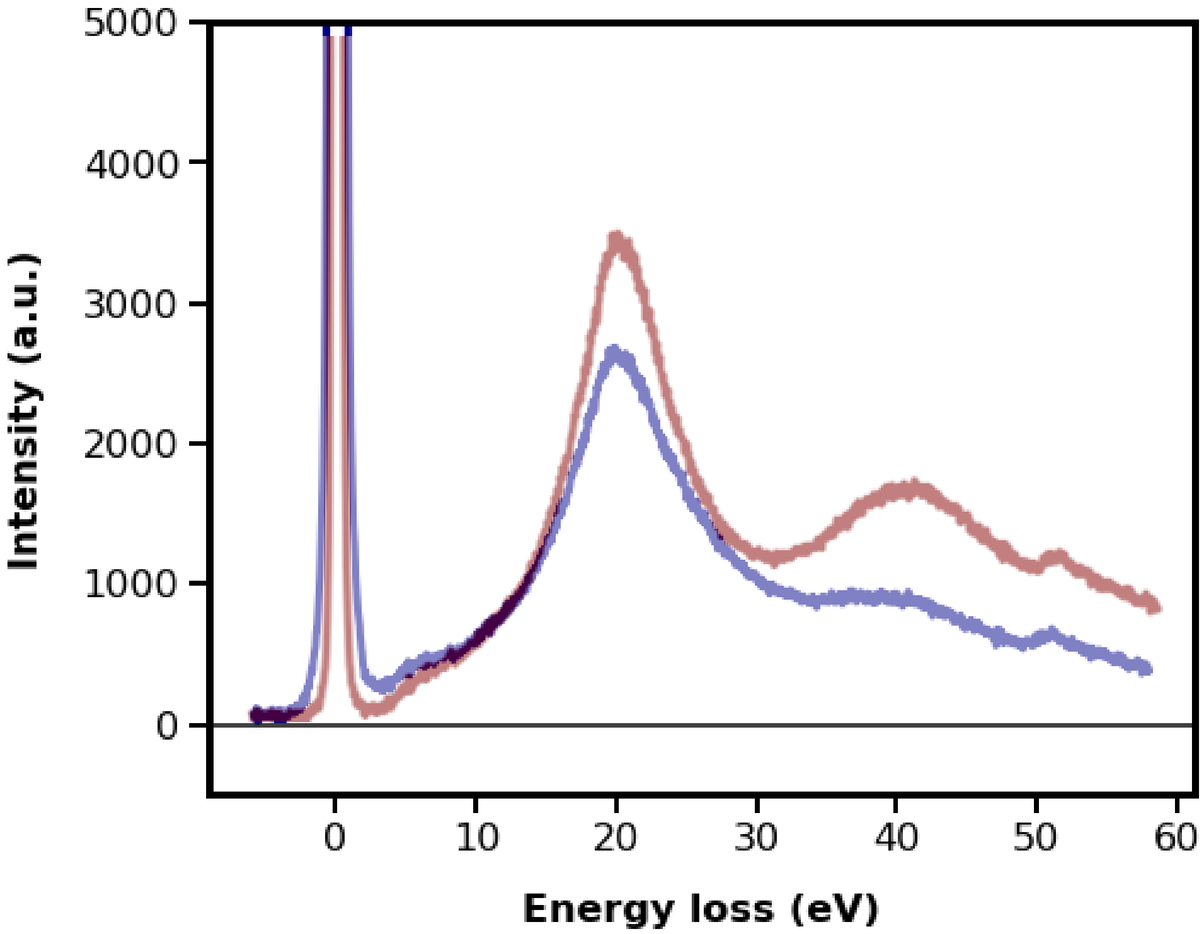
Key concepts:

- **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

From Static to Active Learning



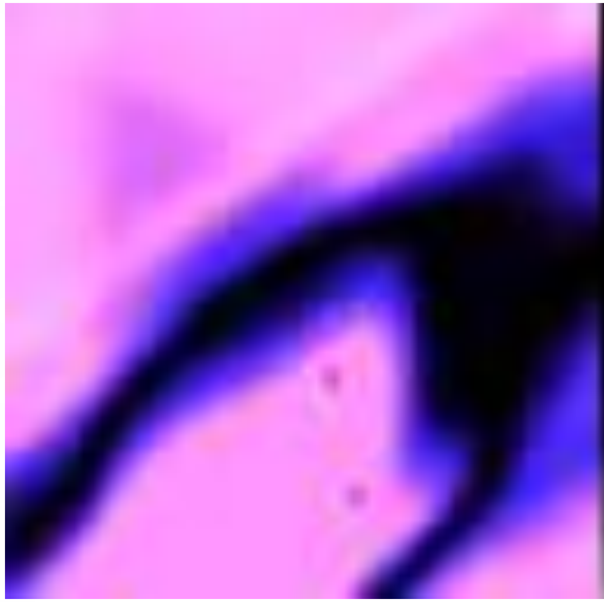


1. What if we have full access to structural information
2. And want to choose locations for (EELS, 4D STEM, CL, EDX) measurements
3. So as to **learn** relationship between structure and spectrum fastest
4. Or **discover** which microstructural elements give rise to specific **desired** spectral features?

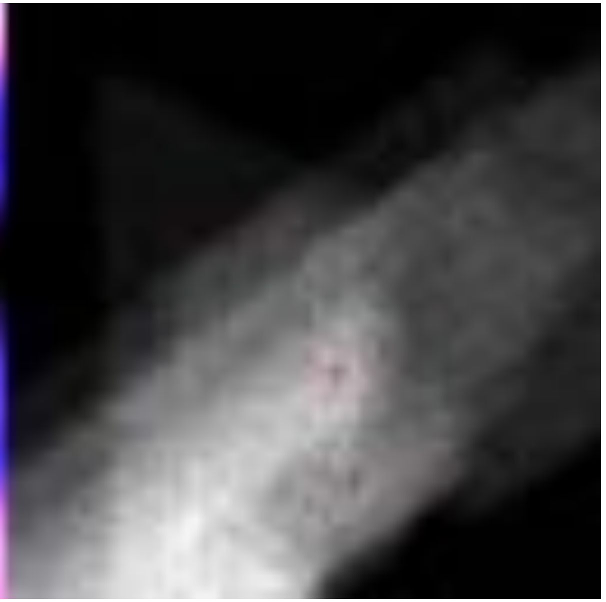
Discovering Regions with Interesting Physics

- **Discovering physics in a “new” material $MnPS_3$**
- **Curve fitting** to help enforce physical processes

Acquisition function

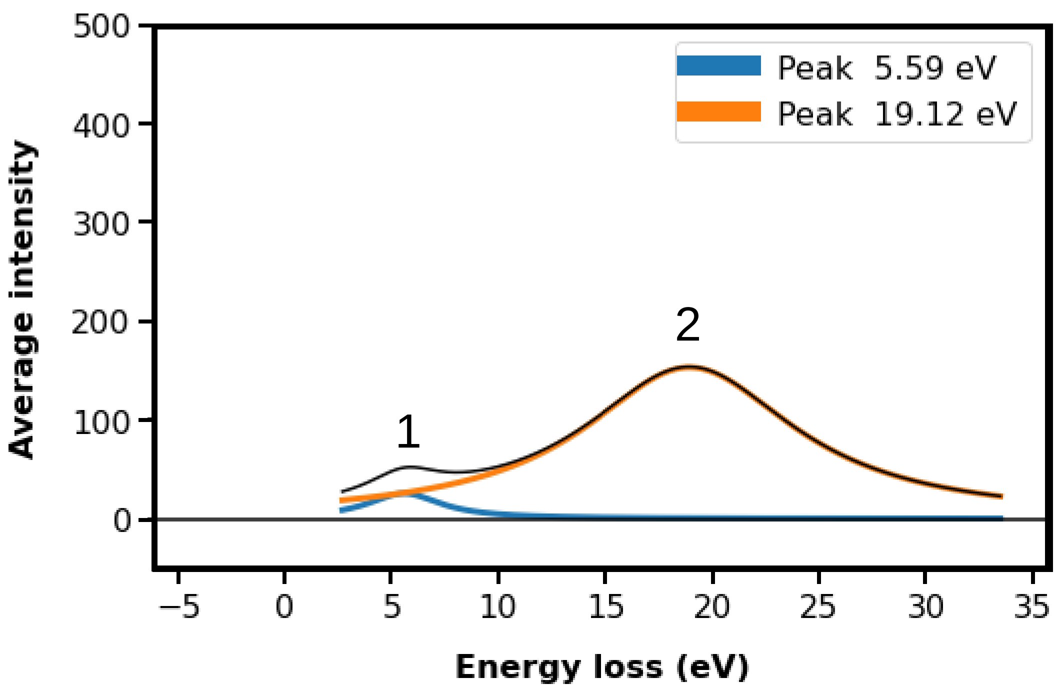


HAADF-STEM

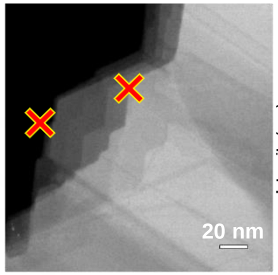


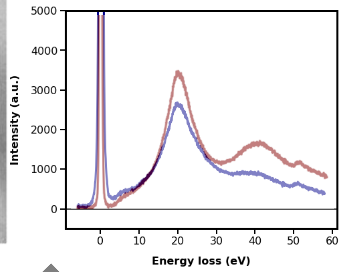
Physics search criteria:

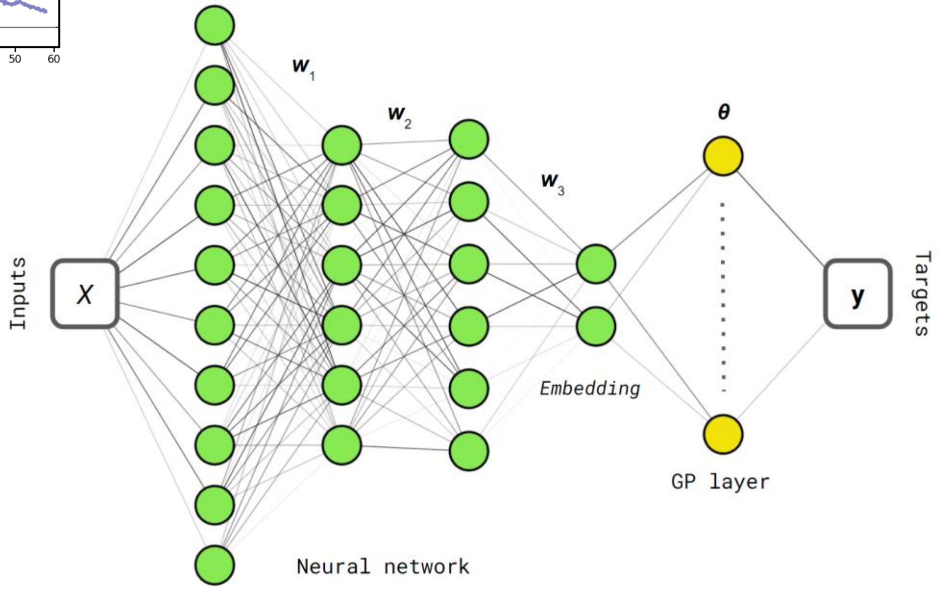
$$\boldsymbol{Ratio} = \boldsymbol{Peak}\,\mathbf{1} \,/\, \boldsymbol{peak}\,\mathbf{2}$$



Deep Kernel Learning







<u>Specify physics criteria</u>

Acquire **structural data**

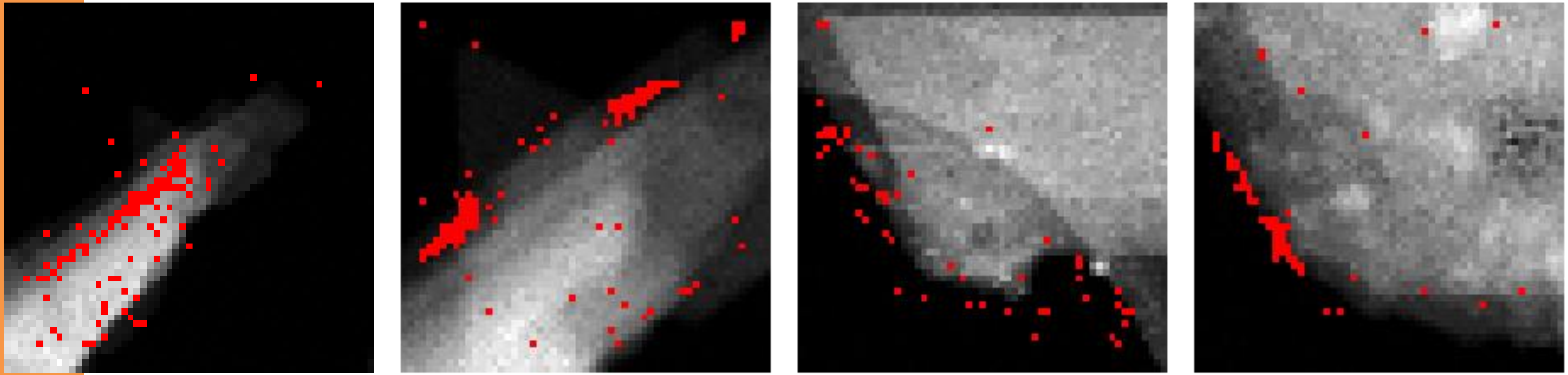
Measure a **spectrum**

Train DKL model with new data

Decide next position (optimize physics criteria)

Allows navigation of the system to search for physics

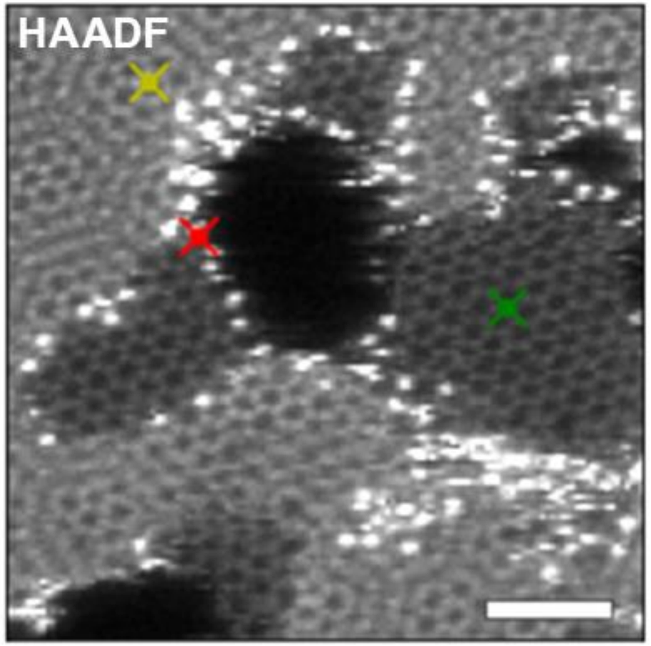
More Examples of Physics Discovery

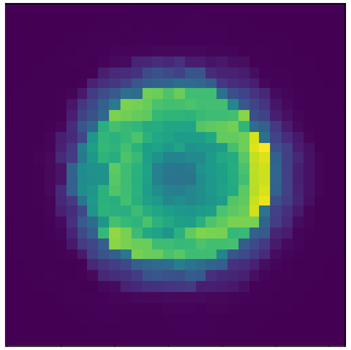


Discovery pathway depends on the reward structure (scalarizer that defines signature of physics we want to discover)!

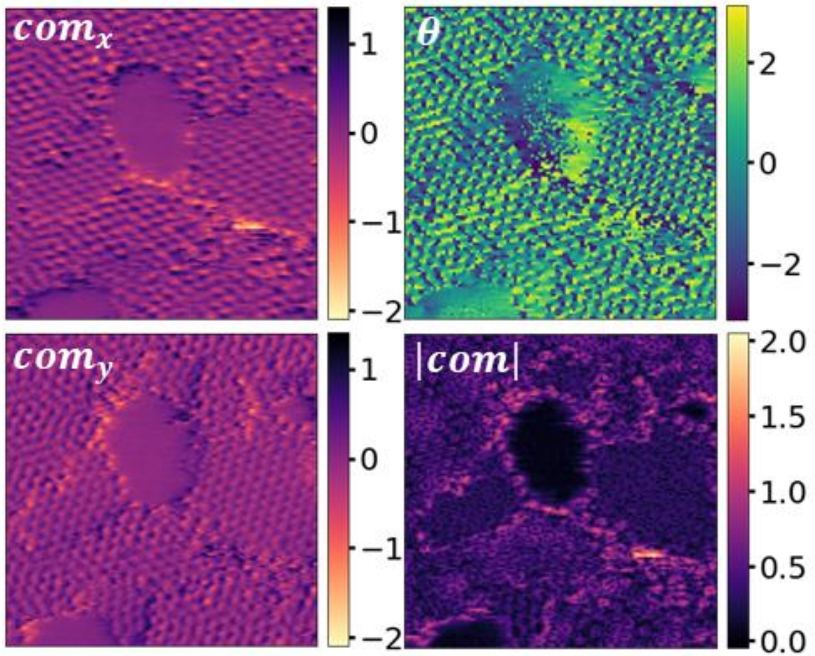
- Currently, we run 4D STEM measurements on a grid.
- What if we want to explore smarter workflows – where microscope chooses where to take 4D STEM measurements?
- **Direct:** We can do it for a priori known objects of interest
- **Inverse:** Or we can aim to discover objects which have predefined signatures of interest in 4D STEM data

4D STEM: Grid, Direct, and Inverse





Quantities to explore
- Electric field
- Potential
- Charge density
- Strain

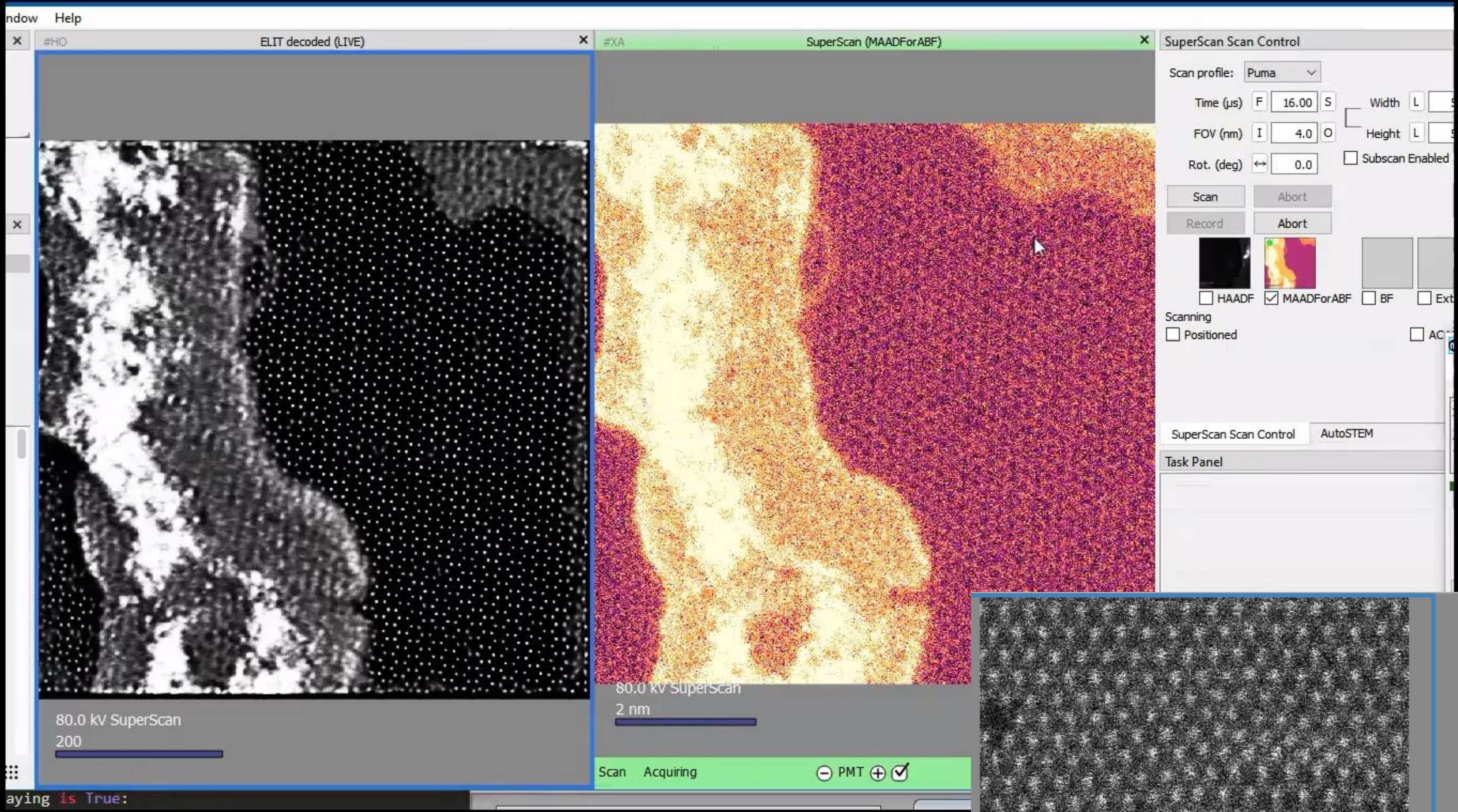


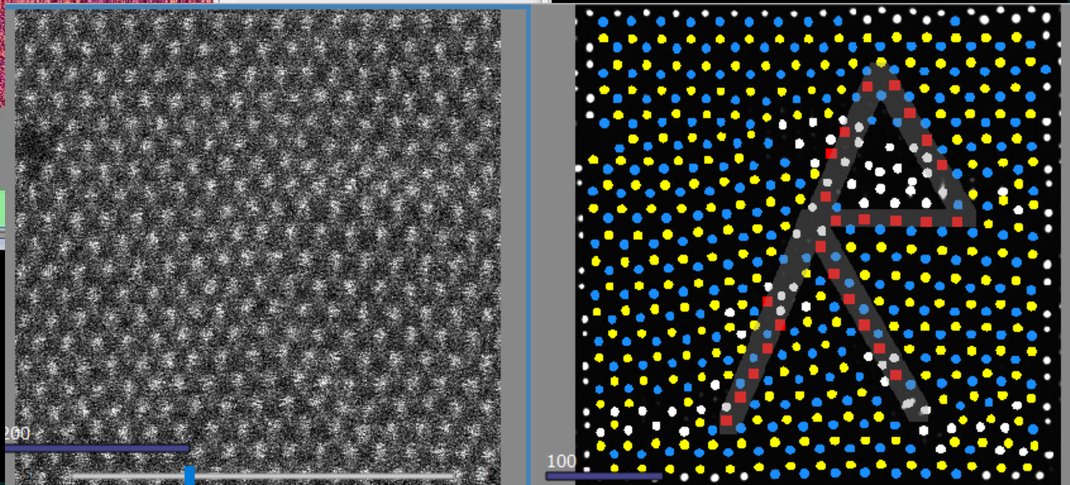
- What can we say about structure?
- Interesting functionalities are expected at the certain structural elements
- We can guess some; we have to discover others
- Multiple goals while running experiment

- **Policy:** **what do we do depending on observation**
- **Reward:** **what do we hope to achieve**
- **Value:** **anticipated reward**

Direct experiment: ELIT (2021)

Implementation: Kevin Roccapriore, Ayana Ghosh, Sergei V. Kalinin & Maxim Ziatdinov





Inverse: Deep Kernel Learning based BO

What physics are we interested in?

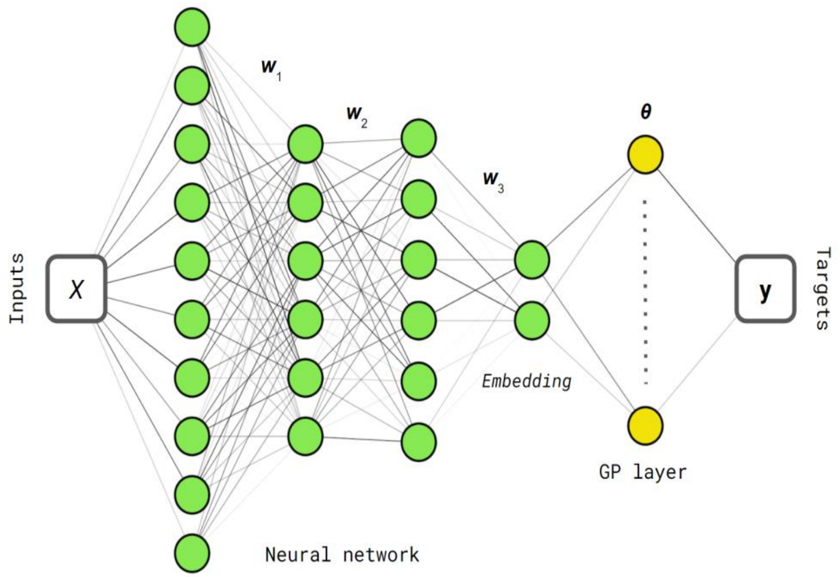
What is our policy to balance exploration and exploitation?

Acquire **structural data** → Measure a **ronchigram** → Define scalarizer → Train DKL model with new data → Decide next position → (back to Measure a ronchigram)

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



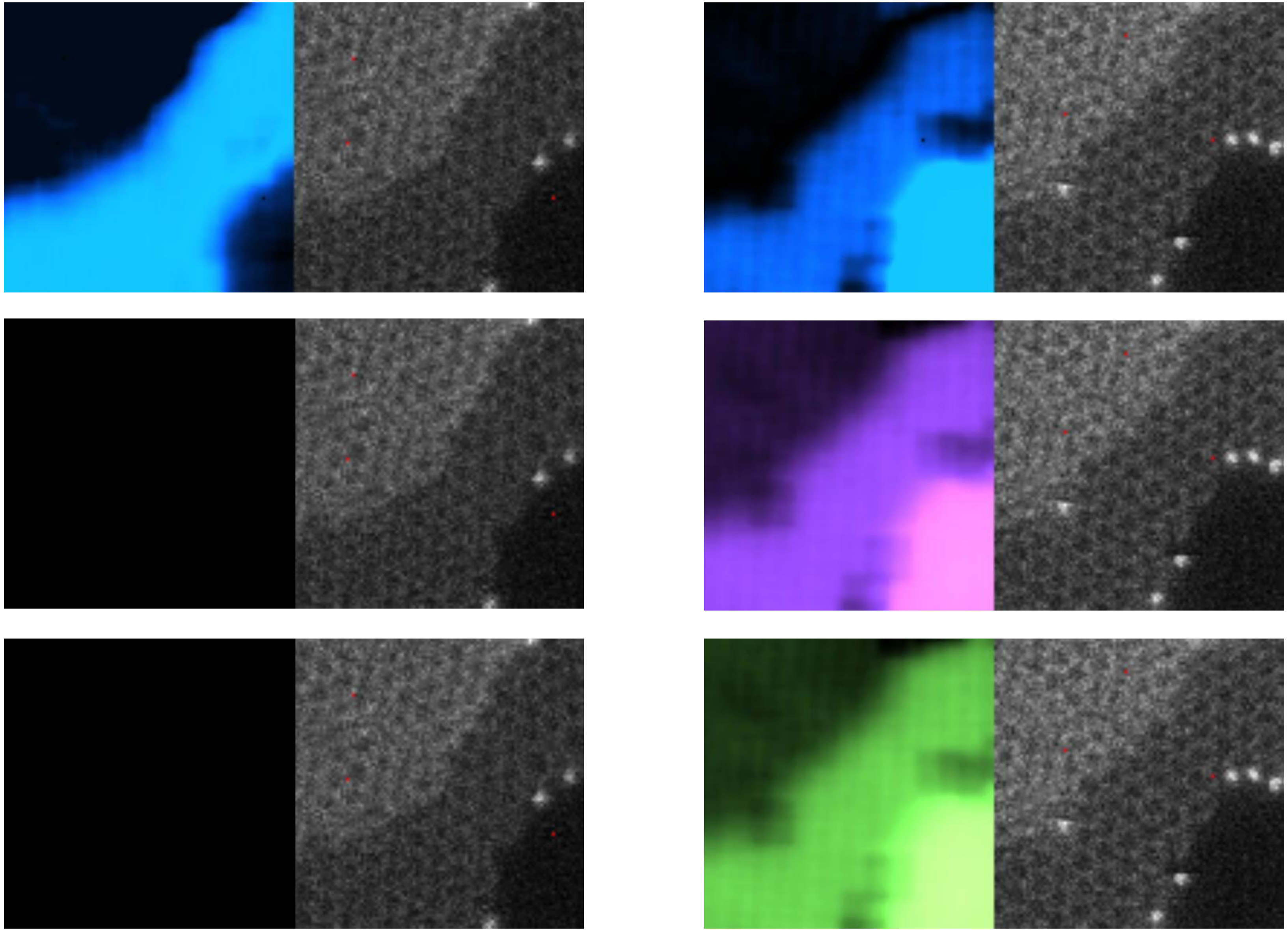
DKL on pre-acquired data

Scalarizer: CoM angle | **Scalarizer: CoM magnitude**

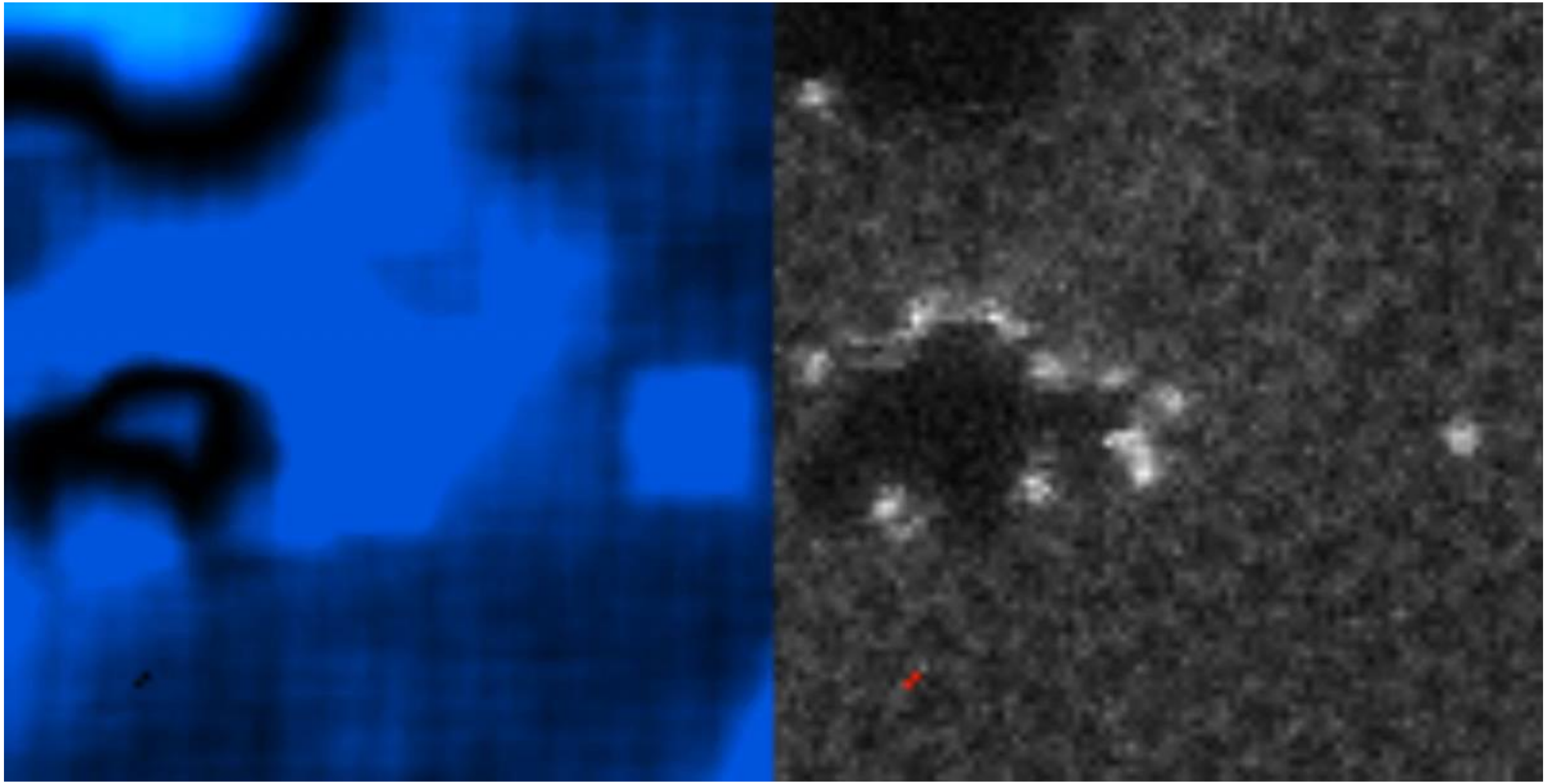
Acquisition function

Prediction map

Uncertainty map



DKL on Active Microscope

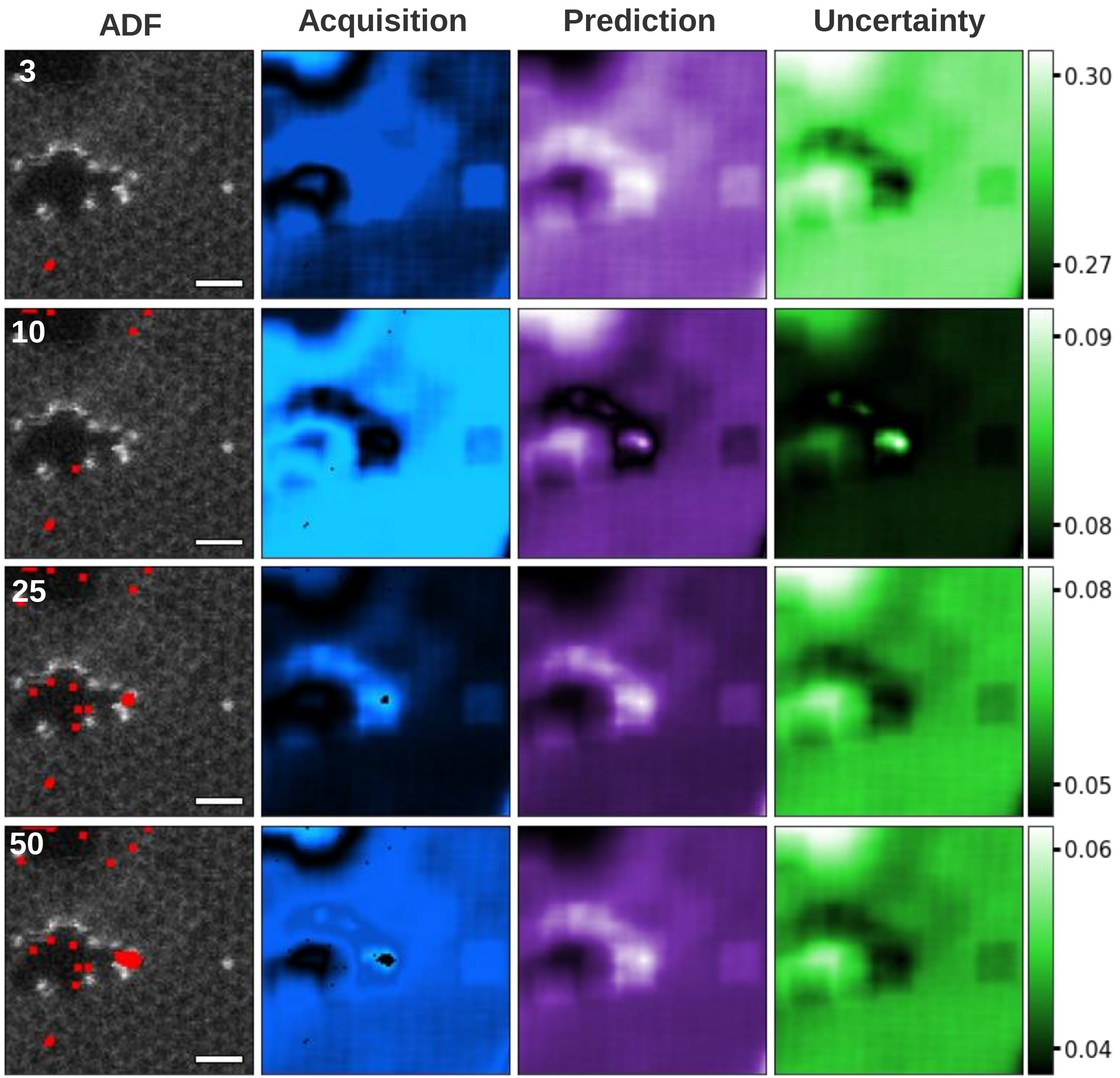


- Different **acquisition functions** can be used:
  - Expected Improvement (**EI**) (usually what was used)
  - Upper Confidence Bound (**UCB**), etc
- Usually based on some combination of **prediction** and **uncertainty**.

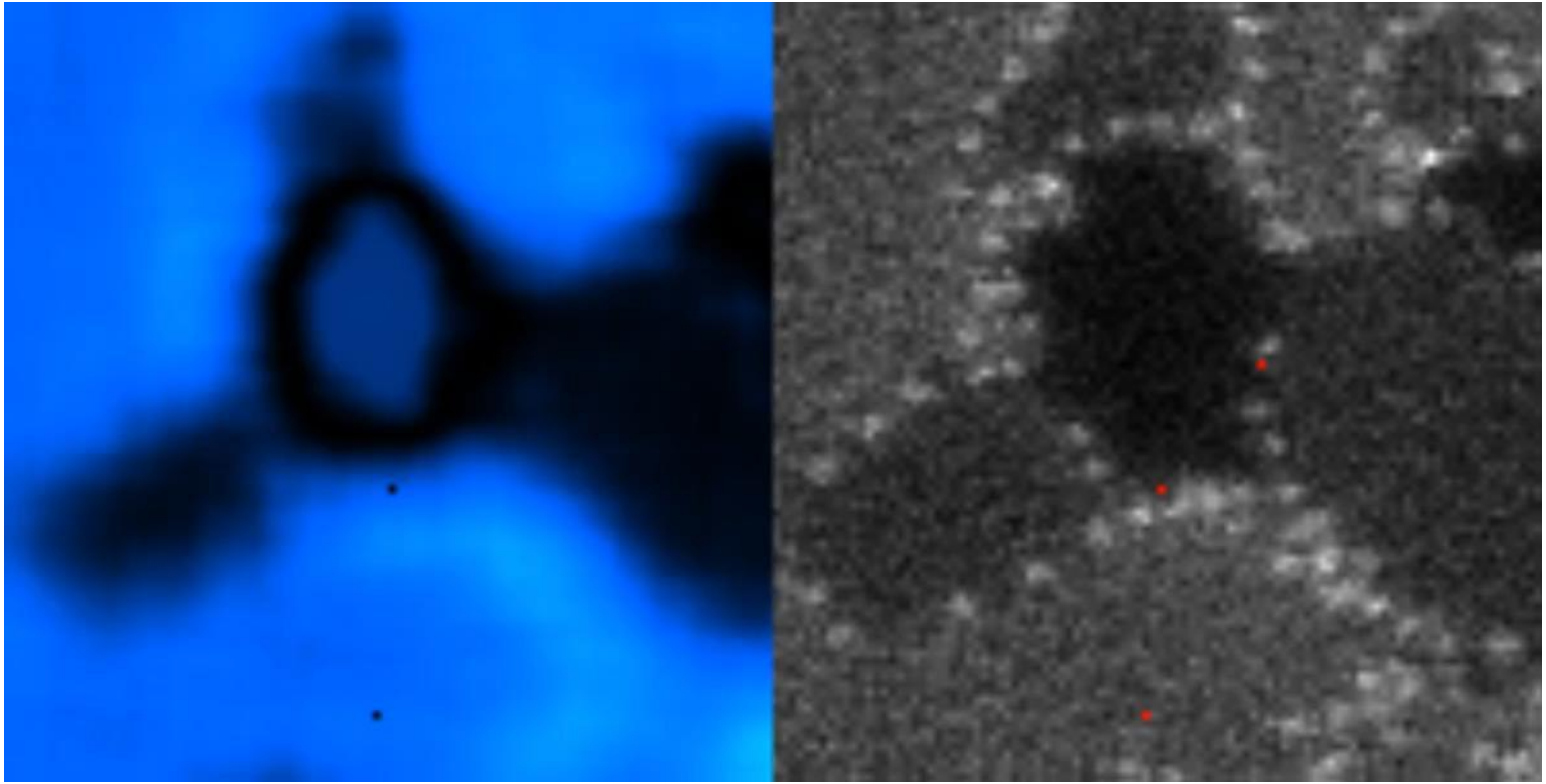
A closer look

Scalarizer: *CoM* magnitude

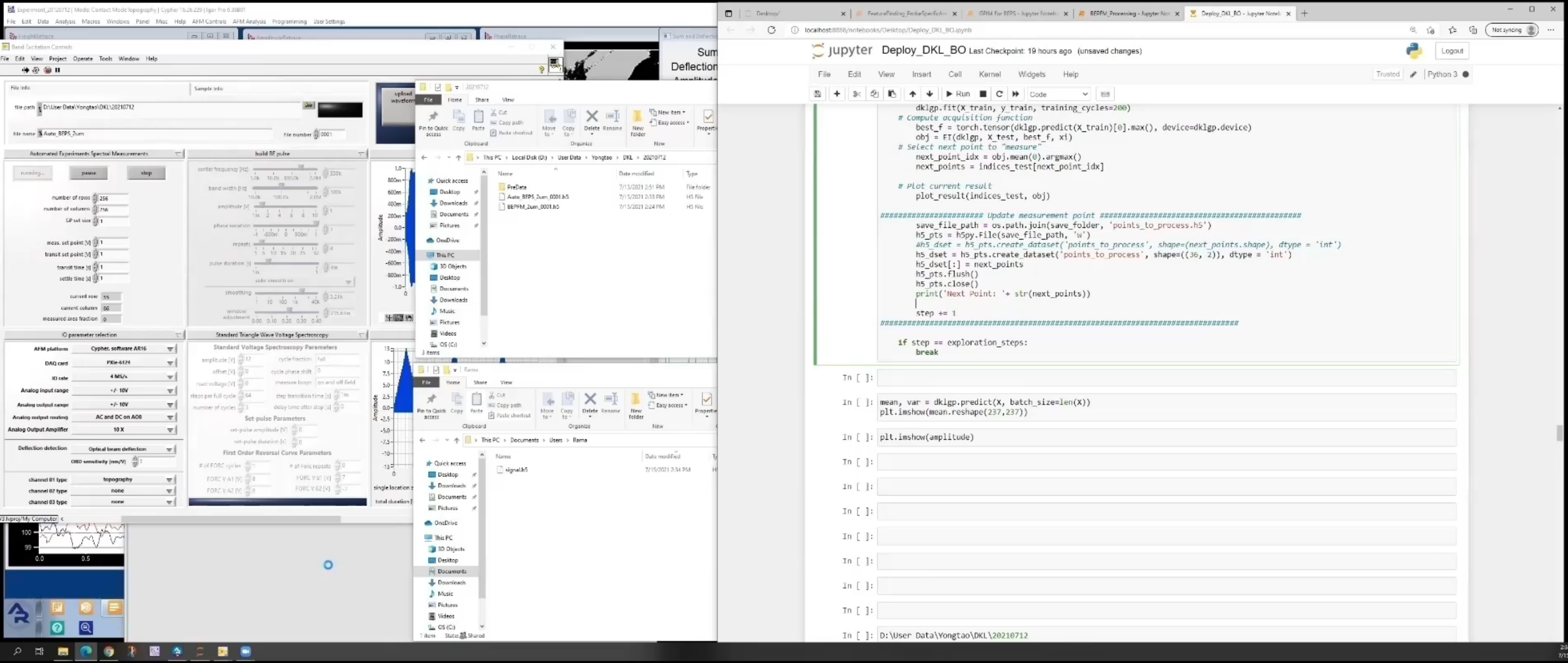
- High uncertainty @ start, but fairly quickly reduces
- Prediction actually doesn't drastically change throughout experiment
  - Structure-property relationship here is fairly rapidly learned
- Note the training can be halted after some criterion is met, making remainder of experiment go much quicker



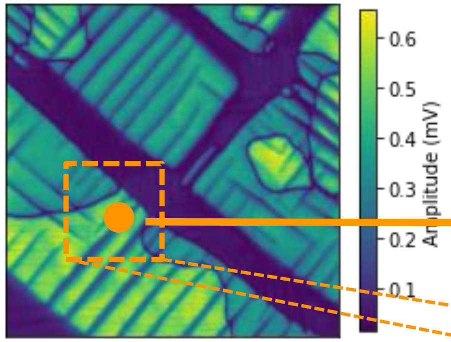
Does it always work?

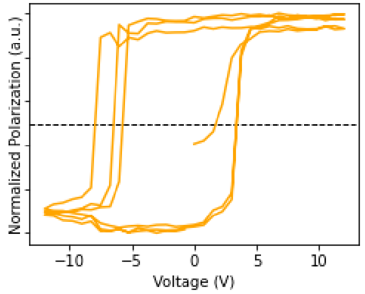


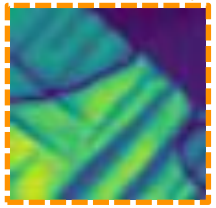
Deep Kernel Learning AE

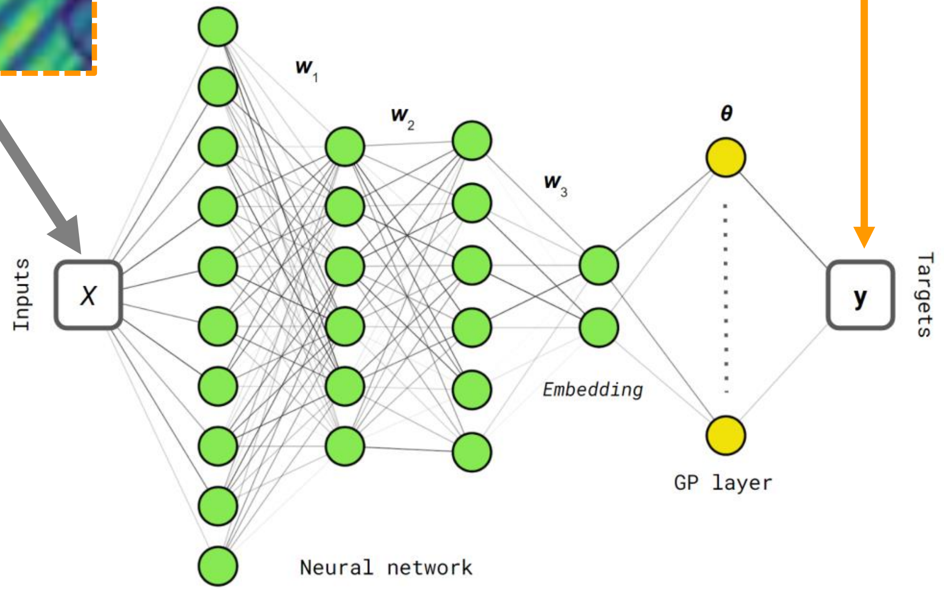


Deep Kernel Learning









- All patches are available in the beginning
- Spectra are made available sequentially
- We define what feature in spectrum are we interested in

<u>Specify physics criteria</u>

Acquire **structural data**

Measure a **spectrum**

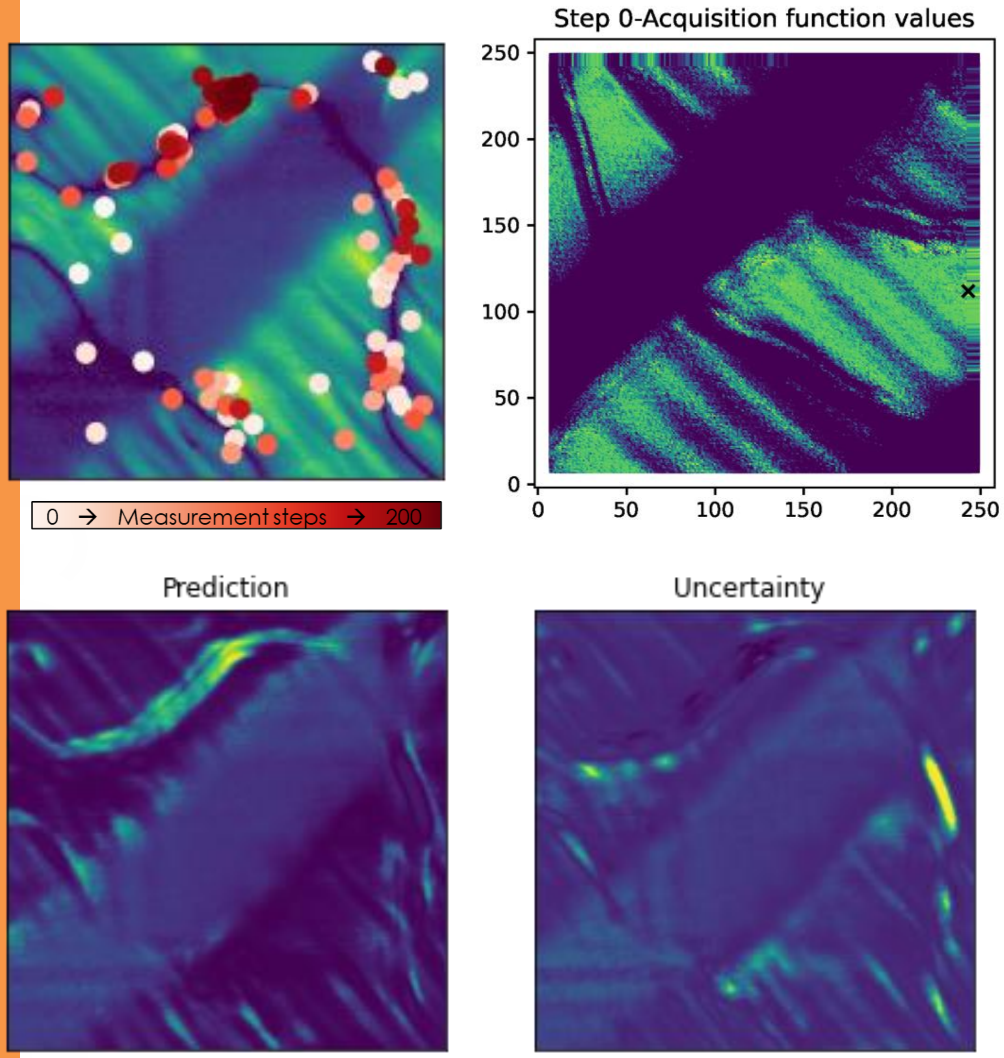
Train DKL model with new data

Decide next position (optimize physics criteria)

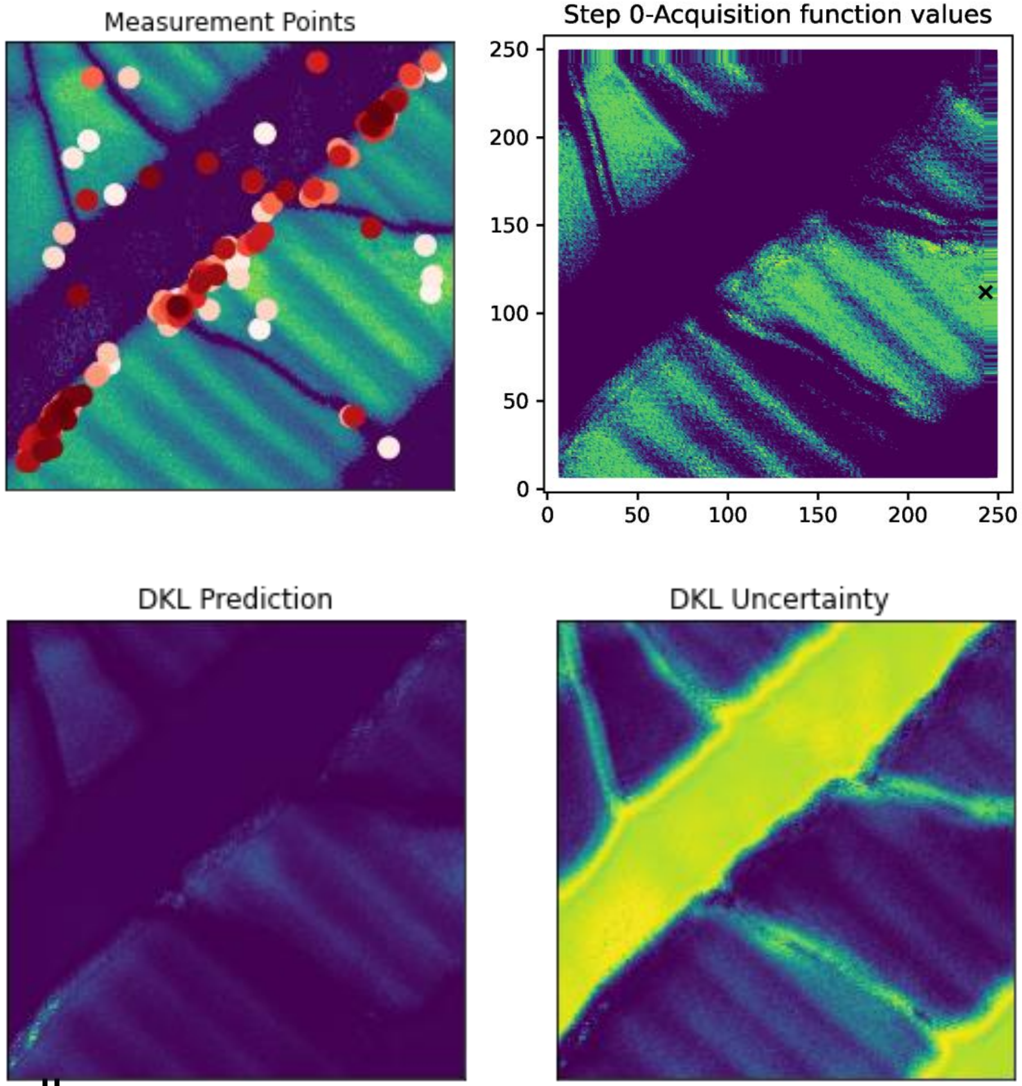
Allows navigation of the system to search for physics

Deep Kernel Learning SPM

Guided by: On field loop area



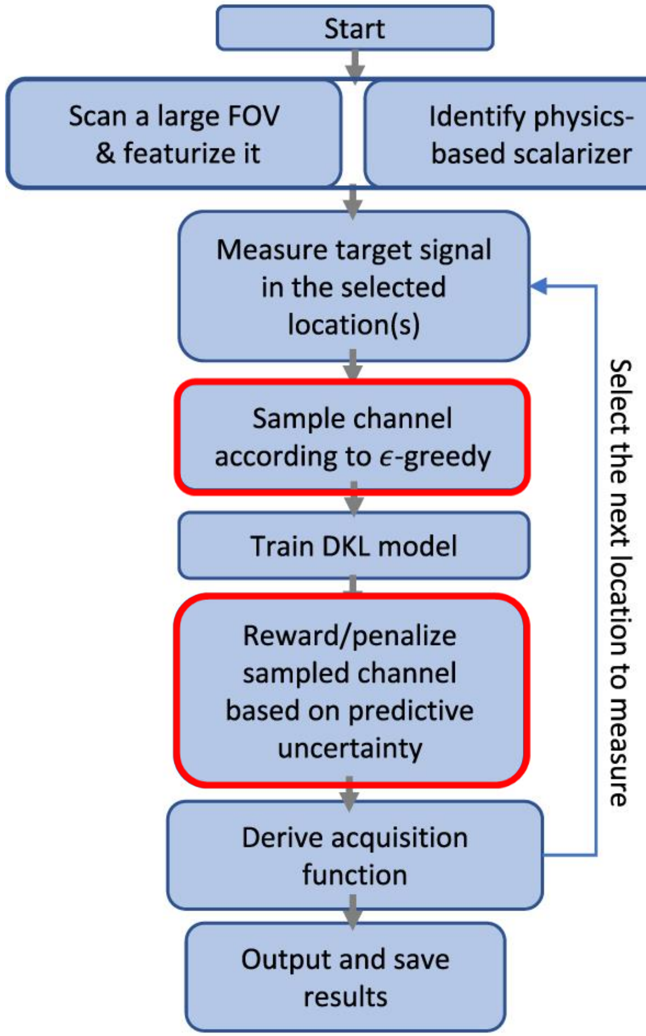
Guided by: Off field loop area

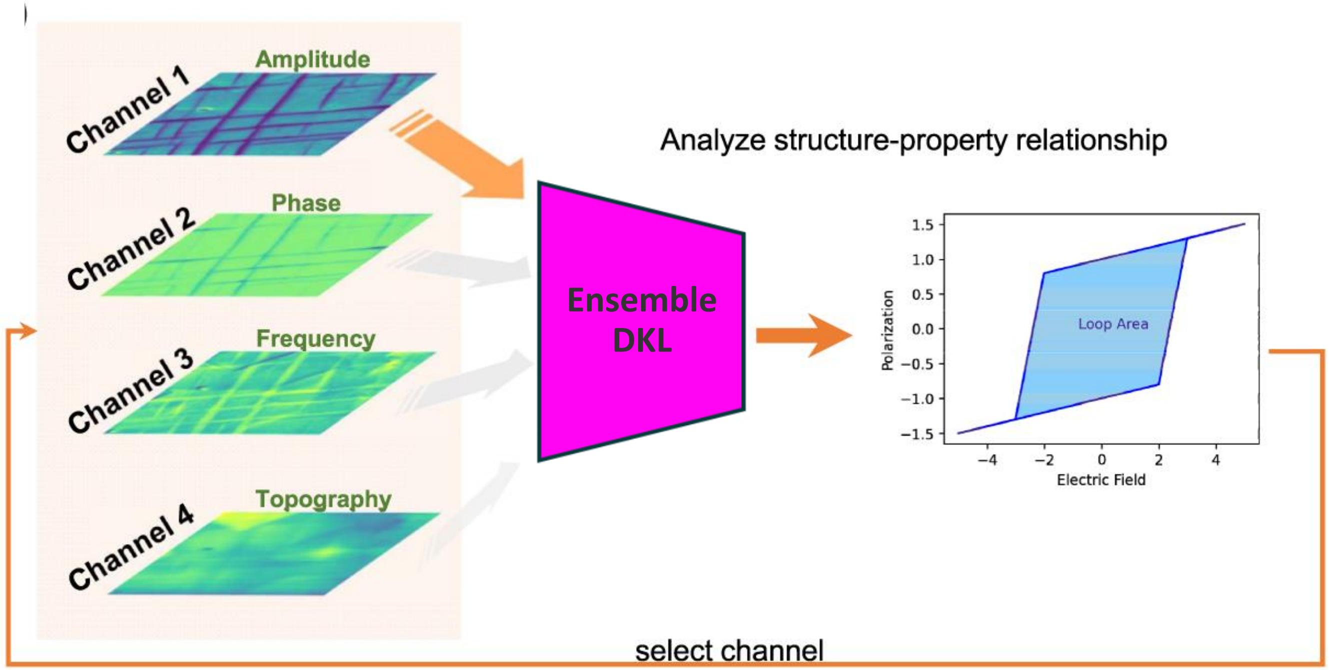


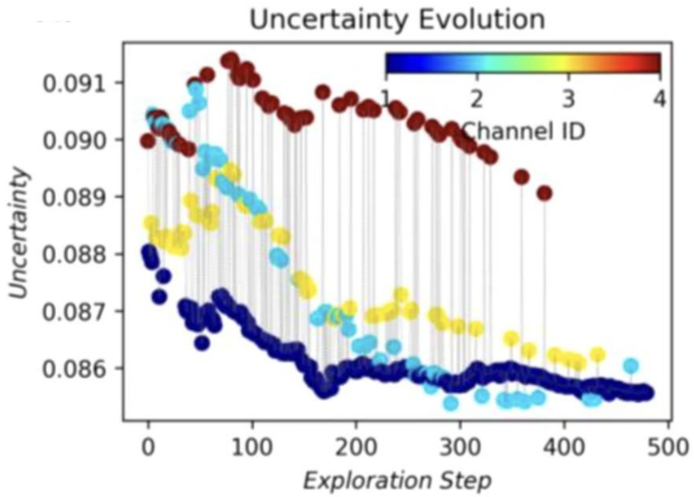
- Large loop opening corresponding 180º domain walls
- This behavior can be attributed to the large polarization mobility of 180º walls

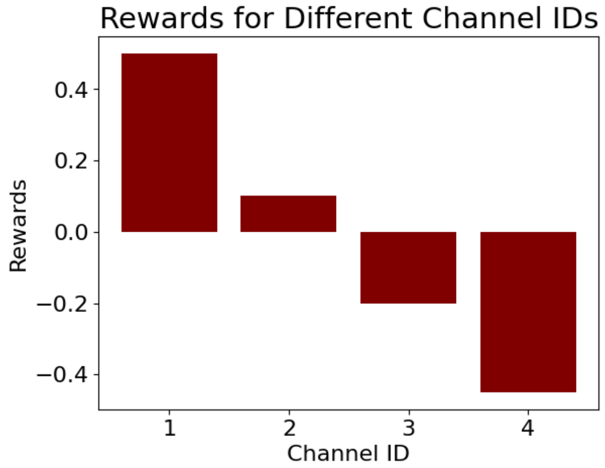
Liu, Yongtao, et al, Nature Machine Intelligence 4, 4 (2022): 341-350.

DKL: Learning the best channel



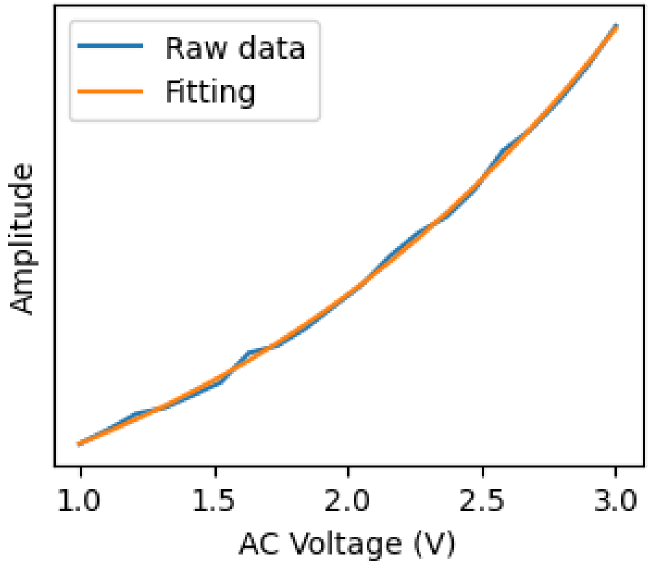


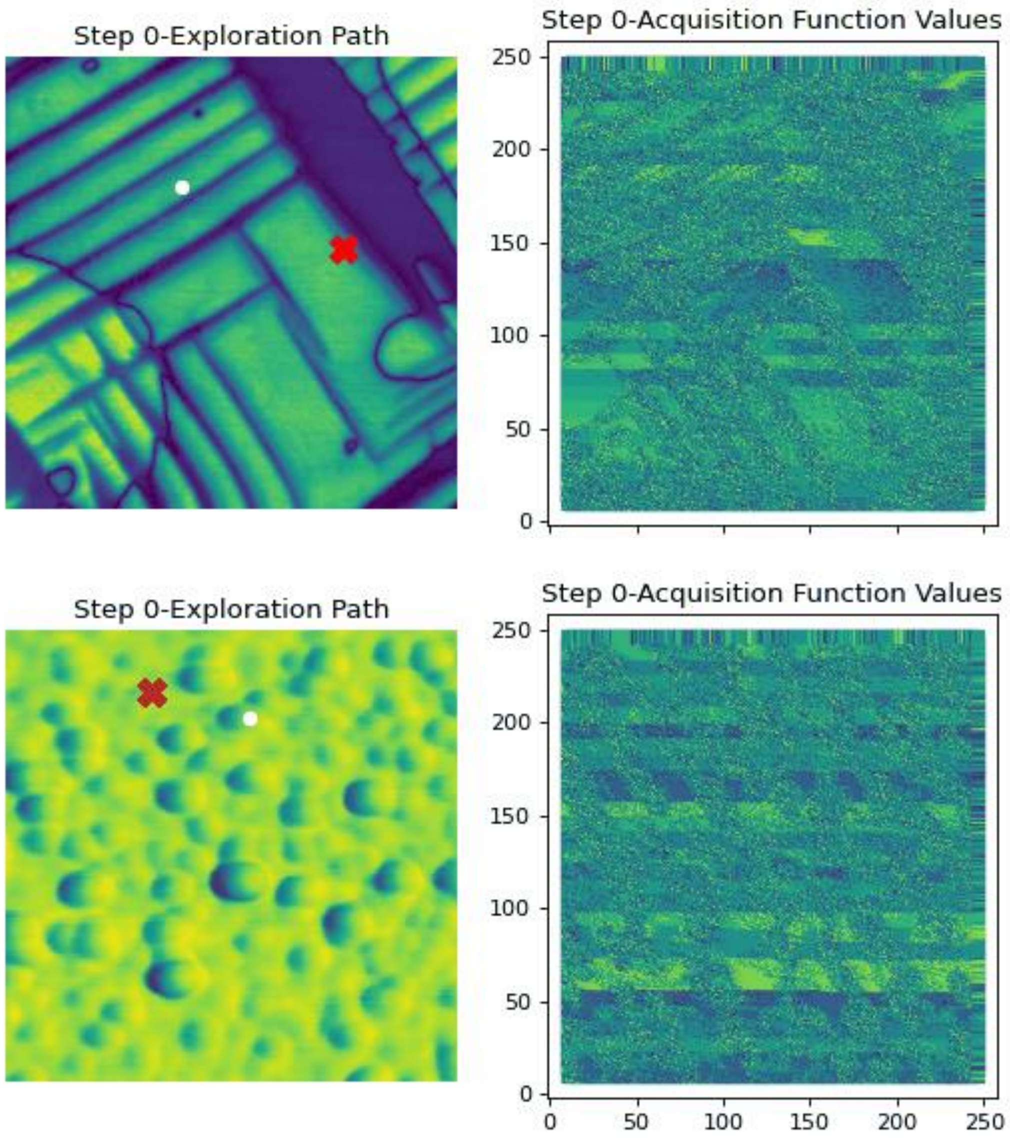




Y. Liu et al. npj Comput Mater **9**, 34 (2023).

Why human in the loop?





- 200-step automated experiment
- PFM amplitude was used as structure ima
- V_{AC} sweep curve at each location was fitte
  $y = Ax^3 + Bx^2 + Cx$
- A, B, C, and A/B were used as the target function to guide DKL-V_{AC} measurement.

The methodologies of classical ML (hyperparameter optimization, cross-validation) are rarely applicable for active learning!

- In conventional microscopy experiment, human runs everything directly – defines scan, positions the probe, defines measurement parameters.

- In AE SPM, the policies are defined before the experiment and do not change. Sometimes it works – but not always.

- How would we:
  - (a) explain the AE progression after the experiment and
  - (b) control it during the experiment ?

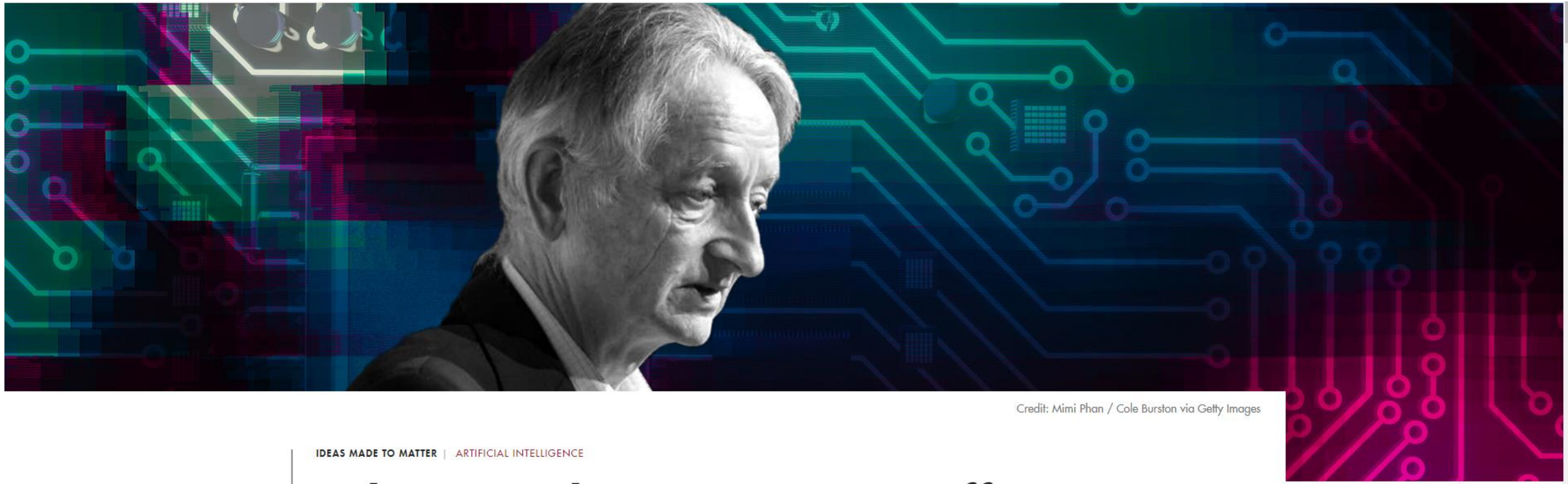
Taking the Human Out of the Loop: A Review of Bayesian Optimization

Citation

Shahriari, Bobak, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. “Taking the Human Out of the Loop: A Review of Bayesian Optimization.” Proc. IEEE 104 (1) (January): 148–175. doi:10.1109/jproc.2015.2494218.

Published Version

doi:10.1109/JPROC.2015.2494218

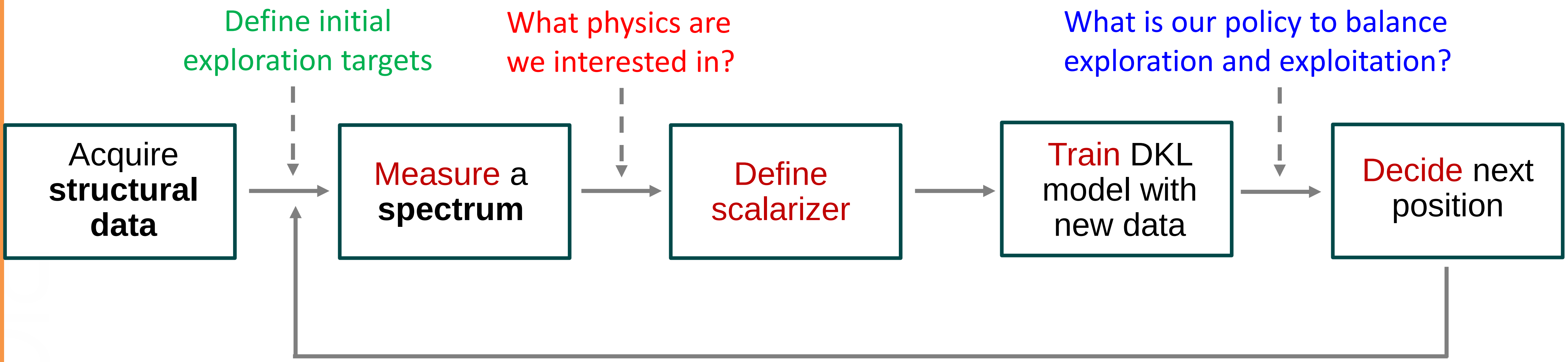


Credit: Mimi Phan / Cole Burston via Getty Images

IDEAS MADE TO MATTER | ARTIFICIAL INTELLIGENCE

Why neural net pioneer Geoffrey Hinton is sounding the alarm on AI

Bringing Human into the Loop



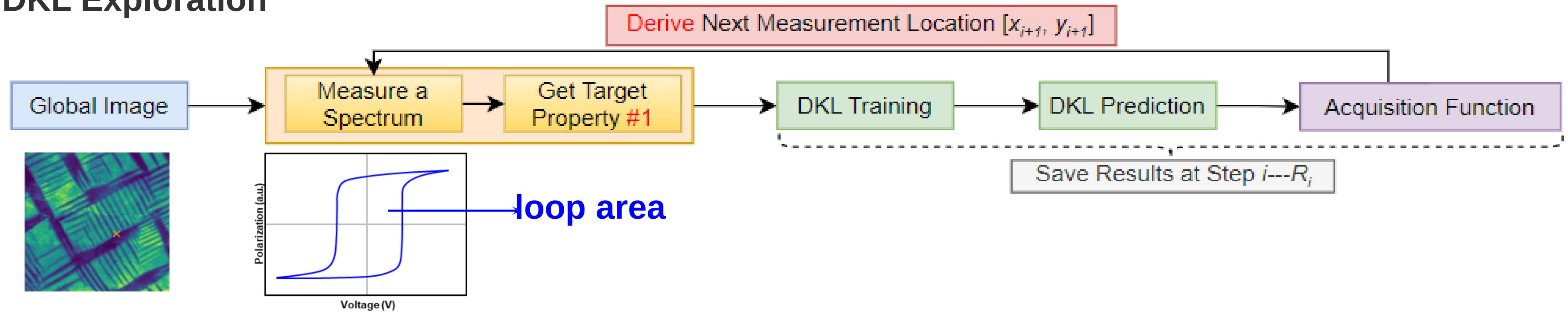
Key concepts:

- **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

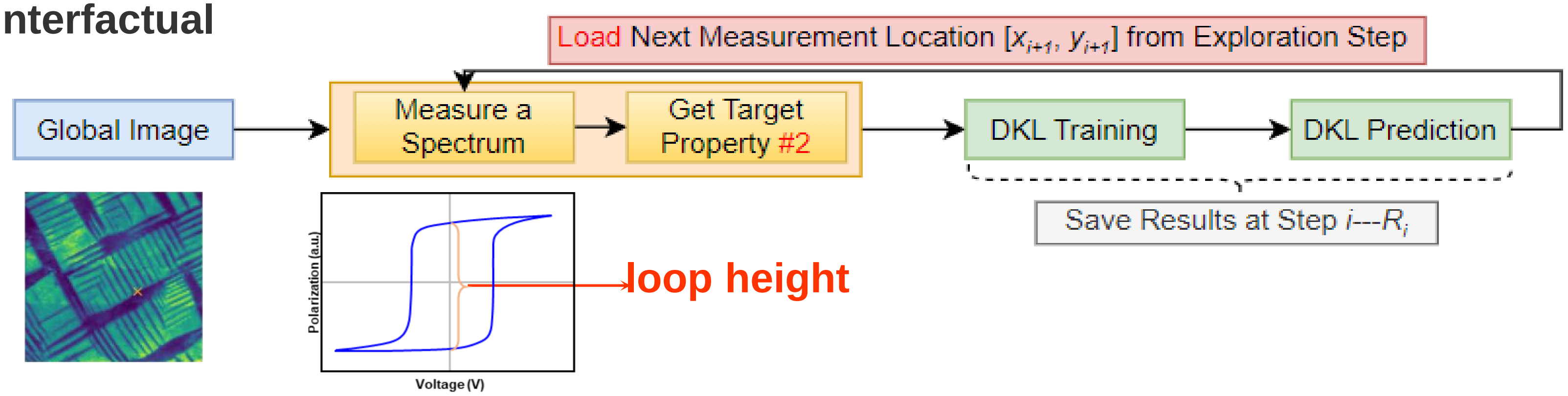
- **Experimental trace:** collection of image patches and associated spectra acquired during experiment. Note that we collect spectra, not only scalarizers

Counterfactual scalarizers

DKL Exploration



DKL Counterfactual



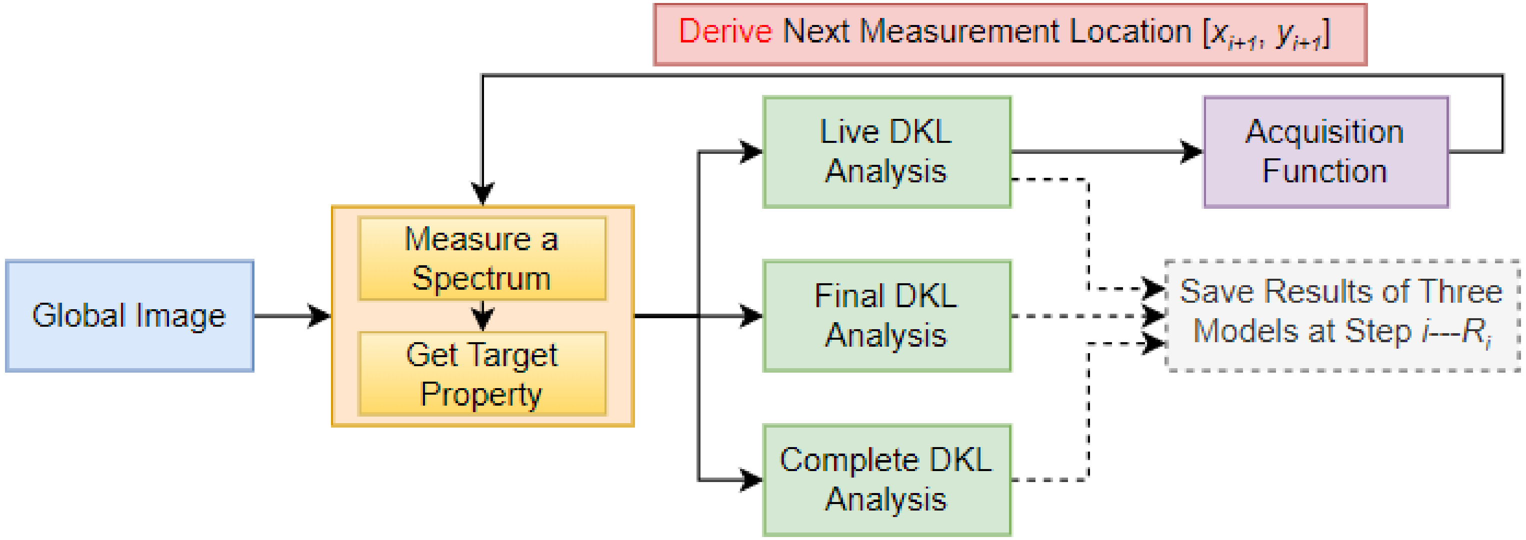
Target properties:

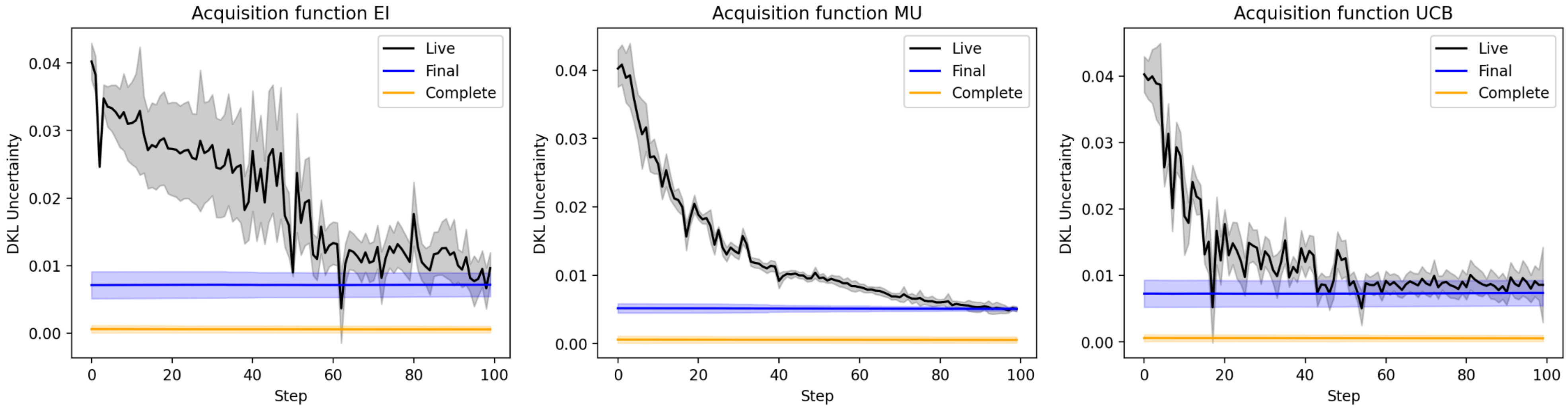
1. Loop Area
2. Loop Height
3. Coercive Field
4. …

- We save the full experimental trace
- What if we follow the actual experimental path – but calculate alternative (counterfactual) scalarizers?

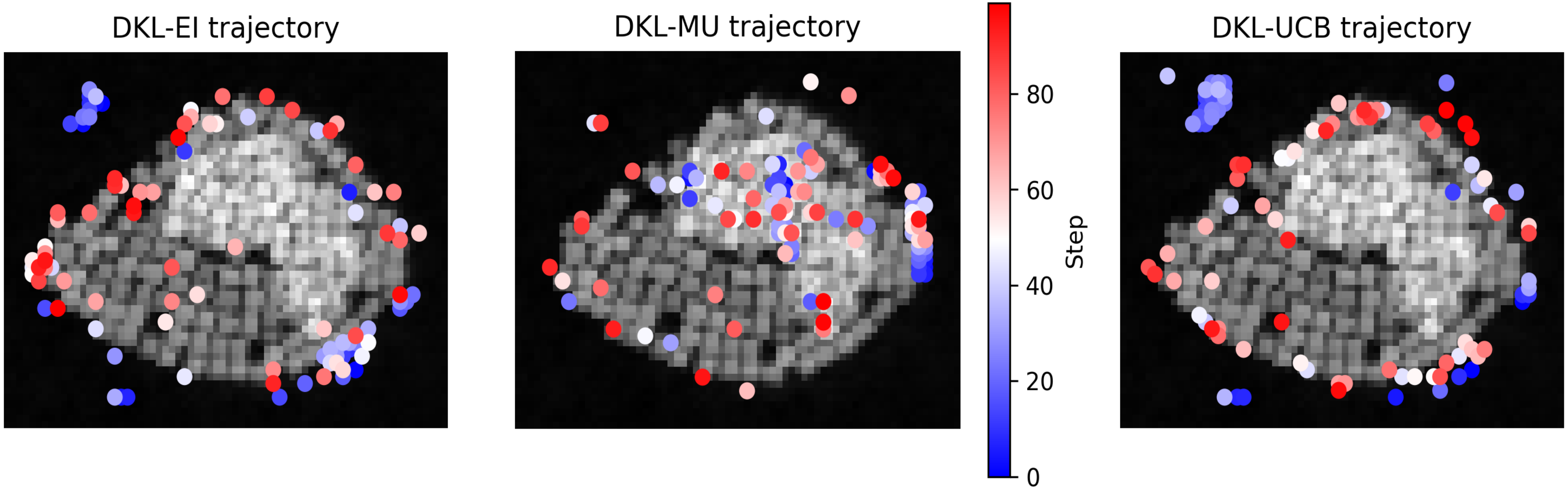
Explainable AE

- During the AE, model learns structure-property relationships.
- What if we retrace the experimental steps – using the fully trained model?



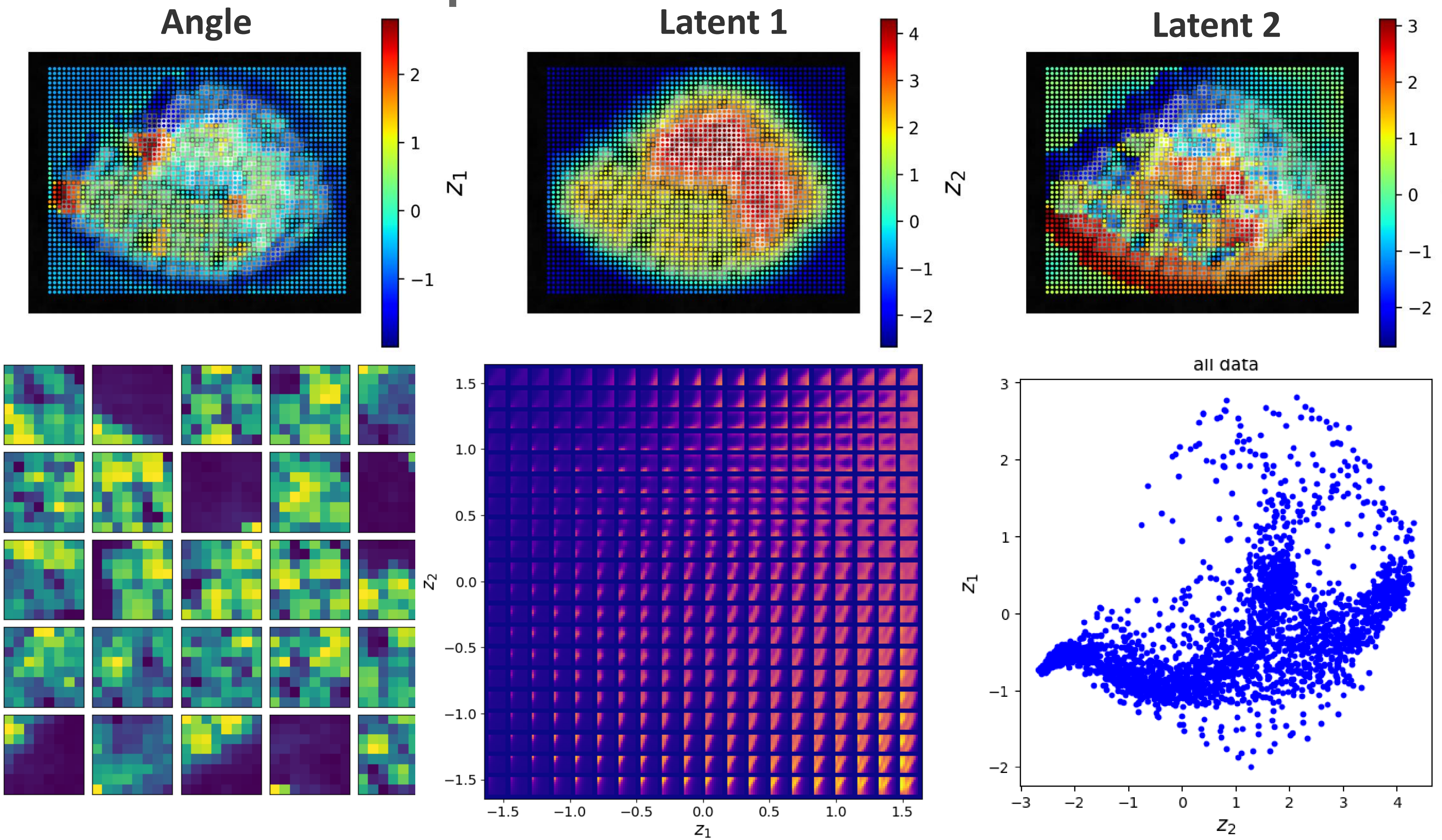


Monitoring the AE



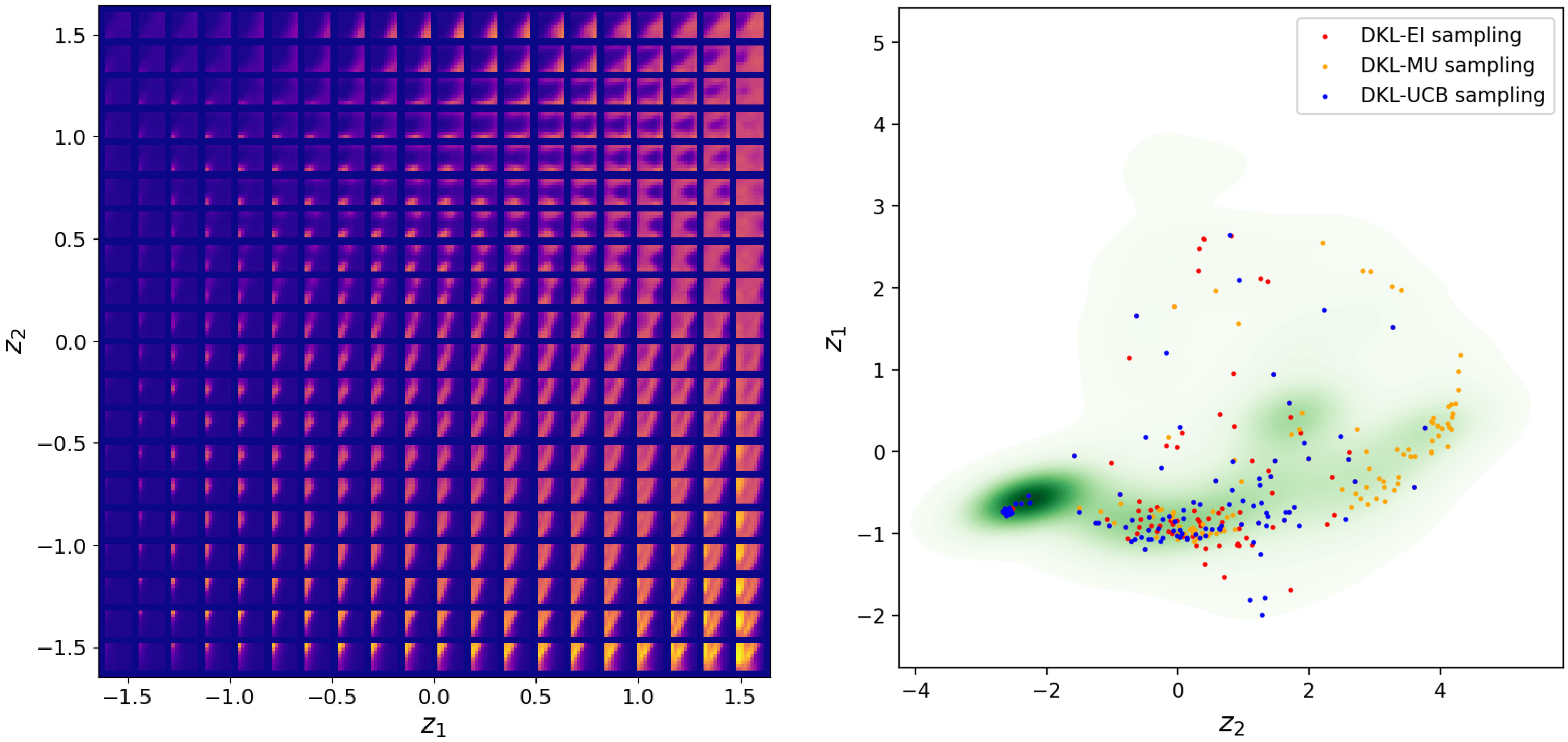
- Different acquisition functions (policies) give different experimental paths for AE
- Can we analyze what is special about points visited?

Global Feature Space

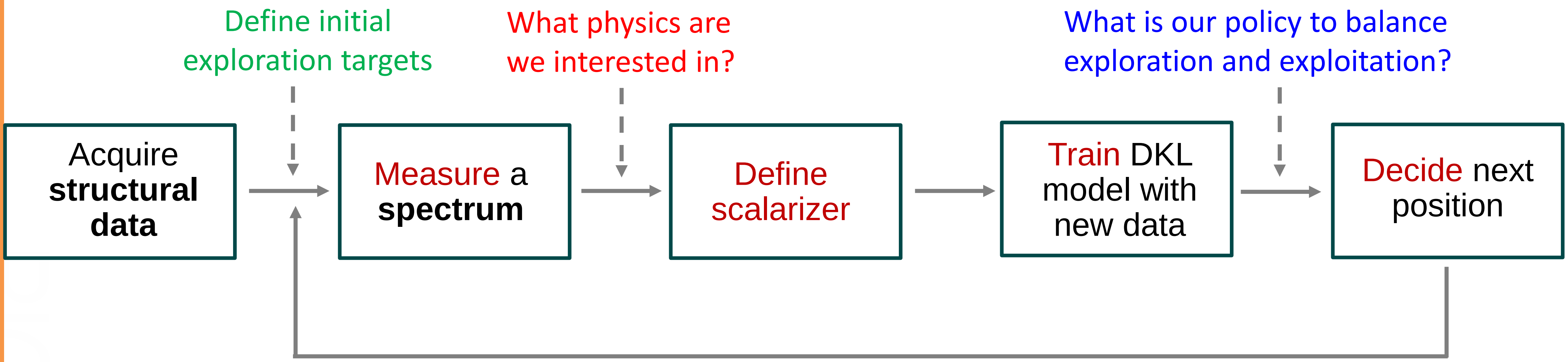


- Global feature space is available from the beginning of the experiment and is stationary
- Experimental trajectories can be visualized in the global feature space

VAE approach: full feature space



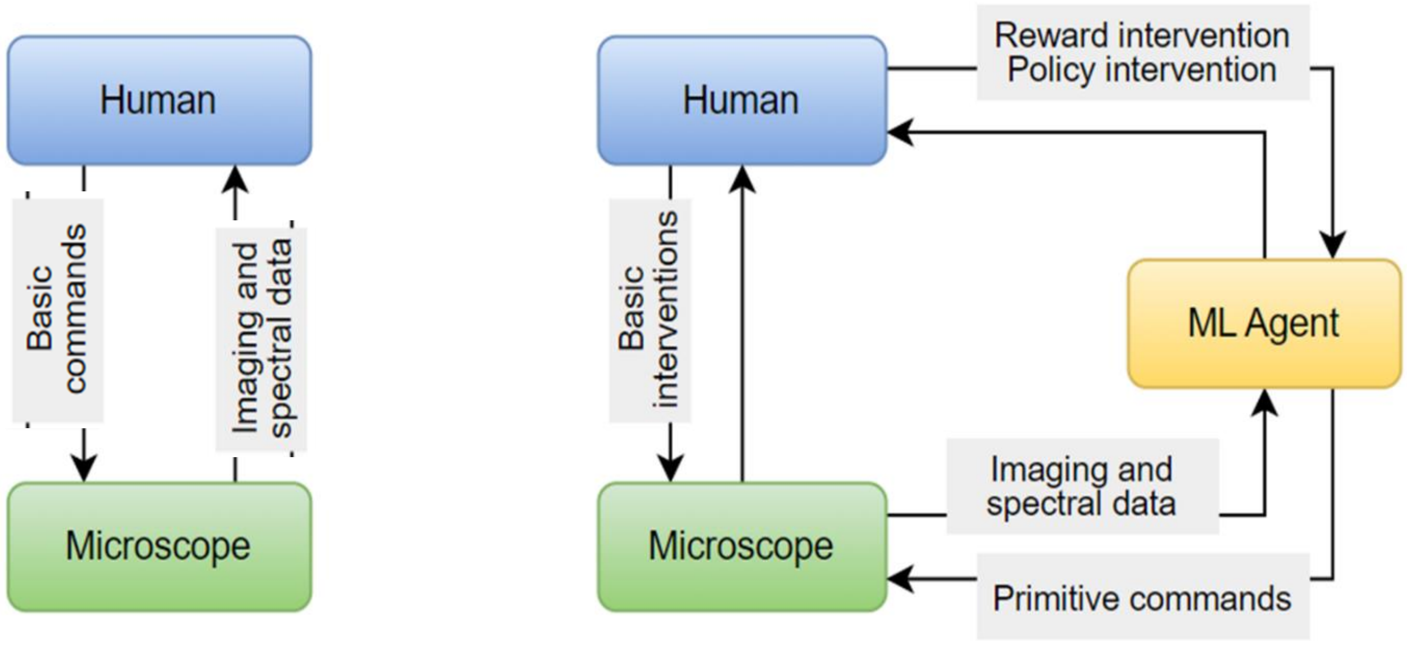
Bringing Human into the Loop

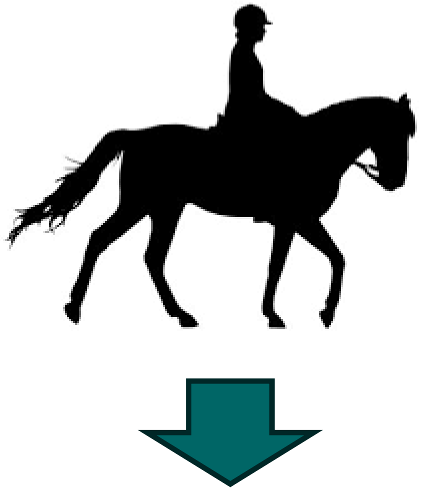


We can intervene on:

- **Policies** (acquisition functions): type and parameters
- **Scalarizers**: what physics are we interested in - type and parameters
- **Knowledge injection:** what microstructures are we interested in?
- **Cost and latencies:** trivial via acquisition functions

Human in the loop AE

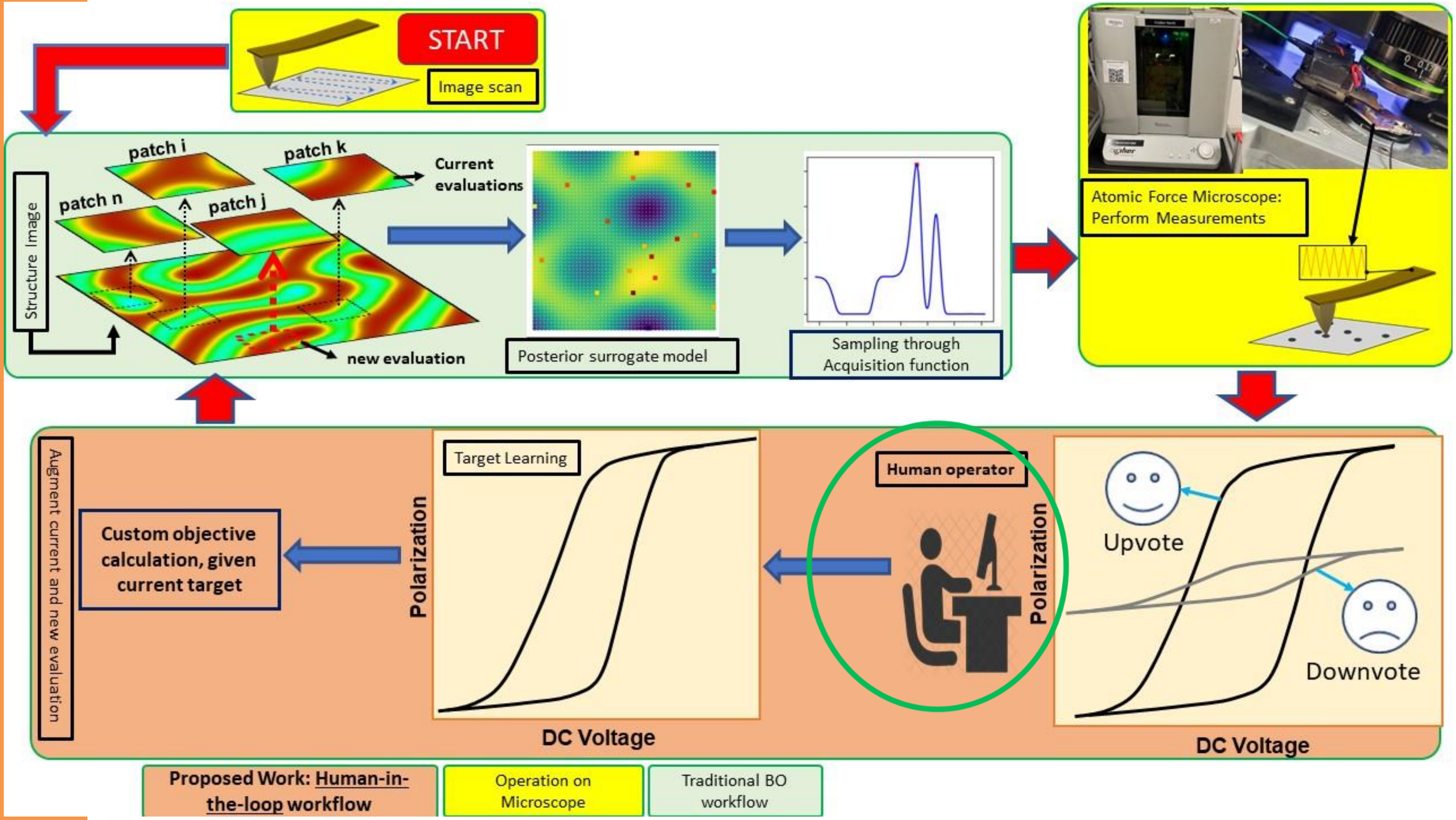




We can intervene on:

- Policies (acquisition functions): type and parameters
- Scalarizers (physics descriptors): type and parameters
- Knowledge injection
- Direct operation

BOARS: Human (partially) in the loop



Arpan Biswas

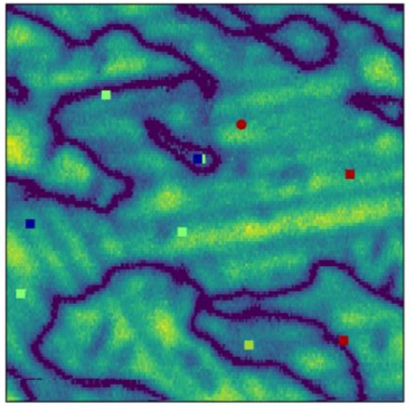
Bayesian optimized Active Recommender System (BOARS)

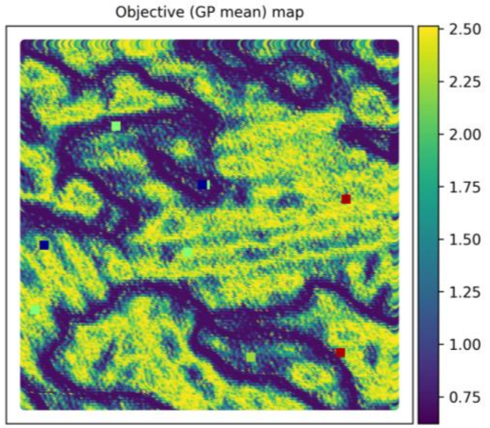
BOARS: Human partially in the loop

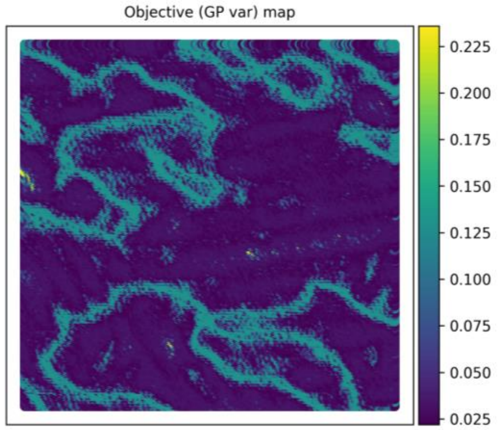
Human Assessment within loop

```
    return ampdat_masked, points_measured

# set parameters
IV = np.copy(amp_masked)
points_measured = np.array(idx)
last_points_measured = np.array(points_measured)
vdc = vdc_vec

train_Y = torch.empty((num_start, 1))
pref = torch.empty((num_start, 1))
init_spec = torch.empty((num_start, spec_length))
# Define a sparse grid to store evaluated spectral locations
eval_spec_y = torch.zeros(img.shape[0],img.shape[0],spec_length)
#Evaluate initial training data
x = torch.empty((1,2))

# First generate target loop, based on initial training data
wcount_good= 0
target_func = torch.zeros(spec_length)

mask = np.isin(points_measured, last_points_measured, invert = True)
new_points_measured = points_measured[mask]
last_points_measured = np.append(last_points_measured, new_points_measured)

for i in range(0, num_start):

    ##******************experiment start******************##
    time.sleep(0.1)
```

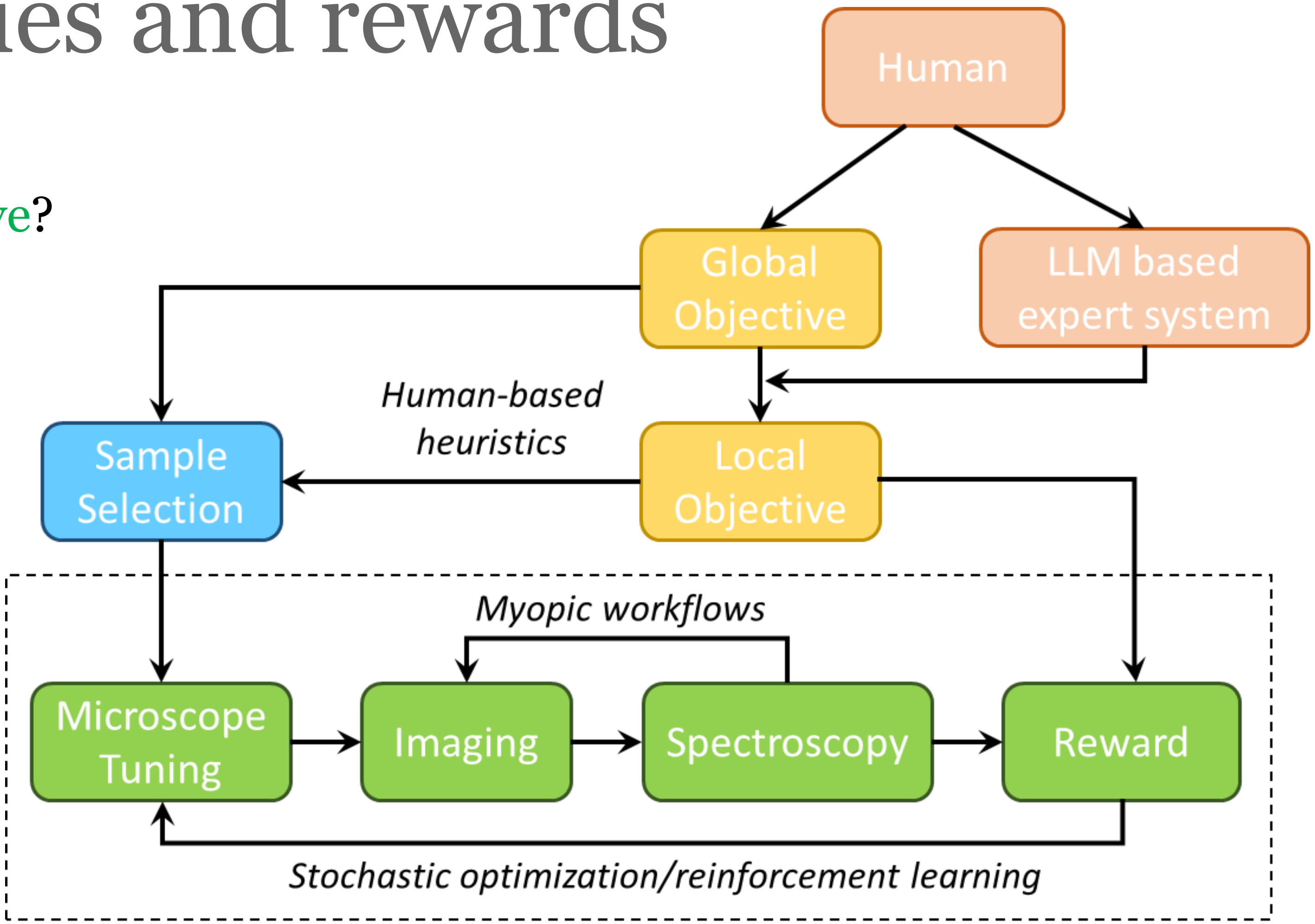
An implementation to SPM







The dance of policies and rewards



Rewards and objectives:

- What is our (hierarchical) objective?
- Can we define reward(s)?

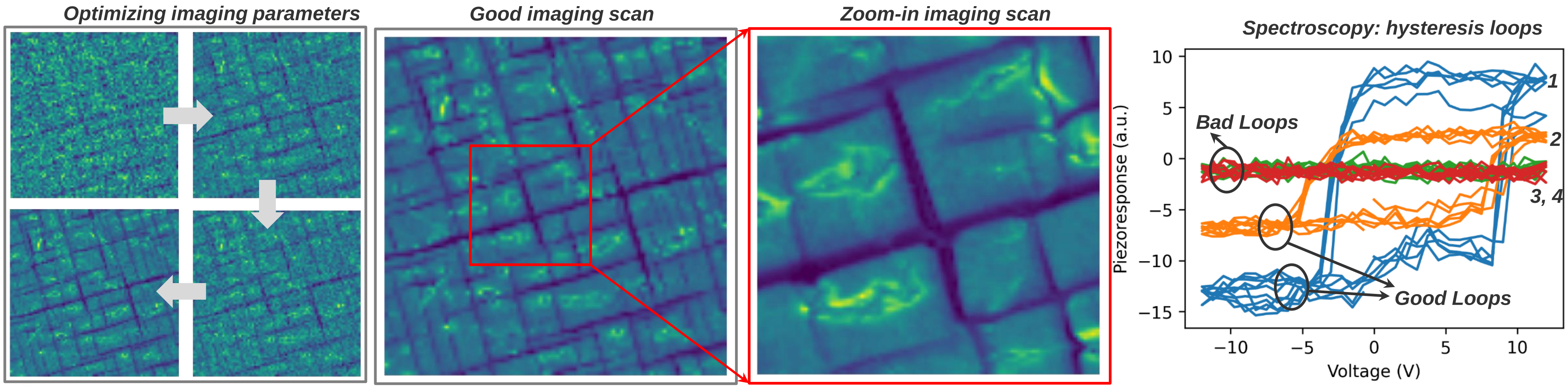
Inferential biases:

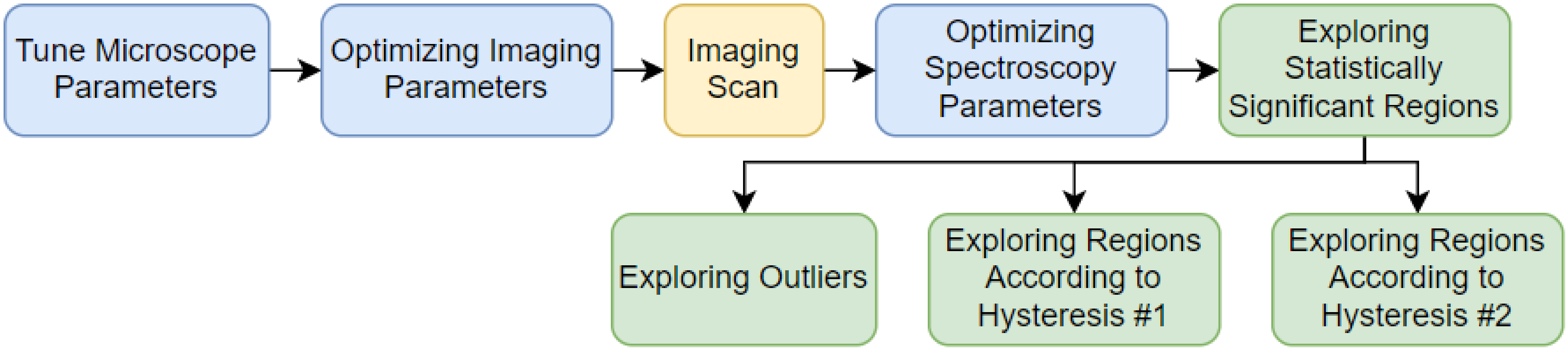
- What do we know before the experiment?
- What do we (hope to) learn after the experiment?

Experiment planning – policies and values

- How do we plan experiment in advance (policies or values based on rewards)?
- Can we ascribe value to certain steps?
- Do we change our policies during experiment?

Future: full workflow optimization





Characteristic	**Definition**	**Availability**
Global image	Initial structural data set available before DKL experiment. Used to create patches for DKL training	Before
DKL latents	The latent variables encoding the structural information in the patches	During**
Scalarizer function	Function defining what characteristic of spectrum guides Bayesian Optimization	Before*
Acquisition function	Function combining DKL prediction and uncertainty of the scalarizer function	Before*
Policy	Principle for selection of next path. Simplest policy is maximization of acquisition function, but can be more complex including epsilon-greedy or switch between multiple scalarizers or acquisition functions. Human in the loop intervention tunes some aspect of the policy	Before*
Experimental trace	Collection of patches (and their coordinates) and spectra derived during experiment. Trace and global image are the results of AE SPM.	During

Characteristic	**Definition**	**Availability**
Live DKL model	DKL model in the state corresponding to the *n*-th experimental step	During
Final DKL model	DKL model in the state corresponding to the end of the experiment	After
Complete DKL model	DKL model trained on the full data set (if available from grid measurements, etc).	
Regret analysis	The difference between predictions of live DKL model and final DKL model after the whole experiment (i.e., after 200 steps in this work)	During** and After
Learning curve	Change of the DKL uncertainty (mean and deviation), indicative of the predictability of the patch-scalarizer relationship	During
Counterfactual scalarizer	The availability of full spectral data as a part of experimental trace allows to estimate what the BO step would be if scalarizer were chosen to be different	During
Trajectory analysis	Real-time trajectory of the probe that can be represented in the global image plane	During
Feature discovery	Analysis of the latent variables and latent representations of image patches and spectra in the trace. Here, we realize only patch analysis but extension to spectra is straightforward.	After
Latent trajectory analysis	Analysis of the experimental trajectory in the latent space of the full collection of the image patches derived from the global image	During**