Machine Learning and Automated Experiment

1. Machine learning is now everywhere…. But sometimes it is difficult to see it in the lab!

2. Applying ML in experimental sciences can be a very daunting proposition:
   a. Understand the problem
   b. Know (some) ML
   c. Develop code: from prototype to operationalization
   d. Implement on the working instrument
   e. Connect to the cloud
   f. Understand the results

3. The code base and libraries can change (TF/Keras -> PyTorch -> Jax -> ?)

4. Basic concepts are often incomprehensible

Therefore…

1. We are not going to try to learn all aspects of it. ML in domain/experiment is always a teamwork.

2. The first step is to define your problem – what is that you want to accomplish?

3. The second step is define your hyper language – what are the things that you know how to do (or can learn to do)?

4. The third step is to work backward from your problem and define the workflow in terms of your hyper language. Do you even need machine learning to solve it?

5. For ML, knowing code is (in some sense) secondary. However, it is critical to understand HOW it works.

What have we learned from lectures 1 and 2:

Lecture 1:

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

Lecture 2:

- Bayesian Optimization
- Bayesian Optimization based on Gaussian Process
- Acquisition Functions

What have we learned from Lectures 3 and 4

Lecture 3:

- Bayesian Inference
- Prior Distribution
- Posterior Distribution
- Least Square Fit
- WAIC

Lecture 4:

- Structured Gaussian Process
- Prior Distributions (kernel, parameters, noise)
- Posterior Distributions (kernel, parameters, noise)
- sGP based BO

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 system's 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

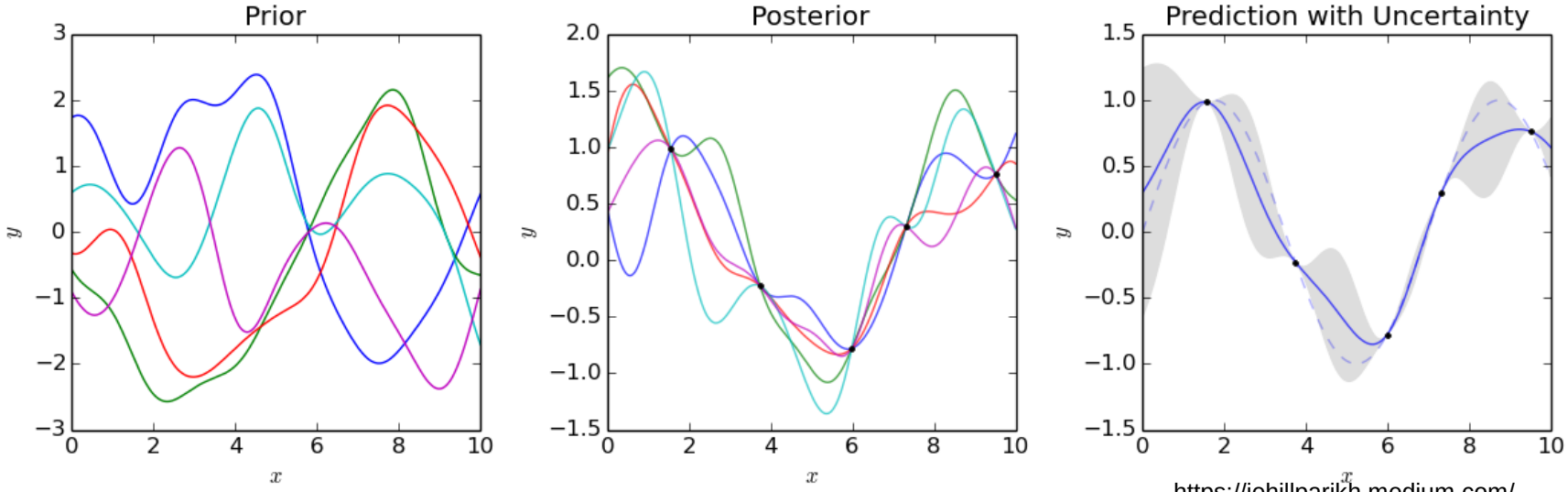
$$\mu_{\boldsymbol{\theta}^i}^{\text{post}} = \mathbf{m}(X_*) + \mathbf{K}(X_*, X|\boldsymbol{\theta}^i)\mathbf{K}\left(X, X\middle|\boldsymbol{\theta}^i\right)^{-1}(\mathbf{y} - \mathbf{m}(X)) \rightarrow \mu_{\boldsymbol{\Omega}^i}^{\text{post}} = \mathbf{m}\left(X_*|\phi^i\right) + \mathbf{K}(X_*,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

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

$\boldsymbol{\Omega}^i = \{\phi^i, \boldsymbol{\theta}^i\}$ 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)



https://jehillparikh.medium.com/

Probabilistic model

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

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

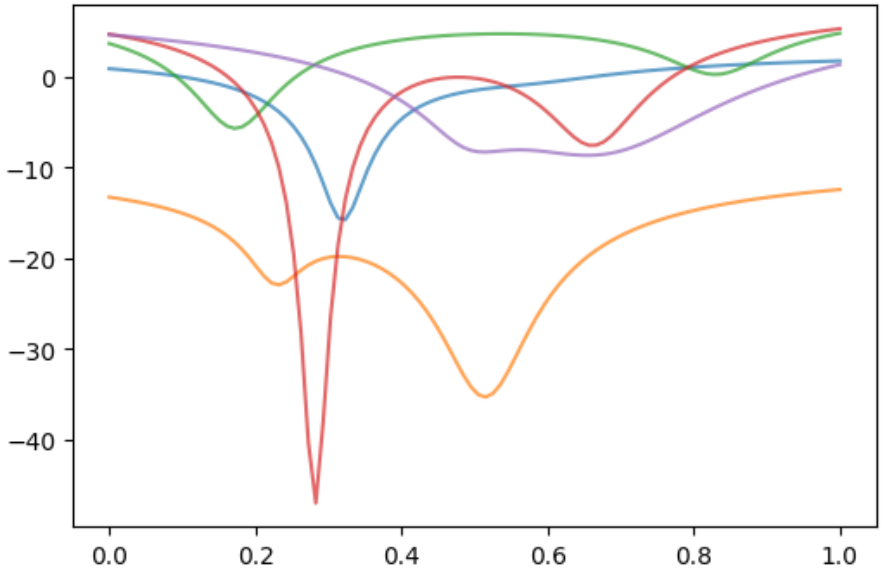
$$L_n \sim \frac{A_n}{\sqrt{(x - x_n^0)^2 + w_n^2)}}$$

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

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

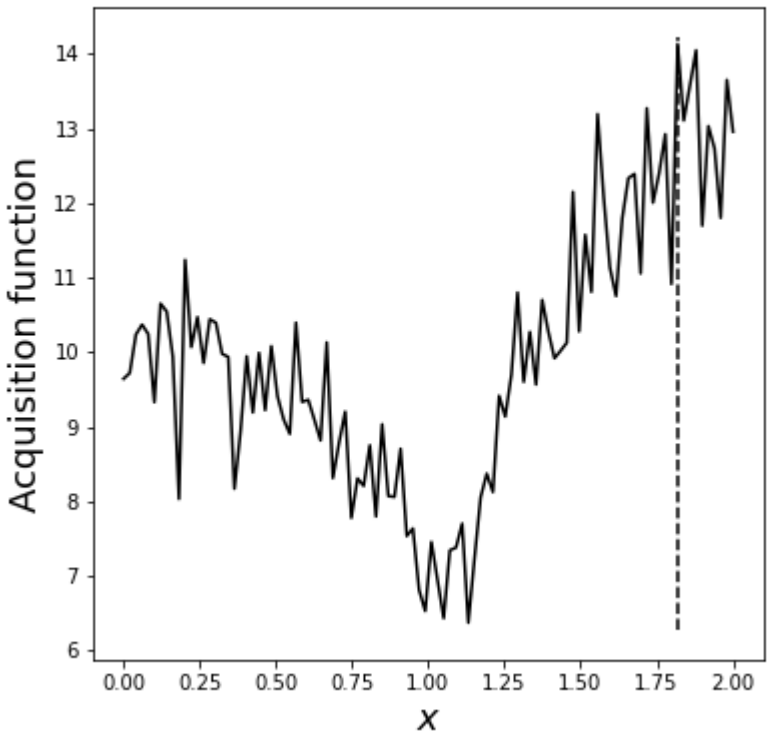
$$x_n^0 \sim Uniform(0, 1)$$

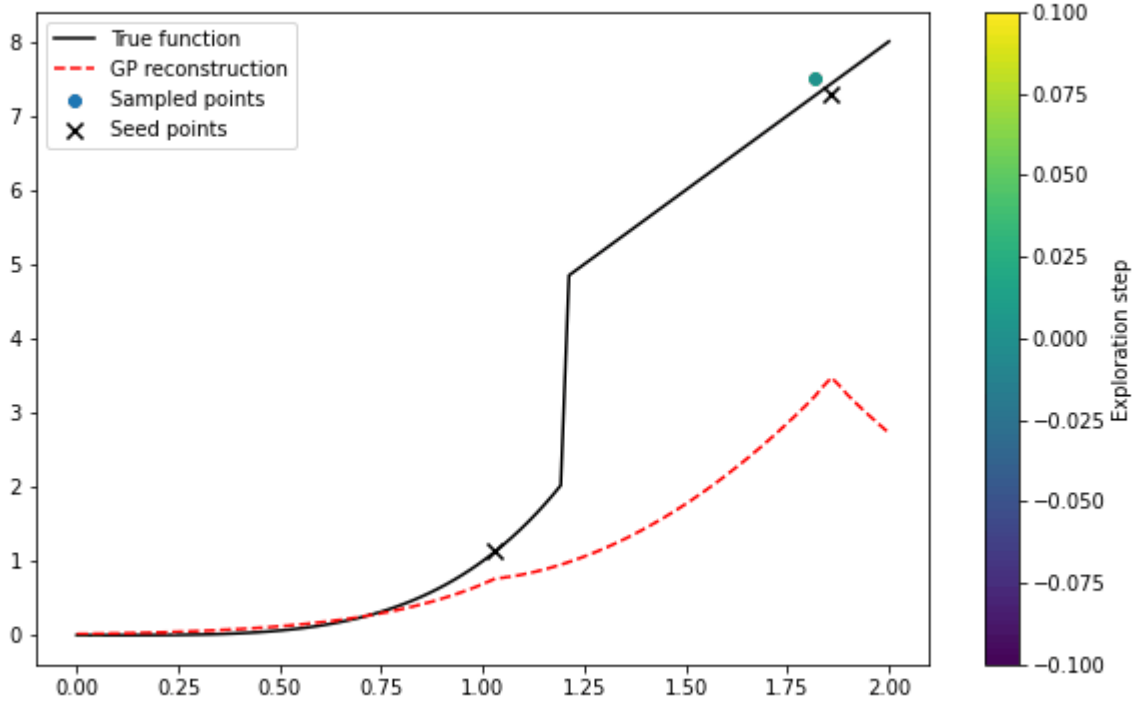
Prior predictive distribution



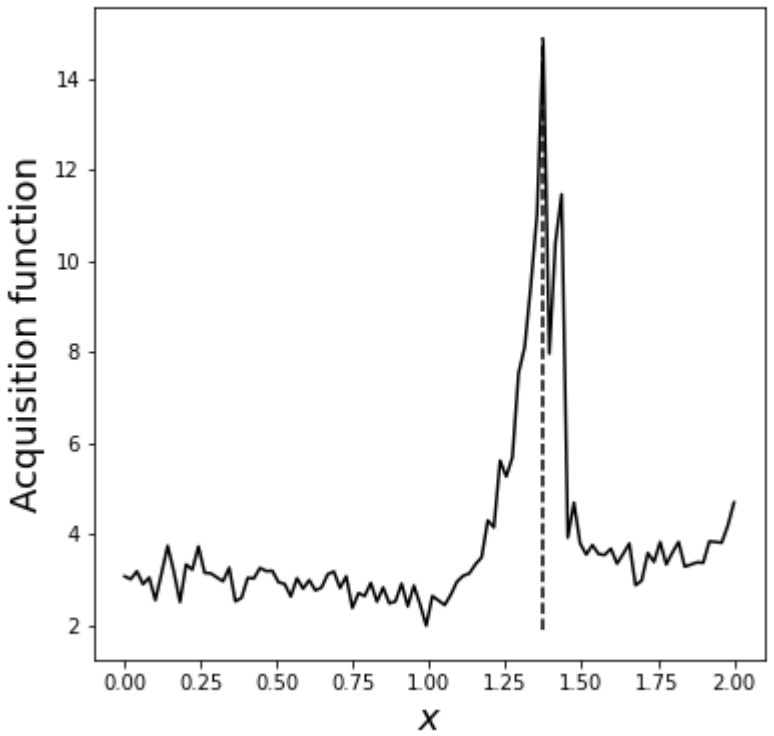
This model simply tells us that there are two minima in our data but does not assume to have any prior knowledge about their relative depth, width, or distance

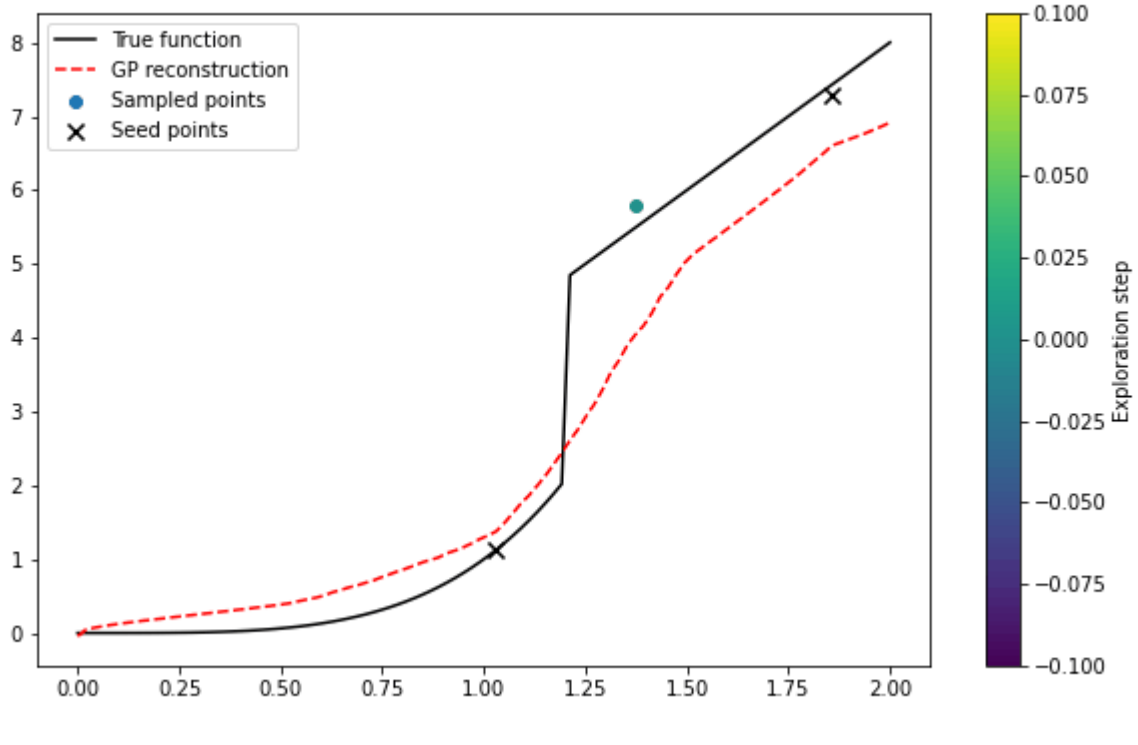
Simple GP search





Structured GP search





Application to Ising model

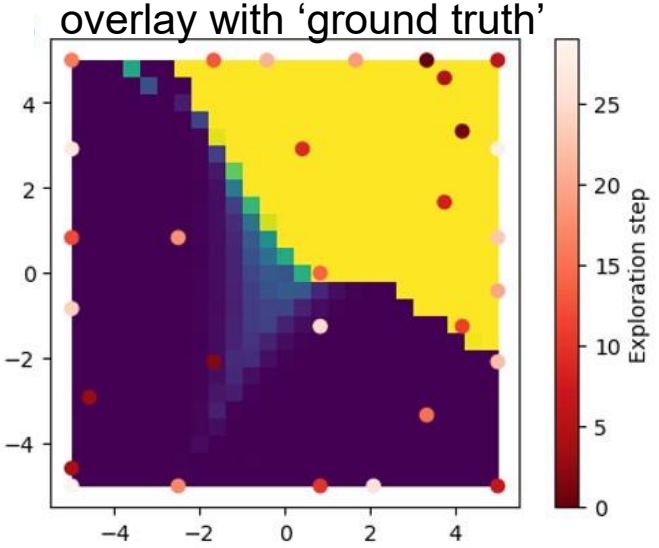
Probabilistic model

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

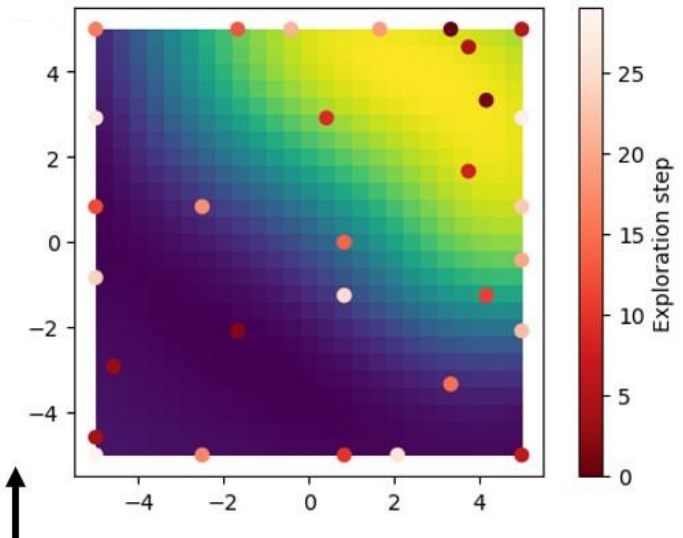
where $f(J)$ is a third-degree polynomial with normal priors on its parameters

Vanilla GP-BO

overlay with 'ground truth'

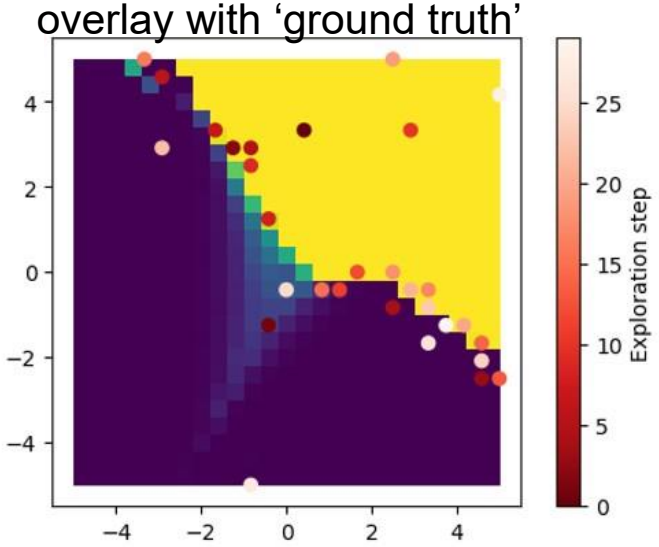


overlay with prediction

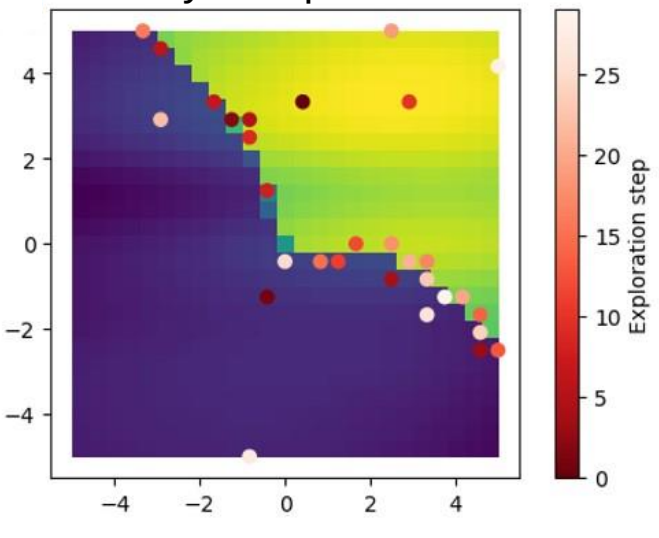


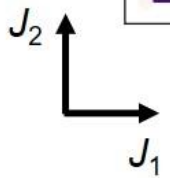
sGP-BO

overlay with 'ground truth'

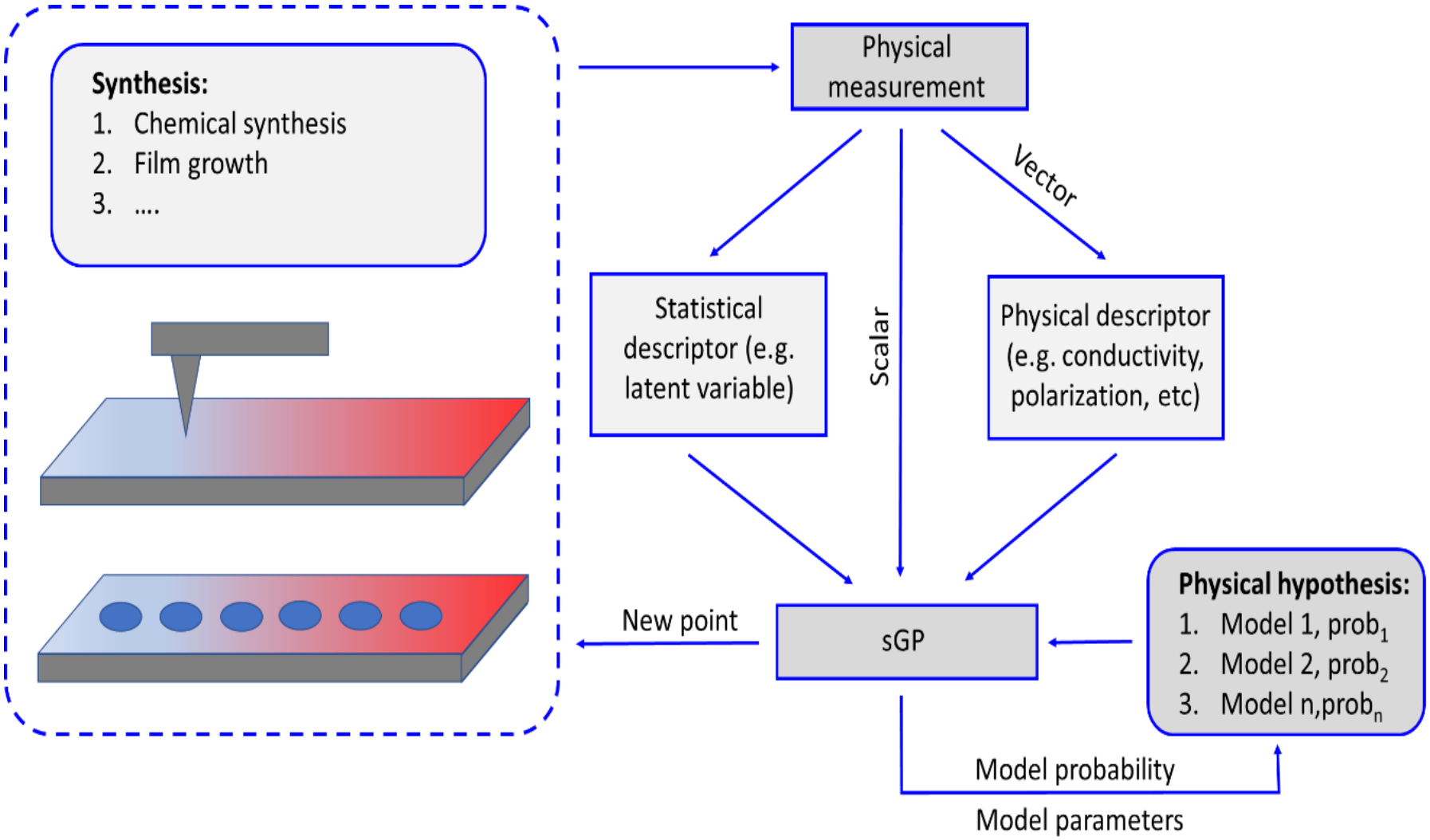


overlay with prediction

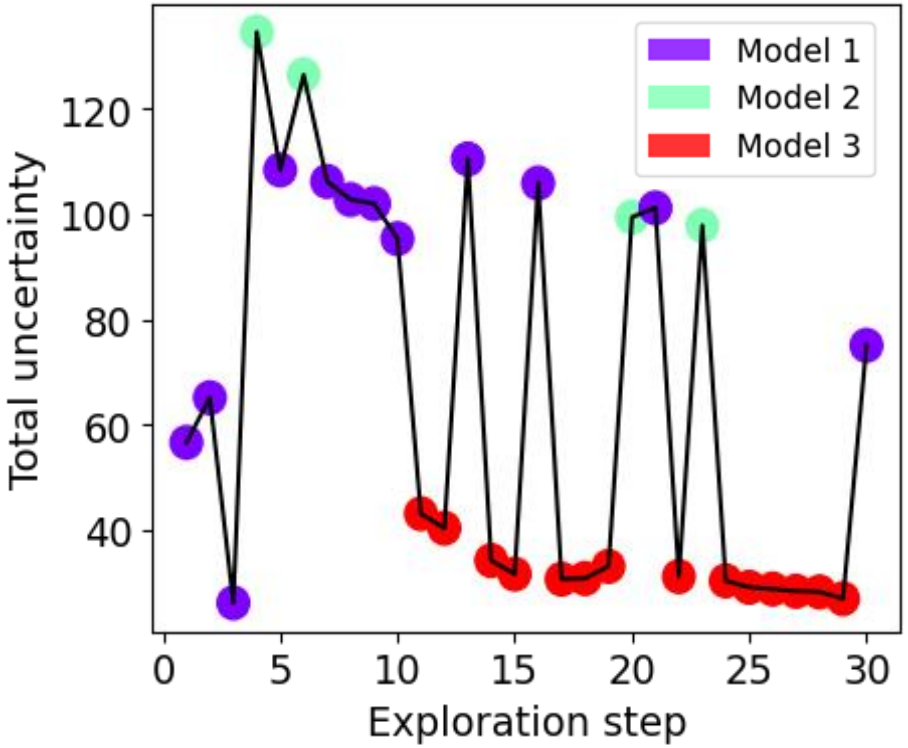


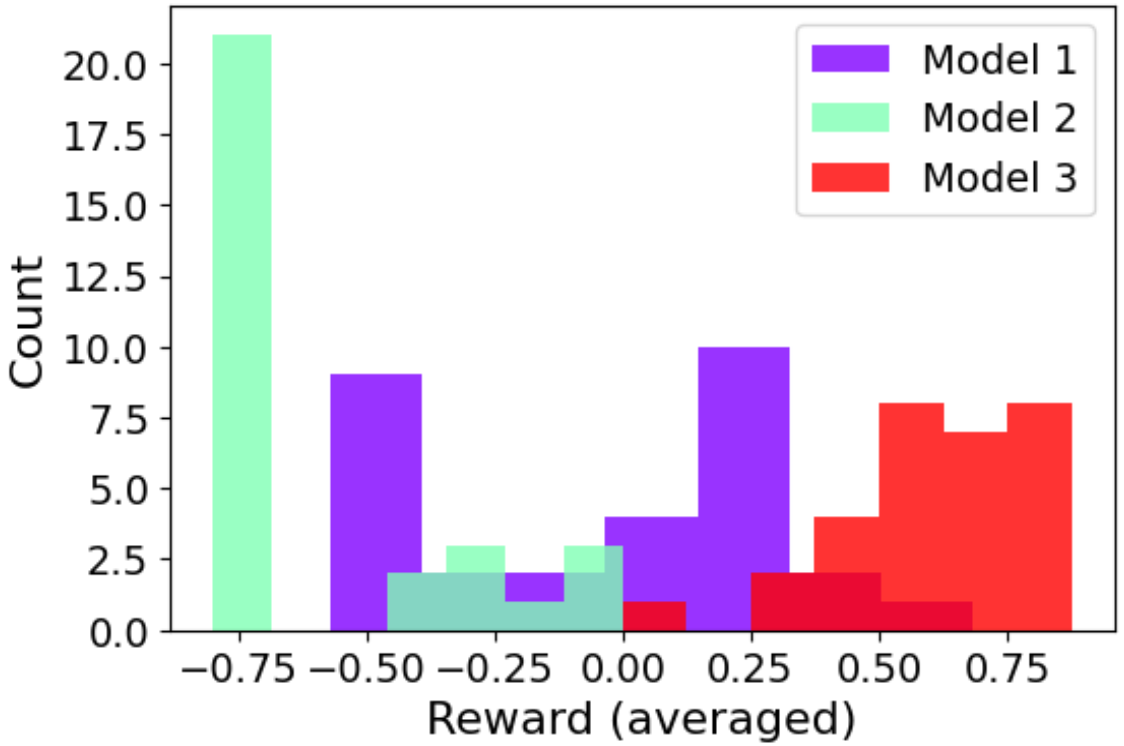


Hypothesis active learning: hypoAL



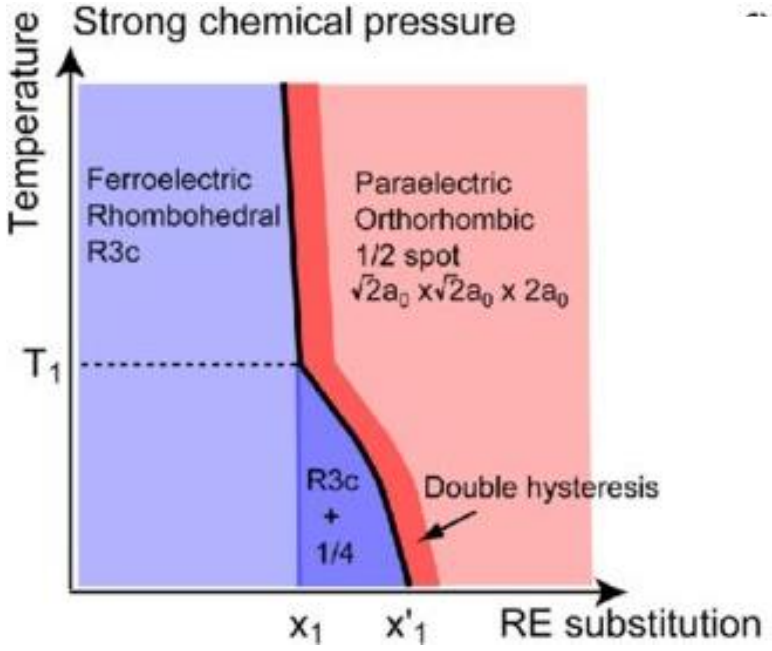
Next step: active model selection

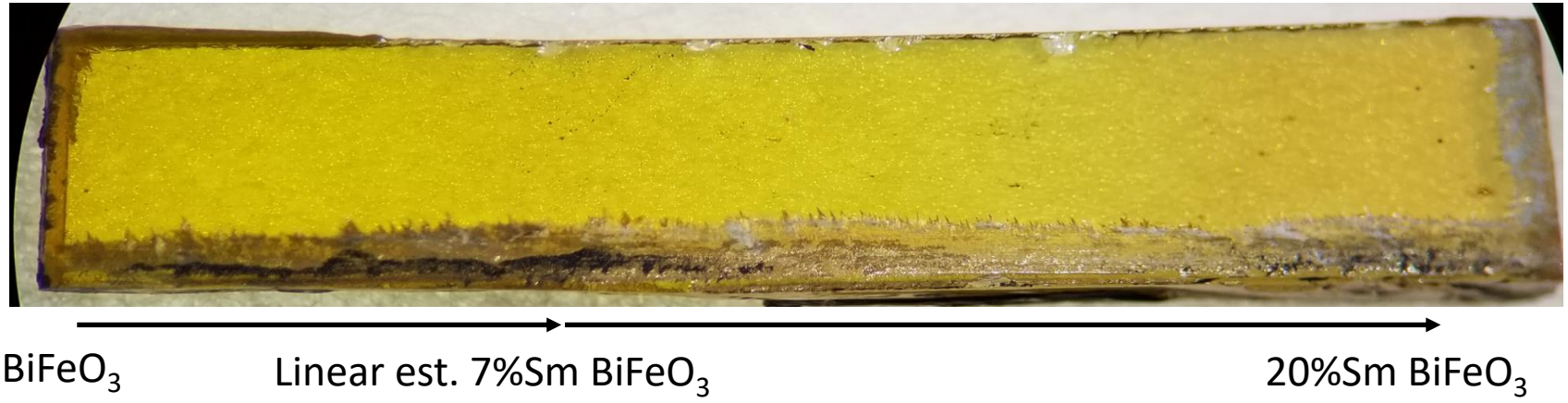


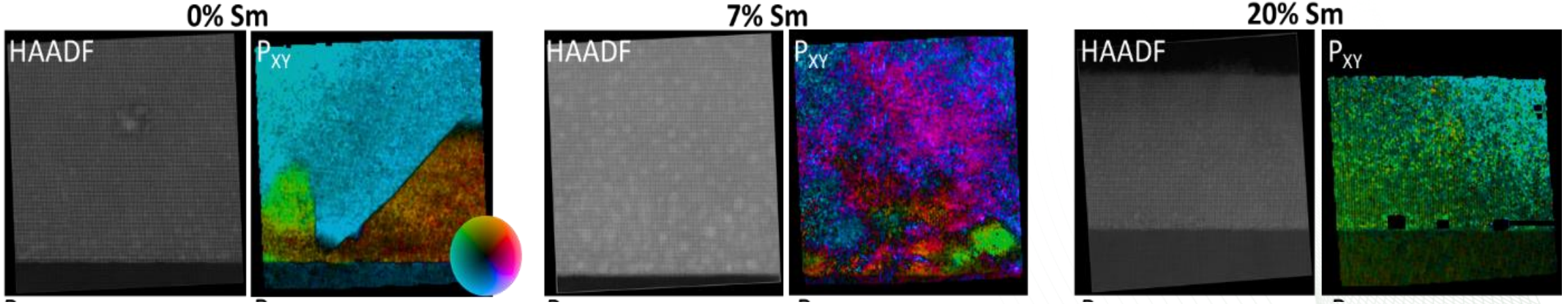


Combinatorial synthesis

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

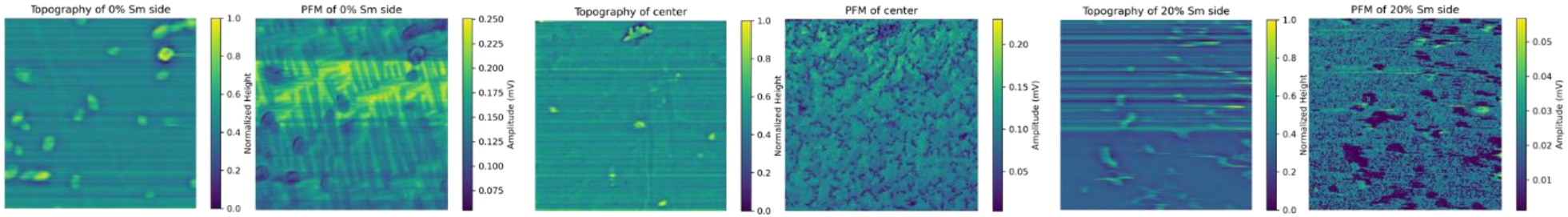






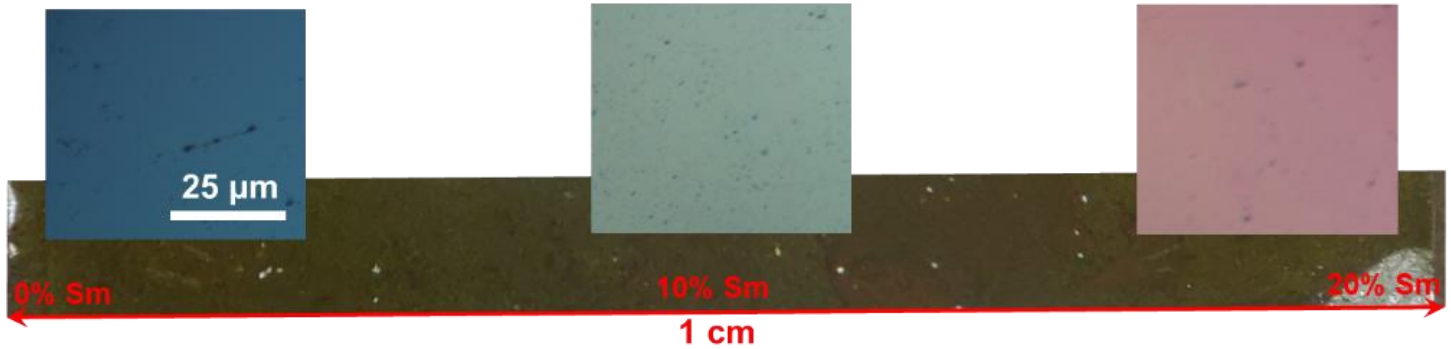
arXiv:2004.11817

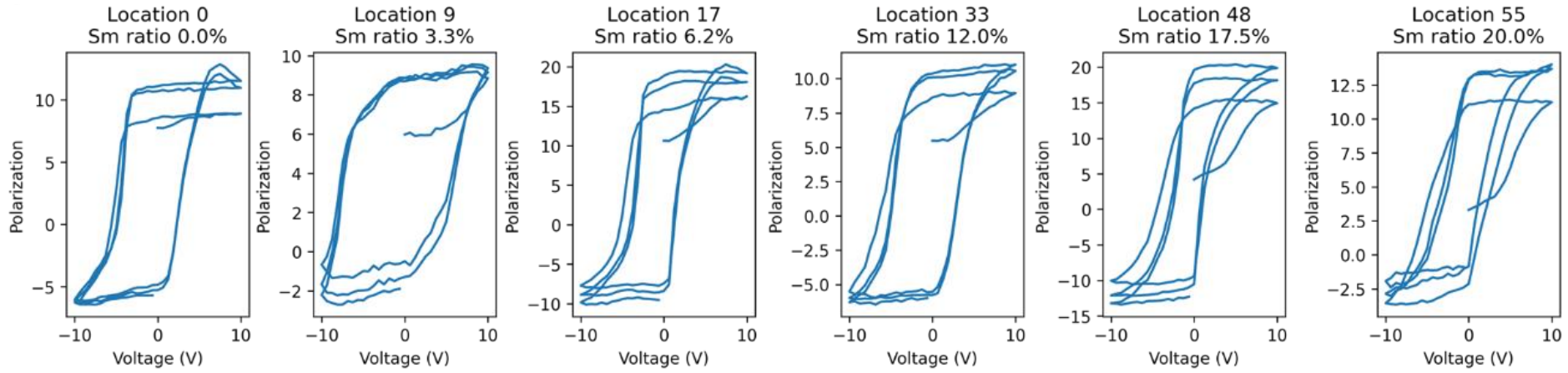
Combinatorial libraries:



optical images

sample picture





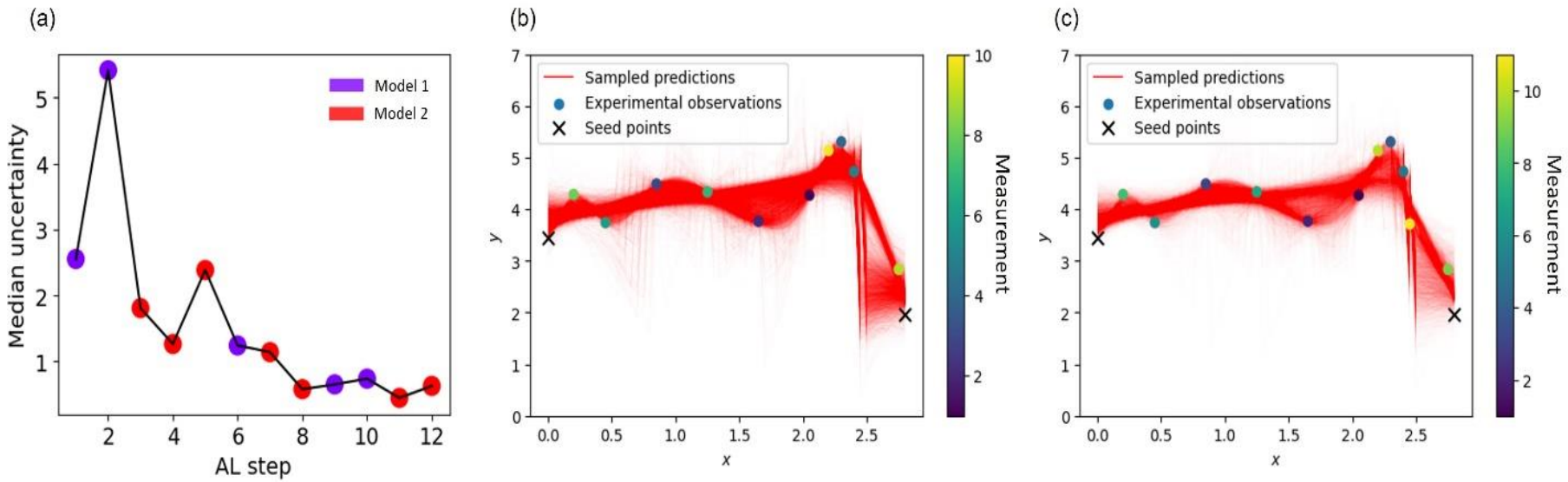
Hypothesis selection for ferroelectric response

Model 1 (second order phase transition):

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

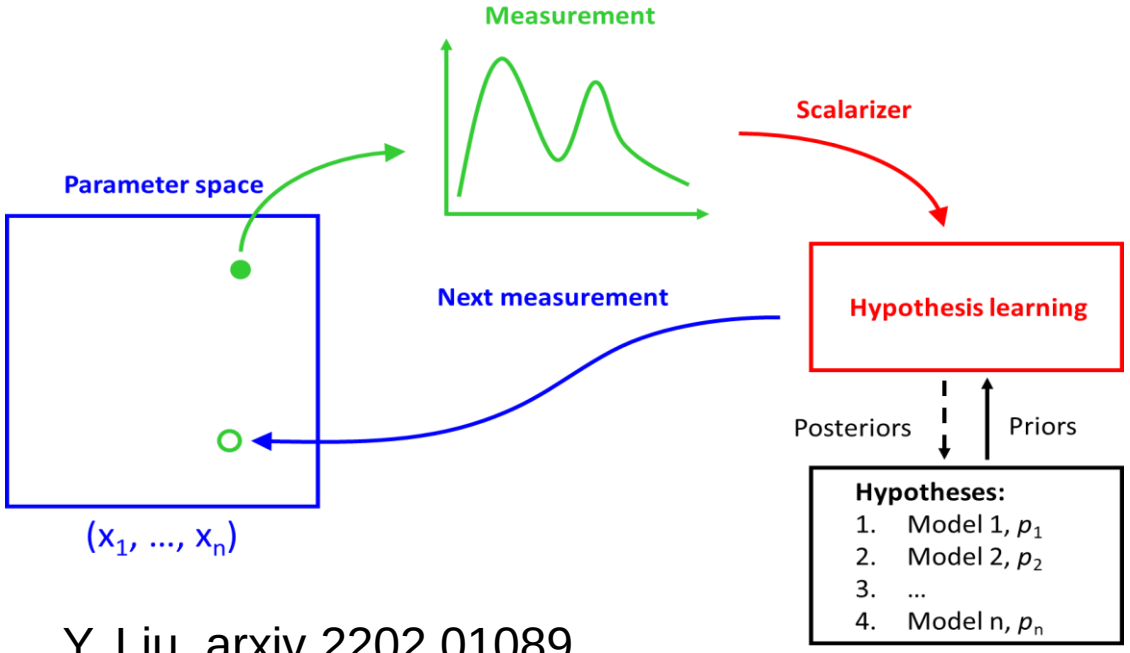
Model 2 (first order phase transition):

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

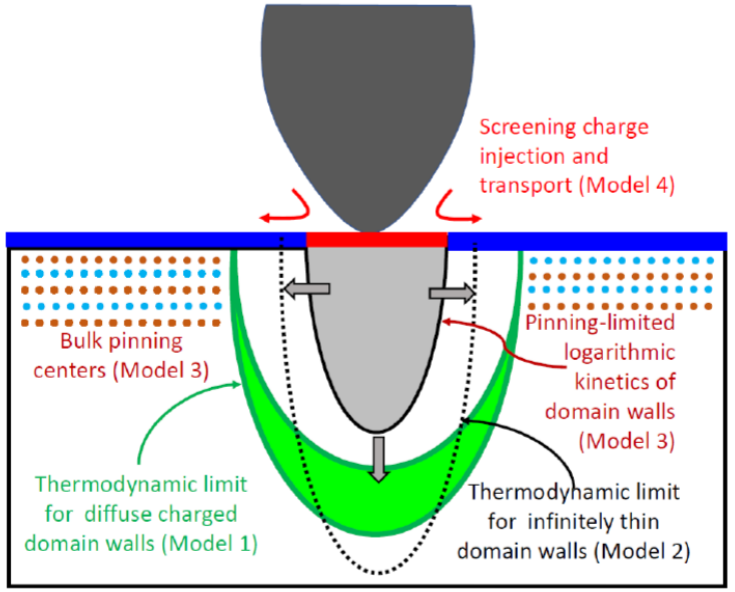


Hypothesis Learning

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



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



Thermodynamic 1

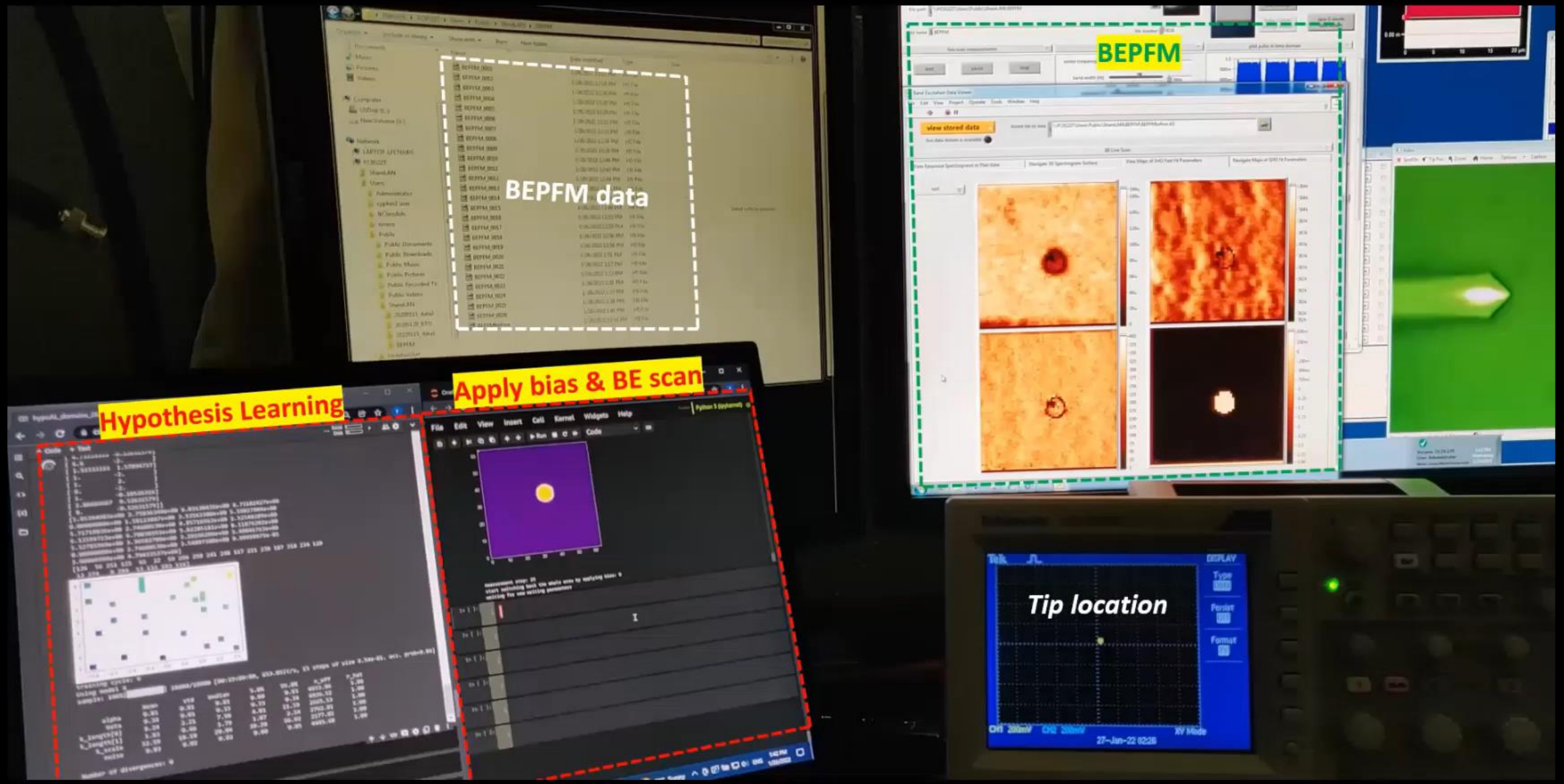
Thermodynamic 2

Wall pinning

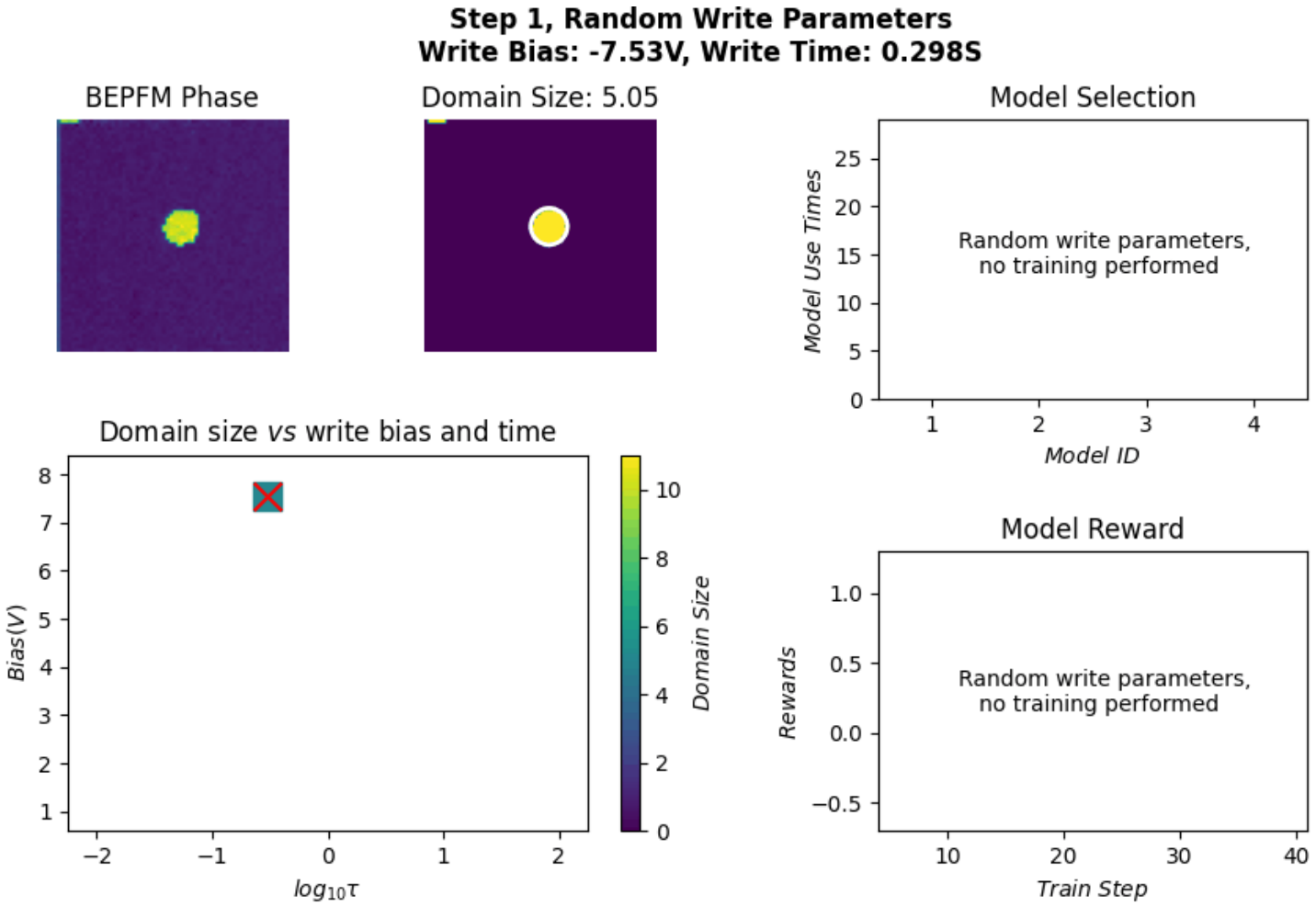
Charge injection

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

Automated Experiment #3—Automated Hypo-learning



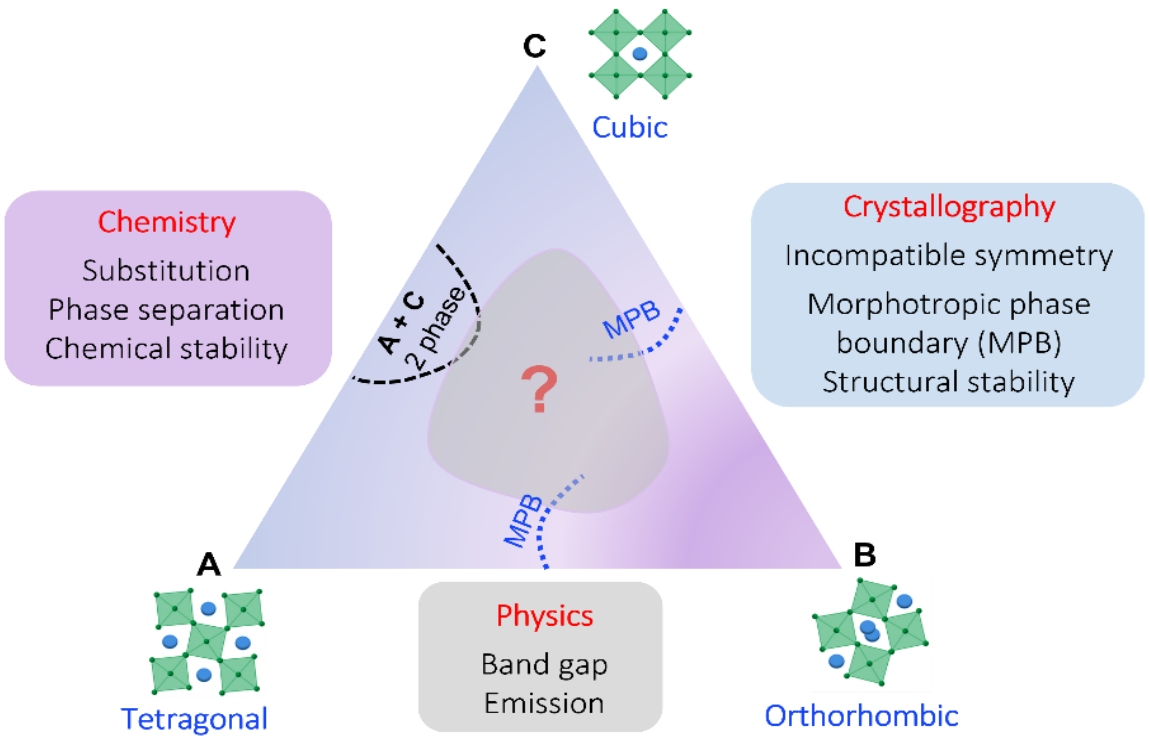
Hypothesis learning in action

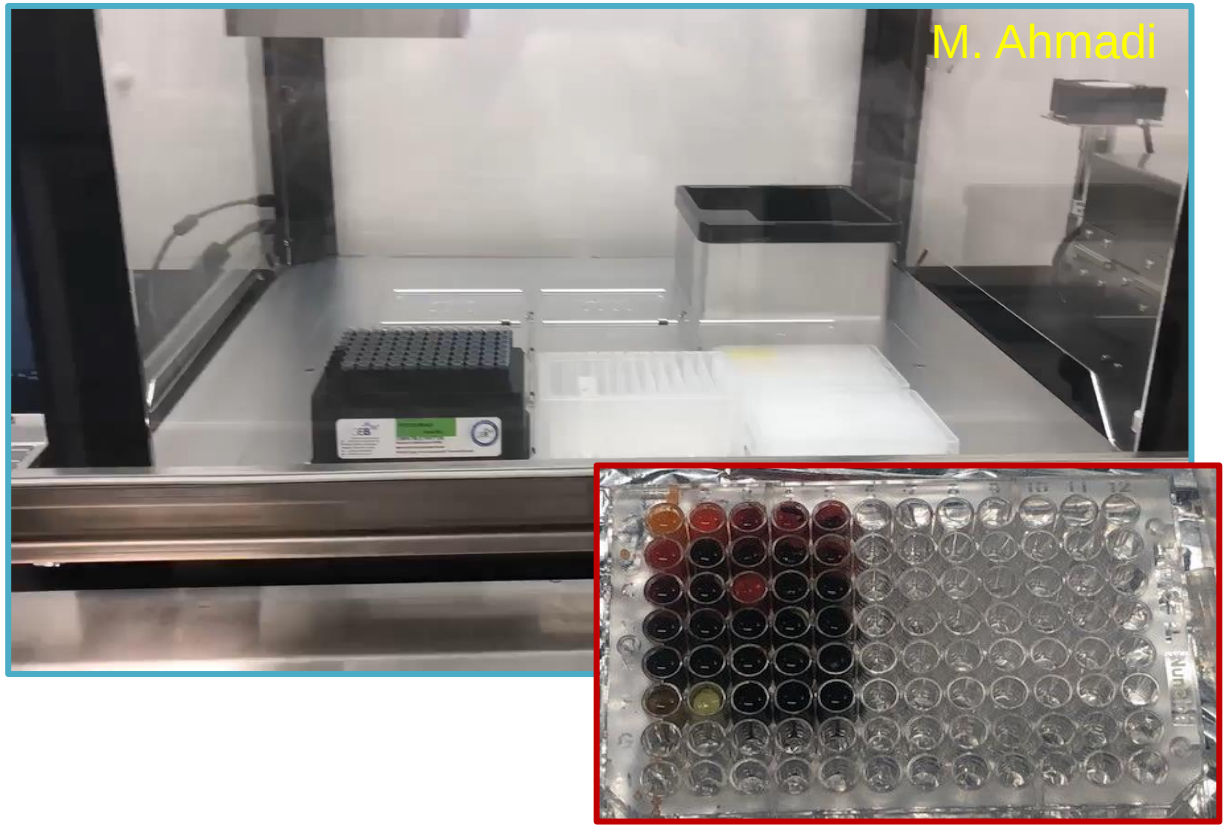


- ML algorithm has 4 competing hypothesis on domain switching mechanisms
- These hypothesis represent full set of possibilities for this system
- The microscope chooses experimental parameters in such a way as to establish which hypothesis is correct fastest
- Important: the same approach can be implemented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

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

Why synthesis (or theory)?

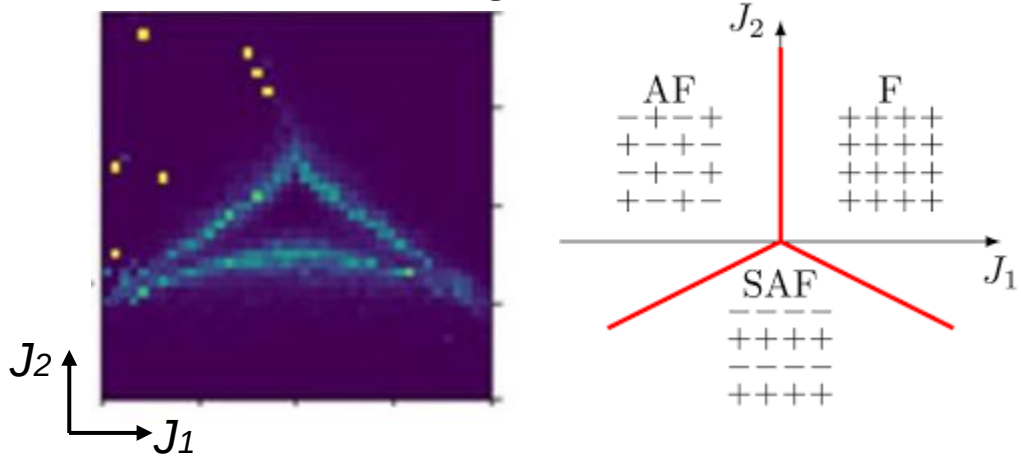




- Automated synthesis in its simplest form requires some way to navigate phase diagrams
- In more complex form, processing space.
- Ideally, incorporate physical knowledge

- Similar problem - theory

Ising model



The World is Bayesian: Physics from Observations

Hypothesis driven science:
What we want to learn

Forward model: Theory

Domain expertise:

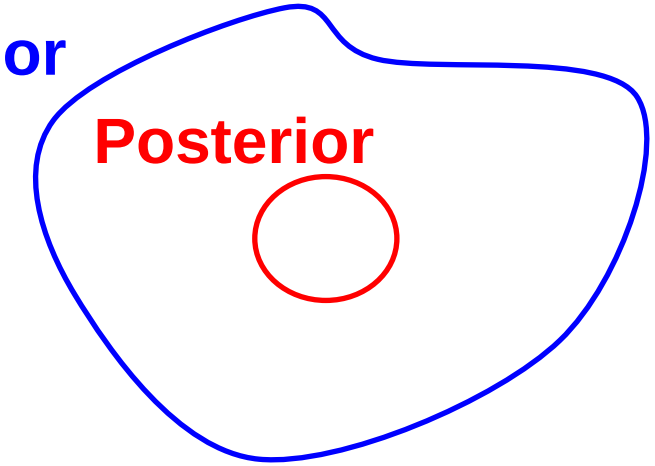
$$P(Theory|Data) = \frac{P(Data|Theory)P(Theory)}{P(Data)}$$

High Performance Computing

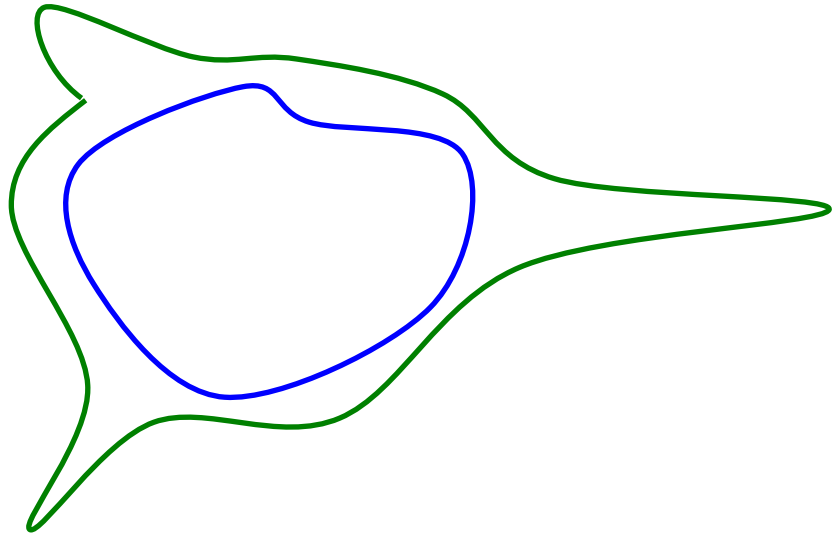
- Experimentalists know the priors. Albeit they do not know that they know it, or how to convert them to algorithmic form
- **However, how do we make guesses about the unknown?**

Prior

Posterior



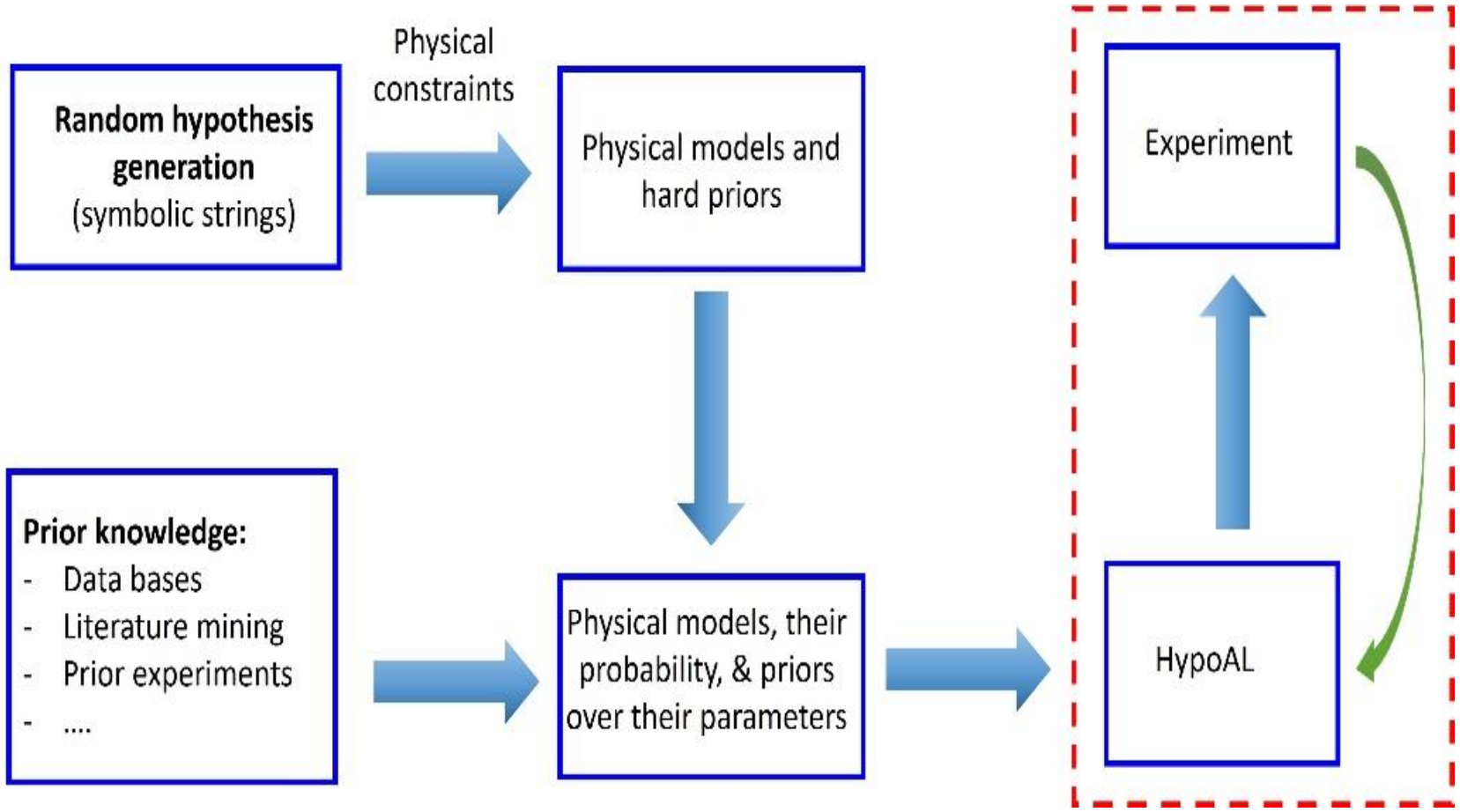
Refinement:
Can be defined as probabilistic model



Hypothesis formation:
How can we do it?

Towards hypothesis learning

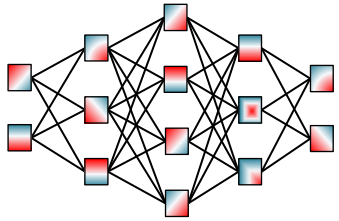
- Data already exists: Eureka, SISSO, SinDY, etc.
- What if we make hypothesis learning a part of active experiment?
- Need policies for hypothesis generation



More then ML: Human in the Loop

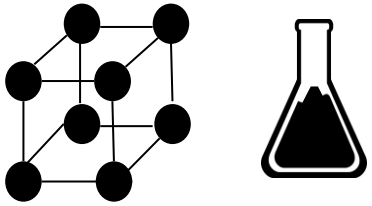
Prior knowledge

- Weights
- Inductive biases



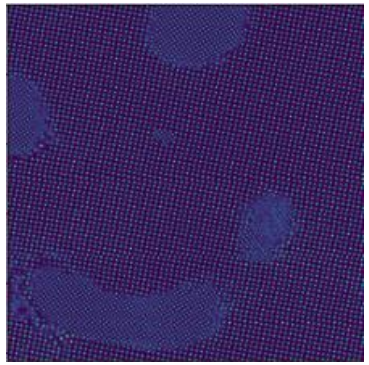
Reward Function

Material Parameters



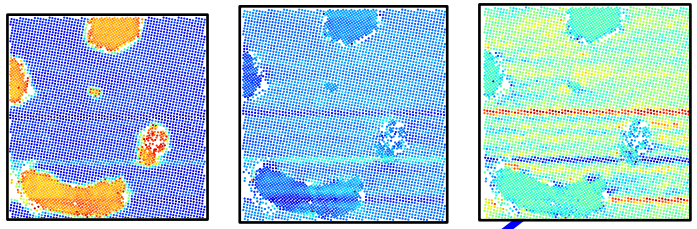
Experiment

Human

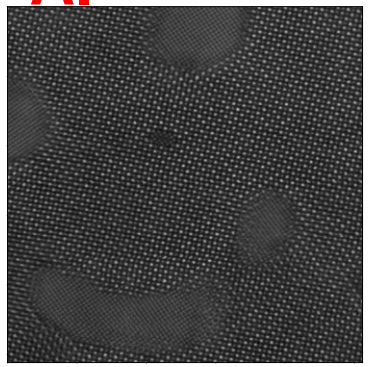


Select analysis method

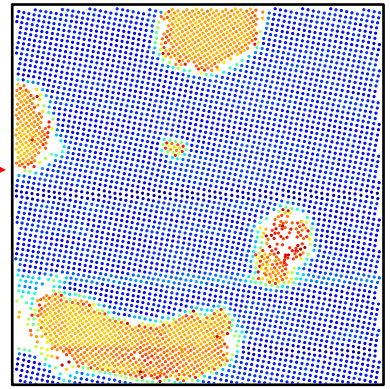
Select component



AI

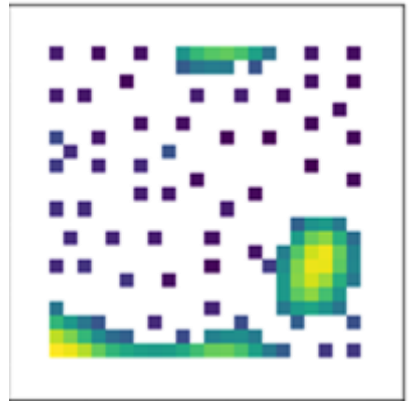


DCNN



rVAE

BO



Discovery pathway depends on the reward structure!