Autonomous Microscopy with Coupled Simulation-experiment Workflows: Requirements and Opportunities

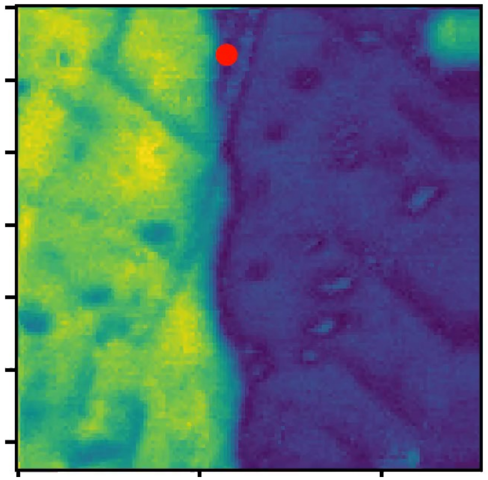
Rama K. Vasudevan

Center for Nanophase Materials Sciences, Oak Ridge National Laboratory

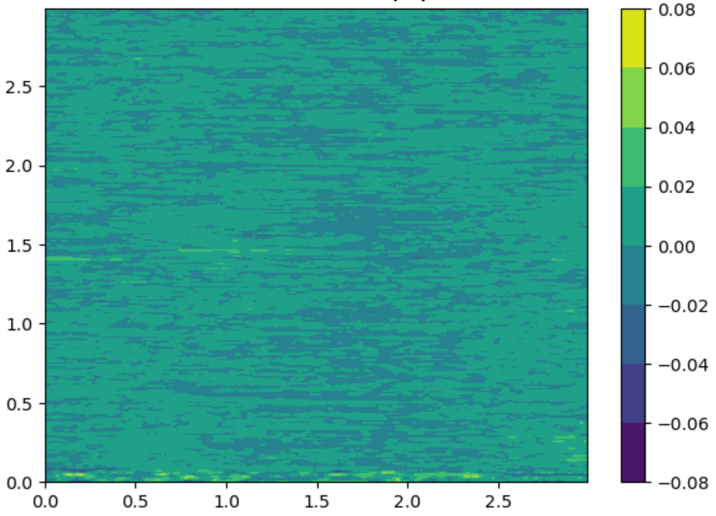
2023 OLCF USER MEETING
18th October 2023

Autonomous Labs: Characterization and Synthesis

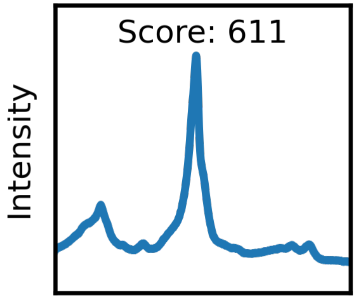
Piezoresponse Force Microscopy

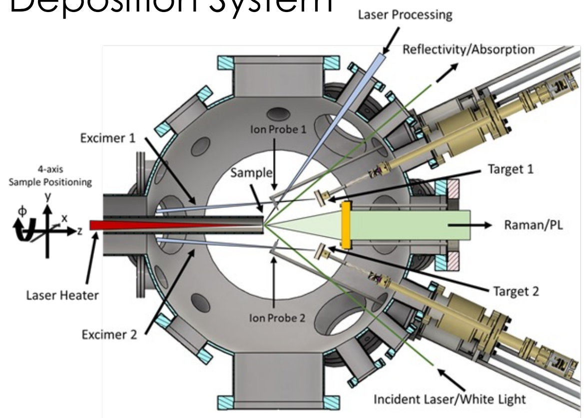


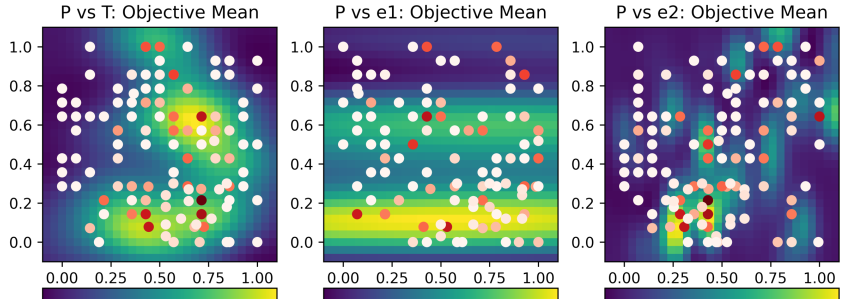
Scanning Tunneling Microscopy



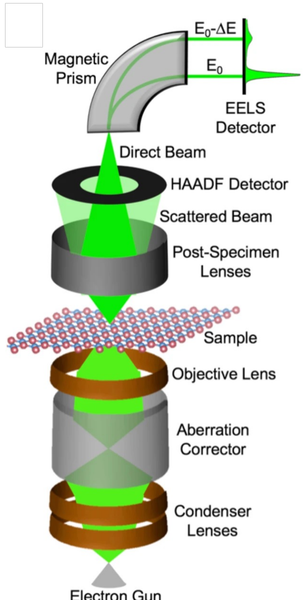
Pulsed Laser Deposition System

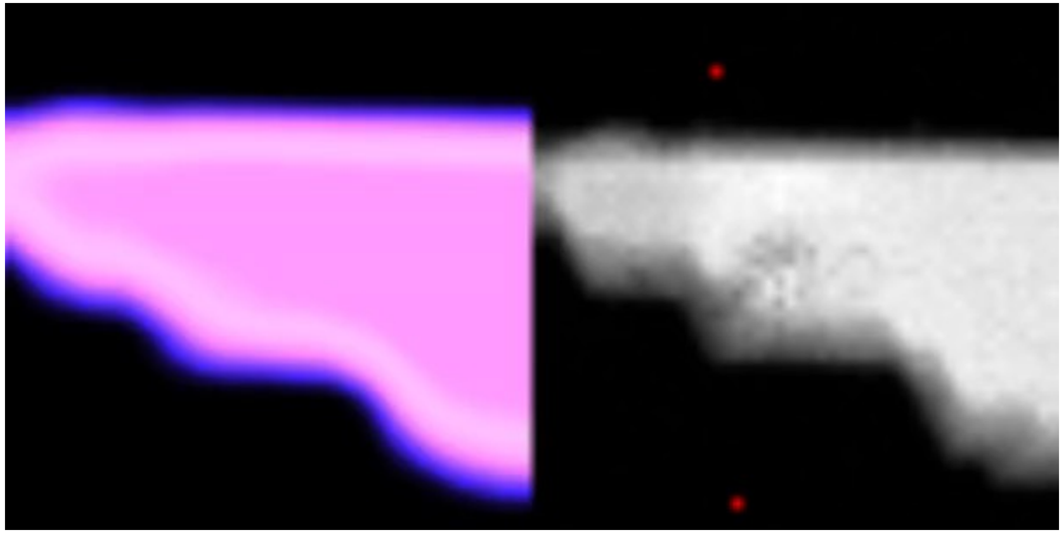






Scanning Transmission Electron Microscopy





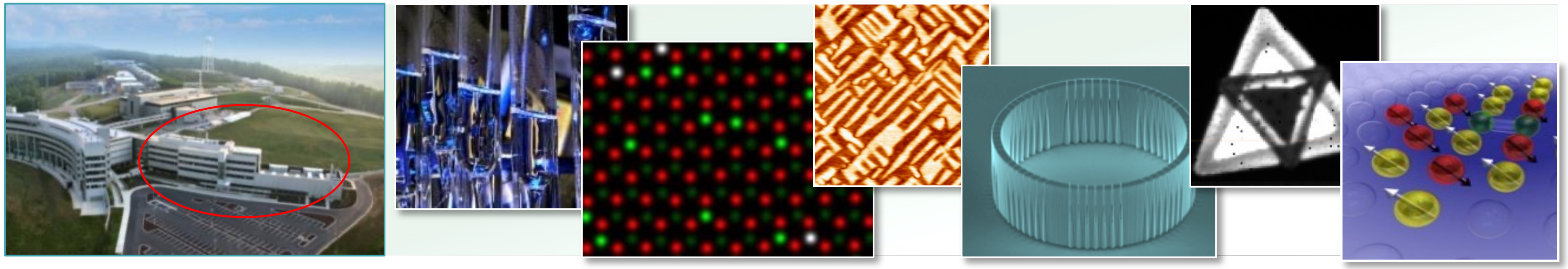
CNMS is a national user facility with a mission to advance nanoscience

About CNMS:

- **Unlike many user facilities, you don't need to have samples to apply for time**
- **Two calls per year for continuous access; anytime for short-term projects**
- **Simple 2-page proposal**
- **Free access to laboratories, equipment and expertise if you agree to publish**
- **Proposal deadlines: early May and mid-October**
- **Joint proposals with neutron sources (SNS, HFIR)**

Research areas:

- **Synthesis –** 2D, precision synthesis, selective deuteration
- **Nanofabrication –** direct-write, microfluidics, cleanroom
- **Advanced Microscopy –** AFM, STM, aberration-corrected TEM/STEM, atom-probe tomography
- **Functional Characterization –** laser spectroscopy, transport, magnetism, electromechanics
- **Theory and Modelling –** including gateway to leadership-class high performance computing



CNMS is a Nanoscale Science Research Center supported by the U.S. Department of Energy, Office of Science, Scientific User Facilities Division

Why do we need smart microscopy?

Control over synthesis pathways

Enabling new experiments

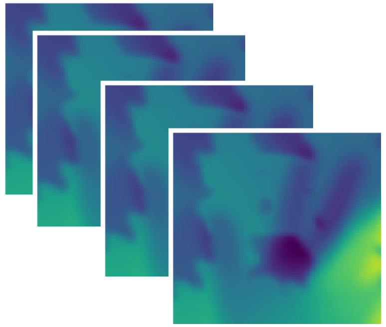
Closing the theory-experiment cycle in a reasonable timeframe

Focus on Science, not Operations

“Smart” workflow necessities

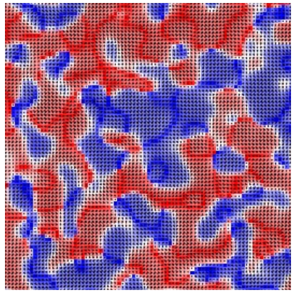
Feedback control

Microscope



Data

Algorithms



Simulations

State
Knowledge

Data Driven **Physics Driven**

Expressive ***Compact***

ML
DL

PINN

Hybrids

Learned
parameters

DFT
QMC

Compute

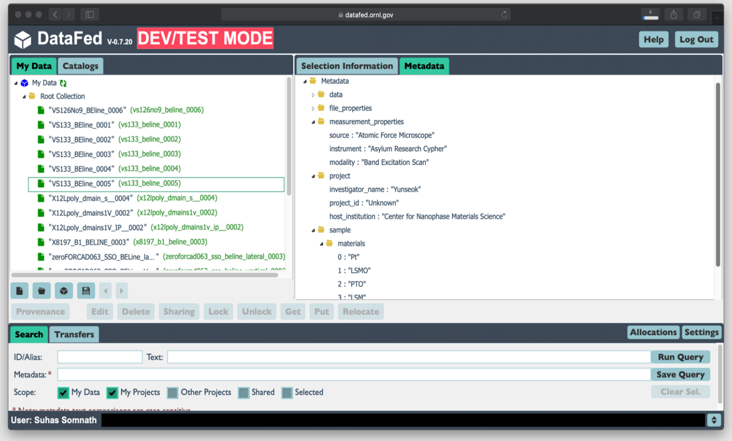
Realizing smart microscopy labs at user facilities

Compute Infrastructure

- Cloud services
- Edge computing – FPGAs
- “Far-edge” – dedicated GPUs for experiments
- Leadership class

Data Infrastructure

- Standardized data models
- Data pipelines / access
- Federated data stores



Software Infrastructure

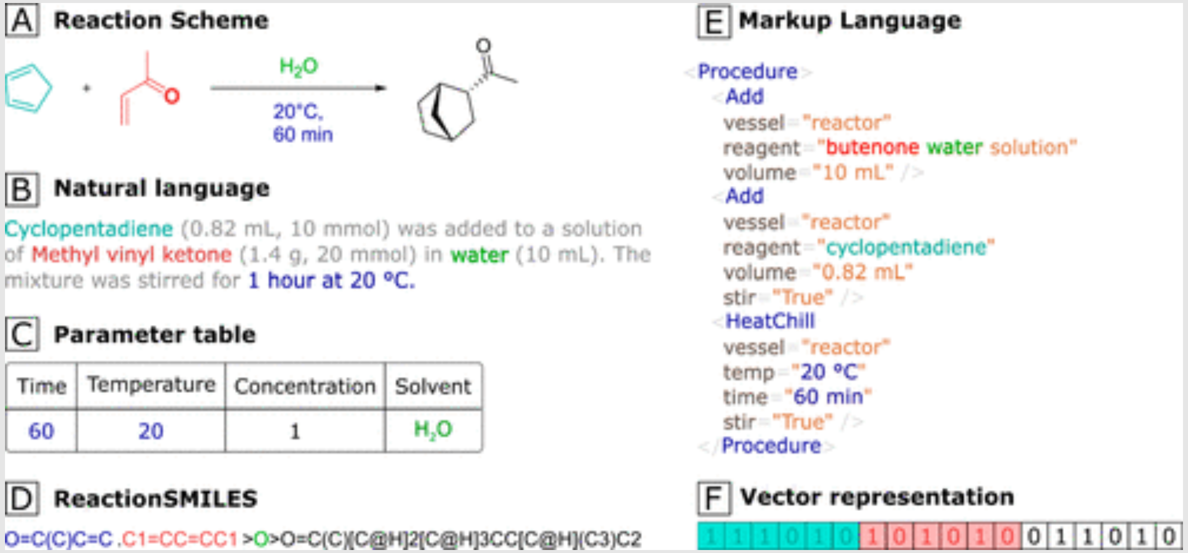
- Open-source analysis packages
- Data pipelines / access / control (INTERSECT)
- Federated data stores

pycroscopy

INTERSECT SDK

Abstractions make the (automated) world go round

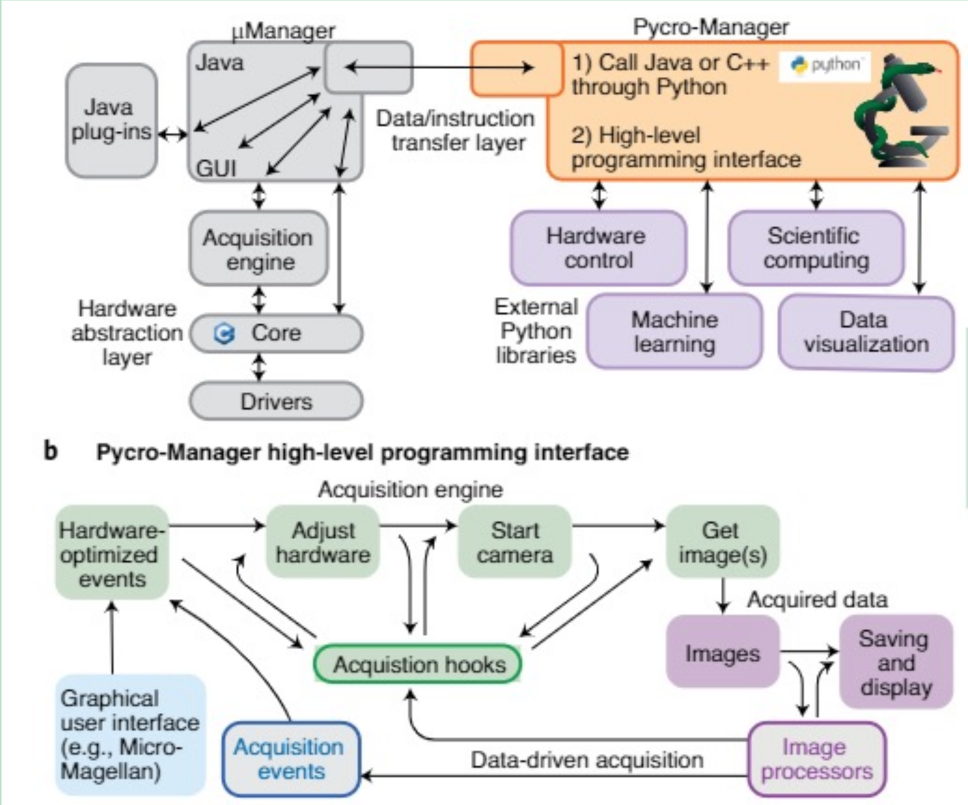
"Chemputation" – Digital chemistry



Hammer, Leonov, Bell and Cronin, JACS 1, 1572 (2021)

A complete programming language for chemistry that can run on open hardware

PycroManager

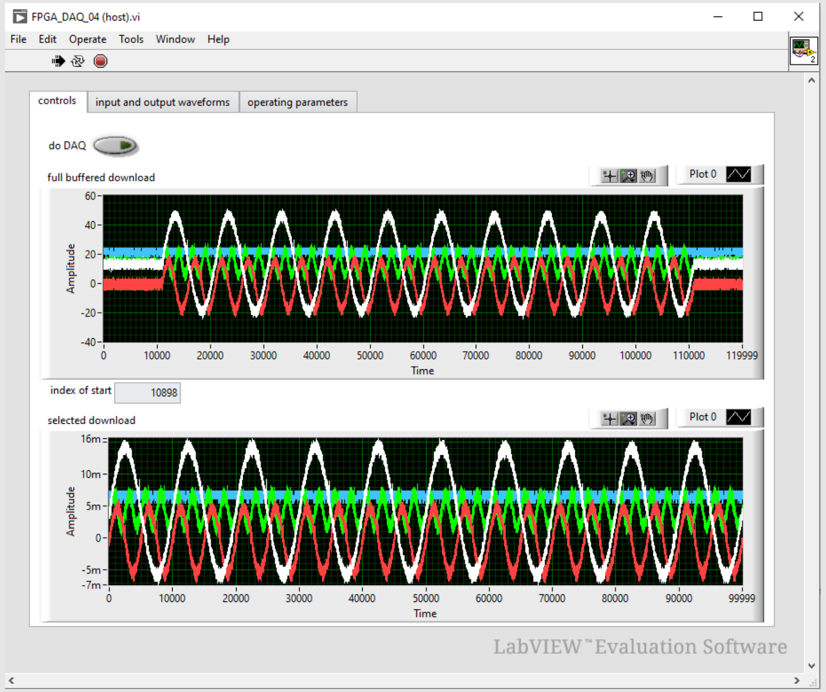


Pinkard, et al. Nat. Methods. 18, 226 (2021)

Computation already has abstractions. But most science characterization tools do not.

Oak Ridge National Laboratory

Hardware abstraction - FPGA

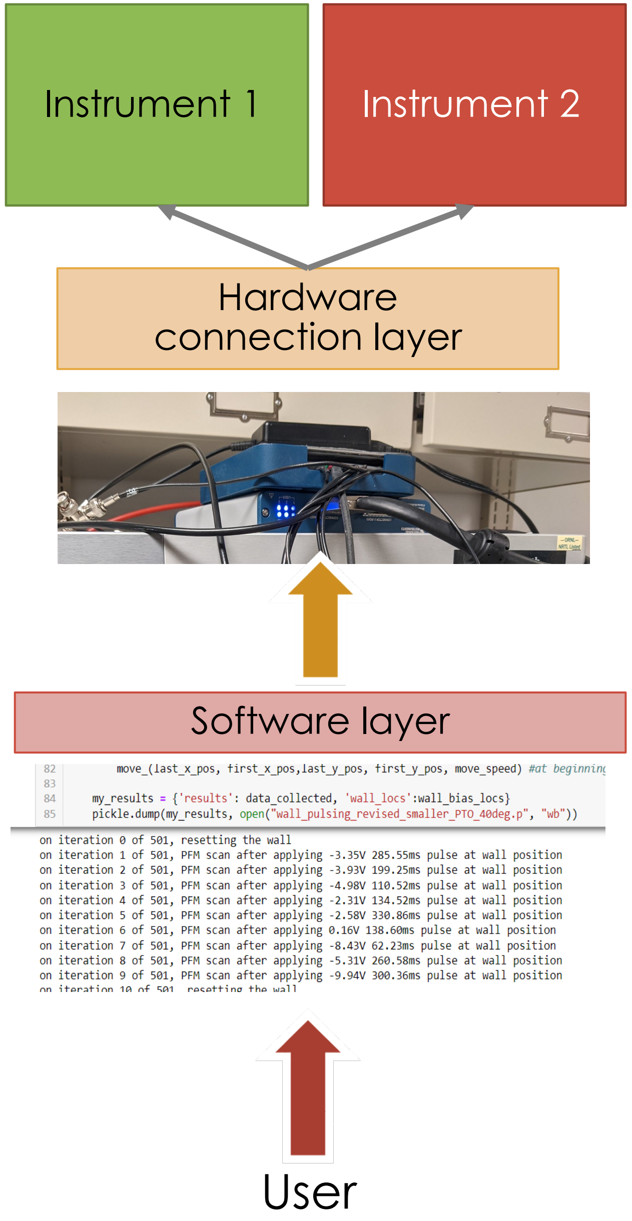


Software abstraction - python

```python
# Pole half the sample
# Locate the position of the wall
# Randomly select a location on the wall, and randomly select a bias pulse magnitude and pulse width
# Move tip to that location and apply a bias pulse, move the tip back to the start
# Then scan to observe effects

wall_bias_locs = []

for k in range(num_iters):
    if k%reset_freq==0:
        print('on iteration {} of {}, resetting the wall'.format(k, num_iters))
        data_output, d, x , y = raster_scan(volt,pix,num_lines, IO_rate,trig,offsetvx,offsetvy, tip_volt=reset_bias,
                                    apply_half=True, left_bias=False)

        last_x_pos = x[-1]
        last_y_pos = y[-1]
        first_x_pos = x[0]
        first_y_pos = y[0]

        move_(last_x_pos, first_x_pos,last_y_pos, first_y_pos, move_speed) #at beginning of scan, move to starting p

        data_output, d, x , y = raster_scan(volt,pix,num_lines, IO_rate,trig,offsetvx,offsetvy, tip_volt=-reset_bias
                                    apply_half=True, left_bias=True)
        move_(last_x_pos, first_x_pos,last_y_pos, first_y_pos, move_speed) #at beginning of scan, move to starting p

        #Do a standard scan, without any bias.
        data_output, d, x , y = raster_scan(volt,pix,num_lines, IO_rate,trig,offsetvx,offsetvy, tip_volt=0.0,
                                                apply_half=False)
        data_collected.append(data_output)

    else:
        #find the mean wall position
        amp_img = data_output[2].reshape(-1, pix*2)[:,:pix]
        phase_img = data_output[2].reshape(-1, pix*2)[:,:pix]
        bias_amp, bias_pw, wall_x_pos, wall_y_pos, xpos_v, ypos_v = get_next_action(amp_img, phase_img, pix=pix)
```

Call low-level functions to control tip position, scanning (e.g., raster, spiral, move etc.), specify voltage waveforms, collect data all in Jupyter notebook

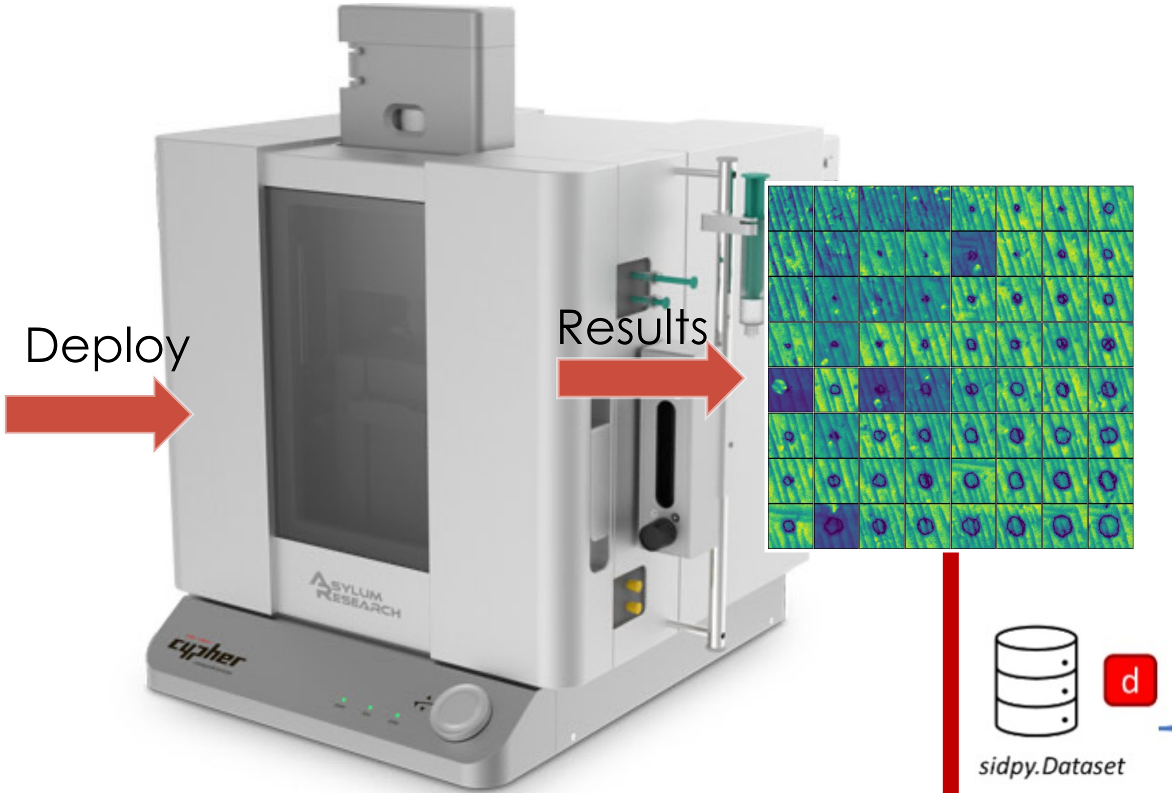
Enables design of complicated automated and autonomous experiments, hardware independent

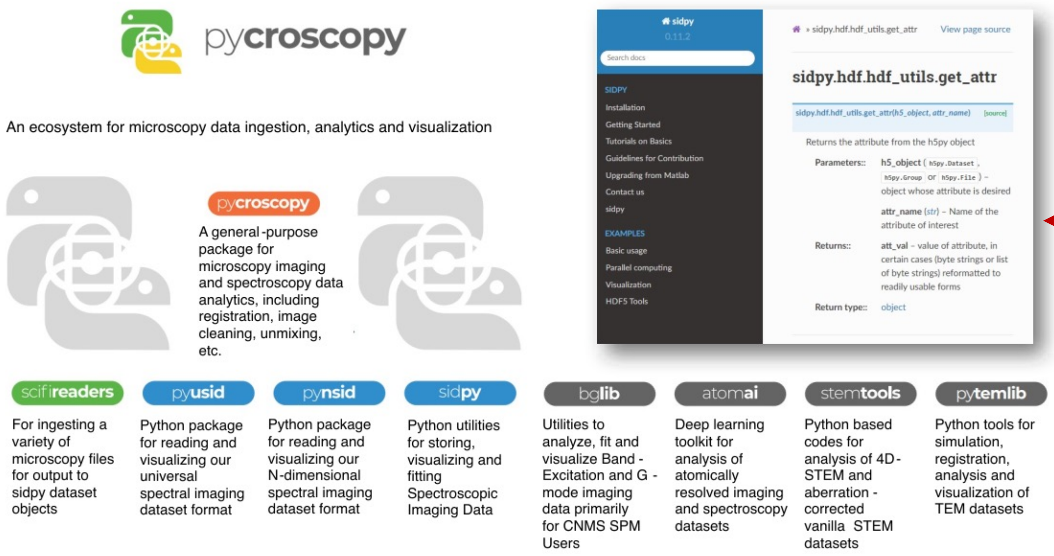


Automating SPM: AEcroscopy

Software Infrastructure

Credit: Yongtao Liu (CNMS/ORNL)

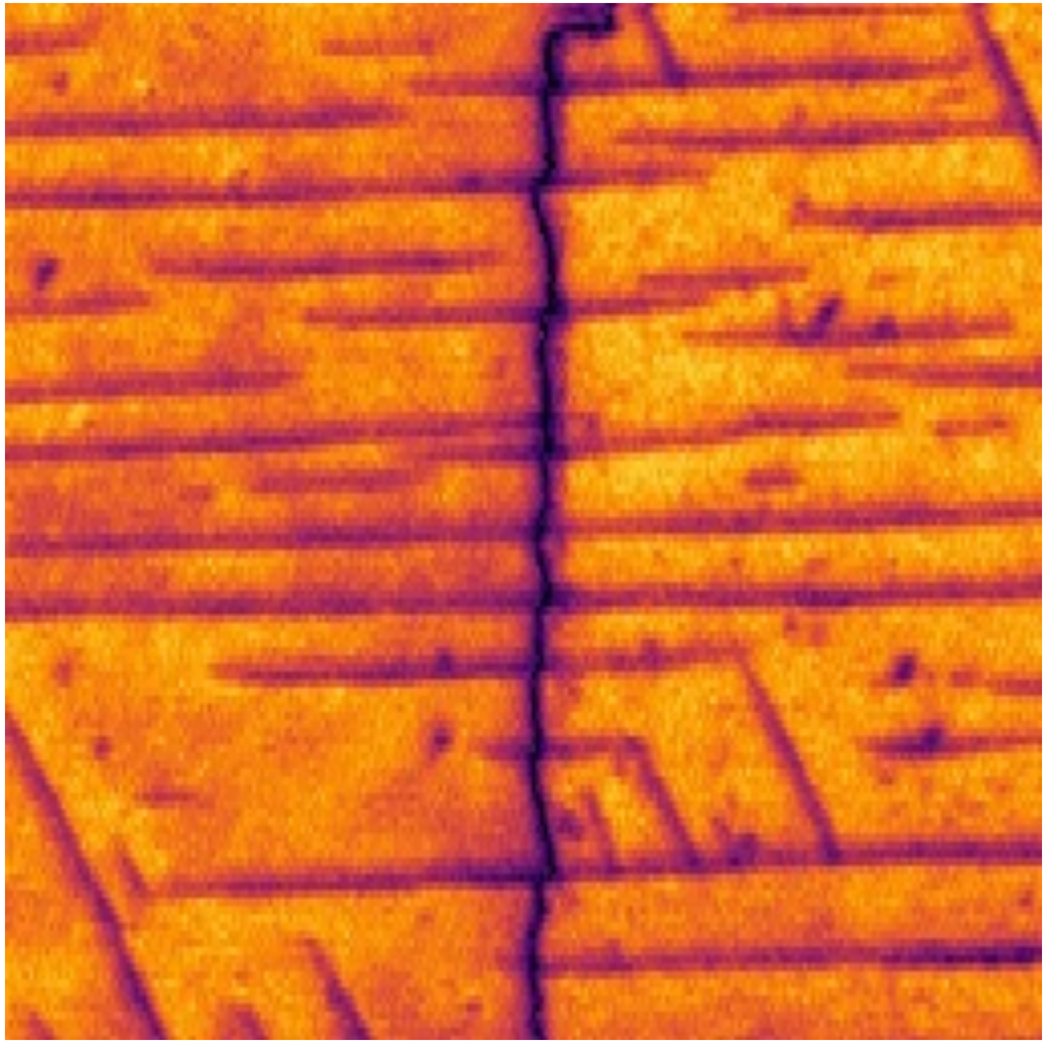


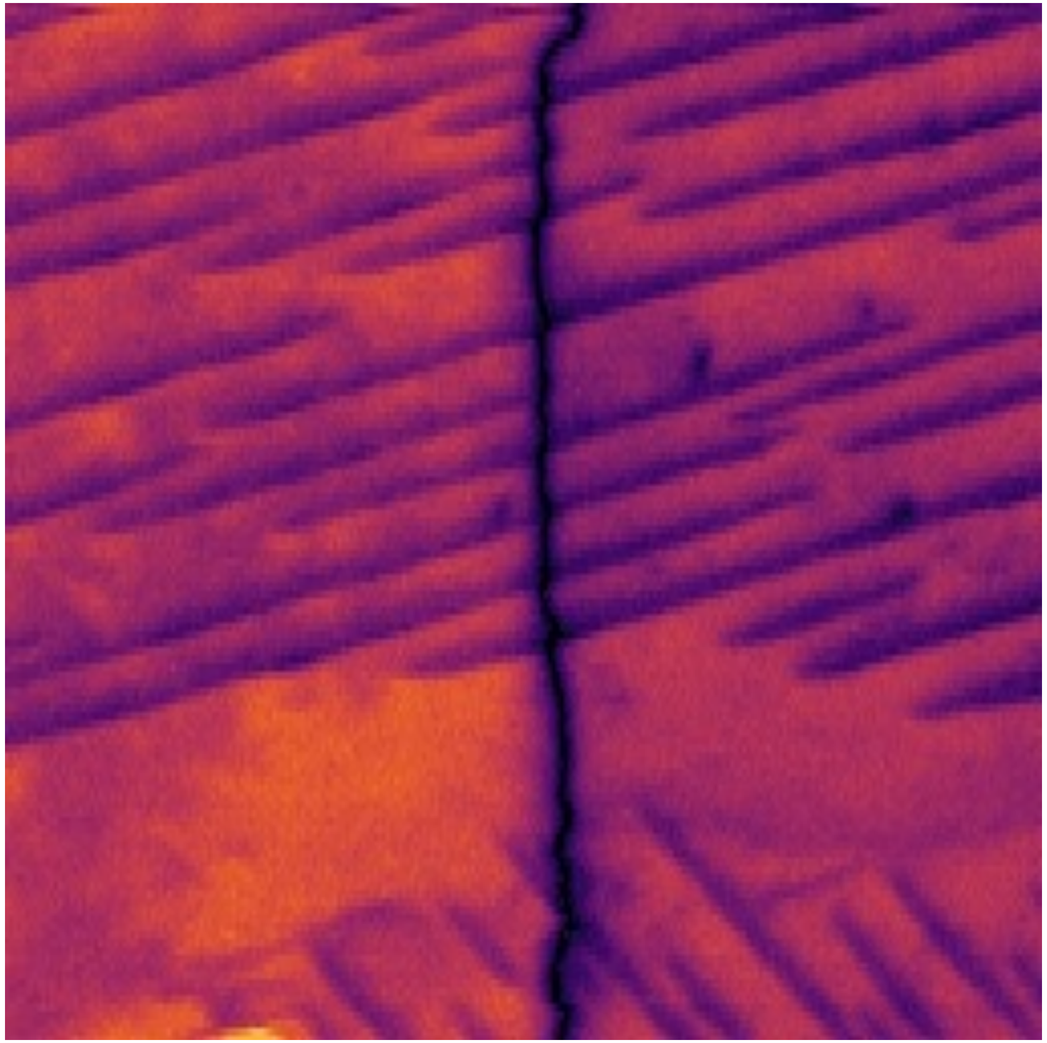


Vasudevan et al. Advanced Theory and Simulations (10.1002/adts.202300247) (2023)

- Standardized data model
- In-built processing and viz utilities

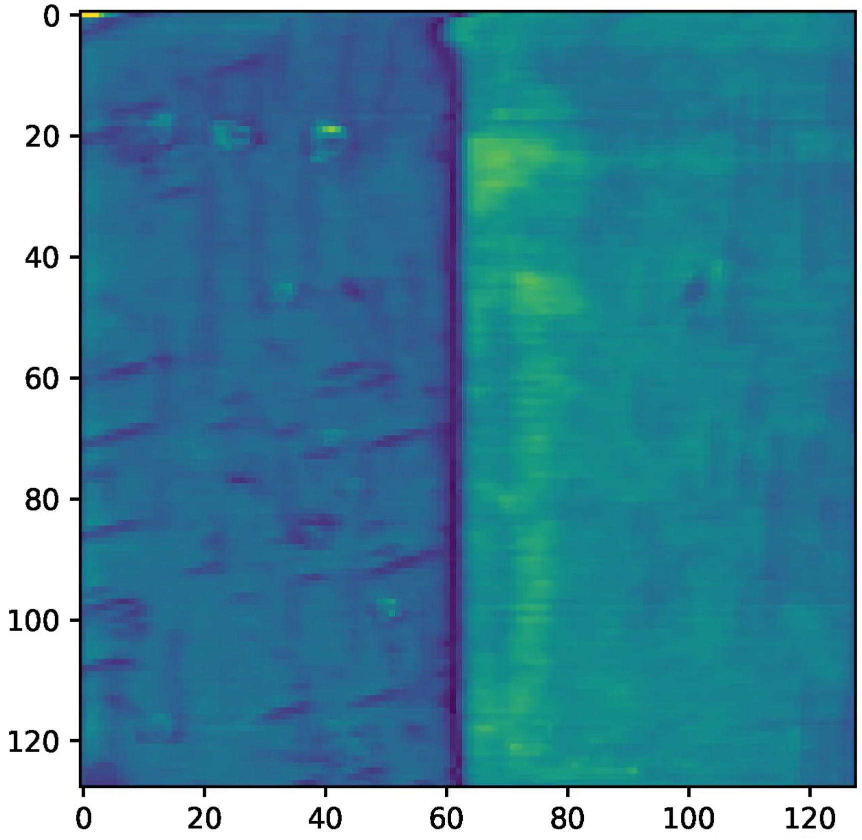
Example automated datasets



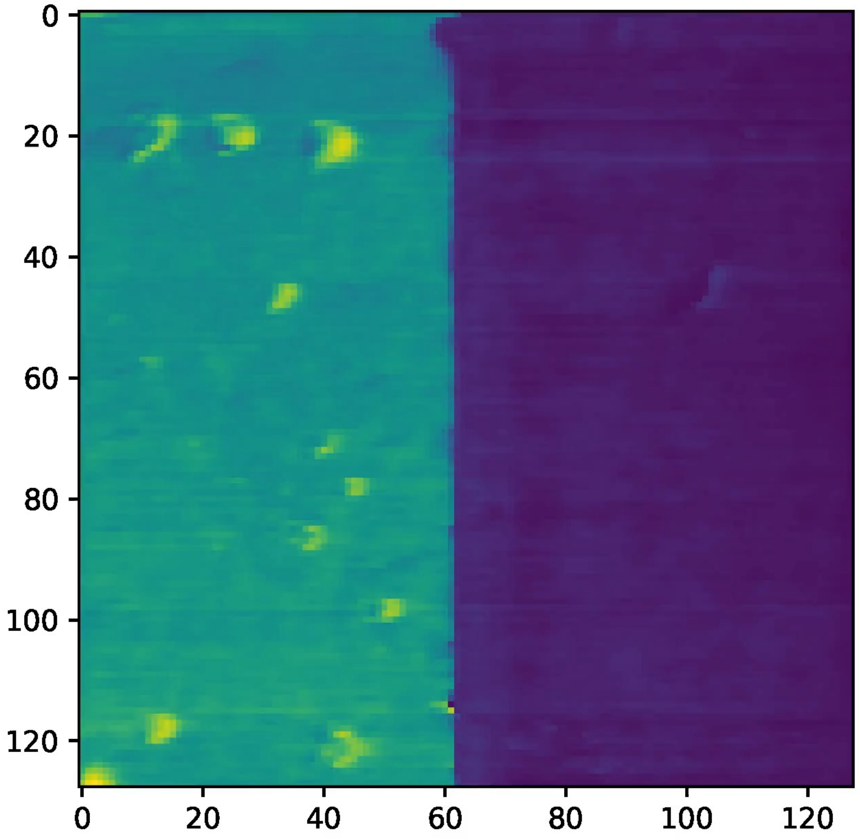


Same experiment, different day

Vertical PFM Amplitude



Vertical PFM Phase



Automated experiments and ML can only go so far

We have access to theory: why not use it?

Ganesh, Rajeev, Shuaifang, CNMS

Scalable Multi-Physics Modeling of Field Driven Coupled Phase Transitions

$$F = \int \left(\psi_{landau}(\boldsymbol{P}) + \psi_{grad}(\boldsymbol{\nabla P}) + \psi_{mech}(\boldsymbol{P}, \boldsymbol{\varepsilon}) + \psi_{elec}(\boldsymbol{P}, \boldsymbol{E})\right) d\Omega,$$

Electric potential ϕ: $\kappa_0 \nabla^2 \phi - \nabla \cdot \vec{\boldsymbol{p}} = 0$

Fully coupled

Polarization (p_1, p_2, p_3)

$$\frac{\partial p_i}{\partial t} = -L \frac{\delta F}{\delta p_i} = -L\left(\frac{\partial F}{\partial p_i} - \nabla \cdot \left(\frac{\partial F}{\partial(\nabla p_i)}\right)\right)$$

Open source software

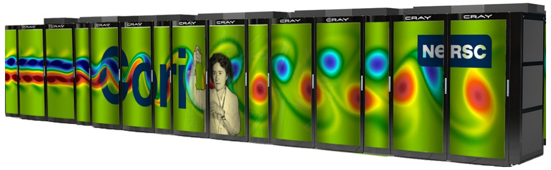
MOOSE

PETSc

Fully coupled

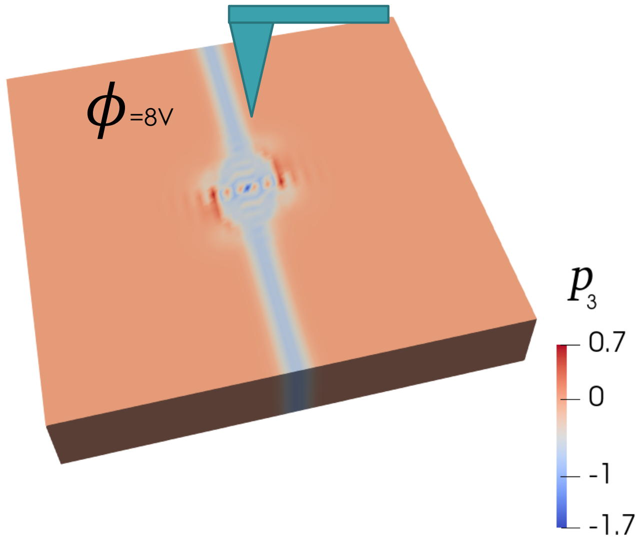
Mechanical equilibrium equation

$$\nabla \cdot \left(C_{ijkl}\epsilon_{kl} - q_{ijkl}p_k p_l\right) = 0$$



CADES

High-Throughput Phase Field



3D simulations of polarization response to a high-field at a domain-wall in $BaTiO_3$

Symbolic Regression

$$\log(t_{\text{switch}}) = -8.777 + \frac{k_{th}}{\sigma}\left[0.197\exp(V_{ij}) - 0.00354\left(\frac{k_{th}}{\sigma}\right)\right], \quad (1)$$

$$\frac{I_{on}}{I_{off}} = -0.597 - 7.676\left(\frac{\sigma}{k_{th}}\right) + 12.637\left(\sqrt{\frac{\sigma}{k_{th}}}\right). \quad (2)$$

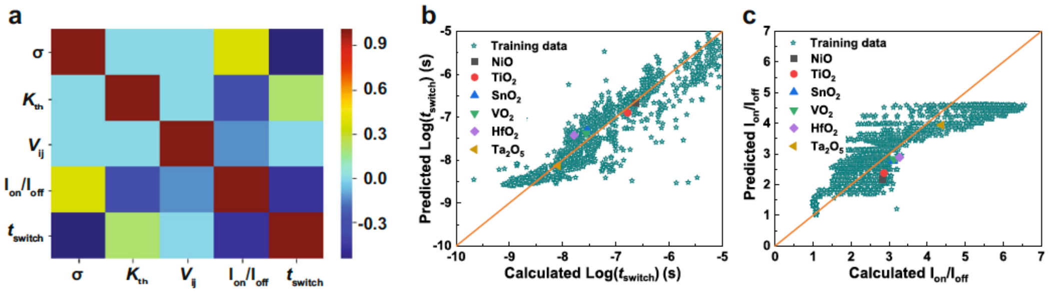
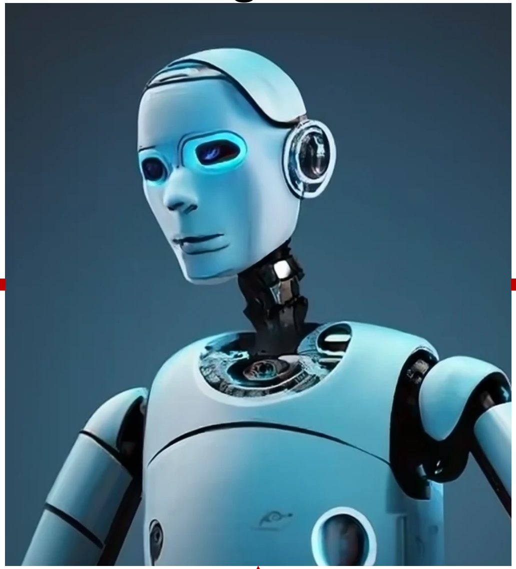


Fig. 5 Machine learning results. a The Pearson correlation plot between the different material-characteristics and the performance metrics. The SISSO model fits to the data using a 2D-descriptor for **b** t_{switch} and **c** I_{on}/I_{off}.

Autonomous theory-experiment workflow requirements

Orchestrating Agent



DoE

Experiment script

AEcroscopy

Welcome to AEcroscopy

Get Started

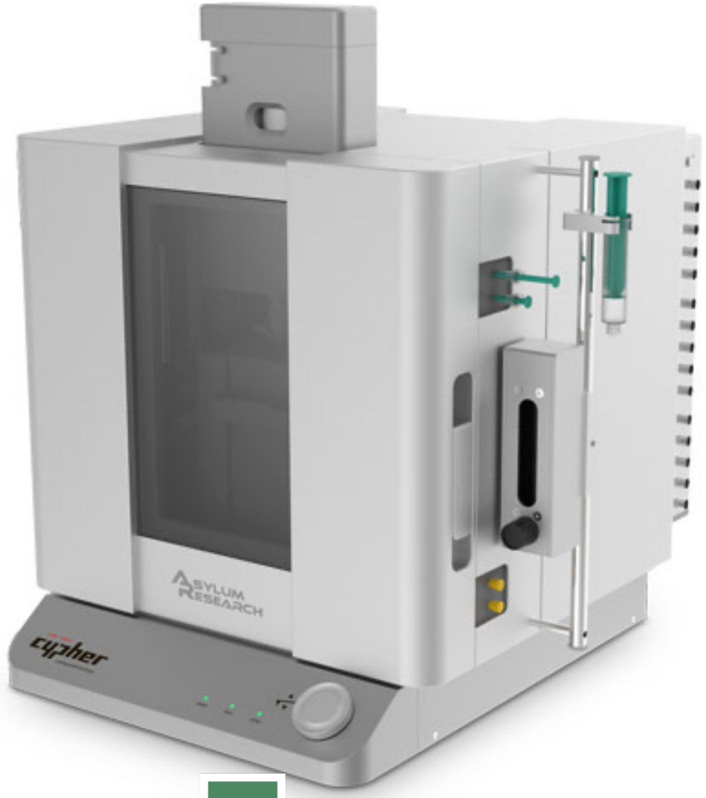
Get Started

Experiment

Experiments

Step 4. Do a BEPFM at the whole exp

```
dset_pfm, dset_chns, dset_cs = newexp.raster_scan(r

f, (ax1, ax2, ax3, ax4, ax5, ax6) = plt.subplots(1,
ax1.imshow(dset_pfm[:,:,0])
ax2.imshow(dset_pfm[:,:,1])
ax3.imshow(dset_pfm[:,:,2])
ax4.imshow(dset_pfm[:,:,3])
ax5.imshow(dset_chns[0,:,:])
ax6.imshow(dset_chns[1,:,:])
plt.show()
```



Results viz. + analysis

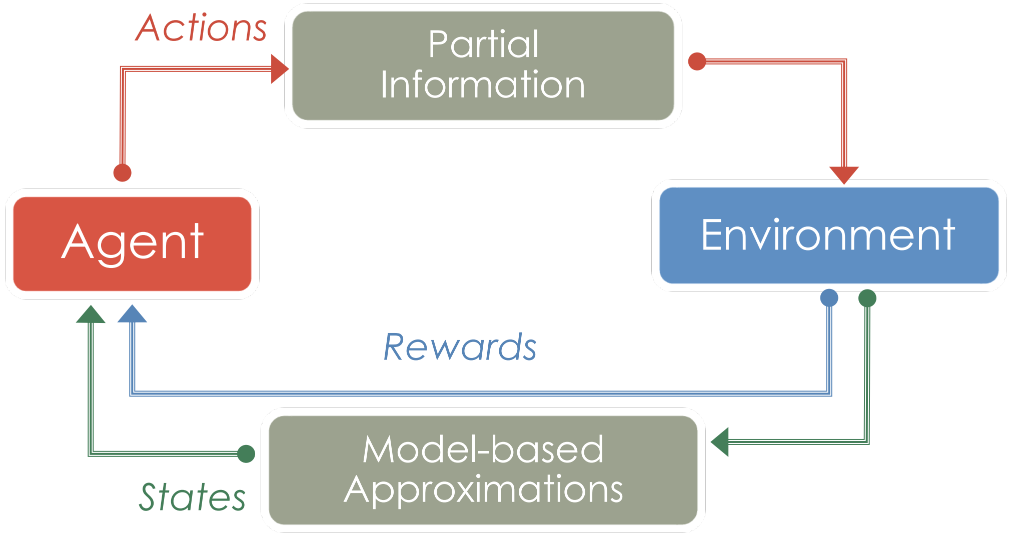
Updated Parameters

Large scale phase-field simulations

MOOSE

Data stores (e.g., cloud)

Reinforcement Learning



- RL is a type of machine learning where an agent learns to a policy in an environment by repeated interactions, with a goal of improving that policy's expected rewards

Policy Objective

$$\pi(a|s) = \mathbb{P}(A_t = a \,|S_t = s)$$

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\,[R(\tau)]$$

Standard Policy Gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_\tau[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)G_t]$$

Actor-Critic Policy Gradient:

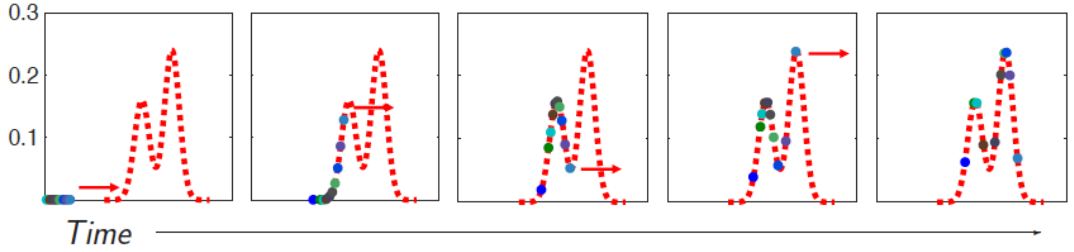
$$\nabla_\theta J(\theta) \sim \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)(r_{t+1} + \gamma V_v(s_{t+1}) - V_v(s_t))$$

$$= \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)A(s_t, a_t)$$

Policy gradient methods: Stein variational policy gradient (SVPG) algorithm

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t + \delta_t \phi(\boldsymbol{\theta}_i^t) \text{ where}$$

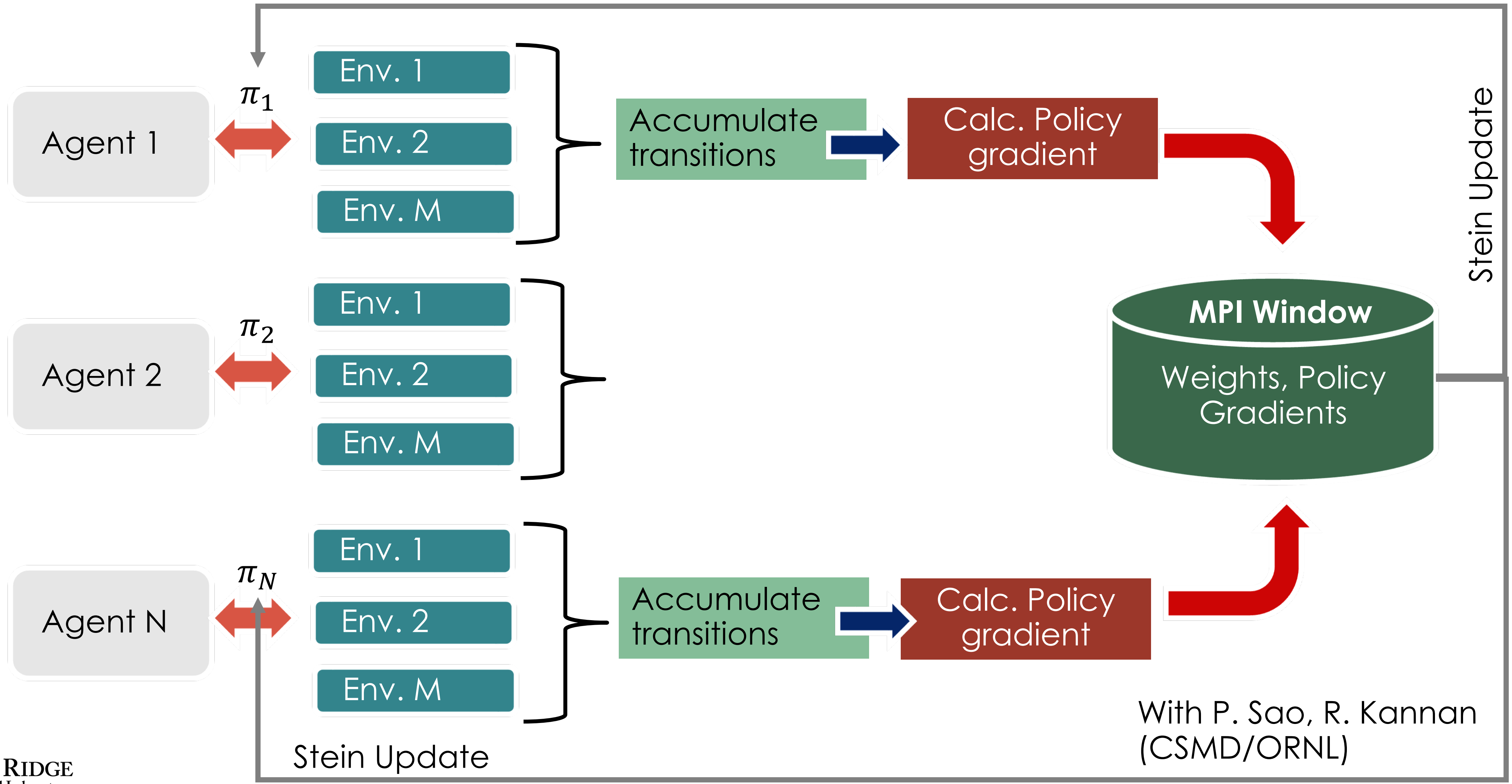
$$\phi(\boldsymbol{\theta}_i^t) = \frac{1}{S} \sum_{j=1}^{S} \left[k(\boldsymbol{\theta}_j^t, \boldsymbol{\theta}_i^t) \nabla_{\boldsymbol{\theta}_j^t} \log p(\boldsymbol{\theta}_j^t) + \nabla_{\boldsymbol{\theta}_j^t} k(\boldsymbol{\theta}_j^t, \boldsymbol{\theta}_i^t) \right]$$



Liu, Yang, et al. "Stein variational policy gradient." *arXiv preprint arXiv:1704.02399* (2017).

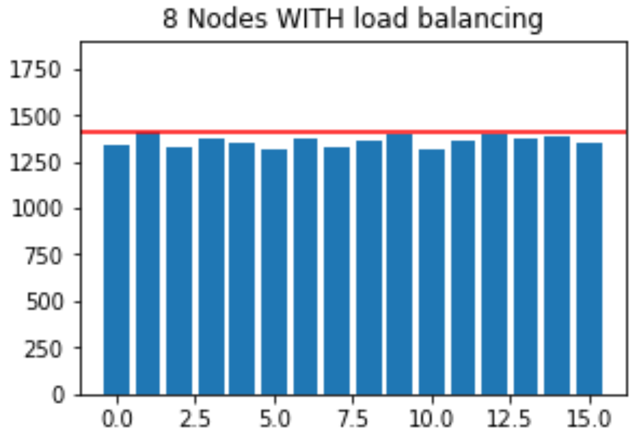
- The first term drives the policy particles towards the high probability regions of $p(\boldsymbol{\theta})$ by following a gradient ascent direction.
- The second term pushes particles away from each other diversifying the policies.
- SVPG balances exploitation (driven by policy gradient) and exploration (driven by repulsion between different policies). Thus, SVPG can learn robust and diverse policies to improve the training convergence.

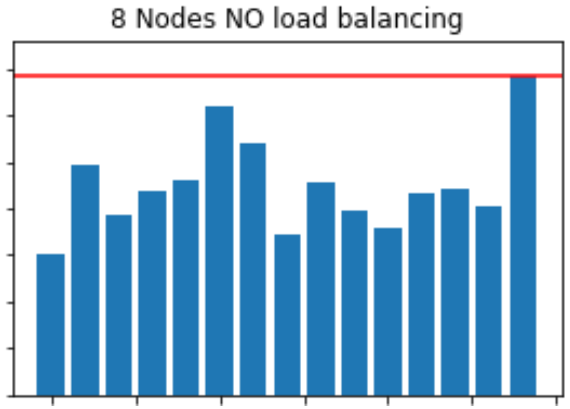
Asynchronous Load-Balanced RL Algorithm

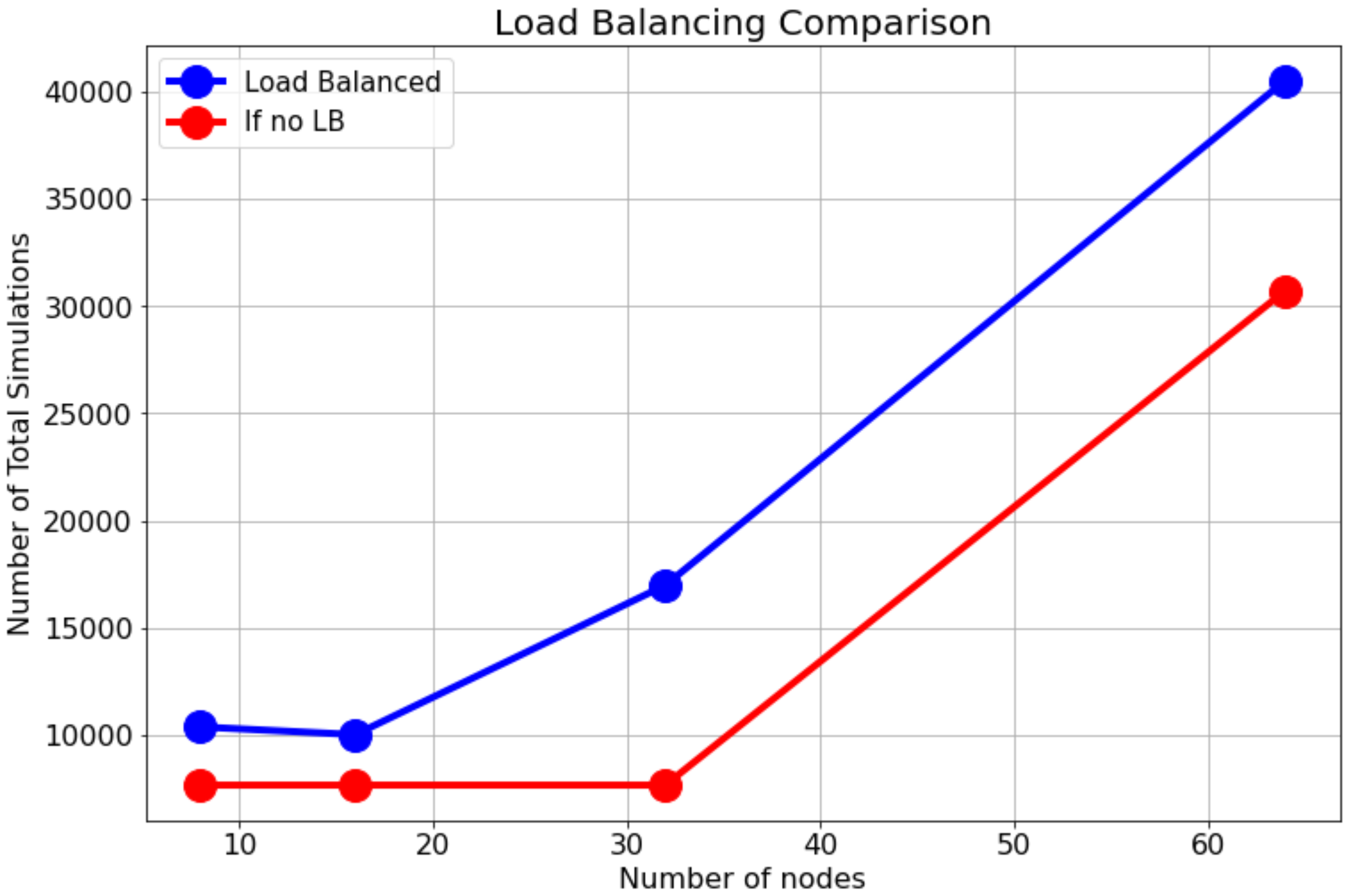


With P. Sao, R. Kannan (CSMD/ORNL)

Results: Load Balanced Asynchronous SVPG-RL



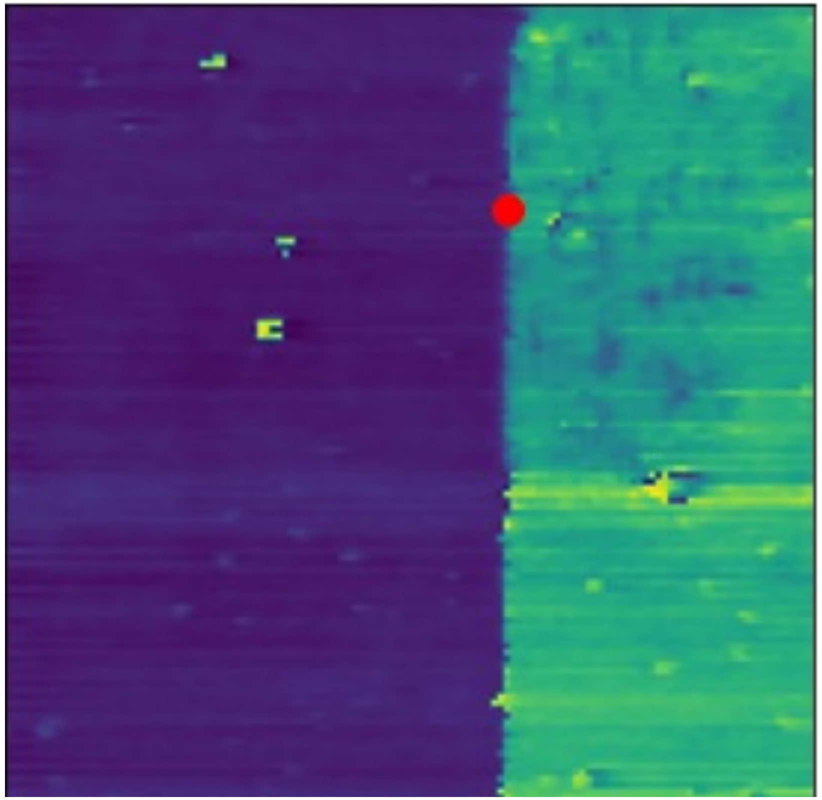




Load Balanced: 465 iterations at 3.02s/it
No Load Balanced: 320 iterations at 5.32s/i
t

Experimental workflow

Let's say we want to optimize
a sequence of bias pulses and
create regions of
high curvature at this
ferroelectric domain wall



Workflow

Write automation script

↓

Collect data on state transitions with policy

↓

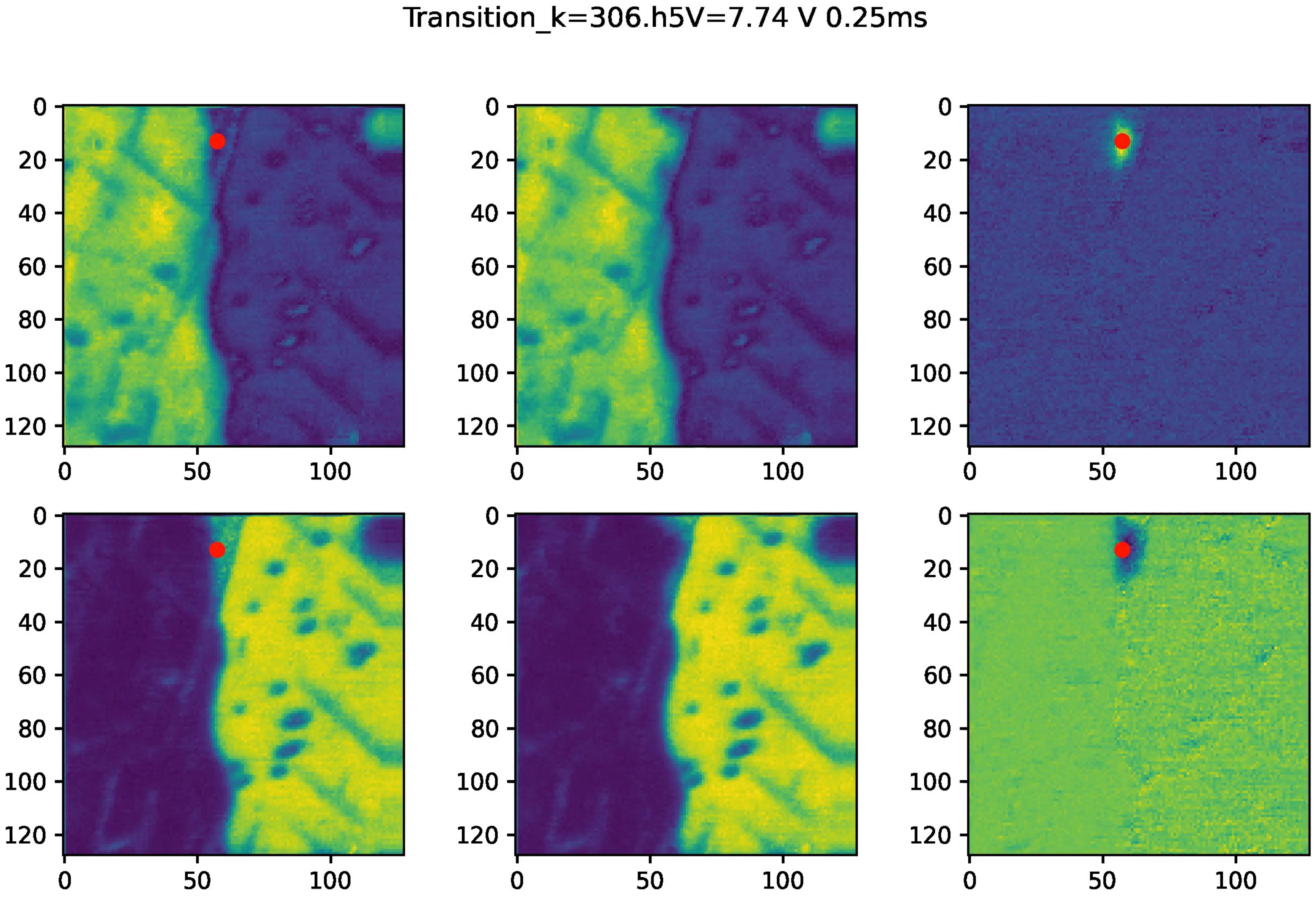
From data, train a surrogate model

Domain knowledge, Simulations →

Train RL agent, obtain policy

Update Policy

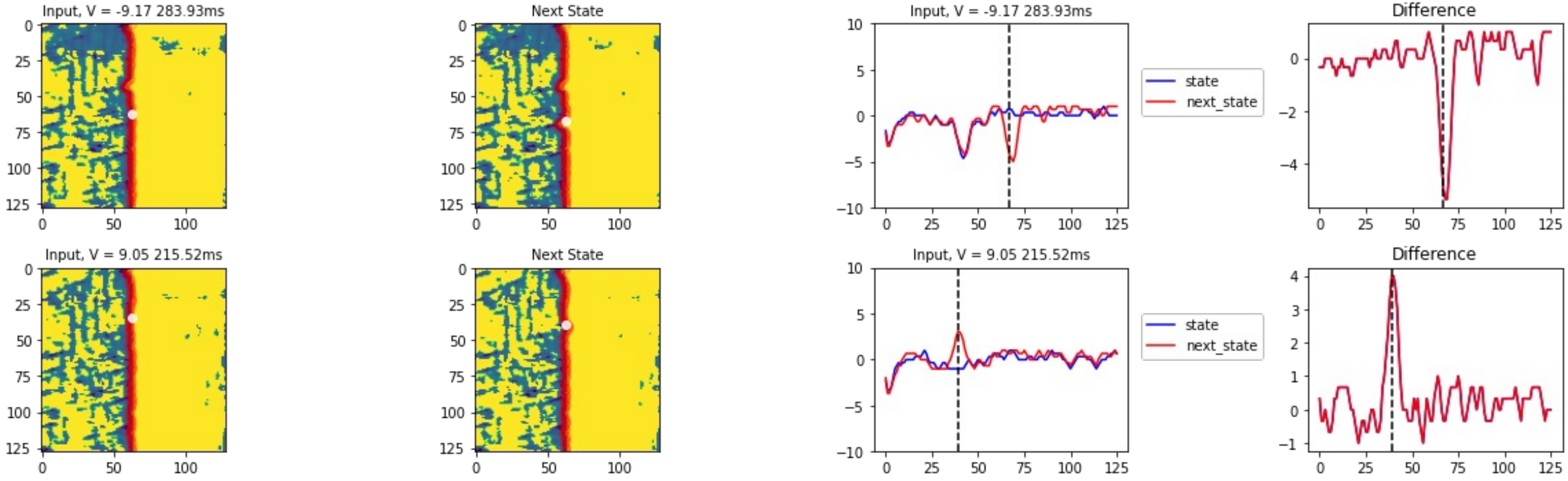
Wall Manipulation in (110) $PbTiO_3$ thin flims



Sample from J. C. Yang (NCKU/Taiwan)

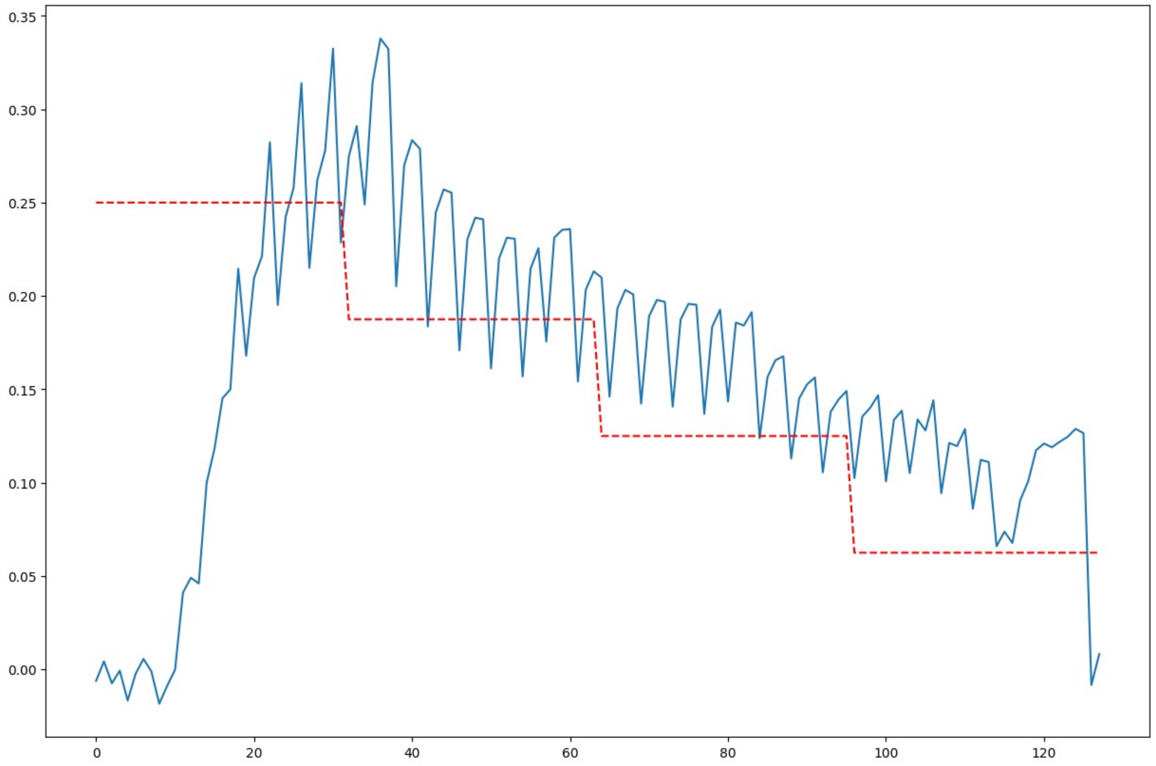
Surrogate Model Training

Example of training data

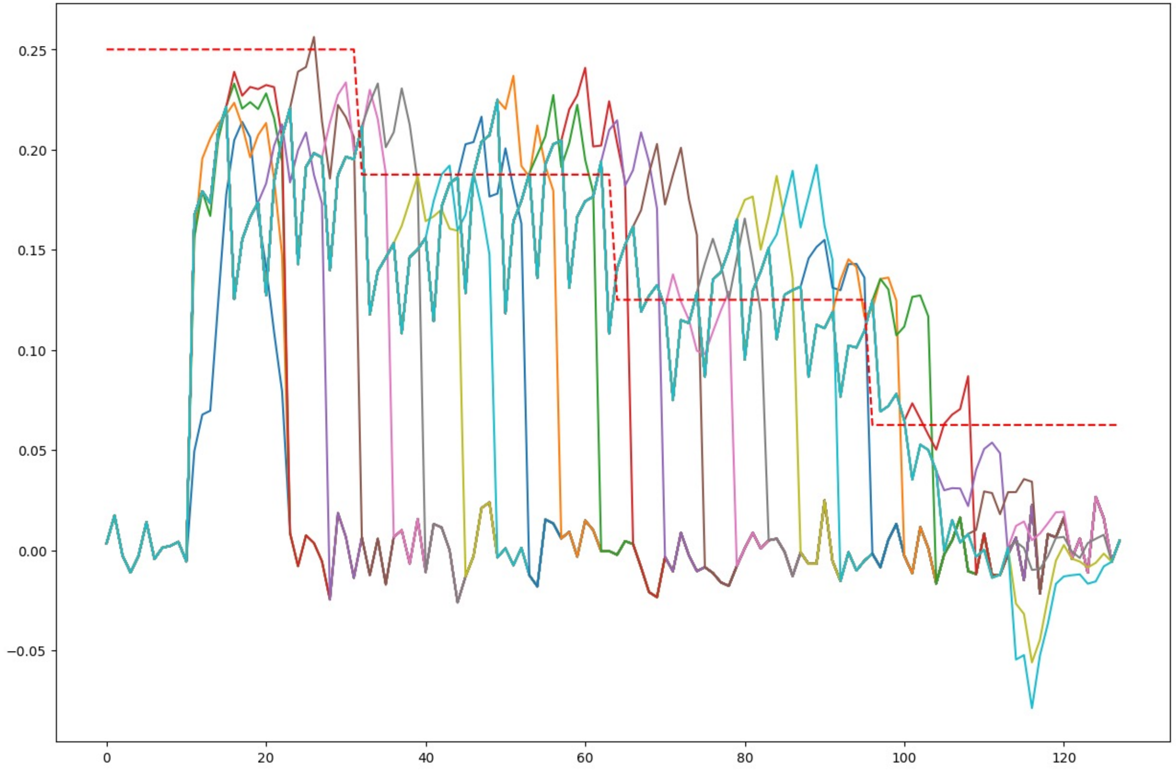


Reinforcement Learning: Autonomous Wall manipulation

HUMAN SOLUTION



RL SOLUTION



Not yet deployed on the instrument: full workflow requires better connections, more data acquisition speeds, and input from simulations

Workflow Requirements

- Algorithms, Simulations ✓
- Instruments connected to compute ✓
- Data infrastructure
- Experiment-Theory workflow orchestration

Workflow

Write automation script

Collect data on state transitions with policy

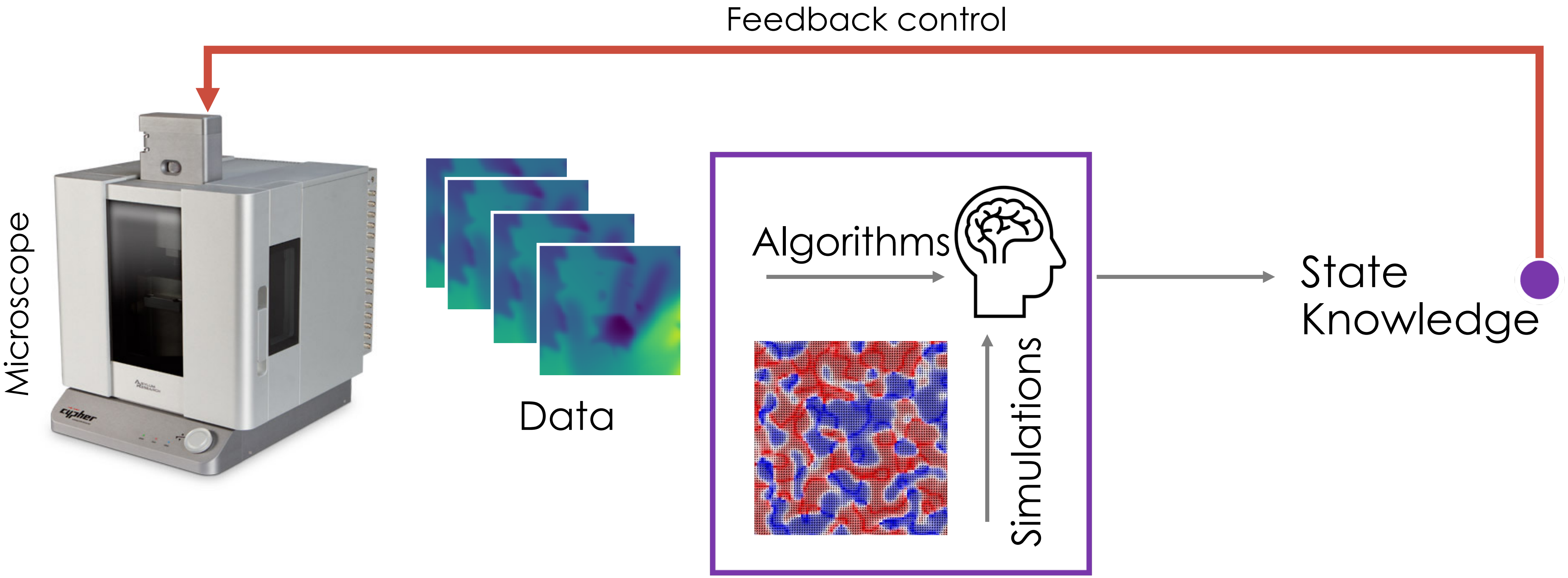
Domain knowledge, Simulations

From data, train a surrogate model

Train RL agent, obtain policy

Update Policy

Summary: “Smart” workflow necessities



Oak Ridge National Laboratory
CENTER FOR NANOPHASE MATERIALS SCIENCES

vasudevanrk@ornl.gov

Thank you