

Toward Automated X-Ray Experiments: An Inspectable Language-Model Sub-Agent for Bragg Peak Centering

David Katz¹, Elias Stengel-Eskin¹, Zhantao Chen^{2,3}

¹ Department of Computer Science, The University of Texas at Austin

² Walker Department of Mechanical Engineering, The University of Texas at Austin

³ Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin



TEXAS
The University of Texas at Austin

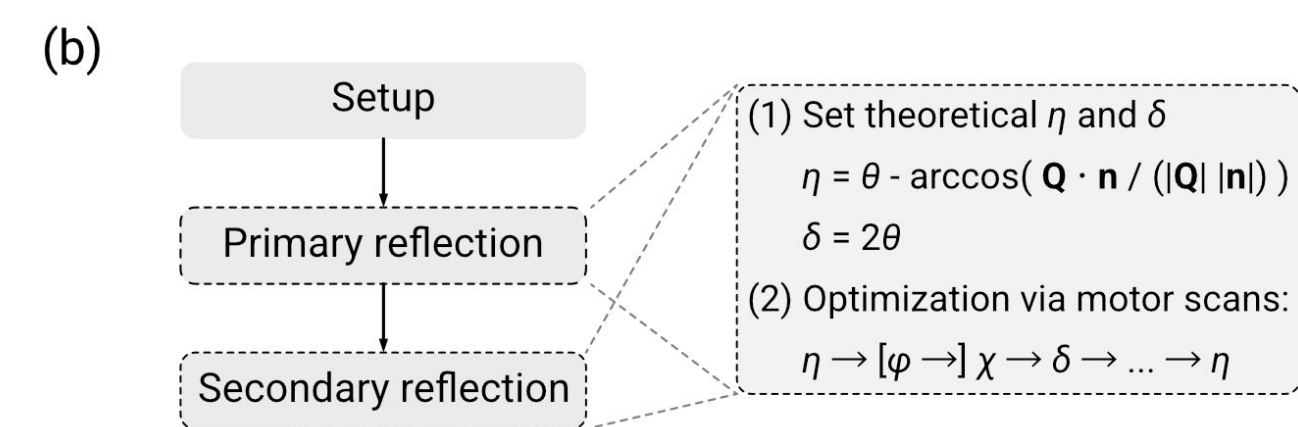
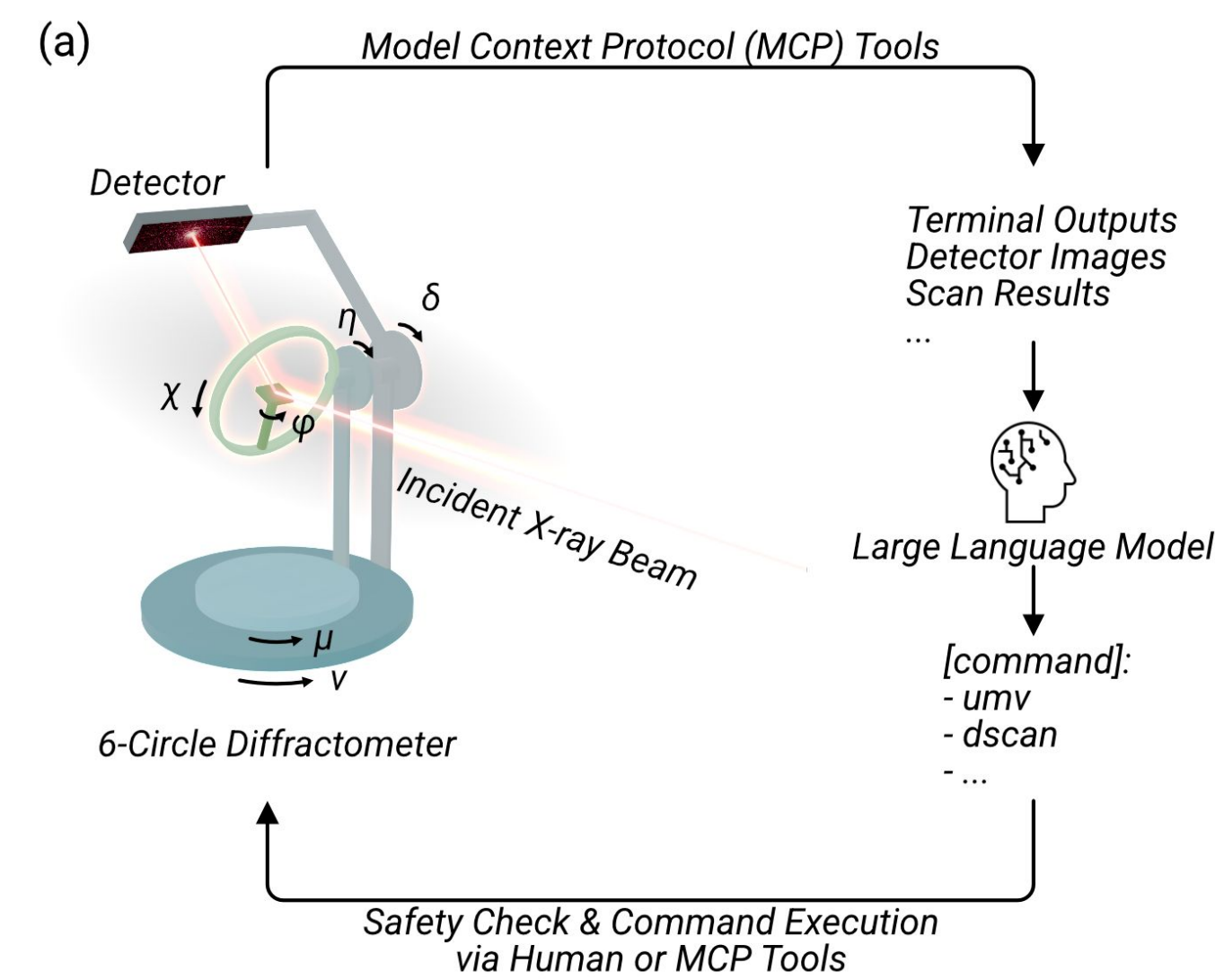
Fine-tuning enabled a small language model to center simulated X-ray diffraction spots

Why automate crystal alignment?

Bragg peaks are sharp X-ray diffraction signals that reveal the regular, repeating arrangement of atoms in a crystalline material. Crystal alignment requires repeated motor adjustments and detector checks.

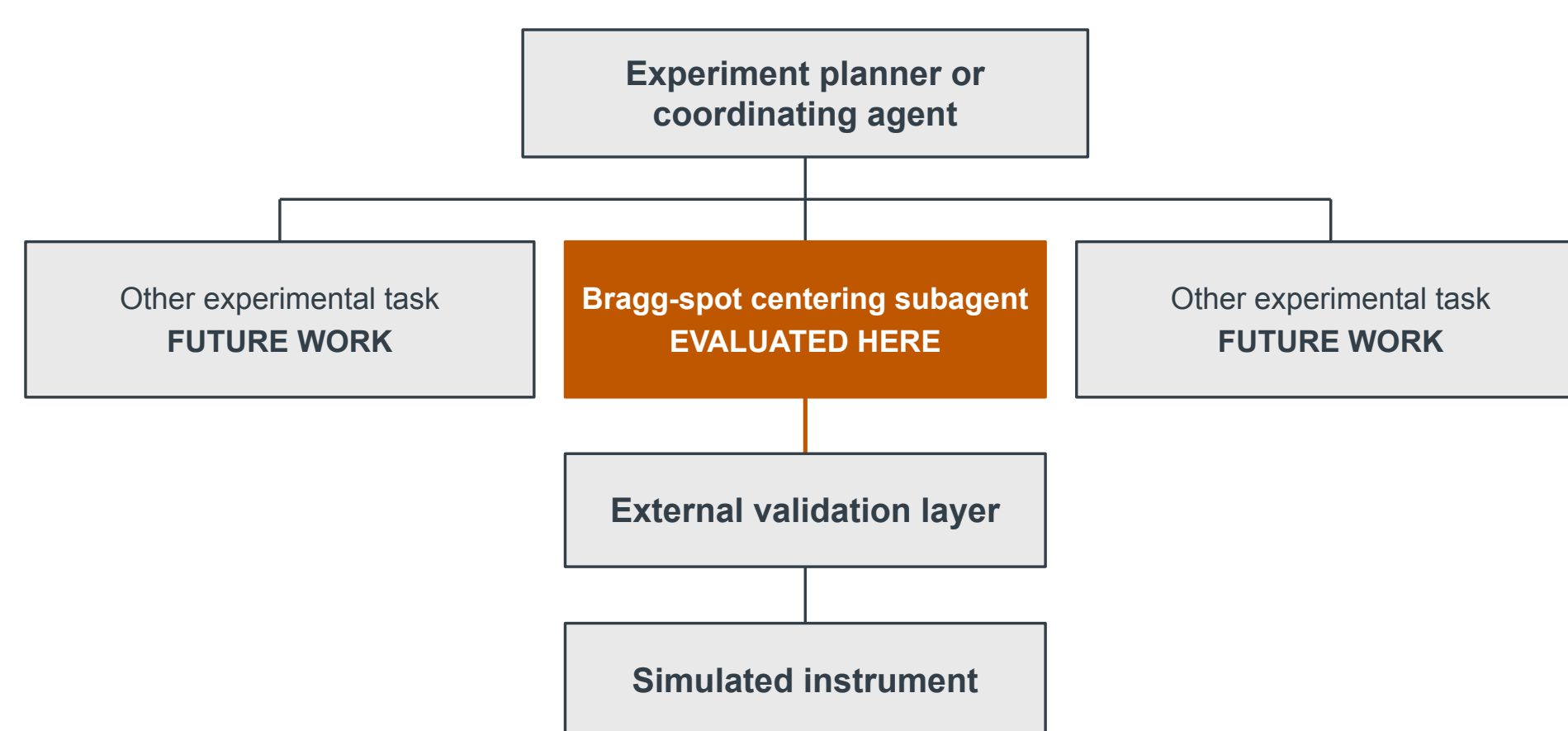
Automating these steps could reduce operator attention and improve use of experimental time.

Here, we study one alignment skill: centering a selected reflection in a simulated diffractometer.



"Prior work: AI-assisted X-ray alignment"
Source: Reprinted from Chen et al., Nat. Mach. Intell. (2026).
DOI: 10.1038/s42256-026-01261-5

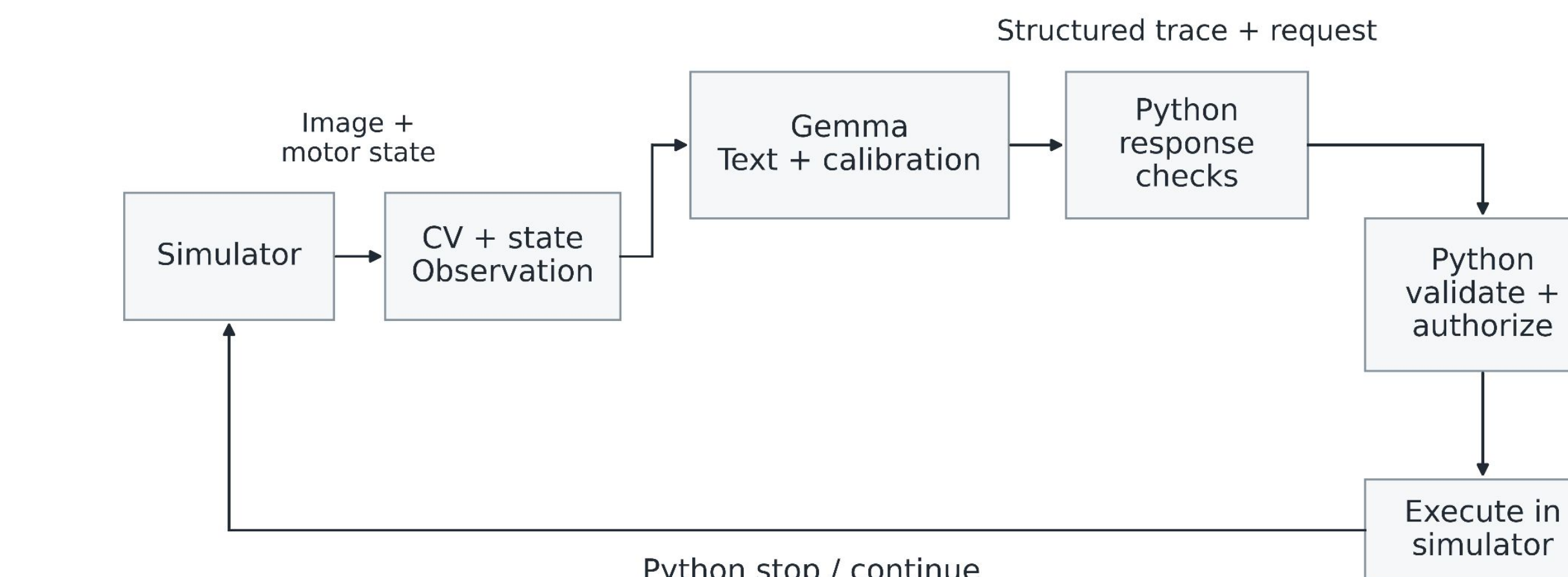
A specialist subagent for a future automated laboratory



Future automated laboratories could use a coordinating agent to plan experiments and delegate individual operations to smaller, task-trained models. Each specialist could be developed and evaluated separately, while external software checks proposed instrument commands before execution.

This study evaluates simulated Bragg-spot centering as one such skill, using a fine-tuned Gemma model and a compact neural-network baseline. Combining multiple specialists into complete experimental workflows and transferring their skills across instruments remain future work.

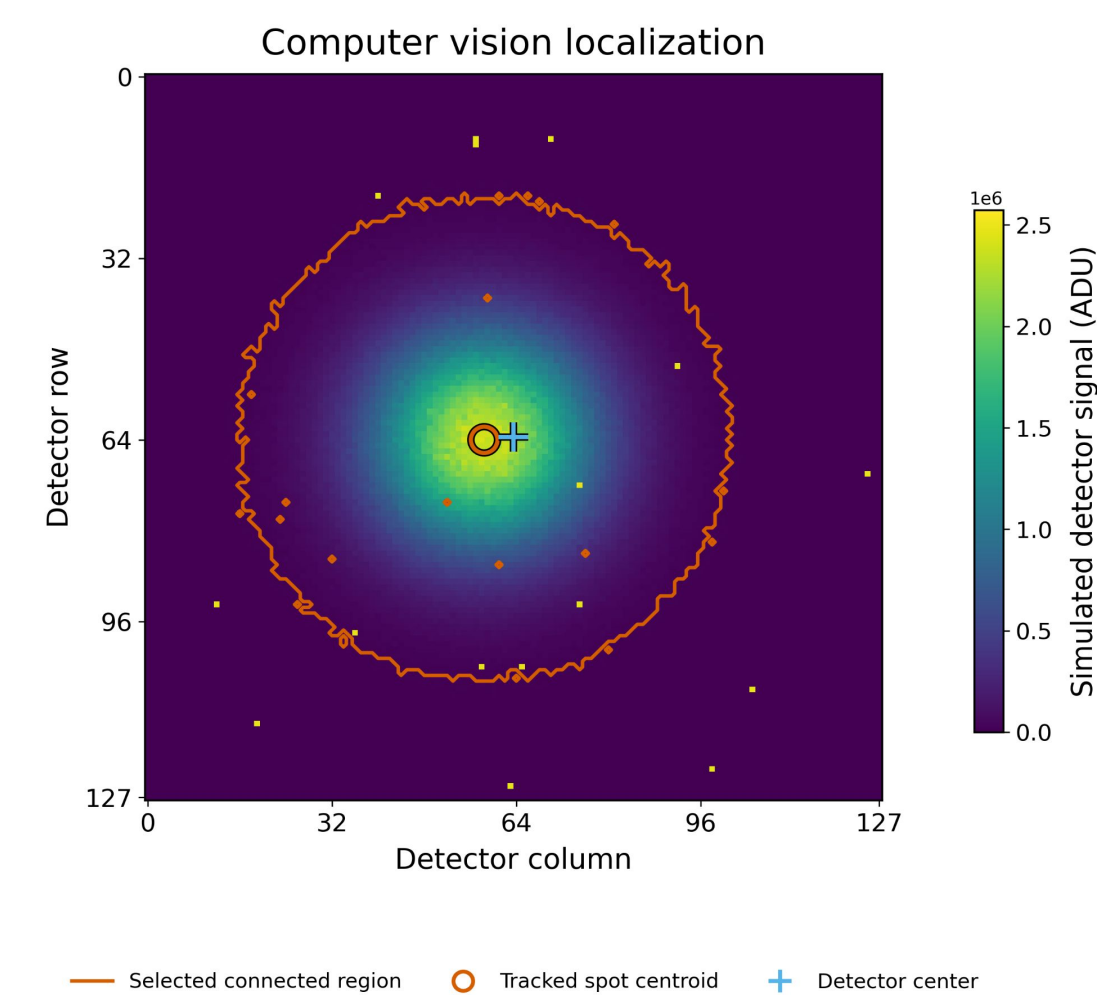
How the centering subagent works



Computer vision measures and tracks the spot's intensity-weighted center.

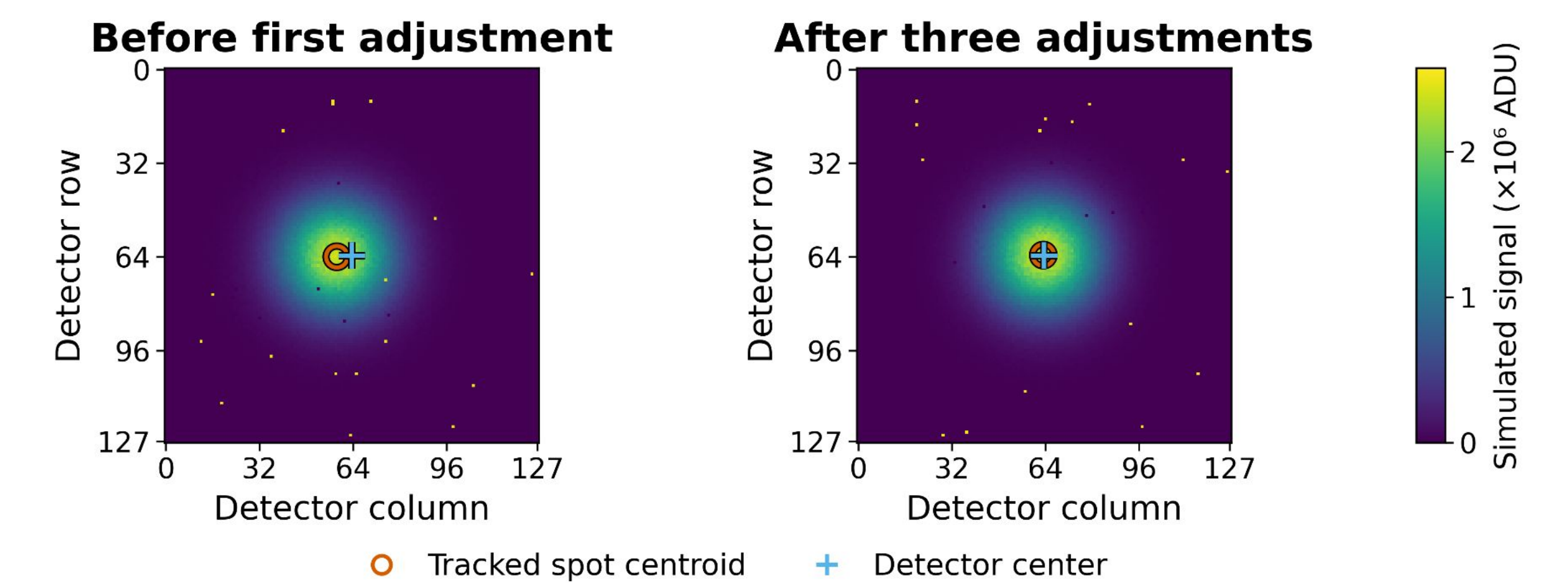
Gemma receives 32 observation and history fields as text, plus calibration instructions. It does not receive the detector image.

Gemma proposes relative delta and chi movements. Python validates each request, and the simulator executes accepted movements.



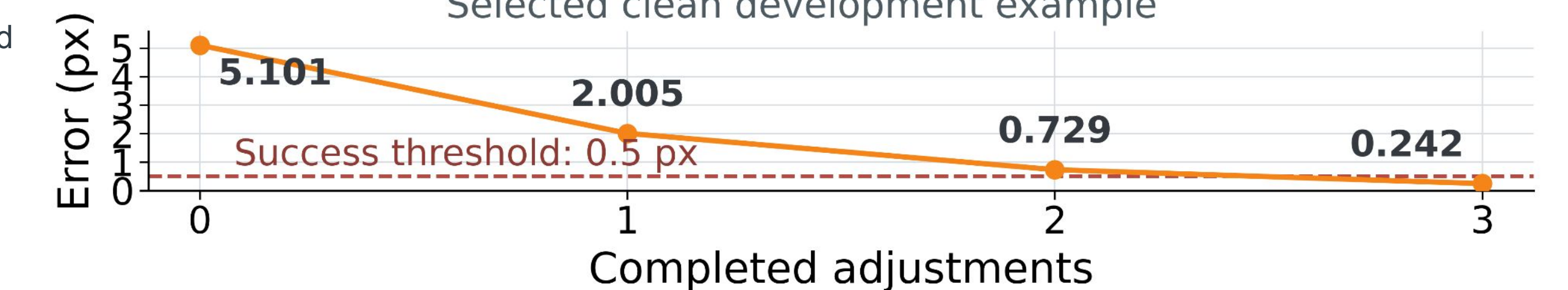
Simulated material: $\text{Co}_3\text{Sn}_2\text{S}_2$
Detector: 128×128 pixels
Controlled axes: delta and chi
Success: centering error ≤ 0.5 pixel
Budget: up to 20 action attempts

One recorded SFT centering episode



Recorded SFT centering episode

Selected clean development example



Illustrative clean-development episode. Detector images were reconstructed from recorded states and checked against archived centroid measurements.

First adjustment: recorded model output

```
strategies delta=nominal_correction,  
chi=nominal_correction;  
plan delta=0.042951, chi=-0.61151  
within observed limits.
```

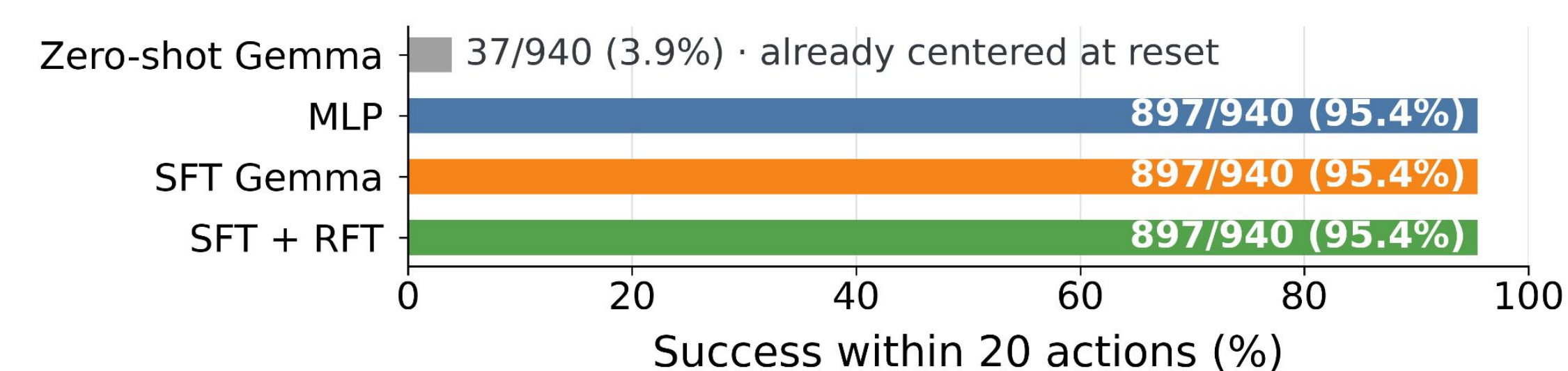
Python accepted the request and the simulator executed it. Centering error fell from 5.101 to 2.005 pixels after this adjustment.

Movement values are relative rotations in degrees.

Controller performance in simulation

Fine-tuning enabled simulated centering

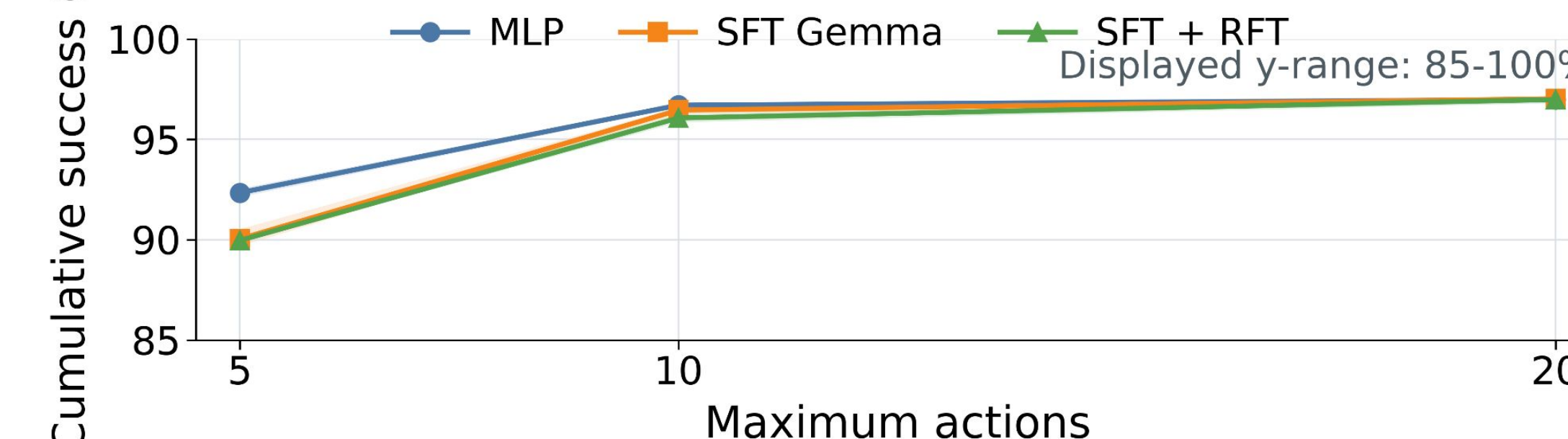
Clean execution · clean-trained checkpoints



All nine trained clean runs reached 89.7/940 by 10 actions.

Mixed-fault success across action budgets

Fault-trained checkpoints · three seeds per trained controller



Development evaluation: 940 paired cases per condition from 43 initialization groups. Trained controllers used three training seeds. Mixed-fault lines show seed means and shading shows the observed seed range. Clean and mixed-fault results use separately trained checkpoints. Simulated motor faults reduced requested movements or prevented movements below a threshold.

Controllers and training

Compact MLP

A 6,402-parameter network trained on numerical teacher movement targets.

Zero-Shot Gemma

Gemma 4 E4B-it received calibration instructions without a trained adapter.

SFT Gemma

Supervised fine-tuning (SFT) with a rank-16 LoRA adapter on structured examples generated by a deterministic controller.

SFT + RFT

The SFT checkpoint received additional reward-guided tuning using model-generated control attempts.

The trained controllers used three independent training seeds.

A deterministic controller generated training examples pairing observations with motor adjustments. The MLP learned numerical movement targets, while SFT Gemma learned structured responses containing those movements. RFT then used feedback from Gemma's own simulator attempts.

Why study a language-model controller?

- A text interface can combine measurements with calibration instructions and experimental constraints.
- Recorded model responses let scientists inspect proposed adjustments before execution.
- Small open-weight models offer a path toward local deployment as specialist laboratory agents.

The MLP remains a strong baseline for this fixed task. The next question is whether a language-model controller can use revised calibration instructions to adapt to a different instrument with less retraining.

What this study shows

- Task-specific training enabled control.** Zero-shot Gemma made no successful off-center corrections. Supervised fine-tuning enabled it to complete centering through repeated observations and motor adjustments.
- Fine-tuned Gemma achieved similar development success to the MLP within 20 actions.** The MLP remained a strong baseline and achieved higher success under mixed motor faults at the five-action budget.
- Additional RFT did not improve aggregate success within 20 actions.** In this study, supervised fine-tuning provided the main improvement in controller performance.
- Model proposals are recorded and checked before execution.** Gemma requests motor adjustments using the measured observations. Python validates each request before the simulator executes it, and the recorded responses let researchers review the proposed actions.

**Group for AI in Materials
Modeling and Analysis**

Computing at TACC

- Vista CPU workloads supported simulation, teacher-data generation, and data preparation.
- Grace Hopper GPUs supported Gemma fine-tuning and inference.
- Slurm arrays distributed evaluation across controllers, conditions, and training seeds.

The authors acknowledge the Texas Advanced Computing Center at The University of Texas at Austin for providing computational resources that contributed to this research.

TACC
TEXAS ADVANCED COMPUTING CENTER

Next steps and project links

- Test transfer across simulated instruments and calibration changes.
- Evaluate unfamiliar starting conditions and investigate tracking failures.
- Extend toward multi-agent workflows and physical-instrument validation.

Methods, results, code, and team contacts

David Katz | davidkatz@utexas.edu |
TACCSTER 2026 | September 21–22, 2026

