Brookhaven National Laboratory

U.S. DEPARTMENT of ENERGY

Towards Autonomous Labs:

Agentic Optimization and Active Surrogate Modeling for CCD Characterization

Prashansa Mukim, Instrumentation Division, Brookhaven National Laboratory

on behalf of the CCD collaboration at BNL and the University of Chicago

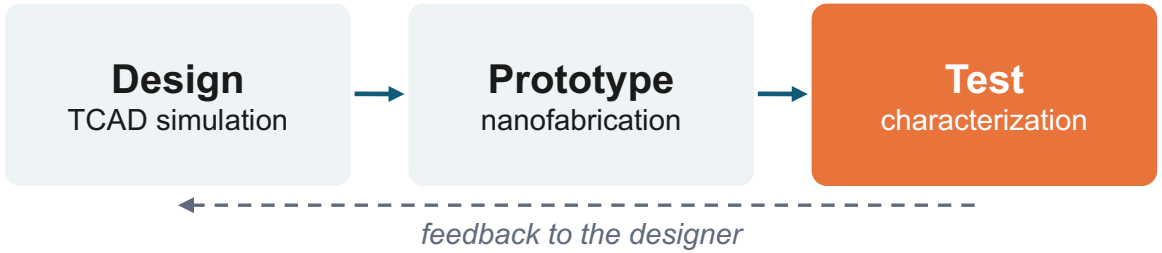
Contributions: Juan Estrada, Julian Cuevas-Zepeda (U. Chicago), Giovanni Pinaroli, Emi Tamura, Santiago Perez (U. Chicago), Anibal Boscoboinik, Hendrik Hamann, Celia Xu, Chuntian Cao, Shane Michtavy, Dominik Wierzbicki

CPAD 2026 · Seattle · October 21, 2026

@BrookhavenLab

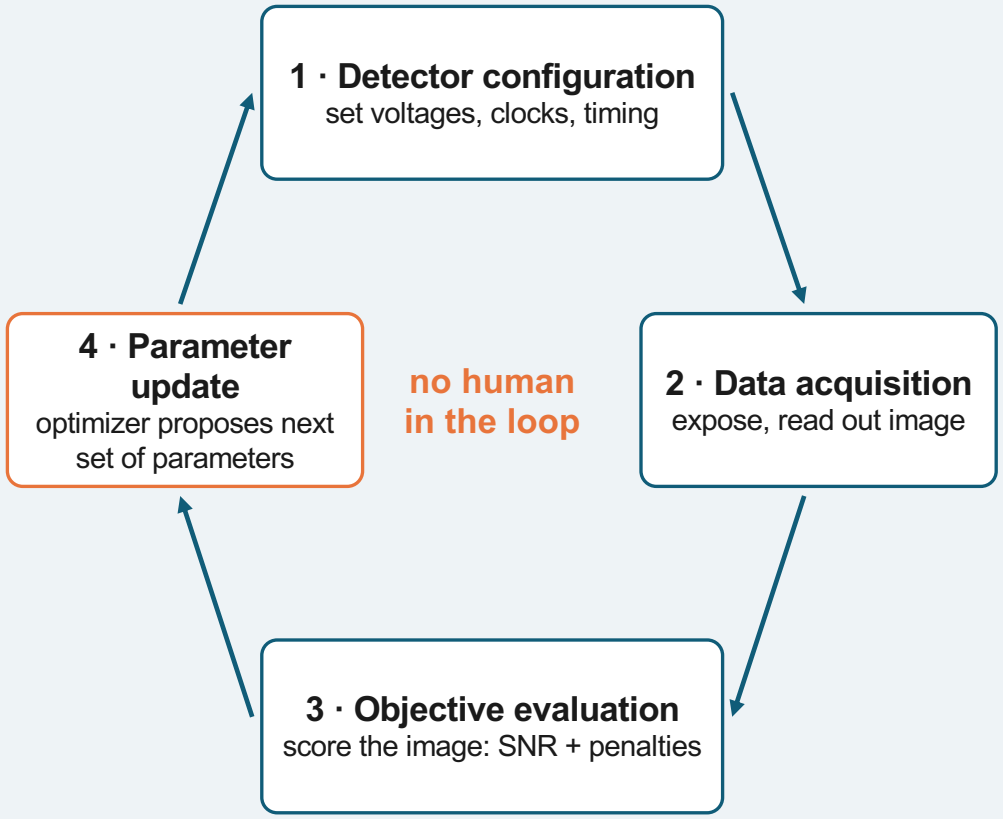
Detector characterization framework

Sensor development cycle

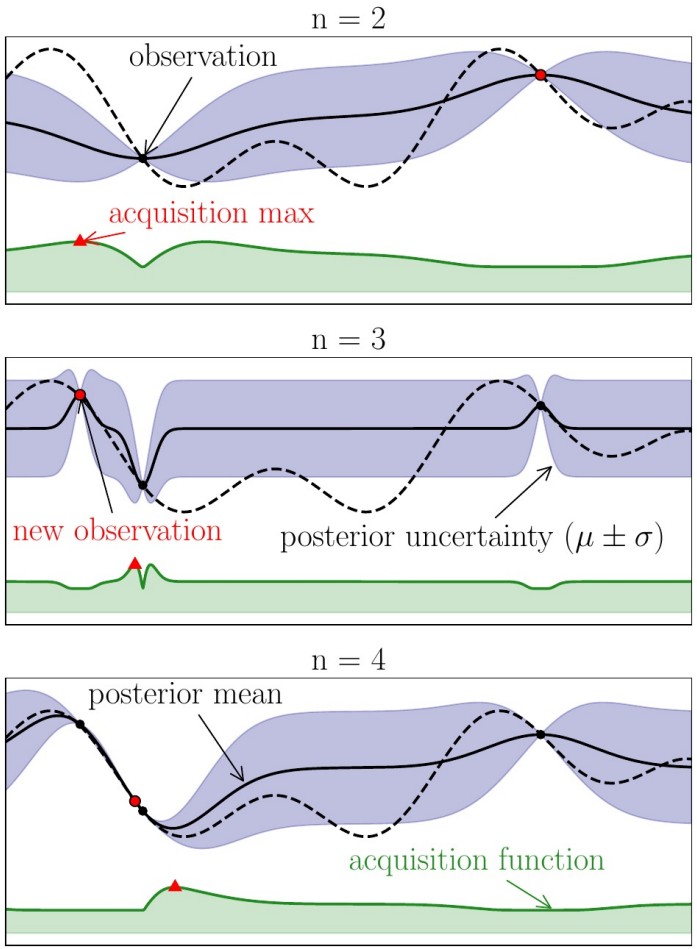


- **Sequential scanning** of a predefined parameter space, with expert oversight guiding every step
- **Strongly coupled parameters:** bias voltages, clock levels, timing; one-at-a-time scans miss interactions, full grid exploration time consuming
- **Image saved for a set of parameters:** set parameters → expose → read out → evaluate SNR, uniformity, saturation, charge-transfer artifacts
- **A year or more** of an expert's time per device; next-generation sensors have more tunable parameters and tighter noise targets

The closed-loop framework



Bayesian optimization (BO) as a baseline optimizer



GP posterior mean ± σ (shaded) tightens with each observation; the acquisition function (green) picks the next point.

B. Shahriari et al., Taking the Human Out of the Loop: A Review of Bayesian Optimization.

Objective: SNR plus penalties (lower is better)

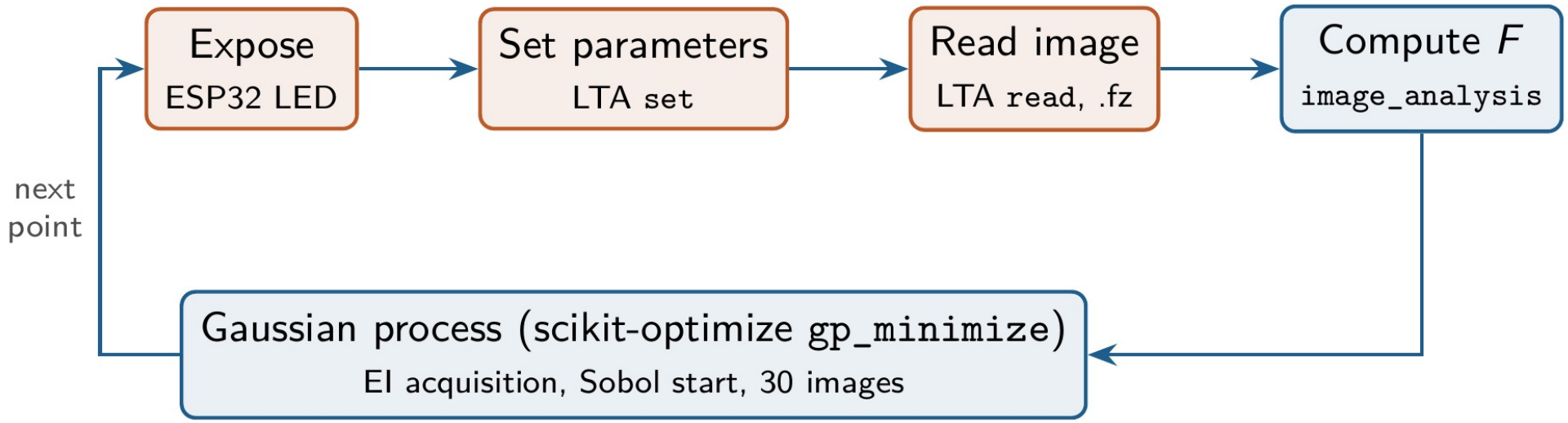
$$F(x_i) = \frac{1}{\mathrm{SNR}(x_i)} + \sum_{j=1}^{5} \alpha_j P_j(x_i), \qquad \mathrm{SNR} = \frac{S}{N}$$

Penalty	Catches	α
$P_1 = \max(0, -S)^{1/3}$	negative gain	1/2
$P_2 = \max(0, N - S)^{1/3}$	noise exceeds signal	1/2
$P_3 = \max(0, 2\sigma_{rev} - \sigma_{active})$	saturation / clipping	1/3
$P_4 = \max(0, U - 1)$	non-uniformity (5 strips)	1/250
$P_5 = \max(0, \mathrm{med}_{ov} - \mathrm{med}_{rev})$	clock-induced charge	1

- **Surrogate:** a Gaussian process models F(x) with uncertainty, updated after every image
- **Acquisition function:** scores candidates by predicted F and expected uncertainty reduction; its maximum is the next measurement
- **Why BO:** derivative-free (noisy F, threshold penalties); explicitly balances exploration and exploitation, sampling uncertain regions more systematically than local search; fewer evaluations than evolutionary algorithms

J. Cuevas-Zepeda et al 2026 JINST 21 P10012.

Baseline framework based on Bayesian optimization



Objective (lower is better)

$$F = \min\left(\frac{\sigma_{\mathrm{os}}}{|\mathrm{gain}|},\ 1000\right) + \sum_i c_i\,\mathrm{penalty}_i^{p_i}$$

gain = median(active) − median(overscan), σ_{os} = std(overscan)

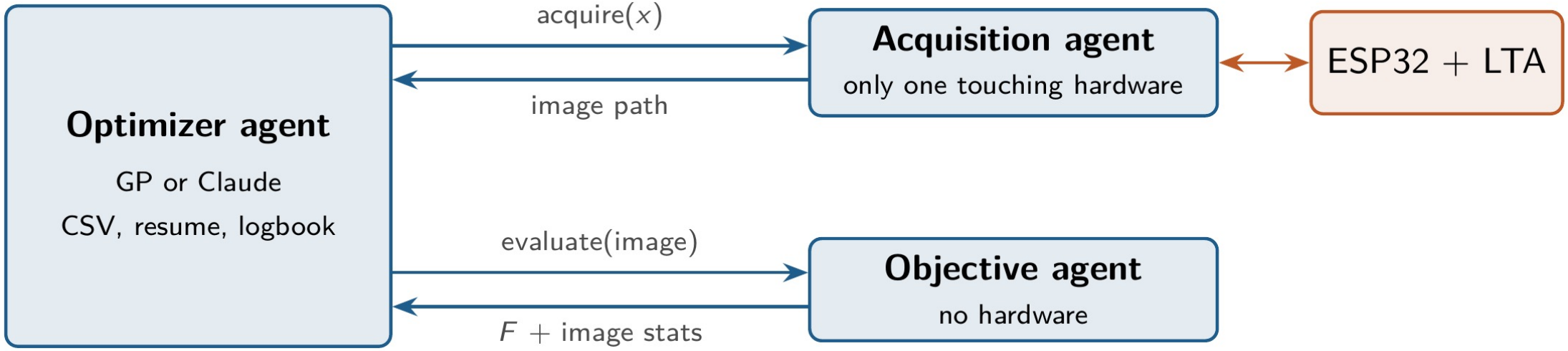
F tracks read noise in units of signal; penalties available but all off

Note the |·|: an inverted signal (negative gain) scores like a real one

Gaussian-process optimizer

- scikit-optimize gp_minimize, expected-improvement acquisition
- 8 Sobol points + 22 guided = 30 images per campaign
- Sees only the parameters and F

Agentic framework based on three cooperating agents



Acquisition agent

The only agent interacting with hardware (ESP32 LED, LTA): sets parameters, exposes, reads the image

Objective agent

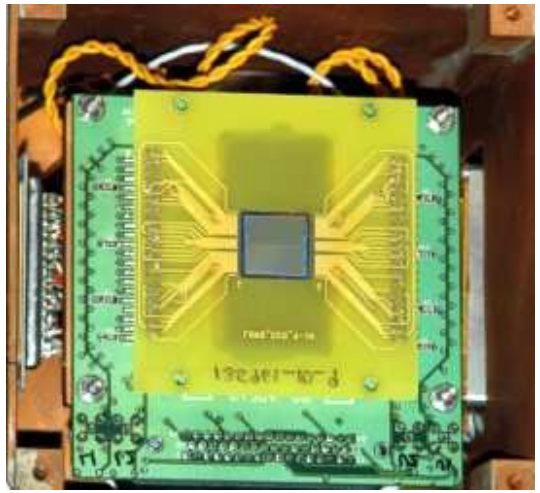
No hardware. Computes F and the image statistics (signal, noise, gain, etc.) for every image

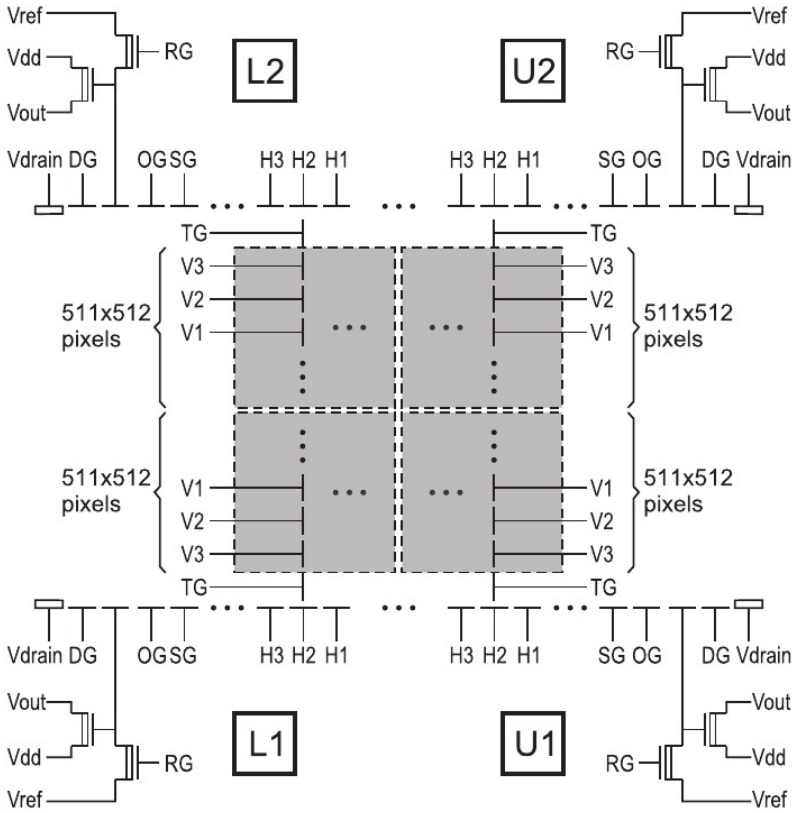
Optimizer agent

GP or Claude. Chooses the next point; keeps CSV, resume state and the lab logbook

Each agent is its own process; messages are plain JSON, all logged.

Agentic parameter optimization: applied to a Skipper-CCD





Skipper-CCD in its package (left) and four-amplifier architecture with floating-gate output stages (right).
G. Fernández Moroni et al., arXiv:1106.1839.

- **Goal:** find the operating point that gives the best images
- **Each evaluation is based on a measurement:** expose (LED) → set parameters → read out → score

Parameter	Role	Range
Vdd	output amplifier drain	[−23, −10] V
Vdrain	dump drain	[−23, −10] V
Vr	reset reference	[−8, −5] V
delay_H_overlap	horizontal clock overlap	[10, 30] clk

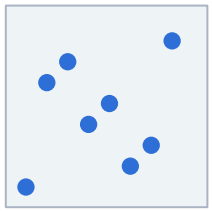
Optimizer works in $[-1, 1]^4$; values mapped to physical units

Replacing the GP with an LLM agent (Claude)

1

Sobol start

8 space-filling points, identical to the GP runs



8 Sobol points

2

Claude chooses every further point

Claude Opus 5.5 via the Claude API (claude-opus-5-5) receives: parameter descriptions and bounds, the objective definition, all measurements in physical units, the image statistics (not seen by GP), the amplifier in use, operator notes and the logbook

3

Structured JSON answer

Next point, clipped to the bounds; thinking summary, tokens and cost logged for every image

Config

```
"optimizer": {
  "type": "claude",
  "n_calls": 30,
  "n_initial_points": 8,
  "model": "claude-opus-5-5",
  "effort": "high" }
```

What Claude sees each turn (real rows, Oct 1)

```
Measurements so far (18):
#   | Vdd   | Vdrain | Vr   | delay | F      | gain
9   | -14.5 | -18.8  | -6.3 | 21    | 0.0472 | 14480
18  | -17.5 | -20.2  | -5.0 | 30    | 0.0242 | 28320
...
Measurements left in this campaign,
including the next one: 12.
Choose the next point.
```

Excerpts from the agent's per-image reasoning

Using memory: re-checking a remembered best point

"The notebook's recorded best is (-14.5,-18.8,-6.3,21), so I should measure that point directly to verify reproducibility and use it as a reference."

Reasoning about noise and bounds

"The repeat of trial #18 showed noticeable noise (~0.01 variation) […] exploring Vr is valuable since we're sitting at a boundary value […] to confirm whether -5.0 is truly optimal or just unexplored territory."

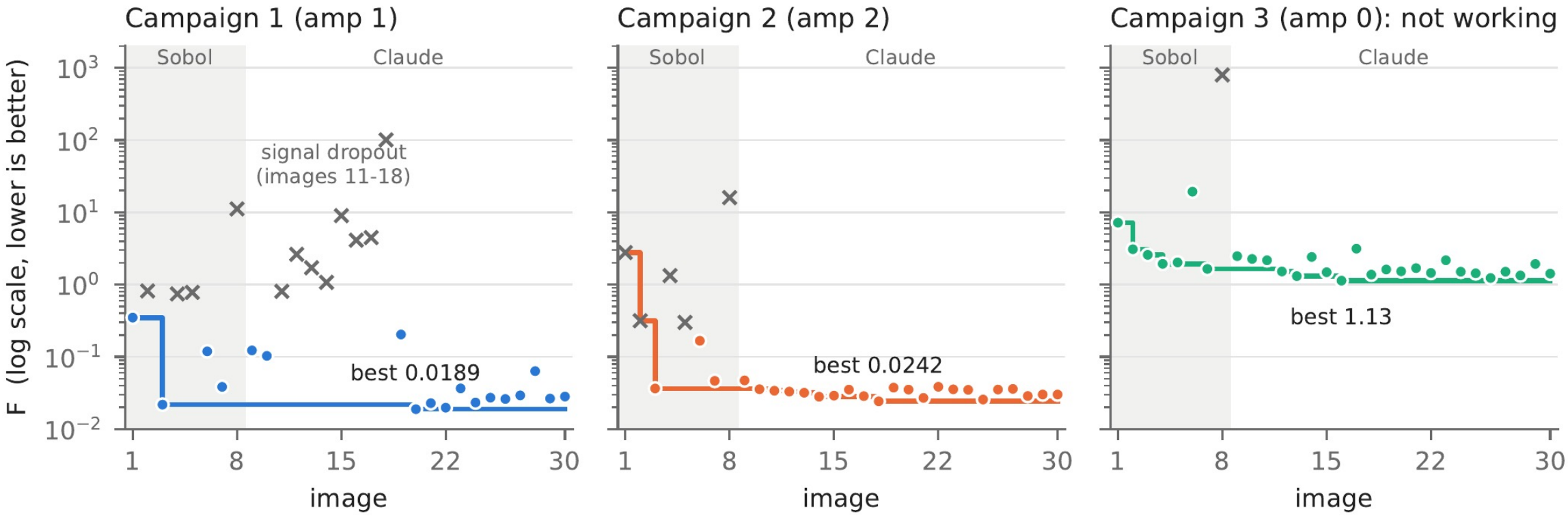
Noticing that the setup changed

"These conditions look drastically different from before – noise is much higher, F values are now in the 1.6-3 range instead of 0.02-0.03 […]"

Learning what a valid image is, unprompted

"Negative gain (#2, #3, #5, #7): charge_active was below charge_overscan, which is not a valid LED response. F uses |gain|, so these look deceptively moderate […] Treat any gain < 0 as a failure."

Convergence trajectory (within a few images after the Sobol start)



8 + 22

Sobol start, then 22 points chosen by Claude in every campaign

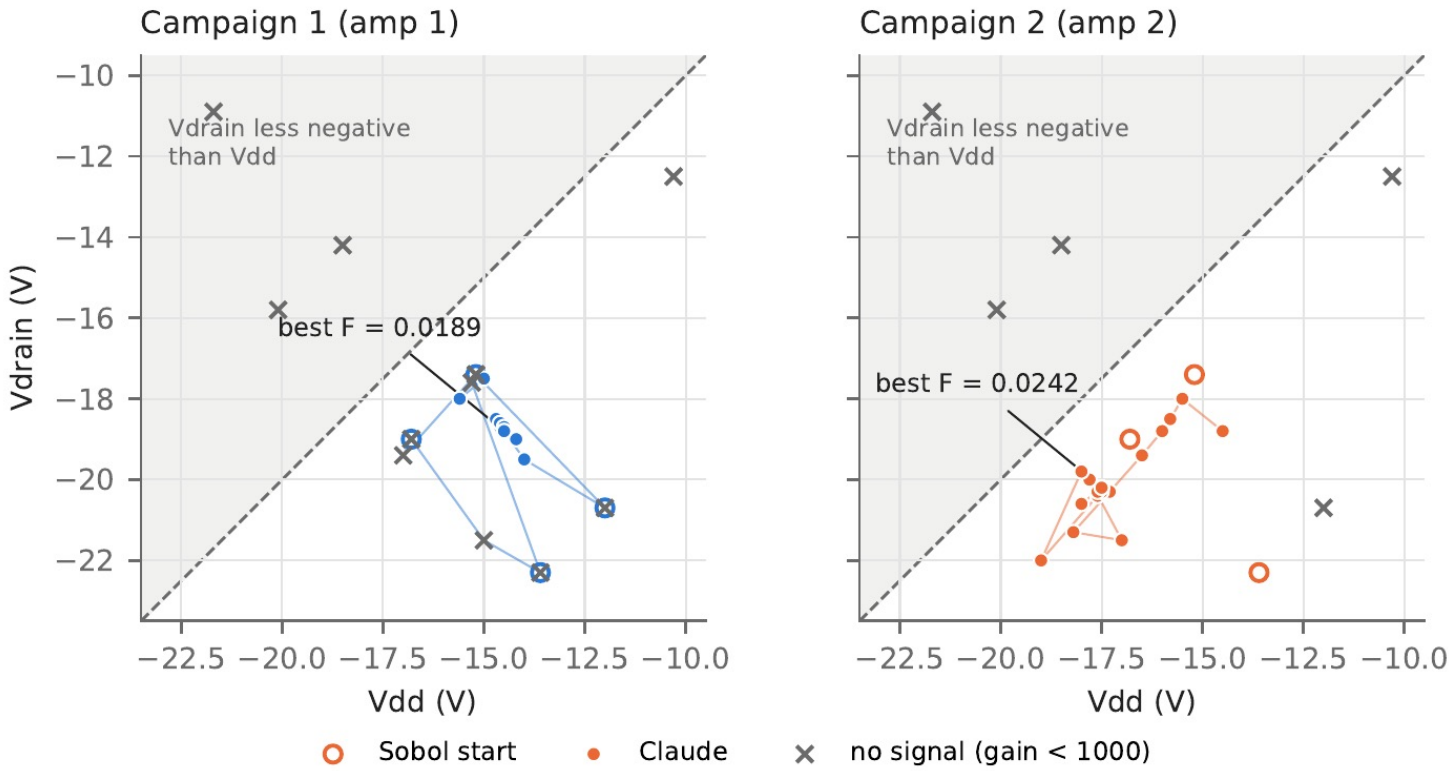
0.019 / 0.024

best F on amplifiers 1 and 2; Claude then refines and re-measures

50×

higher overscan noise on amplifier 0: not working, and Claude flagged it

Reasoning about the Vdd–Vdrain relationship



- **Shaded:** Vdrain less negative than Vdd ⇒ negative gain
- **Good points:** Vdrain 2.5–3 V more negative than Vdd

Agent's conclusion in the logbook

"F is driven almost entirely by gain […] most sensitive: the Vdd/Vdrain relationship"

Open circles: Sobol start; dots: Claude; ×: no signal (gain < 1000)

Comparisons of GP vs. LLM Agent

	Gaussian process	Agent (Claude)
Best F, any sign of gain	**0.769 (gain −1054 ADU) (inverted!)**	**0.428 (gain 1590 ADU)**
Best F with gain > 0	0.920 (gain 850 ADU)	**0.428 (gain 1590 ADU)**
Images with gain > 0	14 of 30 (10 of 22 guided)	**26 of 30 (all 22 guided)**
Best point (Vdd, Vdrain, Vr, delay)	(−23, −23, −6.7, 30)	(−23, −10.5, −7.4, 10)
Time / cost	8 min / free	10 min / $0.28

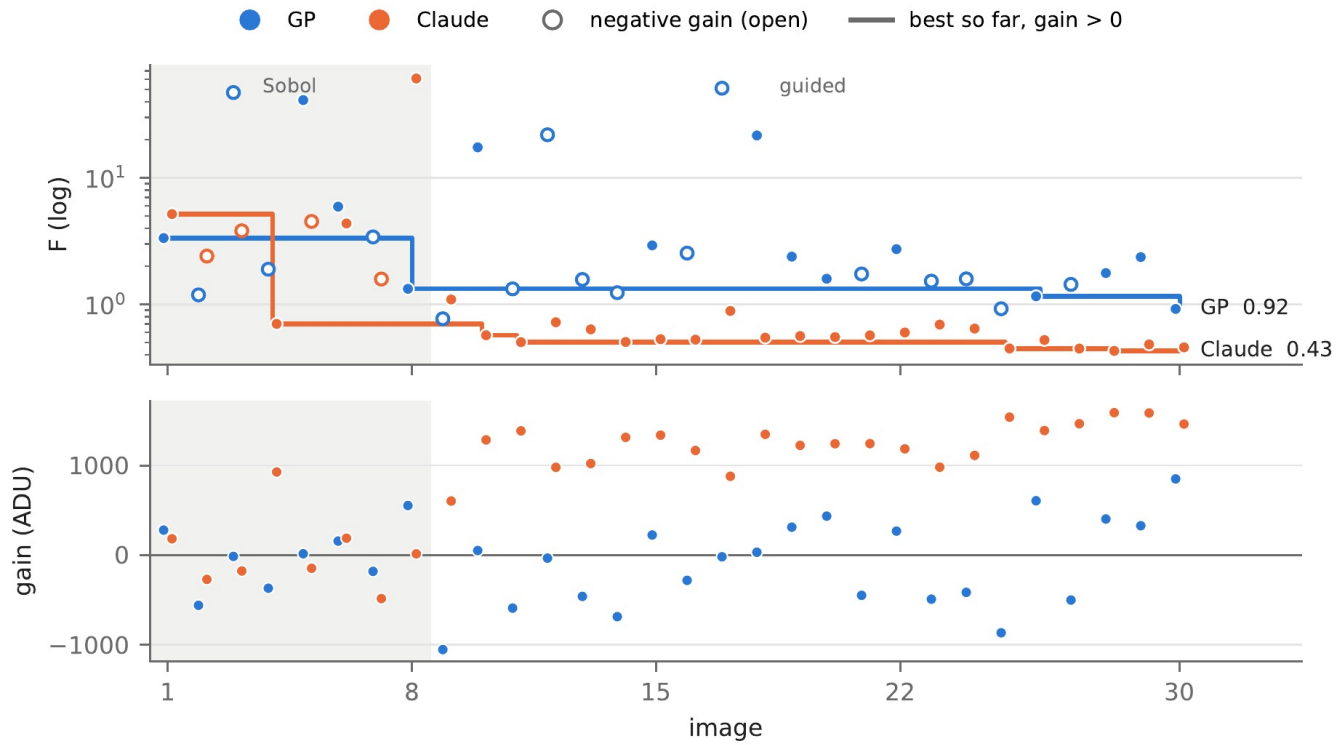
Negative-gain problem flagged:
F uses |gain|, so the GP's nominal best image is an inverted signal. Ranking only valid images (gain > 0) is the fair comparison.

~2×

lower F among valid images (0.920 → 0.428), with ~2× more signal

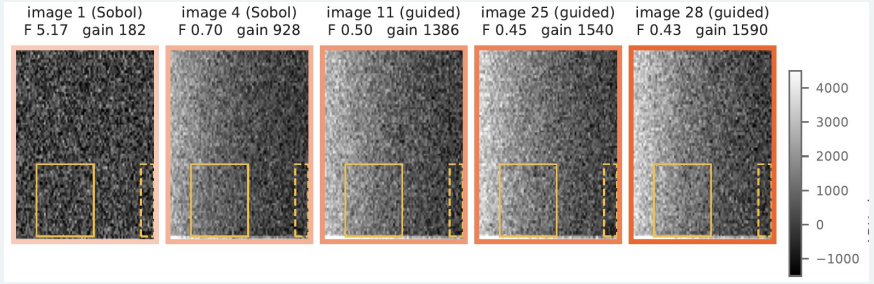
30 images each (8 Sobol + 22 guided) · Claude without logbook

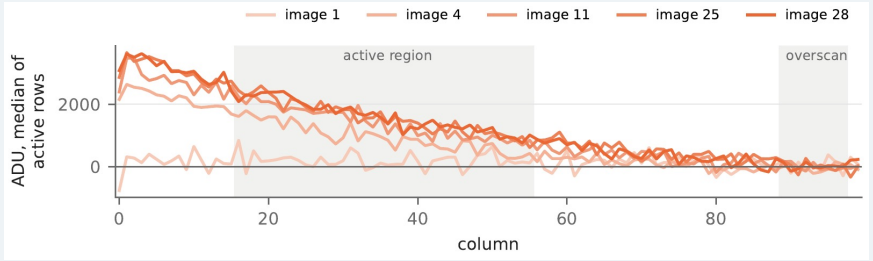
Convergence trajectories: GP vs. LLM agent



Top: F per image; line = best so far among gain > 0 images. Bottom: signal (gain). Open circles: negative gain

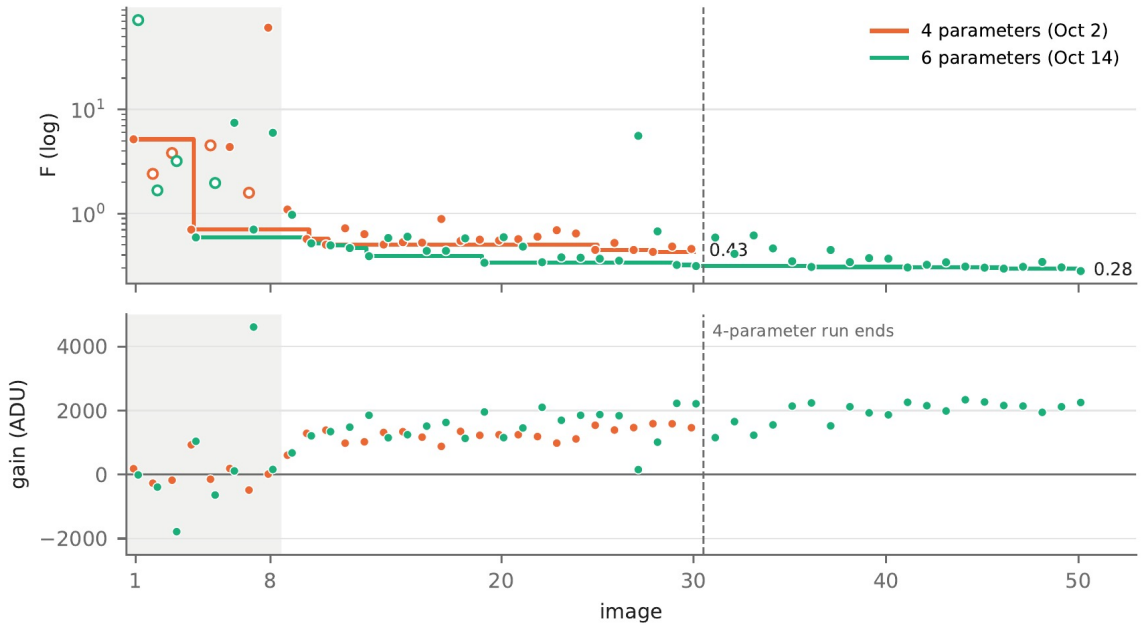
Agent's run: the images improve



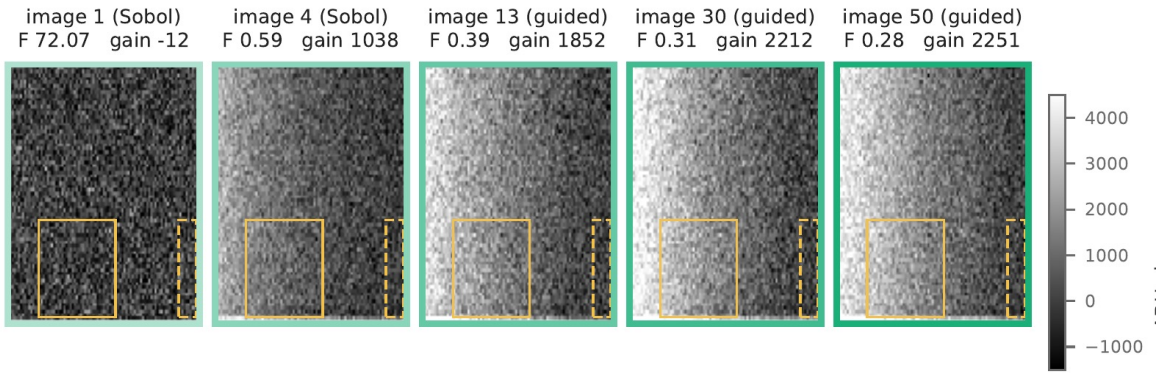


Images where the best F improved (amplifier 3, overscan subtracted, same grey scale). F 5.17 → 0.43, gain 182 → 1590 ADU

Optimizing with six parameters: adding the horizontal clock levels



Claude, amplifier 3, LED only. Line: best F so far (gain > 0). 4 parameters vs. 6 parameters



Added clock parameters
hh (high) and hl (low), each sent to five LTA clock variables; bounds [−10, 10] V; script defaults hh = −2 V, hl = −4.5 V

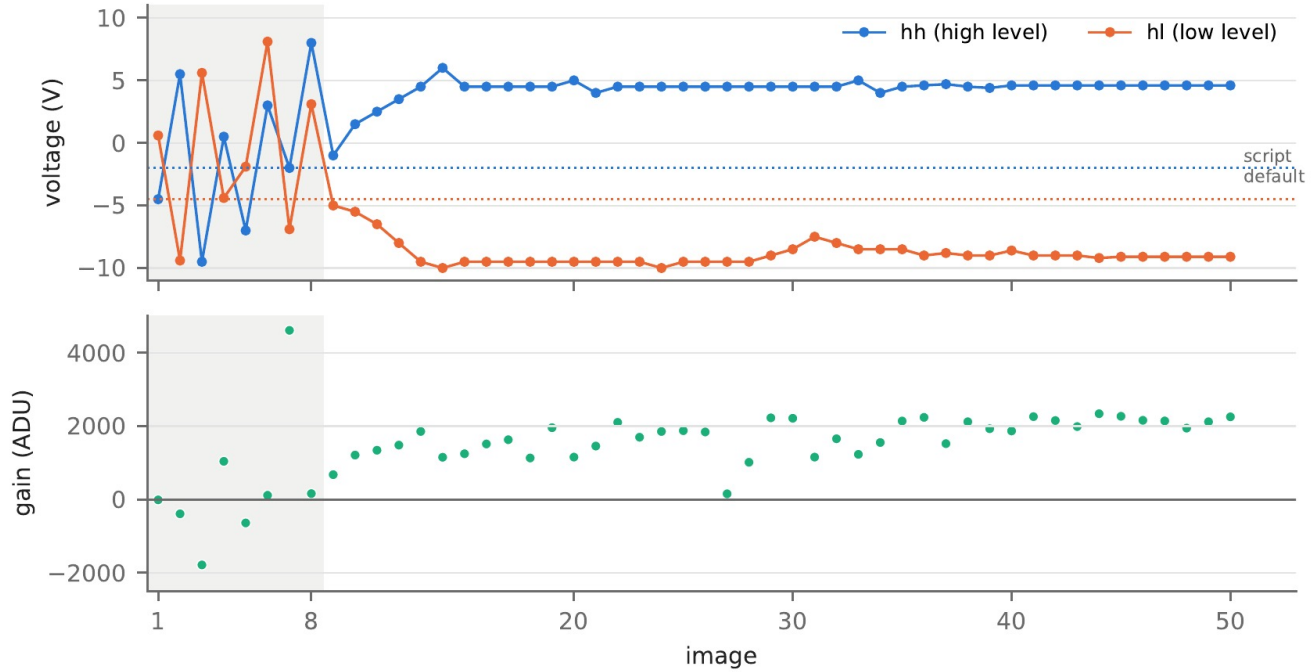
0.43 → 0.28

best F (4 → 6 parameters); a plateau of F = 0.30–0.35, not a lucky image

+40%

gain (1590 → 2251 ADU) at similar noise level (681 → 631 ADU)

Agent's reasoning about the clock levels



Clock levels chosen per image (top) and the resulting gain (bottom); dotted: script defaults

Image 9
"I'll favor the normal hh > hl ordering."

Image 11
"It had a bigger swing between hh and hl, which might be the key factor driving that gain difference."

Image 14
"Increasing the hh–hl swing seems to improve gain."

Image 45
"There's a sharp cliff around hh ~ 4.7 where gain drops dramatically."

Hypothesis formed early (swing → gain), then tested

Claude moves hh up to ~+4.5 V and hl down to ~−9 V step by step after the Sobol start; the gain follows.

Caveats: hh = +4.6 V inverts the usual hh > hl convention; check charge transfer and trails first. F rewards gain/noise, not full image quality.

Agentic optimization: takeaways and next steps

	GP	LLM agent (Claude)
Information used	parameters and F	F, image statistics, operator notes, logbook
Validity of an image	unknown to it	reasons about it (e.g. negative gain)
Explains choices	no	thinking summary per image, logbook entry
Memory across runs	data only (warm restart)	data + written lessons
Cost	free	≈ $0.3 per campaign

What we learned

An LLM agent can run a real detector optimization: it finds a valid, high-signal region within a small number of images, explains why, and turns lessons into written memory.

Next steps

More parameters

clock timing and further bias voltages: where sequential scans become impossible

Ge-CCD

apply the same agentic loop to germanium CCD characterization

Agent-informed objective

add the agent's reasoning back into F (e.g. gain ≤ 0 is a failure)

Run the optimization locally

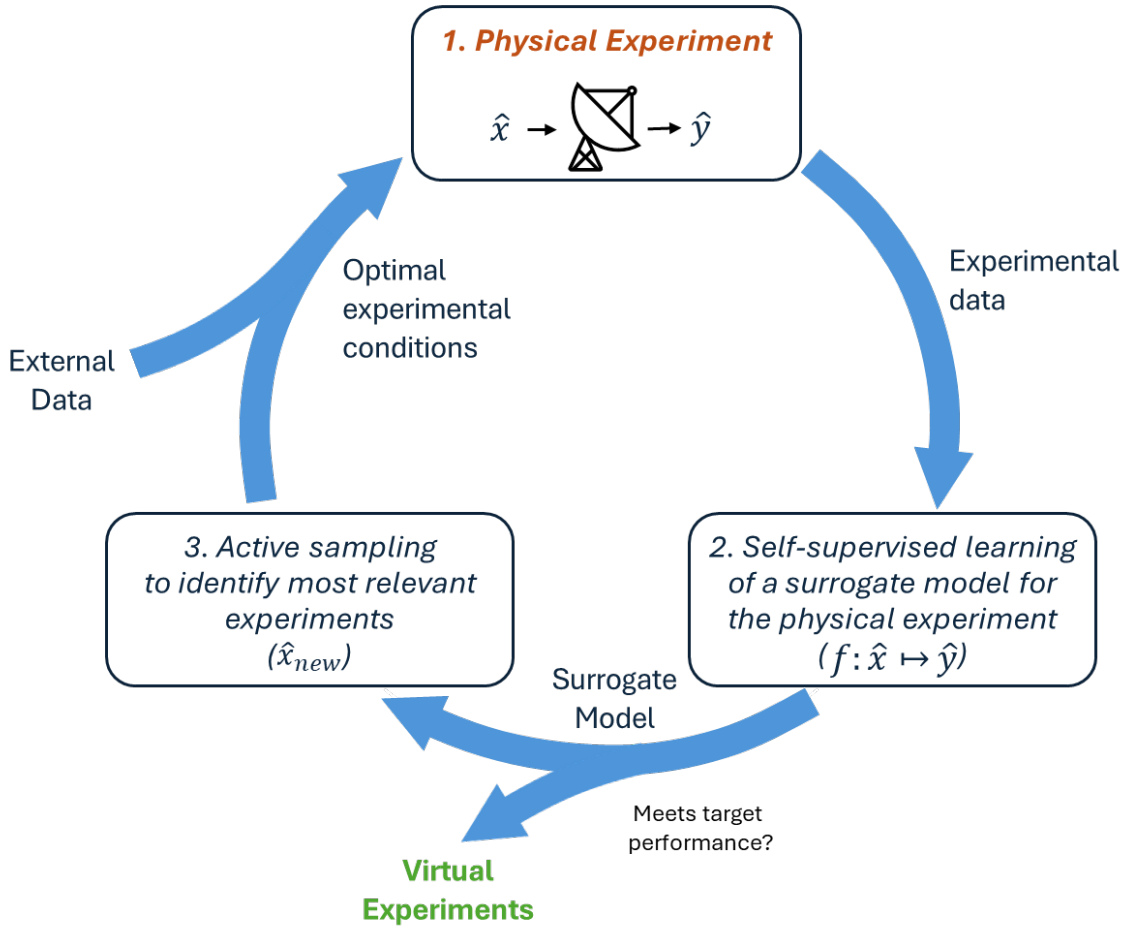
move the loop onto the test-stand side

Images used directly (instead of pre-defined scalars)

give the images directly to the agent and ask it to produce "good" images

Surrogate modeling: virtual experimentation loops

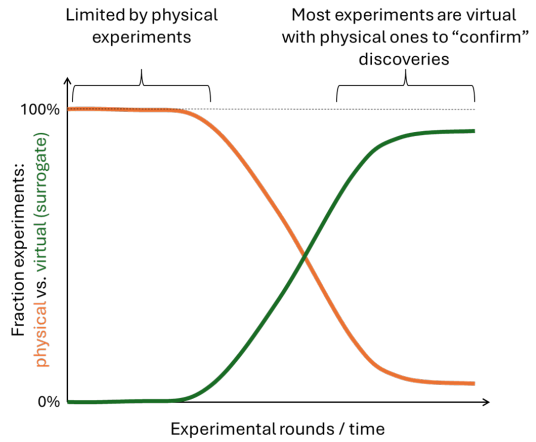
A. Overview:



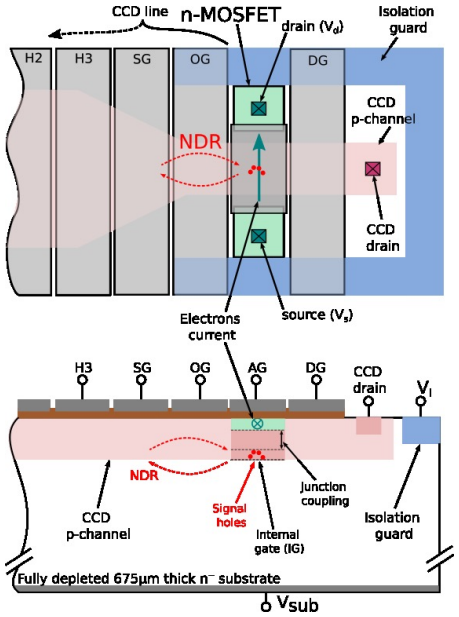
Idea: learn the multidimensional detector response, settings → output metrics, not just a scalar score

- **Surrogate model:** a neural network trained on measured images predicts signal, noise, σ, CIC, SNR
- **Virtual exploration:** scan operating conditions in milliseconds instead of hours
- **Active sampling:** with uncertainty quantification, measure where the model is least certain

B. Physical versus virtual experiments



Test case: SiSeRO CCD database



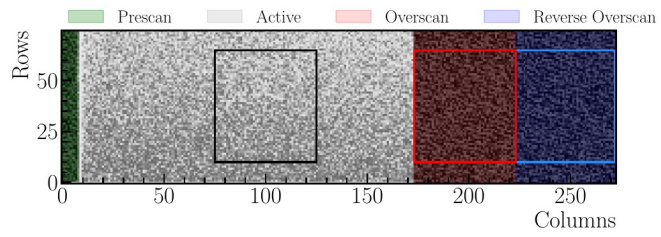
SiSeRO output stage: n-MOSFET in the CCD p-channel (JINST 21 P10012)

What a record holds

113,892 records
Image read out by 4 amplifiers (28,473 files × 4); settings + statistics; all SiSeRO at 130 K; readout by LTA and Archon modules

47 inputs (set before the image)
configuration 7 · conditions 2 · clock & bias voltages 30 · readout geometry 8

26 outputs (measured from it)
median, mean, SD, MAD, RMS per region (20) · gain, CIC, H/V non-uniformity, SNR (6)



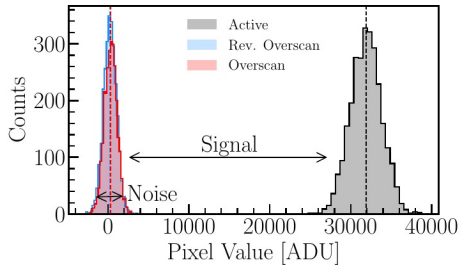


Image regions: S = active level above baseline; N = reverse-overscan width

Model inputs are the 9 voltages actually vary:

Input	Role	Range
OG_High / OG_Low	output-gate clock levels	−3…2 / −9…−4 V
DG_High / DG_Low	dump-gate clock levels	−3…2 / −10…−6 V
RGA_High ≈ Vgs	amplifier gate–source bias (MOSFET operating point)	0…4.8 V
Vdd, Vr	output-stage supply and reference	−20…−6, −15…−6 V
Voff	front-end offset into the ADC range	0…28 V

Model targets (from the image regions)

S = median(active) − median(overscan) **N** = MAD(reverse overscan)
σ_active = MAD(active) **CIC** = median(overscan) − median(rev. overscan)

59,388
LTA images, modules 3, 6, 7

18%
have SNR > 1

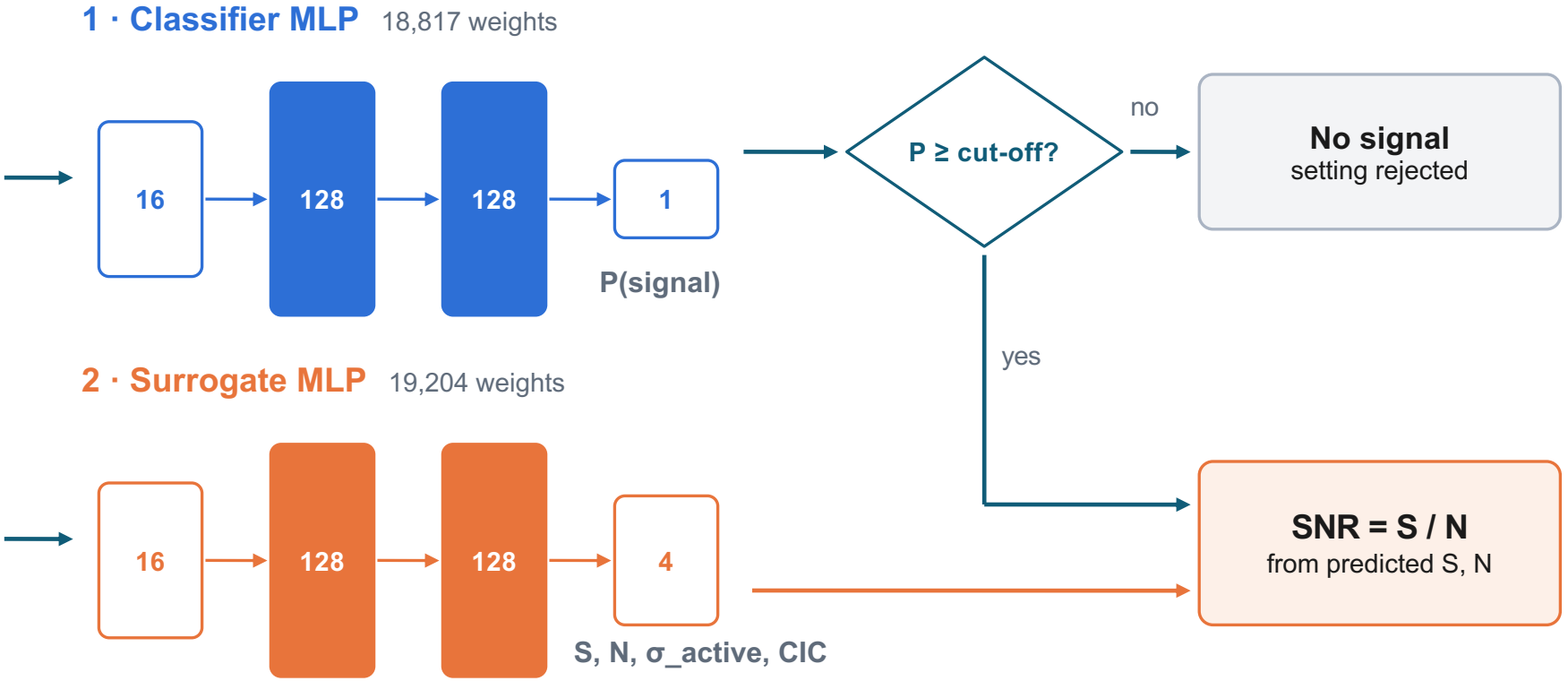
Two-stage model: classification followed by prediction

16 inputs

9 settings (standardized)
RGA_High, OG_High, OG_Low, DG_High, DG_Low, Vdd, Vr, Voff, Total_Rows

Module, one-hot (3)
modules 3, 6, 7

Extension, one-hot (4)
amplifiers 1–4

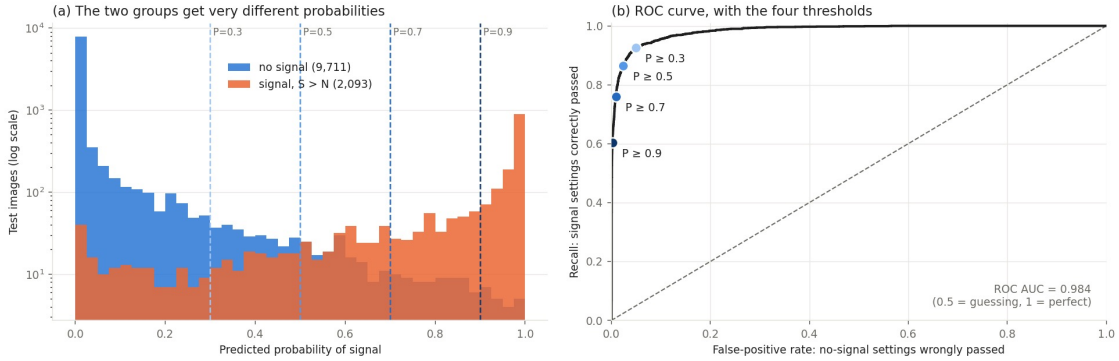


Why split? Predicting output metrics (values) only meaningful when SNR > 1.

Training: 2 × 128 hidden, SiLU, 10% dropout, AdamW. Classifier on all 59,388 images; surrogate on the 10,730 images with S > N. 80/20 split, identical settings kept together.

Results: classifier predicts probability of SNR > 1, surrogate predicts it

1 · Classifier: does a setting lead to SNR > 1?



(a) predicted probability by true class; (b) ROC curve with four cut-offs. 11,804 test images

0.984

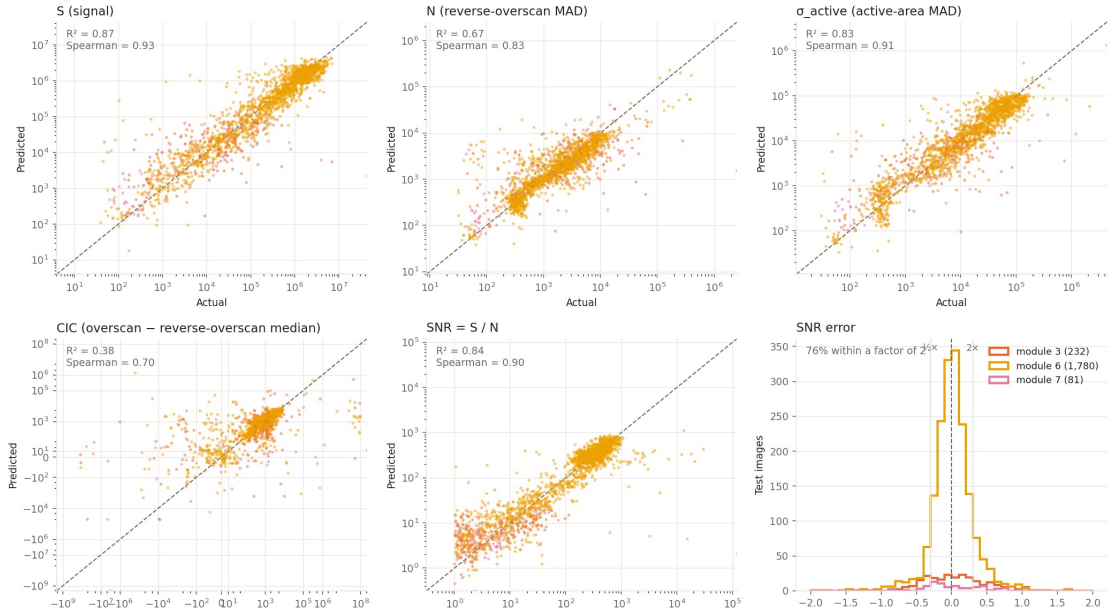
ROC AUC (0.5 = guessing, 1 = perfect)

89% / 86%

precision / recall at P ≥ 0.5; only 2.4% of no-signal settings pass

Settings with SNR > 100 pass 98–100% of the time; mistakes are mostly for borderline values (SNR 1–10)

2 · Surrogate: S, N, σ_active, CIC values predicted where SNR > 1



Predicted vs. actual, 2,093 test images with signal (orange: module 6, 84% of signal data)

R^2	S	N	σ_active	CIC	SNR
log scale	0.87	0.67	0.83	0.38	**0.84**

76% of SNR predictions within ×2. **20/20** top predicted-SNR test settings had signal.

Next steps for surrogate models



Active sampling

Add uncertainty (e.g. ensembles) to the surrogate and measure where it is least certain; link surrogate model to real-time experimental data

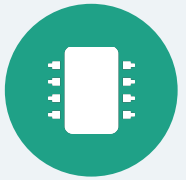
Goal: high fidelity model from a small number of real experiments



Temperature as an input

Train on dense higher temperature (~77 K – 140 K) data plus sparse lower temperature (< 77 K) data

Can the surrogate predict the lower temperature response from mostly higher temperature data?



Readout electronics

Same concept applied for cryogenic-ASIC characterization

Predict cryogenic performance from room-temperature measurements

Idea is to use a few, well-chosen experiments, as a stand in for expensive, exhaustive characterization

Conclusions

1. **Autonomous detector characterization frameworks (based on Bayesian optimization or LLM agents) are quite effective**
   SiSeRO & Skipper CCDs tuned in a few hours instead of a week of expert time, and parameter sensitivities and degeneracies mapped along the way

2. **An AI agent adds context and explanation**
   Claude as a drop-in optimizer reads image statistics, notes and a logbook; ~2× better objective function value than the GP among valid images with a small cost per campaign

3. **Surrogates learn the response itself**
   Idea is to move parameter space exploration and characterization to virtual experiments when possible, framework applicable to multiple platforms (sensors, cryogenic-ASICs,…)

It works, but how and why matter just as much!

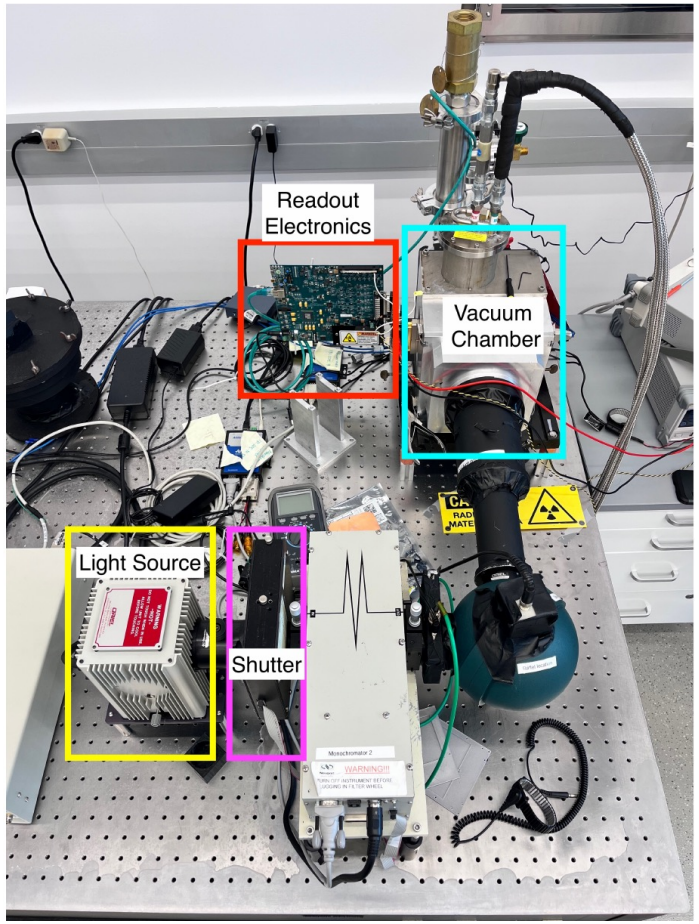
AI expedites characterization tremendously. But speed should not the only goal!

- **Keep learning and training**
  the next generation of detector scientists still need to build the intuition themselves
- **Link reasoning to device physics**
  explain why in terms of how the device operates
- **Use the optimizer as a microscope**
  partial dependence reveals parameter sensitivities and degeneracies
- **Transfer what we learn**
  across devices, temperatures and readout electronics

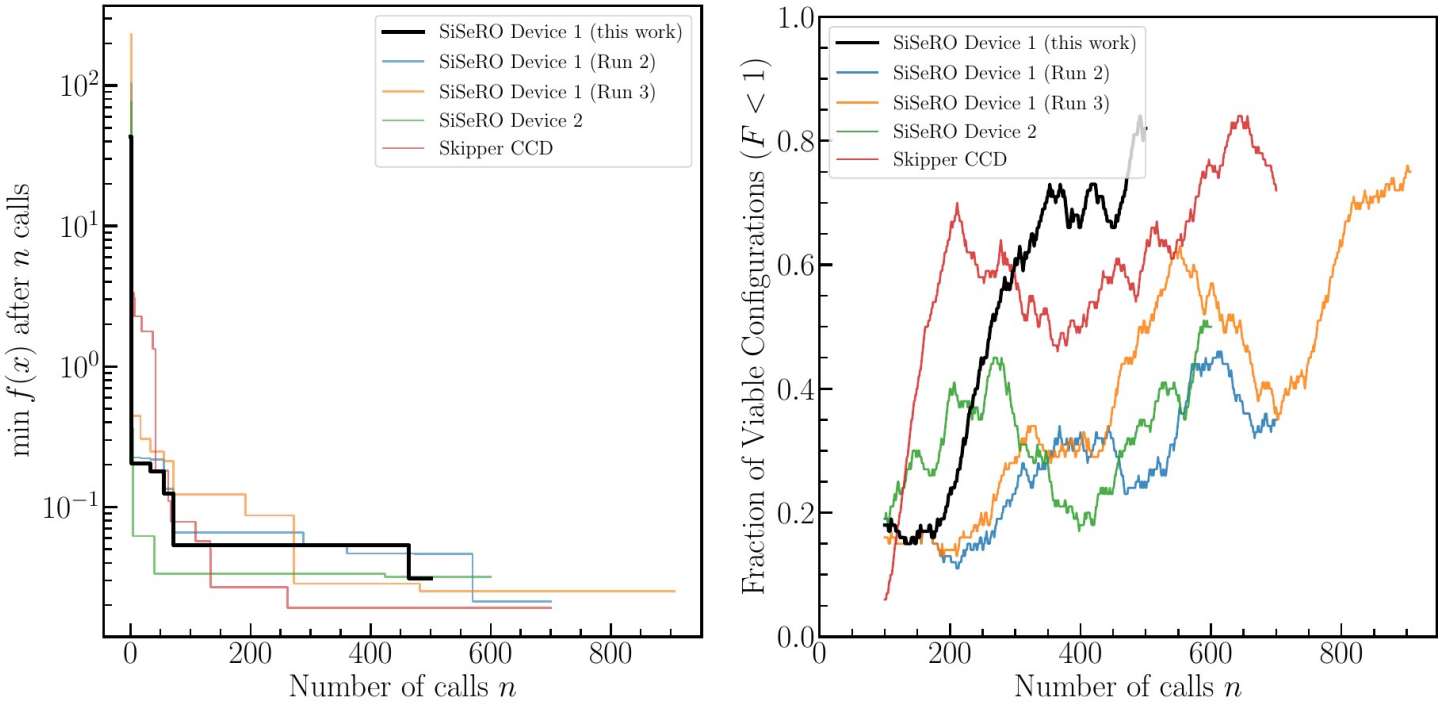
"It works" is half the answer; but knowing how and why is equally important!

Backup

Bayesian optimization applied to SiSeRO CCDs: already shown promise



Automated test stand: light source, shutter, vacuum chamber, readout electronics



Running minimum of F (left); fraction of viable settings, F < 1 (right)

5

coupled parameters: VDS, VGS, VOffset, VI, CDS samples

F ≈ 0.03

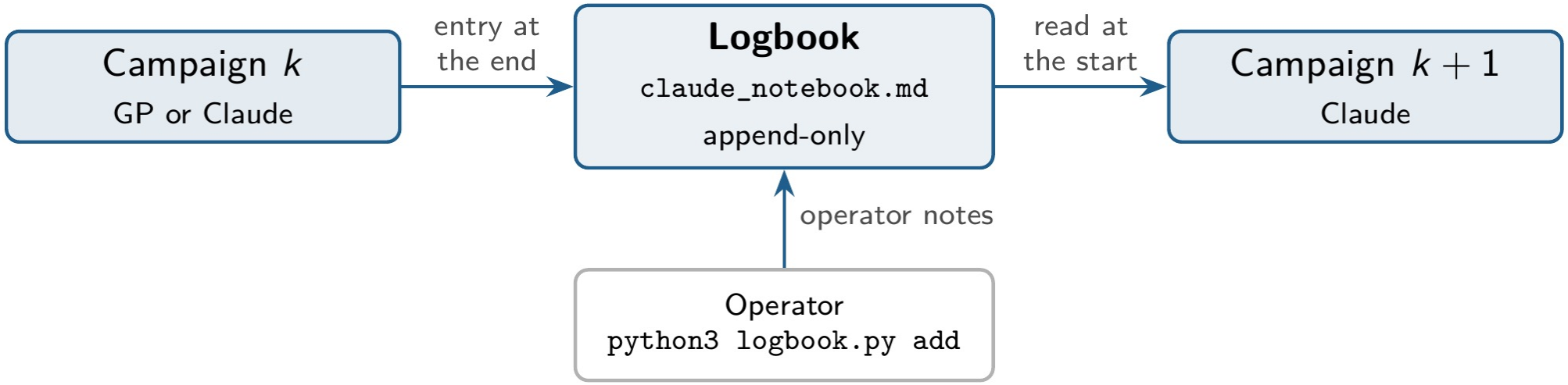
after 500 evaluations (150 random + 350 BO)

< 10 hours

unsupervised, vs. a week or more by hand

[J. Cuevas-Zepeda et al 2026 JINST 21 P10012.]

Retaining memory across campaigns through the lab logbook



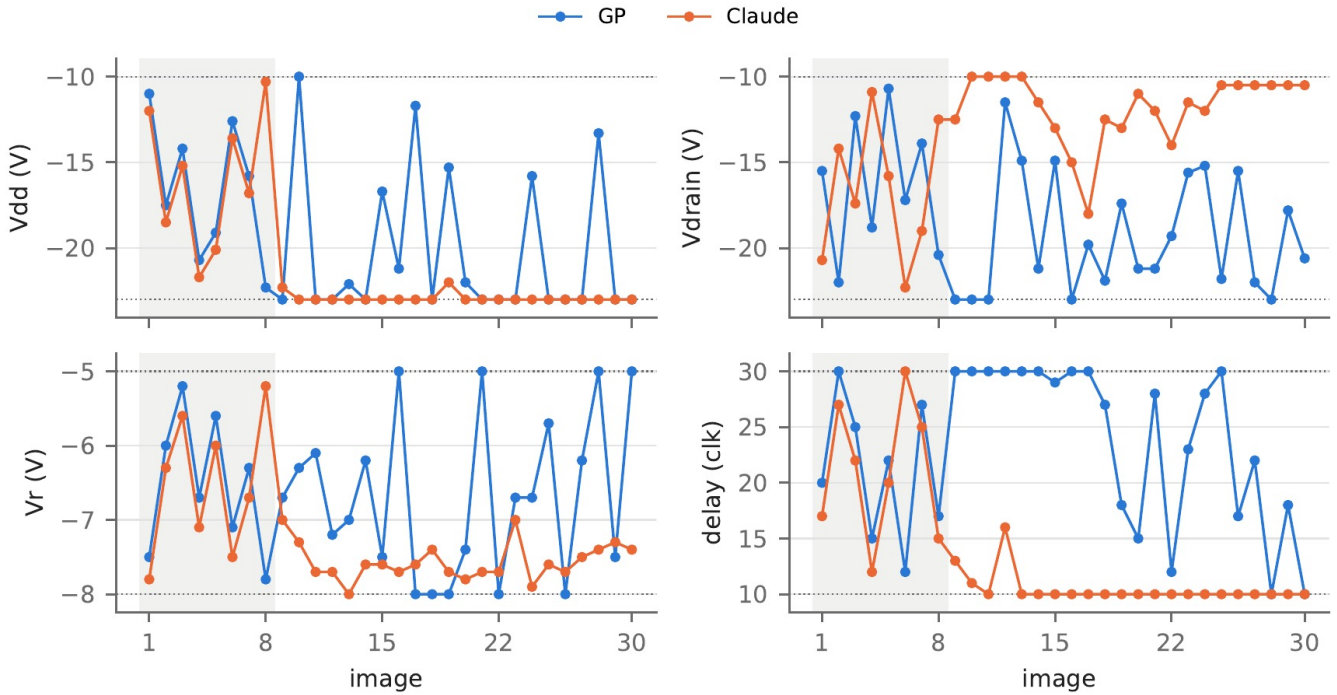
How it works

- Each API call starts from scratch: Agent does not learn between runs by itself
- Every run appends facts (written by the code); Agent adds its analysis
- Read at the start as prior knowledge; current data take priority
- Append-only; can be turned off for fair comparisons

What the logbook caught

- **Broken amplifier:** “noise_overscan […] about 50× […] F values not comparable”
- **Signal decay:** “stop optimizing and investigate the LED or exposure chain”
- **Illumination change:** operator note on light leaks explains the ~15× lower signal on Oct 2

Convergence trajectories: GP vs. LLM agent



Dotted: bounds. Claude converges to (−23, −10.5, −7.5, 10) and then varies one parameter at a time; the GP keeps sampling the edges

- **F uses |gain|:** an inverted signal looks as good as a real one
- **Weak signal:** F is noisy and nearly flat over most of the space, hard for a GP with no notion of a valid image

Claude, unprompted

"Treat any gain < 0 as a failure."

It reads the image statistics that travel with every score; the GP sees only F.

Next: fix the objective for both

- Reject or penalize gain ≤ 0 (the negative_gain penalty exists, currently off)
- Extend bounds: Vdd and delay ended at their limits (−23 V, 10 clk)
- Repeat GP vs. Claude with the corrected objective