

# An Agentic Workflow for Quality Control of Front-end Electronics

Chao Zhang  
BNL

June 17, 2026

Neutrino Physics and Machine Learning 2026, UC Irvine

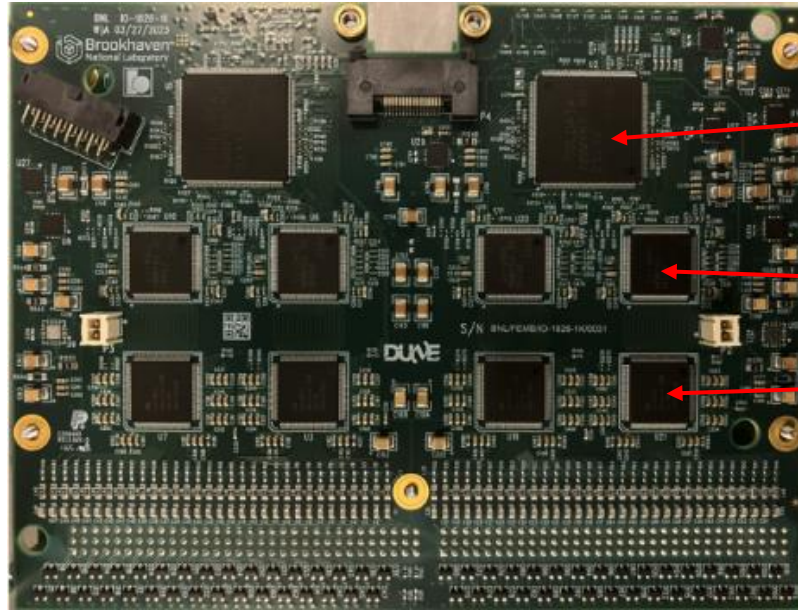


@BrookhavenLab

# Overview

- ❑ The Problem
  - We need reliable Frontend Electronics for physics experiments
  - Quality Control (QC) is a rigorous, but labor-intensive workflow
  - When QC failure happens, diagnosis takes long expert time & efforts
- ❑ We investigated two solutions
  - (A) Fully agentic workflow: In-house agent orchestration (*LangGraph pipeline*) + Retrieval-augmented generation (RAG) + Model Context Protocol (MCP)
  - (B) AI-assisted QC diagnosis: Mature agent harness (*Claude Code*) + SKILLS + Markdown + local tools (ls, grep, sql)
  - Spoiler: one of them reached **expert-level diagnosis**
- ❑ We use DUNE Cold Electronics QC as conceptual examples, but the approaches are agnostic to other QC workflows as well

# Example QC Tasks (for DUNE FD TPC Electronics)



COLDATA x2

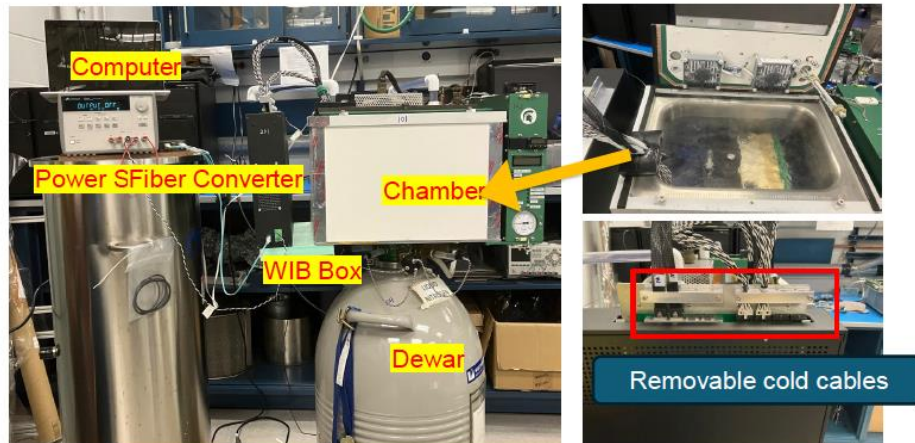
ColdADC x8

LArASIC x8

128-channel  
Front End Mother Board

## DUNE Phase I Far Detectors TPC Cold Electronics

	FD-HD	FD-VD
Readout channel	384,000	245,760
LArASIC	24,000	15,360
ColdADC	24,000	15,360
COLDATA	6,000	3,840
<b>FEMB</b>	<b>3,000</b>	<b>1,920</b>

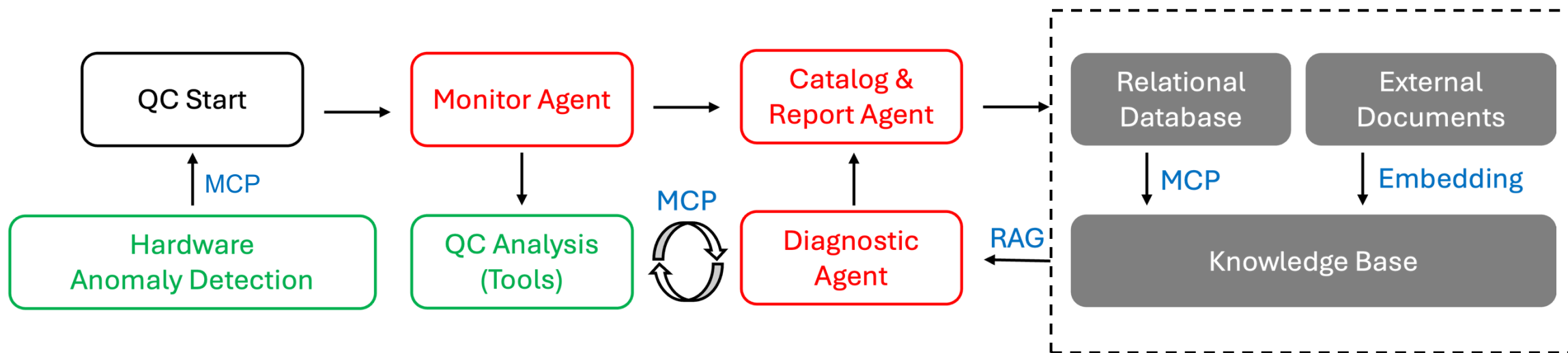


### FEMB QC with Cold Test Setup (CTS) station

- Two boards test simultaneous
- Warm (Room Temperature) + Cold (Liquid Nitrogen)
- Each QC goes through 17 test items

**Challenge:** failure diagnoses take long expert cycles.  
Can we improve by adding AI to this workflow?

# A Fully Agentic QC Workflow: Vision



- ❑ External AI/ML algorithms for Hardware/Data anomaly detection
- ❑ Communicate with Agents via Model Context Protocol (MCP)

- ❑ Agent loops and orchestration via LangGraph

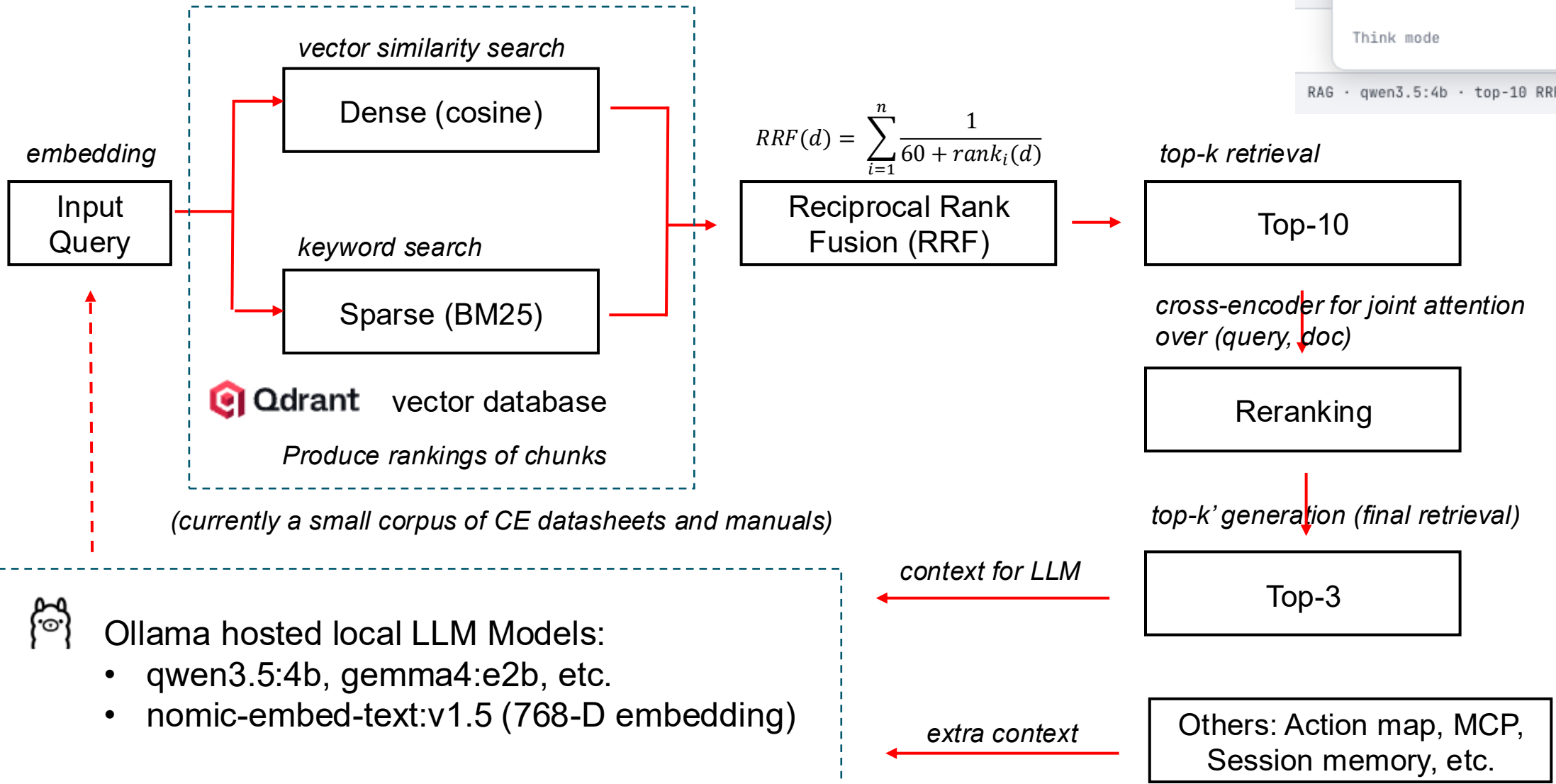
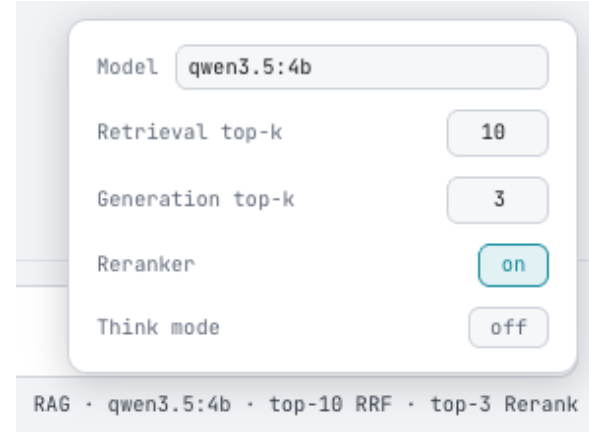
- ❑ Knowledge Base (documents, database, etc.)
- ❑ Communicate with Agents via Retrieval-augmented generation (RAG) and MCP

# Demo 1: Live Monitor + LLM analysis



- ❑ Mock QC DAQ with random failed Test injection
- ❑ Side-by-side view of two boards with 17 Tests (t1-17) each
- ❑ Live-stream via Server-Sent Events (SSE)
  
- ❑ Diagnosis reports generated via a local LLM: RAG + Action Map + MCP
- ❑ Can also chat with the agent during or after the QC session
  
- ❑ FastAPI (backend)
- ❑ Vue 3 (frontend)

# The Knowledge Retrieval Path



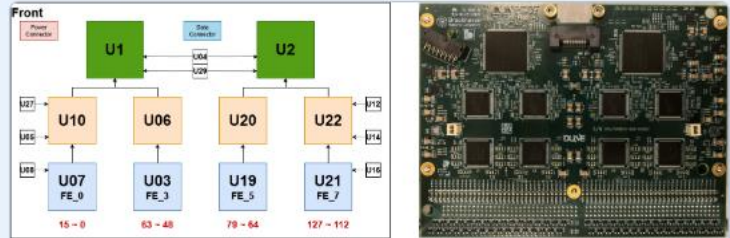
# QC and Hardware Specific Lookups

## ❑ Deterministic action maps

Fault type	Granularity	Description	Recommended action
overcurrent	board	Current draw exceeds rail limit — possible short or damaged component	Power off immediately; inspect FEMB for visible damage or short circuit
undercurrent	board	Current draw below expected minimum — open circuit or failed power domain	Check connector seating and power cable; inspect solder joints on power rails
power_cycle_fail	board	Rail voltages do not recover to nominal after power cycle	Retry power cycle; if persistent, inspect decoupling capacitors and voltage regulators
leakage_high	channel	Channel leakage current exceeds 5 nA spec	Identify affected chip/channel; likely damaged input protection diode or contamination
dead_channel	channel	No pulse/calibration response, or RMS effectively zero	Check wire-bond continuity; if entire chip is dead, suspect COLDATA I2C config failure; re-run <code>wib_adc_autocali.py</code> for the affected chip
gain_error	channel	Charge response deviates >20% from expected gain curve	Re-configure LArASIC registers; compare with adjacent channels on same chip; baseline-dependent error (t6 vs t13) suggests LArASIC bias issue; if both 200 mV and 900 mV curves fail, gain path is damaged
high_noise	channel	Channel RMS exceeds 2.5x median of all 128 channels	Check cable shielding and ground connections; verify ADC bias settings via <code>wib_cfgs</code> ; recurring single-channel excess after re-test → suspect the LArASIC serving that channel
rail_voltage_error	board	VFE or VCD rail voltage outside ±5% of nominal	Check power supply output and cable resistance; inspect WIB power delivery path
dac_error	board	FE or COLDATA DAC output deviates >10% from programmed value	Re-write DAC registers / re-run <code>wib_adc_autocali.py</code> ; if persistent, suspect the chip

## ❑ MCP server exposes sql database query tools

channel # → chip location, serial #, etc.



CETS / FEMB / IO-1865-1L/00034

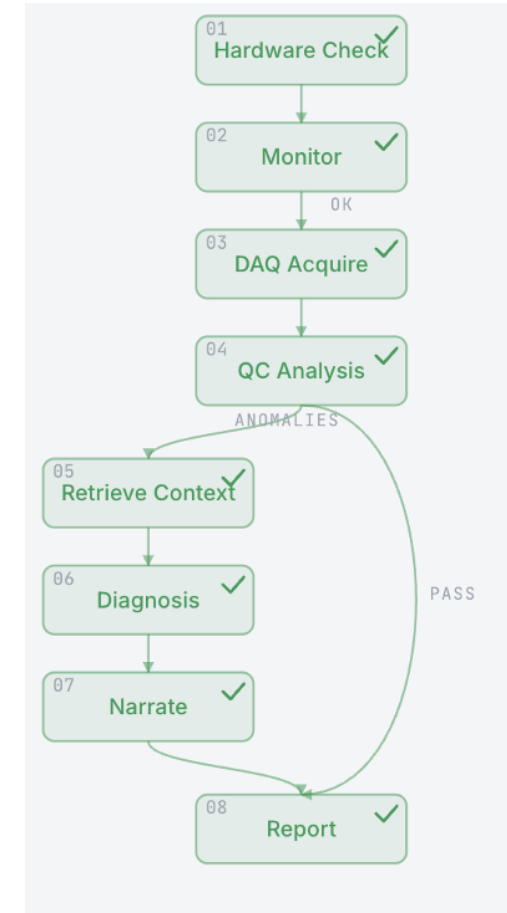
**FEMB IO-1865-1L/00034** + Add note ← Back to list

LArASICs		ColdADCs		COLDATAs	
SERIAL NUMBER	POSITION	SERIAL NUMBER	POSITION	SERIAL NUMBER	POSITION
011-00010	B1	2422-08706	B1	2506-03145	F1
011-00067	B2	2422-08686	B2	2506-03146	F2
011-06332	B3	2422-08705	B3		
011-06336	B4	2422-08685	B4		
011-00237	F1	2422-08671	F1		
011-00217	F2	2422-08667	F2		
011-06355	F3	2422-08639	F3		
011-06458	F4	2422-08635	F4		

# Demo 2: Fully Agentic QC Pipeline (Mock Data)

## LangGraph orchestration

- ❑ stateful agent control
- ❑ cyclical graph execution
  - branching & loops
  - human-in-the-loop



# What worked and did not work so well

## ❑ Proved working on Mock Data

- The full agentic loop: (local LLM + RAG + MCP) works offline
- Small LLM capable of
  - summarizing and simple Q&A
  - Deterministic routing
  - schema-validated tool calls

## ❑ Not working so well for small LLMs

- Expert-level diagnosis via deep reasoning
  - read the failed QC reports
  - inspect per-test plots (RMS, gain/linearity, ENC, monitors)
  - recall this board's **history**: past runs, repairs, known issues
  - decide: re-run? swap a chip? inspect solder? no fault at all?

## ❑ Next try: Frontier LLMs with agent harness, and more tool exposure

# Mature agent harness + Markdown + local tools (ls, grep, sql)



- ❑ Agent Harness: Claude Code
- ❑ Knowledge: curated Markdown files organized as wiki links
- ❑ Local tools
  - read access to on-disk QC data and report files
  - access to full QC test history and relations via direct SQL read-only queries to the CETS database, an existing backend built specifically to track all QC tests
- ❑ No RAG, No MCP

Q Search by FEMB serial or version... 77 FEMBs

FEMB	CHIPS	REPAIRS	QC TESTS	CHK TESTS	FIRST TEST	LATEST TEST ↓	NOTES
<a href="#">I0-1865-1L/00003</a>	18	—	5	7	2026-03-25 01:14	2026-06-01 02:47	—
<a href="#">I0-1865-1L/00012</a>	18	—	5	7	2026-03-25 01:14	2026-06-01 02:47	—
<a href="#">I0-1865-1K/00023</a>	—	—	26	37	2025-12-29 22:08	2026-05-29 17:10	—
<a href="#">I0-1865-1K/00026</a>	—	—	24	39	2026-01-20 21:05	2026-05-29 17:10	—
<a href="#">I0-1865-1K/00020</a>	—	—	20	27	2026-02-03 16:32	2026-05-28 21:38	

## FEMB I0-1865-1K/00029

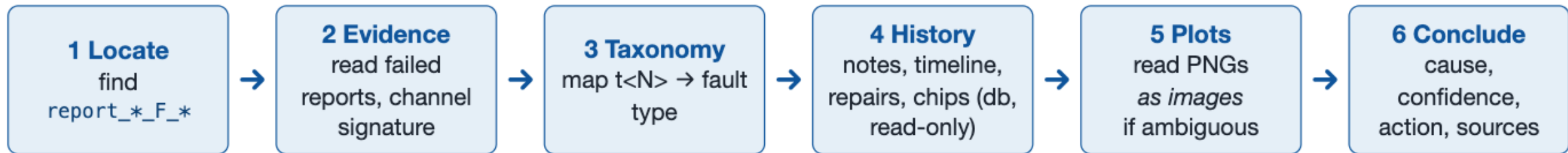
FEMB tests 8 runs

TIMESTAMP	TYPE	TEMPERATURE	RESULT	REPORT
2025-10-07 21:25:29	QC	RT · 295 K	—	<a href="#">Link</a>
2025-10-07 18:28:56	QC	LN · 77 K	—	<a href="#">Link</a>
2025-09-18 18:48:04	CHK	RT · 295 K	<span style="color: green;">● pass</span>	<a href="#">Link</a>
2025-09-18 15:28:04	QC	LN · 77 K	<span style="color: red;">● fail</span>	<a href="#">Link</a>
2025-09-18 15:26:01	CHK	LN · 77 K	<span style="color: green;">● pass</span>	<a href="#">Link</a>
2025-09-18 15:03:20	CHK	LN · 77 K	<span style="color: green;">● pass</span>	<a href="#">Link</a>
2025-09-18 13:47:05	QC	RT · 295 K	<span style="color: green;">● pass</span>	<a href="#">Link</a>
2025-09-18 13:45:16	CHK	RT · 295 K	<span style="color: green;">● pass</span>	<a href="#">Link</a>

# /ce-diagnose SKILL for Claude Code

name	ce-diagnose
description	Diagnose a FEMB QC failure from a run directory, report file/path, FEMB serial, or pasted error. Maps symptoms to the fault taxonomy, checks test/repair history in the cets db, escalates to report plots when ambiguous, and recommends next actions. Use when the user asks "what's wrong with this error/run/test", points at a report_*_F_*.md or a report_filename from the web UI, or invokes /ce-diagnose.

## Diagnose a FEMB QC failure



- ❑ ~100 lines of markdown instructions: a procedure, not a program
- ❑ LLM is the control flow: procedure and conditions explained in plain English
  - **Fast iteration**: a diagnosis weakness is fixed by editing the SKILL text, no scripting needed
- ❑ Output: *what failed* → *most likely cause (+ confidence)* → *history* → *recommended action* → *sources*

# Demo 3: Claude Code Session

```
$ claude
|
```

- ❑ Fresh session (no previous context leakage)
- ❑ Opus 4.8 model, effort high
- ❑ Single SKILL, with failed QC report location as argument
- ❑ Full diagnosis done in ~3 min. Playing at ~10x speed.
- ❑ Cost estimate is calculated by API charge per token

\*in practice much cheaper with Claude Pro/Max subscription

# Report

## FEMB 00029 (IO-1865-1K) — LN QC failure diagnosis

- **Run:** bnl/Time\_2025\_09/18\_15\_28\_04\_CTS\_BNL\_S0BNL\_FEMB\_IO-1865-1K\_00002\_S1BNL\_FEMB\_IO-1865-1K\_00029\_LN\_QC/FEMBBNL\_FEMB\_IO-1865-1K\_00029\_S1 (LN QC, slot 1, 2025-09-18)
- **Diagnosed:** 2026-06-04
- **Board:** FEMB IO-1865-1K 00029

### What failed

4 of 16 items, all pointing at the same single channel — **ch 70, chip index 4 (femb\_pos F3):**

Item	Test	Symptom
t4	Pulse response	25 mV/fC rows at 200 mV BL flagged — PPK std ~500–570 vs ~200 elsewhere, inflated by one outlier channel
t5	RMS noise	Failing-channel signature [[70], [4]]. Ch 70 RMS spikes in <b>every</b> interface mode at 200 mV BL: SE OFF ~850, SE ON ~420, SE-LC ~590, DIFF ~85 (median elsewhere ~5–25)
t6	Cali_1, SE_OFF 200 mV	ENC at 14 mV/fC reads 1007 ± 5930 — one wild channel; per-channel gain plot shows ch ~70 dropping out
t14	Ext-pulse cal, 200 mV BL	128-ch gain distribution dips to ~0 at ch ~70; its amplitude-vs-input curve is kinked (non-linear)

All 900 mV-baseline items (t7, t9, t13) and the ADC-test-mode item (t15, sync pattern) **passed** — the ADC digital path is clean; the fault sits in the analog input path and is worse (or only present) at the 200 mV baseline setting.

The same-day **RT QC passed all 16 items** (13:47 run) — the fault is **cold-only**.

### Most likely cause

**high\_noise / channel-level analog-path fault in the chip-4 (F3) site — most likely PCB/assembly (solder joint or passive in the LArASIC U19 / ColdADC U20 region), not the chips themselves.** Confidence: medium-high for the site, lower for the exact component.

Reasoning:

- Board notes record a recurring chip-4 analog-path issue: Aug 2025 ch 79 (chip 4 ch 15) stuck baseline; LArASIC U19 replaced; ColdADC U20 later replaced after ADC test-mode came back clean. **This run is the first QC after the ColdADC swap and still fails — now on ch 70 (chip 4 ch 6).**
- Two chip swaps not clearing the fault, and the bad channel *moving* within the same chip site (ch 79 → ch 70), argue against a single bad die and toward the chip site itself — reflow/solder quality or a damaged trace/passive that stresses differently each assembly, manifesting under thermal contraction at LN.
- The 200 mV-only signature would normally suggest a LArASIC bias issue (taxonomy: baseline-dependent error), but that chip has already been replaced once; a marginal joint on a bias/decoupling net at the U19 site fits both the baseline dependence and the cold-only behavior.

- Alternative (can't be excluded): the replacement chips are themselves marginal in cold. Distinguisher: a third targeted swap vs. a microscope/X-ray inspection of the U19/U20 solder joints — inspection is cheaper and the notes already point there.

Chips currently at F3 per the db (may pre-date the off-db swaps): LArASIC 009-04763, ColdADC 2502-18536.

### History

- **Recurrence**, not a first failure — third manifestation of the chip-4/F3 analog-path issue (Aug 2025 ch 79; this run ch 70).
- **core\_fembrepair** is **empty** for this board — both chip swaps are off-db, known only from the board note (flagged UNCONFIRMED in the note itself; verify with CE before treating as formal repair history).
- Timeline: 09-18 RT CHK pass → RT QC **pass** → LN CHK pass × 2 → **LN QC fail (this run)** → RT CHK pass. Oct 7 runs were partial (t6, t7 single items, both passed — not a full re-qualification).

### Recommended action

1. **Do not re-run blind** — RT passes; only a full LN QC exercises the fault, and history says it will recur.
2. **Inspect the U19/U20 site** (microscope / X-ray of solder joints, plus the bias/decoupling passives on the chip-4 LArASIC channel inputs) before any further chip swap — two swaps have already failed to clear it.
3. If inspection is clean and a discriminating re-run is wanted: full LN QC, watching whether the failing channel moves again within chip 4 (moving channel → site/assembly; same channel → chip pin or trace).
4. Confirm and record the off-db repair history (U19, U20 swaps) in **core\_fembrepair** — the note is explicitly marked unconfirmed.

### Sources

- Final report: web · tmp/femb/bnl/Time\_2025\_09/18\_15\_28\_04\_CTS\_BNL\_S0BNL\_FEMB\_IO-1865-1K\_00002\_S1BNL\_FEMB\_IO-1865-1K\_00029\_LN\_QC/FEMBBNL\_FEMB\_IO-1865-1K\_00029\_S1/Final\_Report\_FEMB\_BNL\_FEMB\_IO-1865-1K\_00029\_S1.md
- Failed item reports: t4 · t5 · t6 · t14
- Key plots: RMS per-channel (200 mV, all configs) · CALI6 gain/linearity · CALI1 ENC
- Knowledge: docs/knowledge/fault-taxonomy.md (t4/t5/t6/t14 definitions, high\_noise/gain\_error fault types, chip-index → femb\_pos map)
- Database (read-only): core\_femb id 10 (notes), core\_fembtest timeline, core\_fembrepair (empty), core\_laras1c/core\_coldadc femb\_id 10

Generated by /ce-diagnose with claude-opus-4-8 in 3m 6s.

	output	input	cache read	cache write	total
tokens	33,063	3,060	1,511,053	129,768	
cost	\$0.83	\$0.02	\$0.76	\$0.81	<b>\$2.42</b>

Confirmed by CE experts to be an accurate, thorough, expert-level analysis

# How it reasons

Item	Test	Symptom
t4	Pulse response	PPK std ~500–570 vs ~200 elsewhere — one outlier channel
t5	RMS noise	ch 70 spikes in <b>every</b> interface mode at 200 mV BL (~850 vs ~5–25)
t6	Calibration (SE_OFF)	ENC reads $1007 \pm 5930$ — one wild channel, gain drops out
t14	Ext-pulse cal	gain dips to ~0 at ch 70; amplitude curve kinked

## ❑ Symptom aggregation

- 4 items fail, all pointing at **one channel**

## ❑ Configuration narrowing

- fails at 200 mV baseline but not 900 mV;
- ADC sync passed → digital path clean, fault is **analog**
- same-day Room Temperature QC passed → fault is **cold-only**

## ❑ **History**: recurrence after two chip swaps, channel moving within a site → **solder / assembly, stressed by LN contraction**

The skill encodes **where to look**. the model supplies the **multi-hop reasoning**.  
The LLM reasoning logic matches how an expert would diagnose such a problem.

# Model Comparison: Sonnet 4.6 vs Opus 4.8 on the same /ce-diagnose run (n=1)

Dimension	Sonnet 4.6	Opus 4.8	Winner
Diagnosis (root cause)	Correct	Correct	tie
Accuracy of stated facts	1 imprecise claim, 1 unsupported inference	All claims verified	Opus
Comprehensiveness	Single-board, single-run view	Sibling board, LN cross-check, follow-up cycle	Opus
Explanation quality	Clear, correct mechanism	Adds the decisive evidence (LN $\approx$ "RT" values)	Opus
Cost	\$0.97	\$2.00 (2.1x)	Sonnet
Time	2m 40s	3m 46s (1.4x)	Sonnet

- ❑ For routine check (e.g. was this a real fault?) **Sonnet** got the right answer at **half the cost and ~1 minute faster**.
- ❑ For a definitive record (solid evidences that would survive review): **Opus**'s report is the one to keep.

# Compare the Two Approaches

	(A) In-house agent orchestration	(B) Mature agent harness + SKILLS
Reasoning	local LLM (Qwen3.5-4B) + action maps;	frontier model (Claude Opus 4.8) + harness
Retrieval	hybrid RAG, wins at large corpus	curated markdown, lazy-loaded into 1M context
Data access	MCP tools in a fixed pipeline	agentic file/db free navigation
Plots	not multimodal	Can read and analyze images
Deployment	fully offline, \$0 per run	cloud API, ~\$1–2 /diagnosis, or subscription
Engineering	full-stack system (FastAPI, Vue, Qdrant, Ollama, LangGraph)	~100 lines of markdown, needs pre-built infrastructure (QC database backend/tools)
Choose when	data can't leave site / zero cost per run / can fund the full stack R&D	frontier model permitted / cost per run permitted / engineering time is the scarce resource

- ❑ **Different axes:** reasoning quality  $\neq$  orchestration control. B leads on reasoning today; A wins when steps must be guaranteed, auditable, and hardware-wired.
- ❑ **Not either/or:** Use B to generate training data for A's fine-tuning corpus; Use frontier models in A for hard cases; Use A and B together for different tasks.
- ❑ **a local "B":** open harness (e.g. pi) + large open-weight LLM (Qwen3.5-122B, Gemma4) gives B's design fully offline; matching the Claude Code experience is an R&D project, not a free swap

# Summary

## ❑ In-house agentic pipeline

- Live monitoring and the full agentic pipeline works now
- Expert-level diagnosis needs more R&D

## ❑ /ce-diagnose + Claude Code

- Expert-level post-hoc diagnosis, working now
- ~\$2 and 3 minutes per case

A combination of both approaches will be the future R&D.

## The bigger lessons learned

- ❑ Frontier AI reasoning has crossed a threshold: it can follow a procedure, navigate our data, and reason like a domain expert.
- ❑ If you can write down how an expert reasons, you can delegate the investigation: QC here; detector calibration, data-quality check, analysis debugging, etc., all can be in our daily workflow.

# Backup Slides

# Demo: Session Chat

The screenshot displays a monitoring interface for a Cold Electronics QC session. The top navigation bar includes 'Cold Electronics QC', version 'v0.1.0', and 'QC-CTS-BNL-1'. The main area is divided into two columns. The left column shows a list of test sessions, with details for 't8 · calibration\_3', 't13 · calibration\_5', and 't15 · adc\_sync\_pattern'. Each test entry includes a 'DONE' status, source files, and a detailed failure report with 'What failed', 'Likely cause', and 'Next action' sections. The right column shows details for 't9 · calibration\_4', including a failure report and a 'RUN SUMMARY' section. The bottom status bar indicates 'connected', '8 ASICs', 'FW r417', '2 MHz', '128 ch', and 'idle 14:13:26 UTC'.

CE Cold Electronics QC v0.1.0 · QC-CTS-BNL-1 Console Monitor Documents Reports bench online · 87.2 K

SESSION 07/06 10:12:34 · LN\_QC local (fake QC) start fake DAQ 2026-06-07T10:12:34 LN\_QC session failed idle Stop watching

**t8 · calibration\_3** DONE

SOURCES ColDAC\_datasheet.pdf LArASIC\_datasheet.pdf

**What failed.** Channel-level calibration t8 (caL\_3) on FEMB BNL\_FEMB\_IO-1865-1L\_00010 reported a gain error, indicating that measured charge responses deviated by more than 20% from the expected 14 mV/IC curve.

**Likely cause.** The failure is likely caused by an incorrect configuration of analog registers for baseline or peak selection; specifically, there are existing conflict configurations in the LArASIC DAC logic where register settings (such as SG or ST bits) do not match the required caL\_3 parameters ( 200 mV baseline and 2  $\mu$ s peaking), resulting in a distorted waveform that fails the metric check.

**Next action.** Reset the FEMB registers using the COLDATA sequence to clear conflicting state, then manually re-configure channel 1865 analog settings (ensuring SG=00 for gain and ST=11 for peak time) before re-running t8 .

**t13 · calibration\_5** DONE

SOURCES LArASIC\_datasheet.pdf

**What failed.** The gain\_error metric exceeded spec for FEMB channel S1 during calibration test t13; measured charge response deviated significantly from the expected curve at a 900 mV baseline (despite passing when tested under different conditions). This indicates the LArASIC is failing to reproduce the configured 4.7 mV/IC gain relative to its specified bias point for this specific run environment.

**Likely cause.** The root cause is an unstable or incorrect internal DAC output on the FEMB's calibration registers, resulting in a mismatched baseline voltage (likely drifting away from the expected ~900 mV) or improper analog chain scaling rather than a simple digital register error. Since t13 passed previously and only failed now during cold operation with injected failure markers, this points to temperature-dependent DAC drift where the reference voltages required for 900 mV gain configuration cannot be generated accurately by the chip's internal circuitry in its current state.

**Next action.** Immediately check channel S1 I2C register writes and read the raw calibration voltage levels (if accessible via debug terminal) to confirm if the DAC output is stabilizing at exactly 900 mV; if voltages are drifting or unstable, perform a full FEMB reset followed by re-testing t6/t7 for baseline stability before retrying t13.

**t15 · adc\_sync\_pattern** DONE

SOURCES COLDATA\_datasheet.pdf ColDAC\_datasheet.pdf

**What failed.** The FEMB BNL\_FEMB\_IO-1865-1L\_00010 failed test t15 (adc\_sync\_pattern), which requires all 16 ADC channels to lock onto the SHA synchronization pattern. Although the report mentions DC noise metrics, the core failure mode identified in the specification is an **ADC sync loss**, indicating that at least one of the four chips within this FEMB slot failed to synchronize its data lanes with the external test signal (SHA0-SHA7).

**Likely cause.** The most probable root cause is a lack of synchronization between the Chip-Level Timing reference and the internal clock generation required for pattern matching. The retrieved context confirms that adc\_sync\_mode and adc\_test\_mode are enabled, but any offset in the 62.5 MHz WIB clock or a mismatch between the external SHA input frequency/phase and the chip's internal timing recovery logic would prevent all channels from locking simultaneously to the specific analog pattern.

**Next action.** Execute the script command wib\_coldata\_reset.py targeting slot S1 (BNL\_FEMB\_IO-1865-1L\_00010). If the failure persists after a reset, proceed immediately with wib\_adc\_autocali.py .

**RUN SUMMARY** 3 / 17 failed

The FEMB QC run for BNL\_FEMB\_IO-1865-1L\_00010 failed with three specific test items (t8, t13, t15) missing from the pass list despite a misleading green header. Although only 17 of 24 expected tests appear in this excerpt due to formatting, the explicit failure count confirms issues persisted across critical calibration and pulse response stages; verifying data continuity for t10 through t16 is now essential before considering the unit fit-for-purpose.

connected 8 ASICs FW r417 2 MHz 128 ch idle 14:13:26 UTC

- ❑ Can chat with the agent during or after the QC session
- ❑ Memory is kept by adding chat history to the context window (Limited by small LLM's context window size ~128K)