BNL-230305-2026-TECH

Ernest: A Retrieval-Grounded AI Assistant for EIC Accelerator Physics

V. Ranjbar

September 2026

Electron-Ion Collider Project

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC), Nuclear Physics (NP)

Notice: This technical note has been authored by employees of Brookhaven Science Associates, LLC under Contract No. with the U.S. Department of Energy. The publisher by accepting the technical note for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this technical note, or allow others to do so, for United States Government purposes.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Ernest: A Retrieval-Grounded AI Assistant for EIC Accelerator Physics

Technical Note

Vahid H. Ranjbar
Collider-Accelerator Department, Brookhaven National Laboratory

September 30, 2026
Companion to the EIC seminar "Meet Ernest", September 30, 2026

Abstract

The EIC project's technical knowledge, including lattice files, technical notes, controlled documents, presentations and simulation code, is large and spread across many repositories and formats. General-purpose language models answer questions about it fluently but without grounding, so a plausible but wrong parameter value can look exactly like a correct one. This note describes *Ernest*, an assistant built to close off that failure mode. Ernest pairs an agentic language model with (i) a local, retrieval-only knowledge server exposed over the Model Context Protocol (MCP), (ii) an audited, provenance-carrying snapshot of canonical per-ring parameters for the modelled accelerator chain, (iii) *skills*, which package validated workflows that Ernest follows and can learn, kept separate from the searchable index, (iv) memory tiers organized by lifetime, from a crash-safe per-project terminal history to a shared, git-versioned ledger of learned facts, and (v) direct access to accelerator modelling codes (MAD-X, Bmad/Tao, SciBmad) so it can check numerical claims computationally. A single launcher, `ernest claude|codex|opencode`, starts it from any working directory with any of three agent harnesses. The central design rule is that every answer separates documented facts, calculated results, hypotheses and model inferences. We describe the architecture, show representative queries and a live lattice-building demonstration, and review four campaigns of RCS lattice work carried out with the assistant between July and September 2026, which ended in the adoption of RCSV6S0 as the new 18 GeV baseline. We close with limitations, open questions and future directions.

Contents

1 Motivation **2**

2 The namesake **2**

3 Architecture **3**
3.1 Retrieval layer . . . 3
3.2 Building and sharing the index . . . 4
3.3 The audited baseline . . . 4
3.4 Claim discipline . . . 5
3.5 Modelling codes and read-only lattices . . . 5
3.6 Launching Ernest: one command, any harness, any directory . . . 5
3.7 Memory, organized by lifetime . . . 6

3.8 Skills: workflows Ernest can learn . . . 7
3.9 Lineage . . . 7

4 Demonstrations **7**
4.1 Representative queries . . . 7
4.2 Live demonstration: building and matching a lattice . . . 8

5 RCS lattice campaigns, July–September 2026 **8**
5.1 Campaign 1: 10 m RF slots at 18 GeV . . . 8
5.2 Campaign 2: rematching the 9 GeV lattice for SRF drift space . . . 8
5.3 Campaign 3: the road to RCSV6S0 . . . 10
5.4 Campaign 4: do we need stronger extraction kicks? . . . 11

6 Limitations and open questions **12**

7 Future directions **12**

8 Summary **12**

---

1 Motivation

EIC design knowledge is spread across lattice repositories, technical notes, presentations, controlled documents (specifications, SOWs, RODs), simulation toolkits and working notes. Much of the design history is found only in people's memories and old e-mail threads. Finding the *current* value of a parameter, and knowing that it is current, takes real effort.

A general-purpose LLM, asked such a question, will produce an answer. The risk is not that the answer is useless. The risk is that a confidently wrong answer looks just like a real one. A typical example is a superseded design value quoted without any sign that it has been superseded. Ernest is designed so that answers come with provenance, conflicts between sources are reported rather than silently resolved, and "not encoded" or "not found" are acceptable answers.

2 The namesake

Ernest D. Courant (1920–2020), together with M. Stanley Livingston and Hartland Snyder, discovered the strong-focusing (alternating-gradient) principle at Brookhaven in 1952 [1]. The 1958 Courant–Snyder paper [2] developed the formalism of beta and alpha functions and emittance. That formalism is the language of every twiss table this project handles (Q_x, Q_y, $\beta_{x,\max}$, γ_t). In his 2003 retrospective [3], Courant is explicit about crediting Snyder's optical interpretation of strong focusing and the co-development with Livingston and Blewett. Courant joined BNL in 1948 [4]. The author, who worked with him directly, recalls that he kept coming in to the Lab about once a week until roughly 2015.

The name is a nod to the specific BNL lineage the EIC sits in, and the persona is a matter of voice rather than any claim to be the person. The assistant is meant to be precise, to show its work, to credit its sources and not to overclaim. Those habits are the same as the claim discipline in Section 3.4.

3 Architecture

The design is best summarized as *retrieval plus discipline*, not "a smarter model". Figure 1 shows the components. Ernest draws on three read paths, which are the same for everyone: the searchable index, the audited baseline and the skills. It also keeps memory, some of it scoped to one project directory and some shared across EIC.

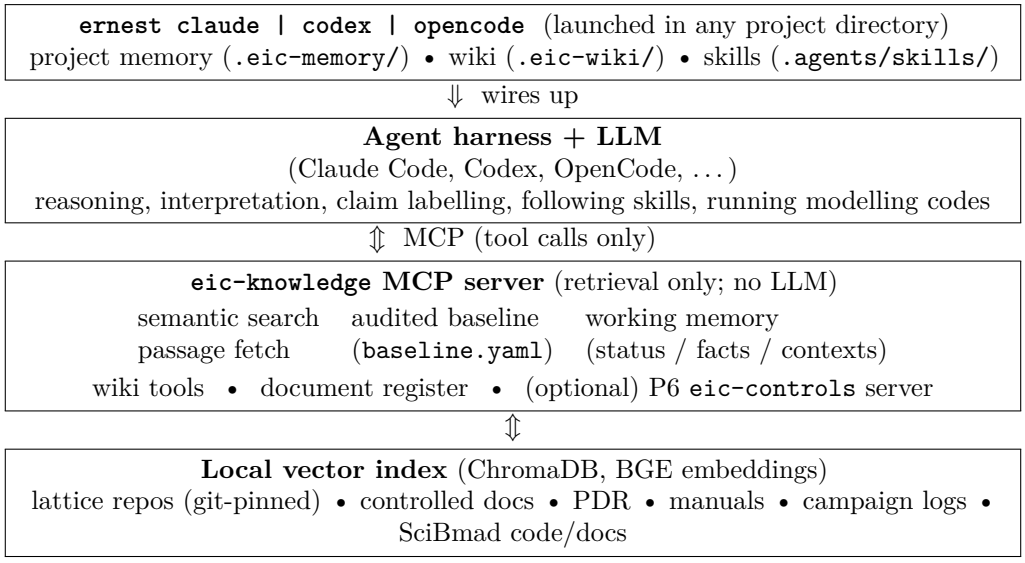


Figure 1: Ernest component overview. All physics reasoning lives in the agent; the knowledge server only retrieves and stores. Skills are read by the agent directly from the project's `.agents/skills/` directory and are not part of the vector index.

3.1 Retrieval layer

The knowledge server (`mcp_server/eic_knowledge_server.py` in the `eic-agent` repository) performs retrieval only. It neither calls an LLM nor summarizes results [5]. Documents are chunked to about 600 tokens with 100-token overlap and embedded locally with `BAAI/bge-base-en-v1.5` (sentence-transformers, on Apple-Silicon MPS, CUDA or CPU) into a ChromaDB collection. No API key or external service is involved [7].

Eighteen sources are configured [8]. They include the `eicorg/einlattices` GitHub repository, the ESR, HSR and hadron-injector lattice sources, the 2025 EIC PDR, controlled documents (with a revision register), RODs, the Bmad, MAD-X and elegant manuals, BEAMBOT's dated campaign logs, and three SciBmad Julia repositories (`SciBmad.jl`, `Beamlines.jl`, `BeamTracking.jl`). On 2026-09-30 the live index held 20 768 chunks from 3 589 files, drawn from the 15 sources that currently have content. For git-backed sources every chunk records the repository, branch and commit. For controlled documents it records the document number, revision, effective date and a status (`latest_in_corpus`, `superseded`, `duplicate`, `needs_review`), and a search can exclude superseded revisions.

The index holds *evidence*, not *method*. Until 2026-09-29 it also held copies of the DA-analysis and tracking workflow notes (HOWTOs and toolkit documentation). Those copies had gone stale; the indexed DA notes file was 53 kB against 126 kB live. They were removed. How-to knowledge now lives only in skills (Section 3.8), where it is followed step by step rather than retrieved in

fragments. The dated campaign logs (DA-with-errors tables, rematch results, tracking benchmarks, the lattice optimization log) are now indexed live from the BEAMBOT working directory, so new entries become searchable at the next index update [8].

Tool	Purpose
`search_eic_knowledge`	semantic search, filterable by source, document type, superseded status
`get_eic_passage`	reassemble the full indexed text of one file
`get_eic_baseline_fact`	canonical per-ring parameter from the audited snapshot
`list/get_eic_source*`	configured sources and index status
`get/set/clear_eic_session_status`	rolling, size-capped log of work checkpoints
`record/list_eic_learned_fact(s)`	permanent, git-committed ledger of durable findings
`save/load/list_eic_context(s)`	named, freeform topic briefings for later reload
`*_eic_project_wiki*`, `record_eic_project_note`	per-project human-readable knowledge map

Table 1: The MCP tool surface, as documented in `AGENTS.md` [5].

3.2 Building and sharing the index

A full rebuild of the index takes about two hours. Rather than have every user rebuild it, snapshots are published as zip files on the `eicorg/eic-agent` GitHub Releases page. A new user downloads the latest snapshot, unpacks it and runs an incremental update to reconcile it with the current sources [6]. Three snapshots have been published so far. The latest (2026-09-10; 14 546 chunks, 1 989 files, 13 sources) already lags the live index, which has since gained the SciBmad repositories, the controlled-document revision register, the PDR and the live campaign logs. The lattices themselves are already on GitHub: the `einlattices`, hadron-injector and HSR sources each mirror their own repository. The ESR lattice is still a manual local copy with no repository.

Whether GitHub Releases is the right long-term, central home for these snapshots, as more collaborators and harnesses come online, is an open question for the project.

3.3 The audited baseline

`config/baseline.yaml` is an audited snapshot of canonical parameters for the four EIC rings (BAR, RCS, ESR, HSR) and the AGS and Booster. For each ring it records circumference, reference energy, tunes, transition gamma, expanded model-element counts, modelled aperture and RF parameters, whenever the audited source established them. Each entry carries a `provenance` block. A companion audit report (`docs/lattice-baseline-audit-2026-09-10.md`) explains every value and every reconciled conflict. A `null` or `not_encoded` entry means the lattice does not encode the value; it does not mean zero.

As an example, the RCS entry is lattice RCSV5S1 with $C = 1507.014835\,\mathrm{m}$, $Q_x = 43.130$ and $Q_y = 56.210$, and a nominal elliptical half-aperture of 18.16 mm. It is sourced from `RCS/RCSV5S1.{twiss,seq,bmad,lte}` in `eicorg/einlattices` at commit `7df41d4b`. The entry also records two reconciliations:

- it replaces an earlier entry with no provenance ($C = 1421.25\,\mathrm{m}$, $Q_x = 41.069$, $Q_y = 58.128$, $\gamma_t = 32.042$) that does not match RCSV5S1;
- the 2022 Master Parameter Tables describe the superseded in-tunnel RCS (3846.17 m, 591 MHz, 60 MV). EIC-ROD-032 (2024-12-31) moved the RCS out of the RHIC tunnel and set injection

at 750 MeV, so the baseline does not mix the old RF values into the current model.

The baseline also records which lattice quantities are canonical, as stated by the RCS system manager: magnet counts, magnet locations, Twiss parameters and tunes are canonical. Apertures are meant to be but contain known errors. RF voltages and phases in lattice files are tracking placeholders, while the 591 MHz frequency is a design fact. Applying this rule exposed a problem. MAD-X's `TWISS` includes powered RF cavities, so the transition gamma in a `.twiss` header written with the placeholder RF on is not a property of the lattice alone. Recomputed with the cavities off, RCSV5S1 gives $\gamma_t = 39.352$ rather than the header's 39.436, and the 9 GeV lattice gives 23.185 rather than 23.889. Both agree with $1/\sqrt{I_1/C}$ computed from the same files' radiation integrals, and the baseline now records the lattice-only values. Similarly, the RCSV5S1 local aperture restriction (9.08 mm at `es.pip04/es.pip05`) was a known error. It was exactly half the ring's 18.16 mm and has been corrected to that value in the lattice repository, pending merge.

The baseline is not itself a primary document. When an answer must rest on documentary evidence, the agent follows the provenance block to the underlying file and cites that file.

3.4 Claim discipline

The agent instructions [5] require every answer to separate documented facts, calculated results, hypotheses and model inferences. They also require that no citation be invented; if nothing supports a claim, the agent says so, and "I don't know" is an acceptable answer. Disagreements between sources are reported rather than resolved silently, and semantic relevance is not taken to imply that a document is current.

Controlled documents are a case where currency matters most. A document register, kept in the `eic-agent` configuration, covers 355 controlled-document files and tags each with a revision and a status. The document number is read from the PDF text, because filenames are unreliable, and specifications and SOWs are cited by document number and revision. A superseded revision may be cited only as history. This register was motivated by a real case: a query about the B1ApF requirements used to rank the superseded Rev 00 first.

Of all the design decisions, this one matters most. Retrieval makes correct answers *available*. The labelling rule makes the *status* of each answer visible, so a reader can tell at a glance what to check.

3.5 Modelling codes and read-only lattices

Production lattice repositories are read-only for the ingestion layer, and the agent is instructed to treat them the same way. Generated or modified variants go to a working area. Numerical claims about lattices are checked by running the codes (MAD-X, Bmad/Tao, SciBmad) rather than taken from a retrieved passage [5]. How to run those codes reliably is captured in skills (Section 3.8).

3.6 Launching Ernest: one command, any harness, any directory

Ernest is started with a single launcher, `scripts/ernest`, installed once onto the user's `PATH` [6]:

```
cd ~/wherever/the/work/is  &&  ernest claude   (or ernest codex, ernest opencode)
```

The launcher wires the chosen harness to the `eic-knowledge` MCP server and to the `AGENTS.md` instructions and persona, using whatever mechanism that harness supports. For Claude Code it passes an inline `--mcp-config` and an appended system-prompt file. For OpenCode it uses the `OPENCODE_CONFIG_CONTENT` environment variable. For Codex it writes a managed

`.codex/config.toml` and `AGENTS.md`, regenerated on every launch and never overwriting a user-owned file. There is one copy of the code and the vector index, shared by every launch. The *launch directory* becomes the project: it receives its own short-term memory, its own wiki and links to the shared skills. If `EIC_CONTROLS_DIR` points at a checkout of the `p6agent` repository, the launcher also attaches that repository's read-only `eic-controls` MCP server, which provides P6 cost and schedule data, next to `eic-knowledge`.

Scoping Ernest to a working directory, rather than to the `eic-agent` repository, is what makes it usable for real work. A seminar folder, a lattice campaign and a controlled-document review each keep separate working state while drawing on the same shared EIC knowledge. The folder this note was written in is itself an Ernest project, with its own memory and wiki.

3.7 Memory, organized by lifetime

Memory is split by how long it should live and who it belongs to [6, 5]. Table 2 lists the tiers.

Tier	Where	What it holds
Terminal history	`.eic-memory/` (project)	Raw, uncurated reconstruction of recent sessions' terminal output. Rolling, about 40 kB. Written by the launcher, not by the agent.
Session status	`.eic-memory/` (project)	Checkpoints the agent writes of work in progress or just finished. Rolling, about 20 k characters, oldest evicted first. Gitignored.
Project wiki	`.eic-wiki/` (project)	Human-readable project pages and dated notes, plus generated inventories of the index, baseline, facts, contexts and skills.
Learned facts	`eic-agent/memory/` (shared)	Append-only ledger of durable findings, tagged by claim type. Committed to git.
Saved contexts	`eic-agent/memory/` (shared)	Named freeform briefings, saved and reloaded by topic. Committed to git.
Audited baseline	`eic-agent/config/` (shared)	Canonical per-ring parameters with provenance, maintained by hand.

Table 2: Memory tiers. "Project" means the directory `ernest` was launched from, and "shared" means the single `eic-agent` repository.

Short-term memory. At the start of every session the agent reads the session-status log and summarizes it to the user before doing anything else. It writes a checkpoint after any substantive task, whether finished or not. The terminal history is a safety net underneath that log. Every launch runs inside `script` with output flushed on every write, so the session is recorded even if it ends in a crash or `kill -9`. On the next launch in the same directory, the recording is rendered to readable text with the `pyte` terminal emulator, folded into `.eic-memory/terminal_history.md` and injected alongside `AGENTS.md`. If the session log is empty but the history shows unfinished work, the agent is told to treat the history as the true record and to write the missing checkpoint. A crashed session is therefore not lost work. Setting `ERNEST_NO_TERMINAL_CAPTURE=1` turns capture off for sensitive work.

Project memory. Findings specific to one project go into that project's wiki through `record_eic_project_note`. They do not go into the shared fact ledger, which is reserved for EIC-wide durable results. The wiki is a map and a working record. It is not primary evidence, so technical claims taken from it still need retrieval and citation.

3.8 Skills: workflows Ernest can learn

Repeatable workflows are packaged as *skills* in the standard `.agents/skills/<name>/SKILL.md` layout. Each skill is a short instruction file, with focused reference documents and, where useful, the scripts it relies on, alongside it [6]. Skills are a read path of their own. They are not embedded in the vector index. The agent reads the whole of a skill when a task calls for it, rather than retrieving fragments of it by similarity.

Four core EIC skills ship with `eic-agent`:

- **Bmad dynamic-aperture analysis**, which bundles W. Bergan's error, correction and DA pipeline;
- **Bmad long-term tracking**, including the RCS RF-ramp script;
- **MAD-X lattice matching**: the match contract, geometry checks before optics, a thin-lens pre-solver and multistart, independent verification (including re-deriving the RF harmonic after a circumference change), and a checked hand-off to Bmad;
- **SciBmad tracking**.

The first three were distilled from BEAMBOT's real campaigns. The launcher links all four into every project without replacing a local skill of the same name.

A skill states when it applies, its prerequisites, verification gates, outputs, failure modes and source lineage. For example, the DA skill first fixes the run contract (lattice, tunes, seeds, emittances, aperture) and flags anything missing. It requires a single-seed smoke test before a full campaign, and a check that apertures are actually active in the translated Bmad model. It treats an idealized DA scan as a screen, not a verdict. Ernest can also *learn* a new skill: when repeated work in a project establishes and validates a reusable workflow, the agent writes it up as a project-local skill. Skills hold method, not results. Campaign-specific numbers stay in the wiki notes and the campaign logs, and a single successful run is not generalized into a rule [5].

Skills are how experience becomes method. The RCS DA and matching campaigns in Section 5 were once a set of scripts and dated notes. They are now checklists that any harness, in any project, will follow, while the dated results remain searchable as evidence.

3.9 Lineage

Ernest grew out of a single-repository prototype (codename BEAMBOT, mid-2026). BEAMBOT had its own local RAG server (ChromaDB with Ollama `nomic-embed-text`) covering RCS, Bmad, MAD-X and elegant documentation. Its instructions already said to distinguish established design values from speculative estimates, and not to treat retrieved notes as authoritative unless their source, date or status supported it. `eic-agent` generalizes that prototype. It is independent of the harness, audited for provenance, and covers the whole modelled accelerator chain rather than one subsystem. What BEAMBOT learned the hard way was kept: its DA, tracking and matching workflows are now Ernest's core skills, with BEAMBOT recorded as their source lineage [6].

4 Demonstrations

4.1 Representative queries

1. *"What is the local minimum aperture in the RCS, and where is it?"*
   The baseline returns a uniform 18.16 mm half-aperture. It also reports that the committed lattice still encodes a 9.08 mm restriction at `es.pip04` and `es.pip05`, which is a known error (half the ring value) with a correction pending merge. The right answer here is partly "the file says this, and it is known to be wrong", which a model with no provenance could not give.

2. *"Isn't the RCS circumference around 3846 m?"* (a deliberate trap)
   The answer surfaces the source-conflict note. 3846.17 m describes the superseded in-tunnel concept, and the current value is 1507.014835 m (RCSV5S1), following EIC-ROD-032.

3. *"How would I run a GPU-accelerated dynamic aperture scan in SciBmad?"*
   Retrieval returns the indexed SciBmad example notebook (`dynamic-aperture-gpu.ipynb`) with its backend options and `dynamic_aperture()` usage, which shows that coverage reaches the code ecosystem and not just EIC PDFs.

4.2 Live demonstration: building and matching a lattice

The second part of the demonstration is unscripted. Ernest is asked, in plain language, to build a small lattice in MAD-X, reusing existing lattice files as templates:

1. match a toy 6 m FODO cell, with 0.5 m quadrupoles, to the classic 90°/cell working point, first with a thin-lens estimate and then with a MAD-X fit;
2. insert a 2 m dipole, bending 1°, in each drift between the quadrupoles.

This is the kind of request the MAD-X lattice-matching skill covers. The skill's thin-lens pre-solver, geometry check and independent re-verification are the steps one wants to see the agent take unprompted.

For reference, the expected answer was computed beforehand with MAD-X 5.08.01. The cell was taken as $\frac{1}{2}$QF–drift–QD–drift–$\frac{1}{2}$QF with 2.5 m drifts. The thin-lens relation $\sin(\mu/2) = L_c/4f$ with $L_c = 6$ m and $\mu = 90°$ gives $f = 2.121$ m, or $|k_1| = 1/(f\,l_q) = 0.9428\,\mathrm{m}^{-2}$. With thick quadrupoles at that strength the cell has $Q = 0.2322$ (83.6°/cell). The MAD-X fit to $Q_1 = Q_2 = 0.25$ gives $|k_1| = 1.0003\,\mathrm{m}^{-2}$, about 6% stronger than the thin-lens estimate, with β_x at the QF centre of 10.04 m. Adding the 1° dipoles, centred in the drifts, barely perturbs the optics: Q_1 moves to 0.25012 and the peak dispersion is 0.141 m. A rematch changes the quadrupole strengths by less than 0.03%. The size of the thin-lens correction, and the fact that the dipoles generate dispersion but almost no tune shift, are useful things to check in the agent's live answer.

The same habit of checking computationally rather than trusting the solver's own report is what caught real problems in the RCS work that follows.

5 RCS lattice campaigns, July–September 2026

Between July and September 2026, four campaigns of RCS lattice work were carried out with the assistant and its predecessor [9, 10, 12, 11]. Together they produced a new 18 GeV baseline (Table 3).

5.1 Campaign 1: 10 m RF slots at 18 GeV

The first versions placed two 5-cell cavities in 10 m RF slots on the 18 GeV, 80-cell-arc lattice, which grew the circumference to 1573.143 m. The following versions shortened the straights from seven to three FODO half-cells, bringing the ring to about 1485.7 m [9]. Both rings departed from RCSV5S1's 1507 m footprint, which the later campaigns preserved.

5.2 Campaign 2: rematching the 9 GeV lattice for SRF drift space

Request and rationale. Extend the drifts around the 8 SRF cavity segments in the RCS9GeV40Cell straight by +0.4 m at `mqa5` and +0.2 m each at `mqa3` and `strrf`. Then recover the optics of the `RCS9GeV40CellV2_rematch` reference ($Q_1 = 28.20019686$, $Q_2 = 27.43751067$,

	Period	Campaign and outcome
1	early July	18 GeV, 80-cell arcs, 10 m RF slots for two 5-cell cavities. The ring grew to 1573 m, then was shortened to about 1486 m with fewer, longer FODO cells.
2	July–August	9 GeV, 40-cell arcs: the same idea on the 40-cell lattice. About 30 studies, with quad-length and SRF-drift rematches. It never reached a good enough DA-with-errors margin.
3	Aug 19 – Sep 11	18 GeV, 80-cell arcs, keeping RCSV5S1's straight length. Nineteen candidates through the full DA-with-errors test. Became RCSV6S0.
4	Aug 11 – Sep 11	Extraction: are stronger extraction kicks needed? No; the existing geometry clears the septum. Written up as a note.

Table 3: The four RCS lattice campaigns.

$\beta_{x,\max} = 32.62\,\text{m}$, $\beta_{y,\max} = 30.98\,\text{m}$, $\mu_y = 2.0$ per straight, $Q'_{1,2} = 1.0$) [10]. The physical motivation, given by the author on 2026-09-14 and recorded in the learned-facts ledger, is to reduce the net wakefield. The 5-cell cavities are doubled up, which means fewer aperture tapers, and selected quadrupole radii are increased from 20 mm to 50 mm, which makes each remaining taper gentler. This rationale was not previously written down in any indexed source.

Method. A full-ring MAD-X match was warm-started from the converged V2-rematch straight quadrupoles. A multistart of 8 Latin-hypercube starts found no better branch. An earlier automated attempt found on disk had applied the drift changes but never actually rematched: its MATCH block was commented out. It was treated as not done and was redone.

Result. A fresh MAD-X `TWISS,CHROM` run, independent of the solver state, gave $Q_1 = 28.20020$, $Q_2 = 27.43751$, $\mu_y = 2.00000$ per straight and $Q'_1 = Q'_2 = 1.0000$. It also gave $\beta_{x,\max} = 33.90\,\text{m}$ and $\beta_{y,\max} = 32.59\,\text{m}$, and Bmad/Tao gave identical tunes and length. The optics growth is modest and expected: +3.9% in $\beta_{x,\max}$ and +5.2% in $\beta_{y,\max}$ relative to target. The circumference change is $12 \times 0.4 + 4 \times 0.2 + 8 \times 0.2 = 7.2\,\text{m}$, taking 1506.933944 m to 1514.133944 m, which matches the lattice exactly.

Side effects checked. The follow-up check also caught an RF harmonic carried over from the shorter ring. It was re-derived for the new circumference ($h = 2985$, about 591 MHz), and a fresh `TWISS` confirmed the linear optics were unchanged [10].

Did it cost DA margin? The same lattice then went through the standard DA-with-errors pipeline: three error seeds (200–202), orbit, tune and chromaticity correction, and DA at $p_z = 0, +0.5, +1\%$, normalized with each lattice's ring-start Twiss parameters and BAR's accepted emittances ($\epsilon_x = 170\,\text{nm}$, $\epsilon_y = 5\,\text{nm}$) [12].

p_z	Baseline	V2 delivered	V2 rematch	**V3 SRF drift**
0%	5.94	5.39	5.97	**6.17**
+0.5%	3.90	3.54	3.87	**4.15**
+1.0%	2.31	1.44	2.08	**2.20**

Table 4: Worst-of-3-seeds horizontal DA-with-errors for the 9 GeV 40-cell family, in units of σ_x, from `DAAnalysis/notes.md`, 2026-08-05 [12]. The vertical plane is not limiting (25–33 σ_y).

The SRF-drift lattice beat the production baseline on momentum and at +0.5%, and came within 5% of it at +1% (2.20 against 2.31) [12]. The wakefield-motivated change therefore cost essentially no DA margin once rematched. The 40-cell family as a whole, however, never reached a margin good enough to adopt, and the next campaign returned to the 18 GeV, 80-cell lattice.

5.3 Campaign 3: the road to RCSV6S0

Goal. For the 18 GeV RCS, keep RCSV5S1's arcs (80 FODO cells per arc) and the 1507.014835 m circumference, and rebuild the two straights to hold four double-5-cell SRF cavity slots with large-bore quadrupoles [10]. Every candidate had to pass the same DA-with-errors test before it could be considered.

What happened. Figure 2 shows the worst-case horizontal DA margin for all 19 candidates in the family, in the order they were run (2026-08-21 to 2026-09-09). Each value was recomputed from the raw per-seed outputs, and they reproduce the campaign notes to within 0.03 σ. Clean linear optics never predicted survival: nine candidates with clean optics collapsed at +1% momentum [12]. The one early survivor sat on a "high" horizontal phase-advance branch (MUX $\approx$ 1.565 per straight, matching RCSV5S1's own Q_x, integer part included). Targeting that branch in a new split-chain straight gave 10.3 σ, the first candidate with no weak point anywhere. The branch turned out to be necessary but not sufficient: a later arc-tune variant with the best idealized DA so far still collapsed. The author's own six-family straight (v2h), retuned in the vertical arc tune to $Q_{y,\mathrm{arc}} = 45.21$, became the final candidate.

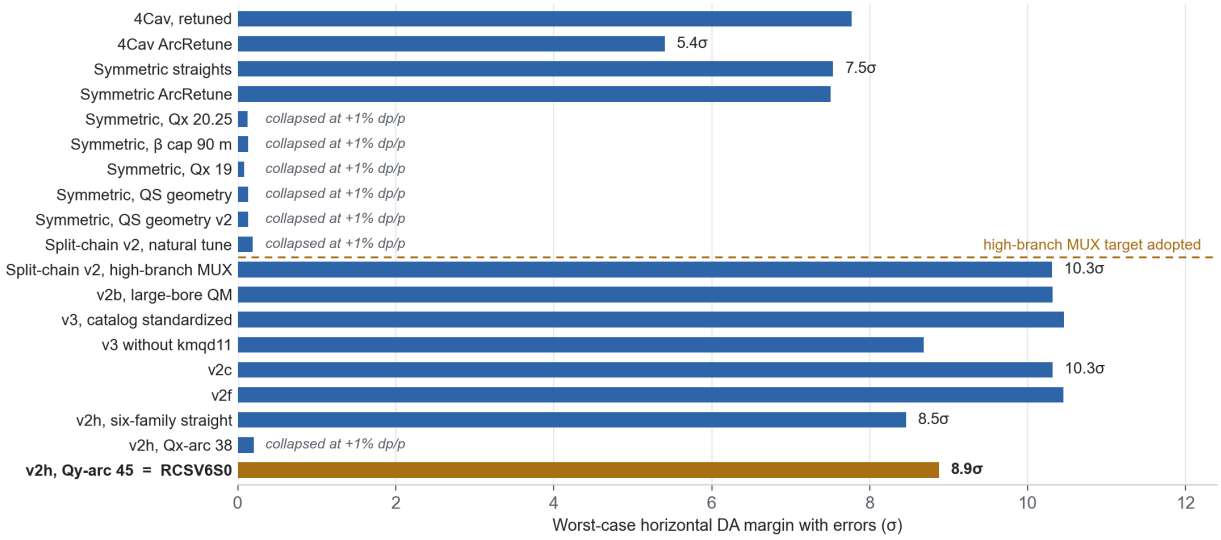


Figure 2: Worst-case horizontal DA-with-errors margin (3 seeds, p_z 0 to +1%) for every 18 GeV straight-redesign candidate, in run order, recomputed from BEAMBOT `DAAnalysis/runs/error_crosscheck_RF10M/run_4Cav*`. Normalized, as in that campaign, with $\epsilon_x = \epsilon_y = 25$ nm and ring-start Twiss parameters. At BAR's 170 nm every value is 2.61 times smaller.

Result. That candidate was checked in byte-identical as `RCS/RCSV6S0` ($Q_x = 43.1297$, $Q_y = 47.2100$). Its figures are: worst-case DA with errors 8.9 σ (at 25 nm; 3.4 σ at 170 nm); idealized DA

11.5–12.0 mm^2, the best in the family; equilibrium emittance 37.8 nm at 18 GeV, below the 50 nm limit; and peak β of 71/47 m, down from about 100 m earlier in the family. It keeps RCSV5S1's footprint, with the straight length and the 1507.01 m circumference unchanged, unlike campaign 1's 1573 m and 1486 m rings. W. Bergan's full multiparticle 750 MeV→18 GeV ramp tracking with errors and correction then showed no losses, and RCSV6S0 was adopted as the new baseline on 2026-09-11 [12]. It was chosen on evidence: only after the full errors-and-correction DA test, and then confirmed by independent ramp tracking.

5.4 Campaign 4: do we need stronger extraction kicks?

A proposal to lengthen the MQD21–septum drift, so the extraction kickers could be weaker, could not be rematched to the design optics in six attempts across three element topologies. The extraction note [11] shows the extension is not needed. The existing geometry, shared by RCS9GeV40CellV3 and RCSV5S1, already gives at least 7 σ septum clearance at 1.36–1.47 mrad per kicker at 18 GeV (1.83–1.93 mrad at 9 GeV), about 9% above the APS-heritage reference kicker's rated current at 18 GeV (Figure 3). The note recommends keeping the drift (`ES.PIP03` = 1.769 m) and driving the kickers with an energy-scaled current.

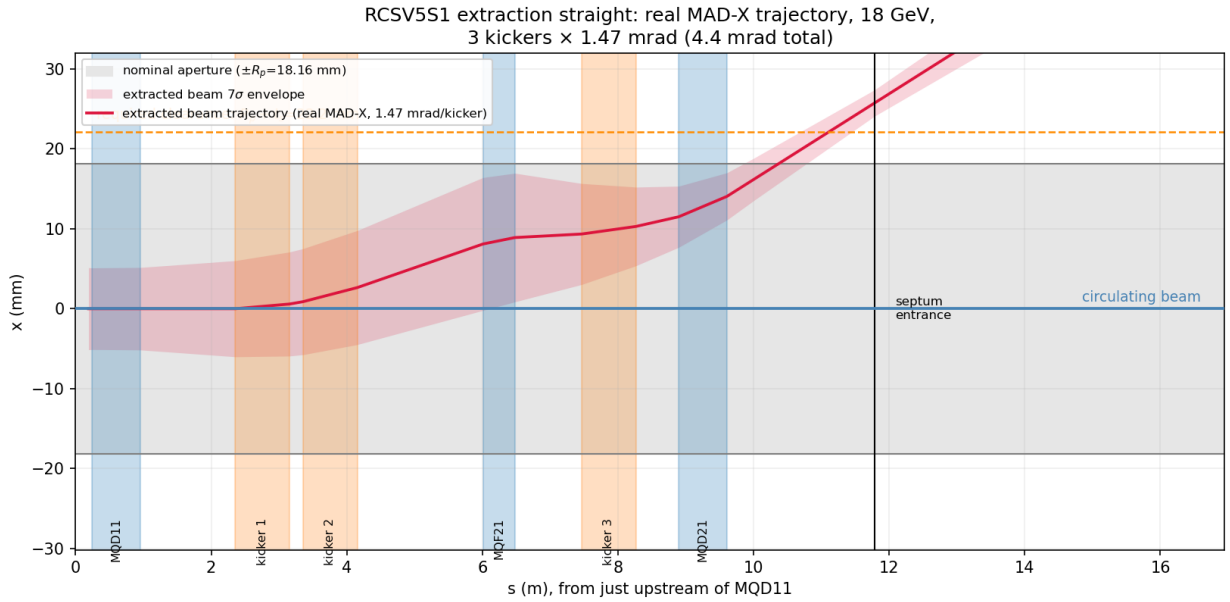


Figure 3: RCSV5S1 extraction straight at 18 GeV: the MAD-X single-pass trajectory with three kickers at 1.47 mrad each, and its 7 σ envelope, clears the septum entrance with a 1.9 mm margin. From the extraction note [11].

RCSV6S0's straight inherited this geometry, but its first kicker settings were checked at only one proxy point. When the author asked whether the full 5 σ beam really fit, a proper check showed a 2.5 mm overshoot. The re-solve, with every kicker's exit constrained, needs no new kicker: moving one 0.408 m corrector/BPM assembly upstream of MQD21 lets the existing three kickers clear 5 σ at 0.97 mrad each, with a worst margin of 0.71 mm [12]. The two results use different criteria (7 σ and 5 σ), so their kick values should not be compared directly. The relocated-corrector file is not yet in `einlattices`.

6 Limitations and open questions

- **Retrieval is not judgment.** The knowledge layer does no reasoning. Interpretation belongs to the LLM the user chooses and, ultimately, to the physicist.
- **Coverage gaps are tracked, not hidden.** The ESR lattice is still a manual-download stopgap and is not in any repository. Its folder name says "7.0" but the file headers say "Version 6.4.2", a discrepancy that is recorded as a hypothesis awaiting confirmation. The local technical-note, presentation and user-document sources are configured but still empty.
- **Staleness is a live risk.** Indexes lag their sources until someone re-runs ingestion, published snapshots lag the live index, and the baseline is a dated snapshot (audit of 2026-09-10). For example, the baseline's RCS entry still points at RCSV5S1, although RCSV6S0 has been adopted upstream.
- **Terminal history records everything.** Capture is unfiltered, which is why it can be switched off per launch. It is also unreviewed, so it is used for recovery and never cited as evidence.
- **Provenance for unstructured sources** such as e-mail, meeting notes and verbal design rationale is weak. The SRF rationale in campaign 2 entered the record only because the author stated it and it was captured in the learned-facts ledger.
- **Where does it help?** An open question for the project is where in EIC workflows a provenance-first assistant actually helps, and where it only adds friction. A second is where the shared index snapshots should live.

7 Future directions

The work is evolving along four lines:

- developing new skills that can be shared across projects;
- a better knowledge structure for short-term and long-term memory;
- adding all of the EIC technical notes to the vector index;
- extending Ernest beyond accelerator physics, to assist systems engineers, project controls and overall project planning. The optional read-only P6 cost and schedule server is a first step.

8 Summary

Ernest's usefulness comes less from the language model than from the constraints placed around it and from what it retains. Answers are grounded in indexed and version-pinned sources. Canonical parameters come from an audited baseline, and lattice claims are checked numerically. Every answer says what kind of claim it is making. Work survives across sessions and harnesses, and validated methods become skills. Courant, in his own account, credited his collaborators and did not claim more than he had done. The same standard, applied at a much smaller scale, is what this system tries to meet: show the sources, say what is not known, and claim only what has been verified.

References

[1] E. D. Courant, M. S. Livingston, H. S. Snyder, "The Strong-Focusing Synchrotron—A New High Energy Accelerator," *Phys. Rev.* **88**, 1190 (1952).

[2] E. D. Courant, H. S. Snyder, "Theory of the Alternating-Gradient Synchrotron," *Ann. Phys.* **3**, 1–48 (1958).

[3] E. D. Courant, "Accelerators, Colliders, and Snakes," *Annu. Rev. Nucl. Part. Sci.* **53**, 1–37 (2003), pp. 12–13.

[4] BNL Newsroom, "Ernest Courant's 100th Birthday" (2020), https://www.bnl.gov/newsroom/news.php?a=217140.

[5] `eic-agent/AGENTS.md`, agent behaviour instructions (repository HEAD `c2a5477`).

[6] `eic-agent/README.md`, sections 6 (index snapshots), 11 ("Using `ernest`"), "Working memory and project knowledge" and "Skills".

[7] `eic-agent/config/rag.yaml`.

[8] `eic-agent/config/sources.yaml` (commit `89dc341`, 2026-09-29, for the move of workflow method from the index to skills).

[9] `BEAMBOT/eic_lattice_optimization_log.yaml` and `eic_lattice_state.yaml`, RF10M and RCS40Cell_RF10M entries.

[10] `BEAMBOT/LatticeDevelopment/notes.md`, entries 2026-08-05 and 2026-08-10.

[11] "RCS Single-Bunch Extraction: Why Extending the MQD21–Septum Drift Is Not Warranted," BEAMBOT `LatticeDevelopment/RCS9GeV40Cell_quadlen/extraction_redesign_2026_08_11/extraction_drift_note.pdf` (2026-08-13).

[12] `BEAMBOT/DAAnalysis/notes.md`, entries 2026-08-03, 2026-08-05, 2026-08-19 and 2026-08-21 to 2026-09-11.