EVALUATING IN-CONTEXT LEARNING FOR ADVANCED LIGHT SOURCE EPICS PROCESS VARIABLE PREDICTION

A. Wu*, J. De Chant, T. Hellert, S.C. Leemann, G. Martino, A. Sulc
Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Abstract

Large language models are becoming increasingly relevant for accelerator operations, where they assist with common tasks like retrieving historical data, preparing analysis scripts, and coordinating multi-step procedures. At the Advanced Light Source (ALS), these operators use their personal jargon (e.g. "sector 4 beam current") to search for the correct PV name from numerous channels, resulting in countless variations of naming conventions. Strong scores on general-purpose benchmarks do not indicate how well a model maps operator jargon to facility-specific EPICS process variable (PV) identifiers. Building on the semantic channel-finding benchmark, we evaluate chat-based large language models on two tasks using 101 ALS expert query–PV pairs. The first probes query-level grounding via single-item testing. The assessment is executed with varying inference-time cues, scored by character-wise correspondence (Levenshstein ratio). The second probes structural understanding by requiring the model to infer character-sequence mapping from the global naming-token vocabulary under prescribed edge-count budgets. We report precision, recall, combined retrieval score (F1), and token overlap (Jaccard similarity). Applied to 27 models, these evaluations split PV retrieval from structural understanding of hierarchical naming patterns, and offer strong dependency of end-to-end PV identification on the ALS control system's naming conventions.

INTRODUCTION

Large-scale experimental facilities, particle accelerators, synchrotron light sources, and free-electron lasers depend on the EPICS control system [1] to orchestrate tens of thousands of hardware channels. Each channel is identified by a Process Variable (PV) name: a structured string that encodes subsystem, location, device, and signal information based on facility-specific naming conventions. Operators routinely translate informal, jargon-rich requests, e.g., "sector 4 ion gauge values", into the correct PV names; automating or assisting this translation with large language models (LLMs) is an active research area [2, 3]. Facility interest in operational LLMs (assistants, log mining, scripted workflows) assumes models are grounded to the correct channels, but LLMs are not trained to "speak EPICS" or a given site's naming rules. A confident wrong PV is significant; it can point monitoring, alarms, or control at an unintended signal.

This paper explores how far current LLMs get on *non-trivial*, facility-grounded tasks: jargon-heavy EPICS-style identifiers and realistic operator queries that stress retrieval and plausible-but-wrong channel suggestions. Hellert *et al.* [2] lay out the conceptual and engineering path for production-ready, LLM-based semantic channel finding; reinforcing that proposal, our work measures that LLM capability via hard benchmark scores on an ALS instantiation of that program, not a deployed control-room integration. Despite rapid progress on general-purpose LLM benchmarks (*e.g.*, MMLU [4] and HumanEval [5]), none specifically target the mapping between operational language and EPICS identifiers. This matters because: (i) PV names follow domain-specific hierarchical conventions that do not appear in public training corpora; (ii) queries use facility shorthand and spoken abbreviations; (iii) correct answers are often sets of correlated channels rather than a single string. Following that framework [2], we instantiate semantic channel finding at the ALS [6] with two tasks: **Task 1** (*PV Name Prediction*) and **Task 2** (*Token Graph Reconstruction*).

Together they yield a reproducible audit of query-level PV retrieval and of naming token graph's structural knowledge, diagnosing hallucination-prone behavior on challenging inputs. Both tasks are fully automated with deterministic scoring, use leave-one-out protocol to avoid train–test leakage, and are evaluated on 27 chat-mode models from eight providers.

BENCHMARK DATASET

The dataset has 101 items, each pairing a natural-language operator request with at least one gold-standard EPICS PV name identified by ALS domain experts, reinforcing facility-grounded query-to-channel supervision in the framework described in [2]. Queries range from terse directives ("beam current") to multi-PV enumerations, e.g., "list the sector 11 ion pump pressures" → 6 PVs, and span beam diagnostics, RF systems, vacuum instrumentation, magnets, radiation monitors, and facility infrastructure.

The target PV names have many ALS conventions (e.g., `BRL2:Gamma:Dose:1Hour`, `SR04C___SHD2___AM00`); see Fig. 1. To expose these names' internal syntax, we tokenize each PV by splitting each non-alphabetic character (digits, underscores, colons, etc.). The alphabetic tokens are in this hierarchy: *System/Ring* → *Section/Type* → *Device/Attribute* → *Signal* → *Qualifier*, shown in Fig. 2. The full token vocabulary has 139 unique tokens connected by 178 gold undirected adjacency edges across 226 PV token sequences.

TASK 1: PV NAME PREDICTION

Task 1 tests the semantic channel-finding capability as seen in [2]: picking operators' correct EPICS PV identifiers.

* wua@lbl.gov; github.com/amyawu/als-language-model-evaluation.git

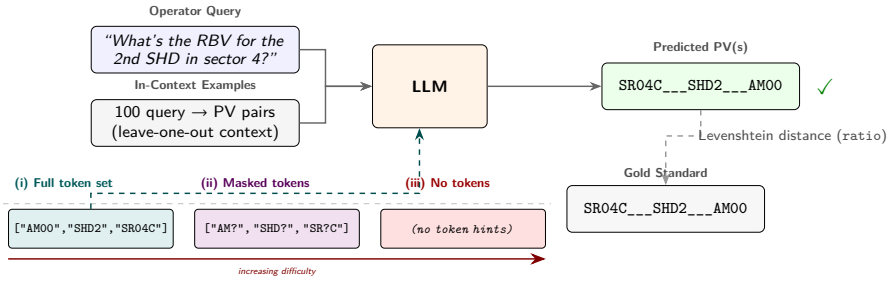

Figure 1: Overview of Task 1 (PV Name Prediction). An operator query and 100 in-context examples are presented to an LLM, which must predict the EPICS PV name(s). Three scenarios provide decreasing levels of token-set hints: (i) the full alphanumeric token alphabet, (ii) the same tokens with digit runs masked by "?", or (iii) no token hints at all.

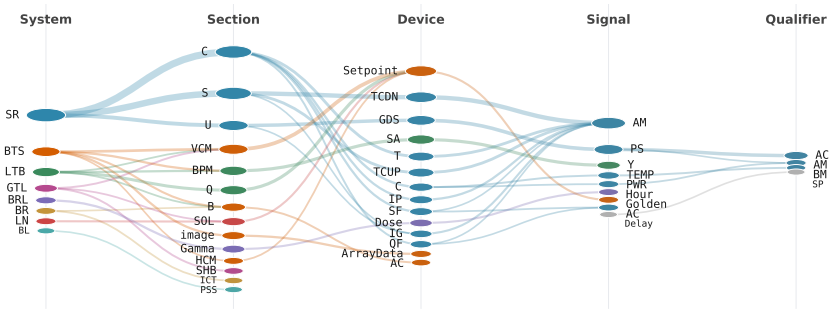

Figure 2: Layered token-flow view of ALS EPICS PV names (split on non-letters; edges join consecutive tokens; ribbon width reflects PV count per link). Gold graph for Task 2 (Token Graph Reconstruction).

Protocol and Scenarios

We use leave-one-out (LOO) over 101 folds: each fold tests one query–gold-PV item while the other 100 pairs are embedded as JSON in the system message. At temperature 0, the model sees a two-message chat (system then user) and must reply with a single JSON object of predicted PV names; chain-of-thought and tools are disabled.

Three scenarios of decreasing difficulty (Fig. 1) vary what the user message adds past the test query: **(i) token alphabet**—alphabetically sorted tokens from the fold's gold PVs, limiting assembly to those tokens plus ALS separators; **(ii) masked tokens**—the same set with digit runs replaced by "?"; **(iii) no token list**—query only from the 100 in-context examples; Table 1 reports endpoints **(i)** and **(iii)**.

Metrics

Exact set match is 1 when the predicted PV strings' multiset equals the gold set (order-insensitive), and 0 otherwise. **Mean ratio** is a sequence alignment score or a softer Levenshtein-based string-similarity score from RapidFuzz [7]: for each gold PV we take the best predicted-string match, normalize the `ratio` score to $[0, 1]$, then average over PVs and folds.

TASK 2: TOKEN GRAPH RECONSTRUCTION

Protocol and Motivation

While Task 1 tests query-level grounding, it conflates PV naming knowledge with in-context retrieval and instruction following. As a probe to the same facility ground truth, Task 2 isolates *structural* PV naming ontology's understanding: given only the token vocabulary and statistical hints, can the model reconstruct which tokens were adjacent in real PV names? The gold graph has 139 nodes and 178 edges.

From all 101 items we build the global token vocabulary and the gold undirected adjacency graph (a token-linkage diagram where an edge joins two tokens if and only if they are consecutive in some PV token sequence). The prompt is (1) a system message with ALS naming conventions, layer semantics, and illustrative adjacencies from general facility knowledge only (*not* from the benchmark), and (2) a user message listing tokens by primary layer with counts, the gold edge total, and per-layer-transition edge budgets. The model must return one JSON object with an `edges` array of pairs; up to two retries are allowed on parse failure.

Metrics

Task 2 is a *set comparison* problem: the gold graph fixes 178 true undirected token adjacencies, and the model outputs another set of candidate pairs. We score agreement with information-retrieval and set-overlap quantities [8]; their use for evaluating predicted versus reference sets is routine in machine learning and natural-language processing.

Operationally, **precision** is the predicted edges' fraction that appear in the gold graph: it measures how trustworthy the model's proposed links are (many wrong pairs imply low precision, analogous to reporting spurious beamline or device associations). **Recall** is the gold edges' fraction that the model builds: it measures the true naming structure's coverage (many missed adjacencies imply low recall). High

Table 1: Combined Results for PV Name Prediction (Task 1) and Token Graph Reconstruction (Task 2). Best Value in Each Metric Column Is Shown in Bold; Cells Are Shaded with a Very Light Column-Wise Grayscale Gradient

	Task 1: PV Name Prediction				Task 2: Token Graph Reconstruction			
Model	**Exact Tok.**	**Exact None**	**Ratio Tok.**	**Ratio None**	**Prec.**	**Recall**	**F1**	**Jaccard**
Llama-4 Scout (LBL vLLM)	55.4	32.7	0.899	0.779	0.129	0.124	0.126	0.067
GPT-OSS-120b (LBL vLLM)	70.3	35.6	0.939	0.797	0.166	0.163	0.164	0.090
Gemini 3 Flash (Vertex)	**87.1**	41.6	**0.977**	**0.821**	0.283	0.303	0.293	0.171
Claude Haiku 4.5 (Vertex)	70.3	37.6	0.928	0.804	0.168	0.275	0.208	0.116
GLM-4.7 (Vertex)	–	–	–	–	0.035	**0.815**	0.068	0.035
GLM-5 (Vertex)	–	–	–	–	0.251	0.264	0.258	0.148
Qwen3-235B (Vertex)	–	–	–	–	0.034	0.663	0.065	0.033
Grok 4.1 fast reasoning (xAI)	74.3	38.6	0.940	0.778	0.269	0.275	0.272	0.158
GPT-OSS-120b (Azure)	–	–	–	–	0.192	0.191	0.192	0.106
GPT-4o	74.3	33.7	0.925	0.764	0.126	0.169	0.144	0.078
GPT-4o-mini	63.4	31.7	0.925	0.766	0.098	0.096	0.097	0.051
GPT-5	79.2	**42.6**	0.966	0.775	**0.303**	0.303	0.303	0.179
GPT-5-mini	73.3	36.6	0.965	0.815	0.191	0.191	0.191	0.106
GPT-5.1	75.2	33.7	0.957	0.743	0.156	0.382	0.222	0.125
GPT-5.2	76.2	30.7	0.957	0.770	0.199	0.326	0.247	0.141
GPT-5.4	81.2	37.6	0.973	0.783	0.283	0.337	**0.308**	**0.182**
o1	77.2	29.7	0.958	0.574	0.270	0.270	0.270	0.156

recall alone is misleading if the model predicts huge numbers of pairs (precision collapses); high precision alone is misleading if it predicts almost nothing (recall collapses). **F1** is the harmonic mean of precision and recall; it only stays large when both are simultaneously respectable, and is a single-number summary in retrieval and classification evaluation [8]. **Jaccard similarity** compares the graph diagram's token overlap, specifically the gold edge set G and predicted set P as $|G \cap P|/|G \cup P|$: the distinct undirected pairs' fraction that appears in both sets, out of all pairs that appear in either. It is symmetric in predicted versus gold, happens when overlap is large relative to the union, and penalizes spurious edges and missing edges even when precision and recall are imbalanced.

EVALUATED MODELS

We evaluate 27 models accessed in OpenAI-compatible chat APIs at temperature 0. Models span three broad categories: *laboratory-hosted open-weight* models in LBNL's CBORG [9] infrastructure (Llama 4 Scout [10], GPT-OSS); *commercial frontier* models from OpenAI [11] (GPT-4o through GPT-5.4), Anthropic [12] (Claude Haiku and Sonnet families), Google [13] (Gemini 3), xAI [14] (Grok), and others; and *reasoning-specialized* models (DeepSeek-R1 [15], o1, Grok 4.1 fast reasoning). All models are queried through a unified harness that handles API dispatch, JSON extraction, retry logic, and deterministic scoring.

RESULTS AND DISCUSSION

Table 1 has exact-set accuracy (%) and mean string ratio (Tok./None) from Task 1 as well as precision, recall, F1, and Jaccard from Task 2.

Task 1: PV Name Prediction

Column-best exact-set accuracy runs 42.6% (no-token, GPT-5) to 87.1% (with-token, Gemini 3 Flash); column-best mean string ratio is 0.977 (Tok.) and 0.821 (None). Open-weight hosted models improve with tokens (Llama-4 Scout 32.7% → 55.4%; GPT-OSS-120b 35.6% → 70.3%) but is below the best frontier scores in both columns. API failures had "–" entries for some models, so live deployment will need availability and accuracy.

Task 2: Token Graph Reconstruction

On the 178-edge gold graph, frontier F1 clusters near ~0.29–0.31 vs. open-weight hosted near 0.13–0.16. Some checkpoints show precision–recall tradeoffs or high recall with very low precision; nine runs yielded no scorable graph (API/JSON), so valid structured output and graph content matter. Although Task 1–Task 2 correlation is left for future work, our results are consistent with the 90-97% query-to-PV accuracy seen in the retrieval-augmented pipeline [2]. Our validation emphasizes the efficacy of using LLM as a human-centric interface for the channel-finder database.

CONCLUSION

We reported an ALS instantiation of semantic channel finding [2] with 101 expert-curated query–PV pairs and two tasks (query-level PV prediction and token-graph reconstruction), evaluated on 27 chat-mode models from eight providers with deterministic leakage-safe protocols. Per-model metrics and headline scores are given in Table 1 and discussed above. The dataset, prompts, and scoring harness are open for reuse and comparison with other facilities adopting EPICS or similar structured naming systems. Future work includes extending the corpus beyond ALS to other EPICS facilities (e.g., NSLS-II, LCLS-II), adding task variants that probe multi-step operational workflows, and establishing a public leaderboard aligned with the benchmark program of [2].

ACKNOWLEDGEMENT

This work was supported by the Director of the Office of Science of the U.S. Department of Energy under Contract No. DEAC02-05CH11231.

REFERENCES

[1] L. R. Dalesio *et al.*, "The experimental physics and industrial control system architecture: past, present, and future", *Nucl. Instrum. Methods Phys. Res. A*, vol. 352, no. 1–2, pp. 13–15, 1994. doi:10.1016/0168-9002(94)91493-1

[2] T. Hellert *et al.*, "From natural language to control signals: a conceptual framework for semantic channel finding in complex experimental infrastructure", 2025. doi:10.48550/arXiv.2512.18779

[3] G. Martino *et al.*, "Autonomous Planning and Execution of Injector Tuning via the Osprey Agentic Framework", presented at IPAC'26, Deauville, France, May 2026, paper MOP6331, this conference.

[4] D. Hendrycks *et al.*, "Measuring massive multitask language understanding", 2020. doi:10.48550/arXiv.2009.03300

[5] M. Chen *et al.*, "Evaluating large language models trained on code", 2021. doi:10.48550/arXiv.2107.03374

[6] T. Hellert *et al.*, "Status of the Advanced Light Source", in *Proc. IPAC'24*, Nashville, TN, USA, May 2024, pp. 1309–1312. doi:10.18429/JACoW-IPAC2024-TUPG37

[7] M. Bachmann, RapidFuzz, github.com/rapidfuzz/RapidFuzz, accessed Mar. 31, 2026.

[8] C. D. Manning, *Introduction to Information Retrieval*. Syngress Publishing, 2008.

[9] Berkeley Lab, "CBorg AI Portal", 2026, https://cborg.lbl.gov

[10] H. Touvron *et al.*, "LLaMA: Open and Efficient Foundation Language Models", 2023, doi:10.48550/arXiv.2302.13971

[11] OpenAI, "OpenAI API Documentation", 2026, https://platform.openai.com/docs

[12] Anthropic, "Claude Models Overview", 2026, https://docs.anthropic.com/en/docs/models-overview

[13] Google, "Gemini API Documentation", 2026, https://ai.google.dev/gemini-api/docs

[14] xAI, "Grok", 2024, https://x.ai/grok

[15] DeepSeek-AI *et al.*, "DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning," *Nature*, vol. 645, pp. 633-638, 2025. doi:10.1038/s41586-025-09422-z