HEPUKE: UNCERTAINTY-AWARE KNOWLEDGE EVOLUTION FOR HIGH ENERGY PHYSICS ANALYSIS AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge bases are essential to agents for high energy physics (HEP), supplying few-shot evidence for previously unseen tasks. Their principal utility lies in strategic guidance for novel analyses rather than ready-made answers or reusable code. Prevailing systems, however, treat all retrieved evidence as equally trustworthy and fail to accrue knowledge from the analyses under assistance. In this paper, inspired by the standard HEP treatment of uncertainty, we propose **HEPUKE**: an **U**ncertainty-aware **K**nowledge **E**volution system for **HEP** that repurposes information entropy to unify statistical uncertainty from LLM generation and systematic uncertainty from HEP methodological choices. Upon it, we construct a self-evolving domain-specific-language (DSL) knowledge base and release a large-scale expert-annotated HEP physics-analysis benchmark. Per-evidence confidence is quantified via the information entropy of sampled DSL generations: high-confidence precedents constrain program synthesis, whereas analyses lacking close precedents are validated and written back, sustaining continual knowledge evolution. To handle the highly redundant yet logically stringent nature of HEP knowledge, we introduce an early-stopping controller that fuses a backward stationarity test on the answer posterior with a forward marginal information gain estimate. On the 1,000-pair BESIII PhysicsQA benchmark, HepUKE reduces retrieval rounds by 43.0% while preserving most answer quality, and on the out-of-domain HotpotQA benchmark, it reduces retrieval rounds by 14.5% with a similar quality–efficiency trade-off. The code and datasets used in this work are available at `https://anonymous.4open.science/r/HepUKE-32FF`.

1 INTRODUCTION

High Energy Physics analyses require substantial domain knowledge, complex software, and large-scale data processing, making them promising targets for agent-based assistance. Recent work has explored HEP agents, retrieval-augmented generation (RAG), domain-specific languages (DSLs), coding agents, and multi-agent workflows (Jiang et al., 2026; Atif et al., 2026; Suresh et al., 2024; Jiao et al., 2026; Desai, 2026; Sun et al., 2026; Cakir & Yerlikaya, 2026; Moreno et al., 2026; Costa et al., 2026; He et al., 2026). However, unseen analyses require more than static retrieval or one-shot generation: agents must extract transferable knowledge from validated precedents and assess its reliability. Existing HEP knowledge bases provide neither iterative retention of validated precedents nor confidence estimation for stored evidence, allowing uncalibrated procedures to introduce systematic bias during knowledge evolution.

Standard RAG implicitly treats semantic similarity as a proxy for evidential value (Lewis et al., 2020; Gao et al., 2024). This assumption is unreliable in HEP, where redundant documents may share terminology while differing in measurement type, physical relevance, or trustworthiness. Similar but distracting passages can degrade generation (Deng, 2026; Cuconasu et al., 2024; Shi et al., 2023; Yoran et al., 2024), and external evidence may amplify model overconfidence (Xiong et al., 2024). Methods such as Self-RAG and CRAG (Asai et al., 2024; Yan et al., 2024) assess generic textual credibility, but do not quantify HEP-specific physical relevance or generation-side confidence. Because an incorrect selection cut or fit configuration can propagate into the final physics result, reliability must be controlled on both sides of the RAG pipeline.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Information entropy provides a low-cost, training-free signal for LLM confidence and hallucination detection (Kuhn et al., 2023; Quevedo et al., 2025). Existing approaches mainly capture statistical uncertainty, namely output fluctuations under fixed evidence (Manakul et al., 2023); systematic uncertainty from methodological choices and retrieved evidence remains largely unaddressed. We therefore use entropy as a two-channel gate: it filters evidence with low physical relevance or high systematic uncertainty and prevents uncertain generations from entering the knowledge base.

Inspired by the HEP treatment of uncertainty, we propose **HEPUKE**, an **U**ncertainty-aware **K**nowledge **E**volution system for **HEP**. HepUKE fuses statistical uncertainty in LLM generation with systematic uncertainty induced by HEP methodological choices into a confidence signal for reranking retrieved evidence. It further introduces (i) confidence-based reranking, (ii) confidence-gated, append-only deposition of verified precedents, and (iii) an early-stopping controller combining forward marginal information gain with backward stationarity detection. Together, these components support calibrated, self-evolving, and efficient RAG-based HEP analysis. Our contributions are:

- We identify the limitations of similarity-only retrieval and propose HepUKE, which jointly controls evidence relevance, generation confidence, and knowledge evolution.
- We fuse statistical and systematic uncertainty into a unified confidence signal for DSL generation and confidence-gated precedent deposition.
- We develop an information-theoretic early-stopping controller that reduces redundant retrieval on HEP's highly repetitive corpus.
- Experiments on BESIII benchmarks show improvements in correctness and calibration with fewer retrieval rounds than standard RAG.

2 Related Work

HEP RAG and agents. HEP agents and RAG systems have been applied to analysis assistance, DSL generation, and multi-agent workflows (Jiang et al., 2026; Atif et al., 2026; Suresh et al., 2024; Jiao et al., 2026; Desai, 2026; Sun et al., 2026; He et al., 2026). Most rely on static retrieval and lack explicit control of physical relevance, methodological uncertainty, and iterative retention of validated precedents. HepUKE addresses these limitations through confidence-gated retrieval and an append-only DSL knowledge base.

Uncertainty estimation for LLMs. Semantic entropy (Kuhn et al., 2023; Quevedo et al., 2025) and multi-sample consistency methods such as SelfCheckGPT (Manakul et al., 2023) estimate uncertainty or hallucination risk, while external evidence can increase overconfidence (Xiong et al., 2024). These methods primarily describe statistical variation under fixed evidence. HepUKE additionally models systematic uncertainty arising from methodological choices in retrieved HEP evidence and combines both components into answer confidence.

Retrieval control, experience retention, and early stopping. FLARE and Adaptive-RAG (Jiang et al., 2023; Jeong et al., 2024) control retrieval using confidence or query complexity. Reflexion, Voyager, and ExpeL (Shinn et al., 2023; Wang et al., 2024; Zhao et al., 2024) retain experience, but their admission rules rely mainly on task performance or self-evaluation rather than scientific validity. BOCPD (Adams & MacKay, 2007) and marginal information gain provide principled change and novelty signals. HepUKE combines backward BOCPD-based stationarity detection with forward JSD-based marginal information gain for externally computable early stopping, without reflection tokens or retraining.

Gaps and positioning. Existing methods do not jointly control evidence-side systematic uncertainty, generation-side statistical uncertainty, and scientifically validated knowledge evolution. HepUKE fills this gap with a unified confidence estimator, a dual gate spanning retrieval and precedent deposition, and an information-theoretic early-stopping controller.

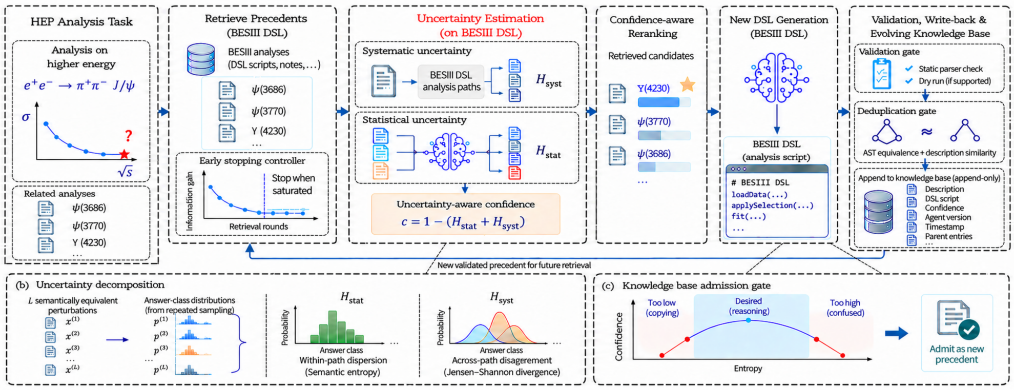

Figure 1: **Overview of HEPUKE.** HEPUKE performs uncertainty-aware precedent retrieval and BESIII DSL generation, using statistical and systematic uncertainty to guide confidence-based reranking and gated knowledge evolution. Bottom: (a) early stopping terminates saturated retrieval, (b) uncertainty is estimated across semantically equivalent DSL analysis paths, and (c) only validated precedents within the confidence gate are admitted to the evolving knowledge base.

3 PRELIMINARIES

3.1 PROBLEMS

Current HEP analysis agents face several challenges when reusing precedents across scientific documents and software workflows. First, retrieval relevance does not imply precedent usefulness: similar examples may introduce unsuitable topologies or numerical cuts. Second, systematic uncertainty introduced by methodology are not estimated, and sampling variation under a fixed instruction is easily confounded with sensitivity to controlled instruction changes. Third, program syntax is insufficient to establish scientific validity, necessitating explicit admission and reuse mechanisms. Fourth, a retrieval controller must allocate retrieval within a question, trading off a stability signal against the estimated value of another round. With redundant, evolving knowledge, over-retrieval wastes calls and hides correct information among near-duplicates. Existing RAG approaches, uncertainty quantification, and continuously evolving knowledge bases do not fully address these issues in scientific program synthesis.

3.2 BESIII EXPERIMENT AND HEPSCRIPT REPRESENTATION

The Beijing Spectrometer III (BESIII) (Ablikim et al., 2010) is an experiment at the large lepton collider Beijing Electron Positron Collider II (BEPCII) (Yu et al., 2016) and has accumulated the world's largest τ-charm dataset. A complete analysis at BESIII typically spans detector simulation, event reconstruction, background suppression, event selection, statistical analysis, and systematic uncertainty evaluation, relying on large HEP frameworks and specific offline software. This workflow is represented and driven by the HEPSCRITP (Jiao et al., 2026), a DSL whose structured syntax enables parsing, comparison, and deduplication of analysis programs—properties unavailable for free-form scripts. Each DSL program encodes a set of observable analysis decisions as shown in Algorithm 1

4 METHODS

We present HEPUKE, an uncertainty-aware knowledge evolution framework for HEP analysis. It operates a retrieval loop on a shared knowledge base $\mathcal{K} = \mathcal{K}_{\text{QA}} \cup \mathcal{K}_{\text{DSL}}$, where $\mathcal{K}_{\text{QA}}$ is a frozen retrieval index over publication chunks and $\mathcal{K}_{\text{DSL}} = \{e_i = (d_i, p_i, r_i)\}$ is a DSL library whose entries consist of a natural language description d_i, a HEPSCRIPT program p_i encoding observable analysis decisions, and its source chunk r_i, thereby retaining provenance for later reuse and evolution. The loop uses two complementary criteria: answer confidence c governs ***what is trustworthy***, and retrieval saturation z governs ***when to stop***. Our method consists of four components:

Algorithm 1 Generic HepScript template.

Step 1 – Datasets
    $D_{\text{data}} \leftarrow$ DATASETMANAGER.REAL_DATA.FIND("703_4260") ▷ At 4.260 GeV
    $D_{\text{incMC}} \leftarrow$ DATASETMANAGER.INCLUSIVE_MC.FIND("703_4260")
Step 2 – Decay card $\mathcal{C}_{\text{sig}}$
    $\mathcal{C}_{\text{sig}} \leftarrow$ Decay $Y(4230)$: $\pi^+\pi^- J/\psi$ (PHSP) ▷ $e^+e^- \to \pi^+\pi^- J/\psi$
Step 3 – Exclusive MC D_{exMC}
    $D_{\text{exMC}} \leftarrow$ CREATE_EXCLUSIVE_MC(related $= D_{\text{data}}$, events $= 10^5$, $\mathcal{C}_{\text{sig}}$)
Step 4 – Algorithm config
    $\mathcal{A} \leftarrow$ ANALYSIS(name = "pipiJpsi", output = root)
Step 5 – Selection Pipeline ▷ Analysis-specific
    TRACK SELECTION
    PHOTON SELECTION
    PARTICLE IDENTIFICATION
    KINEMATIC FIT
Step 6 – Note (un-expressible cut)
    NOTES IN CONTEXT ▷ free-text physics constraints
Step 7 – Bind & Execute
    $\mathcal{A}$.BIND($D_{\text{data}}, D_{\text{incMC}}, D_{\text{exMC}}, \mathcal{C}_{\text{sig}}$)
    ROOT Files $\leftarrow \mathcal{A}$.EXECUTE_ON($[D_{\text{data}}, D_{\text{incMC}}, D_{\text{exMC}}]$)
return ROOT Files

1. an uncertainty-aware confidence estimator that factorizes c into a statistical term H_{stat} and a systematic term H_{syst};
2. a self-evolving DSL knowledge base, where c provides dual gating—weighting precedents and gating write-back—allowing $\mathcal{K}_{\text{DSL}}$ to grow continuously from a static library;
3. an early-stopping controller that infers z from the dynamics of the posterior trajectory $\{p_t\}$ and determines the stopping round τ_t;
4. a retrieval loop that assembles the above modules into an auditable single session.

4.1 UNCERTAINTY-AWARE CONFIDENCE ESTIMATOR

The estimator evaluates whether a candidate precedent reduces the uncertainty of the generated DSL relative to a matched baseline. It computes a statistical entropy term H_{stat} from the language-model token probabilities and two complementary entropy summaries, H_{full} and H_{sel}, over the full generated code and the `selection` module, respectively. Their formal definitions are given in Appendix A. For query q and candidate k, the systematic-uncertainty reduction relative to the matched baseline is defined as

$$\delta H_j(q,k) = H_j^{\text{base}}(q) - H_j^{\text{cand}}(q,k), \qquad j \in \{\text{stat}, \text{full}, \text{sel}\}. \tag{1}$$

A positive value indicates that the candidate reduces the corresponding uncertainty. The raw reductions are standardized globally over all query–candidate pairs, yielding ΔH_{stat}, ΔH_{full}, and ΔH_{sel}. Although the statistical term is smaller than the other terms in the evaluated cells, it is retained in the score for methodological completeness. The candidate score is defined as

$$\text{score}(q,k) = \Delta H_{\text{stat}}(q,k) + \alpha\, \Delta H_{\text{full}}(q,k) + (1-\alpha)\, \Delta H_{\text{sel}}(q,k), \qquad \alpha \in [0,1]. \tag{2}$$

The candidate with the largest score is selected as the precedent. If k^* denotes the selected candidate, its confidence is

$$c(q) = \sigma(\text{score}(q,k^*))\,. \tag{3}$$

Structural disagreement corresponds to the systematic term H_{syst}, nalogous to evaluating systematic uncertainties in HEP by varying methodological choices or analysis paths. $H_{\text{syst,sel}}$ and $H_{\text{syst,glob}}$ are the JSDs of these symbols for the `selection` module and the full scripts, respectively, measuring structural disagreement across paths. Token-level predictive entropy reflects within-path statistical fluctuations, whereas structural disagreement reflects cross-path methodological sensitivity; the two signals are complementary.

4.2 SELF-EVOLVING DSL KNOWLEDGE BASE

The self-evolving DSL knowledge base extends each static entry $e = (d_i, p_i, r_i)$ with a write-back confidence score c_i: $e = (d_i, p_i, r_i, c_i)$. The original DSL script r_i is retained, while provenance records the agent version, timestamp, and parent entries used for synthesis, allowing every entry to be traced to a human-written seed. Precedent retrieval uses the same hybrid dense–sparse index as the QA module and returns the five nearest candidates. A synthesized program is admitted only if it passes two gates. The validation gate requires successful static parsing and, when supported, a clean dry run. The deduplication gate rejects a candidate when an existing entry has an AST-equivalent program—under variable renaming, canonical ordering of commutable cuts and filters, and constant folding—and its description similarity exceeds θ_{dup}. This prevents repeatedly writing the same program and skewing the retrieval distribution.

4.3 EARLY-STOPPING CONTROLLER

The adaptive early-stopping controller terminates retrieval when the posterior has become sufficiently stable and another round is unlikely to provide useful information. After each round, a lightweight probe model constructs a posterior p_t from the retrieval history. The controller combines a BOCPD-based stationarity score Stab_t, a learned estimate $\hat{g}_\theta$ of the next-round information gain, the cumulative search cost, and an all-dry indicator:

$$\psi_t = \left[\hat{g}_\theta(\phi(\mathcal{H}_t)), \mathrm{Stab}_t, t, C_t, \mathbb{1}_{\text{dry}}\right],$$
$$z_t = \sigma\left(\boldsymbol{w}^\top \psi_t + b\right),$$

where C_t is the cumulative number of search calls and $\mathbb{1}_{\text{dry}}$ indicates that the current round returned no documents. The gain predictor is trained by self-supervised regression on the Jensen–Shannon divergence between consecutive posteriors; further details are given in Appendix A.5. Retrieval stops when $z_t > z_{\text{th}}$ and $t \geq r_{\min}$, with $z_{\text{th}} = 0.7$ and $r_{\min} = 2$. Stopping is blocked during cold-start rounds and whenever all subagents return no hits.

5 EXPERIMENTS

In this section, we conduct comprehensive experiments to evaluate the three components in HepUKE – the uncertainty-aware confidence estimator, the self-evolving knowledge base, and the saturation-gated stopping controller. Specifically, we aim to answer the following questions.**Q1** (Confidence Vs Relevance): How does uncertainty-aware confidence estimator perform compared to a cross-encoder relevance model? **Q2** (Self Evolution): How does confidence perform when used as the write-back gate of a self-evolving DSL knowledge base? **Q3** (Efficiency Analysis): How cost-efficient and time-efficient is the adaptive early-stopping controller in knowledge-redundant domains? **Q4** (Ablation Study): What contribution does each component of HepUKE make to the over all performance?

5.1 EXPERIMENTAL SETTING

Datasets. We employ three corpora, all pre-embedded in a single Milvus index that combines dense and sparse retrieval. The primary resource for synthesis and knowledge-evolution studies is BESIII DSL, a collection of 700 expert-authored analysis programs partitioned into a 500-program knowledge library $\mathcal{K}_{\text{DSL}}$ and a held-out set of 200 targets; each entry is a triple $e = (d_i, p_i, r_i)$ linking a natural-language description, the executable DSL program, and its source publication chunk. For retrieval-loop evaluation we use BESIII-PhysicsQA, which comprises 1000 manually reviewed question–answer pairs drawn from 700 BESIII publications, each declaring one of three scorers (exact match, numeric, or set-F1) and accompanied by gold evidence chunks in the frozen index. As an out-of-domain control for the stopping signal we include HotpotQA (Yang et al., 2018), on which no confidence gating is applied. Paper isolation is enforced throughout, so a target's own siblings are never returned during retrieval. We evaluate evidence selection on BESIII-DSL, QA accuracy on BESIII-PhysicsQA, and adaptive stopping on HotpotQA. The construction protocol of the BESIII-PhysicsQA benchmark is detailed in Appendix D

Baselines. We categorize all baselines into three groups according to the decision they replace. (i) *Evidence selection*: We evaluate zero-shot generation without retrieval (No-RAG) as a lower anchor,

the standard cross-encoder RAG pipeline (Relevance top-1), and the post-hoc best candidate under gold as an upper anchor. (ii) *Knowledge admission*: We compare a static library with no write-back, a structural gate that enforces syntactic correctness and completeness, and a confidence threshold swept across a range. (iii) *Stopping*: We contrast fixed-round retrieval, native LLM self-termination, and our saturation-gated adaptive stopping. All configurations share an identical retriever, generator, and prompt, so that each comparison isolates the single decision under study.

Evaluation Metrics. We evaluate our method using several metrics across two categories. For program synthesis and evidence selection, we adopt metrics defined as: (i) *Strict match*, which checks whether datasets and decay cards are specified exactly; (ii) *Structural match*, which measures operator-level and parameter-key consistency while deliberately ignoring concrete cut values; (iii) *Phys-Acc*, which uses an LLM judge to assess whether the generated program encodes the same physics measurement as the ground truth, making this the only category that tolerates structural variation and is sensitive to numerical values. For retrieval efficiency and knowledge-base health, we report: (i) *Contain-Match Accuracy (Contain-Acc.)*, which checks if the correct answer appears in the generated response, (ii) *Costs*, which include rounds, search calls, tokens, and time spent. For knowledge base evolution, the *admission rate*, *rejection breakdown*, and *gate discrimination*—the downstream gap between admitted and rejected DSLs—to track the effect of write-back.

Implementations. All methods share one retrieval stack: local `BGE-M3` encoding (dense 1024-d and sparse) into a Milvus collections, hybrid recall, and `bge-reranker-v2-m3` cross-encoder reranking, with $k = 5$ throughout. All generation uses `deepseek-pro`: DSL synthesis at $T = 0.7$, sampled $N = 5$ per candidate. The saturation probe uses `deepseek-flash` at tempreture $=$ 0.7. Experiments are conducted on the hardware configuration detailed in Appendix B.

5.2 Confidence vs Relevance (Q1)

To address Q1, we fix the candidate pool, generator, and prompt, and vary only the strategy by which one of the $k = 5$ retrieved precedents is selected into the context. We evaluate no-RAG, standard RAG with a reranker, and different confidence-parameter configurations to assess the confidence estimator, and we further perform a module-wise uncertainty evaluation for HepScript. Our key findings are as follows:

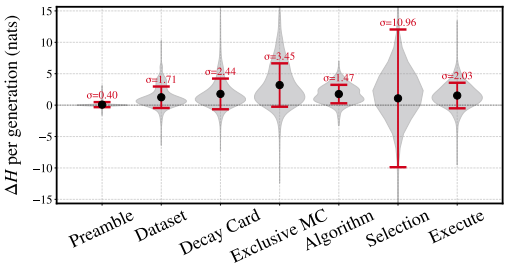

Figure 2: Distribution of the per-generation entropy reduction ΔH (in nats) across the seven DSL modules. Gray violins show the empirical distributions, black markers denote the means, and red bars denote one standard deviation, with the corresponding σ values annotated above each module. Positive ΔH indicates reduced generation uncertainty relative to the matched baseline.

Obs. 1. Uncertainty-aware confidence consistently outperforms relevance across protocols and metrics. Direct DSL generation (No-RAG) performs weakest on the held-out BESIII-DSL targets, reaching only 72.24%, 89.60%, and 58.54% of the Oracle-normalized scores for strict match, structural match, and physical accuracy, respectively. Introducing a relevance-ranked precedent improves all three metrics to 74.94%, 89.72%, and 62.46%. Confidence-aware selection yields further consistent gains, with the best configuration at $\alpha = 0.0$, where the three metrics reach 76.08%, 90.48%, and 65.02%, all exceeding those of relevance Top-1. Notably, performance decreases monotonically as α increases from 0.0 to 1.0, indicating that the selection-focused entropy reduction is more discriminative than the full-code entropy reduction; this is consistent with the observation that global structural uncertainty saturates easily. Nevertheless, although confidence-based selection substantially narrows the gap between model predictions and targets, it does not fully eliminate it.

Obs. 2. Uncertainty variation is concentrated in the task-specific selection segment. Decomposing ΔH by DSL segment shows that *Selection* has by far the widest distribution ($\sigma = 10.96$ nats) and substantial mass on both sides of zero, whereas the other segments are concentrated around positive reductions with $\sigma = 0.40$–3.45 nats. The reusable modules therefore benefit from prece-

Table 1: Precedent selection on 200 BESIII-DSL samples. Scores normalized to Oracle (%).

Strategy	Strict Match	Struct Match	PhysicsAcc.
NO-RAG	72.24	89.60	58.54
RELEVANCE TOP-1	74.94	89.72	62.46
Confidence (α=0.0)	**76.08**	**90.48**	**65.02**
Confidence (α=0.3)	75.59	89.88	64.18
Confidence (α=0.5)	75.17	89.41	64.42
Confidence (α=0.7)	74.76	88.83	64.27
Confidence (α=1.0)	74.57	88.77	63.79
ORACLE	100.00	100.00	100.00

Table 2: Write-back admission over 200 queries.

Gate	θ_{write}	θ_{high}	Admitted		Rej. low-conf		Rej. high-conf		Rej. struct
			n	struct	n	struct	n	struct	struct
Threshold	0.5	∞	191	0.93	2	0.50	0	–	0.52
	0.6	∞	134	0.92	79	0.83	0	–	0.51
	0.7	∞	39	0.89	208	0.83	0	–	0.52
Sweet Spot	0.5	0.95	189	0.94	2	0.50	5	0.48	0.5244
	0.5	0.90	189	0.94	2	0.50	7	0.55	0.52
	0.5	0.85	187	0.93	2	0.50	9	0.64	0.52
	0.5	0.80	183	**0.95**	2	0.50	16	0.57	0.53

dents relatively consistently, with the largest mean reduction observed for *Exclusive MC*. In contrast, the effect on the task-specific selection chain is strongly dependent on precedent compatibility and dominates the residual variation. Consistent with this localization, ΔH_{sel} alone achieves a 39.8% recovery rate, within 1.4 percentage points of the best fused setting.

5.3 SELF EVOLUTION (Q2)

To better understand whether confidence can serve as a dynamic library admission threshold, we conduct a direct evaluation of confidence as a thresholding criterion and, separately, examine whether the observed calibration structure can be converted into an improved selection rule.

Obs. 3. Uncertainty reduction has a sweet spot that only an interval gate can exploit. Table 2 reports a sweep of τ_{write} over the write-back protocol. Under the one-sided gate, increasing τ_{write} from 0.5 to 0.7 reduces admissions from 191 to 39 and decreases admitted quality from 0.9301 to 0.8861, whereas rejected low-confidence programs remain high-quality (0.8284). The interval gate behaves differently: holding $\theta_{\text{write}} = 0.5$ fixed and tightening θ_{high} from 0.95 to 0.80 raises admitted quality from 0.9352 to 0.9457, while rejecting 5–16 high-confidence but low-quality programs (0.4788–0.6435)—precisely the transcription tail that a monotone gate would admit as its most confident candidates.

This yields an inverted-U profile, as shown in Figure 3: quality peaks at 0.942 for $\Delta H_{\text{sel}} \in [+0.005, +0.010)$ ($n = 243$) and declines to 0.598 for $\Delta H_{\text{sel}} \geq +0.04$ ($n = 21$) and to 0.612 for $\Delta H_{\text{sel}} \leq -0.01$ ($n = 79$). The two tails correspond to distinct failure modes: the right tail transcribes precedent-specific cut values, decay modes, and PID thresholds that do not transfer, whereas the left tail conflicts with the target and raises uncertainty. In the peak interval, the precedent fixes the program's structure while the model re-derives the target's specifics, rendering the reference informative without inducing copying. Accordingly, the knowledge base should admit a generated program only when its ΔH_{sel} falls within this sweet spot, thereby rejecting both tails—an interval threshold rather than a monotone confidence threshold.

Table 3: Answer quality and computational cost with and without the early-stopping controller on BESIII PhysicsQA and HotpotQA.

Benchmark	Method	N	Rounds	Tokens		Accuracy (%)
				Prompt	Compl.	
BESIII PhysicsQA	Native	1,000	5.23	66,017	2,984	50.31
	HEPUKE		2.98	36,158	1,883	48.13
	Adaptive-RAG	1000	2.00	14,206	929	43.71
	FLARE		5.00	8,499	476	25.35
HotpotQA	Native	1,000	2.14	—		81.48
	HEPUKE		1.83	—		80.72

5.4 EFFICIENCY ANALYSIS

In order to evaluate the associated efficiency and cost implications of the early-stopping controller, we perform a dedicated analysis of the overhead and contain accuracy of HepUKE during the retrieval stage, comparing token usage and runtime with and without the early-stopping controller on the BESIII dataset. To further validate the domain transferability of the early-stopping method, we also evaluate retrieval overhead on the HotpotQA dataset. The results are presented in Table 3, and we summarize our key observations as follows:

Obs. 4. Early stopping exploits corpus redundancy for a favorable cost–accuracy trade-off. On BESIII PhysicsQA, HEPUKE reduces retrieval rounds from 5.23 to 2.98 (-43.0%), wall-clock time from 66.18 to 42.76 s (-35.4%), and total tokens from 69,001 to 38,041 (-44.9%). Accuracy decreases by only 2.27 percentage points, from 50.31% to 48.04%, retaining 95.5% of native accuracy. In this highly redundant corpus, the controller therefore removes substantial repeated computation while preserving most answer quality. This trade-off is favorable when the application places greater weight on inference cost and latency than on recovering the final few percentage points of accuracy. The other RAG baselines have different retrieval objectives and are not directly comparable; notably, both Adaptive-RAG and FLARE obtain lower accuracies of 43.71% and 25.35%, respectively. On HotpotQA, HEPUKE reduces time by 18.2% with only a 0.76-point decrease in answer F1, from 81.48% to 80.72%.

5.5 ABLATION STUDY

Several component-level ablations have been presented above, including uncertainty gating versus relevance top-1, early stopping versus a fixed retrieval budget, and interval admission versus a monotone threshold. We further examine the stability of confidence gating under retrieval-depth and paraphrase perturbations.

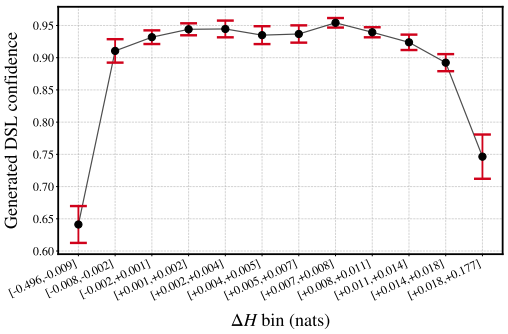

Figure 3: Binned relationship between per-generation entropy reduction ΔH_{bin} and generated DSL confidence. Black markers show mean confidence, and red bars indicate within-bin uncertainty. Confidence is highest for moderate positive ΔH_{bin} and decreases at both extremes.

Obs. 5. Confidence gating remains effective under retrieval and prompt perturbations. Table 4 compares No-RAG, relevance top-1, and confidence gating under a deeper candidate pool ($k = 10$) and paraphrased prompts. With $k = 10$, confidence gating provides consistent but moderate improvements over relevance ranking in all three metrics, increasing Strict Match from 75.7% to 76.3%, Structural Match from 90.6% to 90.8%, and Phys-Acc from 62.6% to 63.3%. Under paraphrase, the advantage is larger: confidence gating reaches 76.0%, 90.7%, and 63.9%, outperforming relevance top-1 by 0.8, 0.4, and 2.1 percentage points on Strict Match, Structural Match, and Phys-Acc, respectively. These results

Table 4: Results of perpetuated retrieval depth and paraphrase. Each variant is evaluated under No-RAG (NR), relevance top-1 (Rel), and confidence gating (Conf).

	Strict (%)			Struct (%)			Phys-Acc (%)		
Perturbation	NR	Rel	Conf	NR	Rel	Conf	NR	Rel	Conf
$k = 3$	71.8	76.2	**76.7**	89.3	91.0	**91.0**	57.6	62.4	**63.4**
$k = 10$	71.8	75.7	**76.3**	89.3	90.6	**90.8**	57.6	62.6	**63.3**
paraphrase	71.8	75.2	**76.0**	89.3	90.3	**90.7**	57.6	61.8	**63.9**
ORACLE		100			100			100	

indicate that the confidence signal remains useful when the retrieval pool is expanded and when the prompt wording changes, with the largest benefit appearing in physics-sensitive evaluation.

6 LIMITATIONS

Our study has several limitations. First, entropy reduction is an operational proxy for precedent usefulness rather than a direct measure of physical correctness. The inverted-U pattern observed in the write-back experiment shows that very large reductions may reflect inappropriate copying of precedent-specific cuts or decay configurations. The interval gate mitigates this behavior, but its thresholds are calibrated only on the current BESIII-DSL setting. Second, the DSL modules respond differently to precedent information: uncertainty reduction is concentrated in the task-specific `selection` module, while other modules exhibit smaller and more consistent changes. Our current estimator combines these signals globally and therefore does not fully exploit this module-specific structure. In addition, the experiments evaluate individual admission decisions rather than long-term knowledge evolution.

7 CONCLUSION

Summary. HEP analysis agents typically start new pipelines with few-shot knowledge from dedicated bases, which offer strategic guidance rather than ready-made solutions. Existing bases, however, treat all retrieved items as equally trustworthy and learn nothing from the analyses they assist. We introduce HEPUKE, which augments LLM statistical uncertainty with an HEP-specific systematic component from methodological choices. This estimator enables an append-only, self-evolving DSL library: high-confidence precedents constrain synthesis, while analyses without close precedents are validated and written back through an interval-shaped gate. For adaptive retrieval termination, we fuse a look-back stationarity test with a look-ahead marginal-information-gain estimate. On BESIII-DSL, confidence-aware selection yields further consistent gains, and confidence gating remains effective under retrieval and prompt perturbations. The interval gate improves admitted-program quality from 0.93 to 0.95 while rejecting the transcription tail. On the 1,000-item BESIII PhysicsQA benchmark, the controller cuts retrieval rounds by 43.0% and prompt tokens by 44.9% at essentially unchanged accuracy, and transfers to HotpotQA with cost savings preserved. The mechanism uses only standard LLM API outputs and is not tied to BESIII or a specific generator.

Future Work. We will further explore how to incorporate physics physical plausibility and logical correctness into the framework, including executable DSL checks, numerical consistency tests, and explicit physics constraints. We will also extend the approach to additional HEP experiments and DSL representations.

AI USE STATEMENT

We used generative AI tools (large language model assistants) in this work. We disclose the following uses.

Uses requiring disclosure: synthetic data generation; implementation of methods (code writing); dataset cleaning; and manuscript polishing. Details of the AI models, prompts, generation and cleaning pipelines, and human verification used for synthetic data generation and dataset cleaning are provided in Appendix C.1 and D.

Uses recommended for disclosure: language polishing; figure preparation; plotting and evaluation code; literature search; and brainstorming. Some categories overlap — for example, language polishing falls under manuscript polishing, and plotting/evaluation code falls under code writing.

Generative AI tools were not use for theoretical model construction or hypothesis generation. We also did not use AI for proving mathematical claims; this study is empirical and contains no formal theorems or proofs. The mathematical tools used in this work (information entropy, posterior stationarity tests, marginal information gain) are established methodological instruments, not claims requiring proof assistance.

All AI-assisted work was reviewed by the authors. The system codebase was written and refactored with the assistance of an LLM coding assistant. All such code was reviewed by the authors and covered by a unit-test suite; several LLM-introduced defects were identified and corrected through testing and manual review. We manually verified the existence and accuracy of each reference and did not retain any LLM-provided citation that we had not personally verified.

The PDF and supplementary materials contain no hidden instructions or invisible text directed at automated review systems. We assume responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide an anonymous code repository containing code, configurations, prompts, and evaluation scripts at `https://anonymous.4open.science/r/HepUKE-32FF`. Details of the uncertainty measures and controller are fully specified in Appendices A and A.5. BESIII-PhysicsQA is constructed following the deterministic seven-step protocol in Appendix D. The physics-accuracy evaluation follows Appendix C. Hardware and software configurations are reported in Appendix B.

REFERENCES

M. Ablikim et al. Design and construction of the besiii detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 614(3):345–399, March 2010. ISSN 0168-9002. doi: 10.1016/j.nima.2009.12.050. URL http://dx.doi.org/10.1016/j.nima.2009.12.050.

Ryan Prescott Adams and David J. C. MacKay. Bayesian online changepoint detection, 2007. URL https://arxiv.org/abs/0710.3742.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Mohammad Atif, Kriti Chopra, Ozgur Kilic, Tianle Wang, Zhihua Dong, Charles Leggett, Meifeng Lin, Paolo Calafiura, and Salman Habib. Celloai: Leveraging large language models for hpc software development in high energy physics. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, PASC '26, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400727344. doi: 10.1145/3815572.3815751. URL https://doi.org/10.1145/3815572.3815751.

Altan Cakir and Ayca Yerlikaya. From experimental limits to physical insight: A retrieval-augmented multi-agent framework for interpreting searches beyond the standard model, 2026. URL https://arxiv.org/abs/2605.02491.

Antonio J. Costa, Caterina Doglioni, Christian Gütschow, Andrew D. Pilkington, and Sukanya Sinha. Agentrivet: an automated system for producing rivet routines from journal publications, 2026. URL https://arxiv.org/abs/2606.13535.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024, pp. 719–729. ACM, 2024. doi: 10.1145/3626772.3657834. URL http://dx.doi.org/10.1145/3626772.3657834.

Jiaqi Deng. Computation over geometry: Meaning identity is computed, not shipped in the embeddings, 2026. URL https://arxiv.org/abs/2609.28290.

Aman Desai. Rooagent: An llm agent for root-based high energy physics analysis, 2026. URL https://arxiv.org/abs/2605.17318.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Mingfeng He, Fayu Jiang, Junkun Jiao, Mingrun Li, Ke Li, Yipu Liao, Beijiang Liu, Tong Liu, Fazhi Qi, Zijie Shang, Weimin Song, Yue Sun, Xiongfei Wang, Hong Wang, Dongbo Xiong, Changzheng Yuan, Bolun Zhang, Zhengde Zhang, and Xuliang Zhu. Dr.sai: An agentic ai for real-world physics analysis at besiii, 2026. URL https://arxiv.org/abs/2604.22541.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *CoRR*, abs/2403.14403, 2024. URL https://doi.org/10.48550/arXiv.2403.14403.

Ruobing Jiang, Dawei Fu, Cheng Jiang, Tianyi Yang, Zijian Wang, Youpeng Wu, Yong Ban, Yajun Mao, and Qiang Li. Agentic hybrid rag for evidence-grounded muon collider analysis, 2026. URL https://arxiv.org/abs/2606.10381.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=WLZX3et7VT.

Junkun Jiao, Tong Liu, Ke Li, Weimin Song, Yipu Liao, Bolun Zhang, Beijiang Liu, Chang-Zheng Yuan, and Yue Sun. Hepscript: A dual-use dsl for human-ai collaborative data analysis workflows in high-energy physics, 2026. URL https://arxiv.org/abs/2605.01423.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *EMNLP*, pp. 9004–9017, 2023. URL https://doi.org/10.18653/v1/2023.emnlp-main.557.

Eric A. Moreno, Samuel Bright-Thonney, Andrzej Novak, Dolores Garcia, and Philip Harris. Ai agents can already autonomously perform experimental high energy physics, 2026. URL https://arxiv.org/abs/2603.20179.

Ernesto Quevedo, Jorge Yero Salazar, Rachel Koerner, Pablo Rivas, and Tomas Cerny. Detecting hallucinations in large language model generation: A token probability approach. In Hamid R. Arabnia, Leonidas Deligiannidis, Soheyla Amirian, Farzan Shenavarmasouleh, Farid Ghareh Mohammadi, and David de la Fuente (eds.), *Artificial Intelligence and Applications*, pp. 154–173, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-86623-4.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=vAElhFcKW6.

Yue Sun, Tong Liu, Yipu Liao, Jingde Chen, and Ke Li. Reliable llm-generated programs for high-energy physics experiments through graph-grounded software knowledge, 2026. URL https://arxiv.org/abs/2609.01095.

Karthik Suresh, Neeltje Kackar, Luke Schleck, and Cristiano Fanelli. Towards a rag-based summarization for the electron ion collider. *Journal of Instrumentation*, 19(07):C07006, jul 2024. doi: 10.1088/1748-0221/19/07/C07006. URL https://doi.org/10.1088/1748-0221/19/07/C07006.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ehfRiF0R3a.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation, 2024. URL https://openreview.net/forum?id=JnWJbrnaUE.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *CoRR*, abs/1809.09600, 2018. URL http://arxiv.org/abs/1809.09600.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZS4m74kZpH.

C.H. Yu et al. BEPCII Performance and Beam Dynamics Studies on Luminosity. In *Proc. of International Particle Accelerator Conference (IPAC'16), Busan, Korea, May 8-13, 2016*, number 7 in International Particle Accelerator Conference, pp. 1014–1018, Geneva, Switzerland, June 2016. JACoW. ISBN 978-3-95450-147-2. doi: doi:10.18429/JACoW-IPAC2016-TUYA01. URL http://jacow.org/ipac2016/papers/tuya01.pdf. doi:10.18429/JACoW-IPAC2016-TUYA01.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *AAAI*, pp. 19632–19642, 2024. URL https://doi.org/10.1609/aaai.v38i17.29936.

A Operational Uncertainty Features

A.1 Scope and Notation

The uncertainty measures used in this work are operational proxies for predictive uncertainty and structural disagreement. They are computed from repeated DSL generations under a baseline condition or a candidate-reference condition. They do not construct an answer-class mixture and do not estimate a signed systematic bias relative to a ground-truth analysis.

Let r denote a generation condition, such as the baseline condition or the condition in which candidate precedent k is provided. For each condition, multiple DSL samples are generated independently. The resulting quantities are first computed within each cell and are then compared between matched baseline and candidate cells.

A.2 Token-Level Predictive Entropy

For a generated token t, let $p_{t,i}$ denote the probability of the i-th token among the reported top-20 log-probability entries. The probability mass outside the reported entries is

$$m_t = 1 - \sum_{i=1}^{20} p_{t,i}. \tag{4}$$

The token-level predictive entropy is defined by

$$H_{\text{tail}}(t) = -\sum_{i=1}^{20} p_{t,i} \ln p_{t,i} - m_t \ln m_t, \tag{5}$$

with the convention $0 \ln 0 = 0$. The residual term accounts for the probability mass not included in the top-20 entries. Let $\mathcal{T}_{\text{code}}^{(n,r)}$ be the set of kept code tokens in sample n under condition r, excluding tokens assigned to the `note` module. The code-only entropy of this sample is

$$H_{\text{stat,code}}^{(n,r)} = \frac{1}{\left|\mathcal{T}_{\text{code}}^{(n,r)}\right|} \sum_{t \in \mathcal{T}_{\text{code}}^{(n,r)}} H_{\text{tail}}(t). \tag{6}$$

For a cell containing N generated samples, the corresponding cell-level quantity is

$$H_{\text{stat,code}}^{(r)} = \frac{1}{N} \sum_{n=1}^{N} H_{\text{stat,code}}^{(n,r)}. \tag{7}$$

For candidate comparison, two additional token-entropy summaries are computed. Let $\mathcal{T}_{\text{full}}^{(n,r)}$ denote all kept code tokens, including tokens in `note` modules, and let $\mathcal{T}_{\text{sel}}^{(n,r)}$ denote the subset belonging to the `selection` module. Then

$$H_{\text{full}}^{(n,r)} = \frac{1}{\left|\mathcal{T}_{\text{full}}^{(n,r)}\right|} \sum_{t \in \mathcal{T}_{\text{full}}^{(n,r)}} H_{\text{tail}}(t), \tag{8}$$

$$H_{\text{sel}}^{(n,r)} = \frac{1}{\left|\mathcal{T}_{\text{sel}}^{(n,r)}\right|} \sum_{t \in \mathcal{T}_{\text{sel}}^{(n,r)}} H_{\text{tail}}(t). \tag{9}$$

Their cell-level means are

$$H_j^{(r)} = \frac{1}{N} \sum_{n=1}^{N} H_j^{(n,r)}, \qquad j \in \{\text{full}, \text{sel}\}. \tag{10}$$

A.3 Structural Disagreement from Canonicalized DSL

Structural disagreement is measured from discrete representations of the generated DSL rather than from token probability distributions. Complete `note` blocks are first removed from each generated program. The remaining source is canonicalized by removing comments, replacing string and

numerical literals with fixed placeholders, collapsing whitespace, and discarding empty lines. The canonicalized source is then mapped to a discrete structural symbol by a deterministic hash function.

Let $s_{\text{sel}}^{(n,r)}$ denote the symbol obtained from the canonicalized `selection` module in sample n, and let $s_{\text{glob}}^{(n,r)}$ denote the symbol obtained from the complete note-stripped program. For either scope $a \in \{\text{sel}, \text{glob}\}$, define the empirical symbol distribution

$$\widehat{p}_a^{(r)}(v) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\left[s_a^{(n,r)} = v\right]. \tag{11}$$

The structural disagreement quantity is

$$H_{\text{syst},a}^{(r)} = -\sum_{v} \widehat{p}_a^{(r)}(v) \ln \widehat{p}_a^{(r)}(v), \qquad a \in \{\text{sel}, \text{glob}\}. \tag{12}$$

The implementation evaluates this quantity through a generalized Jensen–Shannon divergence routine. Because each sample contributes one discrete symbol, its within-sample distribution is one-hot and has zero entropy. Consequently, the implemented quantity reduces exactly to the empirical entropy in Eq. equation 12. It therefore measures disagreement among canonicalized DSL structures, rather than disagreement among token probability distributions.

A.4 Baseline–Candidate Deltas and Confidence Score

For a query identifier q, let $X^{\text{base}}(q)$ be a cell-level feature computed under the baseline condition and $X^{\text{cand}}(q,k)$ the corresponding feature computed when candidate k is used as the reference. The raw candidate-induced change is defined by

$$\delta X(q,k) = X^{\text{base}}(q) - X^{\text{cand}}(q,k). \tag{13}$$

A positive value indicates that the candidate reduces the corresponding entropy or structural-disagreement measure.

For each feature X, the raw deltas are standardized globally over all query–candidate pairs:

$$\Delta X(q,k) = \frac{\delta X(q,k) - \mu_X}{\sigma_X}, \tag{14}$$

where μ_X and σ_X are the mean and standard deviation of the complete set of raw deltas for feature X. No task-specific or dataset-specific grouping is applied.

The candidate score combines the standardized full-code and selection-module entropy reductions:

$$\text{score}(q,k) = \Delta H_{\text{stat}}(q,k) + \alpha\, \Delta H_{\text{full}}(q,k) + (1-\alpha)\, \Delta H_{\text{sel}}(q,k), \qquad \alpha \in [0,1]. \tag{15}$$

The selected candidate is

$$k^* = \arg\max_k \text{score}(q,k), \tag{16}$$

and its confidence is

$$c(q) = \sigma(\text{score}(q,k^*)). \tag{17}$$

Thus, the confidence score increases when the selected precedent produces a larger standardized reduction in full-code and selection-module predictive entropy relative to the matched baseline.

A.5 Controller Implementation Details

The probe posterior comes from a single short forward pass over $\mathcal{H}_t$ and a softmax over the top-B answer-token log-probabilities ($B = 20$). Its normalized certainty is

$$u_t = 1 - \frac{H(p_t)}{\log B}.$$

BOCPD uses a one-dimensional Beta observation model with hazard $H = 0.1$, prior $(\alpha_0, \beta_0) = (1,1)$, and run-length pruning threshold 10^{-6}. The run-length posterior is updated in the standard

way, with sufficient statistics $\alpha_r \leftarrow \alpha_r + u_t$ and $\beta_r \leftarrow \beta_r + 1 - u_t$; a new run restarts from the prior. Observations and Beta-density endpoints are clipped to $[0, 1]$ and 10^{-12}, respectively. If normalization underflows, the posterior resets to the prior and $\mathrm{Stab}_t = 0$.

The predictor input is a fixed feature vector combining round index, posterior statistics, document-set novelty, and recent realized-gain summaries:

$$\phi(\mathcal{H}_t) = (t, H(p_t), \max p_t, \mathrm{Var}(p_t), n_{\mathrm{unique}}, 1-J_{\mathrm{doc}}, \overline{g}_3, \mathrm{Var}(g_3), \mathrm{slope}(g_3), 1-\min(1, n_{\mathrm{unique}}/5)),$$

where J_{doc} is the Jaccard similarity between current and historical document sets. The self-supervised target is the JSD between consecutive posteriors,

$$g_{t+1} = \mathrm{JSD}(p_{t+1} \| p_t),$$

using the shorter prefix when posterior lengths differ.

The fusion vector is

$$\psi_t = (\hat{g}_\theta(\phi(\mathcal{H}_t)), \mathrm{Stab}_t, t, C_t, \mathbb{1}_{\mathrm{dry}}),$$

where C_t is the cumulative number of search-tool calls and $\mathbb{1}_{\mathrm{dry}}$ is the all-dry indicator. The latter is also a hard safety condition, while its learned coefficient is zero. We search a single-stage grid of $14{,}000$ candidates over weights, bias, and threshold, subject to $R_{\mathrm{gold}} \geq R_{\mathrm{gold}}^{\mathrm{base}} - 0.02$, and rank them lexicographically by rounds saved and then R_{gold}, where

$$R_{\mathrm{gold}} = \mathbb{1}\left[\text{gold titles} \subseteq \bigcup_{t \leq t_{\mathrm{stop}}} \text{doc_ids}_t\right]$$

is a lower bound on answer accuracy, since retrieving the gold documents is necessary but not sufficient for an exact answer.

B Experimental Hardware Configuration

Table 5: Hardware and software configuration used in the experiments.

Component	Configuration
Architecture	x86_64
CPU	2 × Intel Xeon Gold 6430
CPU cores	64 physical cores (32 per socket)
Maximum CPU frequency	3.4 GHz
GPU	NVIDIA GeForce RTX 5090
GPU memory	32 GB
NVIDIA driver	580.95.05
CUDA	13.0
L3 cache	120 MB
NUMA nodes	2

C Physics-Acc. Evaluation Protocol

We evaluate the generated HepScripts using a strict, dimension-wise comparison between a reference specification and a candidate specification. The evaluation is performed by an LLM prompted to act as a senior BESIII analyst. It compares the physics content of the two DSL programs while ignoring variable names, comments, indentation, statement ordering, and free-text notes. The comparison covers seven dimensions: decay chain and final state, dataset and energy, topology and tagging, track and photon selection, particle identification, fits and constraints, and MC configuration and decay card. For each dimension, the evaluator records the evidence, assigns a score, and lists substantive and minor differences. A dimension is assigned `null` only when the reference does not address that aspect. If the candidate omits a requirement present in the reference, it receives a score of zero for the corresponding dimension.

Any difference in a discrete physics choice is treated as substantive. This includes the parent or daughter particle, decay mode, tagged side, PID hypothesis set, fit type, generator model, energy

Table 6: Dimensions used for physics-Acc. evaluation.

Dimension	Compared content
Decay chain and final state	Parent particle, complete decay chain, and reconstructed final-state particles.
Dataset and energy	Real-data and inclusive-MC samples, dataset identifiers, and centre-of-mass energy.
Topology and tagging	Single- or double-tag topology, tagged and signal sides, and recoil-mass or direct reconstruction.
Track and photon selection	Track quality, multiplicity, net charge, photon thresholds, timing, isolation, and photon multiplicity.
PID	Identification method, likelihood or probability cuts, tested hypotheses, and particle multiplicities.
Fits and constraints	Fit type, fitted particle list, constraints, and χ^2 requirements.
MC and decay card	Exclusive-MC setup, EvtGen decay chain, generator models, and requested event count.

Table 7: Scoring criteria for physics-Acc. evaluation.

Score	Interpretation
100	Substantively identical; differences are purely notational.
80	Physically equivalent; only minor threshold differences are present, with all relative deviations below 20%.
60	Same overall intent, but one substantive difference or several minor differences that may visibly change the selected sample.
40	Partially overlapping; the dimension is addressed but contains multiple substantive errors.
20	Addressed only nominally; most of the specified content is incorrect.
0	The dimension is absent from the candidate or directly contradicts the reference.

point, or dataset identifier. For continuous requirements, a relative deviation greater than 20% is substantive, whereas a deviation below 20% is minor. An omitted requirement is always treated as substantive.

The *Phys-Acc* score is assigned by an LLM judge according to the above protocol. It measures whether the candidate specification encodes the same physics measurement as the reference, while allowing physically equivalent differences in syntax and structure. The judge returns a JSON object containing the summaries, identified differences, dimension-wise scores, a fatal-mismatch flag, and a final verdict. A fatal mismatch is reported when the candidate describes a fundamentally different parent particle or decay chain that cannot be reconciled by adjusting numerical thresholds.

C.1 LLM JUDGE PROMPT

The following prompt is used to evaluate the *Phys-Acc* of a candidate HepScript.

***Phys-Acc* Evaluation Prompt**

```
You are a senior BESIII collaboration analyst performing a strict technical
review. You are given two BOSS analysis specifications written in a Ruby
domain-specific language: the REFERENCE, which is the accepted specification,
and the CANDIDATE, which is to be reviewed.

Be strict. Do not assume that a difference is acceptable. A candidate that
selects different events is a failure even if it appears superficially similar.

Any difference in a discrete choice is substantive, including the particle,
decay mode, tagged side, PID hypothesis set, fit type, generator model, energy
point, or dataset identifier. For continuous thresholds, a relative deviation
above 20\% is substantive and a deviation below 20\% is minor. An omitted
requirement imposed by the reference is substantive.

Ignore variable names, comments, indentation, statement ordering, and the
```

```
content of free-text note blocks.

Evaluate the following dimensions: decay chain and final state; dataset and
energy; topology and tagging; track and photon selection; PID; fits and
constraints; and MC configuration and decay card. Use the scoring criteria
defined in Table~\ref{tab:physacc-criteria}. Return null only when a dimension
is not addressed by the REFERENCE.

Respond with a single JSON object and nothing else. Fill the fields in order:
evidence, per-dimension scores, and summary fields.

{
  "reference_summary": "<1-2 sentences>",
  "candidate_summary": "<1-2 sentences>",
  "substantive_differences": ["<one difference per item>"],
  "minor_differences": ["<one difference per item>"],
  "scores": {
    "decay_chain_and_final_state": <0-100 or null>,
    "dataset_and_energy": <0-100 or null>,
    "topology_and_tagging": <0-100 or null>,
    "track_photon_selection": <0-100 or null>,
    "pid": <0-100 or null>,
    "fits_and_constraints": <0-100 or null>,
    "mc_and_decay_card": <0-100 or null>
  },
  "fatal_mismatch": true | false,
  "verdict": "<one sentence>"
}
```

D BESIII BENCHMARK DATA DESCRIPTION

D.1 CONSTRUCTION PROTOCOL FOR BESIII-PHYSICSQA

BESIII-PhysicsQA comprises 1,000 retrieval QA pairs, covering 756 published BESIII papers. Construction follows a deterministic seven-step protocol. Its core principle is that the language model determines no facts: hop chains, gold facts, and answer values are all produced by deterministic queries over a prebuilt fact index. The model participates only in structural annotation and question rewriting, and outputs only line numbers or question strings.

Structural annotation. Each paper is annotated with section spans, canonical types, independent discussion units, and table attribution. The model outputs only line numbers, making text-block identifiers a pure function of the frozen artifact and gold locations invariant across reconstructions. The whole corpus is annotated uniformly to avoid different length distributions between gold and distractor blocks.

Text-block materialization and fact extraction. Text blocks are split according to annotations, length budgets are applied by canonical section type, and each table is promoted to a separate text block. Facts are also extracted, and an index and construction report are generated. No model is involved in this step.

Deterministic item selection and assembly. Realistic scientific task scenarios are written first; items are discarded if none can be written. Candidate anchors are queried by category and passed through template-based gates and a Monte Carlo generator consensus gate. Answers are computed by query; systematic-error cases are arithmetic cross-checked, not model-generated.

Isolation validation. Hop count is defined as the minimum text-block set cover over all gold facts, solved exactly. Items are rejected one by one for cases such as a single block containing all gold facts, answer-value leakage, excessively low anchor selectivity, and arithmetic non-reproducibility. Physics papers are highly self-contained; within-paper multi-hop structure is far less common than it appears, so isolation is verified item by item against real text blocks.

Natural-language rewriting. This is the only step that calls the model per item. Inputs are the scenario, fixed hop chain, gold facts, answer, and eight hard checks; only a question string is requested. Checks include:

- prohibit reverse lookup of papers from known parameters;

- restrict length and a single main condition;
- require a scientific-task motivation;
- prohibit artificial hop-chain elongation;
- prohibit leaking locations or arithmetic;
- prohibit pipeline meta-vocabulary;
- prohibit rare-parameter conjunctions as identifiers;
- require a unique, machine-verifiable answer.

If any check fails, rewrite once; if it fails again, discard the item.

Adversarial validation. Gold text blocks are hidden and only retrieved distractors are provided; if the model still answers correctly, the item does not test retrieval and is discarded.

Export. At export, clustering keys are enforced and the correspondence between QA pairs and the index is frozen.