Real-Time Trigger for SCEP Magnetic Monopole Detection

(Dated: April 16, 2026)

Magnetic monopoles are hypothetical particles of fundamental interest in modern physics. Conventional searches are often experimentally complex and difficult to scale. The Search for Cosmic Exotic Particles (SCEP) experiment addresses this challenge with a scalable detection architecture based on room-temperature induction coils. Its proposed three-layer array, with an exposure of 20 000 $\mathrm{m}^2 \cdot \mathrm{year}$, is designed for large-area monopole searches. For such a detector system, the limitations of raw-data storage and transmission make an online coarse trigger essential. Because the induction signal has a relatively low signal-to-noise ratio (SNR), the trigger operates in a high-background regime in which most outputs are expected to be noise-induced candidates. Since the retained candidates can be further examined offline, the goal of the coarse trigger is to preserve as many true signals as possible at a prescribed false-positive rate (FPR), or equivalently, to minimize the false-negative rate (FNR) at fixed FPR. In the original trigger scheme, the waveform is processed using a single fixed template followed by threshold comparison, and the performance degradation caused by template mismatch is not explicitly quantified or controlled. In this work, we first construct a template-bank-based trigger and show that it significantly outperforms the original single-template scheme at fixed FPR. We then develop a physically informed neural-network trigger based on the template-bank response, which further reduces the FNR under realistic non-ideal conditions. In addition, we implement the coarse trigger on an FPGA and demonstrate a steady-state update interval below 1 μs, satisfying the real-time requirement. These results establish a practical coarse-trigger solution for a large-scale induction-coil monopole search.

I. INTRODUCTION

A magnetic monopole (MM), first proposed by Dirac in 1931 [1], is a hypothetical particle carrying an isolated magnetic charge. Magnetic monopoles are of fundamental interest in modern physics because their existence would naturally explain electric-charge quantization through the Dirac quantization condition, and they are also predicted in various grand unified theories (GUTs) [2, 3]. Over the past decades, extensive experimental efforts have been devoted to MM searches using a variety of detection principles. Conventional searches can be broadly divided into two categories. The first relies on induction-based detection with superconducting coils [4, 5], where the passage of an MM produces a quantized change in magnetic flux that can be measured with extremely high sensitivity. The second comprises energy-deposition-based methods, in which a monopole is identified through ionization or excitation signatures in detector media. Representative examples include scintillator-based searches in underground ultra-low-background experiments such as MACRO [6] and Cherenkov-light searches in large-volume neutrino telescopes such as IceCube [7]. Although these techniques have established stringent limits on monopole flux, they often involve substantial experimental complexity and limited scalability when very large exposures are required. To address these challenges, an alternative induction-based approach using room-temperature coils has been proposed. Their simpler experimental implementation and better scalability make them well suited to multilayer, large-area detector configurations. The Search for Cosmic Exotic Particles (SCEP) experiment adopts this strategy through a hybrid detection architecture that combines room-temperature induction coils with plastic scintillators [8]. With a projected exposure of 20,000 $\mathrm{year} \cdot \mathrm{m}^2$ and at least three induction-coil layers, an induction-signal signal-to-noise ratio (SNR) above 4.5 is expected to bring the monopole-flux sensitivity close to the most stringent existing limits.

For a detector system of this scale, continuous raw-waveform readout makes an online coarse trigger indispensable. Assuming an ADC resolution of $b = 20$ bit, a sampling rate of $f_s = 1$ MSps, a coil radius of $r = 10$ cm, and the SCEP design parameters of an effective exposure $\Lambda = 20{,}000\ \mathrm{year} \cdot \mathrm{m}^2$ with $N_L = 3$ induction-coil layers, the total raw waveform data volume required to achieve the target exposure can be estimated as

$$S_{\mathrm{raw}} = \frac{\Lambda}{\pi r^2} N_L\, b\, f_s \approx 151\ \mathrm{EB}. \tag{1}$$

This estimate indicates that, without online preselection, direct storage of all raw waveforms is impractical under realistic engineering constraints. As illustrated in Fig. 1, the online trigger determines whether an input waveform segment should be retained as a candidate event for subsequent offline analysis, while final refined identification can be performed offline using the stored candidates together with plastic-scintillator information and multilayer induction information. Because the induction signal has a relatively low signal-to-noise ratio (SNR), the trigger operates in a high-background regime in which most outputs are expected to be noise-induced candidates. In this regime, the retained data volume approximately scales with the false-positive rate (FPR). Requiring FPR $= 10^{-4}$ reduces the total storage demand from 151 EB/year to approximately 15.1 PB/year, bringing it into a range comparable to that of large contemporary experiments. Since the retained candidates can be further examined offline, the goal of the coarse trigger is to preserve as many true signals as

possible at a prescribed FPR, or equivalently, to minimize the false-negative rate (FNR) at fixed FPR. Accordingly, in this work we adopt the FNR at FPR = 10^{-4} as the primary optimization metric for online coarse triggering.

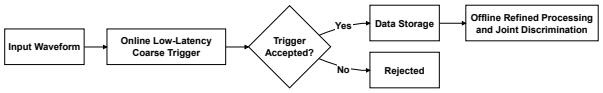

FIG. 1. Schematic of the online coarse-trigger strategy in SCEP. The online trigger processes the streaming induction waveform and decides whether a waveform segment should be stored as a candidate event for subsequent offline refined identification.

The optimal-filter-plus-threshold paradigm has been widely used in rare-event searches, including gravitational-wave searches such as LIGO [9, 10]. In this approach, optimal filtering [11] is first applied to the raw signal waveform. Under the assumed signal and noise model, optimal filtering maximizes the output SNR, and the resulting filter output is then compared with a fixed threshold to determine whether a candidate is present. In the original SCEP study, all candidate signals were processed using a single reference template corresponding to a normally incident monopole passing through the coil center with a velocity of $10^{-5}c$. Although monopole-induced waveforms share similar dominant spectral features, residual template mismatch inevitably degrades the output SNR for nonmatching trajectories and therefore increases the FNR. To mitigate this limitation, we introduce a template bank, apply optimal filtering with each template in the bank, and use the maximum bank response as the trigger statistic. In this way, the trigger effectively selects the template that best matches the input waveform, thereby improving coverage of the physically admissible parameter space and reducing the miss probability. At the same time, because the max operation also enhances the background response, the bank size must be carefully controlled rather than increased without bound.

In practical operation, however, both the signal and noise characteristics may deviate from the ideal model because of manufacturing tolerances, temperature drift, long-term drift, and other non-ideal effects. Constructing a dedicated template bank for every such variation is therefore impractical. This motivates the second stage of optimization considered in this work. Starting from the template-bank response derived under the ideal signal and noise model, we construct a physically informed neural-network trigger, which is expected to provide stronger robustness to non-ideal signals and waveform distortions, thereby improving the detection probability for realistic monopole-like events that deviate from the ideal template model.

Finally, we implement the proposed coarse trigger on an FPGA and verify that the resulting hardware design satisfies the real-time requirement of the SCEP online trigger. The hardware study addresses output consistency with the software reference and steady-state throughput, and shows that the trigger achieves a steady-state update interval below 1 μs. Taken together, these results establish a practical online coarse-trigger solution for large-scale room-temperature induction-coil monopole searches.

The remainder of this paper is organized as follows. Section II presents the software-side optimization of the coarse trigger, including template-bank construction and the physically informed neural-network trigger. Section III describes the real-time FPGA implementation and hardware validation. Section IV concludes the paper.

II. SOFTWARE-SIDE OPTIMIZATION OF THE COARSE TRIGGER

This section presents the software-side optimization of the online coarse trigger. The optimization is carried out in two stages. First, the conventional single-template threshold discriminator is replaced by a template-bank threshold trigger in order to mitigate the performance loss caused by template mismatch over the physically admissible signal space. Second, a physically informed neural-network trigger is constructed based on the template-bank response, so that the decision stage can be further optimized for lower miss probability under realistic non-ideal conditions by exploiting the robustness of neural-network-based discrimination.

A. Template-Bank Construction

For a template signal with spectrum $S(f)$ and noise power spectral density $P_n(f)$, the corresponding optimal-filter transfer function is given by [11]

$$\mathcal{H}_{\mathrm{OF}}(f) = \frac{S^*(f)}{P_n(f)}. \tag{2}$$

In the present work, $S(f)$ denotes the spectrum of the monopole-induced induction signal, while $P_n(f)$ denotes the total noise power spectral density, dominated by coil thermal noise together with additional readout-noise contributions.

For a streaming waveform window $\mathbf{x}$ and a template-dependent optimal-filter kernel $\mathbf{g}_\gamma$, the corresponding filter response can be written as

$$\rho_\gamma = \langle \mathbf{x}, \mathbf{g}_\gamma \rangle, \tag{3}$$

where $\mathbf{g}_\gamma$ denotes the time-reversed optimal-filter template.

For the original single-template trigger, let γ_0 denote the reference template. The trigger statistic is

$$T_{\mathrm{single}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{g}_{\gamma_0} \rangle, \tag{4}$$

and the corresponding threshold decision is

$$\delta_{\mathrm{single}} = \begin{cases} 1, & T_{\mathrm{single}}(\mathbf{x}) \geq \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where τ denotes the trigger threshold.

When a template bank B is used, the trigger statistic is generalized to

$$T_{\mathrm{bank}}(\mathbf{x}) = \max_{\gamma \in B} \langle \mathbf{x}, \mathbf{g}_{\gamma} \rangle, \tag{6}$$

and the corresponding threshold decision becomes

$$\delta_{\mathrm{bank}} = \begin{cases} 1, & T_{\mathrm{bank}}(\mathbf{x}) \geq \tau, \\ 0, & \text{otherwise}. \end{cases} \tag{7}$$

The single-template trigger can be regarded as a special case of the template-bank trigger with $B = \{\gamma_0\}$. Compared with the single-template scheme, the template-bank trigger reduces the mismatch loss by selecting the best-matched template response, while its bank size must be carefully controlled because the max operation also enhances the background response.

The template bank is constructed by a greedy FNR-driven procedure. For a fixed coil configuration, the family of monopole-induced waveforms is defined over the physically admissible parameter space

$$\Gamma = [v_{\min}, v_{\max}] \times [\rho_{\min}, \rho_{\max}] \times [\theta_{\min}, \theta_{\max}] \times [\phi_{\min}, \phi_{\max}], \tag{8}$$

with

$$\begin{aligned} v_{\min} &= 10^{-5}c, & v_{\max} &= 10^{-1}c, \\ \rho_{\min} &= 0, & \rho_{\max} &= R_{\mathrm{inner}}, \\ \theta_{\min} &= 0, & \theta_{\max} &= \pi/2, \\ \phi_{\min} &= 0, & \phi_{\max} &= 2\pi. \end{aligned} \tag{9}$$

A large candidate pool is first sampled over Γ. The construction starts by evaluating each candidate as a single-template bank and selecting the one with the best FNR@FPR $= 10^{-4}$ performance as the initial template. The bank is then expanded iteratively: at each step, every remaining candidate is combined with the current bank to form a temporary bank, and the candidate giving the best improvement is added. The iteration stops when adding a new candidate no longer yields a meaningful reduction in FNR.

The template-bank construction results are summarized in Figs. 2 and 3. All results in this study are evaluated at SNR $= 4.5$ using FNR@FPR $= 10^{-4}$ as the primary metric. For the original single-template trigger, using the reference template $\gamma = (10^{-5}c, 0, 0, 0)$ yields an FNR of 0.635. In contrast, the first template selected by the proposed greedy FNR-driven bank construction already reduces the FNR to 0.231, corresponding to a relative reduction of 63.6%. As the bank is further expanded, the FNR reaches 0.228 at $N = 8$ with an additional 1.7% improvement relative to the first selected template. For comparison, a velocity-only scan with the same number of templates, but without explicit FNR-driven optimization over the full parameter space, consistently gives worse performance and yields an FNR that remains noticeably higher than that of the proposed template bank over the entire range. In particular, the FNR of the velocity-only scan begins to rise slightly near $N = 8$, whereas the template-bank result is already close to saturation at this bank size. This comparison shows that scanning only the velocity dimension is insufficient to capture the waveform diversity induced by the full physically admissible parameter space. At the same time, increasing the number of templates also enhances the background response because the trigger statistic is defined by the maximum response over the bank. As a result, a higher threshold is required to maintain the same FPR. This trend is observed for both methods, but the proposed template bank exhibits a more moderate increase. The proposed greedy template bank not only achieves a lower FNR than the velocity-only scan, but also does so with a smaller background penalty. A bank size of eight templates provides a practical operating point, at which the FNR reduction is already close to saturation while the additional background growth remains limited.

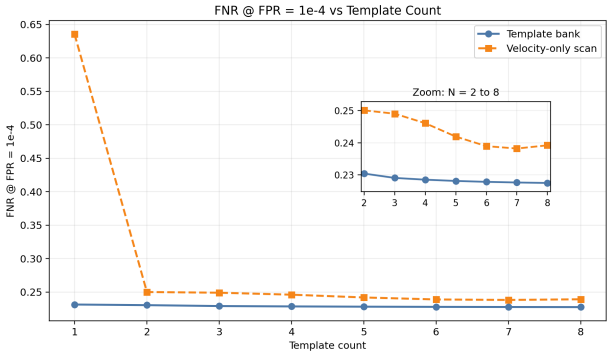

FIG. 2. FNR at FPR $= 10^{-4}$ as a function of template count. The greedy template bank consistently outperforms the velocity-only scan, and the FNR improvement becomes nearly saturated when the template count approaches eight.

B. Physically Informed Neural-Network Trigger

In practical applications, both signal and noise characteristics can deviate from the ideal model because of manufacturing tolerances, temperature variations, long-term drift, and other non-ideal effects. Under such conditions, constructing and maintaining a dedicated template bank for every possible variation is not practical. To address this issue, we introduce a physically informed neural-network trigger initialized from the template-bank response. In this way, physical prior information is retained at initialization, while the subsequent decision function is optimized in a data-driven manner, enabling improved robustness to

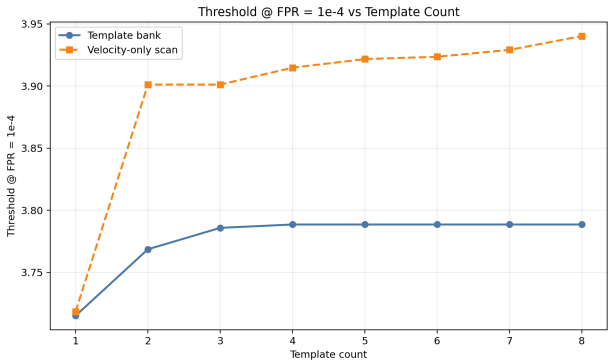

FIG. 3. Threshold at FPR $= 10^{-4}$ as a function of template count. As the template count increases, the threshold must be raised to suppress the stronger background response induced by the max operation. The template bank requires a lower threshold than the velocity-only scan at the same operating point.

realistic signal variations and lower FNR under non-ideal conditions.

Given a template bank $\mathcal{T} = \{\gamma_k\}_{k=1}^{K}$, the first-layer feature vector is written as

$$\mathbf{z} = \mathbf{W}_1\mathbf{x}, \qquad \mathbf{W}_1 = \begin{bmatrix} \mathbf{g}_{\gamma_1}^{\mathsf{T}} \\ \mathbf{g}_{\gamma_2}^{\mathsf{T}} \\ \vdots \\ \mathbf{g}_{\gamma_K}^{\mathsf{T}} \end{bmatrix}, \tag{10}$$

where each row of $\mathbf{W}_1$ is initialized by the corresponding optimum filter from the template bank.

In the conventional template-bank trigger, as defined in Eq. (6), the optimum-filter responses of all templates are first computed, and the final trigger statistic is obtained by taking their maximum. Instead of treating this maximum operation as an external fixed rule, we express it explicitly in neural-network form using ReLU activations and implement it through a multilayer reduction structure. The rectified linear unit is defined as

$$\mathrm{ReLU}(x) = \max(x, 0). \tag{11}$$

Using $\mathrm{ReLU}(\cdot)$, the maximum of two scalars can be expressed as

$$\max(a, b) = \mathrm{ReLU}(b) - \mathrm{ReLU}(-b) + \mathrm{ReLU}(a - b). \tag{12}$$

By recursively applying Eq. (12), the maximum over all template responses can be implemented by a multilayer ReLU network.

In this way, the conventional template-bank trigger can be interpreted as a physically informed neural network whose first layer is initialized by the template-bank optimum filters and whose subsequent layers realize the max-reduction operation. Starting from this structured initialization, the network can then be further trained at the target operating point, leading to improved robustness to realistic signal and noise deviations and hence lower FNR under non-ideal conditions.

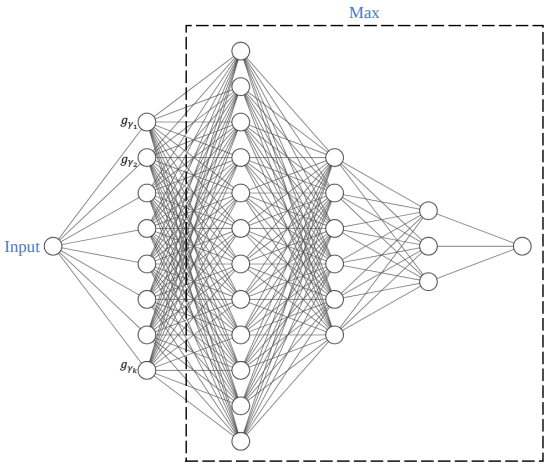

FIG. 4. Physically informed neural-network interpretation of the template-bank trigger. The first layer is initialized from the template-bank optimum filters, while the subsequent layers implement the conventional max-based reduction in neural-network form.

To improve robustness under realistic operating conditions, non-ideal perturbations are included in data generation. On the signal side, the waveform model incorporates $\pm 5\%$ perturbations in detector-related parameters together with temperature perturbations of $\pm 1.67\%$. On the noise side, the thermal-noise temperature is varied by $\pm 1.67\%$, and additional electronic voltage- and current-noise contributions are perturbed by $\pm 10\%$. In this way, the neural-network trigger is trained not only on ideal template-bank responses, but also on realistic signal and noise variations induced by detector non-idealities.

The training, validation, and test datasets are constructed from simulated waveform windows based on the same underlying signal model as in the template-bank stage, but with the non-ideal perturbations described above included during data generation. Their compositions are summarized as follows:

- The training set contains 10^6 samples, with positive and negative samples balanced at 50% each.
- The validation set contains 5×10^5 samples, with 1% positive samples and 99% negative samples.
- The test set contains 10^6 samples, also with 1% positive samples and 99% negative samples.

This dataset design allows stable network training while keeping the validation and test distributions closer to the highly imbalanced operating condition of the online trigger.

The positive and negative samples are defined as follows:

- Positive samples are labeled as $y = 1$ and are constructed from signal-plus-noise waveform windows. Among them, 50% are centered signal windows, while the remaining 50% are shifted signal windows with $0 < |\text{shift}| < 5$.
- Negative samples are labeled as $y = 0$ and consist of 50% pure-noise windows and 50% shifted signal-plus-noise windows with $|\text{shift}| > 20$.

This construction encourages the network not only to distinguish signal from background, but also to suppress false triggers caused by misaligned signal-like patterns.

The network is trained using a loss function tailored to the target operating point $\text{FPR} = 10^{-4}$. Let $\hat{y}$ denote the output logit and $y \in \{0, 1\}$ denote the label. The sigmoid function is

$$\sigma(\hat{y}) = \frac{1}{1 + \exp(-\hat{y})}, \tag{13}$$

and the overall loss is written as

$$\mathcal{L} = \lambda_{\text{BCE}} \mathcal{L}_{\text{BCE}} + \lambda_{\text{pos}} \mathcal{L}_{\text{pos}}, \tag{14}$$

where

$$\mathcal{L}_{\text{BCE}} = -\Big[y \log\big(\sigma(\hat{y})\big) + (1 - y) \log\big(1 - \sigma(\hat{y})\big) \Big] \tag{15}$$

is the standard binary cross-entropy term, and

$$\mathcal{L}_{\text{pos}} = \frac{1}{|P|} \sum_{i \in P} \text{softplus}\left(\frac{t_0(10^{-4}) - s_i^+}{\tau} \right) \tag{16}$$

is an additional positive-sample penalty term designed to reduce missed detections near the target operating point. Here, P denotes the set of positive samples, s_i^+ is the score of the ith positive sample, $t_0(10^{-4})$ denotes the reference threshold corresponding to the target operating point, and τ is a smoothing parameter. The softplus function is defined as

$$\text{softplus}(x) = \log(1 + e^x), \tag{17}$$

which provides a smooth approximation to $\max(0, x)$ and penalizes positive samples whose scores fall below the target threshold.

The network is optimized with the Adam optimizer [12]. Compared with the template-bank threshold trigger, the resulting neural-network trigger can learn decision boundaries that better exploit complementary template responses and are more tolerant to non-ideal waveforms, residual template mismatch, and practical distortions caused by detector non-idealities.

III. REAL-TIME FPGA IMPLEMENTATION AND HARDWARE VERIFICATION

This section presents the FPGA implementation of the proposed coarse trigger and its validation. The goal is to realize on hardware the neural-network-based trigger developed in Section II, while preserving the software behavior and satisfying the real-time throughput requirement of the online trigger. The evaluation focuses on functional correctness, timing closure, resource utilization, and overall engineering feasibility.

A. Deployment Objective and Overall Architecture

The target application is a streaming ADC readout scenario, in which the coarse trigger must process continuous waveform data in real time and retain only candidate events for subsequent offline analysis. The implementation goal is therefore twofold: first, the FPGA output should remain consistent with the software reference after quantization and architectural mapping; second, the parallel hardware implementation should achieve a steady-state update interval below 1 μs, as required by the ADC sampling rate.

The overall hardware architecture is illustrated in Fig. 5. The input sample stream is first written into a circular buffer RAM. A sliding-window module then reorganizes the buffered samples into fixed-length inference windows. Each window is fed into the FCNN inference block implementing the coarse-trigger network. The resulting trigger score is processed by the trigger-decision module to determine whether the current event should be retained. If the event is accepted, an event-extraction module retrieves the required waveform segment and writes the retained candidate event into an SRAM-based event buffer. In this architecture, the FPGA serves as the real-time front end for candidate-event selection.

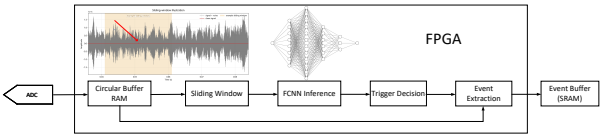

FIG. 5. Overall FPGA architecture for the real-time coarse trigger. The streaming waveform is buffered, reorganized into sliding windows, processed by the FCNN inference engine, and converted into candidate-event outputs for subsequent storage.

B. Implementation of the FCNN Inference

The dominant timing bottleneck of the hardware implementation lies in the first fully connected layer, whose output can be written as

$$y_j = \sum_{i=0}^{20000} x_i w_{j,i} + b_j, \tag{18}$$

where the input window consists of 20001 samples, each represented with 20-bit precision. A straightforward implementation would treat the entire window as a single vector and execute the long dot product serially for each neuron, resulting in an excessively long multiply-accumulate path and poor timing behavior.

To address this issue, a parallel implementation strategy is adopted, as illustrated in Fig. 6. First, the input data are distributed across 96 independent RAM banks so that 96 samples can be read out in parallel in each cycle. Second, each neuron in the first layer performs 96 multiplications in parallel per cycle, and the full first-layer dot product is completed in 209 cycles. Third, the 96 products generated in each cycle are reduced by a 7-stage 96-way adder tree to form one partial sum. Since the 209 partial sums are produced over different cycles rather than simultaneously, they are accumulated by a time-shared accumulator across 209 cycles to generate the final first-layer neuron output.

This architecture significantly shortens the critical path of the first-layer computation. On the one hand, the 96-bank memory organization avoids sequential scanning of the full input window. On the other hand, the combination of parallel multiplication, tree-based reduction, and cross-cycle accumulation prevents the formation of an excessively long serial arithmetic chain. As a result, the design becomes suitable for high-frequency clock operation while preserving the required inference functionality.

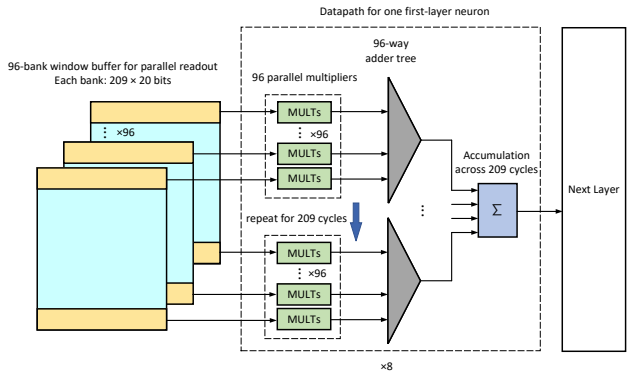

FIG. 6. Parallel datapath for one neuron in the first fully connected layer. A 20 001-sample input window is distributed across a 96-bank window buffer, where each bank stores 209×20 bits and supports parallel readout. At each cycle, 96 samples are processed by 96 parallel multipliers and reduced by a 96-way adder tree. The partial sums are then accumulated across 209 cycles to produce one first-layer neuron output, which is subsequently passed to the next layer.

C. Simulation and Post-Implementation Verification

The proposed FPGA coarse-trigger design is first verified through behavioral RTL simulation and post-implementation analysis. At the behavioral RTL level, the hardware outputs are compared window by window with the software reference, thereby validating the correctness of the quantized weights, the sliding-window organization, and the parallel scheduling strategy. The comparison confirms that the FPGA implementation reproduces the expected software behavior for all verified output windows.

The steady-state throughput is characterized by the interval between consecutive valid outputs. RTL simulation shows that the stable output interval is 236 clock cycles per window. Under a 250 MHz clock, corresponding to a clock period of 4 ns, the steady-state update interval is

$$T_{\mathrm{upd}} = 236 \times 4 \text{ ns} = 944 \text{ ns}, \tag{19}$$

which satisfies the design target of sub-microsecond steady-state throughput. This quantity refers to the steady-state output interval, i.e., throughput, rather than the total latency of a single window. The RTL results therefore demonstrate that the proposed architecture satisfies the required real-time processing capability.

Plots/fpga_functional_verification.pdf

FIG. 7. Behavioral RTL verification and throughput analysis. The FPGA outputs match the software reference on a window-by-window basis, and the steady-state output interval is measured as 236 cycles, corresponding to 944 ns at 250 MHz.

Post-implementation analysis further confirms the engineering feasibility of the design. Under the target

clock frequency of 250 MHz, the implemented design achieves timing closure with zero failing endpoints. The corresponding timing and resource results are summarized in Table I. These results indicate that the current implementation not only satisfies the required timing constraint, but also retains favorable scalability for future extension and system integration.

TABLE I. Post-implementation timing and resource results.

Metric / Resource	Value
Clock frequency	250 MHz
Worst negative slack (WNS)	0.024 ns
Total negative slack (TNS)	0
Failing endpoints	0
LUT utilization	2.76%
FF utilization	0.55%
BRAM tile utilization	4.29%
DSP utilization	14.58%

D. On-Board Validation

After bitstream generation, the design was programmed onto the FPGA board for on-board validation. In the board-level test, the deployed design produced valid trigger outputs through the hardware observation interface, and the observed output sequence was consistent with the expected digital behavior established by the software reference and RTL simulation. In particular, the output logit, hit flag, and valid signal were generated in the intended streaming order, confirming that the buffer organization, parallel inference datapath, and decision logic were correctly integrated in the real hardware environment.

Taken together, the simulation, post-implementation, and on-board results demonstrate that the proposed coarse trigger is not only functionally correct, but also practically deployable on FPGA hardware under the required real-time condition.

IV. CONCLUSION AND OUTLOOK

In this work, we studied the online coarse-trigger problem for magnetic monopole induction signals in a room-temperature induction-coil-based detection framework. Under the storage and throughput constraints of the SCEP experiment, the trigger objective is to preserve as many true signals as possible at a prescribed FPR. Starting from the conventional single-template optimum-filter-plus-threshold trigger, we developed a two-stage optimization strategy consisting of a greedy FNR-driven template bank and a physically informed neural-network trigger. The results show that the proposed template bank substantially outperforms both the original single-template trigger and the velocity-only scan at the target operating point FPR $= 10^{-4}$, while the neural-network trigger further improves robustness under non-ideal conditions. We further implemented the proposed trigger on FPGA and verified that the hardware output is consistent with the software reference, with timing closure achieved at 250 MHz and a steady-state update interval of 944 ns. Taken together, these results establish a practical online coarse-trigger solution for large-scale room-temperature induction-coil monopole searches.

Future work will focus on three aspects. First, the current study is still based mainly on modeled signal and noise variations, and measured detector backgrounds together with long-term operating drifts should be incorporated more explicitly. Second, the robustness of the neural-network trigger to broader classes of detector non-idealities and waveform distortions should be further evaluated. Third, further co-optimization of the trigger algorithm, quantization strategy, and FPGA architecture may improve both discrimination performance and implementation efficiency in future large-scale deployments.

ACKNOWLEDGEMENTS

This project is supported by grants from National Science Foundation of China (No. 12250011), and by the Frontier Scientific Research Program of Deep Space Exploration Laboratory under grant No. 2022-QYKYJH-HXYF-013.

[1] P. A. M. Dirac, Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character **133**, 60 (1931).
[2] H. Georgi and S. L. Glashow, Physical Review Letters **32**, 438 (1974).
[3] J. Preskill, Annual Review of Nuclear and Particle Science **34**, 461 (1984).
[4] P. Eberhard, R. Ross, and J. Taylor, Review of Scientific Instruments **46**, 362 (1975).
[5] B. Cabrera, Physical Review Letters **48**, 1378 (1982).
[6] M. A. et al. (The Macro Collaboration), The European Physical Journal C - Particles and Fields **25**, 511 (2002).
[7] R. Abbasi, Y. Abdou, M. Ackermann, J. Adams, J. Aguilar, M. Ahlers, D. Altmann, K. Andeen, J. Auffenberg, X. Bai, *et al.*, Physical Review D **87**, 022001 (2013).
[8] C. Ye, B. Liu, Z. Cao, L. Han, X. Huang, M. Jiang, D. Liu, Q. Lin, S. Wan, Y. Wu, L. Zhao, Y. Zhang,

X. Peng, and Z. Zhao (SCEP Collaboration), Phys. Rev. D **111**, 023020 (2025).

[9] B. P. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), Phys. Rev. Lett. **116**, 061102 (2016).

[10] S. A. Usman *et al.*, Classical and Quantum Gravity **33**, 215004 (2016).

[11] S. Somalwar, H. Frisch, and J. Incandela, Physical Review D **37**, 2403 (1988).

[12] D. P. Kingma and J. Ba, CoRR **abs/1412.6980** (2014).