

Response to the Editors

Original manuscript ID: TNS-01113-2026

Title: A Deep Neural Network Based Level-1 Trigger for the Magnetic Monopole Search in the SCEP Experiment

Decision: Major Revisions Required / Resubmission Encouraged, 02 October 2026

Decision Letter: Key Statements

Quoted verbatim from the decision e-mail (emphasis added).

Decision letter (Dr. Gian-Franco Dalla Betta)

*“However, **because we believe the work has merit, we invite you to consider submitting a revised manuscript** that takes all the reviewers' and Editors' comments into consideration. It would be given a new Manuscript ID and be fully reviewed again.”*

Senior Editor's note

*“TNS has changed how it treats Major Revision/Resubmit. Per IEEE's current policy, the system will consider your revision a new manuscript, and, therefore, there is no deadline for submission; ... Although your revision will be assigned a new manuscript ID, **it will be linked with all previous versions.**”*

Editor (Jinyuan Wu)

*“I would like to provide **a round of fast review so that you can clean up some quick issues before entering into the full review** that will take much longer time. Improving the clarity of the manuscript will help speeding up the full review in the future.”*

Senior Editor's comments

*“**On the cosmetic side** (see attached template): ...”*

In short, the manuscript was returned after a fast pre-review in which the Editor asked for clarifications before the full review, and the Senior Editor's comments concern formatting only. Under the current TNS policy for Major Revision/Resubmit, the revised manuscript receives a new manuscript ID and is linked with the original submission.

Dear Editors,

Thank you very much for evaluating our manuscript and for your comments and suggestions, which identified the explanations needed before the full review. We have revised the manuscript accordingly. Our point-by-point responses are listed below and marked in green. In the marked version of the revised manuscript, added text is shown in blue with a wavy underline and deleted text in red with a strike-through.

Because a new Fig. 1 (the equivalent circuit of the readout) has been added, the figures of the original manuscript are renumbered: Fig. n of the original manuscript is Fig. n+1 in the revised manuscript. Figure numbers below refer to the revised manuscript unless stated otherwise.

Responses to the Editor: Jinyuan Wu

Comment AE-1

“In Fig. 1, it seems that the noise has a strong sine wave component. Could you explain clearly in your manuscript what is this component? what's its frequency etc.”

Thank you very much for this question. The gray trace in the original Fig. 1 (now Fig. 2) is a simulated realization of the total readout noise. It is a stationary Gaussian random process, not a deterministic sinusoid; no power-line or other periodic interference is included in the model.

Its sine-like appearance is caused by the readout circuit. The induction coil, together with its own capacitance and the amplifier input and stray capacitances in parallel with it, forms an LCR resonant circuit with a resonance frequency of about 54.2 kHz. The dominant noise source, the thermal noise of the coil, acts in series with the coil and is therefore shaped by this resonance. The narrow resonance, with a quality factor of about 100, concentrates the noise power near 54.2 kHz, so the noise oscillates at about this frequency.

Changes in the manuscript

- Section II-A: the resonance of the readout circuit and its effect on the noise are described.
- Fig. 1 (new): equivalent circuit of the induction-coil readout. It shows that the induced electromotive force e_s and the coil thermal noise e_{th} act in series with the coil; all circuit symbols are defined in the caption.
- Fig. 2 (Fig. 1 in the original manuscript): panel (b) shows the signal and a noise realization, and the caption gives the trajectory and the displayed interval.

Comment AE-2

“The signal shown in Fig. 1 seems to be an endless oscillation. It is not obvious to the typical readers like myself. Could you explain clearly the mechanism of the signal shape? Will the signal decay? after how long?”

Thank you very much for this question. The monopole crossing induces a short electromotive-force transient in the coil. This transient excites the resonance of the readout circuit, so the output rings at the resonance frequency of about 54.2 kHz after the crossing. The ringing is a damped oscillation: the coil resistance dissipates the stored energy, and the amplitude decays approximately exponentially with a time constant of about 0.61 ms. The amplitude falls below 1% of its initial value about 2.8 ms after the crossing. The original Fig. 1 showed only 300 μ s after the crossing, about half of one decay time, so the decay was hardly visible.

Changes in the manuscript

- Section II-A: the origin of the oscillation and its decay (below 1% of the initial amplitude about 2.8 ms after the crossing) are described.
- Fig. 2 (Fig. 1 in the original manuscript) has been redrawn with two panels: (a) the signal over 4.3 ms, which shows the complete decay, and (b) the original 0–600 μ s view of the signal and noise.

Comment AE-3

“It is not clear how the multipliers(DSP48) are organized for computing Eq. (9). For each new sample, you need to make $31 \times 113 = 3503$ multiplications. Your system runs at 200 MHz. Do you use 113 multipliers each calculating 31 times or, 31 multipliers each calculating 113 times, or you have other arrangement? Fig. 13 gives readers an impression that 113 templates are implemented in parallel,

which would need 3503 multipliers but Table I shows you only used 72 DSP48E1. Please explain clearly in manuscript.”

Thank you very much for this comment, and we apologize for the confusion caused by the original description and figure. The originally submitted implementation used a third arrangement. The 113 templates were divided into five groups of 24, 24, 24, 24, and 17 templates, which were evaluated one group after another by 24 time-multiplexed filter lanes: within a group, each lane evaluated one template, and the 31 window samples were broadcast to all lanes over 31 clock cycles. The filter therefore needed $5 \times 31 = 155$ clock cycles per input sample, and seven lanes were idle in the last group. The MLP was evaluated by 16 multiply-accumulate (MAC) lanes, so the design contained 40 logical multipliers. In the synthesized design, however, the 24 filter lanes occupied 48 DSP48E1 slices and the 16 MAC lanes occupied 24, i.e., 72 in total, because the synthesis tool duplicated the multipliers whose products were also used outside the accumulation path in the original RTL. These additional slices were not required by the arithmetic. Neither the schedule nor the origin of the 72 slices was described in the original manuscript. Moreover, Fig. 13 of the original manuscript (now Fig. 14) drew the 113 templates as stacked copies of a 31-multiplier filter to illustrate the computation rather than the hardware, which gave the impression of 3503 parallel multipliers.

This grouped schedule is not intuitive and cannot be read directly from (9). To address this comment, we have therefore re-implemented the template-bank filter with the second arrangement mentioned by the Editor, which follows (9) directly: 31 multipliers, each performing 113 multiplications per input sample. The 31-sample window is held in registers. In each clock cycle, the 31 coefficients of one template are read from the template ROM and multiplied by the window samples in the 31 parallel multipliers, and a five-level pipelined adder tree sums the products. The template bank is therefore evaluated one template per clock cycle. Including the delay of the adder tree and the output processing, the filter stage has a latency of 127 clock cycles, which is shorter than the 200-cycle sample interval (1 MHz input, 200 MHz logic).

The filter and the MLP form a two-stage pipeline connected by ping-pong feature buffers, so that the MLP processes the responses of one sample while the filter processes the next. The trigger accepts one sample every 200 clock cycles with a latency of 289 cycles (1.445 μ s). Each of the 31 filter multipliers and each of the 16 MLP MAC lanes (see AE-5) occupies exactly one DSP48E1 slice, because the 20-bit data and 18-bit coefficients fit the 25×18 -bit multiplier of a slice, and no multiplier is duplicated. The adder tree is implemented in general logic, and the trigger core therefore uses $31 + 16 = 47$ DSP48E1 slices, as listed in Table I.

We repeated the RTL simulations and the complete on-board validation with the new implementation. On the board, all samples of the threshold-calibration set D₋ (2.01×10^8 noise-only samples) and of the evaluation set D₊ (3.2×10^6 signal-plus-noise records with 1.6×10^9 samples) were processed without any dropped sample, and the returned outputs agree exactly with the bit-accurate fixed-point reference. The fixed-point arithmetic is unchanged, so the net acceptance is the same as in the original manuscript.

Changes in the manuscript

- Section III-B has been rewritten to give the multiplication counts, the two pipeline stages and their latencies, and the DSP48E1 allocation.
- Fig. 14 (Fig. 13 in the original manuscript) has been redrawn to show a single set of 31 multipliers reused for the 113 templates, the ping-pong buffer, and the 16 shared MAC lanes. Fig. 13 (overall FPGA architecture) has also been redrawn for the new implementation.
- Table I has been updated for the new implementation: 9,458 LUTs, 12,770 registers, 14 BRAM tiles, and 47 DSP48E1 slices.
- Section III-C has been updated with the on-board results of the new implementation.

Comment AE-4

“Please expand the explanation about the MLP training process. Describe number of bits of the coefficients.”

Thank you very much for this suggestion.

Training process. The MLP takes the absolute responses of the 113 templates as its input. Only the weights and biases of the MLP are trained, whereas the template bank is kept fixed. The training set contains 2×10^6 positive and 4×10^6 negative waveform windows generated with the signal and noise models of Section II-A. A signal-injected window is labeled positive when its center lies within $3 \mu\text{s}$ of the true crossing time; pure-noise windows and signal-injected windows whose centers are more than $10 \mu\text{s}$ from the crossing are labeled negative, and the windows in between are excluded. An independent validation set of 4×10^5 positive and 8×10^5 negative windows, with the same signal distribution and labeling rules, is used to monitor overfitting and to select the checkpoint. The network is trained in 32-bit floating point by minimizing the binary cross-entropy with the AdamW optimizer, using an initial learning rate of 10^{-3} , a weight decay of 10^{-4} , and a batch size of 1920 windows. The learning rate is halved whenever the validation loss has not improved for three epochs, the training runs for 24 epochs, and the checkpoint with the lowest validation loss is used.

Coefficient bit widths. For the FPGA implementation, the trained floating-point parameters are converted to fixed point. The template coefficients and the MLP weights are signed 18-bit integers, which match the 18-bit multiplier input of a DSP48E1 slice. The word lengths of the other quantities along the datapath, from the 20-bit input samples to the 32-bit output logit, are given in Section III-B. With these word lengths, the net acceptance of the fixed-point implementation is lower than that of the floating-point reference by only $3/(3.2 \times 10^6)$ in absolute terms, and only 8 of the 3.2×10^6 signal-plus-noise records are classified differently (Section III-C).

Changes in the manuscript

- Section II-E: the training paragraph has been expanded with the trained parameters, the data sets and labeling rules, the optimizer settings, and the training length.
- Section III-B: the fixed-point formats are given together with the template-bank filter in (9) and with the MLP, whose neuron computation, including the bias, is now written as (10).

Comment AE-5

“Write down how many multiplications needed per sample in MLP (I guess it is $113 \times 16 + 16 \times 8 + 8$?) Please describe physical arrangement of the multipliers (DSP48).”

Thank you very much for this suggestion. Yes, the Editor's count is correct. The 113-16-8-1 MLP requires $113 \times 16 + 16 \times 8 + 8 = 1944$ multiplications per input sample. This count is now given in Section III-B.

The per-layer multiplication schedule of the MLP is unchanged from the original implementation, but the multiplier duplication described in AE-3 has been removed. In the revised implementation, the MLP is evaluated by 16 multiply-accumulate (MAC) lanes, each occupying one DSP48E1 slice (24 slices in total in the original implementation), which are reused across the three fully connected layers:

- FC1: each lane computes one of the 16 hidden neurons while one of the 113 input features is broadcast per cycle (113 cycles, 1808 multiplications);
- FC2: all 16 lanes are used, two for each of the 8 neurons, and each lane computes 8 of the 16 products of its neuron (8 cycles, 128 multiplications);
- FC3: eight lanes compute the 8 products in one clock cycle (8 multiplications), and an adder tree then sums them.

Including the memory reads, bias addition, rounding, and the ReLU activations, one MLP evaluation takes 161 clock cycles, which is shorter than the 200-cycle sample interval. The MLP therefore uses 16 of the 47 DSP48E1 slices of the trigger core.

Changes in the manuscript

- Section III-B: the per-sample multiplication count, the 16 shared MAC lanes of the second pipeline stage, and the number of clock cycles taken by each FC layer have been added.
- Fig. 14: the MLP is shown as three layers evaluated sequentially on 16 shared MAC lanes.

Responses to the Senior Editor

Comment SE-1

“please move the Acknowledgement to the funding agency to the footnote at page 1;”

Thank you very much. Done. The funding statement has been moved to the first-page footnote, and the Acknowledgment section has been removed.

Comment SE-2

“in the figures, please avoid having grids in the plot area; also, please enclose the plot area with axes + tick marks along all sides.”

Thank you very much. Done. All data plots (Figs. 2–9, 11, 12, 15, and 16) have been redrawn without grid lines, and each plot area is now enclosed by axes with tick marks on all four sides.

Other Changes

In addition to the changes requested above, we have made the following revisions.

- FPGA implementation (Section III and Conclusions). The FPGA implementation was updated as described in the response to AE-3, and all on-board tests were repeated; Figs. 13 and 14, Table I, and the Conclusions were updated accordingly.
- Reference signal level (Section I). The SCEP proposal requires a processed-signal power SNR of at least 4.5 for a monopole crossing the coil center perpendicularly at 10^{-5} of the speed of light; the manuscript now states this condition and adopts this value as the reference signal level.
- Evaluation (Sections II-A to II-D). The trajectories are sampled with a uniform flux through a sphere of radius 10 cm that encloses the coil, so P_{net} includes the geometric acceptance of the coil; because many trajectories pass outside the coil or cross only part of its turns, the values of P_{net} are relatively small. Definitions of the compact and conventional template-bank triggers, the paired noise-only records, and the mean P_{net} have been added, and the 95% confidence-interval half-width of each mean P_{net} value, about 0.05 pp, is given.
- Window-length scan (Section II-E). Three networks trained with different random seeds are evaluated for each length; their P_{net} values agree within about 0.01 pp, and Fig. 12 now shows their average.
- Notation and format. The notation was made consistent ($\beta = v/c$, L1, FPR, power SNR, and e_s for the induced electromotive force), abbreviations were removed from the abstract, equations and figures are cited as “(n)” and “Fig. n”, subfigure labels were added, Fig. 8 is drawn as a histogram, and the wording was improved in several places.
- References. Some references were added or removed, and the formatting of the reference list was corrected.

Thank you again for your helpful comments, which have improved the clarity and reproducibility of the manuscript. We look forward to your further consideration.

Sincerely,

The Authors