Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Prediction of Magnetic Flux Evolution During Solar Active Region Emergence using Long Short-Term Memory Networks

Eren Dogan

ebd@njit.edu

New Jersey Institute of Technology

Spiridon Kasapis

Princeton University

Sarang Patil

New Jersey Institute of Technology

Jonas Tirona

New Jersey Institute of Technology

John Stefan

New Jersey Institute of Technology

Irina Kitiashvili

Ames Research Center

Mengjia Xu

New Jersey Institute of Technology

Alexander Kosovichev

New Jersey Institute of Technology

Research Article

Keywords: Solar Active Regions, Space Weather Forecasting, Magnetic Flux Emergence, Acoustic Power, Continuum Intensity, Quiet-Sun, LSTM Encoder- Decoder, MagFluxLSTM, Hyperparameter Optimization, Teacher Forcing

Posted Date: April 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9315643/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Solar Physics
DOI: 10.1007/•••••-••••-•••-•••••-•

Prediction of Magnetic Flux Evolution During Solar Active Region Emergence using Long Short-Term Memory Networks

Eren Dogan[1] · Spiridon Kasapis[2] · Sarang Patil[3] · Jonas Tirona[1] · John Stefan[4] · Irina Kitiashvili[5] · Mengjia Xu[3] · Alexander Kosovichev[5,4]

© The author(s) • • • •

Abstract Solar active regions (ARs) are the primary drivers of space weather events, making their early prediction crucial for operational forecasting systems. We develop machine learning models capable of predicting the evolution of magnetic flux during AR emergence using 1D time series of the continuum intensity and solar oscillation power maps for 53 active regions and their surrounding quiet-Sun areas. Each observable is sampled over a fixed $30.66° \times 30.66°$ field of view. These observations capture the temporal evolution of each active region and serve as inputs for training and validation of our `MagFluxLSTM` and `MagFluxEnc-Dec` models. The `MagFluxLSTM` architecture implements a single-stage standard Long-Short Term Memory (LSTM) network. `MagFluxEnc-Dec` represents an LSTM encoder-decoder with teacher forcing. To test and evaluate the models' performance, we use the continuum intensity and oscillation power maps (calculated for several frequency bands from Doppler velocity) as input to predict the magnetic flux. Among the top 100 hyperparameter configurations ranked by validation derivative RMSE, 98% correspond to `MagFluxLSTM`, compared to only 2% for `MagFluxEnc-Dec`. Thus, although the `MagFluxEnc-Dec` architecture has higher model complexity, it leads to poorer generalization to ARs outside the training

✉ E. Dogan
ebd@njit.edu

[1] Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA

[2] Department of Astrophysical Sciences, Princeton University, Princeton, NJ, USA

[3] Department of Data Science, New Jersey Institute of Technology, Newark, NJ, USA

[5] NASA Advanced Supercomputing Division, NASA Ames Research Center, Moffett Field, CA, USA

[4] Department of Physics, New Jersey Institute of Technology, Newark, NJ, USA

set and less stable training than the simpler `MagFluxLSTM`, which can predict magnetic flux emergence 3–10 hours in advance within a 12-hour prediction window in both experimental and operational-type settings for the 5 testing active regions.

Keywords: Solar Active Regions, Space Weather Forecasting, Magnetic Flux Emergence, Acoustic Power, Continuum Intensity, Quiet-Sun, LSTM Encoder-Decoder, MagFluxLSTM, Hyperparameter Optimization, Teacher Forcing

1. Introduction

Solar active regions (ARs) are areas of intense and complex magnetic field concentration on the solar photosphere and are the primary source of space weather phenomena, including solar flares, coronal mass ejections, and solar energetic particle events (e.g., Schrijver 2009). Therefore, early detection and prediction of AR emergence are critical for space weather forecasting, as they can potentially enhance warning capabilities for geomagnetic disturbances that affect Earth's magnetosphere and impact technological infrastructure. Understanding subsurface dynamics and changes in turbulent motions that cause the stochastic excitation of solar oscillations at the photosphere, the power spectrum of which can be sensitive to subsurface perturbations caused by emerging AR (Hartlep et al. 2011; Singh, Raichur, and Brandenburg 2016; Waidele et al. 2023; Kasapis et al. 2023). Such pre-emergence signatures can also be detected through Hα imaging (Liggett and Zirin 1985), surface flow analysis (Grigor'ev, Ermakova, and Khlystova 2011; Schunker et al. 2024), and helioseismic travel-time variations (Kosovichev, Duvall, and Scherrer 1999; Kosovichev 2011; Ilonidis, Zhao, and Kosovichev 2011; Stefan and Kosovichev 2023). A number of helioseismic studies have confirmed that variations of photospheric disturbances could signal the upcoming emergence of magnetic flux before it can be identified visually (e.g., Ilonidis, Zhao, and Kosovichev 2011; Ilonidis, Zhao, and Hartlep 2013; Birch et al. 2012; Leka et al. 2013; Barnes et al. 2014; Stefan and Kosovichev 2023).

Machine learning (ML) models for forecasting space weather events have recently emerged as a powerful approach for analyzing the growing volume of heliophysics data (Camporeale 2020). These methods have been successfully applied to a wide range of predictive tasks, including the prediction of solar flare onset (e.g, Jiao et al. 2020; Georgoulis et al. 2021), solar energetic particles (SEPs; e.g., Kasapis et al. 2026a; Hutchins et al. 2026), and coronal mass ejections (CMEs; e.g, Bobra and Ilonidis 2016; Hegde 2025). One of the most recent advances in this domain is the SuryaFM model (Roy et al. 2025a,b), a 366-million-parameter foundation model for heliophysics designed to learn general-purpose solar representations from multi-instrument SDO observations. In this light, deep learning architectures designed for sequential data offer promising avenues for capturing the temporal dependencies inherent in the emergence of active regions. Long Short-Term Memory (LSTM) networks–an advanced architecture of recurrent neural networks (RNNs)–have demonstrated good performance in time-series prediction tasks across diverse domains (Hochreiter and Schmidhuber

1997; Hua et al. 2019; Sherstinsky 2020), making them particularly suitable for modeling the complex temporal evolution of solar magnetic fields.

Kasapis et al. (2023, 2025) employed LSTM architectures to predict the decrease in continuum intensity associated with the emergence of ARs before they were observed on the solar surface. The LSTM models were trained on Solar Dynamics Observatory/Helioseismic and Magnetic Imager (SDO/HMI; Pesnell, Thompson, and Chamberlin 2012; Scherrer et al. 2012a) observations processed into the 1D ML-ready dataset, the Solar Active Region Emergence Dataset (SolARED; Kasapis et al. 2026b). The LSTM models were designed to predict the onset of continuum intensity decrease associated with AR emergence within a 12-hour window. The results demonstrate that the LSTM-based analysis of oscillation power maps can successfully forecast AR emergence 5–29 hours before the associated continuum intensity decrease, at a stage when the magnetic flux has reached only 4%–9.6% of its eventual maximum.

This study aims to test whether ML models can capture disturbances in solar acoustic power and continuum intensity to predict the magnetic flux emergence before it is visible on the solar photosphere. The presented deep learning framework for predicting the evolution of magnetic flux builds on the methodology of Kasapis et al. (2025) and addresses three key challenges: a) capturing long-term temporal dependencies and pre-emergence signatures in acoustic power and continuum intensity observations, b) demonstrating the ability to predict magnetic flux given (a), and c) achieving operational magnetic flux forecast capabilities with sufficient lead time for space weather applications. The paper is organized as follows: Section 2 presents the methodology followed, such as the data preprocessing steps, the LSTM architectures used, the loss optimization, and the LSTM training procedures. Section 3 describes the prediction results on five testing ARs, including quantitative performance metrics and qualitative analysis of prediction capabilities. Finally, Section 4 provides an overview of the key findings, discusses the models' limitations, and offers some concluding remarks.

2. Data and Methodology

We provide detailed information about the AR emergence dataset (Section 2.1) the different LSTM architectures (Section 2.2), and the model training (Section 2.3) and implementation details (Section 2.4) in the following subsections.

2.1. Dataset

In this study, we train and test the developed ML models on the compressed ML-ready data sets derived from preprocessed solar surface observations of 53 ARs, observed from March 2010 to June 2023 using the SDO/HMI (Scherrer et al. 2012b; Hoeksema et al. 2014) instrument. The dataset's 1D time-series represent the evolution of spatially averaged in 9×9 bins: a) continuum intensity I_c, b) unsigned magnetic flux Φ_m, and c) $M = 4$ frequency ranges of acoustic power P_a (i.e., 2–3 mHz, 3–4 mHz, 4–5 mHz, and 5–6 mHz), all at 1-hour cadence, covering

the full emergence and evolution lifecycle of each AR. The acoustic power maps are derived from an 8-hour series of Dopplergrams with one-hour shifts, capturing the temporal evolution of subsurface dynamics throughout each AR's lifecycle. A detailed description of the Solar Active Region Emergence Dataset (SolARED) is presented in Kasapis et al. (2026b). The dataset is available through the SolARED interactive website at https://sun.njit.edu/sarportal/. The SolARED dataset utilized in this study is already pre-normalized to account for the solar geometric effect.

In the original dataset, the evolution of these observables was tracked and remapped onto the heliographic coordinates using Postel's projection centered on the center of the $30.66^\circ \times 30.66^\circ$ regions to correct for geometrical distortions. The tracked and remapped series of Dopplergrams, sampled at a 45 s cadence, are used to calculate oscillation power maps, which are then divided into 81 tiles arranged in a 9×9 grid. To focus on the primary AR emergence area and balance the active and quiet Sun tiles, the central row of the 9×9 grid is isolated, and the top and bottom four rows are excluded from this analysis. This results in a spatial selection of only 9 horizontal tiles as seen in Figure 1, similar to the tile selection done by Kasapis et al. (2025) and by Tirona et al. (2026). The continuum intensity I_c and acoustic power P_a timelines are stacked together to produce the final input tensor for each AR with dimensions $T \times N \times (M+1)$, where $T = 9$ represents the central row of tiles, $N = 240$ corresponds to the temporal extent (240 hours), and $M+1 = 5$ denotes the acoustic power frequency bands plus one additional channel for continuum intensity. Each of the 9 tiles is treated as an independent spatial sample within the batch dimension during training. An overview of this data accumulation process can be seen in Figures 1 and 2.

The training data comprises of 41 ARs (77%), the validation data includes 7 ARs (13%), and the testing group utilizes the same 5 ARs (10%) evaluated in Kasapis et al. (2025) and Tirona et al. (2026). This split ensures that the model is evaluated on ARs outside the training set, providing a realistic assessment of generalization capability. During training, $W = 110$ hours of observations are used as inputs to the model, with the next $P = 12$ hours of observations reserved for use as targets. This 122-hour (W+P hours) window slides through the 240-hour temporal extent 118 times, generating distinct input–target pairs that are used to train the ML models in a supervised manner. For testing, the input window lengths were adjusted for each test AR to make sure that the very first input intervals capture only pre-emergence phases. While all ARs are tested using 96-hour input timelines, AR11726 is limited to 72 hours due to its emergence closer to the Eastern limb. The SolARED dataset utilized in this study is normalized to account for the solar geometric effect, with all results denormalized for presentation purposes. In addition, min-max scaling is applied independently to each input channel to avoid imposing artificial relationships between physically distinct quantities, while preventing data leakage by fitting the scaler exclusively on the training set.

2.2. Model Architecture

We investigated two LSTM-based model architectures in this study: a) an encoder–decoder LSTM (`MagFluxEnc-Dec`; Figure 1) that employs autoregressive decoding with teacher forcing and b) a stacked LSTM (`MagFluxLSTM`; Figure 2) that performs one-shot, multi-step predictions. Both models take temporal sequences of magnetic and acoustic features as input and generate P-timesteps–ahead forecasts of magnetic flux evolution. As mentioned above, the output length was selected to be $P = 12$ hours, but predictions with shorter or longer output windows are also possible.

2.2.1. `MagFluxEnc-Dec`: Encoder-Decoder LSTM

The `MagFluxEnc-Dec` model implements an encoder-decoder LSTM architecture, which is considered the state-of-the-art approach for sequence-to-sequence prediction tasks with variable-length temporal dependencies. The architecture consists of two symmetric components: a multi-layer LSTM encoder that compresses the input window into a fixed-dimensional context state, and a recurrent LSTM decoder that autoregressively generates the future sequence using this context (see Figure 1).

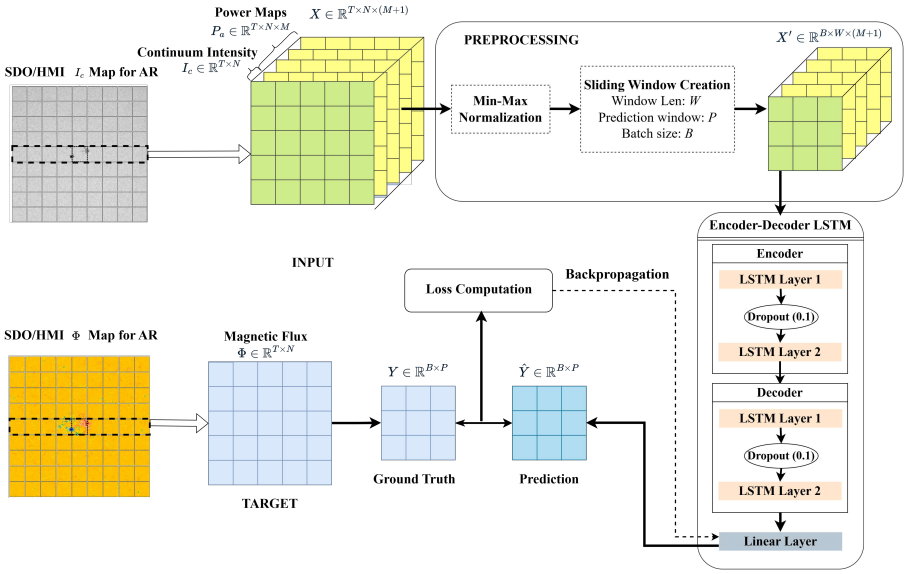


Figure 1. The `MagFluxEnc-Dec` pipeline for magnetic flux prediction. The system extracts central tiles from SDO/HMI continuum intensity (I_c) and magnetic flux (Φ_m) maps, combines them with acoustic power maps to form feature tensor $X \in R^{T\times N\times(M+1)}$ where $T = 9$ central tiles, $N = 240$ timesteps, and $M = 4$ power map frequencies. Sliding windows are generated by continuously sliding a window of length $W = 110$ timesteps across the temporal dimension with unit stride, extracting the following $P = 12$ timesteps as prediction targets, yielding a batch tensor $X' \in R^{B\times W\times(M+1)}$. A two-layer encoder-decoder LSTM is employed to predict magnetic flux evolution $\hat{Y} \in R^{B\times P}$.

The encoder processes the input sequence $\mathbf{x}_{1:W}$ of length $W = 110$ timesteps via a multi-layer LSTM with L stacked layers and h hidden units per layer. The number of layers L is treated as a dynamic hyperparameter optimized during the training search process. At each encoder timestep $t = 1, \ldots, W$, the recurrence relations are:

$$\begin{aligned} \mathbf{H}_t^{(1)}, \mathbf{c}_t^{(1)} &= \mathrm{LSTM}_{\mathrm{enc}}^{(1)}\left(\mathbf{x}_t,\ \mathbf{H}_{t-1}^{(1)},\ \mathbf{c}_{t-1}^{(1)}\right), \\ &\vdots \\ \mathbf{H}_t^{(l)}, \mathbf{c}_t^{(l)} &= \mathrm{LSTM}_{\mathrm{enc}}^{(l)}\left(\mathbf{H}_t^{(l-1)},\ \mathbf{H}_{t-1}^{(l)},\ \mathbf{c}_{t-1}^{(l)}\right), \end{aligned} \tag{1}$$

where $\mathbf{x}_t$ is the input feature vector, and $\mathbf{H}_t^{(l)}$ and $\mathbf{c}_t^{(l)}$ are the hidden state and cell state of the encoder layer $l \in \{1, \ldots, L\}$ at timestep t. The final encoder hidden and cell states summarize the entire window history and are transferred to initialize the decoder. The decoder adopts the same multi-layer LSTM architecture as the encoder, and autoregressively predicts the output sequence $\hat{y}_{1:P}$ over $P = 12$ future timesteps. For each prediction timestep $j = 1, \ldots, P$, the decoder transitions according to:

$$\begin{aligned} \mathbf{H}_j^{(1)}, \mathbf{c}_j^{(1)} &= \mathrm{LSTM}_{\mathrm{dec}}^{(1)}\left(z_j,\ \mathbf{H}_{j-1}^{(1)},\ \mathbf{c}_{j-1}^{(1)}\right), \\ &\vdots \\ \mathbf{H}_j^{(l)}, \mathbf{c}_j^{(l)} &= \mathrm{LSTM}_{\mathrm{dec}}^{(l)}\left(\mathbf{H}_j^{(l-1)},\ \mathbf{H}_{j-1}^{(l)},\ \mathbf{c}_{j-1}^{(l)}\right), \quad l = 2, \ldots, L, \\ \hat{y}_j &= \mathbf{W}_{\mathrm{out}}\mathbf{H}_j^{(L)} + b_{\mathrm{out}}, \end{aligned} \tag{2}$$

where z_j is the input to the decoder at prediction timestep j, determined by the teacher forcing mechanism. $\mathbf{H}_j^{(l)}$ and $\mathbf{c}_j^{(l)}$ are the hidden and cell states of decoder layer l at timestep j. The hidden state of the final decoder layer L, $\mathbf{H}_j^{(L)}$, is projected via the weight matrix $\mathbf{W}_{\mathrm{out}}$ and bias b_{out} to produce $\hat{y}_j$, the predicted magnetic flux value at timestep j. Specifically, z_j is determined by the *teacher forcing* mechanism: during training, with probability P_{tf} (the teacher forcing ratio), z_j is set to the ground-truth target value y_{j-1}; otherwise, z_j is set to the model's own previous prediction $\hat{y}_{j-1}$. This stochastic switching between ground-truth and predicted inputs stabilizes training by mitigating exposure bias and improving convergence. The encoder-decoder framework explicitly models how prediction errors can propagate through the iterative prediction window, making it well-suited for capturing non-stationary, autoregressive temporal dynamics such as those present in solar magnetic flux emergence.

2.2.2. `MagFluxLSTM`: Stacked LSTM

The `MagFluxLSTM` model implements a multi-layer stacked LSTM that processes the entire input window in a single forward pass and produces all P-step future predictions via a single linear output layer. Unlike the encoder-decoder approach,

this architecture does not employ autoregressive feedback; instead, all future timesteps are jointly decoded from the final hidden state of the stacked layers (Figure 2).

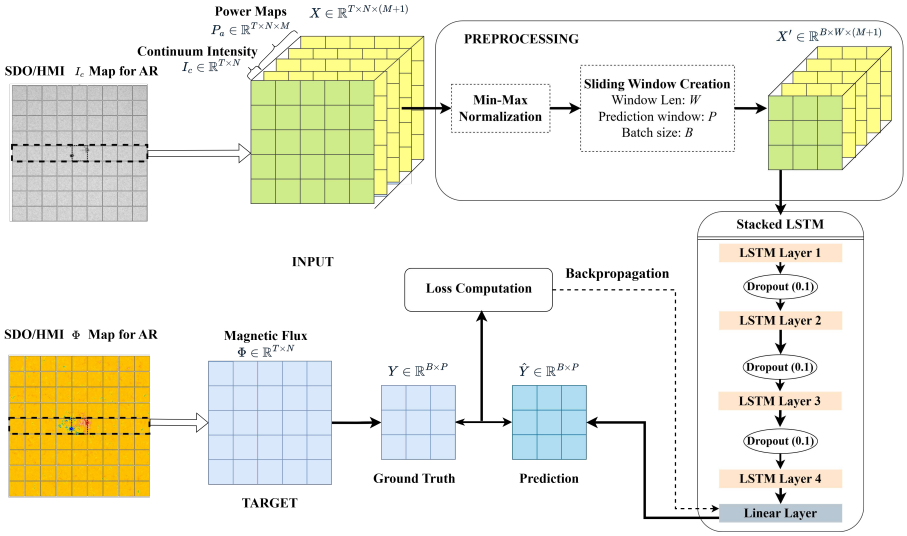


Figure 2. Schematic overview of the `MagFluxLSTM` pipeline for magnetic flux prediction during active region emergence. `MagFluxLSTM` employs stacked LSTM structure instead of the encoder-decoder structure in Figure 1. The system extracts central tiles from SDO/HMI continuum intensity (I_c) and magnetic flux (Φ_m) maps, combines them with oscillation power maps to form feature tensor $X \in R^{T\times N\times(M+1)}$ where $T = 9$ central tiles, $N = 240$ timesteps, and $M = 4$ power map frequencies. Sliding windows are generated by continuously sliding a window of length $W = 110$ timesteps across the temporal dimension with unit stride, extracting the following $P = 12$ timesteps as prediction targets. During training, windowed samples are grouped into batches of size B, yielding a batch tensor $X' \in R^{B\times W\times(M+1)}$. A four-layered stacked LSTM is employed to predict magnetic flux evolution $\hat{Y} \in R^{B\times P}$.

Given an input sequence $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_W)$ of length $W = 110$, the state of the stacked LSTM, with L layers (where L is optimized via the Ray Tune (Liaw et al. 2018)) and hidden size h, evolves as follows. At each timestep $t = 1, \ldots, W$:

$$\begin{aligned} \mathbf{H}_t^{(1)}, \mathbf{c}_t^{(1)} &= \mathrm{LSTM}^{(1)}\left(\mathbf{x}_t,\ \mathbf{H}_{t-1}^{(1)},\ \mathbf{c}_{t-1}^{(1)}\right), \\ &\vdots \\ \mathbf{H}_t^{(l)}, \mathbf{c}_t^{(l)} &= \mathrm{LSTM}^{(l)}\left(\mathbf{H}_t^{(l-1)},\ \mathbf{H}_{t-1}^{(l)},\ \mathbf{c}_{t-1}^{(l)}\right), \quad l = 2, \ldots, L, \end{aligned} \tag{3}$$

where, $\mathbf{x}_t$ is the input feature vector at timestep t.

Between consecutive LSTM layers (i.e., the output of layer l and the input to layer $l + 1$), dropout with rate P_d is applied to improve regularization and reduce overfitting. After processing the entire window, the final hidden state of the top layer $\mathbf{H}_W^{(L)}$ encodes a comprehensive summary of the historical temporal

dynamics. This summary is then projected directly to the prediction horizon via:

$$\hat{\mathbf{y}} = \mathbf{W}_{\text{out}}\mathbf{H}_W^{(L)} + \mathbf{b}_{\text{out}} \in \mathbb{R}^P, \tag{4}$$

where $\mathbf{W}_{\text{out}} \in \mathbb{R}^{P\times h}$ and $\mathbf{b}_{\text{out}} \in \mathbb{R}^P$ are trainable parameters, and $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_P)$ is the complete prediction vector for the next $P = 12$ hours. Critically, this model is *non-autoregressive*: the prediction for timestep j does not depend on the model's prediction at timestep $j-1$, nor is there teacher forcing. Instead, all P predictions are determined simultaneously and independently by the single context vector $\mathbf{H}_W^{(L)}$. This simplicity offers computational efficiency and can outperform autoregressive models when the prediction horizon is relatively short ($P = 12$ hours), and temporal dependencies are not strongly autoregressive in nature.

2.3. Training and Loss Functions

To define magnetic flux emergence, we adopt an emergence criterion similar to that of Kasapis et al. (2025), where emergence is identified when the time derivative of the Φ_m unsigned magnetic flux exceeds a fixed θ value threshold for a sustained duration of t_{sus}:

$$\frac{d\Phi_m}{dt} > 0.01 \quad \text{for} \quad t_{sus} = 4 \quad \text{hours}. \tag{5}$$

This definition is used to evaluate the timing of AR emergence in the LSTM predictions. Because the AR emergence is characterized by changes in temporal gradients rather than absolute magnitudes, a hybrid loss function is designed that explicitly incorporates derivative information to be used alongside the typical mean-squared error (MSE) loss. While magnitude loss functions are effective for predicting the true value for most purposes, the goal of this study is to predict magnetic flux growth using a time derivative. To address this goal, a hybrid loss function that simultaneously optimizes for flux magnitude accuracy and temporal derivative prediction is introduced:

$$\begin{aligned} \mathcal{L}_{\text{hybrid}} &= \alpha\,\mathcal{L}_{\text{mse}} + (1-\alpha)\,\mathcal{L}_{\text{derivative}}, \quad \alpha \in (0,1]. \\ \text{where} \quad \mathcal{L}_{\text{mse}} &= \frac{1}{BP}\sum_{b=1}^{B}\sum_{t=1}^{P}\big(\hat{y}_{b,t} - y_{b,t}\big)^2, \\ \mathcal{L}_{\text{derivative}} &= \frac{1}{B(P-1)}\sum_{b=1}^{B}\sum_{t=2}^{P}\big[(\Delta\hat{y}_{b,t}) - (\Delta y_{b,t})\big]^2, \end{aligned} \tag{6}$$

$$\begin{aligned} \Delta\hat{y}_{b,t} &= \hat{y}_{b,t} - \hat{y}_{b,t-1} \quad (t = 2, \ldots, P), \\ \Delta y_{b,t} &= y_{b,t} - y_{b,t-1} \quad (t = 2, \ldots, P), \end{aligned} \tag{7}$$

The training objective in Equation 6 combines mean-squared error (MSE) terms defined in Equation 7. The first term, $\mathcal{L}_{\text{mse}}$, penalizes absolute discrepancies between predicted magnetic flux $\hat{y}$ and the target y across the prediction horizon. The second term, $\mathcal{L}_{\text{derivative}}$, applies the same penalty to the first-order finite differences ($\Delta\hat{y}$ and Δy), thereby encouraging agreement in temporal gradients and improving sensitivity to the onset and timing of flux emergence. The normalization factors ($1/BP$ and $1/B(P-1)$) ensure that both components are averaged over their respective time indices, matching the common `reduction='mean'` behavior used in practice. The mixing coefficient $\alpha \in (0, 1]$ controls the trade off between amplitude fidelity ($\mathcal{L}_{\text{value}}$) and temporal-dynamics fidelity ($\mathcal{L}_{\text{derivative}}$), with smaller α emphasizing timing accuracy and larger α emphasizing magnitude accuracy.

Because the input data are sampled at a fixed cadence of $\Delta t = 1$ hour, the finite difference terms directly approximate temporal derivatives; therefore, any constant rescaling of Δt can be absorbed into α without affecting optimization. In practice, an overall constant multiplier may also be applied to $\mathcal{L}_{\text{hybrid}}$ to stabilize gradient magnitudes during training. The training workflow for the LSTM models follows a standard temporal learning pipeline with sliding-window sampling and backpropagation through time as summarized in Algorithm 1. After completing all training forward passes, the model is evaluated on the validation set during inference, and the validation loss — computed as the RMSE of the derivative — is used to update the learning-rate scheduler. This process is repeated across all epochs.

2.4. Implementation Details

The complete ML framework discussed is implemented in PyTorch 2.7.0, leveraging CUDA acceleration for efficient training on NVIDIA GPUs. Model optimization employs the Adam optimizer with weight decay (D_w), and an initial learning rate of (lr) selected through systematic hyperparameter optimization using Ray Tune (Liaw et al. 1807) with the Asynchronous Successive Halving Algorithm (ASHA; Li et al. 2020) scheduler combined with Hyperopt search (Bergstra et al. 2013). Learning rate scheduling is implemented using *ReduceLROnPlateau*[1] with a decay factor of 0.2, and a grace period of 10 epochs without validation improvement, allowing adaptive learning rate reduction when training plateaus and preventing overfitting while ensuring the model converges to optimal emergence detection capability.

Training is conducted for a maximum of 100 epochs with early stopping based on validation temporal-gradient RMSE, by the *ASHAScheduler*[2] and custom *TrialPlateauStopper*[3] (more details are provided in A). Training and validation datasets are batched and shuffled as determined by the *Hyperopt*[4] search. The

[1] https://docs.pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

[2] https://docs.ray.io/en/latest/tune/api/doc/ray.tune.schedulers.ASHAScheduler.html

[3] https://docs.ray.io/en/latest/tune/api/doc/ray.tune.stopper.TrialPlateauStopper.html

[4] https://hyperopt.github.io/hyperopt/

Algorithm 1: Training and Validation Procedures

Input: Training data, validation data, hyperparameters
Output: Trained LSTM model parameters

Initialize LSTM, Adam optimizer, learning rate scheduler
Preprocess data: min-max normalize, create sliding windows
Create batches: train (shuffle/no shuffle), validation (no shuffle)
for *epoch* $= 1$ **to** N_{epochs} **do**
    total_train_loss $\leftarrow 0$
    foreach *(input_seq, target_seq) in train_batches* **do**
        Forward pass in train mode
        Compute loss
        Clip gradients, update parameters
        Accumulate train loss
    end
    Train loss $\leftarrow$ average(total_train_loss)
    total_val_loss $\leftarrow 0$
    foreach *(input, target) in val_batches* **do**
        Forward pass in eval mode
        Compute derivative RMSE loss
        Accumulate validation loss
    end
    Validation loss $\leftarrow$ RMSE of prediction differences
    Update scheduler with validation loss
end

baseline training procedure was kept consistent across all experiments. Batching was used to reduce GPU memory demands by training on smaller data subsets, while shuffling was applied to prevent memorization of AR-specific patterns and to promote learning of generalizable temporal dynamics across different active regions. Remember that in order to balance active tiles with quiet tiles and reduce noise overhead, only the middle row of the 9×9 AR grid was used as input, and the top and bottom four rows of tiles were left out.

The best hyperparameters for each model setup chosen by the *Hyperopt* search are shown in Table 1. The model setup involves variations of model type and loss type. At the same time, the top-100 column highlights the percentage of times the model and loss combination was seen in the top 100 results (with different hyperparameters). The best hyperparameters are tabulated on the right-hand side of the table. The best model from this training setup was the `MagFluxLSTM` with a hybrid loss and the given hyperparameters tabulated in Table 1; therefore, that configuration will be discussed in more detail in Section 3.

Across our hyperparameter optimization runs, the `MagFluxLSTM` trained with the hybrid loss function emerged as the top-performing model, surpassing all other configurations. The results show that the *Hyperopt* search strongly favors a) the `MagFluxLSTM` architecture over the `MagFluxEnc-Dec` model, and b) the hybrid loss over MSE loss functions. The top 100 model configurations, ranked

Table 1. Model parameters and corresponding best-performing configurations. Notation: P_d is dropout probability; H is number of hidden layers; L is number of LSTM layers; B is batch size; lr is learning rate; λ is weight decay; α is hybrid loss mixing coefficient; $P_{\rm tf}$ is teacher forcing ratio. All values correspond to the optimal configuration selected after extensive hyperparameter optimization. The top-100 refers to the percentage of times the specific model setups showed up in the top 100 models during trials. The bolded model indicates the best performing model chosen to discuss in Section 3.

Model setup			**Hyperparameters**							
Model	**Loss Type**	**top-100**	P_d	H	L	B	lr	λ	α	P_{tf}
`MagFluxLSTM`	**hybrid loss**	**89%**	**0.1**	**128**	**4**	**64**	$\mathbf{7.06 \times 10^{-3}}$	$\mathbf{1.4 \times 10^{-6}}$	**0.7**	**-**
`MagFluxLSTM`	MSE loss	9%	0.2	64	1	32	2.32×10^{-3}	1.2×10^{-5}	1	-
`MagFluxEnc-Dec`	MSE loss	0%	0.3	32	4	32	1.7×10^{-3}	4.6×10^{-6}	1	0.5
`MagFluxEnc-Dec`	hybrid loss	2%	0.2	64	4	32	9.7×10^{-3}	2.5×10^{-5}	0.9	0.15

by validation temporal-gradient RMSE of the magnetic flux, are presented in Table 1. Notably, the `MagFluxEnc-Dec` model trained with MSE loss does not appear among the top-performing configurations. Overall, these results indicate that the `MagFluxLSTM` with a temporal-gradient–aware loss formulation more reliably captures time-dependent behavior, whereas the `MagFluxEnc-Dec` exhibits higher variance and weaker generalization under limited training data.

3. Results

The results for the best-performing `MagFluxLSTM` with hybrid loss (solid line) along with the other models (dashed lines), validated using the 5 testing ARs, are shown in Figure 3 (AR11726) of the main text and Figures 4, 5, 6 and 7 of the B (ARs 11698, 13165, 13179 and 13183, respectively). Note that the model predicts magnetic flux values in the min-max scaled space, which are subsequently denormalized for visualization purposes, as seen in Figure 3. The timelines plotted in Figures 3–7 are corrected for the solar geometric effect, and therefore the values are expressed in geometrically normalized Gauss units. Table 2 provides a summary of the results visualized in the aforementioned figures.

Out of the seven AR11726 tiles presented in Figure 3, the observed magnetic flux (dark solid line) in five of them (Tiles 41-45) fulfilled the criterion of Equation 5. The bottom right part of Figure 3 shows the positions of the tiles in the original 9×9 grid, before (left; 23:59 UTC, 04-18-2013) and after (right; 23:59 UTC, 04-21-2013) AR11726 emerged. These specific tiles were chosen for demonstration since most of the activity in the tracked 512×512 pixel cutout of the solar surface is centered around them. Note that the ML model's results in Figure 3 (all Φ_m timelines except the solid black) represent a value predicted 12 hours ahead. The same applies to the $d\Phi'_m/dt$ timeline, at the bottom panel of each tile's plots, which represents the derivative of Φ_m that `MagFluxLSTM` with the hybrid loss model predicted. The `MagFluxLSTM` model successfully predicts that there will be emergence in the five tiles where emergence is actually observed, while for the two quiet tiles, the model predictions never fulfilled the emergence criterion, and therefore, an emergence alarm was rightfully never produced for

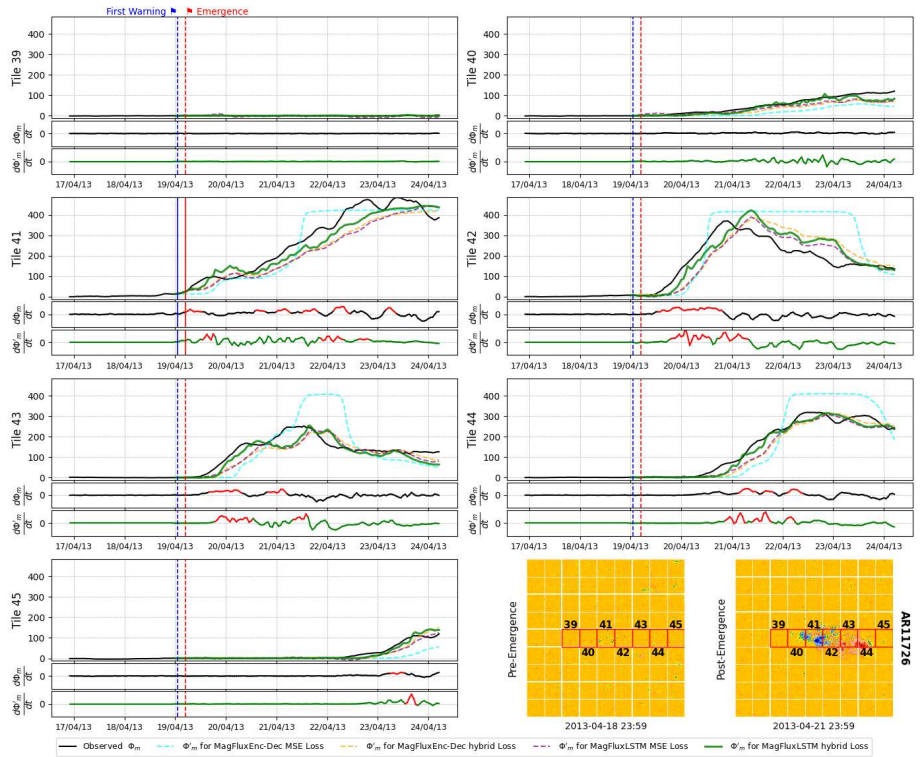


Figure 3. The Φ_m predictions for AR11726 obtained by the models presented in Table 1 for a 12-hour-ahead prediction window, in normalized Gauss units. The green solid line marks the `MagFluxLSTM` with the hybrid loss model, while the dashed lines show the other models. The observed flux is shown in black. The two panels below the main plot present the derivatives for a) the observed magnetic flux and b) the predicted magnetic flux by `MagFluxLSTM` hybrid loss model (top and bottom, respectively). Red segments in the derivative denote the times when the emergence criterion in Equation 5 was fulfilled. Blue vertical dashed lines mark the first predicted emergence across all tiles, whereas the red vertical dashed lines mark the first observed emergence across all tiles. The solid vertical lines indicate the tile in which these first emergence and warning occurred, within the entire AR.

these tiles. The `MagFluxLSTM` Hybrid Loss model achieves an average RMSE of 27.7 across all tiles in Figure 3.

Across the testing ARs, the `MagFluxLSTM` consistently captures the rise in magnetic flux relative to each tile's pre-emergence baseline. Table 2 summarizes the predicted versus observed magnetic flux increase times across the selected tiles. For AR11726 (Figure 3), the first magnetic flux increase occurs in Tile 41, marked by the red vertical line (First Emergence) as defined by the criterion in Equation 5. `MagFluxLSTM` successfully anticipates this emergence 4 hours in advance, which is the first predicted emergence as marked by the blue vertical line in Figure 3. This value is prescribed under the *Emergence* column in Table 2. In tiles 42-45, the model predicts an increase in Φ_m 6, 9, 18, and 5 hours before the observed increase, respectively (Table 2, Φ_m Increase). Remember that these

alarms are defined as the difference between the timestamps of the first red points in the derivative panels of the figure, plus 12 hours, to account for the fact that the $d\Phi'_m/dt$ values are provided to us by `MagFluxLSTM`, 12 hours ahead. For tiles (39 and 40), `MagFluxLSTM` with hybrid loss rightfully does not predict emergence.

Our hyperparameter sweep reveals a notable finding: the comparatively simple `MagFluxLSTM` architecture consistently surpasses the more elaborate `MagFluxEnc-Dec` model (yellow and cyan dotted lines in Figure 3). This preference can be attributed to the reduced model complexity and parameter count, leading to better generalization on the limited dataset of 53 ARs. Given a limited number of available ARs in SolARED, the lower-parameter `MagFluxLSTM` model avoids overfitting. The autoregressive decoder in `MagFluxEnc-Dec` introduces additional degrees of freedom and longer gradient pathways, potentially increasing the sensitivity to noise in the magnetic flux magnitude and derivative targets, negatively affecting the models' performance (Table 1). This may explain the higher ranking of the simpler model by Hyperopt: among the top 100 configurations ranked by validation derivative RMSE, 89% correspond to `MagFluxLSTM` with hybrid loss, compared to 2% for `MagFluxEnc-Dec`. Table 2 highlights the results for `MagFluxLSTM` (bold in Table 1). The ability of `MagFluxLSTM` with hybrid loss to predict not only the AR emergence, but also the increase in Φ_m in separate tiles, better than the rest models we experimented on, is also evident in the results of the rest validation ARs, as seen in Figures 4-7 of B. Table 2 summarizes the `MagFluxLSTM` prediction results for all validation ARs.

Overall, for all validation ARs, `MagFluxLSTM` with hybrid loss produced a positive emergence alarm (the fulfillment of the Equation 5 criterion on the predicted values), varying from 3 to 10 hours. Note that these values were obtained in an experimental setting. As described in Kasapis et al. (2025), a 4-hour delay can be applied to the prediction times to account for the time required to process data in real-time. In Table 2, only the ≥ 4 hour alarms are considered operationally successful — and are therefore marked in green, while the tiles where `MagFluxLSTM` incorrectly produced a Φ_m alarm are marked in red (ILAP; incorrect local activity predictions). When this operational criterion is applied to the magnetic flux increase predictions of the first tile emergence is observed in all validation ARs, we conclude that 3 out of five AR emergences would be predicted if the `MagFluxLSTM` with hybrid loss models were deployed in operations. This study, therefore, presents the first ML model that can correctly predict the time and space where AR magnetic flux emergence should be expected

4. Discussion and Conclusions

This study presents the development of a deep learning framework to predict the increase in magnetic flux associated with the emergence of ARs, addressing critical needs in space weather forecasting. Compared with earlier attempts to use an LSTM-based architecture for emergence prediction (Kasapis et al. 2025), this study explored several alternative approaches. First, min–max normalization is now computed strictly from the training set and applied to the validation and test sets, eliminating the information leakage present in the previous pipeline,

Table 2. Prediction of the magnetic flux increase for the five validation ARs, for 7 select tiles. In green are the operationally successful alarms and the tiles where the `MagFluxLSTM` with hybrid loss correctly predicted that Φ_m would not increase (marked as Quiet). In orange are the alarms that would be delayed in an operational-type setting. An ILAP (incorrect local activity prediction) indicates that the model predicted an increase in Φ_m when in reality there was none.

AR	**Φ_m Increase**							**Emergence**
	Tile 48	Tile 49	Tile 50	Tile 51	Tile 52	Tile 53	Tile 54	
AR11698	Quiet	4h Alarm	ILAP	14h Alarm	13h Alarm	Quiet	Quiet	7h Alarm
	Tile 39	Tile 40	Tile 41	Tile 42	Tile 43	Tile 44	Tile 45	
AR11726	Quiet	Quiet	4h Alarm	6h Alarm	9h Alarm	18h Alarm	5h Alarm	4h Alarm
	Tile 30	Tile 31	Tile 32	Tile 33	Tile 34	Tile 35	Tile 36	
AR13165	Quiet	ILAP	4h Alarm	7h Alarm	Quiet	Quiet	Quiet	7h Alarm
	Tile 39	Tile 40	Tile 41	Tile 42	Tile 43	Tile 44	Tile 45	
AR13179	Quiet	8h Alarm	3h Alarm	1h Alarm	Quiet	Quiet	Quiet	3h Alarm
	Tile 39	Tile 40	Tile 41	Tile 42	Tile 43	Tile 44	Tile 45	
AR13183	Quiet	Quiet	ILAP	8h Alarm	ILAP	Quiet	Quiet	10h Alarm

where scaling was fit to all data. Training was also significantly transformed for traditional techniques: instead of training on each tile and active region sequentially, our model is trained in mini-batches (with optional shuffling), which yields smoother gradient updates and reduces overfitting to AR-specific patterns, such as rapid increases in magnetic flux that are not associated with emergence. Hyperparameter tuning is now performed systematically using Hyperopt and Ray Tune with ASHA early stopping, enabling exploration of a broader and more expressive model space.

Architecturally, we extend the encoder–decoder LSTM used in prior work (Kasapis et al. 2025) by implementing teacher forcing (`MagFluxEnc-Dec`), and introduce a simpler stacked LSTM (`MagFluxLSTM`) that ultimately outperforms `MagFluxEnc-Dec` on derivative-based validation metrics. This new validation metric uses derivative RMSE to align the optimization objective with the scientific goal of accurately predicting the time of flux emergence. Finally, we incorporate a hybrid loss function that optimizes magnitude and gradient agreement together, addressing the prior model's inability to explicitly learn derivative dynamics. Excluding methodological differences, the biggest difference from Kasapis et al. (2025) is the use of continuum intensity and acoustic power variations as inputs and magnetic flux as the output. Given that a significant increase in magnetic flux precedes a decrease in continuum intensity, predicting magnetic flux rather than continuum intensity is more important for space weather forecasting. This lag between the decrease in continuum intensity and the increase in magnetic flux is also evident when comparing the predictions of the two studies, where in the best case of Kasapis et al. (2025), continuum intensity was predicted 36 hours ahead, whereas for magnetic flux, the best prediction was 10 hours ahead for AR13183 (Table 2).

Through systematic comparison of two LSTM-based architectures, a stacked LSTM (`MagFluxLSTM`) and an encoder-decoder architecture (`MagFluxEnc-Dec`), we demonstrate that the simpler stacked LSTM with the hybrid loss formulation

achieves superior predictive performance for this task. Extensive hyperparameter optimization revealed that 89% of the top-100 model configurations utilized the stacked `MagFluxLSTM` architecture, indicating strong convergence to this simpler, more robust approach for magnetic flux prediction. This work presents the first ML model trained for predicting magnetic flux associated with the emergence of solar active regions. Our results show that the `MagFluxLSTM` model with a hybrid loss can predict the emergence of magnetic flux 3-10 hours before it is observed, in an operational-type setting. The increase of magnetic flux associated with the emergence of the respective ARs was successfully predicted, in an operational-type setting, in three out of five cases.

The current model operates on individual tiles without explicit spatial-correlation modeling. Incorporating additional components, such as CNNs, into `MagFluxLSTM` to capture spatial emergence patterns may improve performance for extended AR complexes. While challenges remain in scaling to larger datasets and integrating more diverse inputs, our approach represents a step toward operational forecasting tools that combine physical insight with data-driven methods. Several avenues for future development are promising: integrating vector magnetic field observations could enhance prediction accuracy by providing a more complete characterization of magnetic field topology, and transfer learning approaches could enable application to related prediction problems. Beyond immediate forecasting applications, this work demonstrates the potential of deep learning approaches to extract predictive information from subtle pre-emergence signatures in solar observations. By balancing absolute flux prediction with derivative sensitivity through the hybrid loss we introduced, the `MagFluxLSTM` model captures both the magnitude and temporal evolution of flux emergence, enabling more accurate identification of the magnetic flux onset that can be associated with AR emergence. Furthermore, the model's success in capturing pre-emergence signatures suggests potential applications to other solar activity prediction problems, such as solar flare prediction, filament formation forecasting, and coronal hole evolution modeling and others.

Open Research Section

The model pipeline was implemented in PyTorch 2.7.0 on NVIDIA GPUs. Ray Tune module and ReduceLROnPlateau, TrialPlateauStopper, ASHAScheduler, Hyperopt implementations were used throughout the code. The SolARED capabilities and ML-ready data are available for download from https://sun.njit.edu/sarportal/. The acoustic power maps analyzed in this work are available from the corresponding author upon request. The code used to generate and process the dataset, following the methodology described in the paper, is available at https://github.com/jonastirona/SAR_EMERGENCE_RESEARCH/tree/flip_flux_and_intensity.

Acknowledgements This work was supported by the New Jersey Institute of Technology (NJIT) Grace Hopper AI Research Institute (GHAIRI) Seed Grant; the NASA AI/ML HECC Expansion Program, and the NASA grants 23-HGIO23_2-0077, 20-HSR20_2-0037,

80NSSC19K0630, 80NSSC19K0268, 80NSSC20K1870, and 80NSSC22M0162. We also thank the computing resources provided by the High Performance Computing (HPC) facility at NJIT.

References

Barnes, G., Birch, A., Leka, K., Braun, D.: 2014, Helioseismology of pre-emerging active regions. III. Statistical analysis. *The Astrophysical Journal* **786**, 19.

Bergstra, J., Yamins, D., Cox, D.D., et al.: 2013, Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms. *SciPy* **13**, 20.

Birch, A., Braun, D., Leka, K., Barnes, G., Javornik, B.: 2012, Helioseismology of pre-emerging active regions. II. Average emergence properties. *The Astrophysical Journal* **762**, 131.

Bobra, M.G., Ilonidis, S.: 2016, Predicting coronal mass ejections using machine learning methods. *The Astrophysical Journal* **821**, 127.

Camporeale, E.: 2020, ML-Helio: An Emerging Community at the Intersection Between Heliophysics and Machine Learning. *Journal of Geophysical Research (Space Physics)* **125**, e27502. DOI. ADS.

Georgoulis, M.K., Bloomfield, D.S., Piana, M., Massone, A.M., Soldati, M., Gallagher, P.T., Pariat, E., Vilmer, N., Buchlin, E., Baudin, F., et al.: 2021, The flare likelihood and region eruption forecasting (FLARECAST) project: flare forecasting in the big data & machine learning era. *Journal of Space Weather and Space Climate* **11**, 39.

Grigor'ev, V., Ermakova, L., Khlystova, A.: 2011, The dynamics of photospheric line-of-sight velocities in emerging active regions. *Astronomy reports* **55**, 163.

Hartlep, T., Kosovichev, A.G., Zhao, J., Mansour, N.N.: 2011, Signatures of emerging subsurface structures in acoustic power maps of the Sun. *Solar Physics* **268**, 321.

Hegde, M.: 2025, Predicting CME speed at 20 R$\odot$ using machine learning approaches. *Astrophysics and Space Science* **370**, 91.

Hochreiter, S., Schmidhuber, J.: 1997, Long short-term memory. *Neural computation* **9**, 1735.

Hoeksema, J.T., Liu, Y., Hayashi, K., Sun, X., Schou, J., Couvidat, S., Norton, A., Bobra, M., Centeno, R., Leka, K.D., Barnes, G., Turmon, M.: 2014, The Helioseismic and Magnetic Imager (HMI) Vector Magnetic Field Pipeline: Overview and Performance. *Solar Physics* **289**, 3483. DOI. URL.

Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., Zhang, H.: 2019, Deep learning with long short-term memory for time series prediction. *IEEE Communications Magazine* **57**, 114.

Hutchins, T., Kasapis, S., Farooki, H., Cuesta, M.E., Khoo, L., Pak, S., Czarnota, R., Rankin, J., Szalay, J., Xu, Z., Livadiotis, G., Sarlis, N., McComas, D.: 2026, Solar Energetic Particle Prediction in the Inner Heliosphere Using Deep Learning and PSP/IS$\odot$IS Data. *JGR: Machine Learning and Computation*. Manuscript under review.

Ilonidis, S., Zhao, J., Hartlep, T.: 2013, Helioseismic investigation of emerging magnetic flux in the solar convection zone. *The Astrophysical Journal* **777**, 138.

Ilonidis, S., Zhao, J., Kosovichev, A.: 2011, Detection of emerging sunspot regions in the solar interior. *Science* **333**, 993.

Jiao, Z., Sun, H., Wang, X., Manchester, W., Gombosi, T., Hero, A., Chen, Y.: 2020, Solar flare intensity prediction with machine learning models. *Space weather* **18**, e2020SW002440.

Kasapis, S., Kitiashvili, I.N., Kosovichev, A.G., Stefan, J.T., Apte, B.: 2023, Predicting the Emergence of Solar Active Regions Using Machine Learning. *Proceedings of the International Astronomical Union* **19**, 311.

Kasapis, S., Kitiashvili, I.N., Kosovichev, A.G., Stefan, J.T.: 2025, Prediction of Intensity Variations Associated with Emerging Active Regions using Helioseismic Power Maps and Machine Learning. *The Astrophysical Journal Supplement Series* **280**, 64.

Kasapis, S., Hosseinzadeh, P., Whitman, K., Egeland, R., Georgoulis, M., Vourlidas, A., Papaioannou, A., Lavasa, E., Anastasiadis, A., Giannopoulos, G., Muñoz-Jaramillo, A., Poduval, B., Kitiashvili, I.N., Kosovichev, A.G., Sadykov, V., Filali Boubrahimi, S., Hutchins, T.T., Farooki, H.A., Cuesta, M.E., Khoo, L.Y., Pak, S., Czarnota, R., Rankin, J.S., Szalay, J., Shen, M.M., Livadiotis, G., Xu, Z., McComas, D.J., Sarlis, N., Hristopulos, D., Engell, A.J., AbuBakr Ali, M., Abdelkawy, A.G.A., Shaltout, A.M.K., Beheary, M.M., Lee, C.O., Aminalragia-Giamini, S., Papadimitriou, C., Sandberg, I., Raptis, S., Hamdi, S.M., Laurenza, M., Stumpo, M., Rotti, S., Jackson, I., Ali, A., Baydin, A.G., Schwadron, N., Chatterjee, S., Dayeh, M., Oria, V., Nita, G.M., O'Keefe, P.M., Francis, F., Chong, C.-J., Kosovich,

P., Marroquin, R.D., Aydin, B., Martens, P.C., Zhao, L., Chen, Y., Wang, X., Bobra, M., Gombosi, T., Wang, V., Zhang, M., Torres, J., Chan, P.K., Angryk, R., Nedal, M., Kozarev, K., Arsenov, N., Zhang, P., Moreland, K., Bain, H., Hart, S., Starkey, M.J., Cliver, E.W., Ling, A.G., Alberti, T., Benella, S., Jiang, Y., Ferreira, S.H.: 2026a, Review of Machine Learning Models for Solar Energetic Particle Prediction. Manuscript under review.

Kasapis, S., Dogan, E., Kitiashvili, I.N., Kosovichev, A.G., Stefan, J.T., Butler, J.D., Tirona, J., Patil, S., Xu, M.: 2026b, SolARED: Solar Active Region Emergence Dataset for Machine Learning Aided Predictions. *arXiv preprint arXiv:2601.13145*.

Kosovichev, A.G.: 2011, Advances in Global and Local Helioseismology: An Introductory Review. In: Rozelot, J.-P., Neiner, C. (eds.) *Lecture Notes in Physics, Berlin Springer Verlag* **832**, 3. DOI. ADS.

Kosovichev, A.G., Duvall, T.L., Scherrer, P.H.: 1999, Time-distance helioseismology. *Advances in Space Research* **24**, 163. DOI. ADS.

Leka, K.D., Barnes, G., Birch, A.C., Gonzalez-Hernandez, I., Dunn, T., Javornik, B., Braun, D.C.: 2013, Helioseismology of Pre-emerging Active Regions. I. Overview, Data, and Target Selection Criteria. *ApJ* **762**, 130. DOI. ADS.

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-Tzur, J., Hardt, M., Recht, B., Talwalkar, A.: 2020, A system for massively parallel hyperparameter tuning. *Proceedings of machine learning and systems* **2**, 230.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I.: 1807, Tune: a research platform for distributed model selection and training (2018). *ArXiv:1807.05118*.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I.: 2018, Tune: A Research Platform for Distributed Model Selection and Training. *arXiv preprint arXiv:1807.05118*.

Liggett, M., Zirin, H.: 1985, Emerging flux in active regions. *Solar physics* **97**, 51.

Pesnell, W.D., Thompson, B.J., Chamberlin, P.: 2012, *The solar dynamics observatory (SDO)*, Springer.

Roy, S., Schmude, J., Lal, R., Gaur, V., Freitag, M., Kuehnert, J., van Kessel, T., Hegde, D.V., Muñoz-Jaramillo, A., Jakubik, J., et al.: 2025a, Surya: Foundation model for heliophysics. *arXiv preprint arXiv:2508.14112*.

Roy, S., Hegde, D.V., Schmude, J., Lin, A., Gaur, V., Lal, R., Mandal, K., Singh, T., Muñoz-Jaramillo, A., Yang, K., et al.: 2025b, SuryaBench: Benchmark Dataset for Advancing Machine Learning in Heliophysics and Space Weather Prediction. *arXiv preprint arXiv:2508.14107*.

Scherrer, P.H., Schou, J., Bush, R., Kosovichev, A., Bogart, R., Hoeksema, J., Liu, Y., Duvall, T., Zhao, J., Title, A., et al.: 2012a, The helioseismic and magnetic imager (HMI) investigation for the solar dynamics observatory (SDO). *Solar Physics* **275**, 207.

Scherrer, P.H., Schou, J., Bush, R.I., Kosovichev, A.G., Bogart, R.S., Hoeksema, J.T., Liu, Y., Duvall, T.L., Zhao, J., Title, A.M., Schrijver, C.J., Tarbell, T.D., Tomczyk, S.: 2012b, The Helioseismic and Magnetic Imager (HMI) Investigation for the Solar Dynamics Observatory (SDO). *Solar Physics* **275**, 207. DOI. URL.

Schrijver, C.J.: 2009, Driving major solar flares and eruptions: A review. *Advances in Space Research* **43**, 739.

Schunker, H., Roland-Batty, W., Birch, A.C., Braun, D.C., Cameron, R.H., Gizon, L.: 2024, A flux-independent increase in outflows prior to the emergence of active regions on the Sun. *Monthly Notices of the Royal Astronomical Society* **533**, 225.

Sherstinsky, A.: 2020, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* **404**, 132306.

Singh, N.K., Raichur, H., Brandenburg, A.: 2016, High-wavenumber solar f-mode strengthening prior to active region formation. *The Astrophysical Journal* **832**, 120.

Stefan, J.T., Kosovichev, A.G.: 2023, Exploring the Connection between Helioseismic Travel Time Anomalies and the Emergence of Large Active Regions during Solar Cycle 24. *The Astrophysical Journal* **948**, 1.

Tirona, J., Patil, S., Kasapis, S., Dogan, E., Stefan, J., Kitiashvili, I.N., Kosovichev, A.G., Xu, M.: 2026, Forecasting Continuum Intensity for Solar Active Region Emergence Prediction using Transformers. *arXiv preprint arXiv:2601.13144*.

Waidele, M., Roth, M., Singh, N., Käpylä, P.: 2023, On Strengthening of the Solar f-Mode Prior to Active Region Emergence Using the Fourier-Hankel Analysis. *Solar Physics* **298**, 30.

Appendix

A. Hyperparameter Optimization

Hyperparameter optimization was performed using Ray Tune using the Hyperopt search algorithm and the Asynchronous Successive Halving Algorithm (ASHA) for early-stopping of underperforming trials. The optimization objective was set to minimize the validation root-mean-square error (RMSE) of magnetic flux prediction. Maximum 100 epochs were allowed per trial, but trials could be terminated when their RMSE plateaued for 10 consecutive epochs as determined by the `TrialPlateauStopper` or by ASHA. The complete search space and tuning configuration can be found in Listing 1.

Listing 1: Ray Tune + Hyperopt search configuration for LSTM hyperparameter optimization.

```
search_space = {
    "learning_rate": hp.loguniform("learning_rate", log(1e-5), log(1
        e-2)),
    "hidden_size": hp.choice("hidden_size", [2, 4, 8, 16, 32, 64,
        128]),
    "num_layers": hp.choice("num_layers", [1, 2, 3, 4]),
    "dropout": hp.choice("dropout", [0, 0.1, 0.2, 0.3]),
    "batch_size": hp.choice("batch_size", [32, 64]),
    "weight_decay": hp.loguniform("weight_decay", log(1e-6), log(1e
        -3)),
    "n_epochs": 100,
    # Dataset
    "shuffle": hp.choice("shuffle", [True, False]),
    # Model architecture | Conditional Search Space
    "model": hp.choice(
        "model_branch",
        [
            {
                "model": "\texttt{MagFluxLSTM}",
            },
            {
                "model": "\texttt{MagFluxEnc-Dec}",
                "teacher_forcing_ratio": hp.choice(
                    "teacher_forcing_ratio", [0, 0.1, 0.15, 0.25,
                        0.5]
                ),
            },
        ],
    ),
    "lossFn": hp.choice(
        "lossFn_branch",
        [
            {
                "lossFn": "hybrid",
                "alpha": hp.choice("alpha", [0.1, 0.3, 0.5, 0.7,
                    0.9]),
            },
            {"lossFn": "value"},
        ],
    ),
```

```
}

# Scheduler to early-stop bad trials
scheduler = ASHAScheduler(
    metric="RMSE",
    mode="min",
    grace_period=15,  # Min epochs before a trial can be stopped
    reduction_factor=2,
)

# Search algorithm
search_alg = HyperOptSearch(space=search_space, metric="RMSE", mode=
    "min")

early_stopper = TrialPlateauStopper(
    metric="RMSE",
    mode="min",
    grace_period=10,  # Number of epochs to wait for improvement
)

# Set up the Tuner
ray.init(num_cpus=32, num_gpus=1, include_dashboard=False, _temp_dir
    ="/tmp/ray")
train_ref = ray.put(tensor_train)
val_ref = ray.put(tensor_val)
tuner = tune.Tuner(
    tune.with_resources(
        tune.with_parameters(
            main,
            train_ref=train_ref,
            val_ref=val_ref,
        ),
        {"gpu": 1 / 32, "cpu": 1},
    ),
    tune_config=tune.TuneConfig(
        num_samples=parse_args()[
            "sample_size"
        ],  # Number of different hyperparameter combinations to try
        scheduler=scheduler,
        search_alg=search_alg,
    ),
    run_config=ray.train.RunConfig(
        name="lstm_hyperparameter_search",
        stop=early_stopper,  # Max epochs per trial
    ),
)
```

B. Additional Results

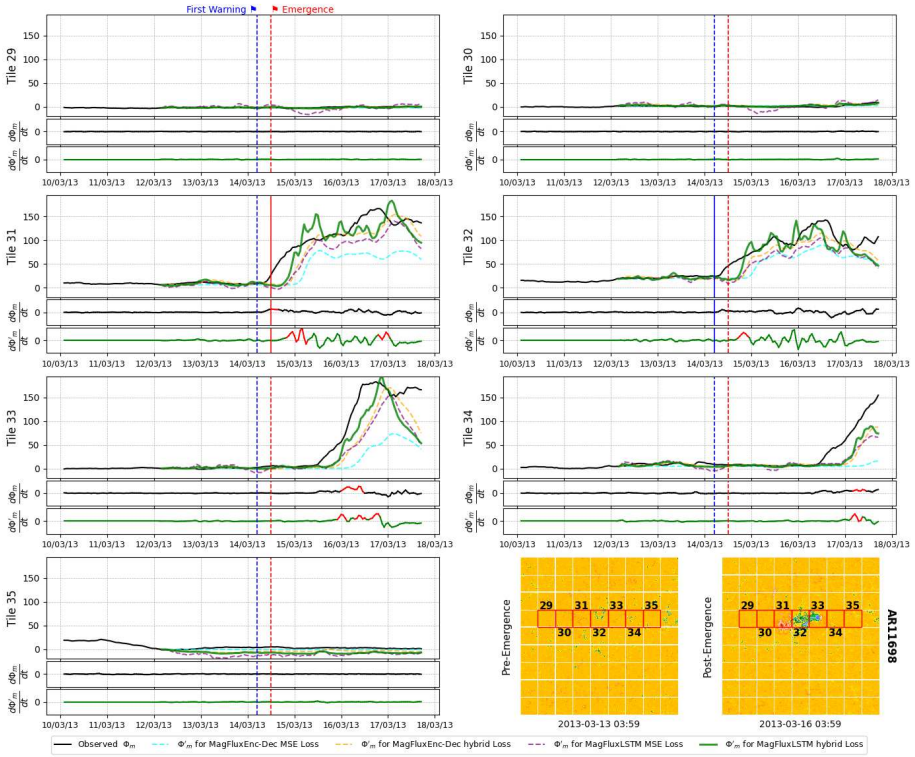


Figure 4. Same as Figure 3 for AR11698.

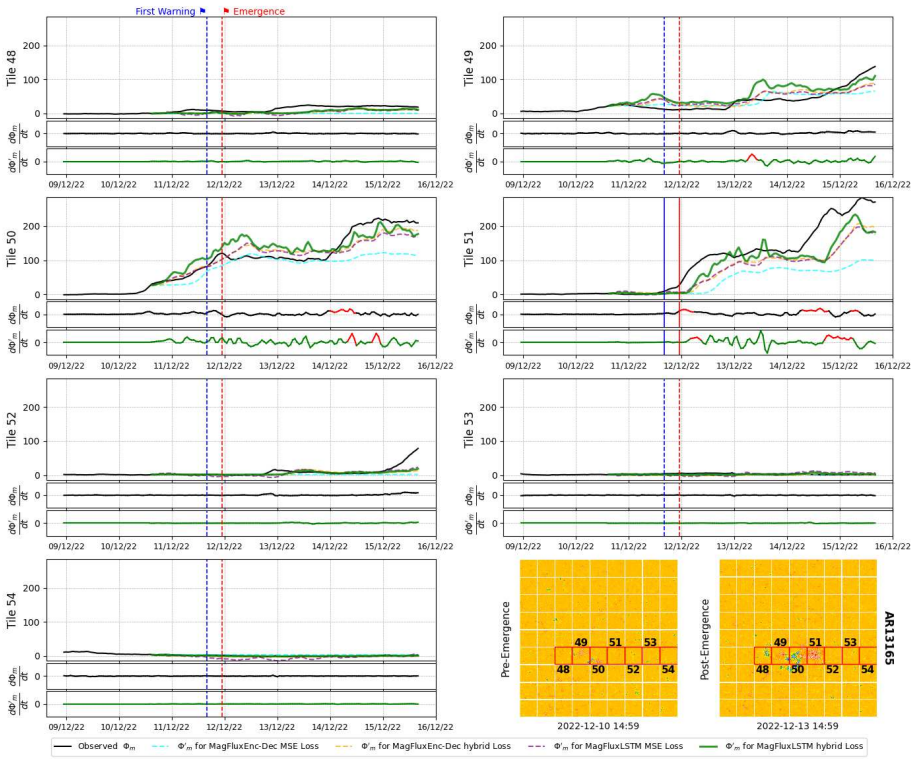


Figure 5. Same as Figure 3 for AR13165.

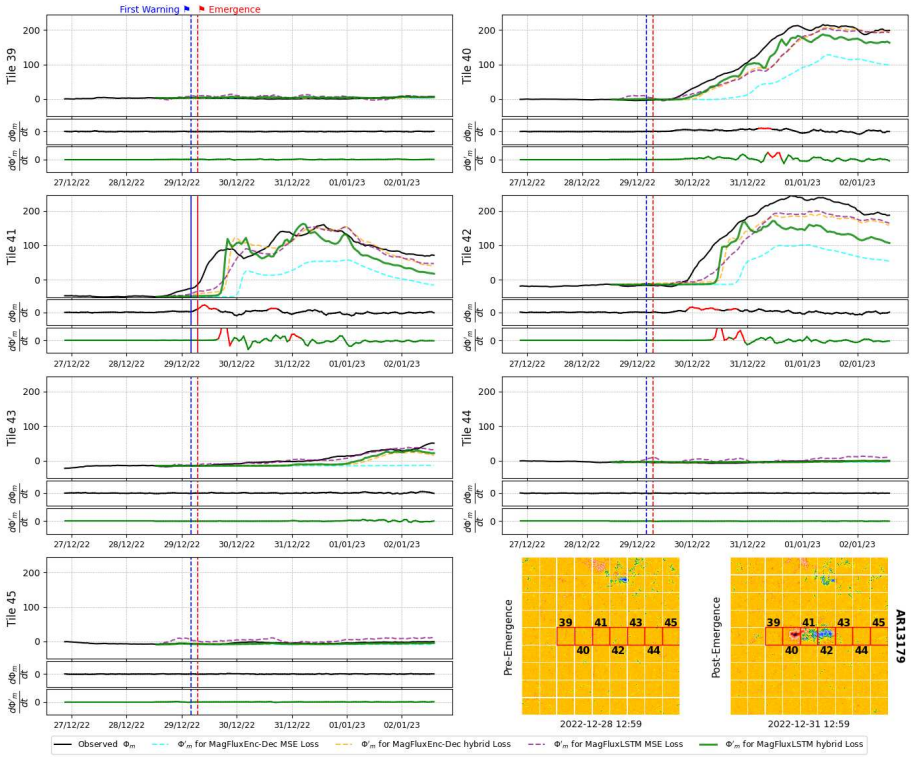


Figure 6. Same as Figure 3 for AR13179.

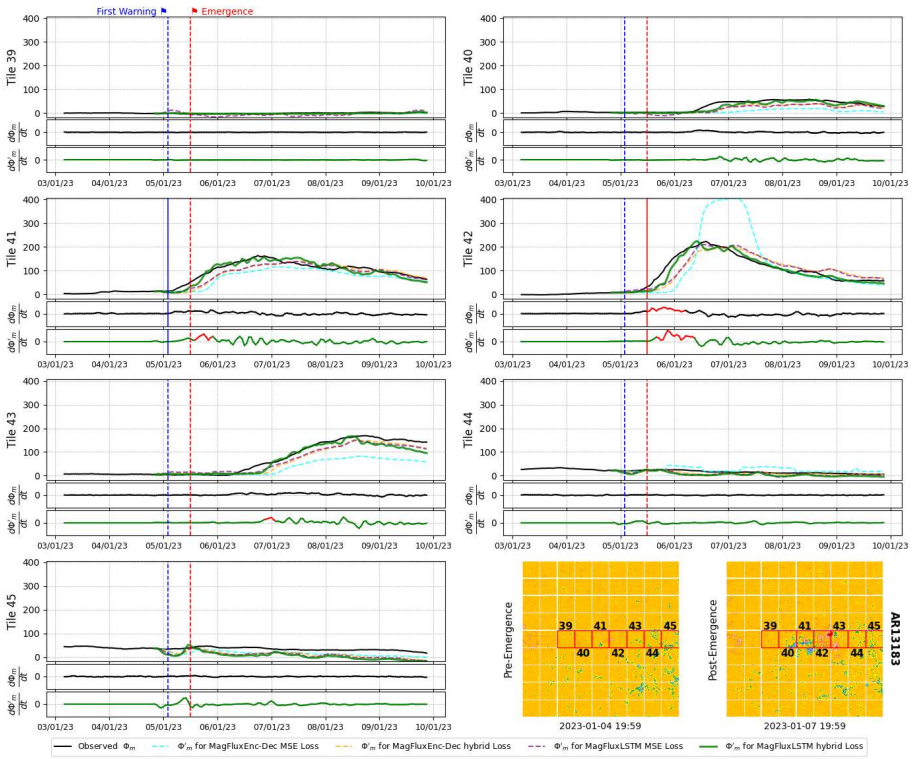


Figure 7. Same as Figure 3 for AR13183.