Deep Space Weather Model:
Long-Range Solar Flare Prediction from Multi-Wavelength Images

Supplementary Material

A. Additional Related Work

Datasets and benchmarks. Observations from space have significantly enhanced our understanding of astronomical phenomena and played a crucial role in advancing solar physics. For example, star tracking and localization have improved thanks to recent advances like Chin et al.'s event-based pipeline [12] and the StarNet dataset [15] for narrow-field star localization. In fields like satellite pose estimation, datasets such as SPEED [36] and SwissCube [31] address challenges such as scale variations and adverse illumination. Similarly, in remote sensing, datasets like DOTA [70] and EarthNet2021 [44] are used to study dynamic terrestrial processes. These examples highlight the importance of tailored benchmarks for developing and evaluating task-specific methods.

Solar flare prediction. Numerous methods have been proposed for solar flare prediction, including early approaches using Multi-layer perceptrons (MLPs) and more recent methods employing Convolutional Neural Networks (CNNs) and Recurrent Neural Networks, such as Long Short-Term Memory networks (LSTMs). DeFN incorporates 79 features extracted from sunspot images, including features related to coronal hot brightening and X-ray intensity trends specifically chosen for operational forecasting. Subsequently, Li et al.[40] propose a CNN model trained on Spaceweather HMI Active Region Patch (SHARP) magnetograms to predict flares, leveraging the ability of CNNs to extract spatial features. Concurrently, [42] develops an LSTM to capture the temporal evolution of active regions using both magnetic parameters and flare history, demonstrating the importance of temporal dynamics in flare prediction. However, these traditional approaches, including MLPs, CNNs, and LSTMs, while demonstrating potential for solar flare prediction, primarily rely on heuristic physical features or often have limitations in capturing long-range spatio-temporal dependencies.

More recently, Transformer-based models have been explored. For example, SolarFlareNet [1] utilizes a transformer-based framework to predict flares from time series of SHARP parameters, extending the prediction window to 72 hours.

Despite their strength in modeling long-range dependencies, the computational cost of Transformers scales quadratically with sequence length. This cost is a significant challenge when applying Transformers to the long, multi-channel, full-disk solar image time series considered in our work, where computational and memory demands can become prohibitive.

Table 4. Correspondence between flare classes and peak X-ray flux intensities

Flare Class	Peak X-ray Flux (I) [W/m^2]
X	$I > 10^{-4}$
M	$10^{-5} < I \leq 10^{-4}$
C	$10^{-6} < I \leq 10^{-5}$
O	$I \leq 10^{-6}$

Masked autoencoders. The Masked Autoencoder (MAE) approach introduced by He et al.[29] has inspired a wide range of extensions and adaptations across various domains. Following the MAE, numerous extensions have explored diverse applications and architectures, geospatial representation learning [55], motion forecasting [11], 3D point clouds [71], and facial landmark estimation [72]. For instance, Traj-MAE [10] adapts MAE for trajectory prediction in autonomous driving, using diverse masking strategies and a continual pre-training framework to capture social and temporal interactions. In the realm of video understanding, several MAE-based methods have been proposed [26, 64, 67, 69]. VideoMAC [53] addresses the resource-intensive nature of many Vision Transformer-based approaches by combining masked autoencoders with ConvNets, using a dual encoder architecture for inter-frame consistency.

Deep SSMs. Deep SSMs are founded on the Linear State-Space Layer [23], inspired by classical state space models in control theory [33]. They achieve efficient sequence modeling by leveraging the HiPPO matrix [22], which enables the memorization of input sequences through optimal polynomial approximation. For example, S4 [25] introduces a method for learning the HiPPO matrix. Building upon S4, S5 [61] proposes a new state space layer that utilizes a single multi-input, multi-output SSM instead of S4's bank of single-input, single-output SSMs. Furthermore, S5 uses an efficient parallel scan for computation, removing the need for the convolutional approach used in S4 and its associated convolution kernel computation.

B. Flare Class Definitions

Table 4 shows the standard classification of solar flares based on their peak X-ray flux, I, measured in the 1-

8 Å wavelength range by the X-ray Sensor (XRS) on board the Geostationary Operational Environmental Satellites (GOES). This classification is widely used in solar physics and space weather forecasting.

Within the X-class, flares are further categorized by a linear scale. An X1.0 flare corresponds to a peak flux of 10^{-4} W/m^2. The number following the 'X' indicates a multiple of this base value. For example, an X2.0 flare has a peak flux of 2×10^{-4} W/m^2, an X3.0 flare has a peak flux of 3×10^{-4} W/m^2, and so on. This same linear scaling applies within the M and C classes as well.

C. Deep Space Weather Model

C.1. Multi-channel Representation Beyond Conventional RGB

Our approach is analogous to how the three channels of an RGB image represent colors within the visible spectrum. However, by incorporating HMI and AIA images, we extend the concept of multi-channel representation to higher dimensionality, encompassing a broader range of the electromagnetic spectrum.

C.2. Parallel 2D and 3D Convolutions

The DCSM begins by applying two parallel convolutional operations to $\mathbf{h}_{\mathrm{ds}}^{(l)}$:

$$\mathbf{F}_{\mathrm{fused}} = \mathrm{Conv3D}(\mathbf{h}_{\mathrm{ds}}^{(l)}) + \mathrm{Conv2D}(\mathbf{h}_{\mathrm{ds}}^{(l)}), \quad (4)$$

where Conv3D and Conv2D denote a 3D convolution and a 2D convolution applied independently to each frame, respectively. The 3D convolution is intended to capture spatio-temporal patterns across channels. By contrast, the 2D convolution focuses on spatial features within each channel (e.g., sunspot structures). Their outputs are summed element-wise to produce the fused feature map $\mathbf{F}_{\mathrm{fused}}$.

C.3. Justification for Adopting S5

We adopt the S5 [61] to model these multi-channel solar images accurately based on the following considerations:

Time-invariance for continuous modalities. Time-variant SSMs, such as Mamba [21], introduce selection mechanisms that may degrade performance on continuous modalities like solar images, especially multi-wavelength observations. By contrast, time-invariant (LTI) SSMs are suggested to perform better on continuous signals [21]. Supporting this, the experiment on YouTubeMix in [21] indicates that LTI models, such as S5, may be more suitable when the input is a continuous modality.

MIMO structure for multi-channel efficiency. Single-input, single-output (SISO) setups, commonly used in previous deep SSM architectures (e.g., [24, 25]), cannot fully leverage the multi-channel nature of the input. In contrast to the SISO approach, the multi-input multi-output (MIMO) structure of S5 permits direct modeling of inter-channel dependencies and offers improved computational efficiency.

C.4. S5 Mathematical Formulation

The core of the S5 layer is a MIMO SSM, which can be represented in continuous time. Drawing inspiration from continuous system equations, an input $\mathbf{u}(t) \in \mathbb{R}^D$, a latent state $\mathbf{x}(t) \in \mathbb{R}^P$, and an output $\mathbf{y}(t) \in \mathbb{R}^D$ are considered. The general form of a continuous-time linear SSM can be defined as:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \quad (5)$$

where $\mathbf{A} \in \mathbb{C}^{P \times P}$, $\mathbf{B} \in \mathbb{C}^{P \times D}$, $\mathbf{C} \in \mathbb{C}^{D \times P}$, and $\mathbf{D} \in \mathbb{R}^{D \times D}$ denote the state matrix, input matrix, output matrix, and feedthrough matrix, respectively. To enable efficient parallel computation, $\mathbf{A}$ is diagonalized as $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, where $\mathbf{\Lambda} \in \mathbb{C}^{P \times P}$ is a diagonal matrix and $\mathbf{V} \in \mathbb{C}^{P \times P}$ is the matrix of eigenvectors, enabling us to rewrite the system dynamics as:

$$\frac{d\tilde{\mathbf{x}}(t)}{dt} = \mathbf{\Lambda}\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}\mathbf{u}(t), \quad \mathbf{y}(t) = \tilde{\mathbf{C}}\tilde{\mathbf{x}}(t) + \mathbf{D}\mathbf{u}(t), \quad (6)$$

where $\tilde{\mathbf{x}}(t) = \mathbf{V}^{-1}\mathbf{x}(t)$, $\tilde{\mathbf{B}} = \mathbf{V}^{-1}\mathbf{B}$, and $\tilde{\mathbf{C}} = \mathbf{C}\mathbf{V}$. This is the reparameterized system with a diagonal state matrix, which is crucial for the efficiency of S5.

This diagonalized system is then discretized using the Zero-Order Hold (ZOH) method with learnable timescale parameters $\mathbf{\Delta} \in \mathbb{R}^P$. The ZOH method assumes that the input function remains constant over each interval defined by the timescale parameter. In practice, the feedthrough matrix $\mathbf{D}$ is restricted to be diagonal. Thus, the S5 layer has the learnable parameters $\tilde{\mathbf{B}} \in \mathbb{C}^{P \times D}$, $\tilde{\mathbf{C}} \in \mathbb{C}^{D \times P}$, $\mathrm{diag}(\mathbf{D}) \in \mathbb{R}^D$, $\mathrm{diag}(\mathbf{\Lambda}) \in \mathbb{C}^P$, and the timescale parameters $\mathbf{\Delta} \in \mathbb{R}^P$. The performance of S5 is sensitive to the initialization of $\mathbf{A}$. It is initialized with a diagonalized HiPPO-N matrix.

C.5. Loss Function

Our loss function comprises three components: the conventional cross-entropy loss (L_{CE}), the BSS loss (L_{BSS}), and the GMGS loss (L_{GMGS}). We employ the BSS and GMGS losses, introduced by [34]. The BSS loss is used to optimize the BSS, a proper scoring rule that comprehensively evaluates probabilistic predictions. Furthermore, the BSS loss is differentiable, enabling efficient optimization through gradient-based methods. The GMGS loss is used to improve the GMGS using its own score matrix for the weights in the loss calculation.

Table 5. Experimental setup for the proposed method.

Epoch (first stage)	20
Epoch (second stage)	15
Batch size	32
Optimizer	AdamW ($\beta_1 = 0.9,\ \beta_2 = 0.95$)
Learning Rate (first stage)	4.0×10^{-5}
Learning Rate (second stage)	4.0×10^{-5}
Weight decay (first stage)	5.0×10^{-2}
Weight decay (second stage)	5.0×10^{-2}
$\#L_{\text{SSE}}$	3
$\#L_{\text{LT}}$	1
$\#D$	64
$\#\lambda_{\text{CE}}$	1
$\#\lambda_{\text{BSS}}$	2
$\#\lambda_{\text{GMGS}}$	1
$\#k$	4
$\#m$	672

The BSS loss is defined as:

$$L_{\text{BSS}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{I}\Big(p(\hat{y}_{ni}) - y_{ni}\Big)^2, \tag{7}$$

and the GMGS loss is defined as:

$$L_{\text{GMGS}} = -\frac{1}{N}\sum_{n=1}^{N} s_{i^*j^*}\sum_{i=1}^{I} y'_{ni}\log\Big(p(\hat{y}_{ni})\Big), \tag{8}$$

where

$$i^* = \text{argmax}_i(y_{ni}), \tag{9}$$
$$j^* = \text{argmax}_j(p(\hat{y}_{nj})), \tag{10}$$

N and I denote the number of samples and the number of classes, respectively. y'_{ni} is the label-smoothed version of y_{ni}, $p(\hat{y}_{ni})$ is the predicted probability for the i-th class of the n-th sample, and $s_{i^*j^*}$ denotes the element from the GMGS score matrix corresponding to the true class i^* and the predicted class j^*, respectively.

Our overall loss function is a weighted sum of these three components:

$$L = L_{\text{CE}} + \lambda_{\text{GMGS}}L_{\text{GMGS}} + \lambda_{\text{BSS}}L_{\text{BSS}}, \tag{11}$$

where λ_{GMGS} and λ_{BSS} are the loss weights controlling the contributions of the GMGS and BSS losses, respectively.

D. FlareBench

Table 6. Experimental setup for the Sparse MAE.

Epoch	20
Batch size	32
Optimizer	AdamW ($\beta_1 = 0.9,\ \beta_2 = 0.95$)
Learning Rate	4.0×10^{-3}
Weight decay	5.0×10^{-2}
$\#\alpha$	20
$\#L_{\text{enc}}$	8
$\#L_{\text{dec}}$	12
$\#D_{\text{pre}}$	128
$\#r_l$	0.3
$\#r_h$	0.5
$\#r_f$	0.5

Data sources and composition. In this study, we constructed FlareBench by combining solar observation data from the Joint Space Operations Center (JSOC)[1] with X-ray flux measurements from the Geostationary Operational Environmental Satellites. Our dataset includes:

1). AIA [39] level 1 images in nine wavelengths:
  - Extreme ultraviolet (EUV): 94 Å, 131 Å, 171 Å, 193 Å, 211 Å, 304 Å, and 335 Å
  - Ultraviolet (UV): 1600 Å
  - Visible light: 4500 Å

2). High-resolution (1K) magnetograms from the HMI [58], also obtained from JSOC.

We used the long-wavelength channel (1–8 Å) X-ray flux data from the GOES X-ray Sensor for class labels. Specifically, we collected Science-level data from GOES-15 for the period 2011-2020 and from GOES-16 for 2021-2022.

E. Experimental Setup

E.1. Implementation Details

Deep SWM. Table 5 illustrates the experimental settings of the proposed method. Our model had approximately 1.59M trainable parameters and 4.64G multiply-add operations. The proposed model was trained on an Nvidia H200 with 140GB of GPU memory and an Intel Xeon PLATINUM 8580 processor. It took approximately three hours to train our model. The inference time was approximately 12 ms per sample.

We used the training set to train our model, the validation set for tuning the hyperparameters, and the test set for evaluating the model's performance. We computed the GMGS score on the validation set every epoch. The performance on the test set was given by the model that achieved the highest GMGS score on the validation set.

[1] http://jsoc.stanford.edu.

Sparse MAE. Sparse MAE, as described in Section 4.4, uses an encoder-decoder architecture with a reconstruction loss, similar to the original MAE [29]. Here, we provide details of the specific configurations used in our implementation.

The encoder of our pretraining model, composed of L_{enc} Transformer layers, is trained to encode the masked input resulting from the two-phase masking process applied to $\mathbf{V}_t$ into an intermediate feature representation. This representation is then processed by a decoder consisting of L_{dec} Transformer layers, which aims to reconstruct the original image. The reconstruction loss is calculated as the mean squared error between the original image and the reconstructed image. Following [29], the loss is computed only over the masked pixels.

The experimental setup for MAE pretraining is shown in Table 6. For MAE pretraining, the number of trainable parameters and the number of multiply-accumulate operations for the proposed method are 2.56M and 27.68G, respectively. For MAE pretraining, we followed the same procedure using the training, and validation sets.

E.2. Preprocessing and Data Augmentation

We pre-processed the datasets by performing the following steps sequentially:

1. Resize all images from 1024 × 1024 to 256 × 256 to reduce computational complexity.
2. For HMI images, mask the timestamp information in the bottom-left corner.
3. To align the solar scales between AIA and HMI images, crop the edges of the AIA images, followed by resizing to match the HMI resolution using bilinear interpolation.
4. Perform standardization on each channel of all images in the dataset.
5. Synchronize the HMI images, AIA images, and class labels to a 1-hour cadence through temporal resampling to maintain a consistent time series.

For data augmentation, we applied random rotations, scaling, brightness and contrast adjustments, Gaussian blur, and channel-specific noise addition.

E.3. Classifier Re-training

Given the class imbalance in our dataset, sampling is necessary during training to ensure an equal number of samples across classes. However, oversampling can lead to overfitting, especially for the X-class, which has few samples. Consequently, our proposed method incorporates a two-stage approach. In the first stage, we train the model on the original dataset. In the second stage, we perform model retraining based on Classifier Re-training[35]. This two-stage process mitigates overfitting while addressing the class imbalance.

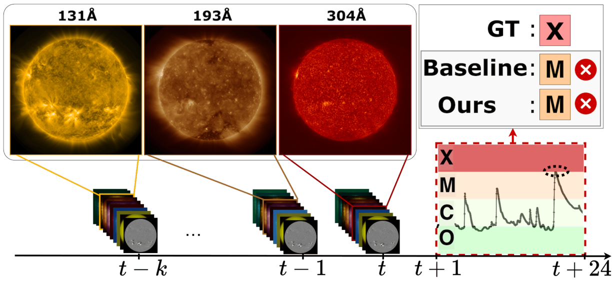

Figure 7. X-ray flux transitions leading up to an X1.0 flare event.

E.4. Evaluation Metrics

As described in Section 5.1, we use GMGS, BSS, and TSS to evaluate the performance of our model. These metrics are defined below.

The GMGS is defined as the trace of the product of a scoring matrix S and a contingency table P:

$$\text{GMGS} = \text{tr}(S^T \cdot P), \tag{12}$$

where S and P denote the I-rank scoring matrix with an element s_{ij} and the I-categorical contingency table with an element p_{ij}, respectively. The GMGS is an important metric in recent solar flare prediction studies [17]. The elements s_{ij} of the symmetric scoring matrix S are defined as:

$$s_{ii} = \frac{1}{I-1}\left[\sum_{k=1}^{i-1} a_k^{-1} + \sum_{k=i}^{I-1} a_k\right] \quad (1 \leq i \leq I), \tag{13}$$

$$s_{ij} = \frac{1}{I-1}\left[\sum_{k=1}^{i-1} a_k^{-1} + \sum_{k=i}^{j-1}(-1) + \sum_{k=j}^{I-1} a_k\right] \\ (1 \leq i < j \leq I), \tag{14}$$

$$a_i = \frac{1 - \sum_{k=1}^{i} p_k}{\sum_{k=1}^{i} p_k} \quad (1 \leq i \leq I), \tag{15}$$

$$p_i = \sum_{j=1}^{I} p_{ij} \quad (1 \leq i \leq I). \tag{16}$$

The BSS, a standard metric for evaluating the reliability of solar flare forecasts [48], is defined as:

$$\text{BSS} = \frac{\text{BS} - \text{BS}_c}{0 - \text{BS}_c}, \tag{17}$$

$$\text{BS} = \sum_{n=1}^{N}\sum_{i=1}^{I}(p(\hat{y}_{ni}) - y_{ni})^2, \tag{18}$$

$$\text{BS}_c = \sum_{n=1}^{N}\sum_{i=1}^{I}(f - y_{ni})^2, \tag{19}$$

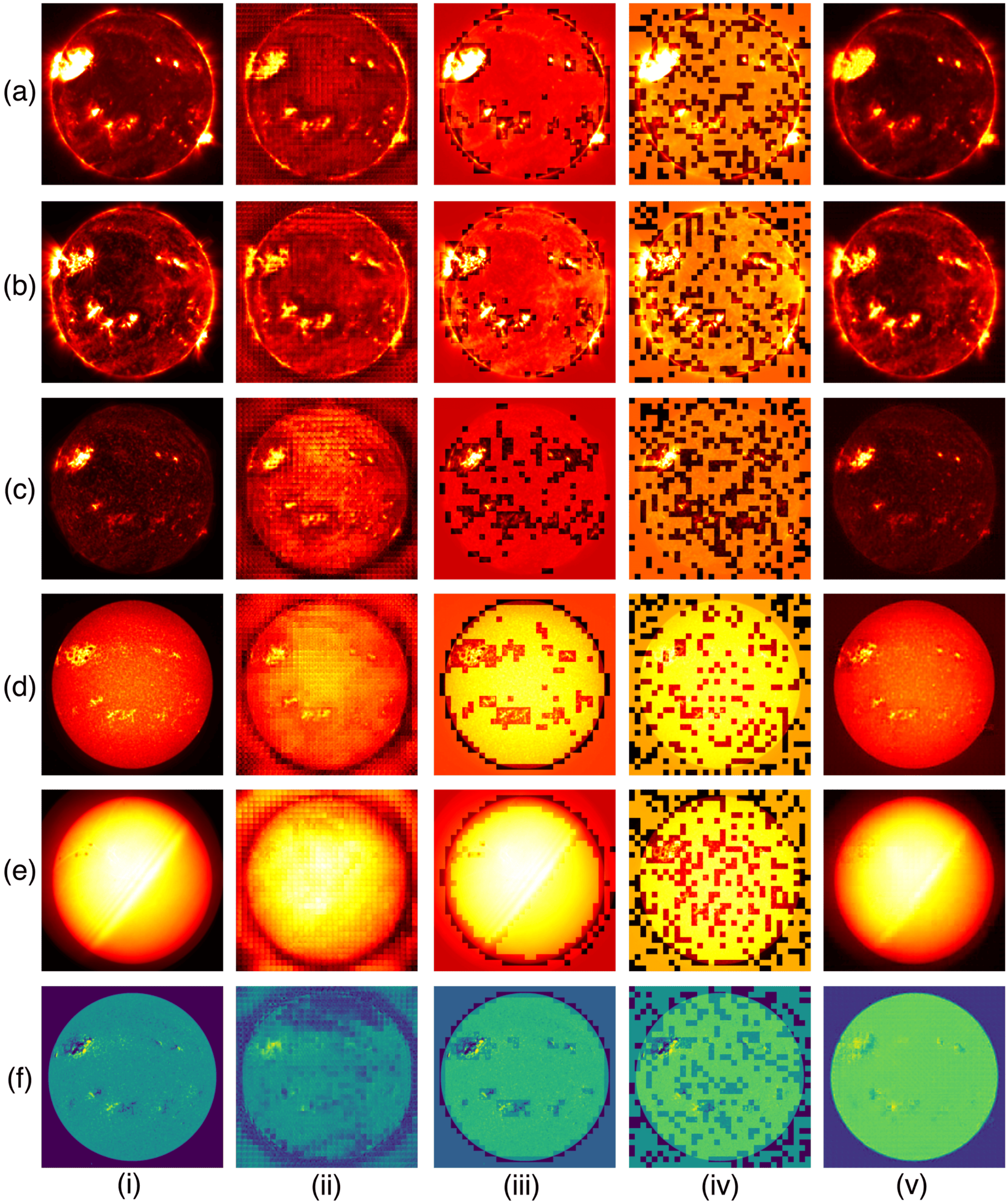

Figure 8. Reconstruction results obtained from the baseline MAE [29] ($\rho = 0.5$) and our proposed Sparse MAE. Rows (a), (b), (c), (d), (e), and (f) show 94 Å, 171 Å, 304 Å, 1600 Å, 4500 Å AIA, and HMI images, respectively, captured three hours before an upcoming X-class flare. Columns (i), (ii), (iii), (iv), and (v) present the original image, the baseline reconstruction, a visualization of patches with the top $\alpha\%$ highest standard deviation highlighted, the spatial-level masking of the Sparse MAE, and the reconstruction of the Sparse MAE, respectively.

Model	Masking	GMGS↑	BSS$_{\geq \mathrm{M}}$↑	TSS$_{\geq \mathrm{M}}$↑	MSE ($r < 0.65$)↓	MSE↓
(1-i)	MAE [29] (ρ=0.75)	0.286 ±0.090	0.067 ±0.306	0.428 ±0.173	9.837 ±0.365	4.147 ±0.378
(1-ii)	MAE [29] (ρ=0.5)	0.420 ±0.062	**0.354 ±0.163**	0.402 ±0.100	7.583 ±0.372	5.887 ±0.329
(1-iii)	Sparse MAE (Ours)	**0.582 ±0.032**	0.334 ±0.299	**0.543 ±0.074**	**3.255 ±0.180**	**2.461 ±0.115**

Table 7. Ablation study: impact of masking strategies.

Model	Architecture	GMGS↑	BSS$_{\geq \mathrm{M}}$↑	TSS$_{\geq \mathrm{M}}$↑
(3-i)	Attention [66]	0.311 ± 0.177	**0.753 ± 0.117**	0.287 ± 0.151
(3-ii)	Mamba [21]	0.364 ± 0.070	0.663 ± 0.032	0.364 ± 0.087
(3-iii)	S5 [61]	**0.582 ± 0.032**	0.334 ± 0.299	**0.543 ± 0.074**

Table 8. Performance comparison of different architectures.

where N, I, y_{ni}, $p(\hat{y}_{ni})$, and f are the number of samples, the number of classes, the true label of the i-th class for the n-th sample, the predicted probability of the i-th class for the n-th sample, and the climatological event rate, respectively.

The TSS is given by:

$$\mathrm{TSS} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} - \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{20}$$

where TP, FP, FN, and TN denote the number of true positives, false positives, false negatives, and true negatives in the contingency table, respectively.

F. Additional Experiments

F.1. Performance of Human Forecasters

The performance of human experts in daily solar flare forecasting operations from 2000 to 2015 was reported by Kubo et al. [37], and is summarized in Table 1. These human experts were engaged in the same forecasting problem as FlareBench: predicting the maximum solar flare class within a 24-hour period. To issue daily forecasts, they utilized solar indicators, including the current and historical X-ray flux, sunspot magnetic field configurations, and chromospheric brightenings in active regions, highlighting their expertise in operational solar flare prediction.

F.2. X-ray Flux Transitions for a Challenging Case

Fig. 5 (d) illustrates a failed X-class prediction. Fig. 7 shows the X-ray flux transitions for this event, including the 24-hour period leading up to the X1.0 flare. Appendix B defines the flare classes, including the notation where a number follows the class letter (e.g., X1.0, M2.5). During the 24 hours preceding the X-class flare, there were two distinct peaks in X-ray flux, each corresponding to M-class flares. Furthermore, the X-class flare represented the boundary between X-class and M-class flares. These factors likely contributed to the difficulty in accurately classifying the flare class.

F.3. Qualitative Results for Pretraining

Fig. 8 illustrates the reconstruction results obtained from the baseline MAE [29] with a mask ratio (ρ) of 0.5 and our proposed Sparse MAE. Rows (a), (b), (c), (d), (e), and (f) show 94 Å, 171 Å, 304 Å, 1600 Å, 4500 Å AIA, and HMI images, respectively, captured three hours before an upcoming X-class flare. Columns (i), (ii), (iii), (iv), and (v) present the original image, the baseline reconstruction, a visualization of patches with the top $\alpha\%$ highest standard deviation highlighted, the spatial-level masking of the Sparse MAE, and the reconstruction of the Sparse MAE, respectively.

As depicted in subfigures (a-ii) and (a-v), and the others, the Sparse MAE reconstructs features in and around sunspots with high fidelity. By contrast, the baseline method struggles to reproduce fine details in these regions. These observations suggest that the enhanced representation of sunspots in the Sparse MAE reconstructions can be attributed to its two-phase masking strategy, which emphasizes preserving essential features such as sunspots.

F.4. Additional Ablation Study

Pretraining ablation. Table 7 presents an ablation study comparing the impact of different pretraining methods on solar flare prediction performance. We evaluate three models: (2-i) using MAE with a mask ratio (ρ) of 0.75, (2-ii) using MAE with ρ = 0.5, and (2-iii) our proposed Sparse MAE. The table presents the GMGS, BSS$_{\geq \mathrm{M}}$, and TSS$_{\geq \mathrm{M}}$ scores for the solar flare prediction task, and the MSE($r < 0.65$) and MSE from the pretraining phase. MSE($r < 0.65$) and MSE are computed as the mean squared error over the masked patches. MSE($r < 0.65$) is restricted to patches within a defined solar region: a circle centered at the image center with a normalized radius of 0.65 (where the distance from the image center to a corner is normalized to 1.0). In this context, a normalized radius of 0.65 defines the boundary of the solar disk. We focus on MSE($r < 0.65$) because accurate reconstruction of the solar disk is more critical for solar flare prediction than reconstruction of the surrounding non-solar regions.

The results reveal differences in performance across the evaluated models. Models (2-i) and (2-ii) underperform Model (2-iii) in terms of both GMGS and MSE($r < 0.65$). Specifically, Model (2-i) underperforms Model (2-iii) by 0.296 and 6.582 points in GMGS and MSE($r < 0.65$), respectively, while Model (2-ii) underperforms Model (2-iii) by 0.162 and 4.328 points in GMGS and MSE($r < 0.65$),

		Predicted Flare Class			
		O	C	M	X
Observed flare class	O	5953	1110	29	167
	C	1427	2307	1211	2329
	M	111	321	433	1394
	X	10	32	30	139

Table 9. Confusion matrix for the test set of the third fold.

Observed class	Predicted class	GMGS_influence
C	X	0.1007
C	O	0.0560
O	C	0.0435
C	M	0.0281
M	X	0.0278

Table 10. The error analysis on the third fold's test set using the $\mathrm{GMGS}_{\mathrm{Influence}}$.

respectively. These results indicate that the improved reconstruction of crucial solar regions, such as sunspots in sparse solar images, achieved with Sparse MAE, leads to extracting features more relevant for solar flare prediction.

Deep SSM ablation. Table 8 presents the performance impact of different architectures in the temporal modeling components. We compared models using the following architectures for capturing temporal dependencies: (3-i) Attention [66], (3-ii) Mamba [21], and (3-iii) S5 [61]. In our method, these architectures replace the ST-SSM, LT-SSM, and their integration mechanism. From Table 8, Models (3-i) and (3-ii) underperformed Model (3-iii) in GMGS by 0.271, 0.218 points, respectively. These results suggest that S5 [61] accurately captures temporal dependencies in solar images.

F.5. Error Analysis

Table 9 presents the confusion matrix obtained using our method on the test set of the third fold. Given the significant impact of X-class solar flares, our model prioritizes their accurate prediction. This prioritization results in more false positives for the X-class, as illustrated in the confusion matrix because correctly identifying these impactful events is paramount.

We defined samples that were incorrectly predicted as failure cases. There were 8,832 failure cases identified in the third fold of the time-series cross-validation. Table 10 categorizes the failed cases. We used the metric $\mathrm{GMGS}_{\mathrm{Influence}}$ (as introduced in [34]) to calculate the influence of failure cases on the GMGS. The influence for each element (i,j) of the confusion matrix is defined as:

$$\mathrm{GMGS}_{\mathrm{Influence}_{ij}} = \frac{c_{ij}(s_{ii} - s_{ij})}{N}, \tag{21}$$

where c_{ij}, s_{ij}, and N represent element (i, j) of the confusion matrix, element (i, j) of the GMGS score matrix (detailed in Subsection 5.1), and the total number of samples, respectively. This metric provides a quantitative measure of how much each type of error negatively impacts the overall GMGS.

Table 10 indicates that the bottleneck is the misclassification of C-class flares as X-class. Given the potential for severe consequences, the model prioritizes predicting X-class flares. This prioritization reduces the risk of missing true X-class events (false positives) but may increase false negatives, as shown in the confusion matrix (Table 9). This trade-off is a deliberate design choice to reduce the likelihood of failing to identify high-impact events.