J. Space Weather Space Clim. 2025, **15**, 34
© M. Cobos-Maestre et al., Published by EDP Sciences 2025
https://doi.org/10.1051/swsc/2025029

JSWSC

Available online at:
www.swsc-journal.org

TECHNICAL ARTICLE OPEN ACCESS

BaLNet: A probabilistic Bayesian neural network for solar wind speed forecasting

Mario Cobos-Maestre[1,*], Manuel Flores-Soriano[2], and David F. Barrero[1]

[1] Department of Computer Engineering, Space Weather Research Group, University of Alcalá, Plaza San Diego s/n, 28801 Alcalá de Henares, Madrid, Spain
[2] Department of Physics and Mathematics, Space Weather Research Group, University of Alcalá, Plaza San Diego s/n, 28801 Alcalá de Henares, Madrid, Spain

Received 31 October 2024 / Accepted 30 June 2025

Abstract – Machine Learning has gained favor as a tool for space weather event forecasting over the last few years. In the context of solar wind forecasting, efforts have been primarily focused on forecasting the bulk speed of solar wind. Although steps have been taken in improving the accuracy of forecasts, a common concern from an operational point of view has been determining the reliability of any given prediction. In this work, we present BaLNet, a variational neural network with a normal probability distribution output as a tool for providing operational forecasts of solar wind speed with uncertainty quantification. For a lead time of five days, we find reductions in root mean squared error between 38% and 55%, depending on the test set, when compared to state-of-the-art neural networks. Similarly, for the same lead time, we obtain a correlation coefficient between prediction and observation of 0.89 to 0.94, compared to the correlation coefficient of 0.15 obtained by state-of-the-art models. Cross-validation results confirm robust generalization across temporal variations, maintaining consistent superiority over baseline methods. We also leverage the properties of the Bayesian inference process supporting the resulting model to estimate the degree and sources of uncertainty for any given forecast. We provide 95% confidence interval for each prediction computed from the standard deviation of the forecasted normal distribution. For a lead time of five days, BaLNet's prediction interval coverage probability is of 95% in the test set, evidencing that the target conditional probability distribution is being succesfully extracted. The method presented allows for a full training of the model to converge in a span of 2.17 h in CPU. This process can be further accelerated by using a GPU to reduce the training time to an average of 15 min.

Keywords: Solar wind forecasting / Machine learning / Operational modeling

1 Introduction

The solar wind is a stream of plasma emitted from the upper atmosphere of the Sun. Coronal holes (CHs) – areas of the solar corona characterized by open magnetic field lines – often result in high speed streams (HSS; Verbanac et al., 2011; Hofmeister et al., 2022) within the solar wind. These HSSs are typically associated with an increase in drag in the upper Earth atmosphere, potentially impacting satellite operation (Nwankwo, 2018). In particularly intense cases, they may lead to geomagnetic storms, potentially disrupting communications and the operation of sensitive services (e.g., Vybornov & Sheiner, 2021).

Studies centered on forecasting solar wind typically focus on its bulk speed, V, or its proton density, ρ. Techniques applied range from physical and empirical modeling methods, like the Wang-Sheeley-Arge (WSA) model (Arge et al., 2003) or EUHFORIA (Pomoell & Poedts, 2018), to deep neural networks like the ones proposed by Upendran et al. (2020), Raju & Das (2021), Brown et al. (2022), and Cobos-Maestre et al. (2024).

Upendran et al. (2020) use a deep neural network with Solar Dynamics Observatory (SDO; Pesnell et al., 2012) Atmospheric Imaging Assembly (AIA; Lemen et al., 2012) extreme ultraviolet (EUV) data as input to predict daily averages of V. The model proposed combines convolutional and long short-term memory (LSTM; Hochreiter & Schmidhuber, 1997) layers to produce a forecast with a lead time of up to four days, obtaining a root mean squared error (RMSE) of 82.3 ± 1.7 km s^{-1}.

Raju & Das (2021) extend the results obtained by Upendran et al. (2020) by applying ballistic tracing to match solar wind speed measurements to specific AIA samples on the 19.3 nm

*Corresponding author: mario.cobos@uah.es

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

wavelength. A convolutional neural network (CNN) is applied in this approach. When forcing a fixed lead time of four days, an RMSE of 76.3 ± 1.9 km s^{-1} is reported, improving upon previous results. The ballistic tracing itself is performed by assuming constant velocity matching the measurement taken at the L1 Lagrange point. The authors note that several different solar wind speed measurements can be mapped to a single image sample, and that some image samples may be left unmapped to solar wind measurements. Because arrival time is not accounted for, uncertainties are explicitly stated to surface in the forecasting process. As a result, the authors highlighted that quantifying predictive uncertainty improves reliability of the resulting model.

Brown et al. (2022) further build upon Raju & Das (2021)'s work by applying attention mechanisms to replace the convolutional layers present in the work of Upendran et al. (2020), and by focusing on a specific subset of the image sample around the center of the image. AIA samples on the 21.1 nm wavelength are used, and provided as a sequence. After these modifications, an RMSE of 71.9 ± 0.4 km s^{-1} is reported, offering a new improvement over previous results.

Most recently, we explored the possibility of applying a fully autoregressive scheme based on the application of self-attention (Cobos-Maestre et al., 2024). We named this model the Solar Wind Attention Network (SWAN), and we found that, by using the persistence of CHs over several Carrington rotations (Cranmer, 2009), we could reduce computational complexity. Notably, for aggregations of one day, training time was reduced to 45 s while using CPU. Without using solar disk information, we obtained an improvement over results previously reported in the literature, with an RMSE of 68.8 ± 1.1 km s^{-1} for a lead time of three days. One of the primary open questions identified in our study was whether incorporating solar disk information could yield a meaningful enough performance boost to offset the added computational cost.

Out of all Machine Learning (ML) works discussed, only Raju & Das (2021) quantify the uncertainty of their model. They do so by applying Monte Carlo dropout (Gal & Ghahramani, 2016) over the parameters of the model, and sampling repeatedly to obtain a measure of stability in the output. The uncertainty quantified in this method corresponds exclusively to that related to the model and the knowledge it has extracted, serving as a quantification of reliability.

The need to ascertain the reliability of any given prediction when applying ML models in Space Weather is discussed at length by Camporeale (2019). Indeed, strategies for uncertainty estimation on this field are later provided in Camporeale et al. (2019), with an emphasis on deferring the prediction of uncertainty to a second, auxiliary model tasked exclusively with predicting the standard deviation σ of a conditional probability distribution. Under this approach, the output of the primary model is treated as the mean of a normal conditional probability distribution function, and the combination of both models yields a measure of uncertainty.

More generally, uncertainty quantification of neural networks is usually performed using ensembles (Rahaman & Thiery, 2021) or Monte Carlo dropout (Gal & Ghahramani, 2016). Both approaches aim to treat the generated model as a conditional probability distribution that can be sampled repeatedly. From the range of sampled values, it becomes possible to extract a measure of the uncertainty associated with any given prediction. However, the adequacy of Monte Carlo dropout as a general method for uncertainty quantification has been brought into question on the grounds of it arguably not being properly Bayesian (Le Folgoc et al., 2021).

An alternative solution is the usage of probabilistic neural networks (PNN; Zaknich, 1998). These modified neural networks, when applied to regression, replace the traditional pointwise output of the neural network with a probability density function. The goal is, then, for this probability density function to estimate the conditional probability of the target output for any given input. To ensure applicability of common gradient descent methods, the so called *reparameterization trick* (Jankowiak & Obermeyer, 2018) is often used. The regression model predicts, then, the parameters defining a probability density function of the desired family, enabling training of the model by backpropagating the divergence from the empirical conditional probability distribution.

Bayesian Neural Networks (BNN; MacKay, 1995; Wilson & Izmailov, 2020; Charnock et al., 2022; Magris & Iosifidis, 2023) replace some or all the weights present in a neural network with Bayesian inference processes (Fortuin, 2022). By representing weights in this fashion, predictions made by the neural network become non-deterministic, instead stemming from the sampling of an underlying probability distribution. This provides a direct framework for estimating prediction uncertainty when applied in a PNN to form a probabilistic Bayesian neural network (PBNN). Indeed, the works of Caceres et al. (2021) and Le Folgoc et al. (2021) experimentally shows that PBNNs offer better reliability in uncertainty estimation than techniques like Monte Carlo dropout.

In this paper, we present BaLNet, a PBNN for solar wind speed forecasting that relies on an initial LSTM layer to extract time series features. We use a combination of solar wind and CH measurements as input for the network, and produce a normal distribution of values as the output. By leveraging the variational nature of the model, we show that it is possible to quantify uncertainty, and assign specific contributions to the input data and to the network's own shortcomings. Furthermore, the architecture is lightweight enough to enable training in a CPU, with an average training time of 2.17 h on it.

The remainder of this paper is structured as follows: Section 2 describes the problem formulation taken and the dataset used. Section 3 describes the BNN proposed, the parameter optimization procedure applied, and an overview of relevant operational considerations. Section 4 describes and discusses the experimental results obtained. Finally, Section 5 provides a summary of the study conducted, as well as conclusions drawn from it.

2 Problem formulation and dataset

2.1 Problem formulation

Following the work of Valdenegro-Toro & Mori (2022), the problem of forecasting V at the L1 point is taken as a regression with uncertainty quantification. To this end, the regression target is an expectation distribution for a dependent variable (Y) given a set of independent variables (X) that can be estimated as

$$P(Y|X = x) \sim \mathcal{N}(\mu_\beta(x), \sigma_\gamma^2(x)), \tag{1}$$

where $\mathcal{N}$ denotes a normal probability distribution with mean defined by a parametric function μ_β and variance defined by a second parametric function σ_γ^2. β and γ refer to the parameters of each function. Because the model is a BNN, β and γ are stochastic functions of X generated through a Bayesian inference process. Thus, the predicted V will be presented as a normal distribution denoting the probability of each possible value.

2.2 Dataset description

The dataset used is comprised of solar wind observations and coronal hole measurements. Solar wind observations are taken from the OMNI 1 h resolution level 2 dataset (Mathews & Towheed, 1995). Solar wind observations in the dataset are comprised of tabular measurements made by the Advanced Composition Explorer (ACE; Stone et al., 1998), Solar wind electron proton alpha monitor (SWEPAM; McComas et al., 1998) and magnetic fields experiment (MAG; Smith et al., 1998) instruments. The measured bulk speed, V, plasma temperature, T, proton density, ρ, and the magnitude of the magnetic field vector, $|B|$ are used as inputs.

To characterize CHs, we use their area, location and average magnetic flux. These measurements are computed from SDO AIA 19.3 nm level 1 image samples and Helioseismic and Magnetic Imager (HMI; Schou et al., 2012) level 1 magnetogram samples. These data have minimal processing and are available in almost real time. As a result, the model is forced to learn how to perform in the presence of corrupted samples that might lead to misidentified or missing CHs. Identification and feature extraction are performed algorithmically, as presented in the next subsection, with a daily sampling rate.

The combined dataset encompasses from 2012-09-26 6:00 to 2021-01-31 23:00, both included, as forecasted dates. A total of 81,210 samples are used for all experiments presented in this work.

2.3 Coronal hole feature extraction

The CH identification process is based on the ASSA algorithm.[1] ASSA is chosen as a basis because it identifies visually dark structures that do not present the magnetic imbalance that defines CHs so that they can be discarded. Our algorithm expands upon this approach by leveraging Gaussian expansion over candidate CH regions to provide closer adherence to CH boundaries. Specifically, it consists of three steps (Cobos Maestre, 2020). First, potential CHs are identified using a thresholding technique based on intensity measurements. Next, a Gaussian expansion is applied to capture transitional regions and merge overlapping candidates into cohesive clusters. Finally, the skewness of the magnetic field distribution in each candidate is evaluated, and those with skewness values below 1 are discarded to remove filaments (Krista & Gallagher, 2009; Boucheron et al., 2016). Figure 1 presents an example of CH identification using this method compared to the original ASSA identification.

Once CHs are identified, three main properties are extracted: area, location, and magnetic flux. The area is measured in arcsec^{-2} using Helioprojective Cartesian coordinates Thompson, 2006). The location is determined using a bounding box defined by its bottom-left and top-right corners in pixel coordinates. The average magnetic flux is calculated by summing the per-pixel magnetic flux values within the CH and dividing by the total area, expressed in G arcsec^{-2}. Including the magnetic flux as a feature results in a reduction of the root mean square error (RMSE) by approximately 10 km s^{-1}. Similarly, removing the area from the input results in an increase in RMSE of $\sim$10 km s^{-1}. Furthermore, using the average magnetic flux rather than the total magnetic flux improves RMSE by an additional 3 km s^{-1}.

To ensure that the entire dataset fits into memory during training, CH features are extracted on a daily basis. During validation, it was verified that increasing the measurement frequency up to once every three hours does not significantly impact numerical results or model behavior, confirming that daily measurements are sufficient.

2.4 Preprocessing

To enable model training, all data must be sampled at the same frequency. Additionally, data cleaning and normalization are essential to ensure model convergence and to prevent invalid features containing non-numerical values from corrupting the graph structure. Since the solar wind data is sampled hourly, the CH measurements – originally recorded at a daily frequency – are resampled to match this rate. This resampling introduces missing values that need to be filled. We initially tested two different methods to handle the missing entries: forward filling and linear interpolation.

In our tests, forward filling increased the required convergence time by approximately 12% without yielding any improvement in numerical results. Because of this, we selected linear interpolation for all subsequent experiments.

Two methods for linear interpolation are explored: one where interpolation is performed first, followed by computing the average absolute magnetic flux of each CH, and another where the average flux is computed first, followed by interpolation. The former is mathematically grounded, since interpolating first reduces the risk of distorting the data by not accounting for potential non-linearities. However, empirical results do not show significant differences in model performance between the two approaches in any of the metrics evaluated. Additionally, interpolating first requires six times more floating point operations, which significantly increases computational cost and scales poorly to large datasets. As results are equivalent, it is decided to perform interpolation last in favor of computational efficiency. To further enhance numerical stability, we standardize all features using the formula:

$$x' = \frac{x - \mu_x}{\sigma_x}, \tag{2}$$

where x is the original feature value, μ_x is the feature's mean, σ_x is its standard deviation, and x' is the standardized value. This ensures that all features have zero mean and unit variance, making the optimization process more stable.

Finally, we split the dataset into training, validation, and testing subsets with a 70%–15%–15% ratio. Table 1 details the exact distribution of samples between these three subsets. The dataset is split sequentially to prevent the risk of information leakage associated with the use of training examples appearing in the time series after the test data. Information

[1] https://spaceweather.kasa.go.kr/assa.

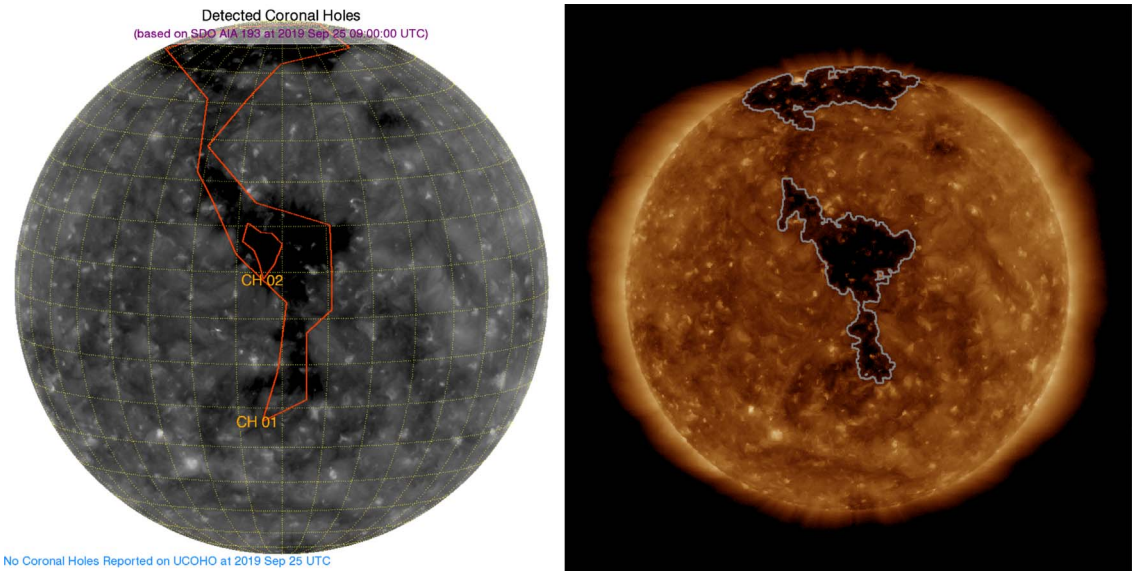

Figure 1. Comparison of CH identification using the ASSA algorithm (left panel) and our algorithm (right panel). The sample shows a large, centered CH during 2019-09-25.

Table 1. Division of the dataset between training, validation, and testing subsets.

Subset	Start date	End date	# samples
Training	2012-09-26 06:00	2019-03-21 09:00	56,812
Validation	2019-03-21 10:00	2020-08-10 16:00	12,199
Testing	2020-08-10 17:00	2021-01-31 23:00	12,199

leakage in this experimental setup could lead to unreasonably low error values by allowing the model to learn from future data, which would not be available in an operational setting (Zhou et al., 2024). Additionally, since the dataset spans different phases of the solar cycle, distributional shifts (Kuznetsov et al., 2015) may distort results obtained through cross-validation (Moreno-Torres et al., 2012).

To further assess the generalizability of the sequential results, five-fold cross-validation is performed, ensuring that the dataset is split sequentially to preserve the causal structure of the time series. Then, each fold is used for testing and for validation once, and the results are collated. The procedure is depicted in Figure 2. To verify the presence and potential impact of information leakage and distributional shift, aggregated and fold-wise results are then compared to those obtained in the sequential setting.This comparison enables testing on the complete solar cycle while controlling for information leakage and distributional shift. Therefore, random sampling is not considered, as it disrupts the sequential dependencies inherent to the time series.

2.5 Input data construction

A fixed-size feature vector is used to represent CH properties for graph construction, limited by available RAM.

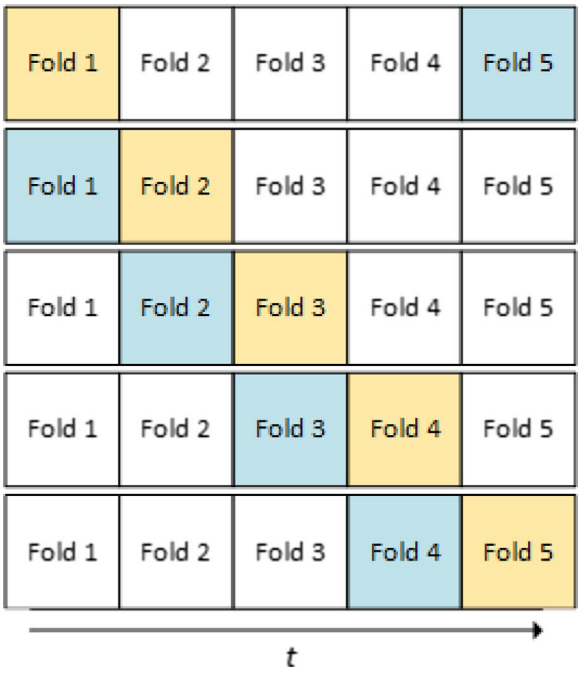

Figure 2. Schematic representation of the 5-fold cross-validation procedure. The time series is split into sequential folds to preserve causality. For each step of the experiment, the fold marked in orange is used as the test set, and the fold marked in blue is used as the validation set. The remainder of the data is used for training.

The CHs are ordered from left to right, corresponding to the direction of solar rotation, and CH information is added following that order until the feature vector is full. This ordering enables the model to always consider CHs that will impact Earth later, allowing for forecasts with a lead time of up to five days. Moreover, when used as a time series, this approach allows BaLNet to account for CHs that may no longer fit into

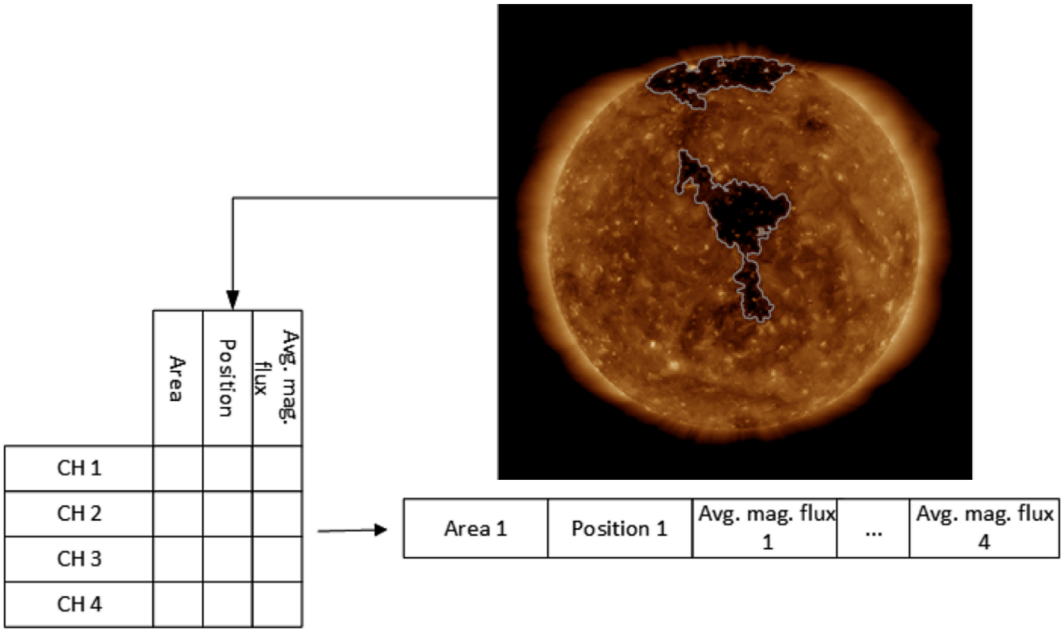

Figure 3. Schematic representation of the CH information vectorization process. First, CHs are identified on available AIA and HMI samples. Then, measurements are taken and stored in tabular form. Finally, CHs are ordered from left to right, and the first four are taken and linearized into a single vector.

the feature vector but were present in previous timesteps. Figure 3 presents a schematic representation of the vectorization process.

Roughly 50% of samples in the dataset include between one and four CHs, while approximately 0.5% of samples contain more than four. An increase in performance is observed when using up to four CHs, with a 14% improvement on all metrics when using four instead of three. However, increasing the number of CHs beyond four leads to a decrease in performance. Using five CHs results in a 2% degradation in performance, along with a 38% increase in training time and a 41% increase in memory requirements. Similarly, using six CHs further degrades performance by 3%, while training time increases by 75% and memory consumption rises by 65%. This trend suggests that using more than four CHs introduces unnecessary complexity. Although the additional CHs may provide useful information, the scarcity of samples with more than four CHs means that BaLNet cannot effectively learn effects on V because of the associated feature sparsity (Petrini et al., 2022). As a result, the model is unable to capitalize on the potential value of this additional information, leading to a performance decline and increased computational costs.

A fixed window size $W = 5$ is used to represent the time series input, as it yielded the best numerical results during preliminary testing. The lead time, denoted as H, specifies the forecast horizon and is set as a parameter for the experiments. We conduct experiments for $H \in \{1, 3, 5\}$ days. Because hyperparameter optimization is performed for a lead time of five days, and no projection correction is applied to CH information, we assess whether decoupling the offset of CH and solar wind data improves performance at shorter lead times. To this end, we introduce a hyperparameter H_{CH}, which allows independent shifting of CH features relative to solar wind targets. For a lead time of one day, we test $H_{\mathrm{CH}} \in \{1, 3, 5\}$, and for a lead time of three days, we explore $H_{\mathrm{CH}} \in \{3, 5\}$. Since the effective lead time is given by $\min(H, H_{\mathrm{CH}})$, this selection ensures that all CH features used in the input would have been observable at the time the forecast is made, preserving operational validity. To capture the persistence of CHs over multiple Carrington rotations, an additional segment of the time series with the same window size W is included as input, positioned C days before the target prediction time. The parameter C serves as an approximation of a Carrington rotation period and is treated as a hyperparameter in the model. For each experiment, we explore values of C ranging from 24.26 to 30.26 days, using a daily step size. Figure 4 provides a schematic illustration of this temporal alignment procedure.

3 The BaLNet model

3.1 Model description

BaLNet is comprised of three main layers, not counting the input: an LSTM layer, a variational dense layer, and a traditional dense layer. This architecture compresses the input time series into a vector representation, which is then used to generate a variational encoding to extract probability parameters from. Dense layers operate by directly applying tensor operations. Given an input vector $\mathbf{x}$, a weights matrix A, a bias vector $\mathbf{b}$, and an activation function θ, the output of a dense layer $d(\mathbf{x})$ is computed as

$$d(\mathbf{x}) = \theta\left(A \cdot \mathbf{x}^{\mathrm{T}} + \mathbf{b}\right). \tag{3}$$

A variational dense layer builds on the general idea of traditional dense layers by introducing stochasticity. The weight

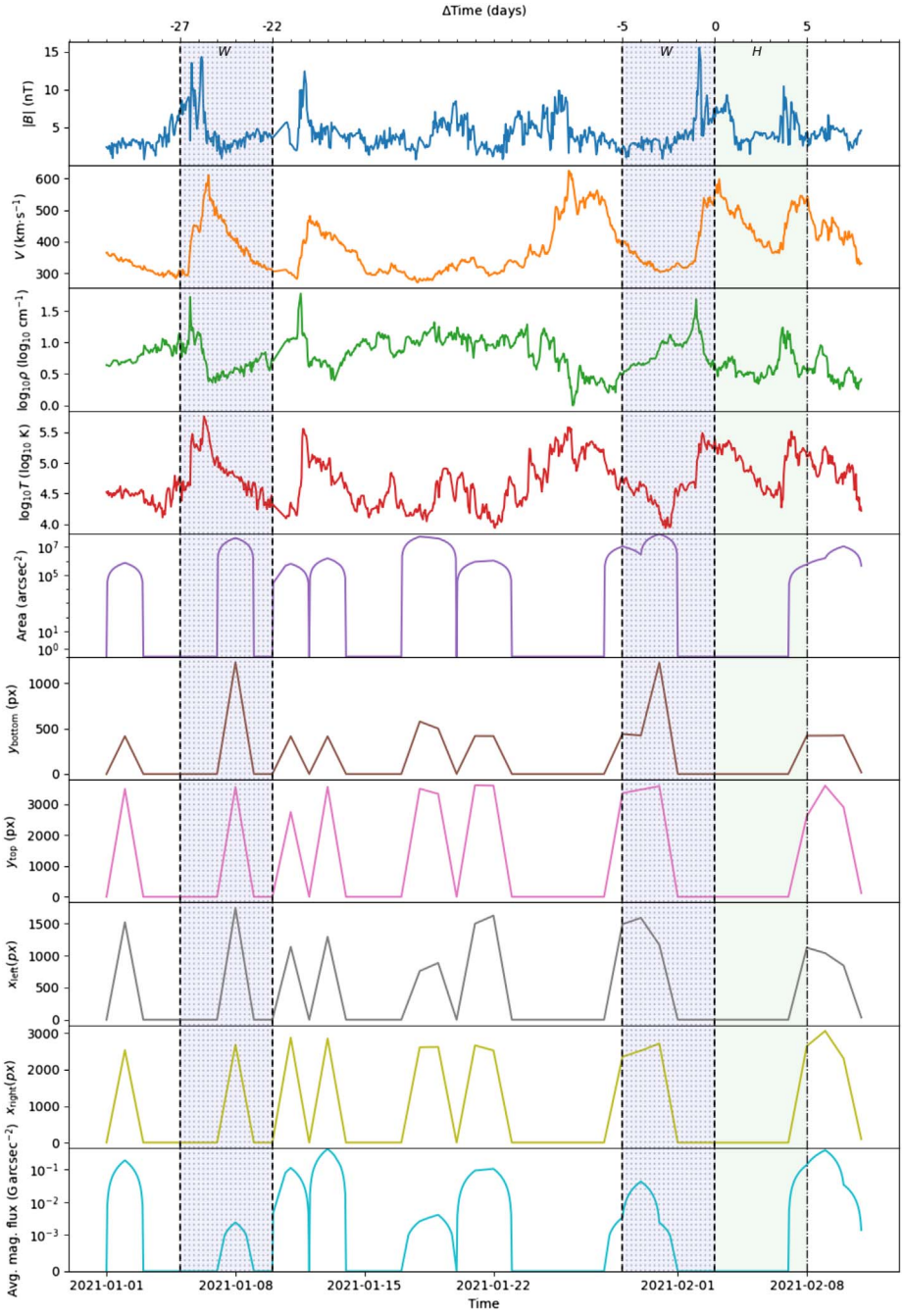

Figure 4. Schematic representation of the temporal alignment scheme. Two time windows of size W are selected and treated as parallel streams of features. The first time window is offset C days from the target forecast date. The second time window is offset some time H with respect to the forecasted time window. In this visual example, a single set of CH measurements is displayed in the interest of brevity, with $W = 5$, $H = 5$, and $C = 27.26$.

matrix is replaced with a Bayesian inference process that generates a multivariate distribution representing the probability of each set of weights. This distribution is generated on a per-inference basis, and sampled to generate the realized weights applied as part of the classic forward propagation. The Bayesian inference process is conditioned on the minimization of the Kullback-Leibler divergence (Kullback & Leibler, 1951) with respect to the empirical conditional probability distribution estimated during backpropagation via Monte Carlo sampling.

Table 2. Weight, mean value (μ) and standard deviation (σ) of V_{t_0} for each multivariate normal used in the Gaussian mixture prior. The rest of the variates, totalling 56, are omitted in the interest of brevity.

Range	Weight (%)	μ (km s^{-1})	σ (km s^{-1})
$V_{t_0} < 400$	53.97	346.0	31.4
$400 \leq V_{t_0} < 600$	46.03	470.4	53.2
$600 \leq V_{t_0} < 700$	5.07	638.9	27.1
$700 \leq V_{t_0} < 800$	0.57	723.6	20.3
$800 \leq V_{t_0}$	0.01	824.5	23.2

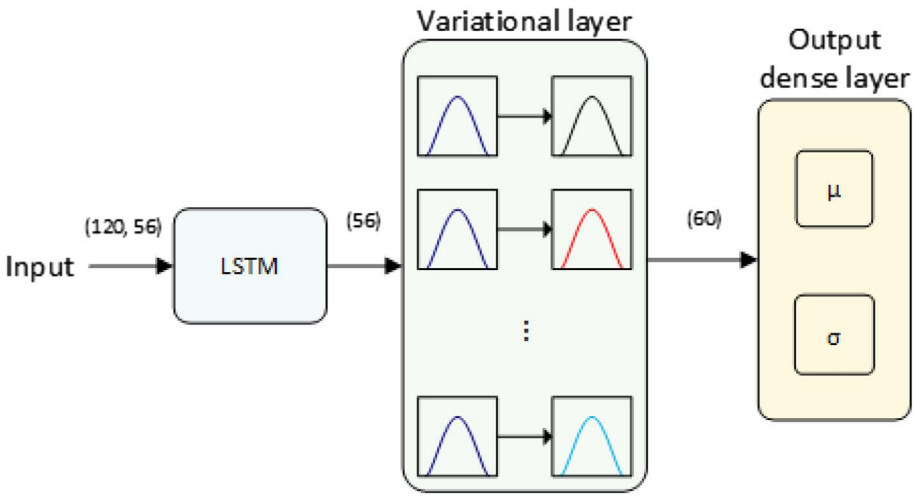

Figure 5. Summary of the proposed architecture. The weights of the variational layer are structured as a Bayesian inference process that generates a normal distribution for each weight. The resulting distribution is then sampled each time the model is run, leading to stochastic outputs.

To construct the prior distribution of the variational dense layer, we assume a mixture of multivariate normal probability distributions. For this purpose, the training dataset is split by the value of V observed at forecast time, t_0, denoted V_{t_0}. We generate five subsets conditioned to V_{t_0} from the training dataset: $V_{t_0} < 400$, $400 \leq V_{t_0} < 600$, $600 \leq V_{t_0} < 700$, $700 \leq V_{t_0} < 800$, and $800 \leq V_{t_0}$; with all speed values expressed in km s^{-1}. For each subset, the mean and standard deviation of each time aligned input feature are computed to configure the corresponding multivariate normal. The mixture weight of each multivariate normal in the final distribution is taken as the proportion of the training dataset represented in it. Values are normalized prior to computing the corresponding statistics. Table 2 presents the weight of each multivariate normal, as well as the corresponding mean, μ, and standard deviation, σ, for V_{t_0}. Because the total number of variates under consideration is 56, and the prior serves only as an initial inductive state, the rest are omitted in the interest of brevity. The values of V_{t_0} are presented in a denormalized form to provide an intuition of the value ranges under consideration for each component.

The posterior distribution is a multivariate normal, with each covariate representing a weight of the dense layer. The activation function chosen is the hyperbolic tangent, tan$h$, because it reseults in consistent convergence during the initial exploration performed. The layer is comprised of 60 neurons. LSTM units are made up of several cells that work in tandem. Each cell has a hidden state path and an input path. The hidden state modulates the processing of inputs, and the result is passed through a forget gate to generate the hidden state for the next cell in the unit. This mechanism enables LSTM layers to effectively capture local temporal relationships and process time series data.

Because of the chosen prior structure, the number of LSTM cells is equal to the number of input variables. The LSTM layer's role is to compress temporal information on a per-input basis, ensuring accurate input for the prior function while encapsulating relevant input features. Again, the tan$h$ activation function is used because only logistic functions have demonstrated consistent convergence.

Finally, the output dense layer is comprised of two independent neurons with linear activation functions. The first neuron represents μ, and the second neuron represents σ. The output is encapsulated into an independent normal distribution to facilitate fitting and usage.

Figure 5 displays a summary of the complete model structure used. This architecture uses a total of 5,878,764 parameters because of the Bayesian inference process applied.

3.2 Optimization procedure

Hyperparameter optimization for the model architecture is focused on a five day lead time. This prioritization is motivated by the goal of extending the predictive range, as well as by the full hyperparameter optimization procedure requiring over 48 h of compute time. To further explore the impact of hyperparameter selection across different lead times, the value of C is systematically evaluated for one day, three day, and five day forecasts.

Because the model's output is a probability density function, traditional metrics such as mean squared error (MSE) and mean absolute error (MAE) are ill-suited for the training task. Instead, the Kullback-Leibler divergence with respect to the empirical conditional probability distribution is minimized.

Direct optimization of the Kullback-Leibler divergence is computationally expensive, and renders the problem practically untractable when applied to the complete model (Blei et al., 2017). Instead, its *Evidence Lower Bound* (ELBO; Duan et al., 2017) is optimized. To this end the combined negative log-likelihood of each batch given the model's current weights is minimized (Oleksiienko et al., 2023).

Optimization is performed using the RMSProp algorithm (Reddy et al., 2018), which is well-suited for non-stationary learning tasks due to its adaptive learning rate properties. To ensure stable training and prevent issues related to exploding gradients, gradient clipping is applied with a maximum norm

of 1.0 (Ramaswamy, 2023). This technique helps control the magnitude of parameter updates, maintaining the stability and convergence of the training process. Early stopping and learning rate scheduling strategies are implemented to prevent overfitting.

3.3 Validation procedure

To validate model results, the model is trained for each combination of C and H seven times to limit the time dedicated to training. A total of 49 training runs are executed, with an average training time in CPU of 2.17 h. The CPU used is an Intel Xeon with a clock speed of 2.20 GHz. All processes are run single threaded and with a single worker. The resulting model's predicted mean is characterized in terms of the following metrics. For the following formulas, Y denotes the observation, $\hat{Y}$ denotes the model's predicted mean, $\bar{Y}$ denotes the mean value of the observation, N refers to the number of elements in Y and $\hat{Y}$, and y_i and $\hat{y}_i$ are elements of Y and $\hat{Y}$, respectively:

- Root mean squared error (RMSE), to validate the average error obtained by the model, weighted proportional to the magnitude of the error. It is defined as

$$\mathrm{RMSE}(Y,\hat{Y}) = \sqrt{\frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}{N}}. \tag{4}$$

- Mean absolute error (MAE), to validate the average error obtained by the model without weighting. It is defined as

$$\mathrm{MAE}(Y,\hat{Y}) = \frac{\sum_{i=1}^{N}|y_i-\hat{y}_i|}{N}. \tag{5}$$

- Pearson's correlation coefficient between observation and prediction (CC), to verify the degree of trend similarity between model prediction and observation. It is defined as

$$\mathrm{CC}(Y,\hat{Y}) = \frac{\mathrm{cov}(Y,\hat{Y})}{\sigma_Y \sigma_{\hat{Y}}}. \tag{6}$$

- Coefficient of determination (R^2), to measure the degree of improvement obtained compared to simply applying a trivial estimation that always outputs the observed distribution mean. It is defined as

$$R^2(Y,\hat{Y}) = 1 - \frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{N}(y_i-\bar{Y})^2}, \tag{7}$$

  where $\bar{Y}$ denotes the mean value found in Y.
- Dynamic time warping distance (DTW) between observed and predicted time series, computed following the algorithm defined by Berndt & Clifford (1994). After computing the optimal warping path P between Y and $\hat{Y}$, the distance is computed as

$$\mathrm{DTW}(Y,\hat{Y}) = \sum_{(i,j)\in P}|y_i-\hat{y}_j|. \tag{8}$$

To contextualize numerical results of the forecasted mean we benchmark with persistence models, a classic linear regression (Zou & Hastie (2005), and an ElasticNet linear regression. We also train and run a CNN following the design proposed by Raju & Das (2021), and compare with SWAN (Cobos-Maestre et al., 2024) to validate the degree of improvement obtained in comparison with state-of-the-art neural networks. Finally, we retrieve WSA-ENLIL (Arge et al., 2003) runs from the NASA Community Coordinated Modeling Center.

To ensure a fair comparison, all Machine Learning models are trained on the same training set. Evaluation is likewise constrained to the exact same test set to prevent deviations in the ground truth from affecting evaluation results. In the case of the CNN and SWAN, we apply the hyperparameter sets provided in their respective papers to guarantee that the estimator follows exactly the same architecture.

To account for WSA-ENLIL being a physical model that predicts multiple physical parameters, we constrain the comparison to the forecasts performed for V at the L1 Lagrange point with lead times of one, three and five days. Because V is the first physical parameter estimated by WSA, it is not affected by propagated errors in previous prediction steps.

To evaluate the quality of the predicted conditional probability distribution, we compute the Prediction Interval Coverage Probability (PICP; Huang et al., 2023). This is calculated using the number of samples within the prediction interval, denoted as N_{in}, and the total number of samples in the test dataset, N:

$$\mathrm{PICP} = \frac{N_{\mathrm{in}}}{N}. \tag{9}$$

We target a 95% prediction interval for calibration. A PICP value close to 95% indicates that the model is well-calibrated. A PICP below 95% suggests the model underestimates uncertainty, while a value nearing 100% indicates an overestimation of uncertainty. To account for the effects of a limited number of random initializations, we consider a model with PICP between 94% and 96% to be well calibrated.

3.4 Operational considerations

BaLNet is designed to provide operational forecasts. During operation, the variational design of the model needs to be considered. The probability of a set of extreme, outlier weights being selected for every selection tends to zero because of the normal distribution implemented. However, fluctuations should still be accounted for.

To obtain a stable prediction, we leverage the central limit theorem Dudley (1978). As per the work of Valdenegro-Toro & Mori (2022), a larger number of evaluations N yields more precise values. A minimum $N = 100$ is suggested to ensure stable results. We investigate whether a smaller N can be utilized by applying the Jensen-Shannon divergence, D_{JS}, to measure stability similarity (Guzmán-Martínez & Alaiz-Rodríguez, 2011). Specifically, a criterion of $D_{\mathrm{JS}} \leq 0.05$ is established to indicate similar stability between values of N. For $N = 10$ and $N = 100$, we find that $D_{\mathrm{JS}} = 3.44 \times 10^{-3}$, which fulfills the criterion. Therefore, $N = 10$ can be used during the operational implementation of BaLNet without compromising the results.

The main advantage of this architecture is the ability to estimate the influence of aleatoric and epistemic uncertainty in the

Table 3. Comparison of numerical results with a lead time of one day. Models denoted as "pers." refer to persistence models. The best numerical results per metric are bolded.

Model	RMSE (km s^{-1})	MAE (km s^{-1})	CC	R^2	DTW ($\times 10^3$)
One day pers.	69.6	50.9	0.64	0.28	**0.17**
Carrington pers.	102.0	75.1	0.21	−0.56	1.80
Linear	59.4	43.4	0.69	0.47	1.67
ElasticNet	82.3	67.5	0.00	−0.02	7.42
WSA-ENLIL	124.0	97.4	−0.06	−1.31	3.20
CNN	69.7	55.4	−0.09	−0.22	3.11
SWAN	64.5 ± 1.3	46.5 ± 1.1	0.64 ± 0.02	0.37 ± 0.03	1.82 ± 0.09
BaLNet	**54.6 ± 3.3**	**40.8 ± 2.7**	**0.79 ± 0.02**	**0.59 ± 0.05**	2.45 ± 0.21

inference process. Aleatoric uncertainty is attributed to the input data used, while epistemic uncertainty stems from the model. To unravel both uncertainties, we use the approach defined by Valdenegro-Toro & Mori (2022): given a predictive variance σ^2, it can be decomposed as

$$\sigma_i^2(x) = \mathbb{E}_i[\sigma_i^2(x)] + \mathrm{Var}_i[\mu_i(x)],$$

where $\mathbb{E}_i[\sigma_i^2(x)]$ denotes the mean variance across the predicted distributions from several model runs, and $\mathrm{Var}_i[\mu_i(x)]$ denotes the variance of their predicted means.

In this expression, the term $\mathbb{E}_i[\sigma_i^2(x)]$ captures the aleatoric uncertainty of the prediction. The term $\mathrm{Var}_i[\mu_i(x)]$ captures the epistemic uncertainty. In an operational context, this approach provides a measure of reliability to facilitate interpretation of the forecasts obtained (e.g., low reliability when a high speed CME is expected to arrive).

4 Results and discussion

4.1 Model evaluation

Tables 3–5 present the numerical results obtained by each model for forecast lead times of one, three, and five days, respectively. For conciseness, only the best configuration of BaLNet (based on the parameter C) is shown in these tables. The optimal value of C is set to 25.26 days for a one day lead time, and to 27.26 days for three day and five day lead times. A detailed breakdown of the results for all BaLNet configurations is available in Appendix.

Overall, BaLNet performs best for a five day lead time, which aligns with architectural hyperparameters having been optimized for that use case. For this lead time, the minimum and maximum PICP values for the 95% confidence interval are approximately 90% and 95%, respectively. In comparison, the one day lead time shows lower PICP values, ranging from 75% to 83%, while the three day lead time achieves PICP values between 85% and 89%. This behavior is consistent with all other evaluation metrics used.

Across all lead times, BaLNet outperforms all other models on the test dataset in terms of numerical metrics. Specifically, for the five day lead time, BaLNet achieves a 55% reduction in RMSE compared to the CNN, while for the one day lead time, the reduction is 15%. Furthermore, the negative R^2 values found suggest a high relative error in most models' predictions, higher than that obtained using the mean value of V as estimation (Kvålseth, 1985). This does not happen in BaLNet's case, further highlighting its accuracy.

The nearly constant performance observed for both ElasticNet and CNN across all lead times reflects a conservative predictive behavior. This behavior exhibits the typical traits of non-convergence associated with excessive regularization: small fluctuations in output, difficulty in tracking the shape of the observations, and consistence of the models' performance with respect to the training set (Ghojogh & Crowley, 2019; Obi & Jecinta, 2023). Furthermore, since regularization is the only common factor between these models, it further underscores its apparent influence on their performance.

Persistence models demonstrate superior performance in short-term DTW evaluations because they replicate the temporal structure of the time series through direct propagation of past values. As a result, persistence models inherently preserve the temporal alignment of features, making them unsuitable for direct comparisons against methods that generate forecasts based on distinct predictive mechanisms. Excluding persistence models from the evaluation, BaLNet achieves the highest agreement with the observed temporal patterns for three and five days lead times, as reflected by its DTW scores. Furthermore, the best initializations of BaLNet outperform Carrington persistence models for the five days lead time.

For the one day lead time, linear regression and the SWAN model show lower DTW values than BaLNet; however, when considering the complete set of evaluation metrics both linear regression and SWAN exhibit numerically inferior results compared to BaLNet. This discrepancy suggests that the high temporal autocorrelation of V, of 0.73 for a one day lead time, results in these models producing imitative behaviors rather than generalizable forecasts. BaLNet, in contrast, presents a clear alignment in all performance metrics, suggesting consistent, stable learning of predictive behaviors.

Table 6 displays the PICP value obtained for each lead time, expressed as a percentage. For the one and three day lead times, BaLNet underestimates the true uncertainty. In contrast, the five day model accurately captures the uncertainty present in the data. These results are consistent with the numerical evaluation of the predicted means, confirming that BaLNet is most reliable when forecasting with a five day lead time.

Figures 6, 7, and 8 illustrate example forecasts for one, three, and five days lead times, respectively, along with their corresponding uncertainty estimates. The most relevant statistics for each figure are summarized in Table 7. A CME arrival occurs between late 2021-11-03 and 2021-11-05, reaching a

Table 4. Same as Table 3 but for a three days lead time.

Model	RMSE (km s^{-1})	MAE (km s^{-1})	CC	R^2	DTW ($\times 10^3$)
Three days pers.	109.5	86.3	0.11	−0.80	**0.85**
Carrington pers.	102.0	75.1	0.21	−0.56	1.80
Linear	85.1	67.8	0.17	−0.09	2.88
ElasticNet	84.7	68.3	−0.04	−0.08	6.55
WSA-ENLIL	105.4	80.8	0.24	−0.67	3.03
CNN	70.0	56.05	−0.11	−0.27	2.80
SWAN	86.2 ± 1.9	64.2 ± 1.7	0.25 ± 0.03	−0.12 ± 0.05	2.26 ± 0.16
BaLNet	**43.7 ± 2.8**	**32.6 ± 2.3**	**0.86 ± 0.02**	**0.74 ± 0.03**	2.20 ± 0.27

Table 5. Same as Table 3 but for a five days lead time.

Model	RMSE (km s^{-1})	MAE (km s^{-1})	CC	R^2	DTW ($\times 10^3$)
Five days pers.	116.3	91.1	−0.01	−1.03	**1.28**
Carrington pers.	102.0	75.1	0.21	−0.56	1.80
Linear	82.9	65.7	0.22	−0.04	3.02
ElasticNet	84.0	68.07	−0.03	−0.06	6.81
WSA-ENLIL	126.9	99.07	−0.07	−1.41	3.00
CNN	70.1	57.1	−0.08	−0.29	3.13
SWAN	86.6 ± 2.6	65.3 ± 1.5	0.15 ± 0.04	−0.13 ± 0.07	2.47 ± 0.20
BaLNet	**31.3 ± 4.4**	**22.8 ± 3.7**	**0.94 ± 0.01**	**0.87 ± 0.03**	1.96 ± 0.21

peak speed of 770 km s^{-1}. For all three lead times, for target times affected by this arrival, BaLNet's prediction increases in mean value until it reaches roughly 550 km s^{-1}, and then stabilizes around that value. Furthermore, for a five days lead time, punctual decreases under this threshold are accurately captured. Using the CME database provided by Richardson & Cane (2010), BaLNet is observed to consistently produce this behavior for CMEs that exceed 550 km s^{-1} bulk speed. This is particularly notable because BaLNet is not provided any CME-specific data, such as coronograph images or extracted features.

Across all lead times, BaLNet captures the general behaviors of solar wind speed. For the one and three days lead times, it primarily captures these general behaviors, while for the five days lead time, the model also predicts finer-grained detail. For example, in the high speed stream starting on 2021-09-15, it can be seen that BaLNet captures the presence of the initial peak for all three lead times. However, the second and third peaks are only captured for a five days lead time. Furthermore, for a five days lead time, BaLNet effectively distinguishes situations where the dominant source of uncertainty is epistemic, as opposed to aleatoric, suggesting its capability to capture the nature of uncertainty even with longer lead times.

4.2 Cross-validation results

The dataset used for this study spans from 2012-10-01 6:00 to 2022-01-01 00:00, covering most of Solar Cycle 24 and the rising phase of Cycle 25. The sequential evaluation, performed in accordance with best practices for time series regression (Clark et al., 2020), is therefore limited to the early stages of Cycle 25. However, this raises the question of whether the numerical results obtained from such a split are generalizable to other phases of the solar cycle.

Table 6. PICP of BaLNet's 95% CI with respect to lead time.

Lead time	PICP (%)
One day	83 ± 5
Three days	88 ± 3
Five days	95 ± 2

To further assess the generalizability of the numerical results across different phases of the Solar Cycle, including data from both Cycles 24 and 25, we perform five-fold cross-validation. The dataset is split into five equally-sized, sequentially ordered subsets, each of which is used as the test set once. The results are then aggregated so that they can be compared to the sequential approach by treating each fold as analogous to an independent run. This approach allows us to evaluate BaLNet's performance across different Solar Cycle phases, despite the limited data. Additionally, the sequential experiment helps identify whether known pitfalls of applying cross-validation to time series, such as information leakage and distributional shift, are present in this case. Table 8 presents a comparison between the numerical results obtained in the cross-validation experiment and those obtained in the sequential setting. While neither any individual folds nor the aggregated results of the cross-validation procedure exceed the sequential setting, this consistency does not indicate evidence of information leakage. However, interfold fluctuations in the relationship between evaluation metrics suggest the presence of distributional shift. This fluctuation is most evident when comparing RMSE and DTW to R^2: the increase in numerical error for both metrics is accompanied by a larger improvement relative to a trivial estimator than in other folds, indicating a shift in the underlying conditional distribution function. Furthermore, using the Kolmogorov-Smirnov test (Massey, 1951), clear differences

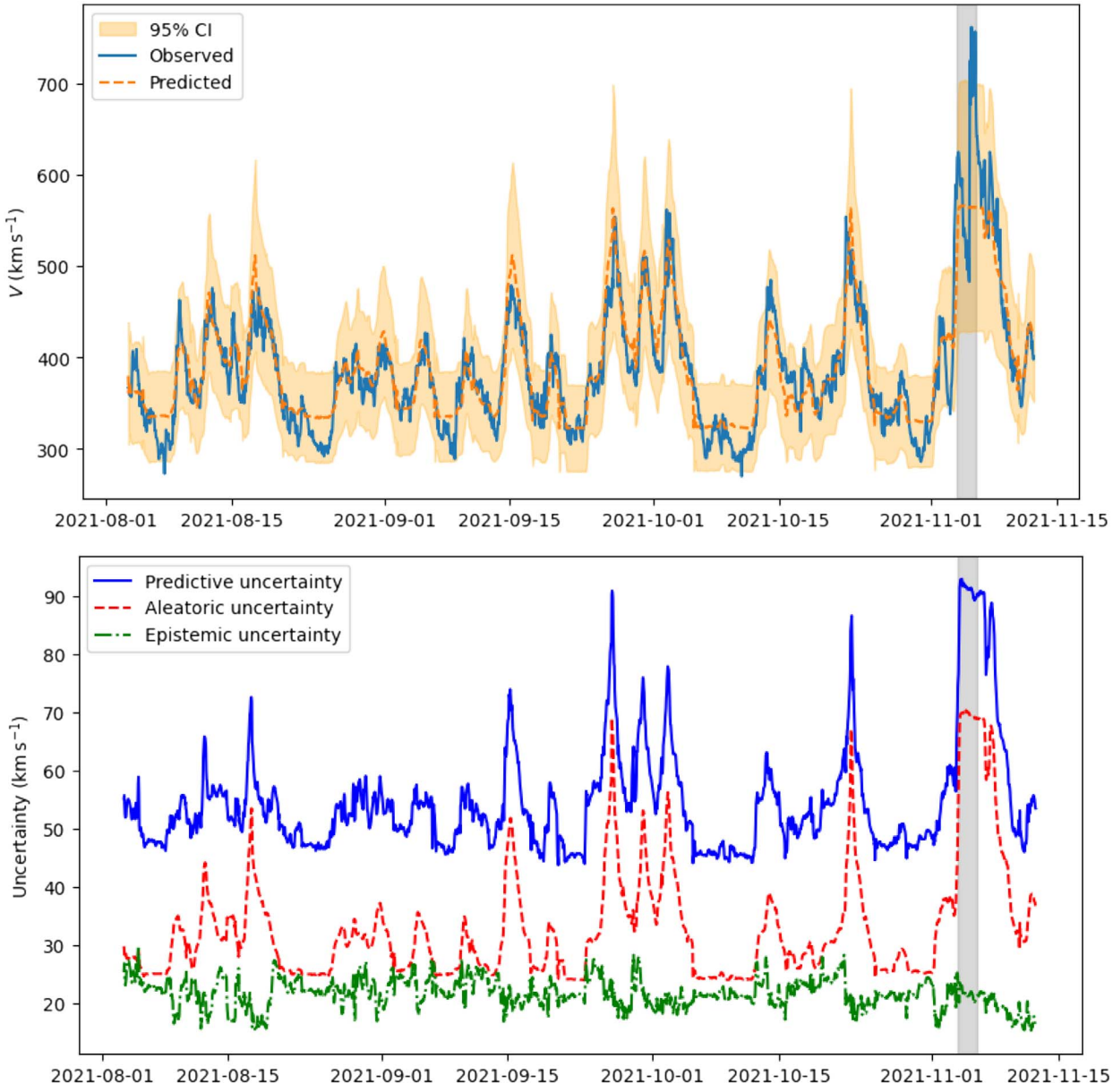

Figure 6. Top panel presents the comparison of forecasted values and observation with a one day lead time. The dashed line corresponds to the mean of the distribution forecasted by the model. The area shaded in orange corresponds to the 95% confidence interval. The bottom panel presents the corresponding uncertainty disentanglement. The area shaded in grey in both panels denotes a CME arrival with a peak speed of 770 km s^{-1}.

are observed between the distributions of metrics obtained using cross-validation and the sequential approach, further reinforcing the presence of distributional shift. The observed shift could be due to a more prominent presence of CMEs during periods of high solar activity. This would make periods around solar minimum ideal to compare models that do not aim to predict CMEs by ensuring minimal impact on the obtained performance metrics.

For all metrics, all folds and the aggregated cross-validation results display numerical improvements when compared to CNN, SWAN, the linear models and the persistence baselines. The highest RMSE obtained represents a reduction by 38% when compared to the CNN's results in the sequential setting, despite the latter being evaluated on a more advantageous subset of the data. The consistent improvements found in the cross-validation procedure suggest that BaLNet is robust to fluctuations in the data, and underline its ability to generalize to different periods of the solar cycle.

Figure 9 displays the forecasts and uncertainty disentanglement produced by the model whose test set encompasses between 2012-10-01 and 2014-08-08, for the period between May 2014 and the end of the test set. Due to the higher mean speed value associated with the maximum of the Solar Cycle (Larrodera & Cid, 2020), the predictive mean stays elevated.

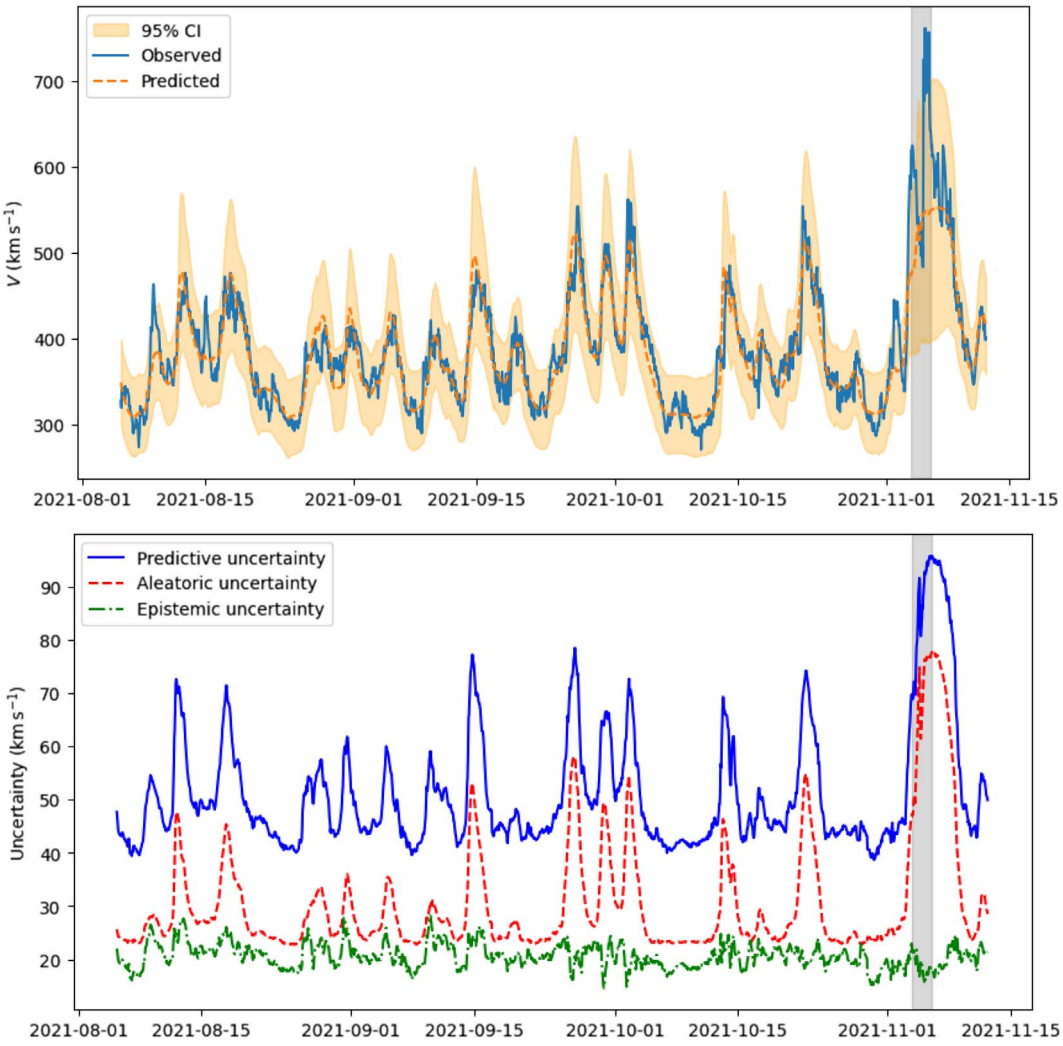

Figure 7. Same as Figure 6, but for a three days lead time.

However, the observed predictive behaviors align with those found in the sequential approach (Fig. 8) when accounting for the observed distributions in each phase, further corroborating BaLNet's stability and its ability to generalize across both minimum and maximum solar conditions. Together with the numerical results, it demonstrates BaLNet's ability to generalize across diverse solar conditions, offering robust predictions even in the presence of solar activity fluctuations.

4.3 Critical analysis of temporal alignment between solar wind data and coronal hole information

Hyperparameter optimization for the previous experiments is performed primarily for a lead time of five days. Furthermore, no projection correction is applied to CH information. It is hypothesized that this should not negatively affect the presence of disk-centered CHs within the input vector for the corresponding lead times. If this hypothesis is wrong, it would lead to the observed reduction in performance for the shorter lead times.

To address this concern, we perform an additional experiment introducing a new hyperparameter, H_{CH}. H_{CH} enables independent offsetting of CH information relative to solar wind data, decoupling their offset to assess whether this improves performance by maximizing the duration for which relevant CHs remain centered on the solar disk within the input window. We evaluate $H_{CH} \in \{1, 3, 5\}$ for a one day lead time and $H_{CH} \in \{3, 5\}$ for a three days lead time. We note that the effective lead time when utilizing this new hyperparameter is $\min(H, H_{CH})$, so the list of values for a lead time of three days is adjusted accordingly. When $H_{CH} = H$, this setting reproduces the previously reported same offset results. Table 9 summarizes the results obtained with each hyperparameter configuration. For a lead time of one day, the time-aligned setting (H_{CH} = H) achieves the best performance in PICP, RMSE, MAE, CC and R^2. H_{CH} = 3 apparent improvement of DTW (∼1.6%) is within the uncertainty bounds of the value found for H_{CH} = 1, and it underperforms in all other measurements. Similarly, for a three days lead time, $H_{CH} = H$ yields the best results for RMSE, MAE, CC, R^2, with a statistically insignificant increase in PICP (∼1.1%) for H_{CH} = 5.

Thus, temporally decoupling CH and solar wind information does not yield any improvements, as observed differences

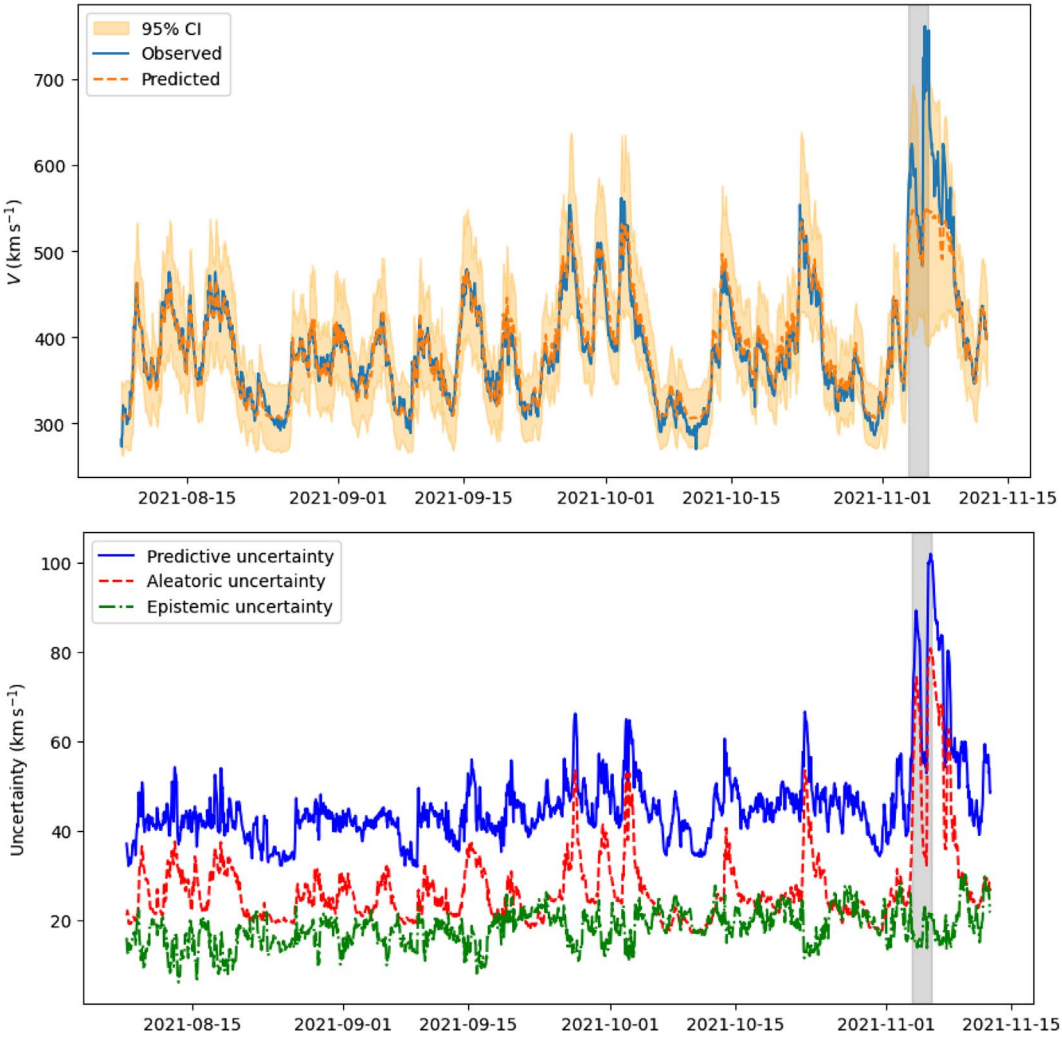

Figure 8. Same as Figure 6, but for a five days lead time.

Table 7. Maximum and minimum predicted $\mu_\beta(x)$, and properties of the distributions of aleatoric and epistemic uncertinaty presented in Figures 6, 7 and 8. All measurements are provided in km s^{-1}.

Lead time	One day (Fig. 6)	Three days (Fig. 7)	Five days (Fig. 8)
Max. $\mu\beta(x)$	572.2	549.4	551.0
Min. $\mu\beta(x)$	315.3	311.1	308.0
Avg. aleatoric	32.6 ± 10.5	31.0 ± 10.8	27.8 ± 9.3
Max. aleatoric	72.2	77.0	77.3
Min. aleatoric	23.0	23.4	20.6
Avg. epistemic	20.0 ± 2.5	17.8 ± 2.1	19.3 ± 2.9
Max. epistemic	28.1	24.2	28.9
Min. epistemic	12.6	11.0	12.6

remain within the uncertainty bounds of the base configuration. This result reinforces that maintaining temporal alignment is preferable within the evaluated conditions, as it minimizes the number of preprocessing operations and ensures full consistency for every lead time.

4.4 Ablation study

To further understand the behavior of the variational dense layer and its impact on model performance, we remove specific components and quantify their impact on the obtained modeling results. Specifically, we reduce the number of components in the prior of the variational layer, we remove the LSTM layer, and we remove the variational layer. Removed layers are replaced with dense layers with the same number of units and activation function.

Table 10 presents a comparison of various performance metrics, including RMSE, CC, DTW, PICP, training time in CPU, and memory usage of the training graph between BaLNet and the ablated models.

The results indicate that removing any components reduces the computational cost of training, at the expense of a reduction in performance. The combination of increased RMSE and reduced PICP suggests that, independently of the element removed, modeling results fail to properly capture the conditional probability distribution of V, with a consistent underestimation of uncertainty. Notably, none of the ablated models are

Table 8. Comparison of results obtained in the cross-validation experiment with results obtained in the sequential experiment for a lead time of five days. Each cross-validation fold is labeled using its start and end dates.

Set	RMSE (km s^{-1})	MAE (km s^{-1})	CC	R^2	DTW ($\times 10^3$)	PICP (%)
2012-10-01 06:00 to 2014-08-08 00:00	35.4	26.0	0.90	0.81	2.86	88
2014-08-08 01:00 to 2016-06-13 18:00	41.4	28.4	0.89	0.78	3.77	90
2016-06-13 19:00 to 2018-04-20 12:00	43.8	28.9	0.94	0.84	4.60	95
2018-04-20 13:00 to 2020-02-25 06:00	33.6	24.1	0.92	0.83	2.98	94
2020-02-25 07:00 to 2022-01-01 00:00	32.4	24.8	0.92	0.84	2.52	95
Cross-validation	37.3 ± 5.0	26.4 ± 2.1	0.91 ± 0.02	0.82 ± 0.02	3.34 ± 0.84	92 ± 3
Sequential	31.3 ± 4.4	22.8 ± 3.7	0.94 ± 0.01	0.87 ± 0.03	1.96 ± 0.21	95 ± 2

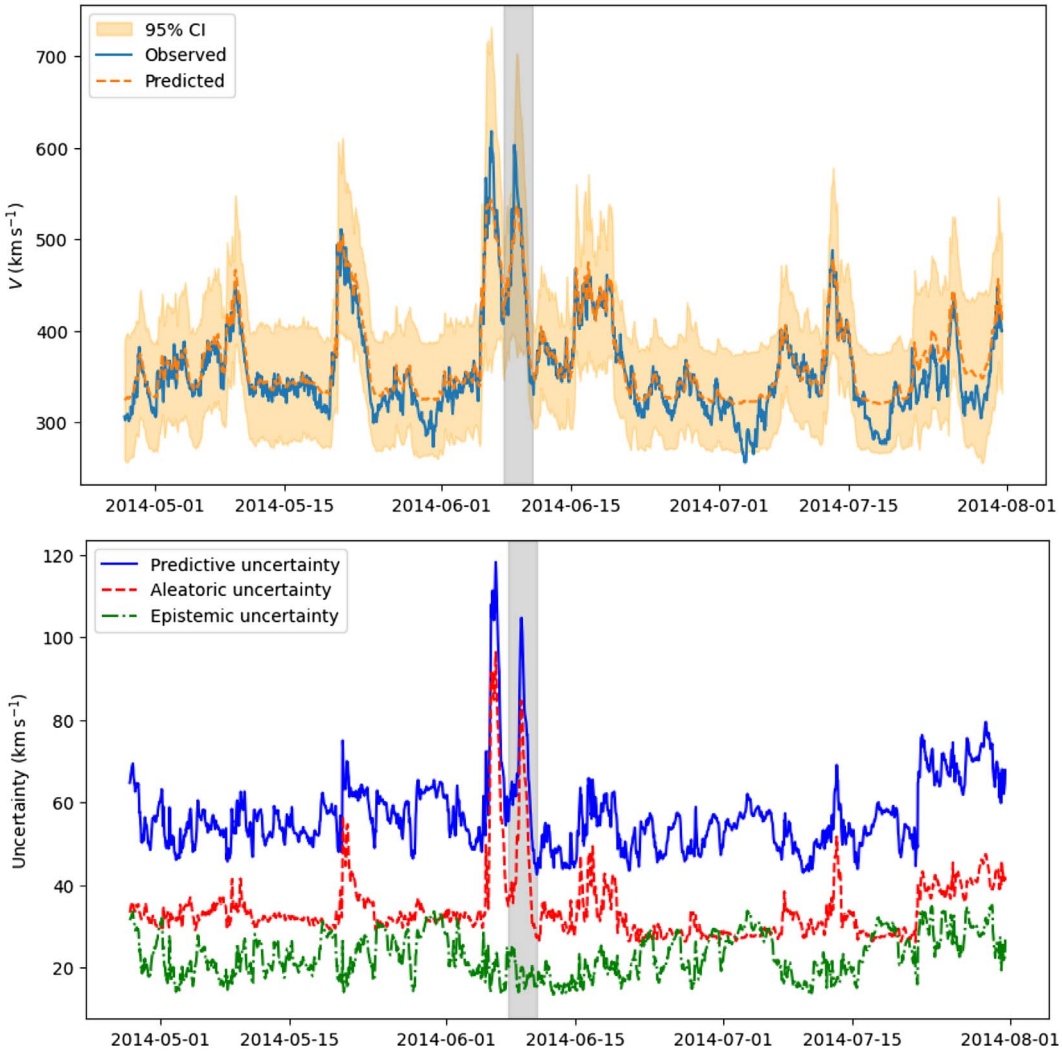

Figure 9. Top panel shows the comparison of forecasted value and observation with a five days lead time for the period between May and early August 2014. The area shaded in orange represents the 95% confidence interval. The bottom panel presents the corresponding uncertainty disentanglement. The area shaded in grey in both panels denotes a CME arrival with peak speed of 610 km s^{-1}.

well calibrated as per our criterion, as their PICP is strictly less than 94%. Furthermore, utilization of GPU reduces the average training time of the full model to an average of 15 min. Thus, the main disadavantage of applying the complete model is minimized.

5 Conclusions

In this paper, we have presented BaLNet, a variational Bayesian neural network architecture for probabilistic solar wind speed forecasting. We optimize model hyperparameters

Table 9. Comparison of performance results obtained using different values of H_{CH}. The best numerical results for each combination of H and measurement are bolded.

H	H_{CH}	PICP (%)	RMSE (km s^{-1})	MAE (km s^{-1})	CC	R^2	DTW ($\times 10^3$)
1	1	**83 ± 5**	**54.6 ± 3.3**	**40.8 ± 2.7**	**0.79 ± 0.02**	**0.59 ± 0.05**	2.45 ± 0.21
	3	78 ± 3	59.6 ± 3.1	44.1 ± 2.5	0.76 ± 0.02	0.51 ± 0.05	**2.41 ± 0.13**
	5	80 ± 2	57.2 ± 2.5	42.4 ± 3.1	0.77 ± 0.02	0.55 ± 0.04	2.57 ± 0.08
2	3	88 ± 3	**43.7 ± 2.8**	**32.6 ± 2.6**	**0.86 ± 0.02**	**0.74 ± 0.03**	**2.20 ± 0.27**
	5	**89 ± 3**	46.4 ± 1.7	34.5 ± 1.4	0.84 ± 0.02	0.70 ± 0.03	2.48 ± 0.36

Table 10. Summary of ablation study results. Models are labeled using the component of BaLNet that is ablated. MAE and R^2 display the same tendencies as RMSE and CC, respectively. The best values for each measurement evaluated are bolded.

Model	RMSE (km s^{-1})	CC	DTW ($\times 10^3$)	PICP (%)	Training time (h)	Graph size (GB)
BaLNet	**31.3**	**0.94**	**1.96**	**95**	2.17	4.25
Reduced prior	43.7	0.86	3.06	82	1.56	4.00
No LSTM	37.6	0.89	2.06	92	1.31	4.50
No variational	41.8	0.88	1.96	89	**0.05**	**3.40**

for a lead time of five days, and evaluate it for lead times of one, three and five days, and find consistent improvements when compared to state-of-the-art solar wind speed forecasting models. Notably, error is reduced by between 38% and 55%, depending on the test set, for a lead time of five days, suggesting a marked increase in forecasting ability. While cross-validation results show slightly higher RMSE and variability due to temporal distributional shifts, BaLNet consistently outperforms benchmark models across all valuation metrics, affirming its generalizability and reliability for operational use. Furthermore, BaLNet provides uncertainty estimation, and can be inspected to disentangle source of uncertainty. This enables users to gain a better understanding of the reliability of any given forecast.

We also evaluated the accuracy of the predicted conditional probability distribution by contrasting the derived 95% confidence interval with the prediction interval coverage probability (PICP) of the computed range. For a lead time of five days, a PICP of between 88% and 95% is obtained, suggesting a strong agreement with the empirical conditional probability distribution function underlying the generative process. The confidence interval provided is observed to lose accuracy when the lead time is reduced. This behavior suggests an excess of confidence by the model in such cases, leading to smaller variances than that observed in the empirical conditional probability distribution.

A key advantage of BaLNet is its lightweight architecture, which enables efficient training on a CPU, with an average training time of 2.17 h. This miniaturization is facilitated by the absence of direct interaction with image data, as the model relies on pre-extracted features from solar observations. While BaLNet includes information from the solar disk, it does so using lower-dimensional feature vectors, reducing the complexity of the model.

We found that removal of the variational component from the architecture results in a loss of predictive power, in exchange for a significant reduction in computational demands. Numerical results were still an improvement over those obtained with state-of-the-art models found in the literature, further highlighting the effectiveness of the Bayesian approach in solar wind forecasting.

In conclusion, BaLNet demonstrates that a Bayesian framework, combined with a lightweight architecture and carefully selected features, can significantly improve solar wind speed forecasting while maintaining computational efficiency. The model's ability to provide well-calibrated uncertainty estimates and to handle high-dimensional input data without directly processing complex images opens new avenues for operational forecasting.

In future, the BaLNet model will be validated in an operational environment to ensure that the results presented in this paper hold during the remainder of the current solar cycle. Additionally, to facilitate the usage of BaLNet as a source of information for models predicting effects on Earth (e.g., Collado-Villaverde et al., 2024), the temporal resolution will be enhanced by reducing the target timestep to 5 min, while maintaining the current five days lead time. Moreover, in this same direction, BaLNet's applicability to forecasting additional solar wind parameters, such as density ρ and T, will be investigated.

Acknowledgments

The authors are grateful to all the members of the Space Weather community making data readily available for usage. We acknowledge the use of ACE SWEPAM and MAG data retrieved from the OMNIWeb database (http://omniweb.gsfc.nasa.gov/). We also acknowledge the use of SDO AIA and HMI samples retrieved from the Joint Science Operations Center (http://jsoc.stanford.edu/). WSA-ENLIL runs were retrieved from the NASA Community Coordinated Modeling Center Runs-on-Request service (https://ccmc.gsfc.nasa.gov/tools/runs-on-request/). The research presented in this paper utilizes the following Python libraries: Astropy (Price-Whelan et al., 2018), Sunpy (The SunPy Community et al., 2020), Numpy (Harris et al., 2020), SciPy (Virtanen et al., 2020), Pandas (McKinney, 2010; jbrockmendel et al., 2020), Tensorflow (Abadi et al., 2016), Keras[2],

[2] https://keras.io.

and Tensorflow Probability (Dillon et al., 2017). The editor thanks two anonymous reviewers for their assistance in evaluating this paper.

Data availability statement

The data and code required to train BaLNet is available in the following GitHub repository: https://github.com/R012/BaLNet. Level 2 solar wind data is provided along with all preprocessing steps required for reproducibility. Coronal hole data measurements are provided after discarding samples marked as corrupt by their FITS headers.

References

Abadi, M, Barham P, Chen J, Chen Z, Davis A, et al. 2016. Tensorflow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), November 2–4, 2016, Savannah, USA, GA.

Arge, CN, Odstrcil D, Pizzo VJ, Mayer LR. 2003. Improved method for specifying solar wind speed near the sun. *AIP Conf Proc* **679 (1)**: 190–193. https://doi.org/10.1063/1.1618574.

Berndt, DJ, Clifford J. 1994. Using dynamic time warping to find patterns in time series. In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, AAAIWS'94*. AAAI Press, pp. 359–370. https://doi.org/10.5555/3000850.3000887.

Blei, DM, Kucukelbir A, McAuliffe JD. 2017. Variational inference: A review for statisticians. *J Am Stat Assoc* **112 (518)**: 859–877. https://doi.org/10.1080/01621459.2017.1285773.

Boucheron, LE, Valluri M, McAteer RTJ. 2016. Segmentation of coronal holes using active contours without edges. *Sol Phys* **291 (8)**: 2353–2372. https://doi.org/10.1007/s11207-016-0985-z.

Brown, EJE, Svoboda F, Meredith NP, Lane N, Horne RB. 2022. Attention-based machine vision models and techniques for solar wind speed forecasting using solar EUV images. *Space Weather* **20 (3)**: e2021SW002976. https://doi.org/10.1029/2021SW002976.

Caceres, J, Gonzalez D, Zhou T, Droguett EL. 2021. A probabilistic Bayesian recurrent neural network for remaining useful life prognostics considering epistemic and aleatory uncertainties. *Struct Control Health Monit* **28 (10)**: e2811. https://doi.org/10.1002/stc.2811.

Camporeale, E. 2019. The challenge of machine learning in space weather: nowcasting and forecasting. *Space Weather* **17 (8)**: 1166–1207. https://doi.org/10.1029/2018SW002061.

Camporeale, E, Chu X, Agapitov OV, Bortnik J. 2019. On the generation of probabilistic forecasts from deterministic models. *Space Weather* **17 (3)**: 455–475. https://doi.org/10.1029/2018SW002026.

Charnock, T, Perreault-Levasseur L, Lanusse F. 2022. Bayesian neural networks. In: *Artificial Intelligence for High Energy Physics*, World Scientific, pp. 663–713. https://doi.org/10.1142/9789811234033_0018.

Clark, S, Hyndman RJ, Pagendam D, Ryan LM. 2020. Modern strategies for time series regression. *Int Stat Rev* **88 (S1)**: S179–S204. https://doi.org/10.1111/insr.12432.

Cobos Maestre, M. 2020. Coronal hole tracking using Big Data and Computer Vision, *Master's Thesis*, Universidad de Alcalá, Alcalá de Henares, Madrid, Spain.

Cobos-Maestre, M, Flores-Soriano M, Barrero DF. 2024. SWAN: A multihead autoregressive attention model for solar wind speed forecasting. *Expert Syst With Appl* **252**: 124128. https://doi.org/10.1016/j.eswa.2024.124128.

Collado-Villaverde A, Muñoz P, Cid C. 2024. Operational SYM-H forecasting with confidence intervals using deep neural networks. *Space Weather* **22 (10)**: 1–2.

Cranmer, SR. 2009. Coronal holes. *Liv Rev Sol Phys* **6 (1)**: 3. https://doi.org/10.12942/lrsp-2009-3.

Dillon, JV, Langmore I, Tran D, Brevdo E, Vasudevan S, Moore D, Patton B, Alemi A, Hoffman M, SaurousRA. 2017. *Tensorflow distributions*. arXiv preprint https://doi.org/10.48550/arXiv.1711.10604.

Duan, H, Yang L, Fang J, Li H. 2017. Fast inverse-free sparse Bayesian learning via relaxed evidence lower bound maximization. *IEEE Signal Process Lett* **24 (6)**: 774–778. https://doi.org/10.1109/LSP.2017.2692217.

Dudley, RM. 1978. Central limit theorems for empirical measures. *Ann Probability* **6 (6)**: 899–929. http://www.jstor.org/stable/2243028.

Fortuin, V. 2022. Priors in Bayesian deep learning: A review. *Int Stat Rev* **90 (3)**: 563–591. https://doi.org/10.1111/insr.12502.

Gal, Y, Ghahramani Z. 2016. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: *Proceedings of The 33rd International Conference on Machine Learning*, MF, Balcan, Weinberger KQ (Eds.), vol. 48 of Proceedings of Machine Learning Research. PMLR, New York, NY, USA, pp. 1050–1059. https://proceedings.mlr.press/v48/gal16.html.

Ghojogh, B, Crowley M. 2019. *The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial*. https://doi.org/10.48550/arXiv.1905.12787.

Guzmán-Martínez, R, Alaiz-Rodríguez R. 2011. Feature selection stability assessment based on the Jensen-Shannon Divergence. In : *Machine Learning and Knowledge Discovery in Databases*, D, Gunopulos, Hofmann T, Malerba D, Vazirgiannis M, (Eds.) Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 597–612. ISBN 978-3-642-23780-5.

Harris, CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, et al. 2020. Array programming with NumPy. *Nature* **585 (7825)**: 357–362. https://doi.org/10.1038/s41586-020-2649-2.

Hochreiter, S, Schmidhuber J. 1997. Long short-term memory. *Neural Comput* **9 (8)**: 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735..

Hofmeister, SJ, Asvestari E, Guo J, Heidrich-Meisner V, Heinemann SG, et al. 2022. How the area of solar coronal holes affects the properties of high-speed solar wind streams near Earth: An analytical model. *A&A*, **659**: A190. https://doi.org/10.1051/0004-6361/202141919.

Huang, Z, Lam H, Zhang H. 2023. Conditional coverage estimation for high-quality prediction intervals. *J Syst Sci Syst Eng* **32 (3)**: 289–319. https://doi.org/10.1007/s11518-023-5560-1.

Jankowiak, M, Obermeyer F. 2018. Pathwise Derivatives Beyond the Reparameterization Trick. In: *Proceedings of the 35th International Conference on Machine Learning, vol. 80 of Proceedings of Machine Learning Research*, J, Dy, Krause A (Eds.), PMLR. https://proceedings.mlr.press/v80/jankowiak18a.html.

Krista, LD, Gallagher PT. 2009. Automated coronal hole detection using local intensity thresholding techniques. *Sol Phys* **256 (1)**: 87–100. https://doi.org/10.1007/s11207-009-9357-2.

Kullback, S, Leibler RA. 1951. On information and sufficiency. *Ann Math Stat* **22 (1)**: 79–86.

Kuznetsov, V, Mohri M. Learning theory and algorithms for forecasting non-stationary time series. In: C, Cortes, Lawrence N, Lee D, Sugiyama M, Garnett R. 2015. *Advances in neural information processing systems*, **vol. 28**. Curran Associates Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/41f1f19176d383480afa65d325c06ed0-Paper.pdf.

Kvålseth, TO. 1985. Cautionary note about R^2. *Am Stat*. **39 (4)**: 279–285.

Larrodera, C, Cid C. 2020. Bimodal distribution of the solar wind at 1 AU. *A&A*, **635**, A44. https://doi.org/10.1051/0004-6361/201937307.

Le Folgoc, L, Baltatzis V, Desai S, Devaraj A, Ellis S, Martinez Manzanera OE, Nair A, Qiu H, Schnabel J, Glocker B. 2021. *Is MC dropout Bayesian?* arXiv preprint arXiv:2110.04286. https://doi.org/10.48550/arXiv.2110.04286.

Lemen, JR, Title AM, Akin DJ, Boerner PF, Chou C, et al. 2012. *The Atmospheric Imaging Assembly (AIA) on the Solar Dynamics Observatory (SDO).* Springer US, New York, NY, pp. 17–40. ISBN 978-1-4614-3673-7. https://doi.org/10.1007/978-1-4614-3673-7_3.

MacKay, DJ. 1995. Probable networks and plausible predictions-a review of practical Bayesian methods for supervised neural networks. *Network Comput Neural Syst* **6 (3)**: 469. https://doi.org/10.1088/0954-898X/6/3/011.

Magris, M, Iosifidis A. 2023. Bayesian learning for neural networks: an algorithmic survey. *Artif Intell Rev* **56 (10)**: 11773–11823. https://doi.org/10.1007/s10462-023-10443-1.

Massey, F, Jr. 1951. The Kolmogorov-Smirnov Test for Goodness of Fit. *J Am Stat Assoc* **46 (253)**: 68–78. https://doi.org/10.1080/01621459.1951.10500769.

Mathews, G, Towheed SS. 1995. NSSDC OMNIWeb: The first space physics WWW-based data browsing and retrieval system. *Comput Netw ISDN Syst* **27 (6)**: 801–808. Proceedings of the Third International World-Wide Web Conference. https://doi.org/10.1016/0169-7552(95)00033-4.

McComas, D, Bame S, Barker P, Feldman W, Phillips J, Riley P, Griffee J. 1998. Solar wind electron proton alpha monitor (SWEPAM) for the Advanced Composition Explorer. *Space Sci Rev* **86**: 563–612. https://doi.org/10.1023/A:1005040232597.

McKinney, W. 2010. Data Structures for Statistical Computing in Python. In: *Proceedings of the 9th Python in Science Conference*, S, van der Walt, Millman J (Eds.), pp. 56–61. https://doi.org/10.25080/Majora-92bf1922-00a.

Moreno-Torres, JG, Saez JA, Herrera F. 2012. Study on the impact of partition-induced dataset shift on *k*-fold cross-validation. *IEEE Trans Neural Netw Learn Syst* **23 (8)**: 1304–1312. https://doi.org/10.1109/TNNLS.2012.2199516.

Nwankwo, VUJ. 2018. Space weather: response of the atmosphere to solar activity and its implications for LEO satellites aerodynamic drag. In: *Exploring the universe: from near space to extra-galactic*, B, Mukhopadhyay, Sasmal S (Eds.), Springer International Publishing, Cham, pp. 637–644. ISBN 978-3-319-94607-8. https://doi.org/10.1007/978-3-319-94607-8_49.

Obi, JC, Jecinta IC. 2023. A review of techniques for regularization. *Int J Res Eng Sci* **11 (1)**: 360–367.

Oleksiienko, I, Tran DT, Iosifidis A. 2023. Variational neural networks *Proc Comput Sci* **222**: 104–113. https://doi.org/1016/j.procs.2023.08.148.

Pesnell, WD, Thompson BJ, Chamberlin PC. 2012. *The Solar Dynamics Observatory (SDO).* Springer US, New York, NY, pp. 3–15. ISBN 978-1-4614-3673-7. https://doi.org/10.1007/978-1-4614-3673-7_2.

Petrini, L, Cagnetta F, Vanden-Eijnden E, Wyart M. 2022. Learning sparse features can lead to overfitting in neural networks. In: *Advances in Neural Information Processing Systems*, **vol. 35**, S, Koyejo, Mohamed S, Agarwal A, Belgrave D, Cho K, Oh ACurran Associates, Inc., 9403–9416. https://doi.org/10.1088/1742-5468/ad01b9.

Pomoell, J, Poedts S. 2018. EUHFORIA: European heliospheric forecasting information asset. *J Space Weather Space Clim* **8**: A35. https://doi.org/10.1051/swsc/2018020.

Price-Whelan, AM, Sipőcz B, Günther H, Lim P, Crawford S, et al. 2018. The astropy project: Building an open-science project and status of the v2. 0 core package. *Astron J* **156 (3)**: 123. https://doi.org/10.3847/1538-3881/aabc4f.

Rahaman, R, Thiery A. 2021. Uncertainty quantification and deep ensembles. In: *Advances in Neural Information Processing Systems*, **vol. 34**, M, Ranzato, Beygelzimer A, Dauphin Y, Liang P, Vaughan JW, Curran Associates, Inc., pp. 20063–20075. https://proceedings.neurips.cc/paper_files/paper/2021/file/a70d-c40477bc2adceef4d2c90f47eb82-Paper.pdf.

Raju, H, Das S. 2021. CNN-based Deep Learning Model for Solar Wind Forecasting. *Sol Phys* **296 (9)**: 134. https://doi.org/10.1007/s11207-021-01874-6.

Ramaswamy, A. 2023. Gradient clipping in deep learning: a dynamical systems perspective. In: *The 12th International Conference on Pattern Recognition Applications and Methods ICPRAM, Lisbon, Portugal, February 22–24, 2023*, **vol. 1**, SciTePress, pp. 107–114. https://doi.org/10.5220/0011678000003411.

Reback, J, McKinney W, jbrockmendelden Bossche JV, Augspurger T, et al. 2020. *pandas-dev/pandas: Pandas 1.0.0.* https://doi.org/10.5281/zenodo.3630805.

Reddy, S, Reddy KT, ValliKumari V. 2018. Optimization of deep learning using various optimizers, loss functions and dropout. *Int J Recent Technol Eng* **7**: 448–455.

Richardson, I, Cane H. 2010. Near-earth interplanetary coronal mass ejections during solar cycle 23 (1996–2009): catalog and summary of properties. *Sol Phys* **264 (1)**: 189–237. https://doi.org/10.1007/s11207-010-9568-6.

Schou, J, Scherrer PH, Bush RI, Wachter R, Couvidat S, et al. 2012. Design and ground calibration of the Helioseismic and Magnetic Imager (HMI) instrument on the Solar Dynamics Observatory (SDO), *Sol Phys* **275 (1)**: 229–259. https://doi.org/10.1007/s11207-011-9842-2.

Smith CW, Lheureux J, Ness NF, Acuña MH, Burlaga LF, Scheifele J. 1998. The ACE magnetic fields experiment. *Space Sci Rev* **86(1)**: 613–632.

Stone, EC, Frandsen A, Mewaldt R, Christian E, Margolies D, Ormes J, Snow F. 1998. The advanced composition explorer, *Space Sci Rev* **86 (1)**: 1–22. https://doi.org/10.1023/A:1005082526237.

The SunPy Community, Barnes WT, Bobra MG, Christe SD, Freij N, et al. 2020. The SunPy Project: Open Source Development and Status of the Version 1.0 Core Package. *Astrophys J* **890**: 68. https://doi.org/10.3847/1538-4357/ab4f7a.

Thompson, WT. 2006. Coordinate systems for solar image data. *A&A,* **449 (2)**: 791–803. https://doi.org/10.1051/0004-6361:20054262.

Upendran, V, Cheung MCM, Hanasoge S, Krishnamurthi G. 2020. Solar wind prediction using deep learning. *Space Weather*, **18 (9)**: e2020SW002,478. https://doi.org/10.1029/2020SW002478.

Valdenegro-Toro, M, Mori DS. 2022. A deeper look into aleatoric and epistemic uncertainty disentanglement. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1508–1516.

Verbanac, G, Vršnak B, Veronig A, Temmer M. 2011. Equatorial coronal holes, solar wind high-speed streams, and their geoeffectiveness. *A&A*, **526**: A20. https://doi.org/10.1051/0004-6361/201014617.

Virtanen, P, Gommers R, Oliphant TE, Haberland M, Reddy T, et al. 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods* **17**: 261–272. https://doi.org/10.1038/s41592-019-0686-2.

Vybornov, FI, Sheiner OA. 2021. Coronal mass ejections and high-speed solar wind streams effect on HF ionospheric communication channel. *J Phys Conf Ser* **2131 (5)**: 052096. https://doi.org/10.1088/1742-6596/2131/5/052096.

Wilson, AG, Izmailov P. 2020. Bayesian deep learning and a probabilistic perspective of generalization. *Adv Neural Inform Process Syst* **33**: 4697–4708.

Zaknich, A. 1998. Introduction to the modified probabilistic neural network for general signal processing applications. *IEEE Trans Signal Process* **46 (7)**: 1980–1990. https://doi.org/10.1109/78.700969.

Zhou, P, Shen L, Han Y, Mi L, Xu G. 2024. A short-term wind speed prediction method utilizing rolling decomposition and time-series extension to avoid information leakage. *Energy Sources A Recovery Util Environ Eff* **46 (1)**: 3338–3362. https://doi.org/10.1080/15567036.2024.2318485.

Zou, H, Hastie T. 2005. Regularization and variable selection via the elastic net. *J R Stat Soc Series B Stat Methodol* **67 (2)**: 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x.

Cite this article as: Cobos-Maestre M, Flores-Soriano M & Barrero D 2025. BaLNet: A probabilistic Bayesian neural network for solar wind speed forecasting. *J. Space Weather Space Clim.* **15**, 34. https://doi.org/10.1051/swsc/2025029.

Appendix

A1 Full model characterization

This Appendix provides a complete characterization of the presented architecture for different configurations of the experiment. In the following table, H denotes the lead time of the forecast, and C the time (in days) taken as an approximation of a Carrington rotation. CI refers to the 95% confidence interval extracted from the predicted normal distribution generated by the model.

Table A1. Full modeling results. H denotes the lead time of the forecast. C refers to the time taken as an approximation of a Carrington rotation. CI refers to the 95% confidence interval extracted from the predicted normal distribution generated by the model.

H (days)	C (days)	Maximum CI size (km s^{-1})	Average CI size (km s^{-1})	PICP (%)	RMSE (km s^{-1})	MAE (km s^{-1})	CC	R^2	DTW ($\times 10^3$)
1	24.26	293 ± 24	136 ± 4	81 ± 4	55.1 ± 3.3	40.9 ± 2.7	0.79 ± 0.02	0.58 ± 0.05	2.52 ± 0.19
	25.26	329 ± 83	145 ± 15	**83 ± 6**	**54.3 ± 6.8**	**40.6 ± 5.6**	**0.80 ± 0.04**	**0.59 ± 0.10**	2.53 ± 0.25
	26.26	291 ± 14	141 ± 8	80 ± 3	57.5 ± 5.1	43.4 ± 3.7	0.78 ± 0.04	0.54 ± 0.08	2.69 ± 0.25
	27.26	295 ± 24	137 ± 5	81 ± 4	54.7 ± 3.3	40.8 ± 2.7	0.79 ± 0.02	0.59 ± 0.05	2.45 ± 0.21
	28.26	294 ± 13	136 ± 5	80 ± 2	55.4 ± 3.2	41.3 ± 2.3	0.78 ± 0.02	0.58 ± 0.05	**2.37 ± 0.14**
	29.26	289 ± 14	135 ± 4	77 ± 3	58.1 ± 2.8	43.8 ± 2.3	0.76 ± 0.02	0.53 ± 0.05	2.54 ± 0.09
	30.26	291 ± 18	139 ± 5	75 ± 1	65.0 ± 3.3	48.2 ± 1.9	0.70 ± 0.03	0.42 ± 0.06	2.68 ± 0.11
3	24.26	336 ± 101	136 ± 17	85 ± 3	46.7 ± 1.5	35.3 ± 1.3	0.85 ± 0.02	0.70 ± 0.02	2.27 ± 0.20
	25.26	406 ± 118	144 ± 10	88 ± 2	46.9 ± 2.6	34.6 ± 2.2	0.84 ± 0.02	0.70 ± 0.03	2.23 ± 0.29
	26.26	420 ± 134	151 ± 21	88 ± 3	48.2 ± 2.1	35.9 ± 1.5	0.83 ± 0.02	0.68 ± 0.03	2.45 ± 0.28
	27.26	372 ± 116	138 ± 13	88 ± 3	**43.7 ± 2.8**	**32.6 ± 2.6**	0.86 ± 0.02	**0.74 ± 0.03**	2.20 ± 0.27
	28.26	405 ± 145	147 ± 25	89 ± 4	46.1 ± 2.4	34.4 ± 1.8	0.85 ± 0.02	0.71 ± 0.03	2.31 ± 0.33
	29.26	340 ± 105	136 ± 11	88 ± 1	44.7 ± 1.9	33.2 ± 1.7	**0.87 ± 0.01**	0.72 ± 0.02	**2.13 ± 0.31**
	30.26	365 ± 113	142 ± 16	**89 ± 2**	45.5 ± 1.3	33.7 ± 1.4	0.86 ± 0.01	0.71 ± 0.02	2.19 ± 0.18
5	24.26	491 ± 84	124 ± 6	94 ± 2	32.5 ± 2.8	24.1 ± 1.8	0.93 ± 0.01	0.85 ± 0.03	1.96 ± 0.21
	25.26	506 ± 115	124 ± 9	94 ± 3	31.9 ± 4.9	22.9 ± 3.7	0.94 ± 0.02	0.86 ± 0.04	**1.77 ± 0.28**
	26.26	493 ± 87	127 ± 9	94 ± 3	33.0 ± 4.0	23.6 ± 2.8	0.93 ± 0.02	0.85 ± 0.04	2.04 ± 0.29
	27.26	449 ± 126	127 ± 11	**95 ± 2**	**31.3 ± 4.4**	**22.8 ± 3.7**	**0.94 ± 0.01**	**0.87 ± 0.03**	1.96 ± 0.21
	28.26	464 ± 111	128 ± 11	94 ± 3	34.2 ± 5.4	25.1 ± 3.4	0.93 ± 0.02	0.83 ± 0.05	2.13 ± 0.42
	29.26	496 ± 130	123 ± 9	94 ± 3	31.7 ± 2.1	23.0 ± 1.5	0.93 ± 0.01	0.86 ± 0.02	1.90 ± 0.17
	30.26	451 ± 82	120 ± 7	90 ± 5	32.3 ± 3.4	24.2 ± 3.0	0.93 ± 0.02	0.85 ± 0.03	1.91 ± 0.20