entropy

MDPI

Article

Multivariate Time Series Information Bottleneck

Denis Ullmann, Olga Taran and Slava Voloshynovskiy *

Faculty of Science, University of Geneva, CUI, 1227 Carouge, Switzerland; denis.ullmann@unige.ch (D.U.)
* Correspondence: svolos@unige.ch

Abstract: Time series (TS) and multiple time series (MTS) predictions have historically paved the way for distinct families of deep learning models. The temporal dimension, distinguished by its evolutionary sequential aspect, is usually modeled by decomposition into the trio of "trend, seasonality, noise", by attempts to copy the functioning of human synapses, and more recently, by transformer models with self-attention on the temporal dimension. These models may find applications in finance and e-commerce, where any increase in performance of less than 1% has large monetary repercussions, they also have potential applications in natural language processing (NLP), medicine, and physics. To the best of our knowledge, the information bottleneck (IB) framework has not received significant attention in the context of TS or MTS analyses. One can demonstrate that a compression of the temporal dimension is key in the context of MTS. We propose a new approach with partial convolution, where a time sequence is encoded into a two-dimensional representation resembling images. Accordingly, we use the recent advances made in image extension to predict an unseen part of an image from a given one. We show that our model compares well with traditional TS models, has information–theoretical foundations, and can be easily extended to more dimensions than only time and space. An evaluation of our multiple time series–information bottleneck (MTS-IB) model proves its efficiency in electricity production, road traffic, and astronomical data representing solar activity, as recorded by NASA's interface region imaging spectrograph (IRIS) satellite.

Keywords: multiple time series; forecasting method; information bottleneck; entropy; KL-divergence; mutual information; deep models; RNN; U-Net; partial convolutions

Citation: Ullmann, D.; Taran, O.; Voloshynovskiy, S. Multivariate Time Series Information Bottleneck. *Entropy* **2023**, *25*, 831. https://doi.org/10.3390/e25050831

Academic Editors: Shuangming Yang, Shujian Yu, Luis Gonzalo Sánchez Giraldo and Badong Chen

Received: 22 February 2023
Revised: 10 May 2023
Accepted: 17 May 2023
Published: 22 May 2023

1. Introduction

The scope of this work lies at the intersection of several domains, offering contributions to information theory (IT) applied to machine learning (ML), applied astrophysics, and computer vision (CV). Recently, non-recurrent models, such as transformers [1], among others [2,3], have been used to accurately predict forecasts for multivariate time series (MTS). The training of these models is guided by a proposed information–theoretical approach. This study supports the validity of some of these models with an IT approach describing the IB in the context of time series (TS). A CV-based model using partial convolution [4] with an MTS forecasting goal is presented and the link with the IB principle is proved. The approach was tested on astrophysical production, electricity production, and road traffic data. MTS, CV, and IT metrics show the empirical effectiveness of the proposed idea.

TS and MTS predictions are among the key applications of ML. They enable models to forecast the future evolution of data over time, where the time flow is represented as a single scalar for TS and a multi-dimensional vector for MTS. Meteorology, finance, online purchases, epidemic spread, and space weather forecasting are all examples of areas with great interest. Even a marginal improvement, as small as a tenth of a percent, can have a significant monetary or scientific impact. TS and MTS predictions are domains where models, such as recurrent neural networks (RNNs) [5] and long-short-term-memory (LSTM) [6], were historically developed to comply with the specificities of the temporal dimension. Although these models have similarities with classical CV models, they are part of a separate group of models that aim to mimic human memory and attention mechanisms.

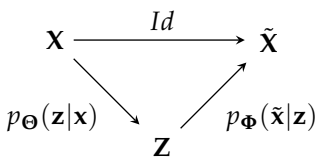
Historically, one-step-forward models were formalized before multi-step-forward models. For the former, the model predicts one step of time after a known time series. Before the rise of deep models, time series decompositions, regressions [7–10], moving averages [11], exponential smoothing techniques [12], and ARIMA [13] models were designed to forecast the most probable outcome of a time step following the given time steps in a series. Later, deep models advocated for learning a larger number of regression parameters through gradient descent [14]. RNNs face a vanishing gradient problem when they consist of more than three layers. LSTMs, a family of RNNs, aim to mimic the memory and synaptic functioning of human brains, as well as solve the vanishing gradient issue, at the cost of a possible explosion of the gradients. Finally, the more recent gated recurrent unit (GRU) [15] is an intermediate version of RNNs that works efficiently in more cases compared to classical RNNs and LSTMs.

The most recent challenges of forecasting with deep models include (a) achieving high-efficiency prediction in the context of MTS, where each time step consists of a multi-dimensional vector, with multi-step-ahead forecasts, where the model predicts multiple time steps ahead in one run, (b) interpretability, and (c) predicting the errors at each forecasted time step. The error prediction is often performed by integrating them as a joint time series that the model has to predict in parallel to the targeted time series [2]. Another option relies on stochastic predictions, where the possible forecast at each time is modeled by probabilistic distributions whose parameters are predicted by the model [2]. Interpretability is often obtained by explicit time representations or decompositions [16–18], as well as hierarchically built models that are supposed to learn classical time series decompositions by trend and seasonality [3]. To our knowledge, the most significant recent performance gains have been obtained by models that first construct a joint embedding representation of the time and space dimensions, along with compression and decompression techniques [2,3,17–20], which are afterward fed into a version of RNN, graph neural networks (GNNs) [21] or self-attention network-like transformers [17], at very high memory costs [18,20].

On the other hand, from the CV family of deep models, the objectives of *image inpainting* [4,22–24] is very similar to TS forecasting. Both attempt to recover some missing data from the information provided by two correlated known dimensions: pixel coordinates for CV and temporal–spatial for MTS. For TS and MTS, the observed temporal evolution of some data provides information to the model to forecast how these data will evolve in the future [25,26]. For image inpainting, the given parts of the image serve as prior information that aids the model in reconstructing the missing parts [24].

Recent works on images denoising [27] and inpainting [4] have shown a high capability to capture prior distributions of images and restore masked or noisy images with high accuracy. They measure accuracy in various ways, including classical Manhattan or Frobenius norms, more advanced styles [28], perceptual losses [29], and human assessments such as the mean opinion score (MOS) [30]. All of these methods use the U-Net architecture [31], with partial convolutions [4], which are particularly efficient for learning outputs that are close to the inputs in terms of similar pixels.

U-Net is a deep convolutional network that was originally designed for image segmentation [31]. It can be sketched by the following Markov chain:



where $\mathbf{Z}$ is a latent representation, $p_{\mathbf{\Theta}}(\mathbf{z}|\mathbf{x})$ is an encoding part modeled by reducing the time dimension through successive strided convolution layers, and $p_{\mathbf{\Phi}}(\tilde{\mathbf{x}}|\mathbf{x})$ is a decoding part performed by an architecture symmetric to $p_{\mathbf{\Theta}}(\mathbf{z}|\mathbf{x})$, such that the posterior $\tilde{\mathbf{X}}$ has the same shape as the prior $\mathbf{X}$. U-Net also allows a direct flow of information with an

identity mapper Id between $\mathbf{X}$ and $\tilde{\mathbf{X}}$, also referred to as the skipping layer. This direct flow of information between the prior and the posterior allows for easy reconstruction of the prior image while the information flow through the latent $\mathbf{Z}$ is responsible for the image segmentation objective. Skipping layers are also present between symmetrical hidden layers of the encoder and the decoder. Without the skipping layers, the U-Net is reduced to an autoencoder (AE) structure [32] sketched by the Markov chain $\mathbf{X} \rightarrow \mathbf{Z} \rightarrow \tilde{\mathbf{X}}$ acting as a principal component analysis dimensional reduction of $\mathbf{X}$ in $\mathbf{Z}$.

Considering the spectral time sequences as images, one can demonstrate that image processing based on generative machine learning techniques can capture the temporal patterns of the physics or logic behind the spatial data, enabling the prediction of short-term evolution. Therefore, our objective is to predict time sequences efficiently with the support of the IB principle rather than classical time sequence modelings, such as LSTM or RNN. The problem with time sequence prediction is, in this way, very similar to *image inpainting* or an *extension* problem [4,22,23].

Recently, IT approaches have formalized deep models through IB [33]. This shows that deep models are guided to find the most informative yet compressed representations for given tasks. Deep models must compress the input information into a format that ideally contains only sufficient statistics to recover posterior targeted data. To our knowledge, very few past works [34–37] have attempted to describe the IB principle in the TS and MTS contexts, whereas an extensive body of literature exists that focuses on deriving the IB principle in CV models [33,38–42].

Very few past works have used IT to design or explain the TS and MTS forecasting models that they proposed. Some attempted to estimate the entropy of TS or MTS in order to quantify their variability [34,35,37]. More interestingly, the IB principle was formulated in the RNN context but without compression of the time dimension, such that only the information present at each time step was compressed and decompressed [36]. Their work claims that each time step can be formulated by its own IB principle and that a time series with N time steps can be modeled by N IB steps. In our work, we claim that the compression of time is key for efficient forecasting and most of the existing models are realizations of a single IB principle with the compression of time and spatial dimensions.

From an information–theoretical point of view, Tishby [33] proposed the information bottleneck principle (IB), which aims to compress the input $\mathbf{X}$ and filter out all task-irrelevant information while preserving sufficient statistics in the bottleneck $\mathbf{Z}$; this was in order to decode the compressed representation into the task-specific representation, denoted as $\tilde{\mathbf{X}}$ in this context. The goal of the model is to find the parameters of the compression encoder $\boldsymbol{\Theta}$ and the decoding $\boldsymbol{\Phi}$ by solving the following optimization problem:

$$(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Phi}}) = \underset{\substack{(\boldsymbol{\Theta},\boldsymbol{\Phi}) \\ I_{\boldsymbol{\Phi}}(\mathbf{Z};\tilde{\mathbf{X}}) \geq \alpha}}{\operatorname{argmin}} I_{\boldsymbol{\Theta}}(\mathbf{X};\mathbf{Z}), \tag{1}$$

where $I_{\boldsymbol{\Theta}}(\mathbf{X};\mathbf{Z})$ represents Shannon's mutual information between $\mathbf{X}$ and $\mathbf{Z}$, which is parameterized by the parameters $\boldsymbol{\Theta}$ of the network $f_{\boldsymbol{\Theta}}(\cdot)$ mapping $\mathbf{X}$ to $\mathbf{Z}$, defined by:

$$I_{\boldsymbol{\Theta}}(\mathbf{X};\mathbf{Z}) = \mathbb{E}_{p(\mathbf{X},\mathbf{Z})}\left[\log_2 \frac{p(\mathbf{X},\mathbf{Z})}{p(\mathbf{X}),p(\mathbf{Z})}\right], \quad \text{where} \quad \mathbf{z} \sim p_{\boldsymbol{\Theta}}(\mathbf{z}|\mathbf{x}) \text{ for all } \mathbf{x} \sim \mathbf{X}. \tag{2}$$

α represents the lower bound on the mutual information between the genuine $\tilde{\mathbf{X}}$ and the compressed $\mathbf{Z}$. This lower bound ensures sufficient statistics of the genuine in the compressed $\mathbf{Z}$ in order to allow the decoder to decode $\tilde{\mathbf{X}}$. Equation (1) can also be refined with a Lagrange multiplier β, such that the parameters of the compression and decompression are solutions of:

$$(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Phi}}) = \underset{(\boldsymbol{\Theta},\boldsymbol{\Phi})}{\operatorname{argmin}} \underbrace{I_{\boldsymbol{\Theta}}(\mathbf{X};\mathbf{Z}) - \beta I_{\boldsymbol{\Phi}}(\mathbf{Z};\tilde{\mathbf{X}})}_{\mathcal{L}(\boldsymbol{\Theta},\boldsymbol{\Phi})}. \tag{3}$$

Past works [43–46] have studied the IB of AEs, often by empirically estimating the information plane (IP), i.e., the temporal graph of the training relation between mutual information, $I(\mathbf{X};\mathbf{Z})$ and $I(\mathbf{Z};\tilde{\mathbf{X}})$. For all studies, the estimation of mutual information is not exact and requires the tuning of some hyperparameters. In [43,46], the authors studied the IP at training times for different types of AEs. They show that sparse autoencoders (SAEs) significantly compress the information of MNIST data in the bottleneck, unlike the other AE, such as the variational autoencoder (VAE), for which the compression is not clear for the data (even though the VAE provides high constraints on the distribution of the bottleneck). More details about the variational decomposition of the IB and its variational approximations are provided in [47]. In [44], the authors studied the IP of vanilla AE on MNIST with different hyperparameters for mutual information estimation. Their work shows the compression of information at each hidden layer, extending from the input to the bottleneck layer. It also provides an interpretation of the link between the dimensions of the bottleneck and the compression of information. It shows that when the bottleneck dimensions are relatively small, compared to the entropy of the source, further compression is forced due to the limitation imposed by the bottleneck dimension. When the bottleneck dimensions are relatively large, there are no such limitations. Our broad interpretation of this outcome is that the AE training follows Shannon's separation theorem from the joint source and channel coding theory [48] because of the large capacity of the channel formed by the AE. In [45], the authors studied the rate-distortion performance of an AE where the IB was used as the dimensionality reduction with a fixed number of noisy information channels; they applied this AE strategy to efficiently store analog data on an array of phase-change memory (PCM) devices. The IB of AEs showed efficient rate-distortion results in this context; the authors provided theoretical insights by utilizing Shannon's separation theorem from the joint source and channel coding theory [48].

We propose a general formulation of the IB principle for MTS forecasting. We show that the U-Net architecture with source masking and an approximation of the IB loss can be regarded as a particular instance of the formulated IB principle for MTS. We provide an extensive evaluation of the proposed model on some astrophysical data of interest, and we compare the model to concurrent ones with MTS, CV, and astrophysical metrics. Interestingly, without fine-tuning, and with an approximation of the IB loss, our models based on the IB principle formulation can achieve top results on different datasets involving astrophysics solar activity prediction, electricity production, and road traffic.

One important direction of this work is the application of the IB principle to astrophysical data. The accurate prediction of solar activity, solar flares, in particular, is still an open issue. Solar flares occur as a result of the reconfiguration of magnetic fields in the corona. These energetic events accelerate highly energetic particles into space and toward the solar surface, where they cause heat and emissions in a broad range of wavelengths. Solar flares are major protagonists in space weather and can cause adverse effects, such as disruptions in communications, power grid failures on Earth, and damage to satellites and other critical infrastructures. Many attempts to predict flares exist [49–53], as well as works on flare detection [54] and analyses [55–58].

2. Methods

Forecasting models take TS data as input, noted as $X_{1:T} = [X_1 : X_T]$, or MTS data, noted $\mathbf{X}_{1:T} = [\mathbf{X}_1 : \mathbf{X}_T]$, both of length T. For TS, each time step X_t, $(t \in [1, \ldots, T])$ is scalar, whereas for MTS, each time step $\mathbf{X}_t = \left[X_t^1, \ldots, X_t^M\right]$, $(t \in [1, \ldots, T])$ is a vector of length M. As a consequence, $\mathbf{X}_{1:T}$ represents a two-dimensional tensor, with the first dimension being *temporal* of length T, and the second typically referred to as the *spatial* dimension of length M.

The goal of the forecasting models is to predict the time continuation of the input data by forecasting one step ahead or multiple steps ahead; this is denoted as $X_{T+1:T+F} = [X_{T+1}, \ldots, X_{T+F}]$ for TS, and $\mathbf{X}_{T+1:T+F} = [\mathbf{X}_{T+1}, \ldots, \mathbf{X}_{T+F}]$ for MTS, where F refers to the number of steps ahead to forecast.

In this paper, the input series $X_{1:T}$ or $\mathbf{X}_{1:T}$ is referred to as the *prior*, the true forecast $X_{T+1:T+F}$ (or $\mathbf{X}_{T+1:T+F}$) is referred to as *genuine*, and the forecast predicted by the model $\tilde{X}_{T+1:T+F}$ or $\tilde{\mathbf{X}}_{T+1:T+F}$ is referred to as *posterior*. $\mathcal{D}$ refers to the training dataset, which is also denoted as $\{\mathbf{X}_{1:T+F}\}_{\mathcal{D}}$ or $\{(\mathbf{X}_{1:T+F}, K)\}_{\mathcal{D}}$ when the data are labeled; K denotes the label, or it can be represented as $\mathbf{K}$ in the case of categorical vectors. Table A1 recalls most of the notations used in the paper.

2.1. IB-Based Optimal Compression for Time Series Forecasts

The general IB principle proposes to compress the input into a latent representation while ensuring the preservation of sufficient statistics, which are crucial for the downstream task. In the context of TS forecasting, this implies that the effective compression of the time dimension needs to be employed in order to achieve accurate forecasting. Using our notations, and according to the IB formulation of Equation (3), the goal of a model is to find the parameters of the compression encoder $\boldsymbol{\Theta}$ and of decoding $\boldsymbol{\Phi}$ by solving the following optimization problem:

$$(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Phi}}) = \underset{(\boldsymbol{\Theta},\boldsymbol{\Phi})}{\operatorname{argmin}} \underbrace{I_{\boldsymbol{\Theta}}(\mathbf{X}_{1:T}; \mathbf{Z}_{ib\_tr}) - \beta I_{\boldsymbol{\Phi}}(\mathbf{Z}_{ib\_tr}; \mathbf{X}_{T+1:T+F})}_{\mathcal{L}(\boldsymbol{\Theta},\boldsymbol{\Phi})}, \tag{4}$$

where the input $\mathbf{X}_{1:T}$ represents the previous T time steps of values, and the output $\mathbf{X}_{T+1:T+F}$ represents the subsequent F time steps. As a consequence, the bottleneck variable $\mathbf{Z}_{ib\_tr}$ should then hold the necessary part of information of prior time steps $1{:}T$ for the model to be able to forecast the time steps $T+1{:}T+F$. The index *ib_tr* explicitly reflects the nature of the MTS bottleneck task as an IB learning statistics of transitions $1{:}T \rightarrow T+1{:}T+F$. Taking inspiration from works [39], Proof of Equation (5) in the Appendix A shows that an upper bound $\tilde{\mathcal{L}}(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ on the loss $\mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ of Equation (3) can be reduced as follows:

$$\tilde{\mathcal{L}}(\boldsymbol{\Theta}, \boldsymbol{\Phi}) = H_{p_{\boldsymbol{\Theta}}}(\mathbf{Z}_{ib\_tr}) - H_{p_{\boldsymbol{\Theta}}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T}) + \beta H_{p_{\boldsymbol{\Theta},\boldsymbol{\Phi}}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr}), \tag{5}$$

where $p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib\_tr}) = \mathbb{E}_{p_{\mathcal{D}}}[p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T})]$, where $\mathcal{D}$ represents the training data consisting of pairs of priors and their corresponding known forecasts: $\mathcal{D} = \{(\mathbf{x}_{1:T}, \mathbf{x}_{T+1:T+F}) \sim \mathbf{X}_{1:T+F}\}$ and $H(.)$ stands for Shannon's entropy, such that the upper bound on $\mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ is composed by these three components:

$$\begin{aligned}
\mathcal{L}_1(\boldsymbol{\Theta}) &= H_{p_{\boldsymbol{\Theta}}}(\mathbf{Z}_{ib_tr}) \\
&= -\mathbb{E}_{p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib_tr})}[\log_2 p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib_tr})], \\
\mathcal{L}_2(\boldsymbol{\Theta}) &= -H_{p_{\boldsymbol{\Theta}}}(\mathbf{Z}_{ib_tr}|\mathbf{X}_{1:T}) \\
&= \mathbb{E}_{p_{\boldsymbol{\Theta}}(\mathbf{X}_{1:T},\mathbf{Z}_{ib_tr})}[\log_2 p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib_tr}|\mathbf{X}_{1:T})], \\
\mathcal{L}_3(\boldsymbol{\Theta}, \boldsymbol{\Phi}) &= H_{p_{\boldsymbol{\Theta},\boldsymbol{\Phi}}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib_tr}) \\
&= -\mathbb{E}_{p_{\mathcal{D}}(\mathbf{X}_{1:T+F})}\Big[\mathbb{E}_{p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib_tr}|\mathbf{X}_{1:T})}[\log_2 p_{\boldsymbol{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib_tr})]\Big].
\end{aligned} \tag{6}$$

$\mathcal{L}_3(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ is the average cross-entropy $H(p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T}), p_{\boldsymbol{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr}))$. Moreover, if the decoding distribution $p_{\boldsymbol{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$ is assumed to follow the Laplacian distribution, ref. [39] shows that the loss $\mathcal{L}_3(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ can be reduced into the average Manhattan distance, which is also referred to as the mean average error loss (MAE) between the genuine and the model estimation:

$$\mathcal{L}_3^{Lap}(\boldsymbol{\Theta}, \boldsymbol{\Phi}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{X}_{1:T+F})}\Big[\mathbb{E}_{p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T})}\big[\|\mathbf{X}_{T+1:T+F} - g_{\boldsymbol{\Phi}}(\mathbf{Z}_{ib\_tr})\|_1\big]\Big]. \tag{7}$$

Throughout the years, different forecasting models have been proposed; the MAE of $\mathcal{L}_3^{Lap}(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ has been used for training the initial models and for evaluating the performances of the forecasting models. We show in Figure 1 how existing models design

a bottleneck $\mathbf{Z}$ that compresses the time dimension, similar to $\mathbf{Z}_{ib\_tr}$ in Equation (4). The following paragraphs briefly explain the time compression and a few differences between the models selected in Figure 1.

LSTM [6], GRU [15], and DEEP-AR [2] operate with an RNN [5] over the time dimension, which is compressed in the hidden memory channel $\mathbf{Z}$. The constituting cell operates only on one time step and predicts another unique time step ($T = 1$ and $F = 1$). DEEP-AR predicts the mean and standard deviations of the forecast value, enabling the model to exhibit stochastic behavior and to learn the uncertainty on the forecast.

NBeats [3] directly operates on all prior times $\mathbf{X}_{1:T}$ and uses a hierarchical RNN structure to capture trends, seasonality, and repetitions, resulting in a hidden representation $\mathbf{Z}$. From this time compression, the model can reproduce the prior TS and forecast multiple time steps ahead $\mathbf{X}_{T+1:T+F}$. The hierarchical structure allows for interpreting the time series.

Transformers [19] encode the given TS with multiple self-attention layers in order to capture repetitions and logic between time steps. Each time step is input into its own self-attention layer. Once the time dimension is effectively compressed into $\mathbf{Z}$, the model decodes the compressed representation to generate one (or multiple) time step forecast(s) $\mathbf{X}_{T+1:T+F}$, depending on the model.

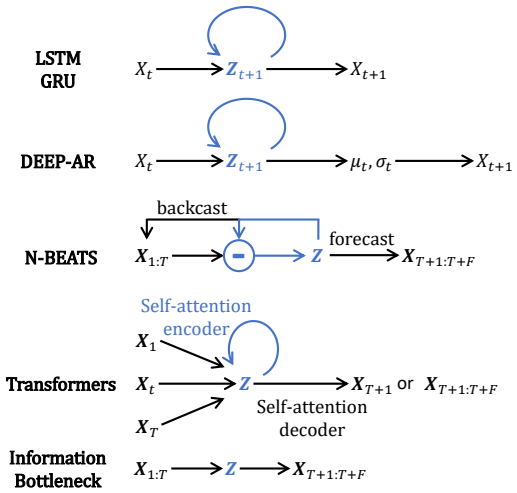


Figure 1. Comparison of Markov chains for a selection of deep TS predictors: the blue parts correspond to the compressed representations of the time dimension. Some of these may accept additional inputs (correlated context) but we did not include them in these diagrams because that would overload the global understanding, and the time dimension is compressed in the same way. A bold $\mathbf{X}$ is used when the model accepts vectors as input.

2.2. Compression by Source Masking

According to Tishby's original IB formulation, the downstream task was a classification. The form of information minimization in the IB is not necessary via the dimension reduction or the addition of noise, but it can be via any lossy operation, such as lossy compression or masking. We propose to address the IB principle for MTS via source masking, dimension reduction, and prediction of the masked parts. Using masks, Equation (4) can be rewritten as:

$$\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}) = I_{\mathbf{\Theta}}(\mathbf{X}_{1:T+F} \odot \mathbf{M}_{1:T}; \mathbf{Z}_{ib\_tr}) - \beta I_{\mathbf{\Phi}}(\mathbf{Z}_{ib\_tr}; \mathbf{X}_{1:T+F} \odot \mathbf{M}_{T+1:T+F}), \tag{8}$$

where $\odot$ is the element-wise product, also known as the Hadamard dot product, where $\mathbf{M}_{1:T}$ and $\mathbf{M}_{T+1:T+F}$ are binary time masks that have ones at the indexed time positions, $1{:}T$ or $T+1{:}T+F$, and zeros at the other time positions. Note that Equation (8) is very close to the formulation of IB for AEs in Equation (1) but with additional masks. Without

masks, the bottleneck not only holds statistics for the transitions $1{:}T \rightarrow T+1{:}T+F$, but also contains all statistics for the reconstruction of the entire sequence $1{:}T+F$. In that case, the bottleneck is reduced to an AE bottleneck $\mathbf{Z}_{ib\_ae}$ with dimension reduction purposes only. In contrast, with masking, the bottleneck $\mathbf{Z}_{ib\_tr}$ is designed to learn transition statistics $1{:}T \rightarrow T+1{:}T+F$.

2.3. Compressing Multi-Dimensional Data by Extreme Spatiotemporal Dimension Reduction

Previous subsections have not specifically taken into account the multi-dimensional aspects of MTS. Instead of scalar values X_t for TS, each time step is a vector $\mathbf{X}_t = [X_t^1, \ldots, X_t^M]$ or tensor for MTS, usually referred to as the *spatial dimension*. Previous works [59] show that models designed for TS usually fail to capture the dependencies between spatial and temporal dimensions. This difficulty has been addressed in two ways: at each time step, adding a model for the spatial interdependencies [60], or designing the spatiotemporal interdependencies jointly [18]. The first proposition models the joint spatial distributions $p(X_t^1, \ldots, X_t^M)$ for each t, either explicitly with GNN [21] or implicitly with the spatial compression [1]. The second proposition models the spatiotemporal joint distribution $p(X_1^1, \ldots, X_1^M, \ldots\ldots, X_T^1, \ldots, X_T^M)$ with joint spatiotemporal attention [18], as well as an encoder–decoder structure made of attention layers for each spatiotemporal scalar variable. The latter performs better than the first one because the spatiotemporal dependencies $X_{t'}^{m'}|X_t^m$ are explicitly designed, whereas the first type decomposes spatiotemporal dependencies in two steps: spatial plus temporal $X_t^m \rightarrow X_t^{m'} \rightarrow X_{t'}^{m'}$ or temporal plus spatial $X_t^m \rightarrow X_{t'}^m \rightarrow X_{t'}^{m'}$.

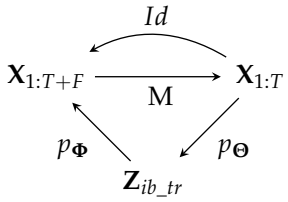
We propose to handle the spatiotemporal compression of the masked MTS data $\mathbf{X}_{1:T+F} \odot \mathbf{M}_{1:T}$ present in Equation (8) using successive two-dimensional (temporal and spatial)-strided partial convolutions *PConv* [4]. When the stride is 2, each *PConv* layer divides the 2 spatiotemporal dimensions by 2, such that, with an adequate number of hidden *PConv* layers, the spatiotemporal dimensions of the bottleneck $\mathbf{Z}_{ib\_tr}$ are reduced to 1×1, and the resulting mask $\mathbf{M}_{ib\_tr}$ is a 1×1 unit matrix. This means that $\mathbf{Z}_{ib\_tr} \odot \mathbf{M}_{ib\_tr} = \mathbf{Z}_{ib\_tr}$ and the posterior $\mathbf{X}_{1:T+F} \odot \mathbf{M}_{T+1:T+F} = \mathbf{X}_{T+1:T+F}$ can be decoded from $\mathbf{Z}_{ib\_tr}$ without any masking considerations. Finally, our approach proposes to assimilate the MTS forecasting problem as an image extension problem, where MTS, being two-dimensional, can be visualized as pseudo-images and processed with classical CV layers as convolutions.

Convolutions also present non-negligible advantages because they locally model spatiotemporal dependencies present in these MTS pseudo-images. Moreover, because of the spatiotemporal compression structure, each successive strided *PConv* hidden layer creates a more global model of the spatiotemporal dependencies, such that, in the end, the bottleneck, $\mathbf{Z}_{ib\_tr}$ fully models the global spatiotemporal dependencies.

2.4. Performing the Forecast

2.4.1. Decoder

Section 2.3 shows that the IB for MTS forecasting is equivalent to a pseudo-image extension problem, where the masked source can be compressed with successive *PConv* layers. A very efficient image inpainting model that also uses masks and *PConv* layers is proposed in [4]. Instead of simply decoding the masked posterior from the bottleneck, it proposes to use a U-Net structure with partial convolution in order to output the full image with the unmasked parts reconstructed and the masked parts predicted. We propose designing the decoder in the same way, i.e., using skipping layers and a structure similar to the encoder, consisting of *PConv* hidden layers, such that the global model is a U-Net architecture with successive *PConv* layers that extends to the bottleneck, followed by successive partial deconvolution layers *PDConv* [4]. The information flow of such a model is sketched by the following Markov chain:



For this configuration, and under a Laplacian assumption for the distribution $p_{\mathbf{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$, the third term of the IB loss $\mathcal{L}_3(\mathbf{\Theta}, \mathbf{\Phi})$ in Equation (8) becomes equivalent to:

$$\mathcal{L}_3^{Lap,UNet}(\mathbf{\Theta}, \mathbf{\Phi}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{X}_{1:T+F})}\left[\mathbb{E}_{p_{\mathbf{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T+F}\odot\mathbf{M}_{1:T})}\left[\left\|\mathbf{X}_{1:T+F} - \tilde{\mathbf{X}}_{1:T+F}\right\|_1\right]\right], \quad (9)$$

where $\tilde{\mathbf{X}}_{1:T+F}$ is the output of the U-Net. This equivalence and more details are given in Proof of Equation (A13) and Remark A1 of the Appendix A.

2.4.2. Partial IB Loss with U-Net

In Section 2.1, we show that for MTS, the IB principle imposes the three losses defined in Equation (6). Moreover, if one assumes a Laplacian distribution for $p_{\mathbf{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$, Equation (9) shows that the cross-entropy $H(p_{\mathbf{\Theta}}, p_{\mathbf{\Phi}})$, which is the third part, $\mathcal{L}_3$, of the upper bound on the IB loss, can be reduced to $\mathcal{L}_3^{Lap,UNet}$, an average Manhattan distance, which is also referred to as MAE between the output $\tilde{\mathbf{X}}_{T+1:T+F}$ and the genuine $\mathbf{X}_{T+1:T+F}$, when one samples $\mathbf{Z}_{ib\_tr}$ from the prior samples $\mathbf{X}_{1:T}$ of the training data, assuming that the Laplacian distribution of $p_{\mathbf{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$ is too restrictive for real MTS datasets, and a simple MAE cannot provide the best forecasting performance.

Image extension and inpainting processing works [4,23,61] use the same design for their models. They develop powerful image inpainting models, which involve recovering masked or missing parts of images, not only in central regions but also on the borders. These models use U-Net [31] with gated, partial, or dilated convolutions and complex losses based on the Manhattan distance MAE, such as style loss by the Gram matrix computation [62], perceptual loss by the VGG-16 hidden layer value computation [29], and/or adversarial loss [63]. One of the most efficient of these works [4] decomposes the loss into six partial losses, each of which is responsible for optimizing specific errors between the global genuine $\mathbf{X}_{1:T+F}$ and the output $\tilde{\mathbf{X}}_{1:T+F}$. The rest of this section will provide interpretations of these losses in the context of MTS. The first partial loss, $\mathcal{L}_{valid}$, is related to the *valid* or *prior* parts of the pseudo-images that can be easily reconstructed with the skipping layers:

$$\mathcal{L}_{valid} = \frac{1}{N_{pi}}\|\mathbf{M}_{1:T} \odot (\tilde{\mathbf{X}}_{1:T+F} - \mathbf{X}_{1:T+F})\|_1 = \frac{1}{N_{pi}}\|\tilde{\mathbf{X}}_{1:T} - \mathbf{X}_{1:T}\|_1, \quad (10)$$

where $N_{pi} = M \times (T + F)$ is the size of the MTS pseudo-image $\mathbf{X}_{1:T+F}$ or the number of pixels. As explained in Section 2.4.1 and Proof of Equation (A13) from the Appendix A, this part of the loss is responsible for the equivalence between the partial IB loss $\mathcal{L}_3$ defined in Equation (6) and $\mathcal{L}_3^{Lap,UNet}$, which is the Manhattan distance MAE between the U-Net output $\tilde{\mathbf{X}}_{1:T+F}$ and the genuine $\mathbf{X}_{1:T+F}$ due to the presence of the skipping layers. It ensures an easy reconstruction of the known parts, and forces the bottleneck $\mathbf{Z}_{ib\_tr}$ to learn the transition statistics $1{:}T \rightarrow T+1{:}T+F$ but not the reconstruction statistics $1{:}T \rightarrow 1{:}T$. As a consequence, in some way, it also softly minimizes $\mathcal{L}_1(\mathbf{\Theta}) = H_{p_{\mathbf{\Theta}}}(\mathbf{Z}_{ib\_tr})$, but without forcing this loss to reach a local minimum. The second loss is noted $\mathcal{L}_{hole}$ for the masked parts of the pseud-images that need to be forecasted:

$$\mathcal{L}_{hole} = \frac{1}{N_{pi}}\|(1 - \mathbf{M}_{1:T}) \odot (\tilde{\mathbf{X}}_{1:T+F} - \mathbf{X}_{1:T+F})\|_1 = \frac{1}{N_{pi}}\|\tilde{\mathbf{X}}_{T+1:T+F} - \mathbf{X}_{T+1:T+F}\|_1. \quad (11)$$

This loss is exactly the IB partial loss $\mathcal{L}_3 = \mathcal{L}_{hole}$ of Equation (6) if we assume $p_{\mathbf{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$ to follow a Laplacian distribution. Under this assumption, this loss forces the encoder to actually let $\mathbf{Z}_{ib\_tr}$ represent the transition statistics between the prior and the posterior; this loss also lets the decoder generate well the posterior from this bottleneck representation. It is interesting to note the $\mathcal{L}_{valid} + \mathcal{L}_{hole} = \mathcal{L}_3^{Lap,UNet} = \mathcal{L}_3$ under the Laplacian assumption of $p_{\mathbf{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$ and for the U-Net flow of information. The three next losses are also MAE, but instead of being directly computed on the genuine $\mathbf{X}$ and prior $\tilde{\mathbf{X}}$, they are computed on deep representations of these variables. These three losses are referred to as *perceptual* and *style* losses:

$$\begin{aligned}
\mathcal{L}_{perceptual} = 2 &\sum_{p \in H_{VGG}} \frac{\|\mathbf{\Psi}_p(\tilde{\mathbf{X}}_{T+1:T+F}) - \mathbf{\Psi}_p(\mathbf{X}_{T+1:T+F})\|_1}{N_{\mathbf{\Psi}_p(\mathbf{X}_{1:T+F})}} \\
&+ \sum_{p \in H_{VGG}} \frac{\|\mathbf{\Psi}_p(\tilde{\mathbf{X}}_{1:T}) - \mathbf{\Psi}_p(\mathbf{X}_{1:T})\|_1}{N_{\mathbf{\Psi}_p(\mathbf{X}_{1:T+F})}}, \\
\mathcal{L}_{style_{out}} = &\sum_{p \in H_{VGG}} \frac{\|\mathbf{\Gamma}_p(\tilde{\mathbf{X}}_{1:T+F}) - \mathbf{\Gamma}_p(\mathbf{X}_{1:T+F})\|_1}{N_{\mathbf{\Psi}_p(\mathbf{X}_{1:T+F})}}, \\
\mathcal{L}_{style_{comp}} = &\sum_{p \in H_{VGG}} \frac{\|\mathbf{\Gamma}_p([\mathbf{X}_{1:T}, \tilde{\mathbf{X}}_{T+1:T+F}]) - \mathbf{\Gamma}_p(\mathbf{X}_{1:T+F})\|_1}{N_{\mathbf{\Psi}_p(\mathbf{X}_{1:T+F})}},
\end{aligned} \tag{12}$$

where $\mathbf{\Psi}_p$ are selected hidden layers of a pre-trained VGG-16 [64] deep image model classifier, where H_{VGG} is the set of selected hidden layer indices, and $\mathbf{\Gamma}_p(\mathbf{X})$ are Gram operators on the hidden layers of VGG [29], as defined by $flatten[\mathbf{\Psi}_p(\mathbf{X})] \cdot flatten[\mathbf{\Psi}_p(\mathbf{X})]^T$. While the MAE of $\mathcal{L}_3^{Lap,Unet}$ is equivalent to the IB partial loss $\mathcal{L}_3$ from Equation (6) with the Laplacian assumption of $p_{\mathbf{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$, for these losses, MAE is applied to deep hidden representations of $\mathbf{X}$ and $\tilde{\mathbf{X}}$. In a similar manner to the normalizing flow [65], where each hidden layer of a deep network modifies the distribution, hidden layers of the VGG and Gram operators are deterministic mappers that modify the distribution of $\mathbf{X}_{T+1:T+F}$. As a consequence, we assume that the MAEs of $\mathcal{L}_{perceptual}$ and $\mathcal{L}_{style}$ can provide equivalents to the IB partial loss $\mathcal{L}_3$ from Equation (6) for other distributions $p_{\mathbf{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$ than the simple Laplacian. Hidden layers of the VGG-16 model capture the statistics related to the prediction of the humanly recognizable class to which an image may belong. The losses using MAE on these hidden layers are assumed to measure human perceptual features, and Gram operators are known to capture styles in an image [62]. Finally, a combination of $\mathcal{L}_{valid}$, $\mathcal{L}_{hole}$, $\mathcal{L}_{perceptual}$, $\mathcal{L}_{style_{out}}$, and $\mathcal{L}_{style_{comp}}$ provides an equivalent to the IB partial loss $\mathcal{L}_3$ for a less restrictive assumption than the simple Laplacian distribution of $p_{\mathbf{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$. The last loss is referred to as *total variation*:

$$\mathcal{L}_{tv} = \|\mathbf{X}_T - \tilde{\mathbf{X}}_{T+1}\|_1. \tag{13}$$

In image extension or inpainting problems, this loss forces the borders of the predicted holes to be smooth, which is a valid assumption for large images with high definition. In the context of MTS, this loss forces the first forecasted time step $T+1$ to be similar to the last known time step T. This smoothness assumption is also valid for the majority of real-world observations, where most of the functions are continuous.

The U-Net structure does not impose specific distributions for $p(\mathbf{Z}_{ib\_tr})$ and $p(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T})$, such that the partial IB losses $\mathcal{L}_1$ and $\mathcal{L}_2$ are intractable, but U-Net imposes an extreme spatiotemporal compression in the bottleneck $\mathbf{Z}_{ib\_tr}$. The source masking, combined with U-Net's skipping layers, allows for a limitation of $\mathcal{L}_1$, as the transition statistics $1:T \rightarrow T+1:T+F$ rather than the reconstruction ones $1:T \rightarrow 1:T$ are learned. For these reasons, the loss used for the model in the experiments is the following partial IB loss:

$$\mathcal{L}_{total} = \mathcal{L}_{valid} + a_1\mathcal{L}_{hole} + a_2\mathcal{L}_{perceptual} + a_3(\mathcal{L}_{style_{out}} + \mathcal{L}_{style_{comp}}) + a_4\mathcal{L}_{tv}, \tag{14}$$

where a_1, a_2, a_3, and a_4 are empirically defined hyperparameters. In practice, we use $a_1 = 6$, $a_2 = 0.05$, $a_3 = 120$, and $a_4 = 0.1$, and only the hidden layer activations $\mathbf{\Psi}_p$ with indices $p = 3,\ 6$, and 10 of the VGG-16 [64] are used to compute $\mathcal{L}_{style_{out}}$, $\mathcal{L}_{style_{comp}}$, and $\mathcal{L}_{perceptual}$, such that $H_{VGG} = \{3, 6, 10\}$, such as in [4]. These parameters were fine-tuned in [4] for image datasets, where pixels take values between 0 and 1. The pseudo-images created from the IRIS, AL, and PB datasets also have pixels ranging from 0 to 1. Because of the good results we achieved with these parameters, we assumed that it was enough to prove the efficiency and adaptability of the described method on different types of MTS data, and we did not attempt to further fine-tune these hyperparameters for each dataset evaluated.

2.4.3. IB Interpretation with the Partial Loss

The encoder has a mapping form that consists of two parts. The first encoding corresponds to the masking, i.e., vector $X_{1:T+F}$, only $X_{1:T}$ is retained as the input to the second part. Thus, in principle, the masking part can be any stochastic map that masks the parts to be predicted. This technique is similar to recent methods referred to as *masked image modeling* (MIM) [66,67] and it is often used in the pretraining of image autoencoders or transformers [66,68]. The second encoding part is a nonlinear embedding implemented as a deterministic encoder, which is the compression part of the U-Net, along with its connecting layers. This second compression is guided by successive masked convolutions with strides of the order of 2 to obtain a bottleneck by the dimension reduction of shape $1 \times 1 \times K$, i.e., where the spatial and temporal dimensions are reduced to 1. This compressed representation is noted as $\mathbf{Z}_{ib\_tr}$ and referred to as the bottleneck in the paper. The masking, together with the deterministic nonlinear embedding, form a stochastic mapping, and can be considered as the equivalent part of the stochastic encoder in the IB framework.

In the end, a nonlinear decompression implemented in a form of a deterministic decoder predicts $\mathbf{X}_{1:T+F}$ from this bottleneck representation $\mathbf{Z}_{ib\_tr}$ and from the skipping layers that map the prior $\mathbf{X}_{1:T}$ information between the input and output. Theoretically, in Section 2.3, we show that the bottleneck should only retain the necessary information of the transition statistics $1 : T \rightarrow T + 1 : T + F$ between the prior and the posterior, and not the statistics of the reconstruction of the prior $\mathbf{X}_{1:T}$. This is because all of the information from the prior $\mathbf{X}_{1:T}$ is transmitted to the output via the skipping layers. This shortcut flow of information is specific to the structure of U-Net, is performed without compression, and preserves the spatiotemporal positions of the prior information. As such, theoretically, only the statistics of the transitions $1 : T \rightarrow T + 1 : T + F$ are retained in the bottleneck $\mathbf{Z}_{ib\_tr}$ and the following Markov chain holds:

$$\mathbf{X}_{1:T} \rightarrow \mathbf{Z}_{ib\_tr} \rightarrow \mathbf{X}_{T+1:T+F}. \tag{15}$$

Moreover, to better understand the role of the bottleneck, we can consider these two thought experiments:

- If we remove the skipping layers, the bottleneck should not only retain the statistics of the transitions from the prior to the posterior but also the reconstruction statistics of the prior.
- If we also remove the source masking of the posterior in $\mathbf{X}_{1:T+F}$, the model is reduced to an autoencoder (AE) and the bottleneck is supposed to perform a dimension reduction of the MTS. Because of the curse of dimensionality, this technique is commonly used to further perform better classifications on the bottleneck representation than on the raw high-dimensional MTS data.

In our case, instead of imposing a distribution of the latent space, such as for VAE, we apply special masking jointly with a dimensionality reduction. This framework can be considered as the lossy part of the information encoding.

2.5. Proposed Model

The model designed in Section 2.4 corresponds to a traditional U-Net complemented with masks and partial convolutions [4,31]. Multidimensional successive *PConv* and *PDConv* layers with a stride of 2 are used when the data are MTS ($M > 1$). The bottleneck must have a 1×1 spatiotemporal shape; because of stride 2, the input pseudo-images must be zero-padded to become squares with a power of 2. As a consequence, if $2^l \times 2^l$ is the spatiotemporal shape of the pseudo-image, the designed U-Net must have l successive *PConv* followed by l successive *PDConv*. Training is performed with 100 epochs and the Adam optimization of gradient descent with a 2×10^{-4} learning rate for the loss defined in Equation (14). The model could be generalized to N-dimensional *PConv* and *PDConv* layers when several dimensions are necessary to model each time step. For instance, in videos, each time step is an image with horizontal and vertical spatial dependencies.

Example of architecture when $128 < \max(M, T+F) \leq 256$**:** The maximum spatiotemporal size is $\max(M, T+F)$, which can also be interpreted as the maximum width or height of the pseudo-images. In this situation, input pseudo-images are zero-padded to obtain a square shape 256×256, and the investigated model is sketched in Figure 2; it uses the classical image extension architecture, U-Net [31], which has a symmetrical structure made of an encoder and a decoder, both with 8 layers. Implementation details of the architecture are given in the Appendix B, Table A2. The encoded representation $\mathbf{Z}_{ib\_tr}$ has a size of $1 \times 1 \times 512$. Each layer of the encoding part divides the width and height with strides of a factor of 2, and increases the number of channels up to 512. The decoder has a symmetrical structure, but the inputs of each layer are concatenations of upsampled versions of the previous layer's outputs with the output of the symmetrical encoding layer. It is combined with partial convolutions (PCs) that were proposed in [4] to handle the masked data. These convolutions are applied at each hidden step and are designed to not take into account the missing data, such that $\mathbf{X}_{T+1:T+F}$ from $\mathbf{X}_{1:T+F}$ at the input layer. At each step of the encoding, the proportion of the masked part is reduced. Each PC is followed by a batch normalization and a ReLU activation, but for the last output layer, the activation is a sigmoid. For the training, we used input images of size 240×240, which were center-padded by zeros to make an image of size 256×256 for fitting the U-Net input size. Because of the 2 strides at the encoding steps and the 1×1 size of the latent representation, a U-Net with 8 encoding layers requires input sizes of $2^8 = 256$.

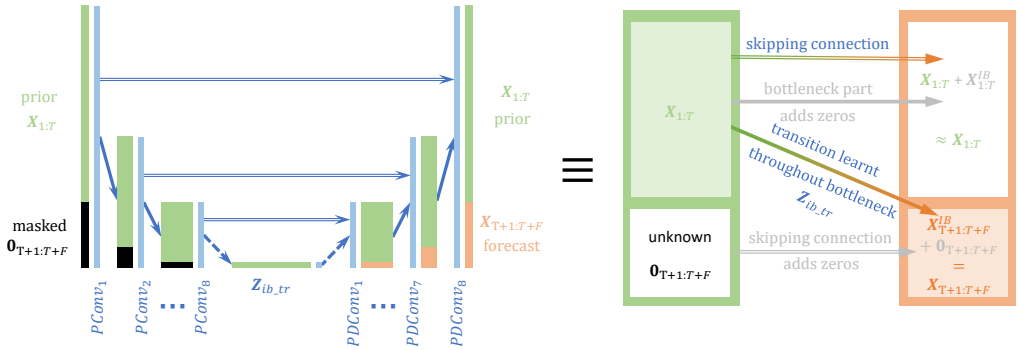


Figure 2. Schematic analogy between the IB principle and image extension: (**Left**) schematically shows the time prediction under the IB principle, with compression and decoding, using *PConv* and *DPConv* and skipping connections to form a variant of U-Net. (**Right**) is an equivalent representation seen as the image extension, where the skipping layers connect $\mathbf{X}_{1:T}$ from the input to the output, and the bottleneck principle allows predicting $\mathbf{X}_{T+1:T+F}$ from $\mathbf{X}_{1:T}$.

When the input of the maximum spatiotemporal size $\max(M, T+F)$ is smaller than 128, the model needs less than 8 *PConv* and *PDConv* layers for the encoder and decoder parts of the U-Net. In general, the number of layers must be $\log_2(\max(M, T+F))$ to ensure a 1×1 spatiotemporal size in the bottleneck $\mathbf{Z}_{ib\_tr}$.

In [4], they specify that because of the masks, batch normalization prevents training from converging. This is because the mean and standard deviation values of batch normalization layers for each sample are biased by the masks. As a consequence, they train the first half of epochs with trainable batch normalization layers, and they freeze the batch normalization layers for the second half of epochs. This is done in TensorFlow by setting the layer parameter $trainable = False$ during training. In our case, all sample masks have the same size and spatiotemporal positions. As a consequence, the mean and standard variations of the samples are not very biased by those masks. Actually, the experimentation showed that freezing the batch normalization for the last epochs did not lead to improved performance.

2.6. IRIS Dataset

IRIS is NASA's interface region imaging spectrograph satellite [69]. IRIS observes regions of the atmosphere of the Sun with many different settings of possible observations recorded in a specified cadence. A time sequence is encoded into a two-dimensional representation in the form of images. We used the designed model to predict time sequence data provided by the IRIS mission. The basis of the IRIS satellite data retrieval is shown in Figure 3. Each observed event is composed of a maximum of four videos of a selected region on the surface of the Sun, together with spectral videos, where a slit is positioned to perform the diffraction [70].

In this work, only the spectral data from *MgII h&k* lines, between 2793.8401Å and 2806.02Å, were considered. This wavelength's range is represented by a vector of size 240. According to modern solar physics theories, spectral data are supposed to contain most of the information on the physics of the Sun, and *MgII h&k* lines are considered some of the best lines to recover information from the chromosphere [71]. The predictions of solar spectral data are crucial for different reasons, including the solar flares forecasting and solar activity in general.

IRIS data are publicly available (iris.lmsal.com/data.html) (accessed on 20 February 2023) but only part of the data is labeled [72]. Only three types of solar activities were considered for this study: quiet Sun (QS), where nothing special appears, active region (AR), where some activity is observed, such as solar prominence, filaments, jets, and flaring profile (FL), when a flare appears during the observed event. Each event is assigned a hierarchical label, such that an event is labeled FL even when it includes AR time steps. The data are normalized at each time step by its maximum value, such that the maximum value at each time step, or the *intensity* of the signal, is reduced to one. This allows for an easier comparison of spectral profiles at each time step, and simplifies the process in terms of the ML.

2.6.1. Problem Formulation

The IRIS restrictions of online observations: Figure 3 explains the IRIS observations in the atmosphere of the Sun with images of given wavelengths and spectra of given positions. Despite the very high precision of IRIS and its capacity to observe a very wide range of astrophysical parameters in time and space, significant difficulties inherent to online observations remain. Spectral observations are limited in time and space as they only correspond to the position of the slit at a given time, which may vary, and the satellite has to store the data before sending them to Earth-based stations [70]. IRIS observations are, therefore, very sparse in all of the potential observable parameters and they may lack a lot of data from other spatial positions. We may also be interested in further observations after the termination of the acquisition/recording session limited by the IRIS storage memory capacity.

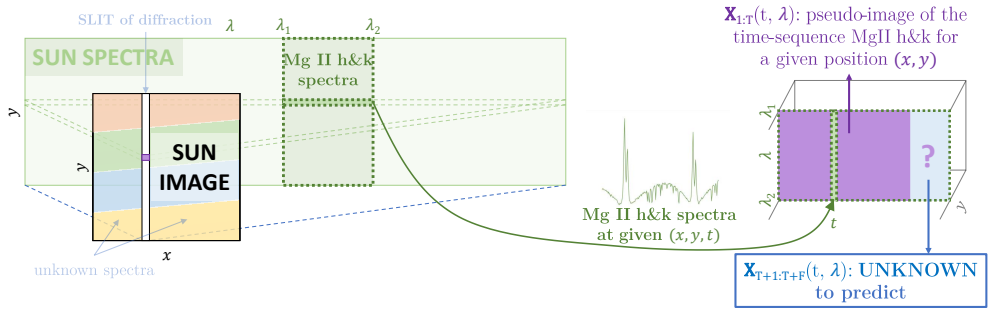


Figure 3. Problem formulation: (x, y) represent the spatial coordinates, λ and t, respectively, represent the spectral and time coordinates. NASA's IRIS satellite integrates a mirror from which the *Sun image* or videos are captured by a sensor paired with a wavelength filter chosen among 1330 Å, 1400 Å, 2796 Å, and 2832 Å. This mirror holds a vertical slit from which the diffraction occurs. The x position of the slit can vary in time and is chosen before the observation. A sensor behind the mirror captures the *Sun spectra* for each vertical position y of the *Sun's image*, but only at the x position of the slit. We only consider the MgIIh/k data, which are between $\lambda_1 = 2793.8401$ Å and $\lambda_2 = 2806.02$ Å, and we consider all available time sequences.

Spectral time sequence forecasting represents a significant step forward in flare forecasting and assists in planning satellite observations. Figure 3 presents the solar–physical interpretation of the spectral time sequence data, represented as images $\mathbf{X}^{1:M}_{1:T+F}$ of physical dimensions $time \times wavelength$ with $1 \leq time\ t \leq T + F$, $1 \leq wavelength\ \lambda \leq M$; the left part $\mathbf{X}^{1:M}_{1:T}$ $(1 \leq t \leq T, 1 \leq \lambda \leq M)$ corresponds to the known prior sequence, and the right part, $\mathbf{X}^{1:M}_{T+1:T+F}$ $(T + 1 \leq t \leq T + F, 1 \leq \lambda \leq M)$, masked, predicted, or genuine, is the sequence that has to be predicted by the model.

Non-homogeneous cadences of the data time series modeling are usually performed by RNNs [73] or LSTMs [6], as briefly summarized in Figure 1. These models are designed for time series with fixed given cadences; Figure 4 shows the wide variety of our data cadences, making the use of RNN or LSTM difficult. To represent the time sequences $\mathbf{X}_{1:T+F}$ of the data under a common cadence, one should represent them by a cadence equal to the greatest common divisor of all of the cadences, which would obviously make those time sequences $\mathbf{X}_{1:T+F}$ highly sparse and penalize the learning of transitions between time steps.

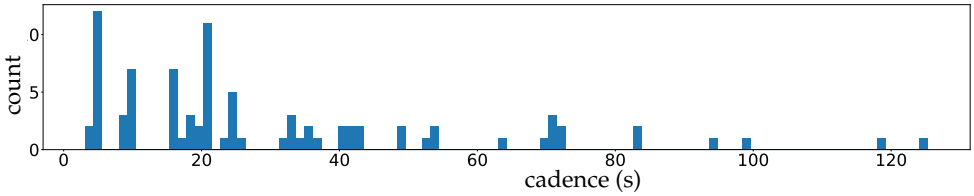


Figure 4. Histogram of the cadences in seconds/time steps.

Clustering spectral data: The 53 clusters of MgIIh/k lines found in [55] allow interpreting the physics on the surface of the Sun. We can compare the original and predicted time sequences through their clustered time sequences in order to prove the utility of our forecasting model in solar–physics by conserving the types of activities.

Astrophysical features: In [72], the authors defined ten solar spectra features to be used as dimensional reductions of spectral data for activity classification purposes. We studied the conservation of these features in the forecasted sequences to show the applicability to astrophysics.

2.6.2. Proposed Approach

As described in Section 2.6.1, a solar spectral time sequence is represented by an image $\mathbf{X}_{1:T}$ and the model has to predict the continuation, which is the time sequence equivalent to the image $\mathbf{X}_{T+1:T+F}$. $\mathbf{X}_{T+1:T+F}$ should be the right extension of the image $\mathbf{X}_{1:T}$, where each

column of the image represents a spectrum at a growing time step from the left to the right. Because of the architecture of the image extension models, the input and output images have the same dimensions. The targeted output image $\mathbf{X}_{T+1:T+F}$ is the concatenation of $\mathbf{X}_{1:T}$ with $\mathbf{X}_{T+1:T+F}$ on the right of it, whereas the input image $Concat([\mathbf{X}_{1:T}, \mathbf{0}_{T+1:T:F}])$, has a blank image $\mathbf{0}_{mask}$ of the same shape as $\mathbf{X}_{T+1:T+F}$ on the right of $\mathbf{X}_{1:T}$.

For both recurrent and image extension models, the input has the same information, organized differently, and the output of the image extension models differs by the left concatenation of the input. Figure 2 shows the skipping layers in the image extension models that help the transition of $\mathbf{X}_{1:T}$ from the input to the output.

2.7. Other MTS Dataset

Two other datasets are used to provide a baseline evaluation between our IB-designed models and concurrent ones.

- **AL** dataset: The solar power dataset for the year 2006 in Alabama is publicly available (www.nrel.gov/grid/solar-power-data.html, accessed on 20 February 2023). It contains solar power data for 137 solar photovoltaic power plants. Power was sampled every 5 min in the year 2006. Preprocessing was conducted to only extract daily events by ignoring nights when data were zero. At each 5-min interval, the data consisted of vectors with 137 dimensions, and these vectors were normalized by their maximum coordinates. For example, in the case of IRIS data, the maximum value at each time was always set to 1.
- **PB** dataset: PeMS-BAY data [74] are publicly available (https://zenodo.org/record/5146275#.Y5hF7nbMI2w, accessed on 20 February 2023) and were selected from 325 sensors in the Bay Area of San Francisco by the California State Transportation Agency's Performance Measurement System [75]. The data represent 6 months of traffic speeds ranging from January 1 to May 31 2017. At each 5-minute interval, the data consist of vectors with 325 dimensions, and these vectors are normalized by their maximum coordinates. For example, in the case of IRIS data, the maximum value at each time was always set to 1.

2.8. Complementary Classifiers to Show Consistency with Applied Sciences

The experimental proof was conducted on IRIS-labeled data to show that the proposed model is not simply capable of predicting possible images but also capable of predicting the information logic behind them. It is common for MTS data (that are to be forecasted) to be classified based on types of activity. There are multiple examples of MTS types of activity, including displacement, boom, euphoria, profit-taking, and panic classifications. In astrophysics, when dealing with solar observations, the types of activity can be categorized as quiet, active, and flaring.

We implemented a classifier composed of eight strided convolutional layers, with dense ending layers, which allowed it to output a vector of size corresponding to the number of classes. This classifier was trained on labeled MTS data $\{(\mathbf{X}_{1:T+F}, \mathbf{K})\}_{\mathcal{D}}$, where $\mathbf{K}$ stands for the categorical one-hot vector representing the class activity for the series $\mathbf{X}_{1:T}$. Once trained, the classifier was used to classify the prior, the genuine, and the predicted forecasts. The classification accuracy between the genuine and the predicted forecasts was evaluated together with the true skill statistic (TSS), which is also known as the Hansen and Kuiper skill score [76], and the Heidke skill score (HSS) [77], which is also known as *kappa* [78]. These two scores were evaluated globally and for each class of prediction. For one class, these scores are defined as follows:

$$TSS = \frac{tP \times tN - fP \times fN}{gP \times gN}, \quad \text{and} \quad HSS = 2\frac{tP \times tN - fP \times fN}{gP \times pN + gN \times tP}, \tag{16}$$

where t stands for *true*, f is *false*, g is *genuine*, p is *predicted*, P represents *positives*, and N represents *negatives*. In a classification with more than two classes, [79] shows that these scores can be defined by generalization, as follows:

$$TSS = \frac{trace(CM - ICM)}{trace(CM^* - ICM^*)}, \quad \text{and} \quad HSS = 2\frac{trace(CM - ICM)}{trace(CM^* - ICM)}, \tag{17}$$

where $trace(.)$ is the diagonal sum operator for a matrix of dimension $m \times m$, which is eventually larger than 2×2. $CM = \left(Count_{g_i,p_j}\right)_{i,j}$ is the confusion matrix holding the joint counts of genuine classification cases g_i (rows) versus forecast classification cases p_j (columns), and $ICM = \left(\frac{Count_{g_i} \times Count_{p_j}}{total\ count}\right)_{i,j}$ is the confusion matrix expected when the genuine and forecast classifications are independent events. $CM^* = diag(Count_{g_i})$ is the expected diagonal confusion matrix when the classifications are ideal (an ideal classification is defined by $Count_{g_i,p_j} = 0$ for all $i \neq j$, such that the confusion matrix CM is diagonal); ICM^* is the corresponding expected confusion matrix when genuine and forecast classifications are independent events.

2.9. Comparison with Other Models

To our knowledge, for almost all MTS datasets, current state-of-the-art (SOTA) datasets are achieved by TS decomposition networks, such as NBeats [3] and SCINet [80], and by pre-trained models designed as graph neural networks [60,81] or transformers [17]. We compare our proposed IB-MTS model with three types of models:

- Multiple successive IBs, where each time step is an instance of the IB formulation. This includes multidimensional RNN, LSTM, and GRU models [5,6,15].
- Composition of two successive IBs: A spatial IB is followed by a temporal IB. For example, this would involve encoder–decoder models using successive RNN, LSTM, or GRU recurrent models at each layer of compression and decoding [82,83].
- Unique joint spatiotemporal IB: The encoder jointly compresses spatial and temporal dimensions of the prior into a bottleneck with an extreme spatiotemporal dimensional reduction; this is our proposed IB-MTS formulation.
- MTS decomposition model, such as NBeats [3].

Names and details of the concurrent evaluated models are listed below:

- **LSTM** model: An LSTM cell [6] performs the one-step-ahead forecast and is trained to predict $\mathbf{X}_{t+1}$ from $\mathbf{X}_t$. It incorporates one layer with M LSTM units. For instance, for the 240×240 spatiotemporal dimensions of IRIS data, the 180 first time steps are the prior data, and the 60 last time steps are the posterior data to forecast. This model is designed with 240 spatial LSTM/GRU units looped 180 times and all of the cell outputs are returned by the model using TensorFlow option $return\_sequences = True$. This layer returns a 180×240 output and only the last 60 time steps are kept. Moreover, a source masking of the posterior is applied to the input and an identity skipping layer is added to transmit the prior $\mathbf{X}_{1:T}$ to the output at the same temporal positions in $\mathbf{X}_{1:T+F}$, such that the LSTM layer only accounts for predicting the posterior part $\mathbf{X}_{T+1:T+F}$. The number of units is directly determined by the shape of the input and output data. Details of the architecture are given in the Appendix B, Table A3.
- **GRU** model: A GRU [15] cell is trained to predict $\mathbf{X}_{t+F}$ from $\mathbf{X}_t$. The structure and number of units are the same, similar to the LSTM models, but GRU cells are used instead of LSTM ones. Details of the architecture are given in the Appendix B, Table A3.
- **ED-LSTM** model: A version using LSTM cells with an encoder and a decoder was implemented as described in Figure 5. Because of the encoder and decoder structures, we name it ED-LSTM. This model conducts multiple step-ahead forecasts and can forecast $\mathbf{X}_{T+1:T+F}$ from $\mathbf{X}_{1:T}$. The model incorporates LSTM cells organized into four

layers: two layers of encoding into a bottleneck and two layers of decoding from the bottleneck. The first encoding layer is composed of 100 spatial units looped 180 times on the prior IRIS data and all of the cell outputs are returned by the model using TensorFlow option *return_sequences* = *True*, returning a 180×100 spatiotemporal output accounting for a spatial compression. The second encoding layer is composed of 100 spatial units looped 180 times and only the last cell outputs of the recurrences are returned, returning a 100-dimensional bottleneck that accounts for a spatial compression followed by a temporal compression. This bottleneck representation is repeated 60 times for IRIS data in order to model the decoding of 60 posterior time steps to forecast. After this repetition, the data are 60×100 and fed to the first decoding layer with 100 spatial units looped 60 times and initialized with the states obtained from the second encoding layer; indeed, the structure is symmetrical, such that the first and second decoding layers are, respectively, the images of the second and the first encoding layers. All cell outputs are returned by the model using TensorFlow option *return_sequences* = *True*, such that a 60×100 spatiotemporal output is returned. The second layer of the decoder is designed with 100 spatial units looped 60 times and initialized with the states obtained from the first encoding layer, such that a 60×100 output is returned. In the end, a time-distributed dense layer is used to map the 60×100 output data into a 60×180 MTS data format. For these models, the input and output shapes are determined by the data and one can only change the number of spatial cells n_1 and n_2 used, respectively, in the first and second layers of the encoding part. n_2 determines the dimension of the bottleneck and on the IRIS data, $180 > n_1 \geq n_2 \geq 1$. Our experiments show that the results of these models do not depend much on the values of n_1 and n_2, but significantly drop when n_2 is very small, close to 1. Details of the architecture are given in the Appendix B, Table A4.

- **ED-GRU** model: This model follows the same structure as the ED-LSTM but with GRU cells instead of LSTM cells. The structure and number of units are the same as with ED-LSTM models, but GRU cells are used instead of LSTM ones. Details of the architecture are given in the Appendix B, Table A4.
- **NBeats** model: We use the code given in the original paper [3]. This model can forecast $\mathbf{X}_{T+1:T+F}$ from $\mathbf{X}_{1:T}$. The model is used in its generic architecture as described in [3], with 2 blocks per stack, theta dimensions of $(4, 4)$, shared weights in stacks, and 100 hidden layers units. For IRIS data, the prior is 180×240, the forecast posterior is 60×240, and the backcast posterior is 180×240, but we also use a skipping layer to connect the input to the backcast posterior, such that the model is forced to learn the transition between the prior and the forecast posterior. we attempted NBeats with other settings and other numbers of stacks without the gain of performance, and when the number of stacks became greater than 4, the model failed to initialize on our machines.

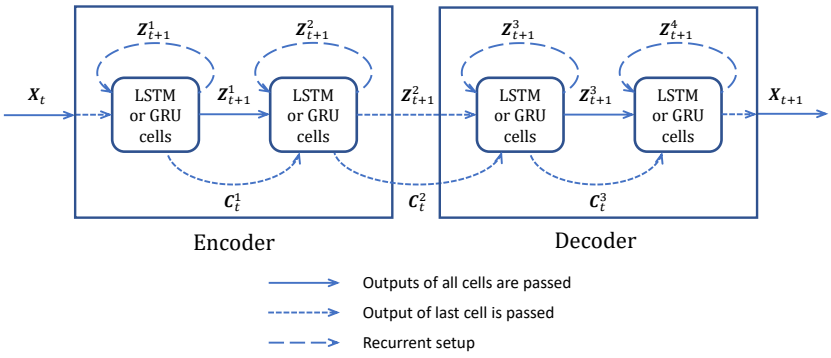


Figure 5. Structure of the ED-LSTM and ED-GRU models used for comparison. $\mathbf{C}^i_t$ represents the hidden state vectors for GRU cells, combined with cell state vectors for LSTM cells.

Table 1 provides the number of parameters for each model. The evident drawback of our proposed IB-MTS is the high amount of parameters inherent to U-Net structures with several levels of deepness. The number of parameters for the concurrent evaluated models is determined by the shape of the input data or by hardware constraints for NBeats. We believe that our model also integrates more parameters because it is based on a jointly spatiotemporal compression whereas the others are based either on a temporal succession of spatial IB or on a combination of a spatial followed by a temporal IB. In theory, the number of parameters for our proposed model should be the square of the number of parameters of the models based on the N successive spatial IB. In practice, it is less than the square, and still able to model a joint distribution, which is not the case for the other ones.

Table 1. Parameters and trained steps of the evaluated models.

	IB-MTS	LSTM	ED-LSTM	GRU	ED-GRU	N-BEATS
Parameters	65,714,097	461,760	401,840	347,040	308,640	100,800
Trained step (ms/sample)	91	83	98	87	98	424

Moreover, a remarkable thing is that our proposed IB-MTS model needs less time to be trained than the others, except for simple RNNs. The differences in duration are even more significant compared to NBeats. NBeats has a very small amount of parameters but is very slow to train compared to the other models. From our experience, instantiating the NBeats model can be quite slow, and in practice, it is impossible to instantiate with much more layers and parameters than the vanilla one. For instance, 120 GB of RAM was not enough when we attempted to set the NBeats model with millions of parameters.

3. Results

Considering that our work may be of interest to readers from various backgrounds, be it MTS prediction, information theory, ML in general, or applied physics, several types of evaluations performed on our model and concurrent ones are listed below; Figure 6 explains the last three listed evaluations.

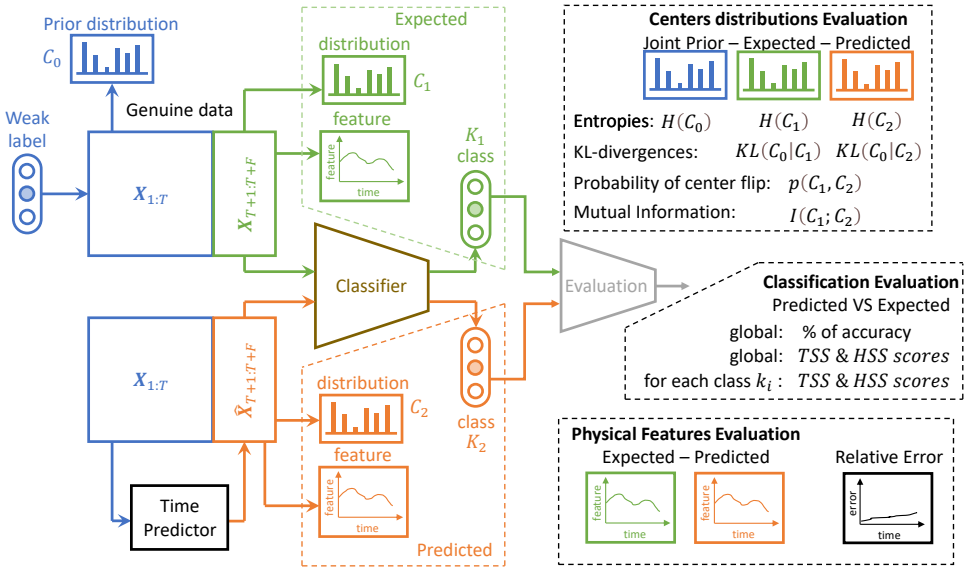


Figure 6. Evaluations performed on the proposed time predictor: center assignments, activity classification, and physical features. Classical MTS and CV evaluations were also performed without appearing in this diagram for readability concerns.

- **MTS metrics:** MAE, MAPE, and RMSE evaluation. These metrics are defined at each time step as the means of $|\mathbf{X}_t - \hat{\mathbf{X}}_t|$ for MAE, $|\mathbf{X}_t - \hat{\mathbf{X}}_t|/|\mathbf{X}_t|$ for MAPE, and the square root of the mean of $\left(\mathbf{X}_t - \hat{\mathbf{X}}_t\right)^2$ for RMSE.
- **CV metrics:** PSNR and SSIM evaluation. The PSNR_t is defined at each time step t as $-10\log_{10}(\text{MSE}_t)$, with MSE_t being the mean of $\left(\mathbf{X}_t - \hat{\mathbf{X}}_t\right)^2$. The larger the PSNR, the better the prediction. The SSIM is defined at each time step by [84]:

$$SSIM_t = \frac{(2\mu_t\hat{\mu}_t + (0.01L)^2)(2\sigma_t\hat{\sigma}_t + (0.03L)^2)(cov_t + (0.0212L)^2)}{(\mu_t^2 + \hat{\mu_t}^2 + (0.01L)^2)(\sigma_t^2 + \hat{\sigma_t}^2 + (0.03L)^2)(\sigma_t\hat{\sigma}_t + (0.0212L)^2)}, \quad (18)$$

  where L is the dynamic range of the pixel values, usually $L = 255$; μ_t and σ_t are the mean and standard deviations of all possible windows of length 7 in the time step data $\mathbf{X}_t$, which are similar for $\hat{\mu}_t$ and $\hat{\sigma}_t$ for the predicted time step data $\hat{\mathbf{X}}_t$. cov_t is the covariance between all corresponding windows of length 7 on $\mathbf{X}_t$ and $\hat{\mathbf{X}}_t$. The SSIM has a maximum of 1 when $\mathbf{X}_t = \hat{\mathbf{X}}_t$ and quantifies the visual structure present in the one-dimensional graph [84].
- **Astrophysical features** evaluation: Twelve features defined in [72] are evaluated for IRIS data. For these data, each time step corresponds to an observed spectral line in a particular region of the Sun. The intensity, triplet intensity, line center, line width, line asymmetry, total continuum, triplet emission, k/h ratio integrated, k/h ratio max, k-height, peak ratio, and peak separation are the twelve measures on these spectral lines. These features provide insight into the nature of physics occurring at the observed region of the Sun. These metrics are evaluated at each time to show that the IB principle and a powerful CV metric are sufficient to provide reliable predictions in terms of physics.
- **The IB evaluation** is performed on centroid distributions in the prior $\mathbf{X}_{1:T}$, genuine $\mathbf{X}_{T+1:T+F}$, and predicted forecasts $\hat{\mathbf{X}}_{T+1:T+F}$. A k-means was performed in [55] for the spectral lines $\mathbf{X}_t$ that are to be predicted over time. The corresponding centroids C are used in this work to evaluate information theory measurements on the quantized data. Entropies for the prior $H(c_0)$, genuine $H(c_1)$, and predicted $H(c_2)$ distributions were averaged on the test data, and a comparison of the distributions between the prediction and the genuine was evaluated by computing the mutual information $I(c_1; c_2)$.
- The **classification accuracy** between the genuine and the forecast classifications was also evaluated. In the context of the IRIS data, three classes of solar activity are considered: QS, AR, and FL. Classifications are compared between the genuine target $\mathbf{X}_{T+1:T+F}$ and predicted forecast $\hat{\mathbf{X}}_{T+1:T+F}$, to assert whether the forecast activity complies with the targeted activity. *TSS* [76] and *HSS* [77] are evaluated globally and for each prediction class. These scores are defined in Section 2.8.

3.1. Evaluations of Predictions on IRIS Data

The model was trained on 240×240-sized images $\mathbf{X}_{1:240}$ representing MTS with 240 time steps and 240 features at each time step, and the last 25% $\mathbf{X}_{180:240}$ was masked at the input and predicted at the output. Each feature corresponds to a specific wavelength.

The data contain events of various durations. They were firstly partitioned into *training*, *validation*, and *testing* events; the model was tested on events that were not even partially seen at training time. All of the events were selected among those that last more than 240 time steps. Each event was divided into several 240×240 sized images $\mathbf{X}_{1:240}$ and paired with the corresponding 240×240 masks.

The model was trained on 12,738 images of size 240×240; 25% of the right part of each image was used as the target for prediction. In the image, each spectrum (column) was normalized, such that the maximum value was 1; this is compliant with [55], where the 53 clusters were obtained after the normalization of the spectra. We directly applied the trained model in order to predict the time sequence with the half-duration continuation of

the input; this is the *direct prediction* procedure. It was tested on 4490 *direct images*. To predict a longer continuation of the input, we adopted a straightforward common procedure and used a sliding 240×240 prediction window; this is the *iterated prediction* procedure. It was tested on 1962 *iterated images*. The evaluation was made by PSNR, SSIM, and the Hamming distance between the original and predicted corresponding cluster sequences. For this last metric, we adopted four different options. The cluster assignments are determined using a k-nearest-neighbor search at each time step, where the accuracy is measured by comparing if the data points have a common nearest centroid. NN_1 refers to $k = 1$ and NN_4 refers to $k = 4$.

Figure 7 presents the forecast by the proposed IB-MTS model and its evaluations for one flaring (FL) sample. The first row of results shows that the genuine and predicted sequences look very similar, with a high PSNR of 35.65 dB. Although, some magnified differences can be naturally observed.

The second row shows results for the prior, genuine, and predicted sequences, in terms of the assignment at each time step to the NN centroid obtained from a k-mean procedure described in [55]. Although the prior distribution differs from the genuine sequence to be predicted, the model was able to generate a sequence that exhibited a similar physical pattern to the genuine one. This shows that the model seems to be able to predict astrophysical patterns even with a CV-based loss that does not specifically measure astrophysical features.

The third row of results evaluates the usual MTS metrics for this specific FL sequence prediction. The three metrics (MAE, MAPE, and RMSE) have small averages over the forecasted times. Interestingly, the MTS errors are very small and below the averages for the first ten predicted time steps. This could be attributed to the temporal proximity with the prior information, as well as the combination of convolutions with the $\mathcal{L}_{tv}$ component of the loss defined in Equation (13), which softens the transition between the prior and the predicted data.

The last twelve plots show the time evolution of the astrophysical features extracted from the genuine (in blue) and predicted (in green) sequences. They show that the genuine features are predicted with more or less errors over time, but the predictions tend to follow the patterns of the genuine sequences. As these features were not considered in the loss used, this shows a non-negligible correlation between astrophysical features and CV metrics that were used to train the model.

Longer Predictions

For longer predictions, a standard well-known method performs recursive forecasts on data forecasted by the model. Under this setup, the prior MTS is a previously performed forecast, and one can legitimately expect an accumulation of errors. This is historically justified by the usage of one-step-forward models, such as the LSTM.

Figure 8 presents the short and long forecasts performed on a normalized IRIS quiet Sun (QS) sample. We can visually see that our approach is able to predict direct predictions as well as correct patters when we iterate the predictions on already predicted data.

3.2. MTS Metrics Evaluation

Table 2 presents the results of the evaluations of MAE, MAPE, and RMSE on IRIS, AR, and PB data. These metrics are averaged over all of the spectral and spatial dimensions, and all of the predicted time steps; for IRIS data, these metrics are also averaged over all types of solar activities. The model was trained and evaluated on each individual dataset. For all three metrics, our model outperforms the concurrent ones on the three datasets.

The descriptive temporal evolutions of these metrics are given in Figure 9 for the prediction errors under the *direct* setup, and in Figure 10 for the prediction errors under the *iterated* setup.

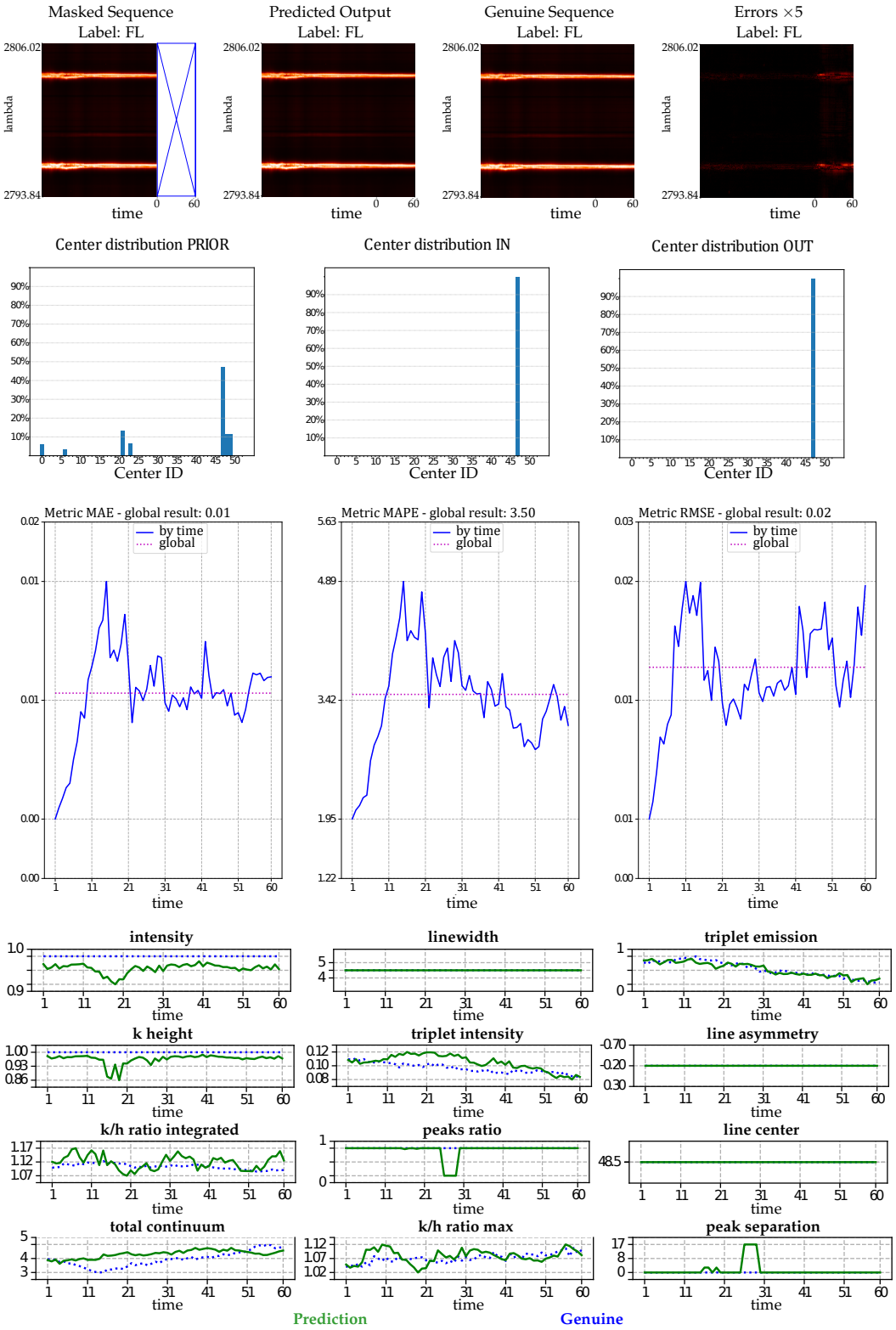


Figure 7. Evaluation of predictions for one flaring (FL) sample performed by the proposed IB-MTS model. The **first row** contains, respectively, the masked input, the predicted output, the genuine data, and the magnified pixel-wise error between the predicted and genuine. **Second row**: Spectral center distribution for the prior, the predicted, and the genuine MTS. **Third row**: MTS evaluation on the prediction. **Last twelve plots**: astrophysical features evaluations; the dotted blues represent the genuine and the green lines represent the prediction.

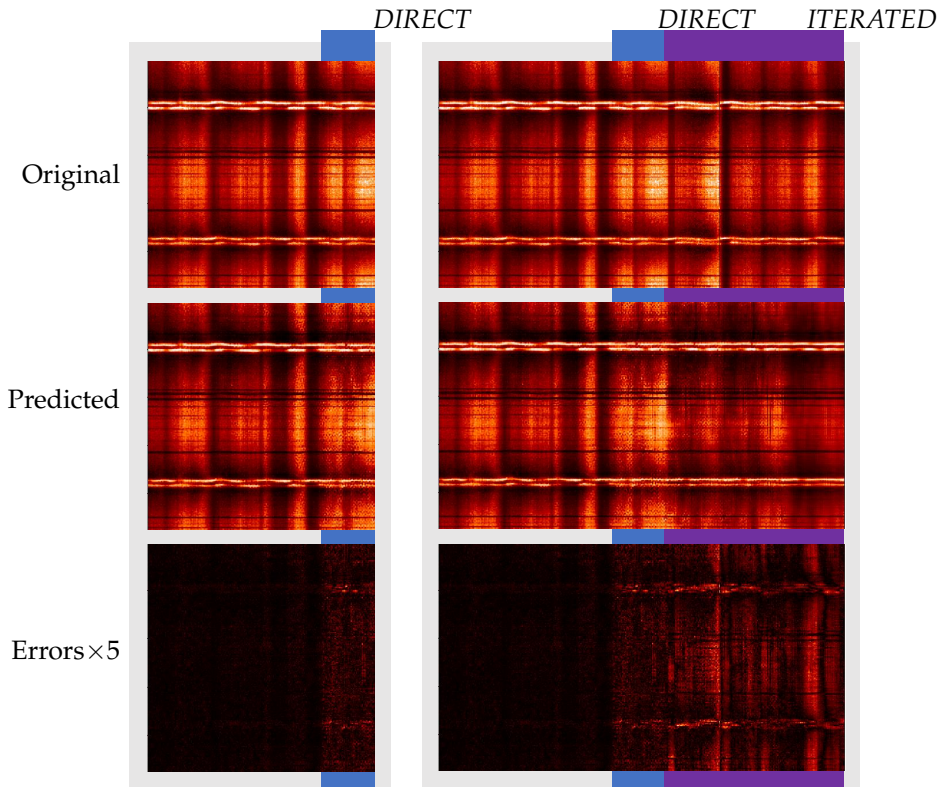


Figure 8. Prediction results: The first column presents the results of the direct predictions (blue part) and the second column presents the iterated predictions (violet part). A masked sample is given from the original sequence (**first row**); the prediction (**second row**) and the magnified (×5) differences (**third row**) are shown.

Table 2. MTS metric results.

Data		Model	IB-MTS	LSTM	ED-LSTM	GRU	ED-GRU	NBeats
IRIS	*direct*	*MAE*	**0.04**	0.05	0.05	0.05	**0.04**	0.10
		MAPE	**2.76**	13.13	4.71	26.84	3.16	4.75
		RMSE	**0.07**	0.08	0.08	0.08	**0.07**	0.19
	iterated	*MAE*	**0.05**	0.06	**0.05**	0.06	**0.05**	0.13
		MAPE	**2.94**	12.35	3.53	26.32	3.22	6.47
		RMSE	**0.08**	0.09	0.09	0.10	**0.08**	0.22
AL	*direct*	*MAE*	**0.08**	0.10	**0.08**	0.10	0.09	0.11
		MAPE	**3.71**	5.27	4.58	5.50	5.20	6.56
		RMSE	0.15	0.16	**0.14**	0.16	0.16	0.18
	iterated	*MAE*	**0.08**	0.19	0.16	0.23	0.16	0.11
		MAPE	**4.00**	11.37	9.10	12.94	9.28	6.23
		RMSE	**0.15**	0.26	0.23	0.30	0.23	0.18
PB	*direct*	*MAE*	**0.19**	0.46	0.46	0.50	0.46	0.22
		MAPE	**4.47**	10.15	10.03	10.76	10.14	5.19
		RMSE	**0.24**	0.54	0.51	0.60	0.52	0.28
	iterated	*MAE*	0.24	0.45	0.45	0.45	0.45	**0.23**
		MAPE	**4.23**	10.00	9.98	10.01	9.98	5.51
		RMSE	0.30	0.51	0.500	0.51	0.50	**0.28**

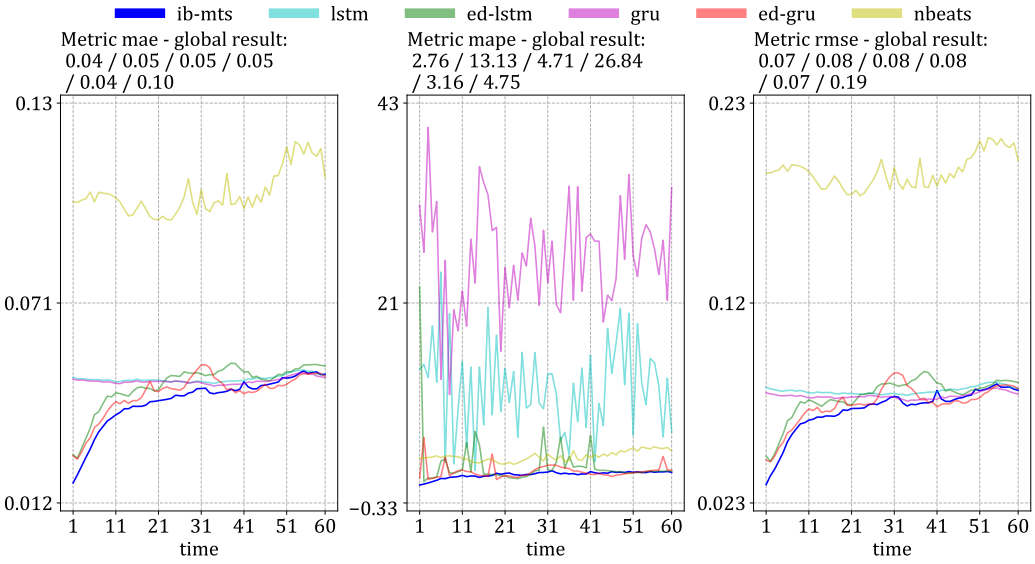


Figure 9. MTS metrics evaluation averaged on the test set for the direct prediction setups on QS, AR, and FL IRIS data.

For the *direct* setup of Figure 9, our proposed model performs better than all of the concurrent ones in more than half of the time steps, and there is even a gap in the performance for the first 30 time steps. The traditional LSTM and GRU have very bad performances in the first 20 time steps compared to the other models. The three models with IB formulations, which propose compressions of the time dimension similar to ED-LSTM and ED-GRU, demonstrate similar performances with a performance gap in the first 20 time steps compared to the proposed IB-MTS. However, ED-GRU manages to recover this gap and even performs slightly better in the last 20 time steps in terms of MAE and RMSE. NBeats does not perform well on the IRIS dataset; the results for MAE and RMSE are among the worst, whereas the MAPE results are average. This means that NBeats produces predictions with significant errors for high values, but it is highly accurate for small values. One reason for this could be the small number of trainable parameters for NBeats, compared with the complexity of the IRIS dataset. In simpler datasets, such as AL and PB, NBeats can achieve the second-best results after IB-MTS and the best results on iterated predictions on the PB dataset. One possible explanation for the weak performances of NBeats on IRIS datasets could be the complexity and the strong spatiotemporal dependencies of spectral data, where the spatiotemporal dependencies are less clear on AL and PB data, with sensors being sorted by their latitudes, ignoring their longitudes. For the MAPE metric, the proposed IB-MTS still performs better than the other ones, even with the last time steps. One interesting fact is that even if the performances are quite far in the first part of the time steps, the classical LSTM and GRU seem to have fewer error variabilities than the ED-LSTM and ED-GRU for MAE and RMSE metrics. This could be explained by the specific designs of NBeats for long-term predictions, suggesting that other currently designed IB models may close the performance gap over the long term while still providing comparable results. In contrast to these conclusions, concerning the MAPE metric, the classical LSTM and GRU models have a lot of error variability and perform worse than the IB-designed models. The variability could be due to larger errors for small values because these types of errors have significant impacts on the MAPE metric.

Concerning the last 20 time steps, Figure A1 from Appendix C can provide some explanation for the gain of performance in terms of the MAE and RMSE of the ED-GRU. This figure provides details of evaluations on the three different types of data included in the training and test data, i.e., QS, AR, and FL data, representing, respectively, 45.6%, 26.2%, and 28.2% of the (short) data. The figure shows that our proposed IB-MTS performs better than all of the others in all of the time steps for QS data, and is the most represented class in the trained and tested data. The other IB models still perform worse than IB-MTS

for the first 30 time steps but they perform better in terms of MAE and RMSE in AR and FL data for the last 20 time steps.

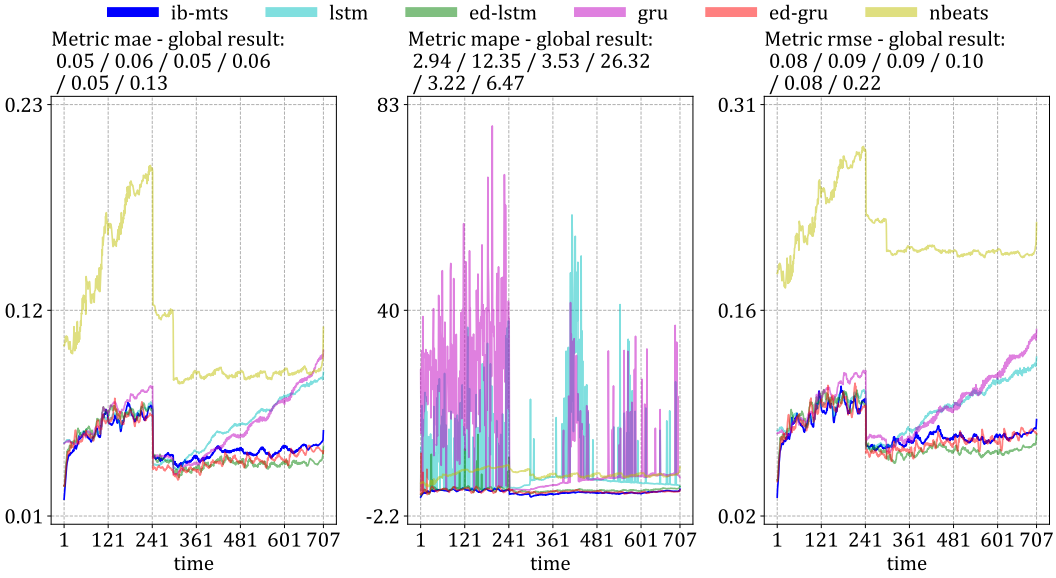


Figure 10. MTS metrics evaluation averaged on the test set for the iterated prediction setups on QS, AR, and FL IRIS data.

Figure 10 presents the evaluations of the MTS metrics in the *iterated* setup, where the model predicts 60 time steps ahead on data that have already been predicted. For this procedure, the proposed IB-MTS does not show a specific improvement in performance, except for the MAPE metric and the first 20 time steps, where the first *direct* prediction is performed. It is also important to note that the traditional LSTM and GRU models have high accumulated errors over time and perform much worse than the others in the last 200 time steps. The specific drop in the curves at 241 time steps is directly related to the variation in the length of the MTS within the testing set.

Figure 11 shows a histogram of the number of QS, AR, and FL data present in the test data by the number of total time steps; the majority of FL data have predicted durations greater than 700 time steps. On the contrary, the majority of QS and AR data have predicted durations smaller than 300 time steps.

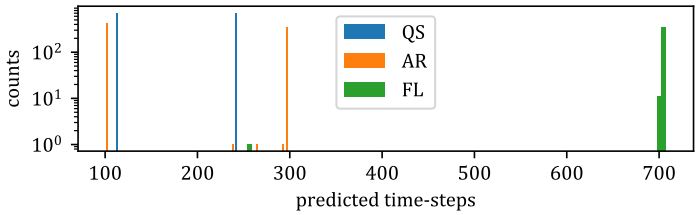


Figure 11. Histogram of the event durations from IRIS data.

Figure A2 from Appendix C presents a detailed comparison of the MTS evaluations for each class of *iterated* data. These graphs firstly explain the specific shapes present in Figure 10 by the fact that no QS data are present after 242 predicted time steps and no AR data are present after 298 predicted time steps. The proposed IB-MTS still performs very well on *iterated* QS data for all of the metrics, as well as for the MAPE metric on all types of data. ED-LSTM and ED-GRU perform better than IB-MTS after 120 time steps for MAE and RMSE metrics. One interpretation of this could be that the proposed IB-MTS is not designed to handle predictions on predictions, and does not include a state channel, such as in LSTM of GRU. Moreover, many variabilities are present for LSTM and GRU on the MAPE metric, whereas the IB-MTS model remains consistently stable and outperforms the

other models in all scenarios. This could be attributed to the errors present for small values because they impact a lot of the results of the MAPE. In fact, an error of 0.1 for a value of 0.2 has a MAPE of $100 \times 0.1/0.2 = 50$, whereas the same error of 0.1 for a value of 0.8 has a MAPE of $100 \times 0.1/0.8 = 12.5$.

3.3. Computer Vision Metrics Evaluation

For the two procedures, *direct* and *iterated*, as described in Section 3.1, Table 3 provides the results of the evaluations in terms of PSNR and SSIM averaged on the test data. The proposed IB-MTS performs better than the concurrent models on IRIS data for both procedures. The gain of performance is very sensible for *direct* predictions. For *iterated* predictions, the results are more grouped and the ED-GRU performs similar to IB-MTS. On AL data, IB-MTS also outperforms concurrent models, except for the PSNR on *direct* data, where ED-LSTM performs a bit better.

Table 3. Accuracy in terms of average PSNR and SSIM for *direct* and *iterated* predictions.

Dataset		Metric	*IB-MTS*	*LSTM*	*ED-LSTM*	*GRU*	*ED-GRU*	*NBeats*
IRIS	*direct*	PSNR	**27.2**	25.6	26.3	25.9	26.7	14.6
		SSIM	**0.897**	0.869	0.887	0.864	0.891	0.673
	iterated	PSNR	**23.8**	23.0	23.4	22.8	**23.8**	13.4
		SSIM	**0.868**	0.821	0.864	0.809	**0.868**	0.586
AL	*direct*	PSNR	17.2	16.0	**17.4**	15.9	16.4	15.7
		SSIM	**0.518**	0.400	0.488	0.377	0.401	0.346
	iterated	PSNR	**16.7**	11.8	13.0	10.6	13.1	15.2
		SSIM	**0.516**	0.046	0.198	0.023	0.166	0.361
PB	*direct*	PSNR	**12.5**	5.4	6.0	4.5	5.7	11.4
		SSIM	0.361	0.013	0.000	0.004	0.000	**0.470**
	iterated	PSNR	10.5	5.9	6.0	5.9	6.0	**11.0**
		SSIM	0.235	0.003	0.003	0.003	0.004	**0.472**

NBeats still fails on IRIS data for the CV metric evaluation but performs well on the simpler datasets (AL and PB). NBeats can even achieve the highest SSIM on PB data. Yet our proposed IB-MTS model is the most stable in terms of the results on various datasets and it outperforms the concurrent ones on almost all metrics and datasets. NBeats also performs the best for the *iterated* procedure on PB data. Our proposed IB-MTS model was designed to predict multiple steps ahead in a *direct* fashion and does not generally extend very well for the *iterated* procedure.

The first row of Figure 12 provides detailed time evolutions of these metrics on IRIS *direct* data. They show a sensible gain of performance for the first 20 time steps, whereas the classical LSTM and GRU models do not have specific time evolutions in terms of performance and they perform worse than the others.

The second row of Figure 12 provides detailed time evolutions of the PSNR and SSIM metrics on the IRIS *iterated* data. ED-LSTM and ED-GRU perform a bit better than IB-MTS under the *iterated* setup for more than 300 time steps whereas the classical LSTM and GRU models perform worse. The specific variation of the curve at 300 time steps is explained by the diversity of the MTS durations evoked in Figure 11.

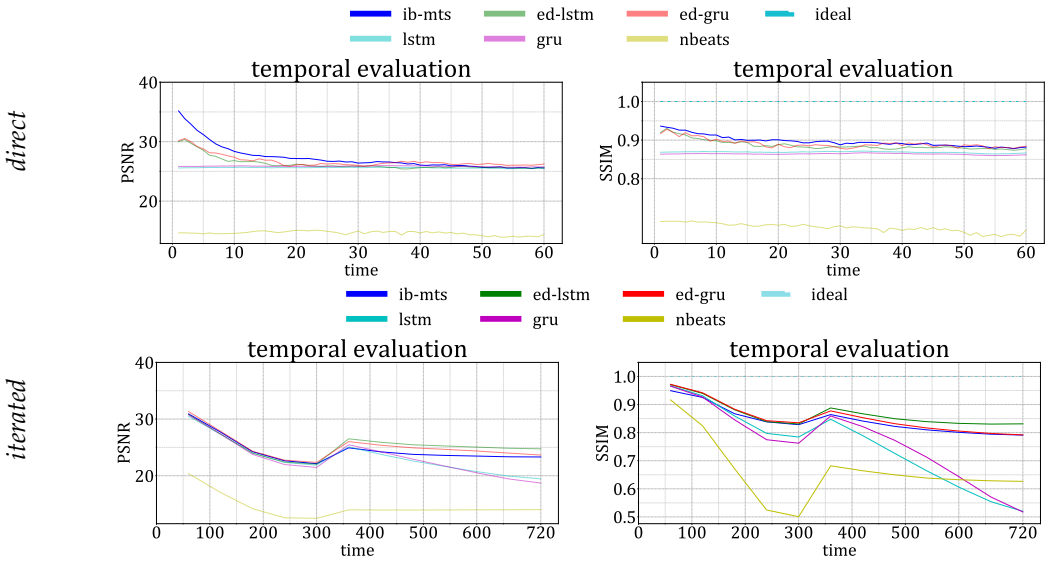


Figure 12. CV evaluation (over time) of the forecast for the direct and iterated predictions on IRIS data.

3.3.1. Information Bottleneck Evaluation on IRIS Data

This subsection evaluates IT-related measures on IRIS prior $\mathbf{X}_{1:180}$, genuine $\mathbf{X}_{181:240}$, and predicted $\tilde{\mathbf{X}}_{181:240}$ data, as explained in Figure 6. Entropies $H(\cdot)$, KL-divergences $KL(\cdot||\cdot)$, and mutual information $I(\cdot,\cdot)$ are estimated by using the centroids obtained by a version of the k-means process performed in [55]. The dictionary of these 53 centroids is publicly available (i4ds.github.io/IRISreader/html/centroid_data.html, accessed on 20 February 2023). Each prior sequence $\mathbf{X}_{1:180}$ contains 180 centroids, one for each time step, with repetitions because the dictionary of centroids only has 53 centroids. In the same manner, each genuine $\mathbf{X}_{181:240}$ and predicted $\tilde{\mathbf{X}}_{181:240}$ sequence contains 60 centroids, with repetitions.

Figure 13 shows the average distributions for prior, genuine, and predicted data. As expected, the average prior and genuine distributions of centroids are very similar and correspond to averages of the observed data. The average distribution of the predicted data remains close to that of the observed data, with only a few deviations in means and standard deviations. In particularly, centroid numbers 42 and 49 are a bit over-represented in the predictions whereas centroid numbers 20 and 44 are a bit under-represented. More details are given in Appendix C with the joint probabilities $p(c_1, c_2)$ of the centers present in the genuine c_1 and pred c_2 forecasts. Figure A3 presents the direct setup of IRIS data and Figure A4 presents the iterated setup. High probabilities on the diagonal $c_1 = c_2$ indicate a high accuracy of the prediction in terms of centroids. These figures show that our proposed IB-MTS ensures the high conservation of the physics behind the spectra, whereas NBeats totally fails in this task.

Table 4 presents the average IT measurements estimated on the prior, genuine, and predicted data. c_0 stands for the centroids present in prior data, c_1 and c_2 are the ones present in genuine and predicted data. Ideally, $KL(c_0||c_2) = KL(c_0||c_1)$, and $H(c_2) = H(c_1) = I(c_1, c_2)$. The higher the $I(c_1; c_2)$, the better. The results show that IB-MTS predicts the best c_2 centroids. This is confirmed by the highest value of mutual information $I(c_1; c_2)$ between the genuine and prediction data. Moreover, the KL-divergence $KL(c0||c_2)$ between the prior and the prediction data is the closest to $KL(c0||c_1)$ between the prior and the genuine data for IB-MTS on *direct* data. The interpretation of a high $I(c1; c2)$ is that the IB-MTS model is the best at predicting the information present in the prediction, even without the need for a lot of extra information beyond what is given in the prior, because $KL(c_0||c_2)$ is the lowest and closest to $KL(c_0||c_1)$. As a consequence, it seems that IB-MTS is the model that best adheres to the IB principles. NBeats failed to predict the correct centroids. The KL-divergence $KL(c0||c_2)$ between the prior and the prediction is the closest

to $KL(c0||c_1)$ but equal to 0; the mutual information $I(c_1;c_2)$ between the genuine and the prediction is 0, which means that, on average, it could not predict the information of the target. One reason for this could be that the intensity information is very important for the NBeats process on IRIS data. The fact that we removed this information from the dataset by normalizing each time step by the maximum values may penalize NBeats on the IRIS data, which is not the case with AL and PB data, where NBeats performs fine.

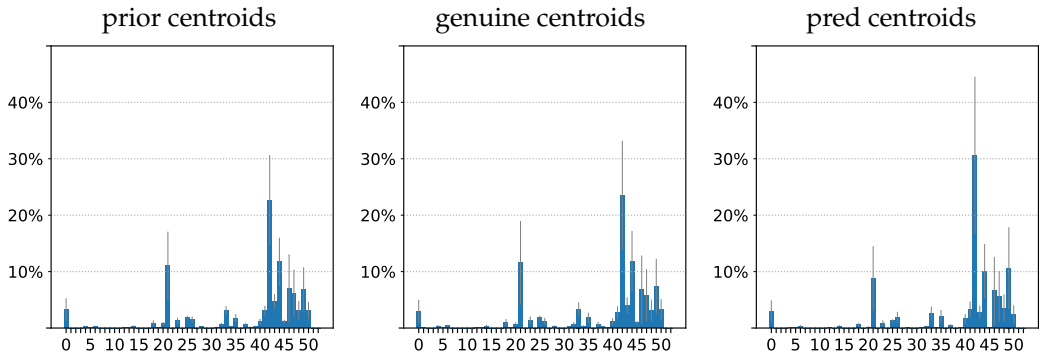


Figure 13. Average distributions of centroids with their standard deviations as vertical gray error bars. The first graph is for the average prior central data, the middle graph is for the average genuine target, and the right graph is the average distribution of predictions performed with IB-MTS.

Table 4. Information comparison between the prior centroids $c0$, the genuine centroids $c1$, and the predicted centroids $c2$ for the IRIS dataset. The results on H, KL, and I are averaged over all testing samples. $H(c_0)$, $KL(c_0||c_1)$, and $H(c_1)$, being statistics on the prior and the genuine, do not depend on the method.

Dataset		Metric	*IB-MTS*	*LSTM*	*ED-LSTM*	*GRU*	*ED-GRU*	*NBeats*
IRIS	*direct*	$H(c_0)$	3.922	3.922	3.922	3.922	3.922	3.922
		$KL(c_0\|\|c_1)$	0.005	0.005	0.005	0.005	0.005	0.005
		$KL(c_0\|\|c_2)$	0.111	0.311	0.111	0.344	0.198	0.000
		$H(c_1)$	3.890	3.890	3.890	3.890	3.890	3.890
		$H(c_2)$	3.567	3.382	3.579	3.280	3.465	0.000
		$I(c_1;c_2)$	**1.753**	1.607	1.613	1.597	1.681	0.000
	iterated	$H(c_0)$	3.968	3.968	3.968	3.968	3.968	3.968
		$KL(c_0\|\|c_1)$	0.003	0.003	0.003	0.003	0.003	0.003
		$KL(c_0\|\|c_2)$	**0.288**	0.575	0.308	0.416	0.486	0.000
		$H(c_1)$	3.957	3.957	3.957	3.957	3.957	3.957
		$H(c_2)$	3.487	3.325	3.385	3.301	3.229	0.000
		$I(c_1;c_2)$	**1.352**	1.276	1.319	1.219	1.314	0.000

Concerning the *iterated* data, the highest values of mutual information $I(c1;c_2)$ are still obtained by the IB models. IB-MTS can achieve the highest mutual information $I(c_1,c_2)$ with the lowest $KL(c_0||c_1)$.

3.3.2. Astrophysical Evaluations

A public dictionary of 53 centroids obtained by a version of k-means was performed on IRIS Mgh&k data [55]. Moreover, astrophysical features defined in [72] allow interpretability of Mghk spectra. This part evaluates the correspondence between centroid assignments for each genuine and predicted time step, and also evaluates the relative time evolution of the error between the genuine and predicted time steps. A k-NN search was performed between the genuine spectra at a given time step and the dictionary consisting of 53 centroids. The same search was performed for the same time step between the predicted spectra and the dictionary, and two sets of k centroids were compared. The considered time steps are successful k-NN assignments when they have at least one center in common.

For the two procedures (*direct* and *iterated*) described in Section 3.1, Table 5 provides the results of the prediction in terms of k-NN metrics averaged on the test data. For comparison, in the third column, we provide the accuracies of a theoretical worst model, which randomly assigns the k-nearest-neighbors among the 53 found in [55]. The most pessimistic accuracy for random classifications is obtained by the following combinatorial calculation:

$$\text{Random}_{k\text{-NN}} = \frac{\binom{53}{k} - \binom{53-k}{k}}{\binom{53}{k}}. \tag{19}$$

For $k = 2$, the proposed IB-MTS model can already predict the sequence of clusters with 82% of accuracy, and even 99% for $k = 5$, whereas a random assignment would give corresponding accuracies of 7.5% and 40%. The NBeats model fails on IRIS data, performing accuracies lower than the random assignments and not being able to predict spectra with the same centroid assignment as the genuine. Figure 14 presents the time evolution of the average k-NN accuracy for the *direct* procedure. There is a clear gain in performance for IB-MTS in the first 20 time steps for 1-NN and 2-NN accuracies. When k is greater than 4, the accounted performances are very similar for all time steps.

Figure 15 presents the time evolution of the average k-NN accuracy for the *iterated* procedure. Similar to the other metrics, the IB-MTS model shows an average performance for the *iterated* procedure, while the other IB models and NBeats perform better when predictions are made iteratively on predicted data.

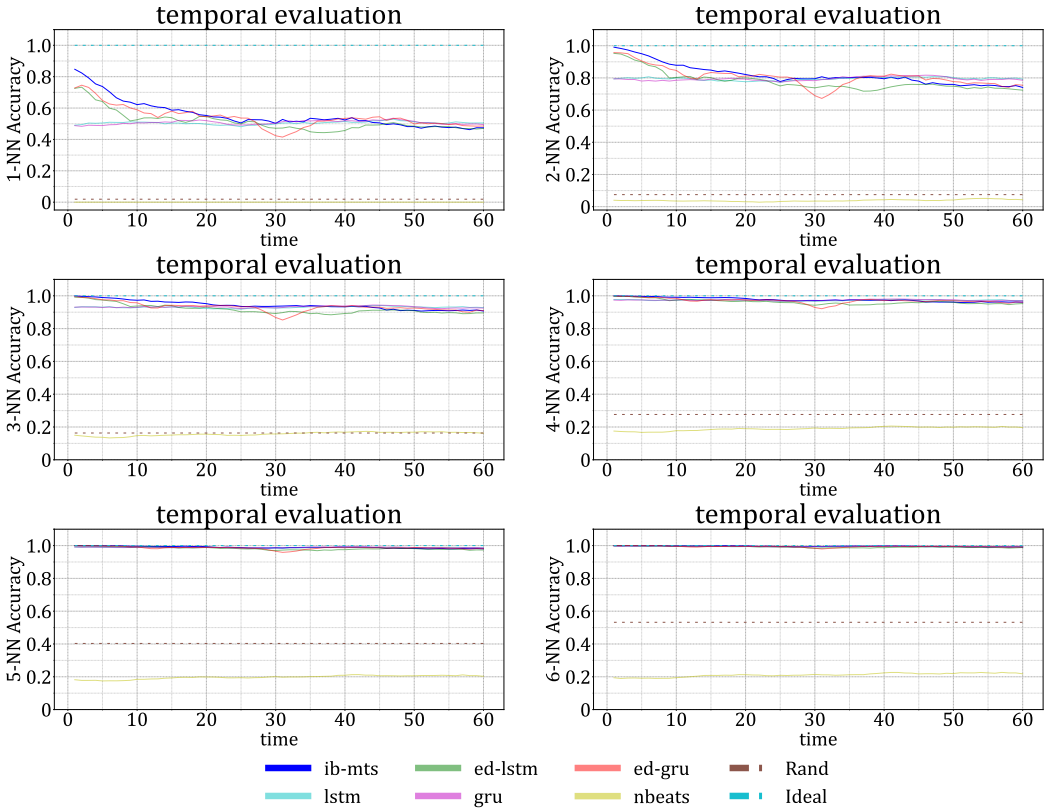


Figure 14. IRIS center assignment evaluation (over time) of the forecasts for direct predictions.

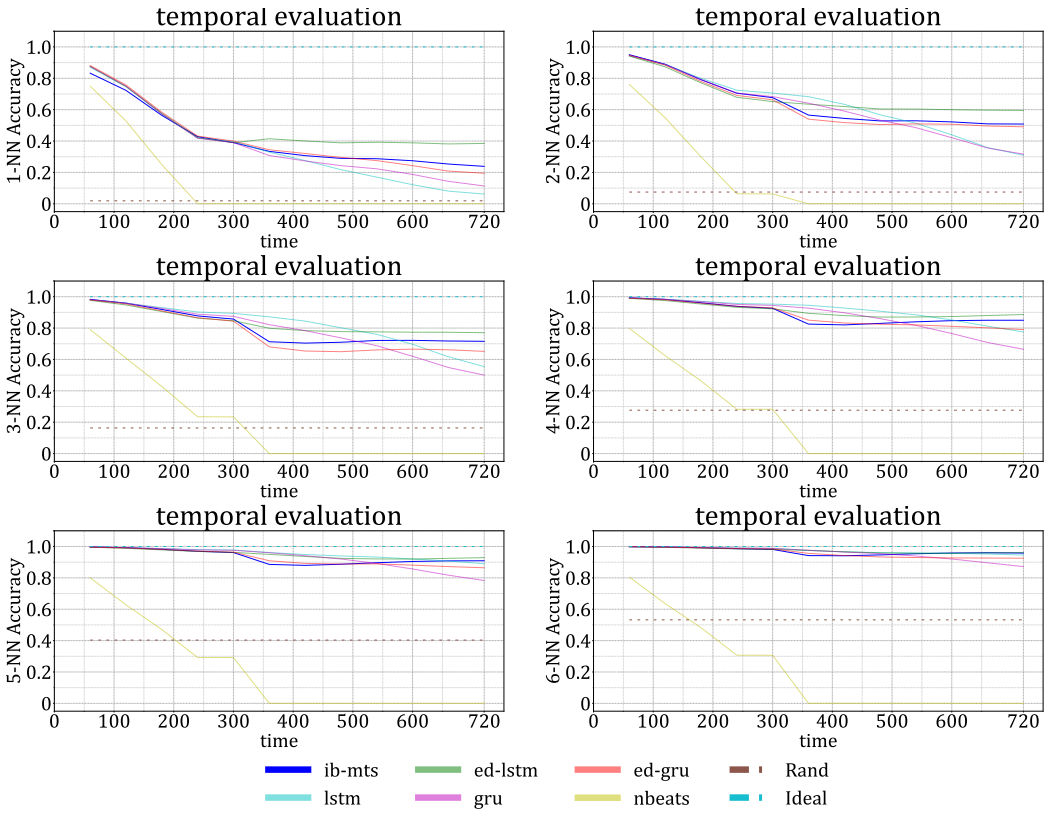


Figure 15. IRIS center assignment evaluation (over time) of the forecasts for iterated predictions.

Table 5. Evaluation on IRIS data: Percentage accuracies in terms of *k*-NN for *direct* prediction of the sizes of the training data and *iterated* prediction using a basic sliding window approach. The random *k*-NN cluster assignment accuracy is given for comparison and corresponds to the worst that can be expected for each *k*-NN assignment.

	Metric	Rand$_{k\text{-NN}}$	IB-MTS	LSTM	ED-LSTM	GRU	ED-GRU	NBeats
direct	1-NN	1.9	**55.7**	50.5	51.4	50.8	54.1	0.0
	2-NN	7.5	**81.9**	79.5	77.8	79.4	80.5	3.8
	3-NN	16.3	**94.3**	93.0	91.8	93.2	93.1	15.8
	4-NN	27.6	**97.7**	97.3	96.3	97.4	97.0	19.0
	5-NN	40.3	**98.9**	98.8	98.2	**98.9**	98.6	19.7
	6-NN	53.2	**99.6**	99.5	99.1	99.5	99.4	21.1
iterated	1-NN	1.9	**45.8**	42.6	43.6	43.4	45.0	0.0
	2-NN	7.5	**73.8**	72.1	70.1	72.0	72.4	3.9
	3-NN	16.3	**89.4**	88.9	87.0	88.3	87.8	16.1
	4-NN	27.6	95.1	**95.2**	93.7	94.5	94.4	19.4
	5-NN	40.3	97.6	**97.8**	96.8	97.3	97.2	20.2
	6-NN	53.2	98.9	**99.0**	98.5	98.6	98.7	21.8

Table 6 provides detailed data on 1-NN center assignment accuracy, as well as HSS and TSS metrics, broken down by label and aggregated for all the data. Concerning the *direct* procedure, IB-MTS performs the best on each label, as well as overall, and ED-GRU obtains comparable results on AR data only. The results for the *iterated* procedure show that IB-MTS performs less favorably compared to other IB models.

The time evolutions of the average relative errors for the astrophysical features are given in Figure 16. Given that each feature of a spectrum is a scalar output of a deterministic function $f_k(\cdot)$, these relative errors are defined at each time step by:

$$ref_{k,t} = \frac{|f_k(\mathbf{X}_t) - f_k(\tilde{\mathbf{X}}_t)|}{f_k(\mathbf{X}_t)}, \tag{20}$$

where $\tilde{\mathbf{X}}_t$ is the time step t estimation of $\mathbf{X}$ by the model. IB-MTS is able to predict the physical features relatively well, with a significant gain of performance in the first 20 time steps in almost all time steps. This is an important result because features, such as the line center, k/h ratios, k-height, and peak separation are related to the positions of specific local maximums on the spectra, and the corresponding functions are not differentiable. We provide experimental proof that IB-MTS is able to predict features that are not easy to integrate in the loss of a deep model. NBeats have high errors in predicting intensities. This is coherent with the comments formulated in Section 3.3.1. It seems that NBeats cannot function without intensity information on IRIS data, which could be the reason for its failure on these data.

Figure 17 shows the time estimations for the relative errors of features under the *iterated* setup. IB-MTS has an average performance under this setup, other IB models perform better, and classical LSTM and GRU performing worse.

Table 6. Evaluation of 1-NN centroid assignment accuracy for the *direct* and *iterated* predictions.

Model	Metric	IB-MTS	LSTM	ED-LSTM	GRU	ED-GRU	N-BEATS
				direct			
Global	% Accuracy	**55.7**	50.5	51.4	50.8	54.1	0.0
	TSS	**0.49**	0.43	0.45	0.43	0.47	0.00
	HSS	**0.50**	0.44	0.45	0.44	0.48	0.00
QS	% Accuracy	**52.5**	47.6	47.7	48.6	49.0	0.0
	TSS	**0.26**	0.18	0.21	0.19	0.22	0.00
	HSS	**0.28**	0.20	0.22	0.20	0.22	0.00
AR	% Accuracy	49.5	44.8	45.7	46.0	**49.9**	0.0
	TSS	**0.43**	0.37	0.39	0.39	**0.43**	0.00
	HSS	**0.43**	0.37	0.39	0.39	**0.43**	0.00
FL	% Accuracy	**63.7**	57.3	60.3	56.3	63.5	0.0
	TSS	**0.58**	0.51	0.53	0.49	0.57	0.00
	HSS	**0.58**	0.51	0.53	0.49	0.57	0.00
				iterated			
Global	% Accuracy	40.4	36.4	**41.8**	37.4	40.0	0.0
	TSS	0.33	0.29	**0.35**	0.30	0.32	0.00
	HSS	0.34	0.29	**0.35**	0.30	0.32	0.00
QS	% Accuracy	**46.3**	43.4	42.7	44.7	43.9	0.0
	TSS	**0.15**	0.10	0.12	0.10	0.12	0.00
	HSS	**0.17**	0.11	0.13	0.12	0.14	0.00
AR	% Accuracy	37.2	38.2	34.8	39.0	**41.3**	0.0
	TSS	0.30	0.30	0.26	0.31	**0.33**	0.00
	HSS	0.30	0.30	0.26	0.31	**0.33**	0.00
FL	% Accuracy	33.0	24.8	**42.6**	26.3	31.8	0.0
	TSS	0.24	0.18	**0.33**	0.18	0.22	0.00
	HSS	0.24	0.17	**0.33**	0.17	0.22	0.00

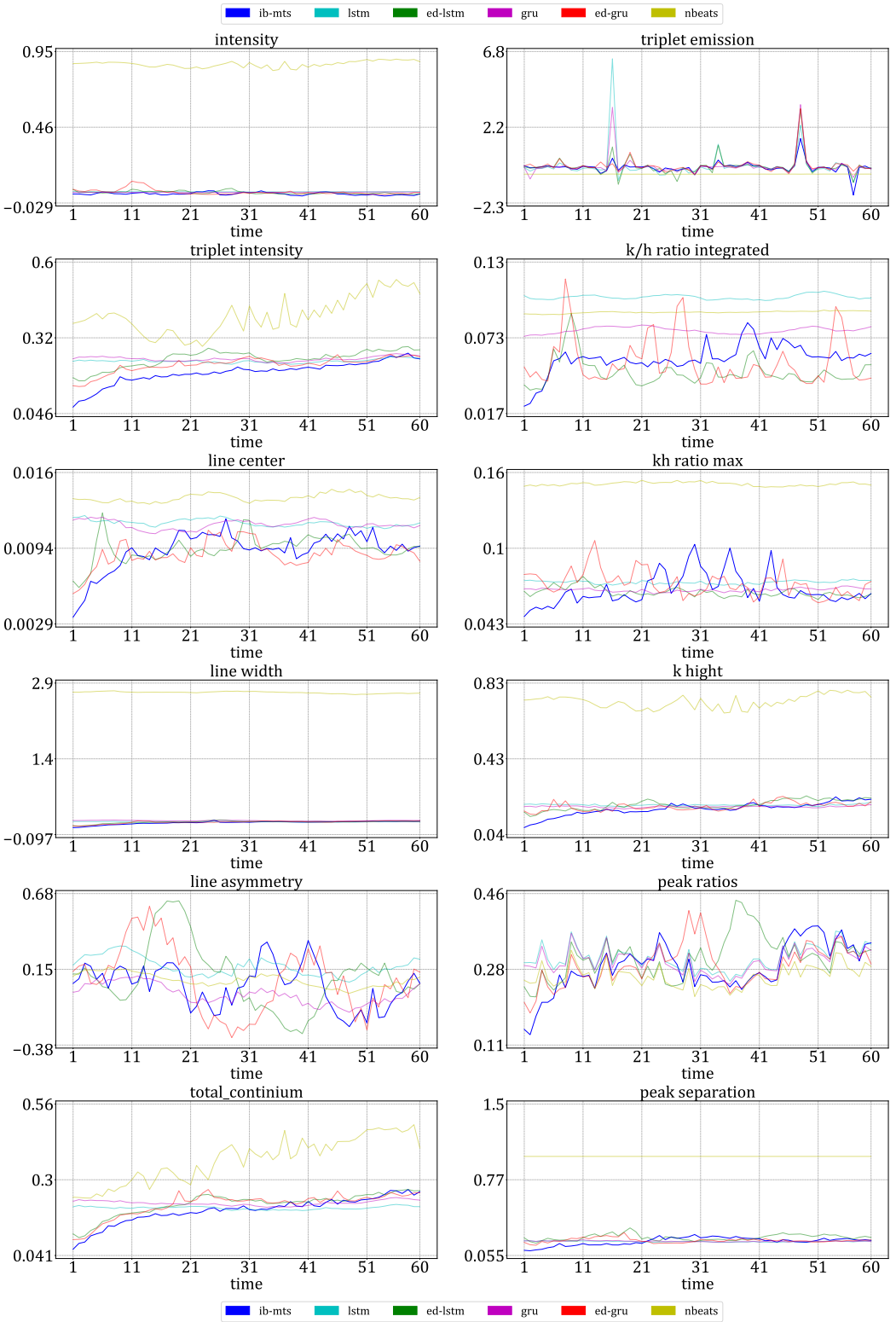


Figure 16. Evaluation of the relative prediction errors for physical features over time of the forecasts for IRIS data and the *direct* setup. The lower the better.

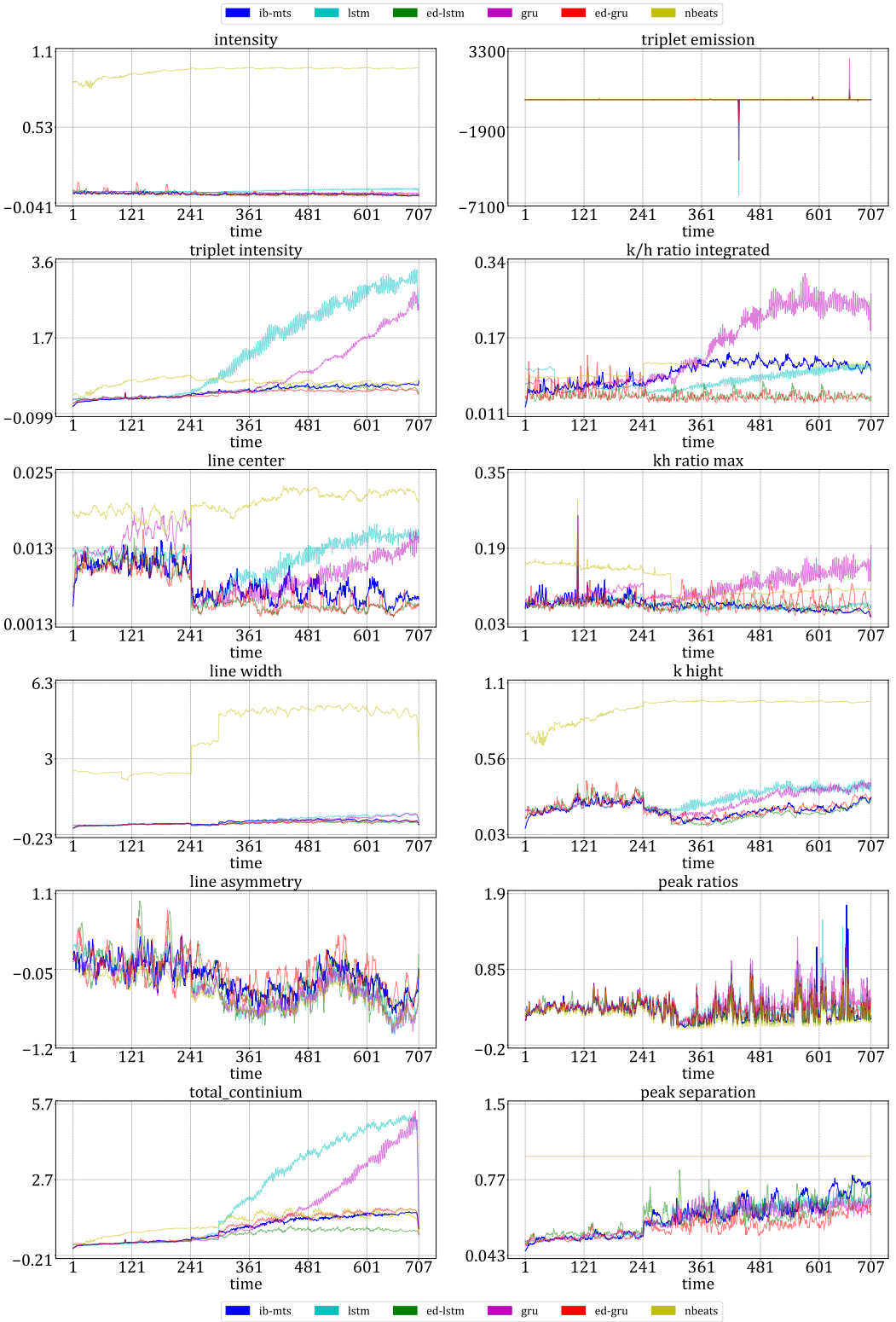


Figure 17. Evaluation of the relative prediction errors for physical features over time of the forecasts for IRIS data and the *iterated* setup.

3.3.3. Solar Activity Classification

This section investigates the solar activity classifications of test data. Part of the IRIS Mghk data were labeled by the type of activity, such as QS, AR, and FL. The labeling was performed globally for each time sequence and not at each time step, such that if an event

presents a flaring FL episode between time steps t_0 and t_1, the sample $\mathbf{X}_{1:T+F}$ will be labeled as FL ($1 < t_0 < t_1 < T + F$).

A previously trained classifier was used to classify the genuine and predicted sequences, $\mathbf{X}_{T+1:T+F}$ and $\tilde{\mathbf{X}}_{T+1:T+F}$. This classifier outputs a vector of size 3 with a categorical assignment. The assigned class corresponds to the minimum of the cosine pseudo-distance between the output of the classifier and the one-hot encoded version of the class. The classification is considered successful when they match. For the *iterated* procedure, the classification is estimated by the minimum cosine pseudo-distance with the average of the categorical outputs obtained by the classifier at each iteration. Table 7 shows the results of the classification in terms of percentage accuracy, HSS, and TSS metrics. For the short and long procedures, there are no specific performance gains for IB-MTS compared to the concurrent models. All models show high activity classification performances, except NBeats; previous results sections showed that the model failed to predict on the IRIS normalized data. IB-MTS has slightly higher TSS and HSS scores than other models.

Table 7. Accuracy of solar activity classifications for the predicted versus genuine MTS with the *direct* and *iterated* prediction setups.

Model (Count)	Metric	IB-MTS	LSTM	ED-LSTM	GRU	ED-GRU	N-BEATS
				direct			
Global (8000)	% Acc	**95**	**95**	**95**	**95**	**95**	88
	TSS	**0.911**	**0.911**	0.906	0.910	0.909	0.805
	HSS	**0.915**	0.906	0.911	0.905	0.914	0.785
QS (3680)	% Acc	**97**	96	96	96	96	94
	TSS	**0.938**	0.911	0.916	0.910	0.918	0.876
	HSS	**0.936**	0.911	0.915	0.910	0.917	0.874
AR (536)	% Acc	96	96	**97**	96	**97**	91
	TSS	**0.613**	0.401	0.327	0.400	0.311	0.000
	HSS	**0.640**	0.371	0.366	0.362	0.349	0.000
FL (3784)	% Acc	98	**99**	98	**99**	**99**	92
	TSS	0.958	0.971	0.965	**0.972**	**0.972**	0.838
	HSS	0.959	0.971	0.965	**0.972**	**0.972**	0.843
				iterated			
Global (8000)	% Acc	94	94	93	94	**95**	86
	TSS	**0.979**	0.899	0.870	0.896	0.895	0.768
	HSS	0.889	0.891	0.869	0.889	**0.901**	0.738
QS (3680)	% Acc	**96**	95	95	95	95	88
	TSS	**0.914**	0.903	0.891	0.892	0.907	0.777
	HSS	**0.915**	0.903	0.893	0.891	0.908	0.767
AR (536)	% Acc	94	95	95	**96**	**96**	92
	TSS	**0.544**	0.387	0.188	0.399	0.277	0.168
	HSS	**0.594**	0.331	0.185	0.346	0.311	0.113
FL (3784)	% Acc	**98**	**98**	97	**98**	**98**	91
	TSS	0.948	0.955	0.930	**0.960**	0.957	0.932
	HSS	0.949	0.957	0.930	**0.962**	0.957	0.920

4. Discussion

4.1. Conclusions

Our proposed model is not perfectly fine-tuned but it already shows very competitive results. Moreover, the integration of the total loss of $\mathcal{L}_1$ and $\mathcal{L}_2$ from Equation (6) will allow for better optimization of the full IB loss from Equation (4). The main goal of this paper was to provide a new theoretical formulation of the IB in the context of MTS, i.e., to bring theoretical and empirical proofs that are not only problems of multiple successive IBs, but unique, joint spatiotemporal IBs. As a consequence, these results and comparisons are convincing and support the presented IB formulation for MTS forecasting. Moreover, we

show that IB models, which utilize compression through encoding and decoding of the time dimension, also perform better than classical recurrent models.

The gain of performance is also very significant for the first 20 time steps across almost all evaluated metrics. This may be due to the combination of convolutions and the transition term $\mathcal{L}_{tv}$ present in the loss in Equation (13), which smooths out the predictions at the border for the first predicted time steps.

4.2. Spatial Sorting of MTS Data

Spatial dimensions in AL and PB are sorted by increasing latitudes, while the IRIS spatial dimension is sorted by increasing the wavelength. Other possible orderings for AL and PB datasets involve increasing longitudes or distances from a specific coordinate. The spatial relationship of the time series in the AL dataset is closely associated with the spatial correlation of the weather and sunshine levels. The spatial relation of the time series in the PB dataset is linked to the connection with the road networks around San Francisco.

Other works [1,17,18,60] highlight the limitations of the models that only consider spatial and temporal dependencies separately. Spatiotemporal dependencies are locally modeled with convolutions on spatiotemporal pseudo-images. Moreover, the spatiotemporal dimensions are compressed by applying successive convolutions with a stride of 2 in the first half of the U-Net architecture. Because of this spatiotemporal compression structure, each hidden layer of the encoding part of the U-Net creates a more global model of the spatiotemporal dependencies, such that, in the end, the bottleneck models the global spatiotemporal dependencies. We did not test the results of permuting the natural spatial ordering of the datasets as it was not part of our research questions. However, we believe that such permutation would likely penalize the training of all models except for the simple LSTM and GRU models.

The bottleneck of our proposed model has 1×1 spatiotemporal dimensions. This extreme shrinking of spatiotemporal dimensions is crucial in our model. Firstly, it allows one to always obtain unmasked data in the bottleneck and predict by decoding the posterior that is masked at the source. Secondly, as explained above, it allows for the global modeling of the spatiotemporal dependencies of MTS.

4.3. Non-Homogeneous Cadences of IRIS Data

The proposed model achieves an extreme reduction of the spatiotemporal dimensions to 1×1 and C channels. We believe that this extreme spatiotemporal reduction helps to tackle the problem of the non-homogeneous cadences of the temporal dimension presented in Figure 4 because it is compressed into a unique spatiotemporal dimension in the bottleneck.

The exact theoretical explanation as to why the model performs well with non-homogeneous data still remains open and is the subject of investigation. Synthetic periodic data with non-homogeneous frequencies may help test the robustness of all models. In our paper, the convincing results on IRIS data show that our proposed model is robust to non-homogeneous data. We did not extend the study of this question to other datasets as we could not find any other MTS dataset with non-homogeneous cadences of observations. Usually, physical observations are designed with a fixed cadence to keep the analysis simple. As a consequence, we do not believe that the elaboration of synthetic non-homogeneous data is a priority for this work and we just showed the good performance of the proposed model on IRIS data.

In addition, the non-homogeneous cadences of observations in the IRIS data may contribute to the poor forecasting performance of NBeats for this dataset. NBeats is based on the decomposition of MTS data into interpretable MTS signals that compose the observations, similar to the trend and seasonality decomposition. It is very challenging and perhaps impossible to find the trend and seasonality of MTS observations with non-homogeneous cadences. In AL and PB data, the cadences were the same for all of the data,

and NBeats performs much better; however, the best results are given by our proposed model for these data.

4.4. Pros

Interestingly, while the model is trained on a CV loss that is not designed to measure accuracies in activity classifications or astrophysical information, the model is still able to predict the information with significant accuracy. This is in favor of interpretability and we believe that this is a great plus.

Unlike most recent studies in MTS forecasting, where spatiotemporal embeddings are designed prior to being fed to the models, our model jointly works straight on spatiotemporal dimensions seen together as images. Because of the fully masked convolutional structure, we believe that such simplicity is the advantage of our model, which can be easily applied to data, and generalized to more dimensions, such as for videos where the spatial information is composed of two joint dimensions.

4.5. Cons and Possible Extensions

Because of the activations used, the models were tested on normalized data, where each time step had a maximum of one. While this is suitable for many applications where only the spatial distribution of the data is important at each time step, such as the shape of a spectral line, in some applications, it is also important to know the real maximum value or *intensity* of the data at each time step. We believe that in this case, intensity forecasting is a simpler issue than the one solved in our work; one can train a classical RNN model to predict the intensities in parallel to the spatiotemporal MTS.

Our model has much more parameters to train than classical RNNs, but still less than the most recent transformers. Without significant difficulties, we could train our model on data with 240×240 dimensions. Moreover, the performant CV loss used to train the model comes at the expense of memory comes at the expense of to save the parameters of the VGG-16 used in the loss.

We believe that the model could be further improved by integrating more interpretability and prediction of the forecast errors. These features were not targeted in this step and are not in the scope of this work.

4.6. Future Research Directions

Recent understandings and implementations of the IB principle could also further improve the performance of our presented IB-MTS. For instance, [39] showed that the general IB principle is equivalent to several adversarial and CV losses present at the output and at the compressed bottleneck levels, where we only considered the particular case with the CV loss at the output. In particular, introducing a loss at the bottleneck should help the compression and enhance data representations at the bottleneck, leading to increased interpretability. This approach would certainly bring new improvements, following the ones achieved by this work, showing the importance of the IB formulation in the context of MTS forecasting. Moreover, for a better estimation of the forecasting error, one could attempt to amend the output of the network and predict the means and standard deviations of Gaussian distributions for each predicted pixel instead of a singular value. This would make the model stochastic and facilitate the confidence estimation in the forecasted outputs.

Author Contributions: Conceptualization, S.V. and D.U.; methodology, D.U.; software, D.U.; validation, S.V. and D.U.; formal analysis, D.U.; investigation, D.U.; resources, D.U.; data curation, D.U.; writing—original draft preparation, S.V., D.U. and O.T.; writing—review and editing, S.V., D.U. and O.T.; visualization, D.U.; supervision, S.V.; project administration, D.U.; funding acquisition, S.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Swiss National Science Foundation SNSF, NRP75 project no. 407540_167158, SNSF Sinergia project no. 193826, and the University of Geneva.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The code used in this work and selected classified IRIS data can be found at github.com/DenisUllmann/IB-MTS, accessed on 20 February 2023. [IRIS] Author: NASA; IRIS is a NASA small explorer mission developed and operated by LMSA. LMission operations are conducted at the NASA Ames Research Center, and significant contributions to downlink communications are funded by the ESA and the Norwegian Space Centre. For more information, please visit iris.lmsal.com/search/, accessed on 20 February 2023. [AL] Author: The National Renewable Energy Laboratory, U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, operated by the Alliance for Sustainable Energy LLC; Solar Power Data for the year 2006 in Alabama at www.nrel.gov/grid/solar-power-data.html, accessed on 20 February 2023. [PB] Author: California State Transportation Agency (CalSTA) Performance Measurement System (PeMS) and Yaguang Li; PeMS-BAY traffic data [74] at dx.doi.org/10.5281/zenodo.5146275, accessed on 20 February 2023.

Acknowledgments: We would like to acknowledge Lucia Kleint, Brandon Panos, and Cedric Huwyler for their assistance in providing access to IRIS satellite data, as well as for their valuable explanations and labeling of the data. Additionally, we utilized the Python library https://github.com/i4Ds/IRISreader, accessed on 20 February 2023 to work with the IRIS data and accomplish the objectives of this work.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AL	solar power dataset for the year 2006 in Alabama
AR	solar active region
ARIMA	autoregressive integrated moving average model
CV	computer vision
DNN	deep neural network
FL	solar flare
GNN	graph neural network
GRU	gated recurrent unit
HSS	Heidke skill score
IB	information bottleneck
IRIS	NASA's interface region imaging spectrograph satellite
IT	information theory
LSTM	long short-term memory model
MAE	mean absolute error
MAPE	mean absolute percentage error
ML	machine learning
MOS	mean opinion score
MSE	mean square error
MTS	multiple time series
NLP	natural language processing
NN	neural network
PB	PeMS-BAY dataset
PC	partial convolution
PSNR	peak signal-to-noise ratio
QS	quiet Sun
RAM	random access memory
RGB	red–green–blue
RMSE	root mean square error
RNN	recurrent neural network
SSIM	structural similarity
TS	time series
TSS	true skill statistic

Appendix A. Theory

Table A1. Summary table of the notations.

Random Variables	
$\mathbf{X}$	generic spatiotemporal data
$\tilde{\mathbf{X}}$	estimation of $\mathbf{X}$
T	scalar duration of the prior sequence
F	scalar duration of the posterior sequence
M	spatial size of spatiotemporal data
$\mathbf{X}_t$	multidimensional data at time step t
X_t^m	scalar value at time step t and spatial index m
$\mathbf{X}_{1:T} = \mathbf{X}_{1:T}^{1:M}$	prior sequence
$\tilde{\mathbf{X}}_{1:T}$	prior sequence estimation
$\mathbf{X}_{T+1:T+F} = \mathbf{X}_{T+1:T+F}^{1:M}$	posterior genuine sequence
$\tilde{\mathbf{X}}_{T+1:T+F}$	posterior genuine sequence estimation
$\mathbf{X}_{1:T+F} = \mathbf{X}_{1:T+F}^{1:M}$	full sequence
$\tilde{\mathbf{X}}_{1:T+F}$	full sequence estimation
$\mathbf{Z}$	bottleneck
$1:T \rightarrow T+1:T+F$	transition from prior to posterior
$\mathbf{Z}_{ib\_tr}$	IB bottleneck for transition $1:T \rightarrow T+1:T+F$ learning
$\mathbf{Z}_{ib\_ae}$	IB bottleneck of AE
$\mathbf{M}$	generic mask for spatiotemporal data
$\mathbf{M}_{1:T}$	prior mask
$\mathbf{M}_{T+1:T+F}$	posterior mask
$\mathbf{M}_{1:T+F}$	full mask
$\mathbf{M}_{ib\_tr}$	mask at the IB bottleneck for transition $1:T \rightarrow T+1:T+F$ learning
$\mathbf{1}$ & $\mathbf{1}_{len}$ & $\mathbf{1}_{row \times col}$	vectors and matrices of ones, eventually with specified length *len* or *row* and *col* sizes.
$\mathbf{0}$ & $\mathbf{0}_{len}$ & $\mathbf{0}_{row \times col}$	vectors and matrices of zeros, eventually with specified length *len* or *row* and *col* sizes.
K & $\mathbf{K}$	scalar & categorical labels
c_0, c_1 & c_2	prior, genuine, and predicted centroid assignments
Information Theory	
$p_{\mathcal{D}}$	data distribution
$p_{\boldsymbol{\Theta}}$ & $p_{\boldsymbol{\Phi}}$	(encoding/decoding) distribution with parameter ($\boldsymbol{\Theta}/\boldsymbol{\Phi}$)
$\mathbb{E}_{p_{\mathcal{D}}}[\cdot]$	mean by sampling from $p_{\mathcal{D}}$
$\mathbb{E}_{p_{\boldsymbol{\Theta}}}[\cdot]$	mean by sampling from $p_{\boldsymbol{\Theta}}(\mathbf{Z}\|\mathbf{X})$
$H(\cdot)$ & $H(\cdot,\cdot)$	generic entropy and cross-entropy
$H_{p_{\boldsymbol{\Theta}}}$	entropy parametrized by the encoder
$H_{p_{\boldsymbol{\Theta}},p_{\boldsymbol{\Phi}}}$	cross-entropy parametrized by the encoder and the decoder
$KL(\cdot\|\|\cdot)$ & $I(\cdot;\cdot)$	KL-divergence & mutual information
$I_{\boldsymbol{\Theta}}$ & $I_{\boldsymbol{\Phi}}$	Encoding and decoding mutual information
Layers & mappers	
Id	Identity mapper
$Concat$	Concatenation of tensors
$(-/B/P)Conv$	(-/Binary/Partial) Convolutional layer
$(-/B/P)DConv$	(-/Binary/Partial) Deconvolutional layer
Losses & metrics	
$\mathcal{L}$	generic loss
$\tilde{\mathcal{L}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$	upper bound on the loss $\mathcal{L}$
$\mathcal{L}_3^{Lap}$	$\mathcal{L}_3$ with Laplacian assumption of $p_{\boldsymbol{\Phi}}(\mathbf{X}_{T+1:T+F}\|\mathbf{Z}_{ib\_tr})$
$\mathcal{L}_3^{Lap,UNet}$	$\mathcal{L}_3^{Lap}$ for a U-Net architecture
$ref_{k,t}$	relative error for feature k at time step t

Proof of Equation (5). This proof is highly inspired by [39] but without variational approximation considerations. Let us consider the loss defined in Equation (4) and let us simplify the TS notations that are not relevant to this proof:

$$\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}) = I_{\mathbf{\Theta}}(\mathbf{X}_{1:T}; \mathbf{Z}_{ib\_tr}) - \beta I_{\mathbf{\Phi}}(\mathbf{Z}_{ib\_tr}; \mathbf{X}_{T+1:T+F}) = I_{\mathbf{\Theta}}(\mathbf{X}; \mathbf{Z}) - \beta I_{\mathbf{\Phi}}(\mathbf{Z}; \mathbf{Y}). \tag{A1}$$

The first term is mutual information that is parameterized by $\mathbf{\Theta}$, which represents the parameters of the encoding part. This term can be decomposed:

$$\begin{aligned} I_{\mathbf{\Theta}}(\mathbf{X}; \mathbf{Z}) &= \mathbb{E}_{p_{\mathbf{\Theta}}(\mathbf{x},\mathbf{z})}\left[\log \frac{p_{\mathbf{\Theta}}(\mathbf{z}|\mathbf{x}) p_{\mathcal{D}}(\mathbf{x})}{p_{\mathbf{\Theta}}(\mathbf{z}) p_{\mathcal{D}}(\mathbf{x})}\right] \\ &= \mathbb{E}_{p_{\mathbf{\Theta}}(\mathbf{z})}[\log p_{\mathbf{\Theta}}(\mathbf{z})] - \mathbb{E}_{p_{\mathbf{\Theta}}(\mathbf{x},\mathbf{z})}[\log p_{\mathbf{\Theta}}(\mathbf{z}|\mathbf{x})] \\ &= H_{p_{\mathbf{\Theta}}}(\mathbf{Z}) - H_{p_{\mathbf{\Theta}}}(\mathbf{Z}|\mathbf{X}). \end{aligned} \tag{A2}$$

The second term of (A1) is the mutual information parametrized by Φ, which represents the parameters of decoding, and can be decomposed:

$$\begin{aligned} I_{\mathbf{\Phi}}(\mathbf{Z}; \mathbf{Y}) &= \mathbb{E}_{p_{\mathbf{\Phi}}(\mathbf{z},\mathbf{y})}\left[\log \frac{p_{\mathcal{D}}(\mathbf{y}|\mathbf{z}) p_{\mathbf{\Phi}}(\mathbf{z})}{p_{\mathcal{D}}(\mathbf{y}) p_{\mathbf{\Phi}}(\mathbf{z})}\right] \\ &= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{y})}[\log p_{\mathcal{D}}(\mathbf{y})] - \mathbb{E}_{p_{\mathbf{\Phi}}(\mathbf{z},\mathbf{y})}[\log p_{\mathcal{D}}(\mathbf{y}|\mathbf{z})] \\ &= H_{p_{\mathcal{D}}}(\mathbf{Y}) - \mathbb{E}_{p_{\mathbf{\Phi}}(\mathbf{z},\mathbf{y})}\left[\log p_{\mathcal{D}}(\mathbf{y}|\mathbf{z}) \frac{p_{\mathbf{\Phi}}(\mathbf{y}|\mathbf{z})}{p_{\mathbf{\Phi}}(\mathbf{y}|\mathbf{z})}\right] \\ &= H_{p_{\mathcal{D}}}(\mathbf{Y}) - \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x},\mathbf{y})}\left[\mathbb{E}_{p_{\Theta}(\mathbf{z}|\mathbf{x})}[\log p_{\mathbf{\Phi}}(\mathbf{y}|\mathbf{z})]\right] + \mathbb{E}_{p_{\mathbf{\Phi}}(\mathbf{z},\mathbf{y})}\left[\log \frac{p_{\mathbf{\Phi}}(\mathbf{y}|\mathbf{z})}{p_{\mathcal{D}}(\mathbf{y}|\mathbf{z})}\right] \\ &= H_{p_{\mathcal{D}}}(\mathbf{Y}) - H_{p_{\Theta,\mathbf{\Phi}}}(\mathbf{Y}|\mathbf{Z}) + KL(p_{\mathbf{\Phi}}(\mathbf{y}|\mathbf{z})||p_{\mathcal{D}}(\mathbf{y}|\mathbf{z})) \\ &\geq H_{p_{\mathcal{D}}}(\mathbf{Y}) - H_{p_{\Theta,\mathbf{\Phi}}}(\mathbf{Y}|\mathbf{Z}), \end{aligned} \tag{A3}$$

because the Kullback–Leibler divergence $KL(p_{\mathbf{\Phi}}(\mathbf{y}|\mathbf{z})||p_{\mathcal{D}}(\mathbf{y}|\mathbf{z}))$ is positive.

As a consequence, by putting the results of Equations (A2) and (A3) in Equation (A1), one can obtain the following upper bound on the loss:

$$\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}) \leq H_{p_{\mathbf{\Theta}}}(\mathbf{Z}) - H_{p_{\mathbf{\Theta}}}(\mathbf{Z}|\mathbf{X}) + \beta H_{p_{\mathbf{\Theta},\mathbf{\Phi}}}(\mathbf{Y}|\mathbf{Z}) - \beta H_{p_{\mathcal{D}}}(\mathbf{Y}), \tag{A4}$$

and $H_{p_{\mathcal{D}}}(\mathbf{Y})$ being fixed by the data, the upper bound on the loss can be reduced to:

$$\tilde{\mathcal{L}}(\mathbf{\Theta}, \mathbf{\Phi}) = H_{p_{\mathbf{\Theta}}}(\mathbf{Z}) - H_{p_{\mathbf{\Theta}}}(\mathbf{Z}|\mathbf{X}) + \beta H_{p_{\mathbf{\Theta},\mathbf{\Phi}}}(\mathbf{Y}|\mathbf{Z}), \tag{A5}$$

□

Remark A1 (Details on *PConv* and *PDeconv* for MTS). *With T representing the time steps of prior values and F representing the time steps of forecasted values, let us recall Equation (4):*

$$\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}) = I_{\mathbf{\Theta}}(\mathbf{X}_{1:T}; \mathbf{Z}_{ib\_tr}) - \beta I_{\mathbf{\Phi}}(\mathbf{Z}_{ib\_tr}; \mathbf{X}_{T+1:T+F}). \tag{A6}$$

This corresponds to the general formulation of the IB principle with the MTS notation. The bottleneck variable $\mathbf{Z}_{ib\_tr}$ should then hold part of the information of prior time steps $1:T$ and forecast time steps $T+1:T+F$ in order to learn the time transition. Equation (A6) can be rewritten using temporal masks on the source $\mathbf{X}_{1:T+F} = Concat[\mathbf{X}_{1:T}, \mathbf{X}_{T+1:T+F}]$:

$$\begin{aligned} \mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}) = &\underbrace{I_{\mathbf{\Theta}}([\mathbf{X}_{1:T}, \mathbf{X}_{T+1:T+F}] \odot \mathbf{M}_{1:T}; \mathbf{Z}_{ib\_tr} \odot \mathbf{M}_{ib\_tr})}_{A} \\ &- \beta \underbrace{I_{\mathbf{\Phi}}(\mathbf{Z}_{ib\_tr} \odot \mathbf{M}_{ib\_tr}; [\mathbf{X}_{1:T}, \mathbf{X}_{T+1:T+F}] \odot \mathbf{M}_{T+1:T+F})}_{B}, \end{aligned} \tag{A7}$$

where $\odot$ *is the element-wise product, also known as the Hadamard dot product, with* $\mathbf{M}_{1:T}$ *and* $\mathbf{M}_{T+1:T+F}$*representing binary time masks having ones at the indexed time positions,* $1:T$ *or* $T+1:T+F$*, and zeros at the other time positions. Binary masks* $\mathbf{M}_{1:T}$ *and* $\mathbf{M}_{T+1:T+F}$ *lie in the same manifold, such as the MTS* $\mathbf{X}_{1:T+F} = Concat[\mathbf{X}_{1:T}, \mathbf{X}_{T+F}]$*, and* $\mathbf{M}_{ib\_tr}$ *in the same manifold, such as the compressed variable* $\mathbf{Z}_{ib\_tr}$*, such that* $\mathbf{M}_{1:T}, \mathbf{M}_{T+1:T+F} \in \mathbb{R}^{(T+F)\times M}$*,* $\mathbf{M}_{1:T} = Concat[\mathbf{1}_{T\times M}, \mathbf{0}_{F\times M}]$*,* $\mathbf{M}_{T+1:T+F} = Concat[\mathbf{0}_{T\times M}, \mathbf{1}_{F\times M}]$*;* $\mathbf{M}_{ib\_tr}$ *is a binary mask for the compressed manifold of* $\mathbf{Z}_{ib\_tr}$ *and is filled only by ones:* $\mathbf{Z}_{ib\_tr} \odot \mathbf{M}_{ib\_tr} = \mathbf{Z}_{ib\_tr}$*.*

A straightforward approach to designing the compression, described in part A of Equation (A7) for pseudo-images $\mathbf{X}_{1:T}$*, consists of a convolution Conv and binary convolution BConv, both with strides larger than 2 for spatiotemporal dimension reduction. Without a loss of generality for the explanation of Conv and BConv layers, let us consider the example with one prior time step* $T = 1$*,* $\mathbf{X}_{1:T} = \mathbf{X}_1$*, one posterior time step to forecast* $F = 1$*,* $\mathbf{X}_{T+1:T+F} = \mathbf{X}_2$*, and 2 spatial dimensions* $M = 2$*, such that* $\mathbf{X}_{1:2} \in \mathbb{R}^{2\times 2}$*; the bottleneck* $\mathbf{Z}_{ib\_tr} = \mathbf{Z}_{1\to 2}$ *learns the transition* $1 \to 2$*:*

$$\mathbf{Z}_{1\to 2} = Conv_{\mathbf{\Theta}}([\mathbf{X}_1, \mathbf{X}_2] \odot \mathbf{M}_1) \in \mathbb{R}^{1\times 1\times C} \quad and \quad \mathbf{M}_{1\to 2} = BConv(\mathbf{M}_1) = [1], \tag{A8}$$

where C *is the number of channels for* $Conv_{\mathbf{\Theta}}$*. Proof of Equation (A8) is given in Appendix A. The combination of* $Conv_{\mathbf{\Theta}}$*, BConv with corresponding masks* $\mathbf{M}_1$ *and* $\mathbf{M}_{1:2}$ *is usually referred to as* partial convolution *PConv*$_{\mathbf{\Theta}}$ *[4], such that:*

$$PConv_{\mathbf{\Theta}}(\mathbf{X}_{1:2}, \mathbf{M}_1) = (\mathbf{Z}_{1\to 2}, \mathbf{M}_{1\to 2}). \tag{A9}$$

In a symmetrical approach, still considering the example when $T = 1, F = 1, M = 2$*, the decoding described in part B of Equation (A7) can be designed by a deconvolution DConv and a binary deconvolution BDConv, both with strides of 2, but in this case, the decoder performs as well as a backcast of the prior* X_1 *because the deconvolution of* $\mathbf{M}_{1\to 2}$ *returns* $\mathbf{M}_{1:2}$ *instead of* $\mathbf{M}_2$ *of B:*

$$DConv_{\mathbf{\Phi}}(\mathbf{Z}_{1\to 2} \odot \mathbf{M}_{1\to 2}) = \left[\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2\right] \in \mathbb{R}^2 \quad and \quad BDConv(\mathbf{M}_{1\to 2}) = \mathbf{M}_{1:2} = \mathbf{1}_{2\times 2}, \tag{A10}$$

and Proof of Equation (A10) is given in Appendix A. The combination of $DConv_{\mathbf{\Phi}}$*, BDConv with corresponding masks* $\mathbf{M}_{1\to 2}$ *and* $\mathbf{M}_{1:2}$ *is usually referred to as* partial deconvolution *PDConv*$_{\mathbf{\Phi}}$*, such that:*

$$PDConv_{\mathbf{\Phi}}(\mathbf{Z}_{ib\_tr}, \mathbf{M}_{ib\_tr}) = (\tilde{\mathbf{X}}_{1:2}, \mathbf{M}_{1:2}). \tag{A11}$$

When $T + F$ *and* M *are larger than 2, pseudo-images* $\mathbf{X}_{1:T}$ *and masks* $\mathbf{M}_{1:T}$ *must be zero-padded to a square pseudo-image, whose size is a power of* 2*, and successive PConv and PDConv layers must be used to obtain a* $1 \times 1 \times C$ *bottleneck. For instance, when the padded input image is* 254×256*, 8 successive PConv and 8 successive PDConv layers must be used, and the model is deep. This forecast model design finally makes use of more traditional CV layers, such as convolutions, compared to classical TS deep models, such as LSTMs; however, the model is directly justified by the IB formulation of the time dimension compression.*

A drawback of this design is that the binary deconvolution decoder outputs $\mathbf{M}_{1:T+F} = \mathbf{1}_{(T+F)\times M}$ *instead of* $\mathbf{M}_{T+1:T+F} = Concat[\mathbf{0}_{T\times M}, \mathbf{1}_{F\times M}]$ *from part B of the IB Equation (A7). As a consequence, only with these PConv and PDConv layers would it have to perform a backcast and output an estimation of the prior* $\tilde{X}_{1:T}$ *in addition to the estimated forecast* $\tilde{X}_{T+1:T+F}$*, which means that the bottleneck does not only learn the transition statistics* $1:T \to T+1:T+F$ *but also the reconstruction of* $1:T$*. This is not the right formulation of the IB bottleneck for MTS forecasting. As a consequence, a skipping connection between the masked input* $\mathbf{X}_{1:T+F} \odot \mathbf{M}_{1:T}$ *and the output is then added to the model in order to let all of the information about the prior* $\mathbf{X}_{1:T}$ *pass without any parameter learning, bypassing the bottleneck* $\mathbf{Z}_{ib\_tr}$*. Complemented by these skipping layers, the bottleneck of the network only learns the statistics of the transition between the prior and the forecast. The resulting output for the designed model is then given by:*

$$\begin{aligned} UNet_{\mathbf{\Phi},\mathbf{\Theta}}(\mathbf{X}_{1:T+F}, \mathbf{M}_{1:T}) &= \left[\mathbf{X}^{IB}_{1:T}, \mathbf{X}^{IB}_{T+1:T+F}\right] \odot \mathbf{M}_{1:T+F} + \mathbf{X}_{1:T+F} \odot \mathbf{M}_{1:T} \\ &= \left[\mathbf{X}^{IB}_{1:T} + \mathbf{X}_{1:T}, \mathbf{X}^{IB}_{T+1:T+F}\right], \end{aligned} \tag{A12}$$

where $UNet_{\mathbf{\Phi},\mathbf{\Theta}}$ is the network with the PConv, PDConv, and skipping layers. $\mathbf{X}^{IB}_{1:T}$ is the estimation of the prior $\mathbf{X}_{1:T}$ performed through the bottleneck $\mathbf{Z}_{ib\_tr}$, whereas $\mathbf{X}_{1:T+F} \odot \mathbf{M}_{1:T} = \mathbf{X}_{1:T}$ is the prior input directly connected to the output by the skipping layer. The output of the network is still not $[\mathbf{X}_{1:T}, \mathbf{X}_{T+1:T+F}] \odot \mathbf{M}_{T+1:T+F} = [0, \mathbf{X}_{T+1:T+F}] = \mathbf{X}_{T+1:T+F}$ from part B of the IB loss in Equation (A7), but we will now show the equivalence. With the described design of the prior compression, bottleneck decoding, and prior connection to the output, the upper bound $\mathcal{L}_3$ loss defined in Equation (7) then becomes equivalent to the following loss on the model estimation output from Equation (A12):

$$\mathcal{L}_3^{Lap,UNet}(\mathbf{\Theta},\mathbf{\Phi}) \equiv \mathbb{E}_{p_{\mathcal{D}}(\mathbf{X}_{1:T+F})}\left[\mathbb{E}_{p_{\mathbf{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T})}\left[\|\mathbf{X}_{1:T+F} - UNet_{\mathbf{\Phi},\mathbf{\Theta}}(\mathbf{X}_{1:T+F}, \mathbf{M}_{1:T})\|_1\right]\right]. \tag{A13}$$

This equivalence and more details are given in Proof of Equation (A13) of this Appendix A.

Proof of Equation (A8).

$$\begin{aligned} \mathbf{Z}_{1\to 2} &= Conv_{\mathbf{\Theta}}([\mathbf{X}_1, \mathbf{X}_2] \odot \mathbf{M}_1) \\ &= Conv_{\mathbf{\Theta}}([\mathbf{X}_1, \mathbf{X}_2] \odot [\mathbf{1}_M, \mathbf{0}_M]) \\ &= Conv_{\mathbf{\Theta}}([\mathbf{X}_1, \mathbf{0}_M]) \\ &= \sigma(\mathbf{\Theta}_1 \times \mathbf{X}_1 + \mathbf{\Theta}_2 \times \mathbf{0}_M) \in \mathbb{R}, \end{aligned} \tag{A14}$$

where σ is a nonlinear activation and $Conv$ is a one-dimensional convolution with parameter $\mathbf{\Theta} = [\mathbf{\Theta}_1, \mathbf{\Theta}_2]$, and:

$$\mathbf{M}_{1\to 2} = BConv(\mathbf{M}_1) = BConv([1, 0]) = [1], \tag{A15}$$

where $BConv$ is a one-dimensional binary convolution defined by:

$$BConv(\mathbf{M}) = Binary(Conv_{[1,1]}(\mathbf{M})) \quad \text{and} \quad Binary(m) = \begin{cases} 1, & \text{if } m > 0 \\ 0, & \text{otherwise} \end{cases} \tag{A16}$$

□

Proof of Equation (A10).

$$\begin{aligned} DConv_{\mathbf{\Phi}}(\mathbf{Z}_{1\to 2} \odot \mathbf{M}_{1\to 2}) &= DConv_{\mathbf{\Phi}}(\mathbf{Z}_{1\to 2} \odot \mathbf{1}) = DConv_{\mathbf{\Phi}}(\mathbf{Z}_{1\to 2}) \\ &= \sigma([\mathbf{\Phi}_1 \times \mathbf{Z}_{1\to 2}, \mathbf{\Phi}_2 \times \mathbf{Z}_{1\to 2}]) = [\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2] \in \mathbb{R}^2, \end{aligned} \tag{A17}$$

where σ is a nonlinear activation and $DConv$ is a one-dimensional deconvolution of stride 2 with parameter $\mathbf{\Phi} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2]$, and:

$$BDConv(\mathbf{M}_{1\to 2}) = BDConv([\mathbf{1}]) = [\mathbf{1}, \mathbf{1}], \tag{A18}$$

where $BConv$ is a one-dimensional binary convolution of stride 2, defined by:

$$BDConv(\mathbf{M}) = Binary(DConv(\mathbf{M})). \tag{A19}$$

□

Proof of Equation (A13). $\mathcal{L}_3^{Lap}(\mathbf{\Theta},\mathbf{\Phi})$ is the third component of the upper bound on the IB loss from Equation (6), where $p_{\mathbf{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$ is assumed to be Laplacian, and $\mathcal{L}_3^{Lap,UNet}(\mathbf{\Theta},\mathbf{\Phi})$ is the theoretical loss for our proposed model designed with U-Net and a Laplacian assumption of $p_{\mathbf{\Phi}}(\mathbf{X}_{T+1:T+F}|\mathbf{Z}_{ib\_tr})$:

$$\begin{aligned} \mathcal{L}_3^{Lap}(\mathbf{\Theta},\mathbf{\Phi}) &= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{X}_{1:T+F})}\left[\mathbb{E}_{p_{\mathbf{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T})}\left[\|\mathbf{X}_{T+1:T+F} - g_{\mathbf{\Phi}}(\mathbf{Z}_{ib\_tr})\|_1\right]\right] \\ &= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{X}_{1:T+F})}\left[\mathbb{E}_{p_{\mathbf{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T})}\left[\left\|\mathbf{X}_{T+1:T+F} - \mathbf{X}^{IB}_{T+1:T+F}\right\|_1\right]\right], \end{aligned} \tag{A20}$$

$$\begin{aligned}\mathcal{L}_3^{Lap,UNet}(\boldsymbol{\Theta},\boldsymbol{\Phi}) &= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{X}_{1:T+F})}\left[\mathbb{E}_{p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T})}\left[\|\mathbf{X}_{1:T+F} - UNet_{\boldsymbol{\Theta},\boldsymbol{\Phi}}(\mathbf{X}_{1:T+F},\mathbf{M}_{1:T})\|_1\right]\right] \\ &= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{X}_{1:T+F})}\left[\mathbb{E}_{p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T})}\left[\left\|\mathbf{X}_{1:T+F} - \left[\mathbf{X}_{1:T} + \mathbf{X}_{1:T}^{IB}, \mathbf{X}_{T+1:T+F}^{IB}\right]\right\|_1\right]\right],\end{aligned} \tag{A21}$$

where $\mathbf{X}_{1:T}$ and $\mathbf{X}_{T+1:T+F}$ are the prior and genuine forecasts given in the training dataset, and$\mathbf{X}_{1:T}^{IB}$ and $\mathbf{X}_{T+1:T+F}^{IB}$ are the predicted prior and forecast after the compression and decompression using the bottleneck $\mathbf{Z}_{ib\_tr}$. Starting with the norm in Equation (A21), we can prove:

$$\begin{aligned}\left\|\mathbf{X}_{1:T+F} - \left[\mathbf{X}_{1:T} + \mathbf{X}_{1:T}^{IB}, \mathbf{X}_{T+1:T+F}^{IB}\right]\right\|_1 &= \left\|\left[\mathbf{X}_{1:T}^{IB}, \mathbf{X}_{T+1:T+F} - \mathbf{X}_{T+1:T+F}^{IB}\right]\right\|_1 \\ &= \left\|\mathbf{X}_{1:T}^{IB}\right\|_1 + \left\|\mathbf{X}_{T+1:T+F} - \mathbf{X}_{T+1:T+F}^{IB}\right\|_1,\end{aligned} \tag{A22}$$

and by the linearity of the expected value function:

$$\mathcal{L}_3^{Lap,UNet}(\boldsymbol{\Theta},\boldsymbol{\Phi}) + \mathbb{E}_{p_{\mathcal{D}}(\mathbf{X}_{1:T+F})}\left[\mathbb{E}_{p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T})}\left[\left\|\mathbf{X}_{1:T}^{IB}\right\|_1\right]\right] = \mathcal{L}_3^{Lap}(\boldsymbol{\Theta},\boldsymbol{\Phi}), \tag{A23}$$

such that $\mathcal{L}_3^{Lap}(\boldsymbol{\Theta},\boldsymbol{\Phi}) \leq \mathcal{L}_3^{Lap,UNet}(\boldsymbol{\Theta},\boldsymbol{\Phi})$, because the norm is positive. Moreover, the minimum of $\mathbb{E}_{p_{\mathcal{D}}(\mathbf{X}_{1:T+F})}\left[\mathbb{E}_{p_{\boldsymbol{\Theta}}(\mathbf{Z}_{ib\_tr}|\mathbf{X}_{1:T})}\left[\|\mathbf{X}_{1:T}^{IB}\|_1\right]\right]$ is 0, obtained when $\mathbf{X}_{1:T}^{IB} = 0$ for all $\mathbf{X}_{1:T} \sim p_{\mathcal{D}}(\mathbf{X}_{1:T})$. The two losses, $\mathcal{L}_3^{Lap,UNet}(\boldsymbol{\Theta},\boldsymbol{\Phi})$ and $\mathcal{L}_3^{Lap}(\boldsymbol{\Theta},\boldsymbol{\Phi})$, share the same minimum. In fact, minimizing $\mathcal{L}_3^{Lap,UNet}(\boldsymbol{\Theta},\boldsymbol{\Phi})$ is equivalent to minimizing $\mathcal{L}_3^{Lap}(\boldsymbol{\Theta},\boldsymbol{\Phi})$ while forcing the model to not learn a backcast $\mathbf{X}_{1:T}^{IB}$; instead, the bottleneck $\mathbf{Z}_{ib\_tr}$ only learns the sufficient statistics of transitioning from the prior $\mathbf{X}_{1:T}$ to the forecast $\mathbf{X}_{T+1:T+F}$. □

Appendix B. Models

Table A2. Model summary of IB-MTS for IRIS data. The encoding part has 7 successive repetitions indexed by $i \in [0:7]$ of *PConv* and *BatchNormalization* layers followed by *ReLu* activations, except when $i = 0$, no *BatchNormalization* layer is included. The decoding part has 7 successive repetitions indexed by $i \in [0:7]$ of *UpSampling*, *Concatenation*, *PConv*, and *BatchNormalization* layers followed by *ReLu* activations, except when $i = 7$, no *BatchNormalization* layer is included. EncPConv2D$_0$ is connected to [zero_pad2d$_1$, zero_pad2d$_2$]. DecUpImg$_0$ is connected to [EncReLu$_7$]. DecUpMsk$_0$ is connected to [EncPConv2D$_7$[1]]. DecConcatImg$_7$ is connected to [zero_pad2d$_1$[1], DecUpImg$_7$]. DecConcatMsk$_7$ is connected to [zero_pad2d$_2$[1], DecUpMsk$_7$].

Model: IB-MTS for IRIS Data				
Layer (Type)	**Output Shape**	**Kernel**	**Param #**	**Connected to**
inputs_img (InputLayer)	[(240, 240, 1)]	–	0	[]
inputs_mask (InputLayer)	[(240, 240, 1)]	–	0	[]
zero_pad2d$_1$ (ZeroPad2D)	(256, 256, 1)	–	0	[inputs_img]
zero_pad2d$_2$ (ZeroPad2D)	(256, 256, 1)	–	0	[inputs_mask]

Table A2. *Cont.*

Model: IB-MTS for IRIS Data				
Layer (Type)	**Output Shape**	**Kernel**	**Param #**	**Connected to**
EncPConv2D_i (PConv2D)	$i = 0$ (128,128,64) $i = 1$ (64, 64, 128) $i = 2$ (32, 32, 256) $i = 3$ (16, 16, 512) $i = 4$ (8, 8, 512) $i = 5$ (4, 4, 512) $i = 6$ (2, 2, 512) $i = 7$ (1, 1, 512)	7 5 5 3 3 3 3 3	18880 410240 1639168 2361856 4721152 4721152 4721152 4721152	[EncReLu_{i-1}, EncPConv2D_{i-1}[1]]
$\text{EncBN}_{i\neq 0}$ (BatchNorm)				[EncPConv2D_i[0]]
EncReLu_i (Activation)				[EncBN_i]
DecUpImg_i (UpSampling2D)	$i = 0$ (2, 2, 512) $i = 1$ (4, 4, 512) $i = 2$ (16, 16, 512) $i = 3$ (32, 32, 512) $i = 4$ (64, 64, 256) $i = 5$ (128, 128, 128) $i = 6$ (256, 256, 64) $i = 7$ (256, 256, 3)	3 3 3 3 3 3 3 3	9439744 9439744 9439744 9439744 3540224 885376 221504 3621	[DecLReLU_{i-1}]
DecUpMsk_i (UpSampling2D)				[DecPConv2D_{i-1}[1]]
DecConcatImg_i (Concatenate)				[EncReLu_{6-i}, DecUpImg_i]
DecConcatMsk_i (Concatenate)				[EncPConv2D_{6-i}[1], DecUpMsk_i]
DecPConv2D_i (PConv2D)				[DecConcatImg_i, DecConcatMsk_i]
$\text{DecBN}_{i\neq 7}$ (BatchNorm)				[DecPConv2D_i[0]]
DecLReLU_i (LeakyReLU)				[DecBN_i]
outputs_img (Conv2D)	(256, 256, 1)	1	4	[DecLReLU7]
OutCrop (Cropping2D)	(240, 240, 1)	–	0	[outputs_img]
Total params: 65,724,969				

Table A3. Model summary of LSTM for the IRIS data. The ED-GRU model replaces the LSTM layer with a GRU layer consisting of 240 units and has a total of 347,040 parameters.

Model: LSTM for IRIS Data				
Layer (Type)	**Output Shape**	**Units**	**Param #**	**Connected to**
inputs_seq (InputLayer)	[(240, 240)]	–	0	[]
slice_0 (SlicingOp)	(180, 240)	–	0	[inputs_seq]
lstm (LSTM)	(180, 240)	240	461760	[slice_0]
slice_1 (SlicingOp)	(60, 240)	–	0	[lstm]
concat (TFOp)	(240, 240)	–	0	[slice_0, slice_1]
Total params: 461,760				

Table A4. Model summary for ED-LSTM for the IRIS data. The ED-GRU model replaces the LSTM layers with GRU layers, each consisting of 100 units. The total number of parameters in the model is 164,100.

Model: ED-LSTM for IRIS Data				
Layer (Type)	**Output Shape**	**Units**	**Param #**	**Connected to**
inputs_seq (InputLayer)	[(240, 240)]	–	0	[]
slice (SlicingOp)	(180, 240)	–	0	[inputs_seq]
$lstm_0$ (LSTM)	[(180, 100), (100)]	100	136400	[slice]
$lstm_1$ (LSTM)	[(100), (100)]	100	80400	[$lstm_0$[0]]
repeat_vector (RepeatVector)	(60, 100)	–	0	[$lstm_1$[0]]
$lstm_2$ (LSTM)	(60, 100)	100	80400	[repeat_vector, $lstm_0$[0][2]]
$lstm_3$ (LSTM)	(60, 100)	100	80400	[$lstm_2$[0], $lstm_1$[1]]
time_distributed (TimeDistributed)	(60, 240)	240	24240	[$lstm_3$]
concat (TFOp)	(240, 240)	–	0	[slice, time_distributed]
Total params: 401,840				

Appendix C. Results

IRIS QS Data

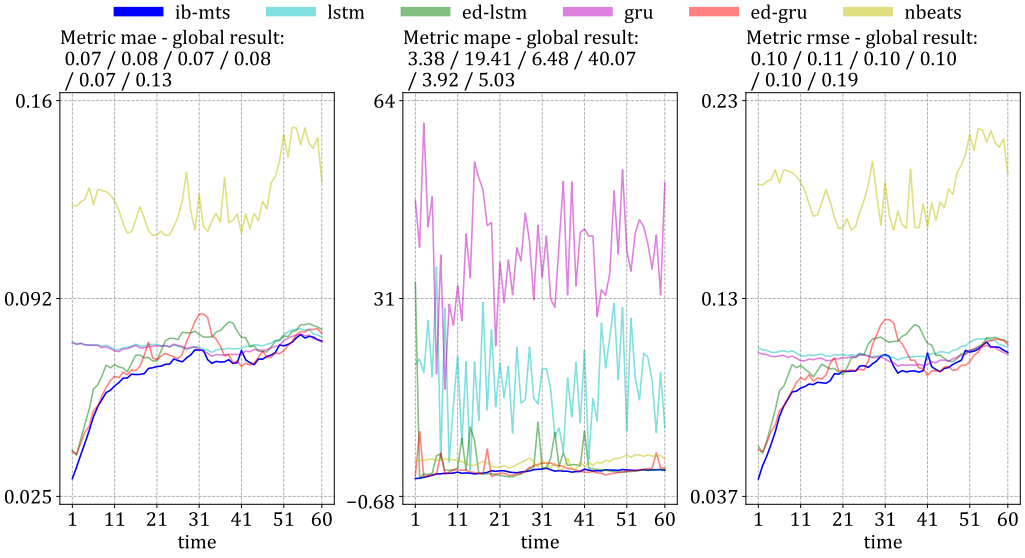


Figure A1. *Cont.*

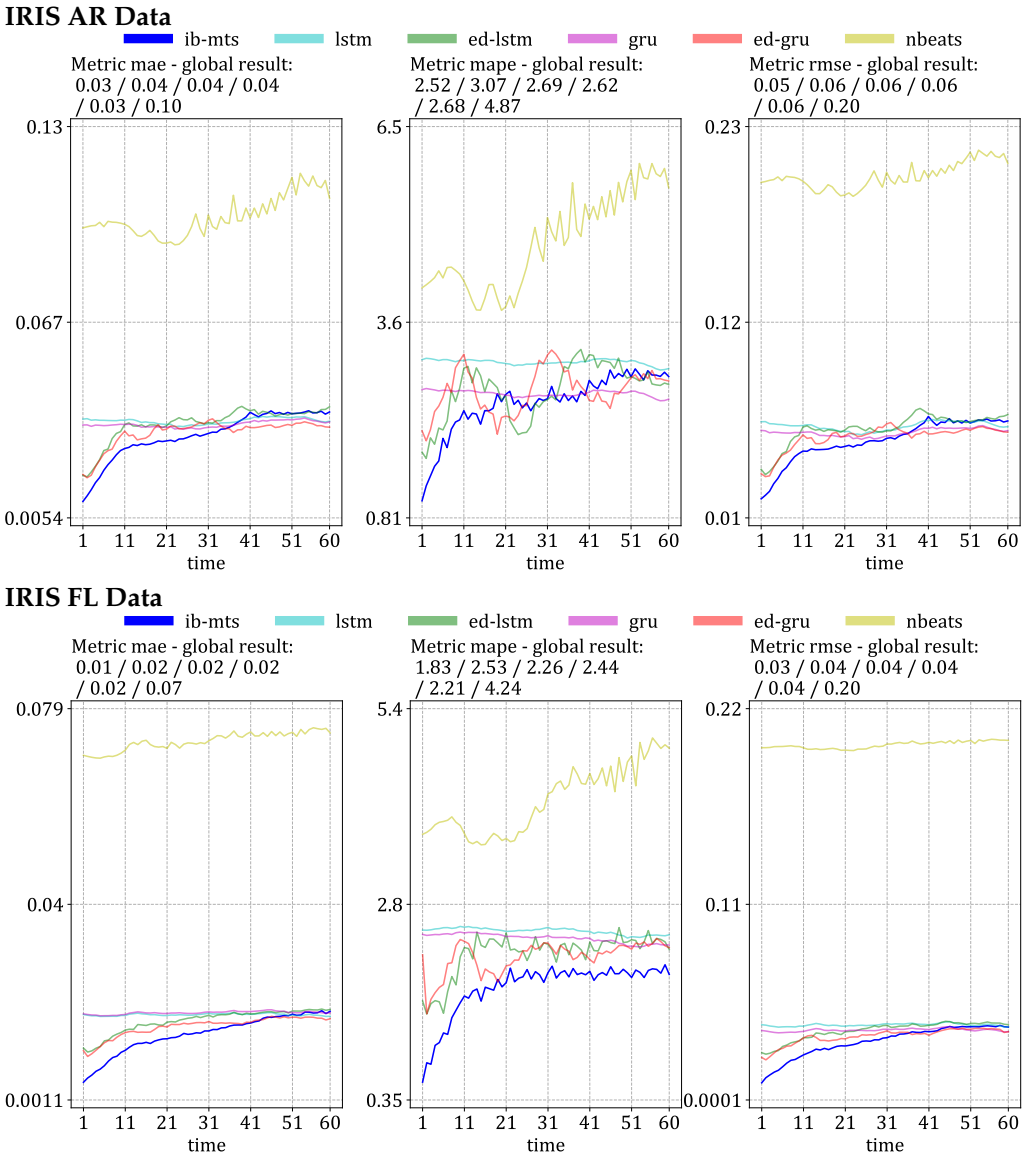


Figure A1. Detailed MTS metrics evaluation on the test set for the direct prediction setup. The evaluations are given for each solar activity: the first row of results is for QS activity, the second row is for AR, and the last row is for FL.

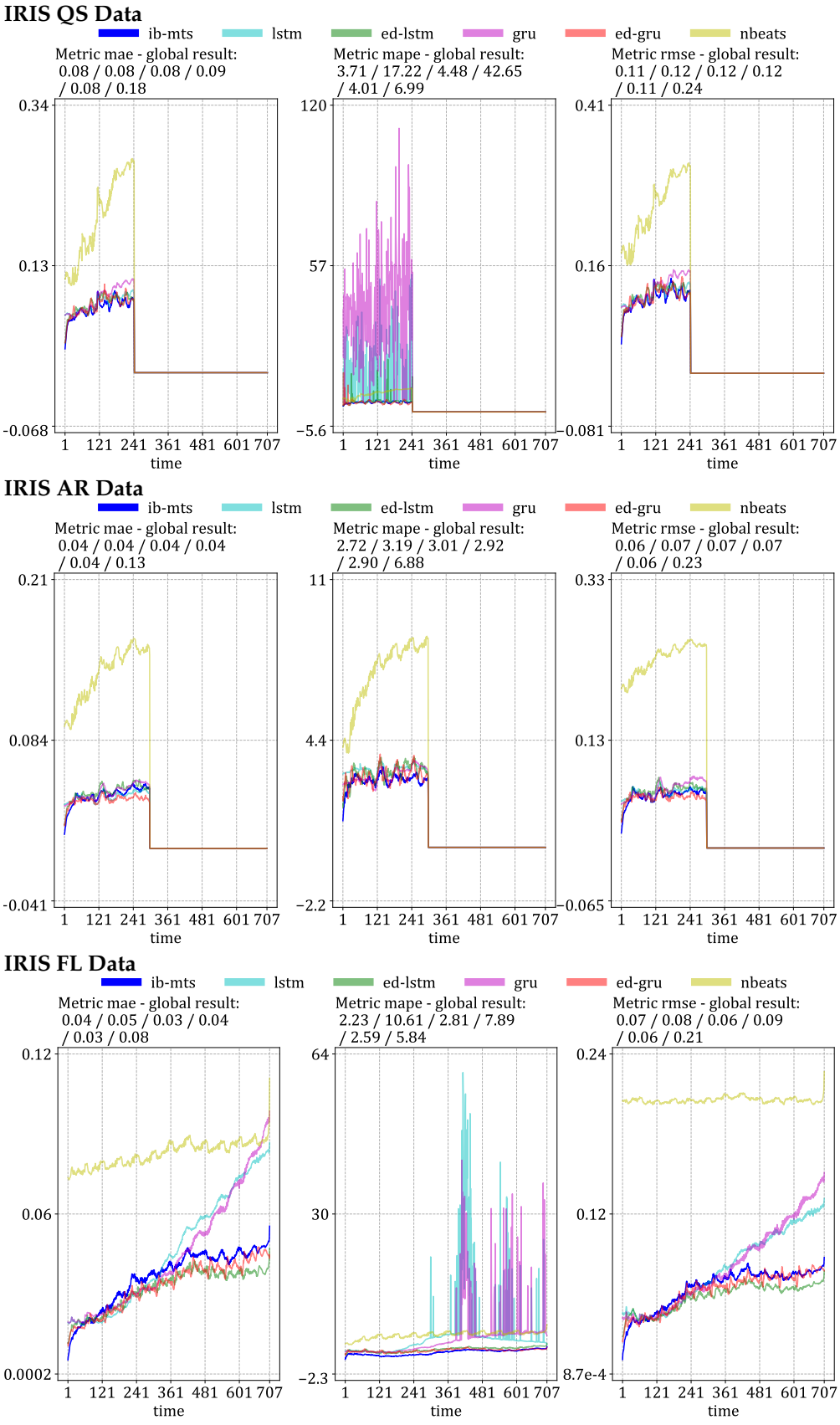


Figure A2. Detailed MTS metrics evaluation on the test set for the iterated prediction setup. The evaluations are given for each solar activity: the first row of results is for QS activity, the second row is for AR, and the last row is for FL.

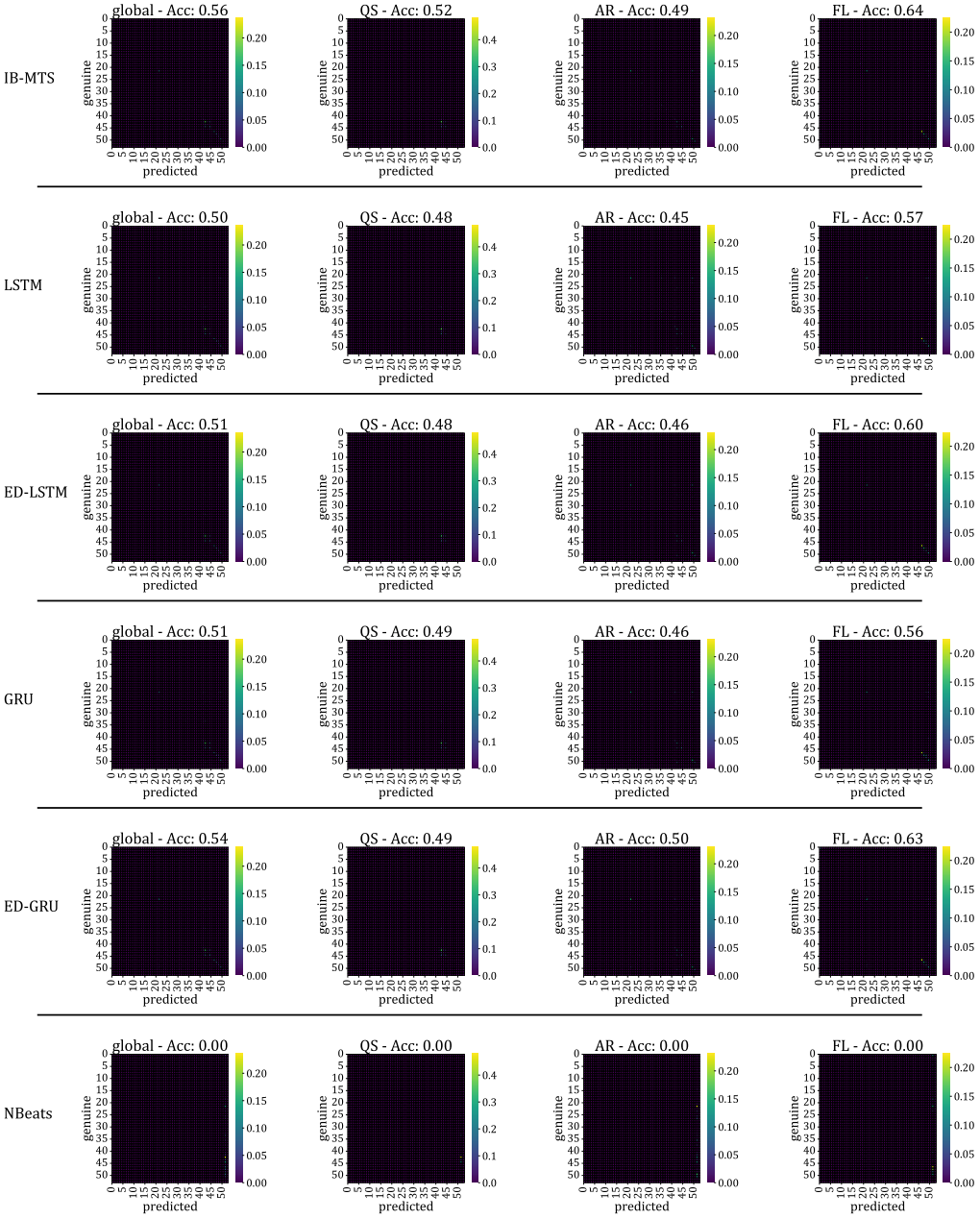


Figure A3. Confusion matrices for the prediction of centroids on IRIS data, for the direct procedure. We used the 53 centroids from [55]. Each row of results corresponds to a model. Columns are organized by data labels: *global* aggregate results for QS, AR, and FL data; other columns present the result for of each label, taken separately. Each confusion matrix gives results in terms of join probability distribution values between the genuine and the predicted. Probability values are displayed with color maps, where violet is the lowest probability and yellow is the highest.

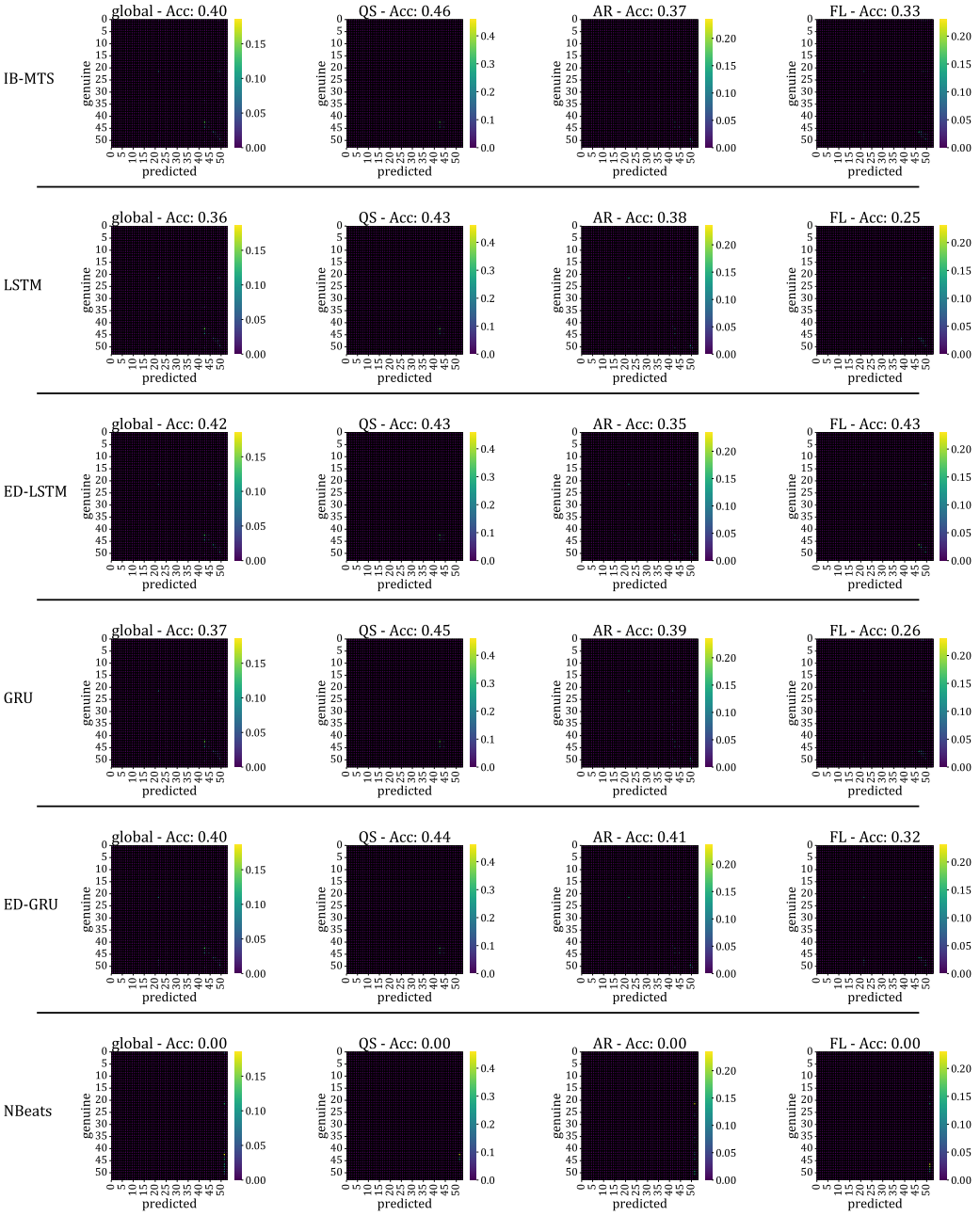


Figure A4. Confusion matrices for the prediction of centroids on IRIS data, for the iterated procedure. We used the 53 centroids from [55]. Each row of results corresponds to a model. Columns are organized by data labels: *global* aggregate results for QS, AR, and FL data; other columns present the results for of each label, taken separately. Each confusion matrix provides results in terms of join probability distribution values, between the genuine and the predicted. Probability values are displayed with color maps, where violet is the lowest probability and yellow is the highest.

References

1. Gangopadhyay, T.; Tan, S.Y.; Jiang, Z.; Meng, R.; Sarkar, S. Spatiotemporal Attention for Multivariate Time Series Prediction and Interpretation. *arXiv* **2020**, arXiv:2008.04882.
2. Flunkert, V.; Salinas, D.; Gasthaus, J. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. *arXiv* **2017**, arXiv:1704.04110.
3. Oreshkin, B.N.; Carpov, D.; Chapados, N.; Bengio, Y. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv* **2019**, arXiv:1905.10437.
4. Liu, G.; Reda, F.A.; Shih, K.J.; Wang, T.; Tao, A.; Catanzaro, B. Image Inpainting for Irregular Holes Using Partial Convolutions. *arXiv* **2018**, arXiv:1804.07723.

5. Rumelhart, D.E.; McClelland, J.L. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*; The MIT Press: Cambridge, MA, USA, 1987; Volume 1, pp. 318–362.
6. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
7. Dobson, A. *The Oxford Dictionary of Statistical Terms*; Oxford University Press: Oxford, UK, 2003; p. 506.
8. Kendall, M. *Time Series*; Charles Griffin and Co Ltd.: London, UK; High Wycombe, UK, 1976.
9. West, M. Time Series Decomposition. *Biometrika* **1997**, *84*, 489–494. [CrossRef]
10. Sheather, S. *A Modern Approach to Regression with R*; Springer: New York, NY, USA, 2009. [CrossRef]
11. Molugaram, K.; Rao, G.S. Chapter 12—Analysis of Time Series. In *Statistical Techniques for Transportation Engineering*; Molugaram, K., Rao, G.S., Eds.; Butterworth-Heinemann: Oxford, UK, 2017; pp. 463–489. [CrossRef]
12. Gardner, E.S. Exponential smoothing: The state of the art. *J. Forecast.* **1985**, 4, 1–28. [CrossRef]
13. Box, G.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*; Holden-Day: Cleveland, Australia, 1976.
14. Curry, H.B. The method of steepest descent for nonlinear minimization problems. *Quart. Appl. Math.* **1944**, *2*, 258–261. [CrossRef]
15. Cho, K.; van Merrienboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv* **2014**, arXiv:1409.1259.
16. Kazemi, S.M.; Goel, R.; Eghbali, S.; Ramanan, J.; Sahota, J.; Thakur, S.; Wu, S.; Smyth, C.; Poupart, P.; Brubaker, M. Time2Vec: Learning a Vector Representation of Time. *arXiv* **2019**, arXiv:1907.05321.
17. Lim, B.; Arik, S.O.; Loeff, N.; Pfister, T. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. *arXiv* **2019**, arXiv:1912.09363.
18. Grigsby, J.; Wang, Z.; Qi, Y. Long-Range Transformers for Dynamic Spatiotemporal Forecasting. *arXiv* **2021**, arXiv:2109.12218.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2021**, arXiv:1706.03762.
20. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *arXiv* **2020**, arXiv:2012.07436.
21. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [CrossRef] [PubMed]
22. Bertalmio, M.; Sapiro, G.; Caselles, V.; Ballester, C. Image Inpainting. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 23–28 July 2000; ACM Press/Addison-Wesley Publishing Co.: Boston, MA, USA, 2000; SIGGRAPH '00; pp. 417–424. [CrossRef]
23. Teterwak, P.; Sarna, A.; Krishnan, D.; Maschinot, A.; Belanger, D.; Liu, C.; Freeman, W.T. Boundless: Generative Adversarial Networks for Image Extension. *arXiv* **2019**, arXiv:1908.07007 2019.
24. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Deep Image Prior. *Int. J. Comput. Vis.* **2020**, *128*, 1867–1888. [CrossRef]
25. Dama, F.; Sinoquet, C. Time Series Analysis and Modeling to Forecast: A Survey. *arXiv* **2021**, arXiv:2104.00164.
26. Tessoni, V.; Amoretti, M. Advanced statistical and machine learning methods for multi-step multivariate time series forecasting in predictive maintenance. *Procedia Comput. Sci.* **2022**, *200*, 748–757. [CrossRef]
27. Lehtinen, J.; Munkberg, J.; Hasselgren, J.; Laine, S.; Karras, T.; Aittala, M.; Aila, T. Noise2Noise: Learning Image Restoration without Clean Data. *arXiv* **2018**, arXiv:1803.04189.
28. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image Style Transfer Using Convolutional Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2414–2423.
29. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *arXiv* **2016**, arXiv:1603.08155.
30. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv* **2016**, arXiv:1609.04802.
31. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597.
32. Kramer, M.A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.* **1991**, *37*, 233–243. [CrossRef]
33. Tishby, N.; Zaslavsky, N. Deep Learning and the Information Bottleneck Principle. *arXiv* **2015**, arXiv:1503.02406.
34. Costa, J.; Costa, A.; Kenda, K.; Costa, J.P. Entropy for Time Series Forecasting. In Proceedings of the Slovenian KDD Conference, Ljubljana, Slovenia, 4 October 2021. Available online: https://ailab.ijs.si/dunja/SiKDD2021/Papers/Costaetal_2.pdf (accessed on 20 February 2023).
35. Zapart, C.A. Forecasting with Entropy. In Proceedings of the Econophysics Colloquium, Taipei, Taiwan, 4–6 November 2010. Available online: https://www.phys.sinica.edu.tw/~socioecono/econophysics2010/pdfs/ZapartPaper.pdf (accessed on 20 February 2023).
36. Xu, D.; Fekri, F. Time Series Prediction Via Recurrent Neural Networks with the Information Bottleneck Principle. In Proceedings of the 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Kalamata, Greece, 25–28 June 2018; pp. 1–5. [CrossRef]
37. Ponce-Flores, M.; Frausto-Solís, J.; Santamaría-Bonfil, G.; Pérez-Ortega, J.; González-Barbosa, J.J. Time Series Complexities and Their Relationship to Forecasting Performance. *Entropy* **2020**, *22*, 89. [CrossRef] [PubMed]

38. Zaidi, A.; Estella-Aguerri, I.; Shamai (Shitz), S. On the Information Bottleneck Problems: Models, Connections, Applications and Information Theoretic Views. *Entropy* **2020**, *22*, 151. [CrossRef] [PubMed]
39. Voloshynovskiy, S.; Kondah, M.; Rezaeifar, S.; Taran, O.; Holotyak, T.; Rezende, D.J. Information bottleneck through variational glasses. *arXiv* **2019**, arXiv:1912.00830.
40. Alemi, A.A.; Fischer, I.; Dillon, J.V.; Murphy, K. Deep Variational Information Bottleneck. *arXiv* **2016**, arXiv:1612.00410.
41. Ullmann, D.; Rezaeifar, S.; Taran, O.; Holotyak, T.; Panos, B.; Voloshynovskiy, S. Information Bottleneck Classification in Extremely Distributed Systems. *Entropy* **2020**, *22*, 237. [CrossRef]
42. Geiger, B.C.; Kubin, G. Information Bottleneck: Theory and Applications in Deep Learning. *Entropy* **2020**, *22*, 1408. [CrossRef] [PubMed]
43. Lee, S.; Jo, J. Information Flows of Diverse Autoencoders. *Entropy* **2021**, *23*, 862. [CrossRef]
44. Tapia, N.I.; Estévez, P.A. On the Information Plane of Autoencoders. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8. [CrossRef]
45. Zarcone, R.; Paiton, D.; Anderson, A.; Engel, J.; Wong, H.P.; Olshausen, B. Joint Source-Channel Coding with Neural Networks for Analog Data Compression and Storage. In Proceedings of the 2018 Data Compression Conference, Snowbird, UT, USA, 27–30 March 2018; pp. 147–156. [CrossRef]
46. Boquet, G.; Macias, E.; Morell, A.; Serrano, J.; Vicario, J.L. Theoretical Tuning of the Autoencoder Bottleneck Layer Dimension: A Mutual Information-based Algorithm. In Proceedings of the 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, The Netherlands, 18–21 January 2021; pp. 1512–1516. [CrossRef]
47. Voloshynovskiy, S.; Taran, O.; Kondah, M.; Holotyak, T.; Rezende, D. Variational Information Bottleneck for Semi-Supervised Classification. *Entropy* **2020**, *22*, 943. [CrossRef]
48. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
49. Barnes, G.; Leka, K.D.; Schrijver, C.J.; Colak, T.; Qahwaji, R.; Ashamari, O.W.; Yuan, Y.; Zhang, J.; McAteer, R.T.J.; Bloomfield, D.S.; et al. A comparison of flare forecasting methods. *Astrophys. J.* **2016**, *829*, 89. [CrossRef]
50. Guennou, C.; Pariat, E.; Leake, J.E.; Vilmer, N. Testing predictors of eruptivity using parametric flux emergence simulations. *J. Space Weather Space Clim.* **2017**, *7*, A17. [CrossRef]
51. Benvenuto, F.; Piana, M.; Campi, C.; Massone, A.M. A Hybrid Supervised/Unsupervised Machine Learning Approach to Solar Flare Prediction. *Astrophys. J.* **2018**, *853*, 90. [CrossRef]
52. Florios, K.; Kontogiannis, I.; Park, S.H.; Guerra, J.A.; Benvenuto, F.; Bloomfield, D.S.; Georgoulis, M.K. Forecasting Solar Flares Using Magnetogram-based Predictors and Machine Learning. *Sol. Phys.* **2018**, *293*, 28. [CrossRef]
53. Kontogiannis, I.; Georgoulis, M.K.; Park, S.H.; Guerra, J.A. Testing and Improving a Set of Morphological Predictors of Flaring Activity. *Sol. Phys.* **2018**, *293*, 96. [CrossRef]
54. Ullmann, D.; Voloshynovskiy, S.; Kleint, L.; Krucker, S.; Melchior, M.; Huwyler, C.; Panos, B. DCT-Tensor-Net for Solar Flares Detection on IRIS Data. In Proceedings of the 2018 7th European Workshop on Visual Information Processing (EUVIP), Tampere, Finland, 26–28 November 2018; pp. 1–6. [CrossRef]
55. Panos, B.; Kleint, L.; Huwyler, C.; Krucker, S.; Melchior, M.; Ullmann, D.; Voloshynovskiy, S. Identifying Typical Mg ii Flare Spectra Using Machine Learning. *Astrophys. J.* **2018**, *861*, 62. [CrossRef]
56. Murray, S.A.; Bingham, S.; Sharpe, M.; Jackson, D.R. Flare forecasting at the Met Office Space Weather Operations Centre. *Space Weather* **2017**, *15*, 577–588.
57. Sharpe, M.A.; Murray, S.A. Verification of Space Weather Forecasts Issued by the Met Office Space Weather Operations Centre. *Space Weather* **2017**, *15*, 1383–1395.
58. Chen, Y.; Manchester, W.B.; Hero, A.O.; Toth, G.; DuFumier, B.; Zhou, T.; Wang, X.; Zhu, H.; Sun, Z.; Gombosi, T.I. Identifying Solar Flare Precursors Using Time Series of SDO/HMI Images and SHARP Parameters. *arXiv* **2019**, arXiv:1904.00125.
59. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Graph Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. *arXiv* **2017**, arXiv:1707.01926.
60. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting. *arXiv* **2017**, arXiv:1709.04875.
61. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Free-Form Image Inpainting with Gated Convolution. *arXiv* **2018**, arXiv:1806.03589.
62. Gatys, L.A.; Ecker, A.S.; Bethge, M. A Neural Algorithm of Artistic Style. *arXiv* **2015**, arXiv:1508.06576.
63. Wang, C.; Xu, C.; Wang, C.; Tao, D. Perceptual Adversarial Networks for Image-to-Image Transformation. *IEEE Trans. Image Process.* **2018**, *27*, 4066–4079. [CrossRef] [PubMed]
64. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
65. Kobyzev, I.; Prince, S.J.; Brubaker, M.A. Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3964–3979. [CrossRef] [PubMed]
66. Bao, H.; Dong, L.; Piao, S.; Wei, F. BEiT: BERT Pre-Training of Image Transformers. *arXiv* **2022**, arXiv:2106.08254.
67. Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; Hu, H. SimMIM: A Simple Framework for Masked Image Modeling. *arXiv* **2022**, arXiv:111.09886.
68. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked Autoencoders Are Scalable Vision Learners. *arXiv* **2021**, arXiv:2111.06377.

69. Pontieu, B.D.; Lemen, J. *IRIS Technical Note 1: IRIS Operations*; Version 17; LMSAL, NASA: Washington, DC, USA, 2013.
70. LMSAL. *A User's Guide to IRIS Data Retrieval, Reduction & Analysis*; Release 1.0; LMSAL, NASA: Washington, DC, USA, 2019.
71. Gošic, M.; Dalda, A.S.; Chintzoglou, G. *Optically Thick Diagnostics*; Release 1.0 ed.; LMSAL, NASA: Washington, DC, USA, 2018.
72. Panos, B.; Kleint, L. Real-time Flare Prediction Based on Distinctions between Flaring and Non-flaring Active Region Spectra. *Astrophys. J.* **2020**, *891*, 17. [CrossRef]
73. Gherrity, M. A learning algorithm for analog, fully recurrent neural networks. In Proceedings of the International 1989 Joint Conference on Neural Networks, Washington, DC, USA, 16–18 October 1989; Volume 1, pp. 643–644.
74. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In Proceedings of the International Conference on Learning Representations (ICLR '18), Vancouver, BC, Canada, 30 April–3 May 2018.
75. California, S.o. Performance Measurement System (PeMS) Data Source. Available online: https://pems.dot.ca.gov/ (accessed on 20 February 2023).
76. Hanssen, A.; Kuipers, W. On the relationship between the frequency of rain and various meteorological parameters. *Meded. En Verh.* **1965**, *81*, 3–15. Available online: https://cdn.knmi.nl/knmi/pdf/bibliotheek/knmipubmetnummer/knmipub102-81.pdf (accessed on 20 February 2023).
77. Heidke, P. Berechnung des Erfolges und der Gute der Windstarkevorhersagen im Sturmwarnungsdienst (Measures of success and goodness of wind force forecasts by the gale-warning service). *Geogr. Ann.* **1926**, *8*, 301–349.
78. Cohen, J. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **1960**, *20*, 213–220. [CrossRef]
79. Allouche, O.; Tsoar, A.; Kadmon, R. Assessing the accuracy of species distribution models: Prevalence, kappa and the true skill statistic (TSS). *J. Appl. Ecol.* **2006**, *43*, 1223–1232. .: 10.1111/j.1365-2664.2006.01214.x. [CrossRef]
80. Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; Xu, Q. SCINet: Time Series Modeling and Forecasting with Sample Convolution and Interaction. *arXiv* **2022**, arXiv:2106.09305.
81. Shao, Z.; Zhang, Z.; Wang, F.; Xu, Y. Pre-Training Enhanced Spatial-Temporal Graph Neural Network for Multivariate Time Series Forecasting. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; Association for Computing Machinery: New York, NY, USA, 2022; KDD '22; pp. 1567–1577. [CrossRef]
82. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
83. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2014**, arXiv:1409.3215.
84. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]