



UNIVERSITY OF AMSTERDAM

MSC ARTIFICIAL INTELLIGENCE  
INTERNSHIP REPORT

---

# Solar Nowcasting using Deep RNNs

---

by  
MELIKA AYOUGHI  
12224855

12 ECTS  
June 2019 — July 2019

*Supervisor:*  
MSc. Fons de Leeuw

*Assessor:*  
Dr. Stratis Gavves



**DEXTER**

## **Abstract**

Weather forecasting is a long-standing scientific challenge with direct social and economic impact. Solar irradiance forecasting that is closely related to weather forecasting is an essential task for energy companies; The presence and movement of water vapor moisture in the satellite images can help analyze solar properties. The combination of weather and solar irradiance forecasting is a suitable task for deep neural networks due to vast amounts of continuously collected data and a rich spatio-temporal structure of data that does not need annotation. We propose to use a deep neural network that forecasts satellite imagery and solar irradiance up to 4 hours ahead. These predictions are at the high spatial resolution of  $1 \text{ km}^2$  and the temporal resolution of 15 minutes with a latency in the order of seconds. Our model takes as input EUMETSAT satellite images with 11 channels and forecasts the preceding satellite images. We evaluate our model's performance by measuring solar irradiance at particular solar farms and comparing it with Numerical Weather Prediction at forecasts of up to 4 hours on continental Europe.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Introduction</b>	<b>1</b>
<b>Background</b>	<b>2</b>
Imbalance . . . . .	2
Solar Power Prediction . . . . .	3
Numerical Weather Prediction (NWP) . . . . .	4
Neural Weather Models (NWM) versus NWP . . . . .	4
<b>Related Work</b>	<b>6</b>
Precipitation Nowcasting and Video Prediction . . . . .	6
Restructured Recurrent Connections for Spatio-Temporal Modeling . . . . .	7
<b>Methodology</b>	<b>8</b>
Input . . . . .	8
Encoding-forecasting Structure . . . . .	8
Convolutional GRU . . . . .	8
<b>Experimental Setup</b>	<b>10</b>
Datasets . . . . .	10
Implementation Details . . . . .	12
Training Details . . . . .	12
Evaluation Metric . . . . .	13
<b>Results</b>	<b>14</b>
Moving MNIST Results . . . . .	14
EUMETSAT Results . . . . .	14
<b>Conclusion</b>	<b>23</b>

# Introduction

Weather forecasting is the prediction of future weather conditions such as precipitation, temperature, pressure, and wind and is fundamental to both science and society. The field joins the forefronts of physical modeling and computing technology into a single century-long scientific and technological effort [1]. The social and economic benefits of accurate weather forecasting range from improvements in our daily lives to substantial impacts on agriculture, energy, and transportation and prevents human and economic losses through better prediction of hazardous conditions such as storms and floods [2], [3].

In addition to the inherent complexities of the atmosphere and relevant dynamical processes, the ever-growing need for real-time, large-scale, and fine-grained precipitation nowcasting poses extra challenges to the meteorological community. It has aroused research interest in the machine learning community as well [4] [5].

Operational and conventional weather forecasts are based on Numerical Weather Prediction (NWP) using physics laws to simulate the atmosphere's dynamics. NWP has seen substantial advances over the preceding decades due to improvements in the representation of the physics, an increase in observational data, and exponential growth in computing capabilities. These aspects have increased the spatial and temporal resolution of NWP and have advanced the range of skillful weather forecasts by about one day per decade. Despite the advances, several challenges remain for NWP [1]. The computational and power demands of NWP grow as a power of the resolution of the forecast. They also create a trade-off between the accuracy and time taken of the forecast. The mathematical formulation of NWP is derived from our current understanding of atmospheric physics, which might be imprecise or cannot be fully resolved at the model's resolution.

In this project, we propose to use a neural network that forecasts satellite imagery and solar irradiance up to 4 hours ahead. These predictions are at the high spatial resolution of  $1 \text{ km}^2$  and the temporal resolution of 15 minutes with a latency in the order of seconds. Our model takes as input EUMETSAT [6] satellite images with 11 channels and forecasts the preceding satellite images. We evaluate our model's performance by measuring solar irradiance at particular solar farms and comparing it with Numerical Weather Prediction at forecasts of up to 4 hours on continental Europe.

# Background

In the last decade, power grids have faced significant changes in their form and usage. One of the driving forces behind this transformation was increasing electricity consumption. In addition, the ever-growing global awareness of climate change directed energy companies into investing in renewable energy resources, like solar and wind energy, which are clean and serve as a good solution for issues regarding sustainability. Nonetheless, connecting these resources to electric grids has recently raised some concerns regarding the generated power fluctuations and their possible negative impacts on network operations.

As a result of these fluctuations, the electric grid could operate unstable, which leads to technical malfunctions. Furthermore, fluctuations could cause high power swings in the feeders as well as unacceptable power outages. Concerning these circumstances, energy suppliers have to calculate the exact amount of their required electricity at a specific time. In a case when the consumption of electricity does not match the generation, a so-called imbalance price can be calculated. The objective of the energy farms is to minimize these imbalance costs. Also, renewable energy supplies production depends greatly on weather conditions, whereas its usage lies mostly in private consumption. Thus, tools to plan and calculate electricity generation of solar and wind forecasts are essential. Dexter Energy Services provides solar and wind forecasts to energy farm operators.

One of the techniques used for very short-range forecasting is *Nowcasting*. In this method, based on the current weather, and an estimate of its speed and movement direction is used to forecast the weather a short period ahead. Hereby it is assumed that the weather moves without notable changes. A short forecast is needed for a detailed description of the current weather since it takes time to gather and map weather observations. Nowcasting also includes forecasts collected by extrapolation for a period of 0 to 6 hours ahead, where it is possible to forecast small features (like individual storms) with reliable accuracy.

## Imbalance

The supply and demand for electricity should always remain in balance, especially since power cannot be stored efficiently. To preserve this equilibrium and make the supply and demand match, buying and selling electricity should be planned precisely by the electricity system's market players. Hence in order to maintain the balance between the inflowing electricity and the withdrawal of electricity from the electricity grid, regulatory methods such as issuing imbalance costs are used.

The following explanations and imbalance cost calculations are only accurate for a specific company (TenneT), which works in the Netherlands and Germany. In general, methods used to calculate imbalance prices are case-specific and reflect the electricity price that has to be added or removed from the grid. In total are three roles in the balancing system:

- Transmission System Operator (TSO): The grid operator responsible for preserving a stable frequency over the grid. They are set for monitoring, maintaining, and

restoring the balance between supply and demand of electrical power in a region. The company TenneT is, in this case, the operator.

- **Balance Responsible Party (BRP):** All parties connected to the electricity grid must be accredited BRPs. According to the law, each BRP is obliged to send a precise schedule to the TSO each day. Any deviance from this trade schedule is considered as an imbalance. The BRP parties are financially responsible for this imbalance. They pay or receive the imbalance price for each imbalance settlement period (ISP), fixed at 15 minutes.
- **Balancing Service Provider (BSP):** The BSP is a market party that provides the TSO with frequency restoration reserves for its balancing task. The main purpose of BSP is to ensure operational security on the grid. On the other hand, the used resources are called Frequency Restoration Reserves (FRR) and have the ability to supply or to store excess electricity production.

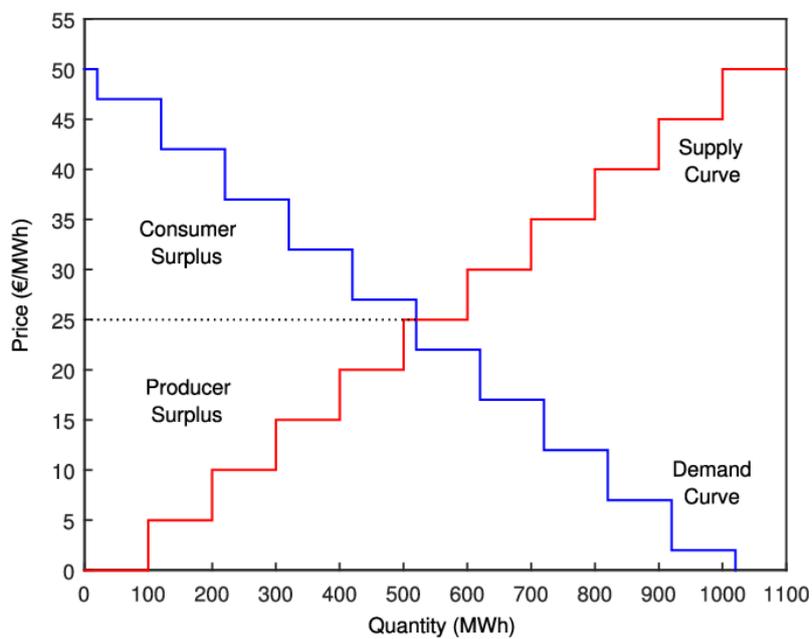


Figure 1: The supply-demand curve that is used for merit order calculating day ahead electricity prices.

## Solar Power Prediction

The amount of solar irradiance exposed to the photo-voltaic panel is a practical metric for calculating solar energy production.

The types of solar irradiance are threefold:

1. **Direct Normal Irradiance (DNI):** The DNI is measured at the unit's surface perpendicular to the sun. The Direct Normal Irradiance only includes the sun's direct beams, while excluding diffuse radiations from other surfaces nearby. Moreover, atmospheric losses are included in the DNI, but they depend on various factors like cloud cover, air pollution, moisture, and others.
2. **Diffuse Horizontal Irradiance (DHI):** The DHI is the second type of solar irradiance, which does not include direct radiations from the sun, but is measured as the light gets scattered by molecules and particles in the atmosphere.

3. Global Horizontal Irradiance (GHI): The GHI is the total irradiance from the sun on a horizontal surface on Earth. It accounts for the zenith angle of the sun.

## Numerical Weather Prediction (NWP)

The most successful framework to perform medium- and long-range forecast to date is Numerical Weather Prediction, which can forecast the weather for the coming up to 6 days with high confidence [1].

The core of NWP models are PDEs and other equations that summarize our current beliefs and information about the dynamical behaviors in the earth's atmosphere. Depending on the use case, target domain, and scale, the exact formulation of NWP models alters. However, the model variables always include surface-based, airborne, and satellite-based measurements, such as temperature, moisture, precipitation, wind fields, and pressure [7]. These models' accuracy relies on the quality of the initial conditions and the mathematical formulation's validity, which means a faithful resemblance of the initial conditions to the atmosphere's current state.

These initial conditions are estimated through a complex data assimilation process in modern NWP models, which leverages a wide range of available information, including past and asynoptic observations, to improve initialization [8]. Despite such effort, initial condition errors are inevitable. These errors negatively affect model performance during the spin-up period [3] and lead to suboptimal short-range forecast. Besides, numerical solving of the PDEs is computationally critical and contains runtimes of more than one hour for regional models such as HRRR.

## Neural Weather Models (NWM) versus NWP

Here we will mention the main differences of NWM and NWP in terms of weather nowcasting [9]:

1. NWM leads directly to a probability distribution, while NWP ensembles a set of deterministic physical simulations using either different initial conditions or model parameters to output probabilistic prediction, see Figure 2.
2. On the other hand, NWM is constructed from general modules that depend on the task's structure but are mostly independent of the underlying domain, such as convolutions, recurrent connections, whereas NWP relies on explicit phenomena-dependent physical equations.
3. NWM performance relies on the data available for the task. Besides, these model's latency does not depend on the target lead time's magnitude so that all lead times can be predicted at once in parallel. While, the simulation in NWP is sequential along time, and its latency scales linearly with target lead time.
4. A comparison between our model and NWP's latency shows that in practice, the latency of DNN based models is in the order of seconds to minutes, for any of the target lead times, whereas for the latter, it is in order of tens of minutes to hours.
5. Another interesting observation investigates the scalability of the currently designed models concerning the underlying resolution. Doubling the spatial resolution requires four times more computation using the current architectural choices for our model, and eight times more computation for NWP [1].

6. NWP's performance is inherently tied to the model's spatial resolution since increased resolution allows direct resolving of more physical phenomena. However, our model's performance is not directly related to its resolution since the network can learn to represent any sub-resolution structure in the hidden layers in a distributed way.

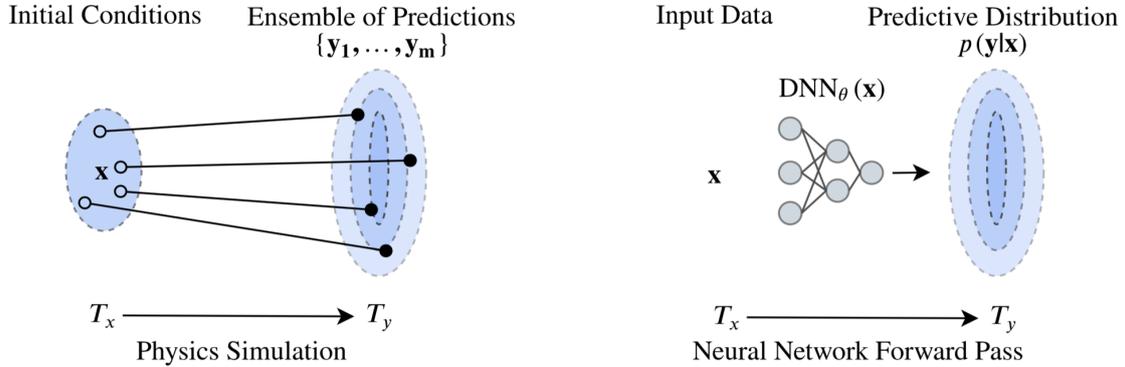


Figure 2: Properties of NWP and NWMs. Left: A deterministic physical simulation starting from the initial conditions is performed by NWP, estimating the predictive uncertainty from an ensemble of predictions at slightly different initial conditions. Right: The NWM approaches the current observations as DNN's direct inputs to directly estimate the distribution over future conditions.  $p(y|x)$ . Image courtesy of [9]

Due to these factors, there is a potential to improve forecasts using real-time and accurate neural network modeling.

# Related Work

Deep neural networks (*DNN*) have seen exceptional progress in recent years due to increased amounts of data, better model architectures, ease of implementation, and powerful specialized hardware such as GPUs and TPUs [10]. The recently introduced DNN architectures can effectively process and use spatial and temporal (spatio-temporal) contexts in the input data [11], [12]. These models can produce probabilistic outputs and represent uncertainty in the predictions. The performance improves with increasing amounts of training data, while the specification of the DNNs remains easy to comprehend and maintain. These properties make DNNs especially promising for weather forecasting/nowcasting:

1. The vast amount of continually collected data from satellites, ground-based radars, and weather stations requiring no human annotation.
2. The immense computational requirements of weather forecasting using conventional simulation methods.
3. The rich spatio-temporal structure of the inputs, and predictions and the inherent need to represent uncertainty.

## Precipitation Nowcasting and Video Prediction

In the precipitation nowcasting problem, radar echo maps' reflectivity factors are first transformed to grayscale and then fed into the prediction algorithm [4]. Precipitation nowcasting can be seen as a video prediction problem with a fixed camera, the weather radar. Hence, the proposed methods for video prediction tasks are applicable to precipitation nowcasting. There are three types of architectures for video prediction: *2D CNN* based models and *3D CNN* based models, and *RNN* based models.

*2D* and *3D CNN* based models are proposed in [13] and [14]. [13] treats the frame sequence as multiple channels and applies *2D CNN* to generate the prediction while [14] treats them as the depth and applies *3D CNN*, see Figure 3 for the general architecture.

The first *RNN* based model for video prediction [16] uses a convolutional *RNN* with  $1 \times 1$  state-state kernel to encode the observed frames. The *LSTM* encoder-decoder network is proposed by [17], which uses one *LSTM* to encode the input frames and another to predict multiple frames ahead. The model's generalization was proposed in [4] by replacing the fully-connected *LSTM* with *ConvLSTM* to better capture the spatiotemporal correlations, which is better suited for image sequence prediction rather than text prediction. Later, the model was extended by predicting the input frame's transformation instead of directly predicting the raw pixels [18], [19]. [20] proposed to use both the *RNN* that captures the motion and the *CNN* that captures the spatial information to generate the prediction.

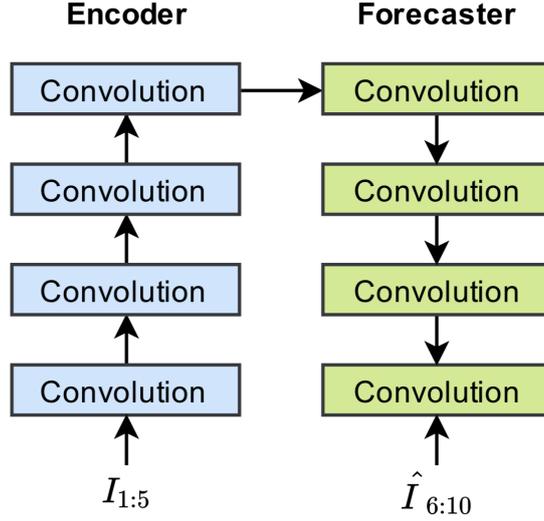


Figure 3: Illustration of the 2D/3D CNN encoder-forecaster. In this example, 4 convolution layers are used to get the representation of the 5 input frames, which is further used to predict the 5 future frames. The convolution blocks can be either 2D or 3D [15].

## Restructured Recurrent Connections for Spatio-Temporal Modeling

From a higher-level perspective, precipitation nowcasting and video prediction are intrinsically spatio-temporal sequence forecasting problems where both the input and output are spatio-temporal [4]. Recently, there has been a trend of replacing the fully-connected structure in the recurrent connections of RNN with other topologies to enhance the network’s ability to model the spatio-temporal relationships. In the *ConvLSTM* [4], the full-connections are replaced with convolutions, which is designed for dense videos. The *SocialLSTM* [21] defines the topology based on the distance between different people and is designed for human trajectory prediction. The *Structural-RNN (S-RNN)* [22] defines the structure based on the given spatiotemporal graph. The *TrajGRU* [15] the model actively learns the location-variant structure for recurrent connections. It aims at learning the local correlation structure for spatiotemporal data. *MetNet* [9] uses a ConvLSTM plus axial self-attention layer to aggregate the global context from a large input patch corresponding to a million square kilometers.

# Methodology

## Input

Our model receives as input a five-dimensional tensor of size [b, t, w, h, c] that corresponds to dimensions batch size, time, height, width and number of channels. The time dimension comprises  $t$  slices sampled every 15 minutes over a 4-hour interval prior to  $T_x$ , where  $T_x$  is the time at which the model makes a prediction into the future. The channels comprise the 11 spectral bands of the EUMETSAT satellite images.

## Encoding-forecasting Structure

We adopt a similar formulation of the precipitation nowcasting problem as in [15]. The satellite input images form a spatio-temporal sequence  $I_1, I_2, \dots, I_x$ . At a given time-step  $x$ , our model generates the most likely next  $K$ -step predictions,  $\hat{I}_{x+1}, \hat{I}_{x+2}, \dots, \hat{I}_{x+K}$ , based on the previous observations. The encoding-forecasting network first encodes the observations into 3 layers of RNN states:  $H_1, H_2, H_3 = h(I_1, I_2, \dots, I_x)$ , and then uses another 3 layers of RNNs to generate the predictions based on these encoded states:  $\hat{I}_{x+1}, \hat{I}_{x+2}, \dots, \hat{I}_{x+K} = g(H_1, H_2, H_3)$ . Figure 4 illustrates the encoding-forecasting structure. There are downsampling and upsampling layers between the RNNs, which are implemented by convolution and deconvolution with stride. The UNet structure helps enhance the low-level states being guided by the high-level states, which have captured the global spatio-temporal representation. Moreover, the low-level states could further influence the prediction. One can choose any type of RNNs like ConvGRU in this general encoding-forecasting structure as long as their states correspond to tensors.

## Convolutional GRU

The main formulas of the ConvGRU [23] used in this work are in Equation 1.  $'*$ ' is the convolution operation and  $'\circ'$  is the Hadamard product. The bias terms are omitted for notational simplicity.  $H_t, R_t, Z_t, H'_t \in R^{C_h \times H \times W}$  are the memory state, reset gate, update gate, and new information, respectively.  $X_t \in R^{C_i \times H \times W}$  is the input and  $f$  is the leaky ReLU [24] activation function with a negative slope of 0.2.  $H, W$  are the height and width of the state and input tensors and  $C_h, C_i$  are the channel sizes of the state and input tensors, respectively. Every time a new input arrives, the reset gate will control whether to clear the previous state and the update gate will control how much the new information will be written to the state.

$$\begin{aligned} Z_t &= \sigma(W_{xz} * X_t + W_{hz} * H_{t-1}) \\ R_t &= \sigma(W_{xr} * X_t + W_{hr} * H_{t-1}) \\ H'_t &= f(W_{xh} * X_t + R_t \circ (W_{hh} * H_{t-1})) \\ H_t &= (1 - Z_t) \circ H'_t + Z_t \circ H_{t-1} \end{aligned} \tag{1}$$

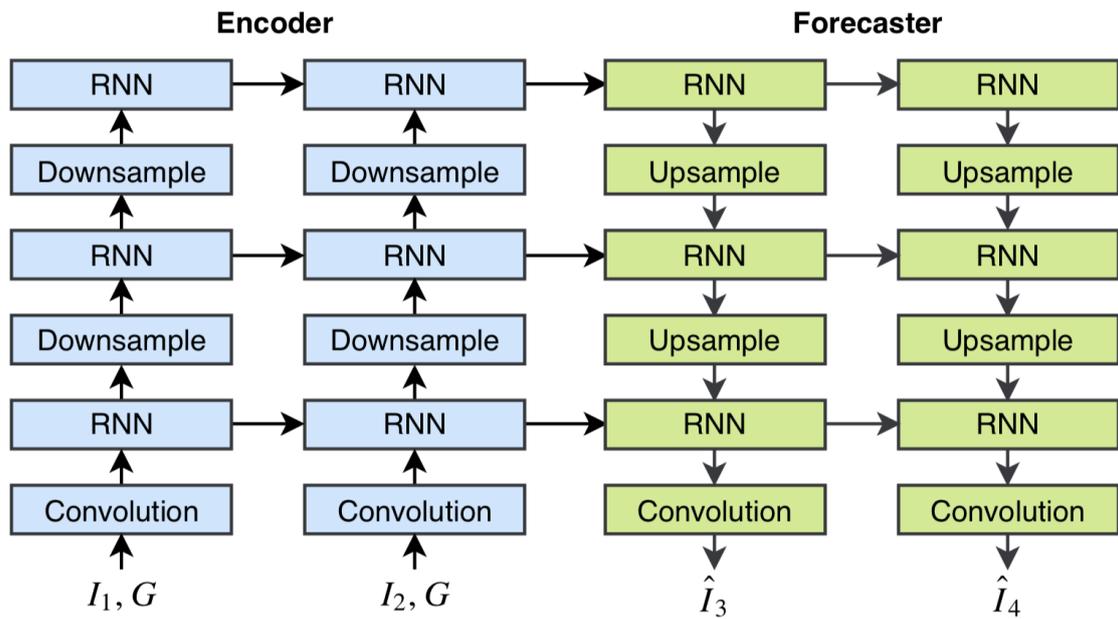


Figure 4: Example of the encoding-forecasting structure used in [15]. In the figure, we use three RNNs to predict two future frames  $\hat{I}_3$  and  $\hat{I}_4$  given the two input frames  $I_1, I_2$ . The spatial coordinates  $G$  are concatenated to the input frame to ensure the network knows the observations are from different locations. The RNNs are ConvGRU blocks. Zeros are fed as input to the RNN if the input link is missing.

# Experimental Setup

## Datasets

### Moving MNIST [4]

In this dataset, each video is 20 frames long and consists of two digits moving inside a  $64 \times 64$  patch. The digits are chosen randomly from the MNIST training set and placed at random locations inside the initial patch. Each digit is assigned a velocity whose direction is chosen uniformly randomly on a unit circle and whose magnitude is also chosen uniformly at random over a fixed range. The digits bounce-off the edges of the  $64 \times 64$  frame and overlap if they are at the same location.

### Dexter Dataset

**EUMETSAT:** EUMETSAT "Rapid Scan High Rate SEVIRI Level 1.5 Image Data - MSG" [6] is an image stream provided by geostationary satellites orbiting the earth. The captured raw data is processed and is one of the Meteosat Second Generation (MSG) system's main products. The designation "Level 1.5" corresponds to image data that has been corrected for all unwanted radiometric and geometric effects. It has been geolocated using a standardized projection and has been calibrated and radiance-linearised. The Level 1.5 data is suitable for deriving meteorological products and further meteorological processing. The SEVIRI instrument is designed to produce the earth's image from a spinning geostationary satellite, see Figure 5.

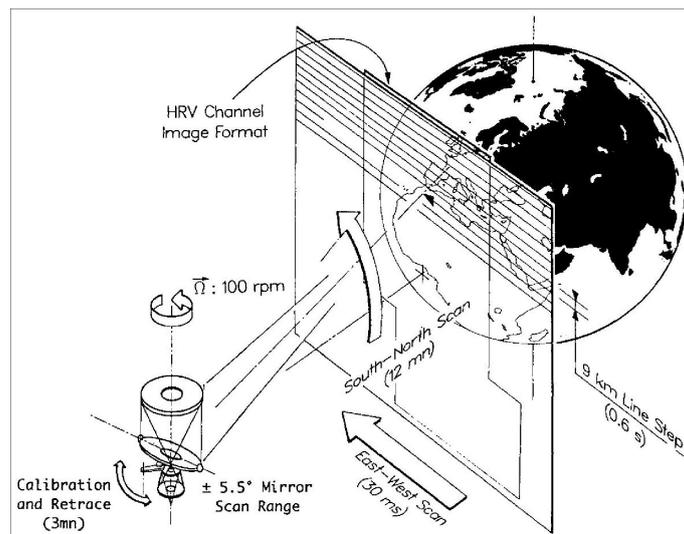


Figure 5: The SEVIRI instrument is designed to produce the image of the Earth from a spinning geostationary satellite. Image courtesy of [25].

It is a time series of images every 5 minutes. The images are provided in 12 channels,

corresponding to the 12 spectral bands. They vary in spectral characteristics, dynamic range, the detectors' operating temperature, and the number of detectors simultaneously acquiring image information. A short description of the 11 spectral channels that were used in this work are the following:

1. VIS0.6, VIS0.8: These channels are in the human visible wavelengths. Used for cloud detection, cloud tracking, location identification, aerosol, land surface, and vegetation monitoring.
2. IR1.6: Near Infrared channel. Discriminates between snow, cloud, ice, and water clouds, provides aerosol information.
3. IR3.9: Primarily for low cloud and fog detection. It also supports the measurement of land and sea surface temperature at night and increases the low-level wind coverage from cloud tracking.
4. WV6.2, WV7.3: These two channels are the water vapor channels for observing water vapor and winds.
5. IR8.7: The channel provides quantitative information on thin cirrus clouds and supports the discrimination between ice and water clouds.
6. IR9.7: This channel contains ozone radiances. It is possible to use it as input to an NWP. Wind motion in the lower stratosphere and the state of the ozone field can be observed.
7. IR10.8, IR12.0: Cirrus clouds, volcanic ash clouds can be detected in this band, and temperatures could be inferred.
8. IR13.4: The CO<sub>2</sub> absorption channel. In cloud-free areas, it will contribute to temperature information from the lower troposphere.

The HRV channel was not used due to the images being much higher resolution and causing errors with memory constraints in the preprocessing phase.

**Train and Validation Splits:** The training data are satellite images from August 2016 to July 2017, from 8:15 to 17:00 every 15 minutes; 36 images per day each with 11 black and white channels, without the HRV channel. The first 18 images per day are given as input, and predictions are compared to the rest of that day's images. The validation data are satellite images with the same characteristics from August 2015 to July 2016.

The solar irradiance measure of 15 solar farms across Europe on the same days is used as ground-truth.

**Preprocessing:** Each image is cropped only to include Europe. Since they are geostationary photos of a geoid shape, they require a computationally heavy preprocessing step to be converted to a near equidistant version, here we use the *Transverse Mercator Projection*. Each image is converted from its geostationary projection with an offset to the east by 9.5° to an equidistant projection. Unfortunately, the ideal equidistant projection is impossible, as the earth is a 3D sphere and the target projection is a 2D plane. The solution is to use a projection where the designated target area, Europe, has little distance error on average. In this project, we use the *Transverse Mercator Projection* because, as shown in Figure 6, Europe is in the middle line of it where there is the least distortion. The images are captured in a geostationary viewpoint. Every pixel represents the same location in all images; thus, a mapping for each pixel to their corresponding latitude and longitude values is created. The PROJ.4 library was used for the conversion. Using a pixel to meters ratio from the data description EUMETSAT (2017), each coordinate's pixel value is calculated and then placed on the original image grid, linearly interpolating between

the pixels of the original image. Finally, the transformed images and a lookup table make the two-way conversion possible, from pixel coordinates to geolocation and reverse. This is useful at the feature extraction phase.

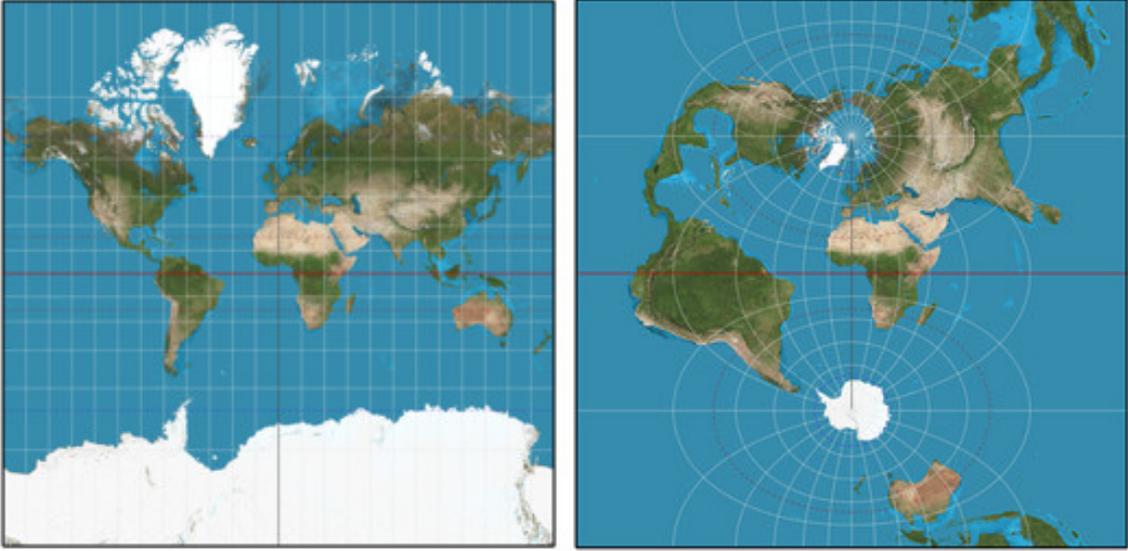


Figure 6: Left: Normal Mercator projection. Right: Transverse Mercator projection. In the tropics, where the cylinder is close to the Earth’s surface, the representations are most accurate. Image courtesy of [26].

**NWP Baseline:** We use the European Centre for Medium-Range Weather Forecasts (ECMWF) [27] data as our baseline for solar nowcasting.

## Implementation Details

The detailed network configurations of our model is described in Table 7.

Name	In Kernel	In Stride	In Pad	State Ker.	State Dila.	Ch I/O	In Res	Out Res	Type	In	In State
econv1	$3 \times 3$	$1 \times 1$	$1 \times 1$	-	-	4/16	$64 \times 64$	$64 \times 64$	Conv	in	-
ernn1	$3 \times 3$	$1 \times 1$	$1 \times 1$	$k \times k$	$d \times d$	16/64	$64 \times 64$	$64 \times 64$	ConvGRU	econv1	-
edown1	$3 \times 3$	$2 \times 2$	$1 \times 1$	-	-	64/64	$64 \times 64$	$32 \times 32$	Conv	ernn1	-
ernn2	$3 \times 3$	$1 \times 1$	$1 \times 1$	$k \times k$	$d \times d$	64/96	$32 \times 32$	$32 \times 32$	ConvGRU	edown1	-
edown2	$3 \times 3$	$2 \times 2$	$1 \times 1$	-	-	96/96	$32 \times 32$	$16 \times 16$	Conv	ernn2	-
ernn3	$3 \times 3$	$1 \times 1$	$1 \times 1$	$k \times k$	$d \times d$	96/96	$16 \times 16$	$16 \times 16$	ConvGRU	edown2	-
frnn1	$3 \times 3$	$1 \times 1$	$1 \times 1$	$k \times k$	$d \times d$	96/96	$16 \times 16$	$16 \times 16$	ConvGRU	-	ernn3
fup1	$4 \times 4$	$2 \times 2$	$1 \times 1$	-	-	96/96	$16 \times 16$	$32 \times 32$	Deconv	frnn1	-
frnn2	$3 \times 3$	$1 \times 1$	$1 \times 1$	$k \times k$	$d \times d$	96/96	$32 \times 32$	$32 \times 32$	ConvGRU	fup1	ernn2
fup2	$4 \times 4$	$2 \times 2$	$1 \times 1$	-	-	96/96	$32 \times 32$	$64 \times 64$	Deconv	frnn2	-
frnn3	$3 \times 3$	$1 \times 1$	$1 \times 1$	$k \times k$	$d \times d$	96/64	$64 \times 64$	$64 \times 64$	ConvGRU	fup2	ernn1
fconv4	$3 \times 3$	$1 \times 1$	$1 \times 1$	-	-	64/16	$64 \times 64$	$64 \times 64$	Conv	frnn3	-
fconv5	$1 \times 1$	$1 \times 1$	$0 \times 0$	-	-	16/1	$64 \times 64$	$64 \times 64$	Conv	fconv4	-

Figure 7: The details of the ConvGRU model. The *In Kernel*, *In Stride* and *In Pad* are the kernel, stride and padding in the input-to-state convolution. *State Ker.* and *State Dila.* are the kernel size and dilation size of the state-to-state convolution.  $k$  and  $d$  are set as stated in [15]. The *In State* is the initial state of the RNN layer.

## Training Details

We use the Mean Squared Error (MSE) loss for training and evaluation. Our model is trained using the Adam optimizer [28] with learning rate equal to  $10^{-4}$  and momentum

equal to 0.5. We use the encoding-forecasting structure introduced previously with three RNN layers. The numbers of filters for the three RNNs are 64, 96, 96 respectively. All RNNs are ConvGRU and all use the same set of hyperparameters. We train for 200,000 iterations with norm clipping threshold equal to 10 and batch size of 4.

## Evaluation Metric

We evaluate the Moving MNIST image results by comparing the MAE of our implementation of ConvGRU with the results in [15], to justify its correct implementation.

We evaluate the EUMETSAT image results of our model against the NWP data by comparing their MSE. The Mean Average Error of GHI solar irradiance values at the specified solar farms between the predictions and the ground-truth is used as an evaluation for how well the model generalizes to solar nowcasting. The evaluation is done for 1/2/3/4 hours ahead and averaged over all solar farms.

# Results

## Moving MNIST Results

Before evaluating our model on the more challenging solar nowcasting task, we first compare the MSE of our implementation of ConvGRU with the results in [15], to justify its effectiveness and correct implementation in Table 1. As the results are close, they suggest correct implementation of the paper.

Table 1: Comparison of ConvGRU implementation and the baseline ConvGRU implemented in [15] on the MovingMNIST dataset.

	our implementation	ConvGRU baseline
#parameters	2.84M	2.84M
Test MSE $\times 10^{-2}$	1.463	1.495

We also evaluate the quality of the generated images by visualizing them in Figures 8, 9, 10. At each iteration, a sequence of 10 images (ground-truth, top row) and the predicted (bottom row) are visualized. The samples are taken randomly every 10000 iterations. We can observe how the model learns to make more precise predictions in time during the training, even with overlapping digits.

## EUMETSAT Results

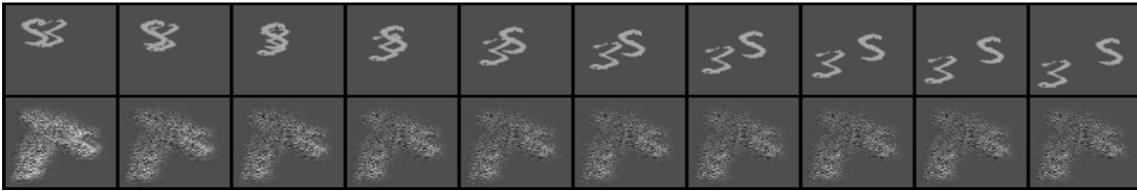
We compare the qualitative performance of our neural weather model with the NWP baseline in Table 2. The closeness of results suggest that we can achieve sufficient performance only using satellite images that use shorter inference time and not as much compute power for weather simulations. As expected, the error is higher as the prediction time-gap increases. The NWP outperforms our model for 1 hour ahead predictions but as time passes by our model outperforms the NWP except for the 4 hour ahead.

Table 2: Comparison of MAE of the NWP and the NWM on the EUMETSAT dataset for the specified solar farms.

	Numerical Weather Prediction	Neural Weather Model
1/2/3/4 hours	<b>0.073</b> /0.098/0.169/ <b>0.184</b>	0.088/ <b>0.097</b> / <b>0.152</b> /0.185

Like for the Moving MNIST dataset, we also evaluate the qualitative performance of the generated images by visualizing only one channel in Figures 11, 12, 13. At each iteration, a sequence of 18 images (ground-truth, top row) and the predicted (bottom row) are visualized. The samples are taken randomly every 10000 iterations. We can observe

how the model learns to make more precise predictions in time during the training. The predictions are less precise and more blurry which is probably why the solar prediction results are not always better than NWP. You can also see the predictions for up to 4 hours ahead are of good quality and probably why our model almost always outperforms the NWP in the long-range.



(a) Iteration 0



(b) Iteration 10000



(c) Iteration 20000



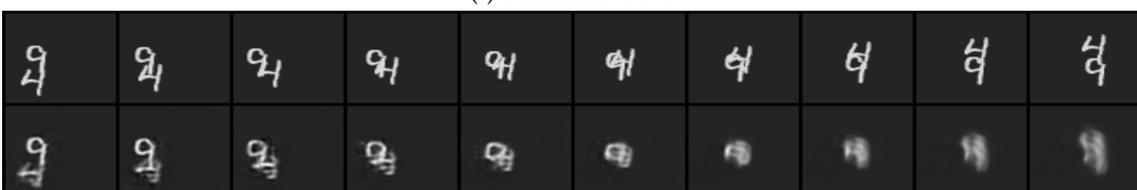
(d) Iteration 30000



(e) Iteration 40000

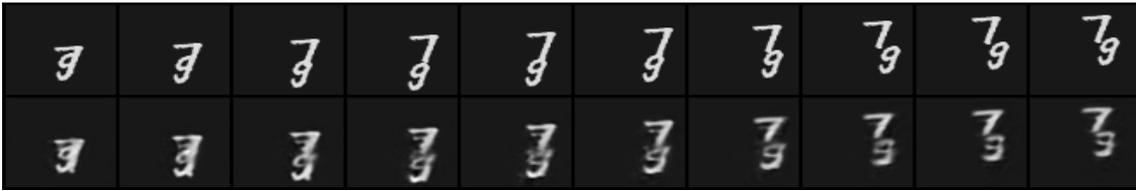


(f) Iteration 50000



(g) Iteration 60000

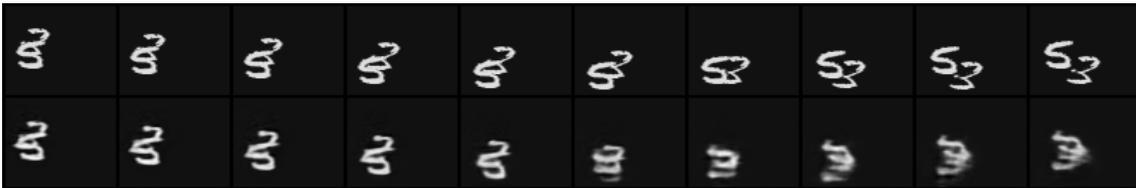
Figure 8: At each iteration, the upper sequence of 10 images are the ground-truth and the lower sequence of 10 images are the predictions. The samples are taken randomly every 10000 Iterations.



(a) Iteration 70000



(b) Iteration 80000



(c) Iteration 90000



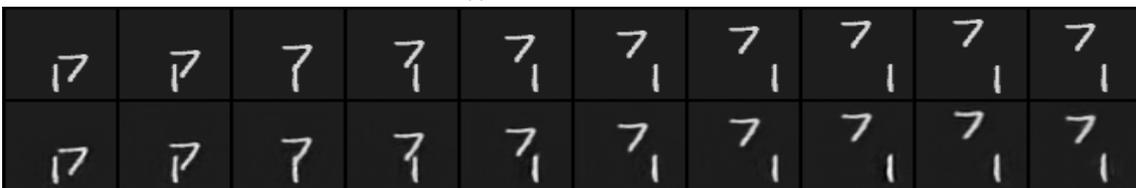
(d) Iteration 100000



(e) Iteration 110000



(f) Iteration 120000



(g) Iteration 130000

Figure 9: At each iteration, the upper sequence of 10 images are the ground-truth and the lower sequence of 10 images are the predictions. The samples are taken randomly every 10000 epochs.



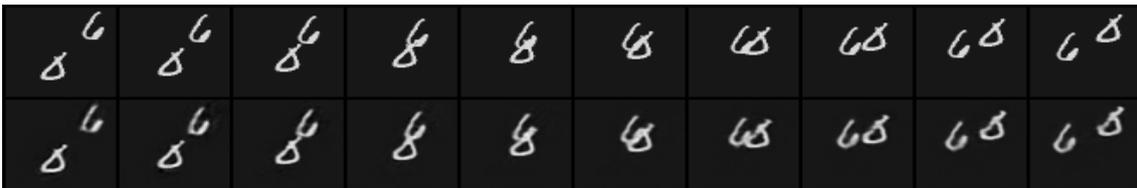
(a) Iteration 140000



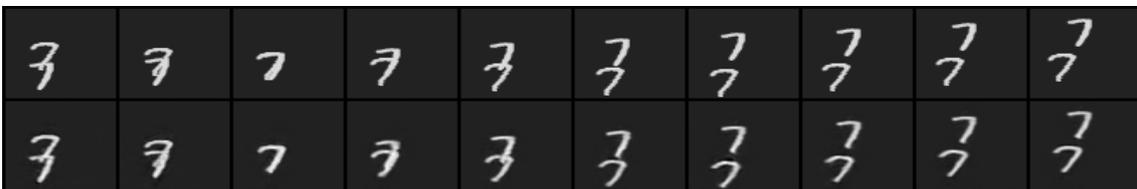
(b) Iteration 150000



(c) Iteration 160000



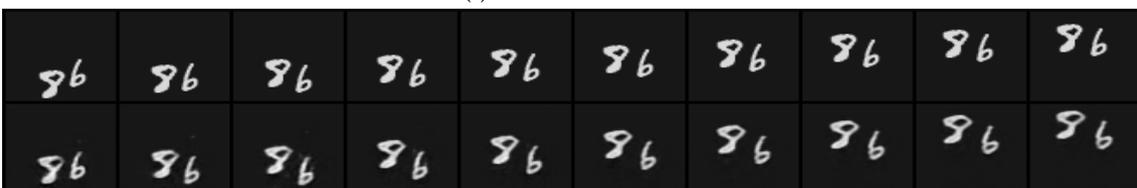
(d) Iteration 170000



(e) Iteration 180000

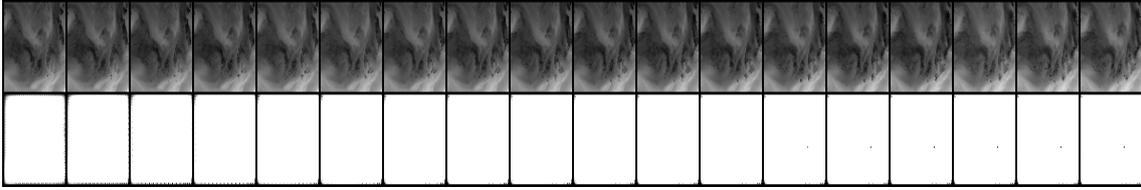


(f) Iteration 190000

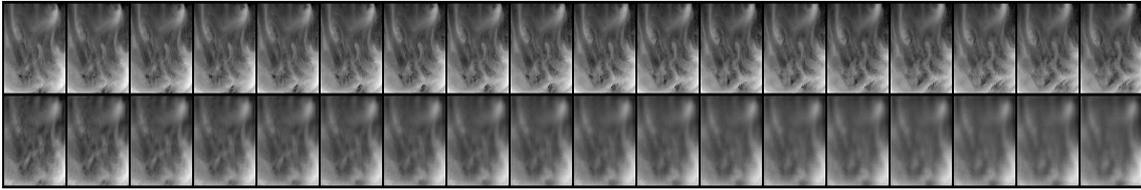


(g) Iteration 200000

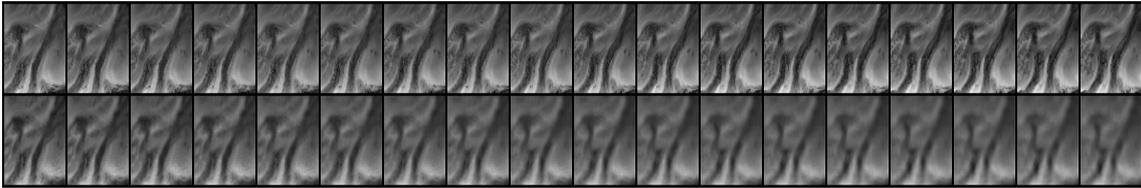
Figure 10: At each iteration, the upper sequence of 10 images are the ground-truth and the lower sequence of 10 images are the predictions. The samples are taken randomly every 10000 epochs.



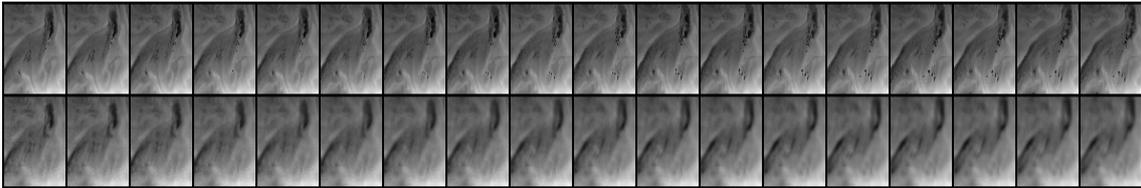
(a) Iteration 0



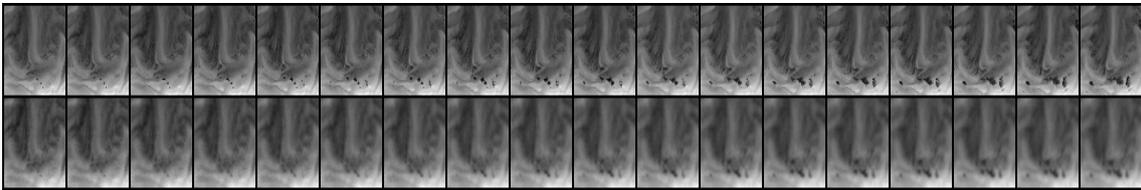
(b) Iteration 10000



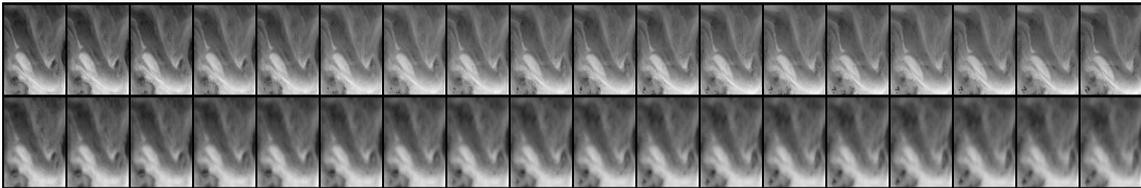
(c) Iteration 20000



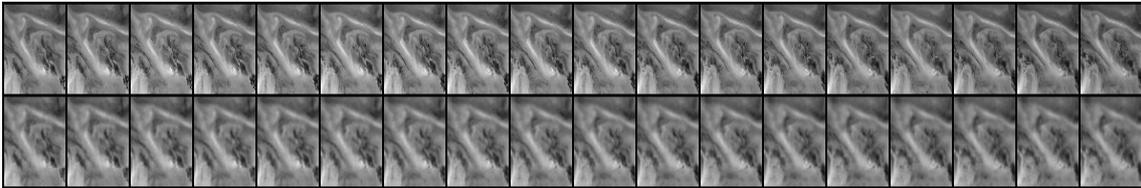
(d) Iteration 30000



(e) Iteration 40000

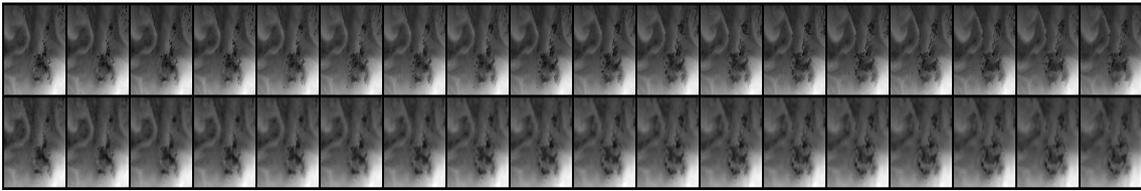


(f) Iteration 50000

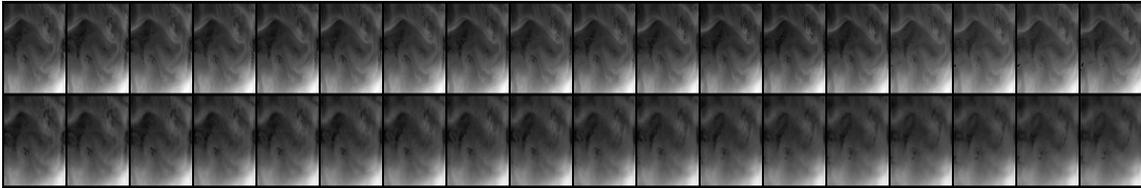


(g) Iteration 60000

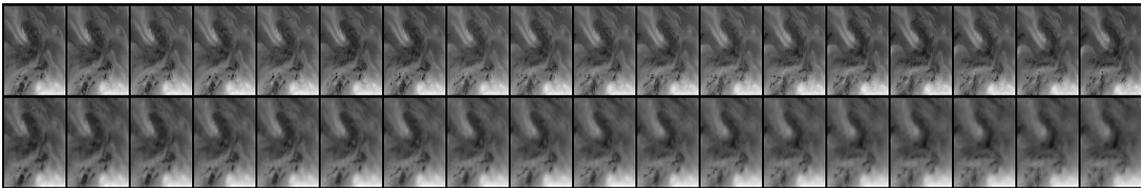
Figure 11: At each iteration, the upper sequence of 18 images are the ground-truth and the lower sequence of 18 images are the predictions for one of the 12 channels. The samples are taken randomly every 10000 iterations.



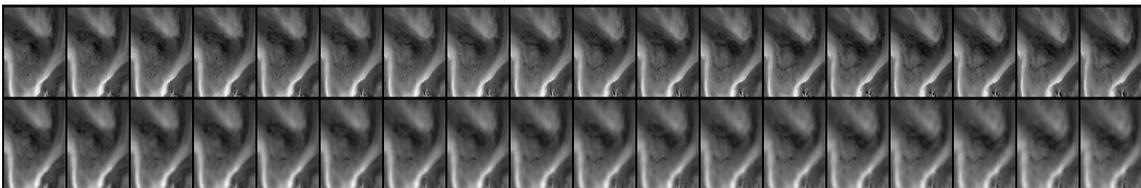
(a) Iteration 70000



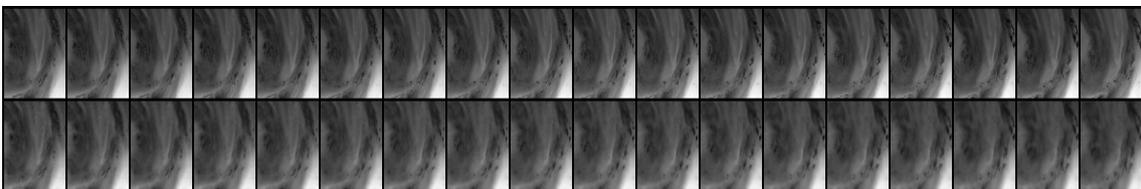
(b) Iteration 80000



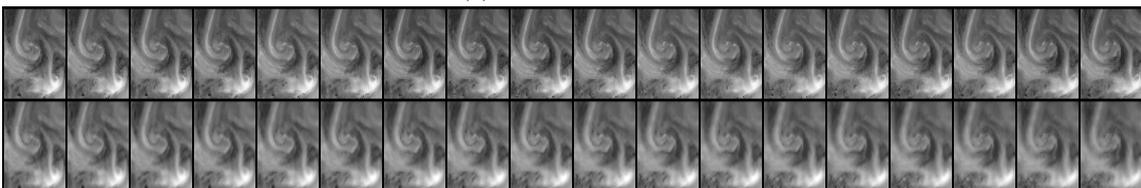
(c) Iteration 90000



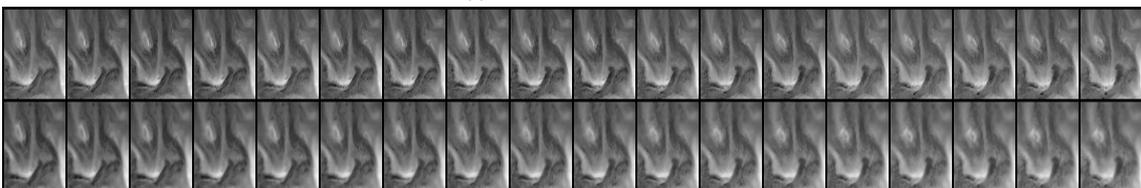
(d) Iteration 100000



(e) Iteration 110000

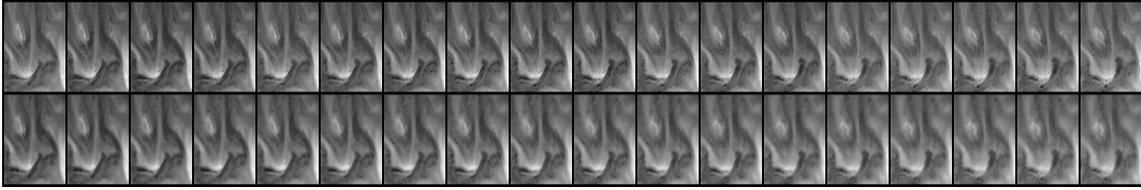


(f) Iteration 120000

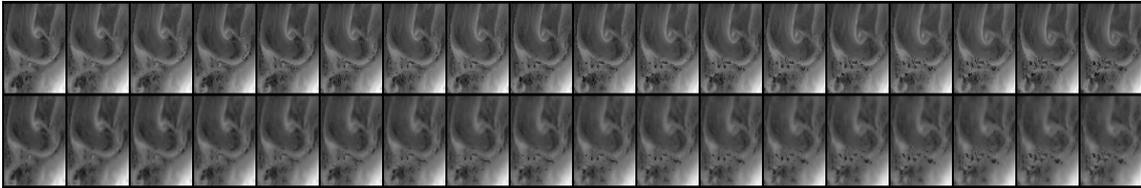


(g) Iteration 130000

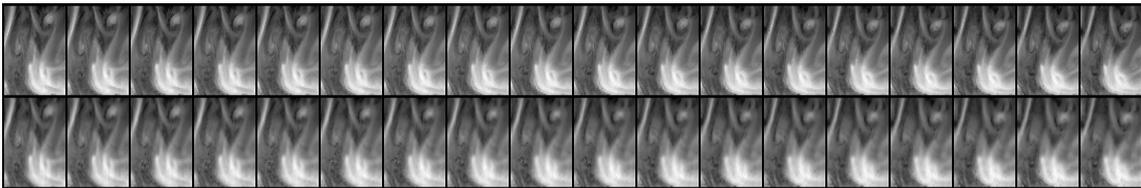
Figure 12: At each epoch, the upper sequence of 18 images are the ground-truth and the lower sequence of 18 images are the predictions for one of the 12 channels. The samples are taken randomly every 10000 iterations.



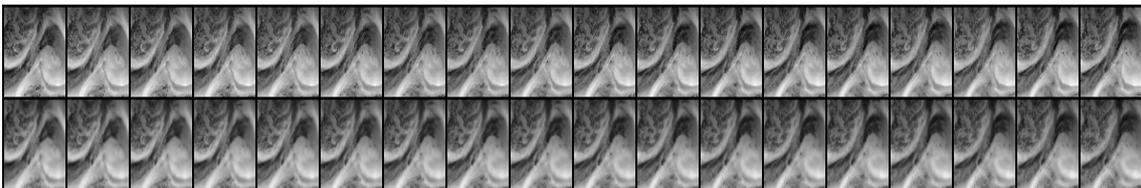
(a) Iteration 140000



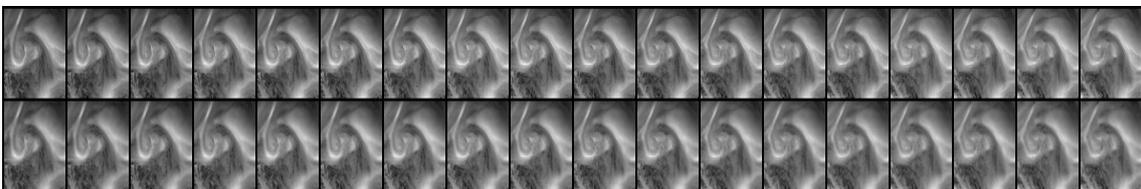
(b) Iteration 150000



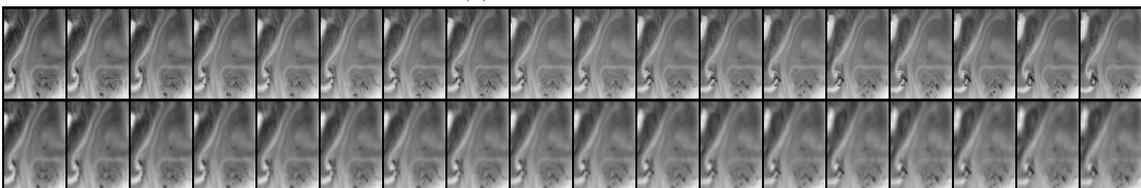
(c) Iteration 160000



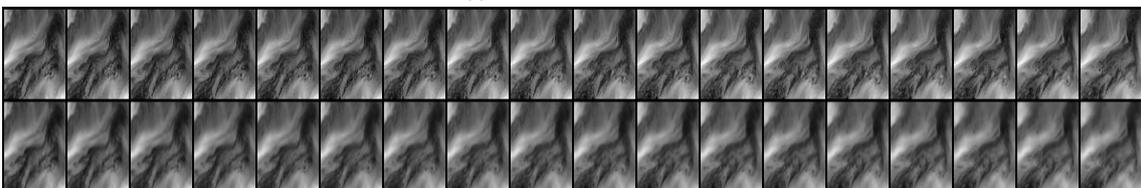
(d) Iteration 170000



(e) Iteration 180000



(f) Iteration 190000



(g) Iteration 200000

Figure 13: At each iteration, the upper sequence of 18 images are the ground-truth and the lower sequence of 18 images are the predictions for one of the 12 channels. The samples are taken randomly every 10000 iterations.

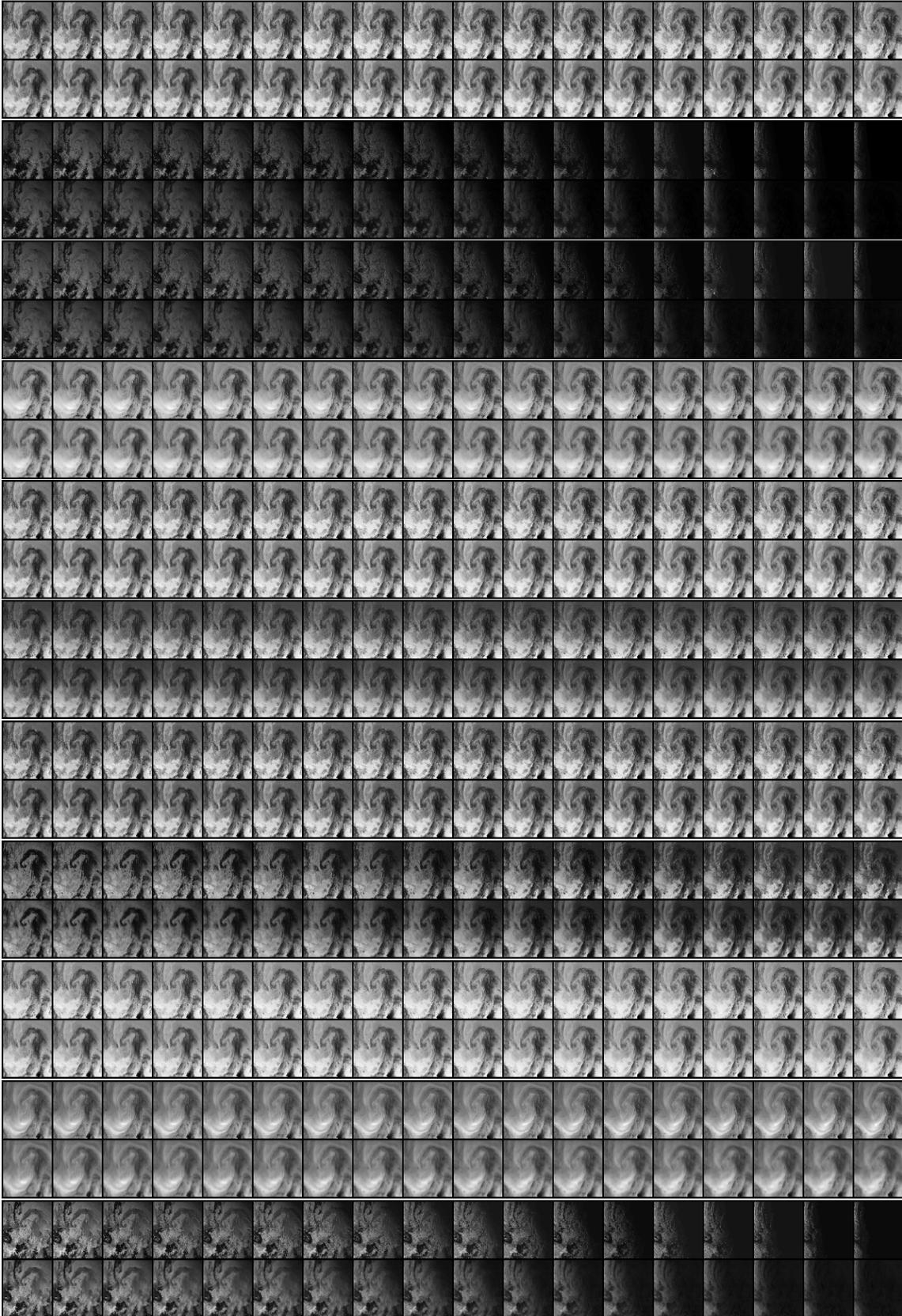


Figure 14: Every 2 rows show the ground-truth and the prediction of a specific weather channel. There are 12 channels in total. This figure shows a random sample from the final trained model.

# Conclusion

In this work, we examined weather forecasting as a long-standing scientific challenge with direct social and economical impact. One of the main weather forecasting applications is in power grid imbalance prediction that helps reduce power fluctuations on the grids. We hypothesized that solar irradiance nowcasting is possible with satellite images since the presence and movement of water vapor moisture in the satellite images can help analyze solar properties. Solar irradiance forecasting is a closely related task to weather forecasting. The combination of weather and solar irradiance forecasting is a suitable task for deep neural networks due to vast amounts of continuously collected data and a rich spatio-temporal structure of data that does not need annotation. We proposed to use a deep neural network that forecasts satellite imagery and solar irradiance up to 4 hours with a spatial resolution of  $1 \text{ km}^2$  and the temporal resolution of 15 minutes and a latency in the order of seconds. We evaluated our model's performance by measuring solar irradiance at particular solar farms and comparing it with NWP at forecasts of up to 4 hours on continental Europe and concluded that neural weather models could achieve almost the same results as in NWP but with less latency and compute power.

# Bibliography

- [1] P. Bauer, A. Thorpe, and G. Brunet, “The quiet revolution of numerical weather prediction,” *Nature*, vol. 525, no. 7567, pp. 47–55, 2015 (cit. on pp. 1, 4).
- [2] S. PC, R. Misumi, T. Nakatani, K. Iwanami, M. Maki, A. W. Seed, and K. Hirano, “Comparison of rainfall nowcasting derived from the steps model and jma precipitation nowcasts,” *Hydrological Research Letters*, vol. 9, no. 4, pp. 54–60, 2015 (cit. on p. 1).
- [3] Y. Hwang, A. J. Clark, V. Lakshmanan, and S. E. Koch, “Improved nowcasts by blending extrapolation and model forecasts,” *Weather and Forecasting*, vol. 30, no. 5, pp. 1201–1217, 2015 (cit. on pp. 1, 4).
- [4] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, *Convolutional lstm network: A machine learning approach for precipitation nowcasting*, 2015. arXiv: [1506.04214](https://arxiv.org/abs/1506.04214) [cs.CV] (cit. on pp. 1, 6, 7, 10).
- [5] J. Sun, M. Xue, J. W. Wilson, I. Zawadzki, S. P. Ballard, J. Onvlee-Hooimeyer, P. Joe, D. M. Barker, P.-W. Li, B. Golding, *et al.*, “Use of nwp for nowcasting convective precipitation: Recent progress and challenges,” *Bulletin of the American Meteorological Society*, vol. 95, no. 3, pp. 409–426, 2014 (cit. on p. 1).
- [6] *Eumetsat: Msg level 1.5 image data format description*, Sep. 2017 (cit. on pp. 1, 10).
- [7] 2017 (cit. on p. 4).
- [8] D. M. Barker, W. Huang, Y.-R. Guo, A. Bourgeois, and Q. Xiao, “A three-dimensional variational data assimilation system for mm5: Implementation and initial results,” *Monthly Weather Review*, vol. 132, no. 4, pp. 897–914, 2004 (cit. on p. 4).
- [9] C. K. Sønderby, L. Espeholt, J. Heek, M. Dehghani, A. Oliver, T. Salimans, S. Agrawal, J. Hickey, and N. Kalchbrenner, *Metnet: A neural weather model for precipitation forecasting*, 2020. arXiv: [2003.12140](https://arxiv.org/abs/2003.12140) [cs.LG] (cit. on pp. 4, 5, 7).
- [10] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, *et al.*, “In-datacenter performance analysis of a tensor processing unit,” in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 2017, pp. 1–12 (cit. on p. 6).
- [11] G. E. Hinton, A. Krizhevsky, and I. Sutskever, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1106–1114, 2012 (cit. on p. 6).
- [12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778 (cit. on p. 6).
- [13] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction beyond mean square error,” *arXiv preprint arXiv:1511.05440*, 2015 (cit. on p. 6).

- [14] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances in neural information processing systems*, 2016, pp. 613–621 (cit. on p. 6).
- [15] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Advances in neural information processing systems*, 2017, pp. 5617–5627 (cit. on pp. 7–9, 12–14).
- [16] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, "Video (language) modeling: A baseline for generative models of natural videos," *arXiv preprint arXiv:1412.6604*, 2014 (cit. on p. 6).
- [17] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*, 2015, pp. 843–852 (cit. on p. 6).
- [18] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Advances in neural information processing systems*, 2016, pp. 667–675 (cit. on p. 6).
- [19] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in neural information processing systems*, 2016, pp. 64–72 (cit. on p. 6).
- [20] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," *arXiv preprint arXiv:1706.08033*, 2017 (cit. on p. 6).
- [21] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971 (cit. on p. 7).
- [22] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016 (cit. on p. 7).
- [23] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," *arXiv preprint arXiv:1511.06432*, 2015 (cit. on p. 8).
- [24] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, 2013, p. 3 (cit. on p. 8).
- [25] *Meteosat second generation - eoportal directory - satellite missions*, <https://directory.eoportal.org/web/eoportal/satellite-missions/m/meteosat-second-generation> (cit. on p. 10).
- [26] *Mercator's marvellous map | thatsmaths*, <https://thatsmaths.com/2015/05/21/mercators-marvellous-map/> (cit. on p. 12).
- [27] *Forecasts | ecmwf*, <https://www.ecmwf.int/en/forecasts> (cit. on p. 12).
- [28] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2014. arXiv: 1412.6980 [cs.LG] (cit. on p. 12).