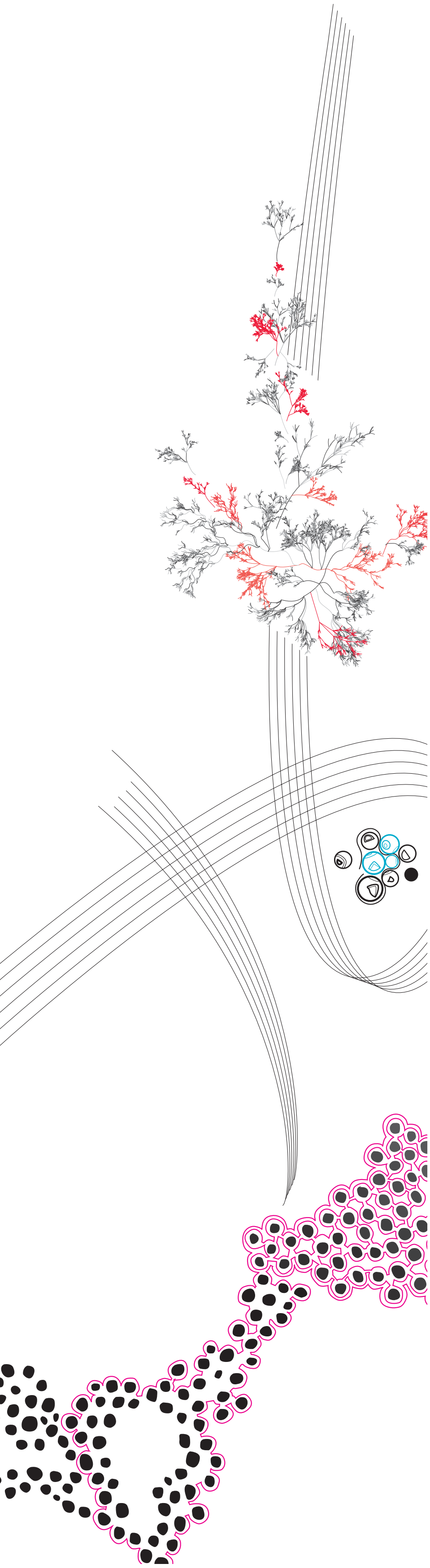
MSc Thesis Applied Mathematics



SG-AIFS: A High-Resolution Deep Learning Approach to Weather Forecasting in Western Europe.

Sophie Buurman

Supervisors: Maurice Schmeits (KNMI)
Kirien Whan (KNMI)
Sophie Langer (University of Twente)
Advisor: Mariana Clare (ECMWF)

December, 2024

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Infrastructuur en Waterstaat

UNIVERSITY OF TWENTE.

SG-AIFS: A High-Resolution Deep Learning Approach to Weather Forecasting in Western Europe.

Sophie Buurman[1]

December, 2024

[1] Email: sophie.buurman@outlook.com

Preface

“ Wees trots, wees een rots, en laat het los ” - Nina (Alex) Buurman

I want to express my gratitude to all the people who have supported me throughout this Master thesis project. First and foremost, I would like to thank KNMI, especially Ben Wichers-Schreur and Werenfried Spit, for providing me with the incredible opportunity to work on such an exciting project. Their sponsorship of my travels to Reading, Bonn, and Oslo allowed me to collaborate with experts from various institutions, which greatly enriched my experience and learning. I am deeply thankful my supervisors at KNMI, Maurice Schmeits and Kirien Whan, for their guidance and for their valuable feedback as well as for the interesting discussions we had during our weekly meetings. My sincere appreciation to Sophie Langer, whose support throughout this project —and even beyond— has been a source of great strength. Thanks to all my colleagues at KNMI, who were always there for a chat during lunch breaks, during which I got to know them professionally as well as personally. I especially want to thank Jasper Wijnands and Bastien Francois, who were there during my travels as part of the ML Pilot Project. Their collaboration and shared enthusiasm helped lighten the burdens and amplify the excitement of the work. Special thanks go to Bert van Ulft and Michal Koutek for their patience and expertise in answering my many NWP-related questions. From ECMWF, I am grateful to Mariana Clare, who has been my advisor, main contact person and support throughout the project. Additionally, I want to extend my gratitude to Florian Pinault, Mario Santa Cruz and Sara Hahner for answering my questions regarding anemoi-datasets, anemoi-graphs and the anemoi imputer, respectively. From MET Norway, I would like to thank the entire BRIS team —Thomas Nils Nipen, Håvard Homleid Haugen, Magnus Sikora Ingstad, Even Marius Nordhagen, Aram Farhad Shafiq Salihi, Paulina Tedesco, Ivar Ambjørn Seierstad, and Jørn Kristiansen— for their collaboration. I am especially thankful to Magnus and Håvard for their patience with my questions and for their enthusiasm and dedication, which made working with them a true privilege. I am incredibly grateful to have friends and family who showed great interest in my work. I am especially thankful to my mom, Louise Wipfler, for her moral and technical support. Her willingness to sit with me during the most stressful moments, calmly working through problems together, was a true lifesaver. My dad, Kit Buurman, and sibling, Nina (Alex) Buurman, have always been my safe harbour, providing comfort and understanding after long and demanding days. To my friends from the AM Master —Nina, Karisma, Jose, Fabio, Renske, and Steven- I am grateful for the wonderful memories we shared together during this time. Sharing this journey with you has been a joy and an inspiration. To my other friends outside the study not mentioned here, please know that your support has meant the world to me, and I am forever grateful for your presence in my life. Finally I would like to thank Alex Duan, who was there for me during my most difficult periods and who always provided a safe space and encouraged me all the way through. Alex, I love you. To everyone else who contributed to this journey: thank you for turning this MSc thesis project into an interesting and enjoyable experience.

Abstract

Traditionally, weather forecasting relies on numerical weather prediction (NWP) models. However, recent advances in the field have demonstrated that deep-learning based weather prediction (DLWP) models can successfully be trained on historical re-analysis (ERA5, 0.25 degree resolution) data, to produce accurate global medium-range forecasts. Two DLWP approaches can be distinguished: Graph Neural Networks (GNNs) and Transformer models. Expanding on these approaches, the European Centre for Medium-range Weather Forecasts (ECMWF) has developed a global Graph-Transformer model based on GraphCast from Google DeepMind, called the Artificial Intelligence Forecasting System (AIFS). This MSc thesis investigates the possibility of extending these new developments to high-resolution modeling on a limited domain, by adapting AIFS to include a stretched grid (SG-AIFS) using refined hidden grid layers in the processor step. This research was done in collaboration with MET Norway, whose evaluation on surface observations has outperformed their operational HARMONIE-AROME NWP model on certain variables, although showing underestimation of extremes [29]. In this research, the DOWA (Dutch Offshore Wind Atlas) dataset - a reanalysis from the 2.5-km HARMONIE-AROME NWP model of KNMI - is used to integrate with the lower-resolution ERA5 data and is subsequently connected to the stretched grid. First, different processor refinements are assessed by evaluating hidden grid sizes using ERA5 data. We find that although increasing processor refinements accelerates training time, it results in marginal improvements over longer lead times. On the other hand, rollout training proved essential in reducing RMSE values across all lead times. These results are employed to train the model on the DOWA dataset, producing high-resolution deterministic forecasts whilst minimizing computational resources. The model provides +6h predictions with some accuracy, although it lacks detailed features and longer lead times show artefacts resembling the processor hidden grid structure.

Keywords: numerical weather prediction, weather forecasting, ECMWF, ERA5, HRES, deep learning, machine learning, data-driven forecast, graph neural networks, stretched grid

Contents

Chapter 1

Introduction

From agriculture to the energy transition and from recreational use to the warning for high-impact weather extremes, the importance of accurate meteorological forecasts is evident in all aspects of life. Numerical weather prediction (NWP) is the established practice used to calculate the future weather state based on our knowledge of the underlying physical processes. The Navier–Stokes and mass continuity equations (including the effect of the Earth's rotation), together with the first law of thermodynamics and the ideal gas law, describe the governing processes that determine the state of the atmosphere. The non-linear Navier-Stokes equations have no analytic solution, hence numerical integration is necessary, starting from the current weather state and using spatial and temporal discretization.

There are numerous small-scale processes essential for the predictability of the weather, that remain uncaptured within these discretizations. The parametrization of these unresolved processes is a requirement for accurate meteorological predictions. Moreover, due to the chaotic nature of the atmosphere, forecasts at longer lead times are inherently sensitive to their initial conditions. The smallest initialization error can lead to large uncertainty at prediction time [7]. This has elicited the introduction of ensemble forecasting, where the initial conditions of an NWP model are perturbed, resulting in probabilistic weather forecasts. Deepened understanding, increased computational power (which has led to increasing resolution) and improved model parameterization quality and ensemble forecasting have steadily advanced NWP models over the past decades and this is also called "The quiet revolution of NWP" [2].

A new development, namely data-driven weather prediction (DLWP), has made a rapid rise in the last few years, trying to improve on NWP models by learning the underlying processes implicitly from historical data. Due to the large amount of available data (originating from observations used in data assimilation), rapid model development, and the optimization of computational resources, deep learning methods have emerged as a promising addition to NWP models. In particular, from the end of 2022 onward, deep learning models have shown comparable skill to NWP models on a limited number of variables [7]. Moreover, DLWP models, once trained, are able to provide forecasts in minutes, whereas NWP models require hours of computation time to produce a single forecast. DLWP methods for probabilistic forecasting have recently been developed, featuring high-quality forecast ensemble members that aim to represent the forecast uncertainty [33].

Since 2013, KNMI (Royal Netherlands Meteorological Institute) has used a high-resolution NWP model called HARMONIE-AROME (HA) to make short-range predictions. HARMONIE is a limited area weather prediction model, which provides an hourly output on a limited domain as can be seen in Figure 4.1. HARMONIE is based on AROME (Application of Research to Operations at Mesoscale), developed by Météo-France. The increased resolution of 2.5km allows the model to resolve deep convective processes (associated with showers) that are hard to model at lower resolutions [4].

An important next step in the development of DLWP models is to consider the extension to limited-area modeling similarly to HARMONIE-AROME [31]. In collaboration with MET Norway [29], a stretched grid approach is investigated in this thesis, where the hidden grid of a global Graph Neural Network (GNN) is locally refined on the area of interest. The stretched grid will be implemented by adapting the Artificial Intelligence/Integrated Forecasting System (AIFS) [24]. AIFS is a global DLWP model, developed by ECMWF, based on the original Graph Neural Network called GraphCast, developed by Google Deepmind [22]. AIFS is trained on ERA5 reanalysis data, and combines the flexibility of GNNs with the speed optimization of Transformer networks.

We are using the DOWA[1] (Dutch Offshore Wind Atlas) dataset [39], based on reforecasts from the high-resolution limited-area HA model, centered on the Netherlands. In this research, we integrated DOWA with the lower-resolution ERA5 data and connected it to the stretched grid. By making clever use of the flexibility of the model architecture, we want to improve the continuity of the information flow and increase the horizontal resolution on a regional domain.

The objective of this MSc thesis is to investigate the effectiveness of the stretched grid model proposed by [29] on the DOWA domain. Specifically, this investigation seeks to address the following research questions:

1. What is the influence of increasing resolution in the processor hidden grid of the SG-AIFS model?
2. How does the performance of the SG-AIFS Transformer model compare to that of the SG-AIFS GNN model?
3. How does including the rollout step described in [22], [23] and [29] influence the model performance?
4. What is the impact of directly training on 2.5km resolution on the relevant variables?
5. How effectively does the 2.5km-resolution SG-AIFS model capture extreme events?
6. How does the performance of the 2.5km-resolution SG-AIFS model compare with the operational HA model?

To address these questions, this MSc thesis begins with a review of the most recent advances in global deep learning for weather forecasting, with a particular emphasis on GNN models. The AIFS model and stretched grid adaptation are introduced, and their capabilities are discussed. The SG-AIFS model is then tuned and validated by training on the lower resolution ERA5 dataset. This process involves evaluating the impact of model architecture choices, such as autoregressive rollout fine-tuning, the number of processor refinements and structural modifications to the processor. Subsequently, a model is trained on the ERA5 reanalysis data at 1 degree latitude/longitude resolution, combined with the DOWA dataset at 2.5km spatial resolution. The performance of this model is assessed through two primary metrics - Root Mean

[1] `https://www.dutchoffshorewindatlas.nl/about-the-atlas/dowadata/data-info`.

Square Error (RMSE) and spectral power analysis - focusing on key variables such as the 2-meter temperature and the 10-meter winds speed. Finally, a case study is presented in which the SG-AIFS model's performance is benchmarked against that of the (previously) operational HA model during a storm event that occurred on February 22-23, 2017. This comparison aims to provide insights into the operational viability, weaknesses and strengths of the SG-AIFS model in capturing extreme weather phenomena.

Chapter 2

Graph Neural Networks

Graph Neural Networks (GNNs) arise in fields where information is defined according to the relationships between points of information. They are permutation invariant, meaning their outcome is not influenced by the ordering of connected nodes. Additionally, GNNs efficiently reuse functions across the graph domain, making the functions easily transferable between different graph structures. As a result, they have been successful in a wide range of problems, such as image classification and machine translation. Furthermore, GNNs have been applied effectively in the modeling of dynamical systems and hence provide a promising direction for meteorological and climatological research. In this chapter, we will introduce the mathematical notion of neural networks, specifically multi-layer perceptrons and convolutional neural networks. From these concepts, GNNs are formally introduced, and the relevant DLWP models are explained.

2.1 Mathematical framework

2.1.1 Neural networks

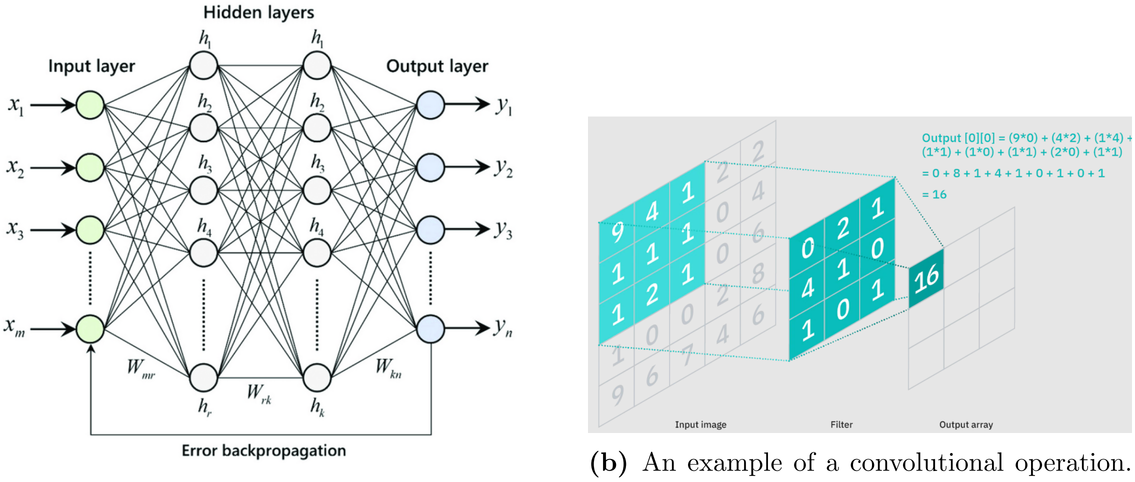
The goal of machine learning is to approximate a function underlying a given dataset, and find the corresponding parameters such that the resulting prediction will perform well on unseen data [12]. A neural network is a type of machine learning model inspired by biological neurons to mimic the structure and functionality of human learning. Vanilla feedforward neural networks, also referred to as multi-layer perceptrons (MLPs), are often represented as a graph, where the neurons are called nodes and the connections between neurons called the weights. Deep neural networks consist of layers of neurons; an input layer, an output layer and L hidden layers in between. An example of a deep neural network can be found in Figure 2.1a.

A deep neural network can mathematically be defined as a function $f_\theta : \mathcal{X} \longrightarrow \mathcal{Y}$, where $\mathcal{X}$ is the input space, $\mathcal{Y}$ the output space and the parameters $\theta \in \mathbb{R}^n$ the weights of the network. The output of the l^{th} layer of the network is denoted by y^l. Each layer processes the output of its preceding layer by a linear transform, followed by an activation function:

$$\begin{aligned} &y^{l+1} = \sigma(\theta^{l+1} y^l + b^{l+1}) \quad l \in \{1, ..., L-1\}, \\ &f_\theta = Y^L \end{aligned}$$

where θ^l and b^l are called the weights and biases of layer $l \in \{1, ..., L\}$ and σ is the activation function, e.g., sigmoid, ReLU or Tanh.

The weights and biases of the network are initialized randomly. The network learns parameters that can be used to optimally carry out a task by *training* on the data. The desired output



(a) An example of a deep neural network. The function takes m inputs and passes it through a number of hidden layers which perform a linear transformation and a non-linear activation to produce n outputs. Image adapted from [16].

(b) An example of a convolutional operation. A dot product operation between the kernel and the input image is performed to obtain the final value in the output array. In a CNN, the kernel values will be the weights of the network. Image adapted from [19].

Figure 2.1: An example of a deep neural network (left) and a convolutional operation (right).

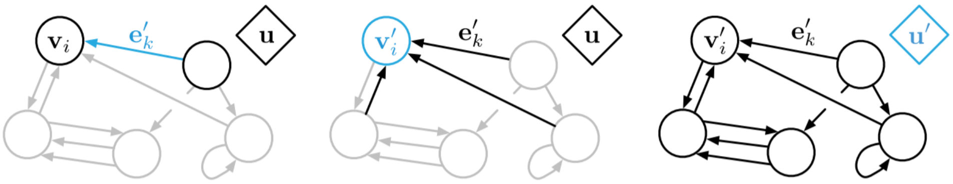
of the network, given a specific input, is provided by the ground truth data. During training, the discrepancy between the output of the network and the desired output is measured by a *loss function*. The loss function is then minimized using the gradient descent method, proceeding backward through the network and adjusting the values of the parameters in each step in the direction of the gradient. This process is called *backpropagation* and the size of the gradient descent step is called the learning rate. After training, the parameters are fixed and the model is expected to generalize to data it has not seen during training.

Convolutional neural networks (CNNs) are developed for structures with a grid-like topology, such as images. A CNN is a neural network that takes an image as input, and processes local image information using convolutional operations. Convolutional operations pass a symmetrical kernel over each position of the image, performing a dot product operation for all overlapping voxels. An example of a convolutional operation can be seen in Figure 2.1b. The kernel's middle value will be replaced with the calculated value in the image. Similar to regular neural networks, the values of the kernel are trained by the CNN. Placing several convolution layers subsequently allows for a growing receptive field and enables the network to extract distinctive features from the image, that can be used for classification, segmentation or regression [18].

2.1.2 Graph Neural Networks

GNNs were introduced in 2009 by [35], extending on CNNs for domains consisting of patterns and relationships. Generally, GNNs consist of a cascade of Graph Network (GN) blocks. The message passing steps inside the GN blocks can intuitively be seen as performing a convolutional step on an irregular grid. Instead of inferring information from a specific rectangular filter, the number of neighbour nodes is not restricted to a grid. The passing of information is generalized to a non-euclidean domain, but still retains the idea of sparsifying the network by aggregating information based on proximity [28].

A graph is defined as $G = (u, V, E)$ where V represent the n vertices of the graph, with each



(a) Edge update (b) Node update (c) Global update

Figure 2.2: Updates in a GN block. Blue indicates the element that is being updated, and black indicates other elements which are involved in the update (note that the pre-update value of the blue element is also used in the update). The notation in the figure differs from the notation used in equation (2.1) - (2.6). Figure taken from [1].

v_i in V, $i \in [n]$ being a vector with the node attributes. E denotes the edge set, with attribute vector $e_{i,j}$ in E if and only if there is a bidirectional edge between node v_i and node v_j. Finally, u contains the global attributes of the graph, which refer to properties of the graph as a whole. A Graph Network (GN) block consists of three update functions ϕ, and three aggregation functions ρ, for updating the edges, nodes and global attributes, respectively. Usually, the ϕ functions are multi-layer perceptrons (MLPs), learning the mappings between the individual components. The ρ functions aggregate the attributes by using a permutation invariant function, usually the sum function, to map the attributes to a single value. Due to the property of graphs to label nodes arbitrarily, it is crucial that the ρ functions are permutation invariant. Note that these functions are all shared across the domain, thus remaining effectively independent of the hidden graph structure. We describe the updates of the ρ and ϕ functions for the edges, nodes and global attributes in order, as shown in Figure 2.2.

In the first step, the edge attributes for all edges are updated, using the attributes from the neighbouring nodes (Figure 2.2a). Applying the edge update function ϕ^e and the aggregation function $\rho^{e\to v}$ gives the following edge update steps:

$$e'_{i,j} = \phi^e(e_{i,j}, v_i, v_j, u) \quad \forall e'_{i,j} \in E \tag{2.1}$$

$$\bar{e}'_k = \rho^{e\to v}(E'_k) \quad \forall v_k \in V, \tag{2.2}$$

where $e'_{i,j}$ are the updated edges (computed using ϕ^e), going from node v_i to node v_j, u are the global attributes and $\bar{e}'_k$ is the aggregation (computed using $\rho_{e\to v}$) of these updated edges.

Secondly, each node is updated by using the edge information from the neighbouring edges (Figure 2.2b). Let v'_k be the updated nodes, then the corresponding node update function ϕ^v is defined as follows (Node update):

$$v'_i = \phi^v(\bar{e}'_k, v_k, u). \tag{2.3}$$

Finally, we update the global attributes using the aggregated information from the updated nodes and edges (Figure 2.2c). Let $\bar{e}'$ be the globally aggregated edge updates, let $\bar{v}'$ be the globally aggregated node updates and let u' be the updated global attributes. Then the edge aggregation function $\rho^{e\to u}$, the node aggregation function $\rho^{v\to u}$ and the global update function

ϕ^u are given as follows (Global update):

$$\bar{e}' = \rho^{e \rightarrow u}(E') \tag{2.4}$$

$$\bar{v}' = \rho^{v \rightarrow u}(V') \tag{2.5}$$

$$u' = \phi^u(\bar{e}', \bar{v}', u). \tag{2.6}$$

Note that these updates do not necessarily have to be in this order. This framework has been shown to excel at the representation of relationships between nodes and learn the generalization of these relationships across the network [1].

2.2 GraphCast

All GNN models discussed subsequently are based on GraphCast, introduced by Google DeepMind [22]. These specific GNNs consist of GN blocks in an encoder-processor-decoder architecture with a so-called 'hidden grid' (see Figure 2.3g). The input graph (in our case, the ERA5 reanalysis latitude/longitude grid) is transformed into a latent representation (the hidden grid) by a GN encoder. A number of graph message passing steps are performed on this latent space graph, after which it is transformed back to the original grid by a final GN decoder. Since network representations have shown to be efficient and accurate in learning complex physical dynamics (e.g. fluids), [22] argue that they should be suitable for predicting future weather states from physics-based training data, especially considering their ability to effectively sparsify the number of interactions. By learning the elements of the latent representation that interact with each other, GNNs transfer information over a long range. This is particularly useful for longer forecast lead times, where the state of the atmosphere has a larger range of influence. Transformers also allow these long-range connections, however they often have to be spatially reduced due to their computational complexity [22]. The authors of GraphCast implement faster long-range connections by utilizing a 'multi-mesh', as depicted in Figure 2.3. This latent-representation consists of an iteratively refined icosahedron, projected on the unit sphere. These refinements are then combined into a multi-mesh and allow for different scales of information transferability.

2.2.1 Network structure

The node attributes of GraphCast contain the weather state information for each input grid point, as well as constants for the hidden grid (such as the sine/cosine of the longitude and latitude). The edge features consist of four attributes calculated from the position on the unit sphere of the connecting nodes. There are no global attributes, so the updates as described in equations (2.4) - (2.6) are not considered. Furthermore, to reduce the dimensionality of the attributes, GraphCast encodes the features described above into a latent space of fixed dimension using five multi-layer perceptrons (MLPs).

The grid input data is connected to the multi-mesh based on proximity. Each hidden grid point is connected to all data points within a range of 0.6 times the length of an edge of the hidden graph layer. The graph connecting the data and the hidden grid is called the encoder graph. Every mesh point corresponds to about 25 grid points on average. Note that since this graph is directed, the order of the edges matters ($e_{i,j} \neq e_{j,i}$). All edges are directed towards the hidden grid. During the encoding step, the information about the state of the atmosphere is transferred by performing a single message-passing step over this encoder graph. During this message passing step, all edges and nodes are updated, but only the hidden grid nodes are aggregated with the information from the edges. This encoding is the first layer of the

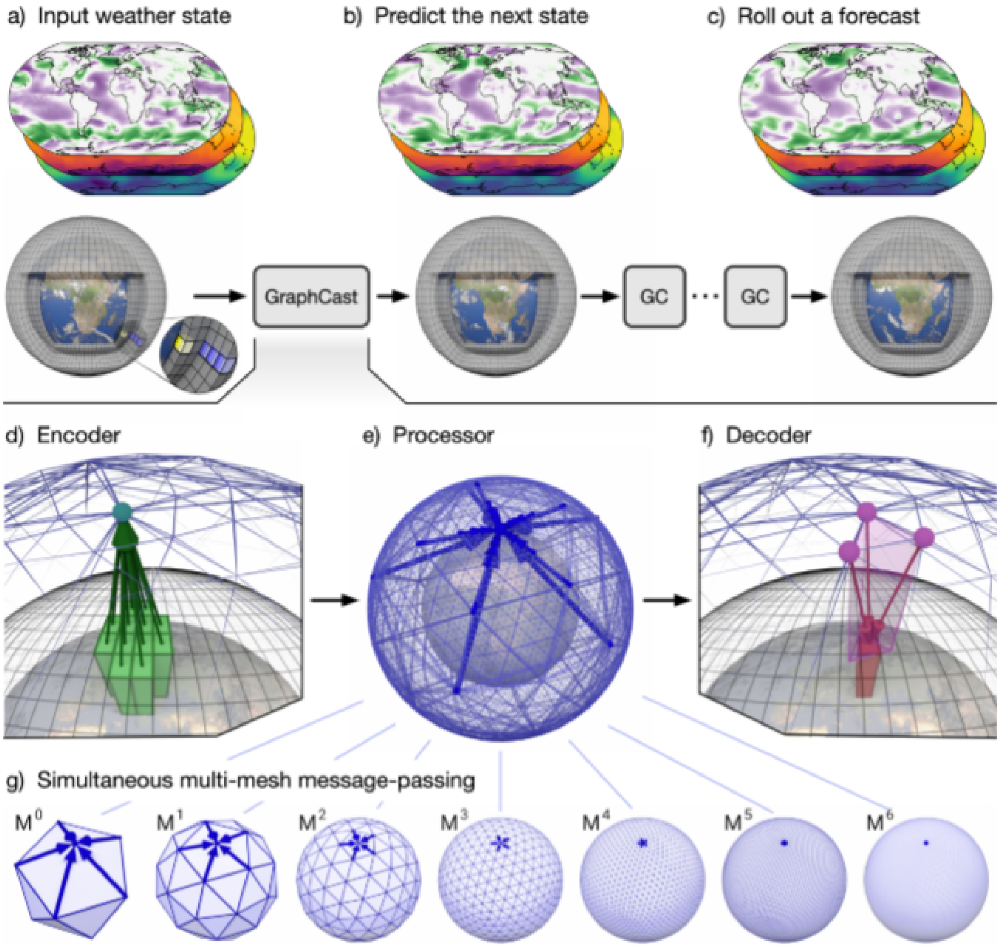


Figure 2.3: Graphcast model schematic. (a) The input weather state(s) are defined on a 0.25° latitude-longitude grid comprising a total of 721 × 1440 = 1, 038, 240 points. Yellow layers in the closeup pop-out window represent the 5 surface variables, and blue layers represent the 6 atmospheric variables that are repeated at 37 pressure levels (5 + 6 × 37 = 227 variables per point in total), resulting in a state representation of 235, 680, 480 values. (b) GraphCast predicts the next state of the weather on the grid. (c) A forecast is made by iteratively applying GraphCast to each previous predicted state, to produce a sequence of states which represent the weather at successive lead times. (d) The Encoder component of the GraphCast architecture maps local regions of the input (green boxes) into nodes of the multi-mesh graph representation (green, upward arrows which terminate in the green-blue node). (e) The Processor component updates each multi-mesh node using learned message-passing (heavy blue arrows that terminate at a node). (f) The Decoder component maps the processed multi-mesh features (purple nodes) back onto the grid representation (red, downward arrows which terminate at a red box). (g) The multi-mesh is derived from icosahedral meshes of increasing resolution, from the base mesh (M^0, 12 nodes) to the finest resolution (M^6, 40, 962 nodes), which has uniform resolution across the globe. It contains the set of nodes from M^6, and all the edges from M^0 to M^6. The learned message-passing over the different meshes' edges happens simultaneously, so that each node is updated by all of its incoming edges. (Figure taken from [22])

GraphCast neural network, after which a residual connection is added. After the information is encoded to the multi-mesh, GraphCast iteratively runs GN updates on the hidden grid. As in equations (2.1) - (2.3), first the edges and then the nodes are updated. To prevent overfitting, a skip connection is added for each update. This layer of the neural network is repeated 16 times in GraphCast, with unshared network weights for each layer. Finally, the latent representation is decoded by applying a single message-passing step on the decoder graph. The decoder graph connects each output grid point to the hidden grid by finding the closest triangle to that grid point and adding edges from each of the corner nodes of that triangle. The edges of the decoder graph are directed towards the output grid, and only these nodes are updated in the final node update. After a final skip connection over these output nodes, the output function is the prediction $\hat{y}_i^G = MLP_{VG}^{Output}(v_i^G)$ for each of the grid nodes v_i^G, where MLP is a multi-layer perceptron that projects the grid attributes to the final 227 predicted variables for that grid node. The prediction represents the difference between two weather states, and thus the next predicted weather state is given by [22]:

$$\hat{X}^{t+1} = GraphCast(X^t, X^{t-1}) = X^t + \hat{Y}^t, \tag{2.7}$$

where X^t represents the weather state representation (ground truth) that is known for time t, $\hat{Y}^t$ is the model output and $\hat{X}^{t+1}$ is the final predicted weather state, given by the sum of the previous weather state X^t and the predicted difference $\hat{Y}^t$.

2.2.2 Training details

The GraphCast network, as described above, is trained using the weighted Mean Squared Error (MSE) loss between the target output and the predicted output over 12-step forecasts of 6 hour lead times (so a total of three days). For longer lead time predictions the forecasts are rolled out as shown in Figure 2.3. The MSE loss takes the average over latitude-longitude, pressure levels, variables, lead times and batch sizes. Weights are applied proportional to the area of a grid cell. Variables are weighted based on the inverse variances of the time differences and based on pressure level, tuned to produce approximately comparable validation performance. For GraphCast, the training data contains ERA5 re-analysis data at a 0.25 degree lat/lon resolution (about 28 square km), ranging from 1979 to 2022. It consists of 6 atmospheric variables at 37 pressure levels, 5 surface variables, 5 static variables and 2 time variables. An overview can be found in Table 2.1.

2.2.3 Evaluation of forecast skill

For deterministic forecasts, the common metrics that are used to evaluate the skill of a forecast are the Root Mean Square Error (RMSE) and the Anomaly Correction Coefficient (ACC). The RMSE relates to the root mean square of the difference between the ground truth and the forecast for a specific lead time τ and variable j. Usually in forecasting, the RMSE is latitude-weighted and is given by:

$$RMSE(j, \tau) = \frac{1}{|D|} \sum_{t_0 \in D} \sqrt{\frac{1}{|G|} \sum_{i \in G} a_i (\hat{x}_{j,i}^{t_0+\tau} - x_{j,i}^{t_0+\tau})^2}, \tag{2.8}$$

where

- D is the test set consisting of the forecast initialization times,
- G contains all of the latitude/longitude coordinates,

- a_i are the areas of the latitude/longitude grid cells (normalized to unit mean),
- $\hat{x}^{t_0+\tau}_{j,i}$ is the prediction of variable j at grid point i and time $t_0 + \tau$,
- $x^{t_0+\tau}_{j,i}$ is the ground truth of variable j at grid point i and time $t_0 + \tau$.

This latitude-weighted RMSE is widely used in geospatial analysis to assess the quality of meteorological variable predictions, although slightly adapted here to follow the convention set by WeatherBench [22]. The ACC is a measure for the evaluation of forecasts in terms of the correlation between the forecasted and observed anomalies, with respect to the long-term averaged climatology. This ACC is typically also latitude-weighted and given as:

$$\mathcal{L}_{ACC}(j,t) = \frac{1}{|D|} \sum_{t_0 \in D} \frac{\sum_{i \in G} a_i (\hat{x}^{t_0+\tau}_{j,i} - C^{t_0+\tau}_{j,i})(x^{t_0+\tau}_{i,j} - C^{t_0+\tau}_{j,i})}{\sqrt{\left[\sum_{i \in G} a_i (\hat{x}^{t_0+\tau}_{j,i} - C^{t_0+\tau}_{j,i})^2\right]\left[\sum_{i \in G} a_i (x^{t_0+\tau}_{i,j} - C^{t_0+\tau}_{j,i})^2\right]}}, \tag{2.9}$$

where $C^{t_0+\tau}_{j,i}$ is the climatological mean of a variable j at spatial location i at time $t_0 + \tau$, with all other components as defined above [22]. The climatological mean is the multi-year average of a weather variable for a given location and time of the year.

2.2.4 Performance

GraphCast was trained on 32 TPUs (Tensor Processing Units) for 4 weeks. It has shown comparable skill to HRES on a number of variables using the normalized RMSE and normalized ACC skill scores as shown on the ECMWF scorecard . The results are summarized in Figure 2.4. Note that GraphCast outperforms HRES for most variables, except for the RMSE of the lowest pressure level (50 hPa). However, similar to other data-driven weather forecasting models, the forecasts seem smoother compared to the HRES forecasts, although this blurring effect did not increase with forecast lead time [7].

2.3 GC-LAM and Hi-LAM

While data-driven models, e.g. [22], show succesful results, the question of the extension of the model to high-resolution forecasts remains. Currently, many institutes around the world use Limited Area Models (LAMs) to provide high-resolution predictions for a specific area at lower computational cost. One such limited area is the MEPS area, which is a rectangular grid centered around Scandinavia, projected using a Lambert conformal conic map projection. [31] designed two models adapting GraphCast to this MEPS area. They adapt the methodology from classic limited area NWP models by using *boundary forcing*. Information on the boundary of the limited area domain is included as an input during the autoregressive rollout of the prediction. After making an initial prediction, the output is replaced by the external boundary information for the grid cells that lay within the boundary area. [31] use ground-truth data, consisting of 3 years of archived forecasts from the operational MEPS system, as the boundary information in their models, but highlight that this could be replaced by an external model (either NWP or data-driven).

The first LAM model (called GC-LAM) adapts the multi-mesh framework from GraphCast, using a connected grid over the regional domain. Their grid is similar to the structure of a CNN, with each node point being connected to nine gridpoints. Unfortunately, the results show circular artefacts around the nodes, as can be seen in Figure 2.5, especially for the nodes with

Table 2.1: ECMWF variables used as input/prediction for Graphcast. The first column indicates whether the variable represents a static property, a time-varying single-level property (e.g., surface variables are included), or a time-varying atmospheric property. The second and third columns are ECMWF's labels. The fourth column is a ECMWF's numeric label, and can be used to construct the URL for ECMWF's description of the variable, by appending it as suffix to the following prefix, replacing "ID" with the numeric code: `https://apps.ecmwf.int/codes/grib/param-db/?id=ID`. The last column indicates whether the variable is taken as input and is predicted, or is only used as input context (the double horizontal line separates predicted from input-only variables, to make the partitioning more visible). Table taken from [22].

Type	Variable name	Short name	ECMWF Parameter ID	Role (accumulation period, if applicable)
Atmospheric	Geopotential	z	129	Input/Predicted
Atmospheric	Specific humidity	q	133	Input/Predicted
Atmospheric	Temperature	t	130	Input/Predicted
Atmospheric	U component of wind	u	131	Input/Predicted
Atmospheric	V component of wind	v	132	Input/Predicted
Atmospheric	Vertical velocity	w	135	Input/Predicted
Single	2 metre temperature	2t	167	Input/Predicted
Single	10 metre u wind component	10u	165	Input/Predicted
Single	10 metre v wind component	10v	166	Input/Predicted
Single	Mean sea level pressure	msl	151	Input/Predicted
Single	Total precipitation	tp	228	Input/Predicted (6h)
Single	TOA incident solar radiation	tisr	212	Input (1h)
Static	Geopotential at surface	z	129	Input
Static	Land-sea mask	lsm	172	Input
Static	Latitude	n/a	n/a	Input
Static	Longitude	n/a	n/a	Input
Clock	Local time of day	n/a	n/a	Input
Clock	Elapsed year progress	n/a	n/a	Input

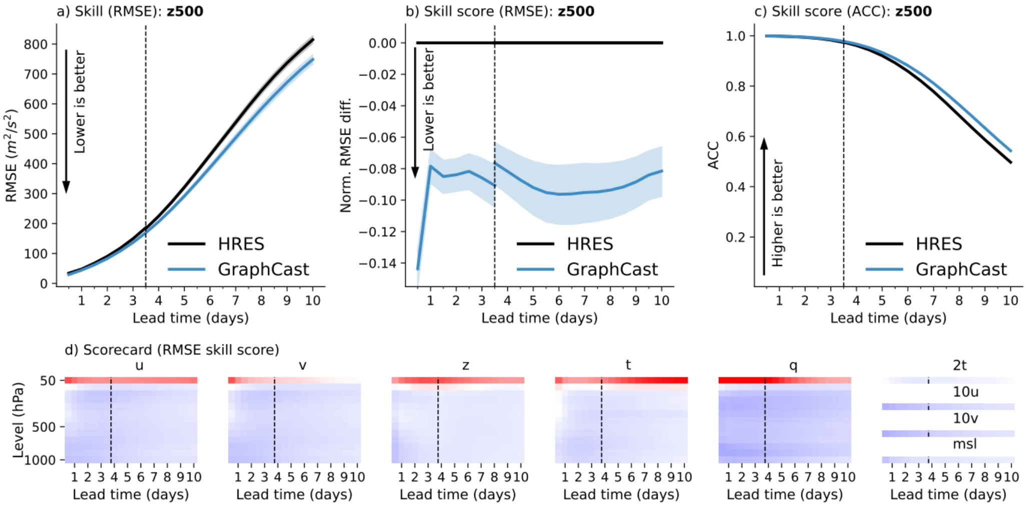


Figure 2.4: Skill and skill scores for GraphCast and HRES in 2018. (a) RMSE skill (y-axis) for GraphCast (blue lines) and HRES (black lines), on z500, as a function of lead time (x-axis). Error bars represent 95confidence intervals. The vertical dashed line represents 3.5 days, which is the last 12 hour increment of the HRES 06z/18z forecasts. The black line represents HRES, where lead times earlier and later than 3.5 days are from the 06z/18z and 00z/12z initializations, respectively. (b) RMSE skill score (y-axis) for GraphCast versus HRES, on z500, as a function of lead time (x-axis). Error bars represent 95%-confidence intervals for the skill score. We observe a discontinuity in GraphCast's curve because skill scores up to 3.5 days are computed between GraphCast (initialized at 06z/18z) and HRES's 06z/18z initialization, while after 3.5 days skill scores are computed with respect to HRES's 00z/12z initializations. (c) ACC skill (y-axis) for GraphCast (blue lines) and HRES (black lines), on z500, as a function of lead time (x-axis). (d) Scorecard of RMSE skill scores for GraphCast, with respect to HRES. Each subplot corresponds to one variable: u, v, z, t, q, 2t, 10u, 10v, msl, respectively. The rows of each heatmap correspond to the 13 pressure levels (for the atmospheric variables), from 50 hPa at the top to 1000 hPa at the bottom. The columns of each heatmap correspond to the 20 lead times at 12 hour intervals, from 12 hours on the left to 10 days on the right. Each cell's color represents the skill score, as shown in (b), where blue represents negative values (GraphCast has better skill) and red represents positive values (HRES has better skill). Figure taken from [22].

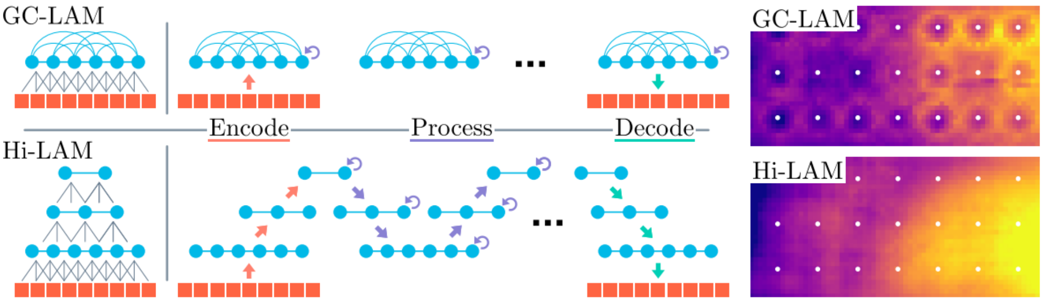


Figure 2.5: Left: Overview of the prediction process of the GC-LAM and Hi-LAM models. Input at the grid nodes (orange squares) are encoded to the mesh nodes (blue circles), processed and decoded back to produce a one-step prediction. Right: Artefacts in GC-LAM prediction (enlarged) centered at mesh nodes with > 8 neighbors (white dots). Figure taken from [31]

.

a large number of connections. As a solution, they propose a second model (Hi-LAM) that uses a *hierarchical* grid that connects the different mesh-levels by adding extra encoding GN layers. This ensures that the higher levels capture the latent representation of the entire domain more effectively (see Figure 2.5). During prediction, information from all mesh levels are incorporated in the final output [31].

2.3.1 Training details

All models are trained using the weighted MSE as loss function, similar to GraphCast. For the GC-LAM model, 4 GN processing layers are used. For the Hi-LAM model only half the amount of processing layers is needed, as each layer is updated twice. The models were trained on a single NVIDIA A100 GPU for about 3-4 days, using a training data set of only two years. The data contains 17 weather variables at a forecasting step time of 3 hours. The horizontal resolution was downsampled from the original 2.5km to 10km.

2.3.2 Performance

The authors compare the performance of both models to a simple model called 1L-LAM, which uses a fully connected graph without any hidden layers or hierarchical structure. Since the models are not compared to any operational forecasts or NWP models on this domain at this resolution, it is hard to evaluate the skill of the forecasts. Qualitative results indicate that the Hi-LAM model predictions are close to the ground truth, although being slightly blurred for some variables. Overall, these results are promising, especially since they show that with using fewer computational resources (less GPUs, fewer training data), we can still expect reasonable results on a high-resolution limited domain.

Chapter 3

AIFS - ECMWF's Data-Driven Forecasting System

The European Centre for Medium Range Weather Forecasting (ECMWF) is an independent intergovernmental meteorological institute. ECMWF produces high-quality global numerical weather predictions for 35 member states, as well as for commercial customers. Following the recent advances in DLWP as described in the previous chapter, ECMWF has developed their own improved medium-range global data-driven forecasting model, called AIFS (Artificial Intelligence/Integrated Forecasting System) [23] [24]. To implement the model in an operational setting, they not only focus on the machine learning framework, but also on developing end-to-end training and inference data pipelines, along with ensuring operational verification. The model is based on the structure of GraphCast, consisting of a GNN-based encoder-processor-decoder structure with 16 hidden processor layers.

3.1 Data and grid structure

Unlike GraphCast, AIFS avoids using a latitude-longitude data grid due to the over-representation of gridpoints at the poles. Instead, spatial data is represented on a reduced Gaussian grid (N320), widely acknowledged by the meteorological community for the reduction in data size and convenience in applying Fast Fourier Transforms [21]. The N320 grid has $N = 320$ equally-spaced latitude lines between the poles and the equator, resulting in a total of $2N = 640$ latitude lines. The points on the latitude lines are calculated from the zeros of the Legendre polynomial of order $2N$:

$$P_{2N}(sin(\theta_k)) = 0, \tag{3.1}$$

where θ_k is the number of points on latitude line k [11]. The solutions of the Legendre polynomial provide (near)-regular distance intervals in both latitudinal and longitudinal directions, resulting in a better spatial distribution across the globe [21]. The GNN's flexibility in handling irregular data structures allows for the smooth implementation of these non-uniform grids. Furthermore, the data-loading pipeline is made more efficient by using the Zarr storage format, allowing arrays to be chunked and loaded more conveniently. Incorporating this data format using the ECML-tools dataloader framework in Pytorch-Lightning results in a significant speedup [24].

The model is trained on six key atmospheric variables at 13 pressure levels and a number of surface variables, following conventions used in other models. To reduce computational costs,

Table 3.1: AIFS training data. The table provides an overview of AIFS inputs and outputs during training.

Field	Level Type	Input/Output
Geopotential, U component of wind, V component of wind, vertical velocity, specific humidity, temperature	Pressure level (hPa): 50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, 1000	Both
Surface pressure, mean sea-level pressure, sea-surface temperature, skin temperature, 2m temperature, 2m dewpoint temperature, 10m U component of wind, 10m V component of wind, total column water	Surface	Both
Land-sea mask, orography, standard deviation of sub-grid orography, slope of sub-gridscale orography, insolation, latitude/longitude, time of day/day of year	Surface	Input

less pressure levels are used (13 instead of the 37 pressure levels of GraphCast), resulting in fewer variables for training/prediction. In addition, convective precipitation and total precipitation are modeled as output variables only. The variables used in the AIFS training are given in Table 3.1.

3.2 Graph-Transformer

As of February 2024, AIFS is able to run at a spatial resolution of 0.25 degrees. This is due to the introduction of attention mechanisms in both the encoder/decoder as well as the processor. In the encoder and decoder, following the ideas of [36], the message aggregation step for the edge updates is adjusted by transforming each source node attribute h_i to a query vector $q_i = W_q h_i + b_q$ and each target node h_j to a key vector $k_j = W_k h_j + b_k$. The edge attributes are encoded in the bias vector $\hat{e}_{ij} = W_e e_{ij} + b_e$, and the value vector v_i is again computed from the source node using the attention operation:

$$\alpha_{ij} = \frac{e^{q_i^T (k_j + \hat{e}_{ij}))/\sqrt{}(d_k)}}{\sum_{u \in \mathcal{N}(i)} e^{q_i^T (k_u + \hat{e}_{iu}/\sqrt{}(d_k)}}, \tag{3.2}$$

where d_k is the dimension of the key vector. The softmax operation is performed over the neighbourhood $\mathcal{N}(i)$ of the source node i. Thus, the attention operation represents the importance of an edge by giving the edge probability distribution per source node. The new node features are computed from a value vector $v_u = W_v h_u + b_v$ weighted by the (permutationally invariant) aggregated attention values [36]:

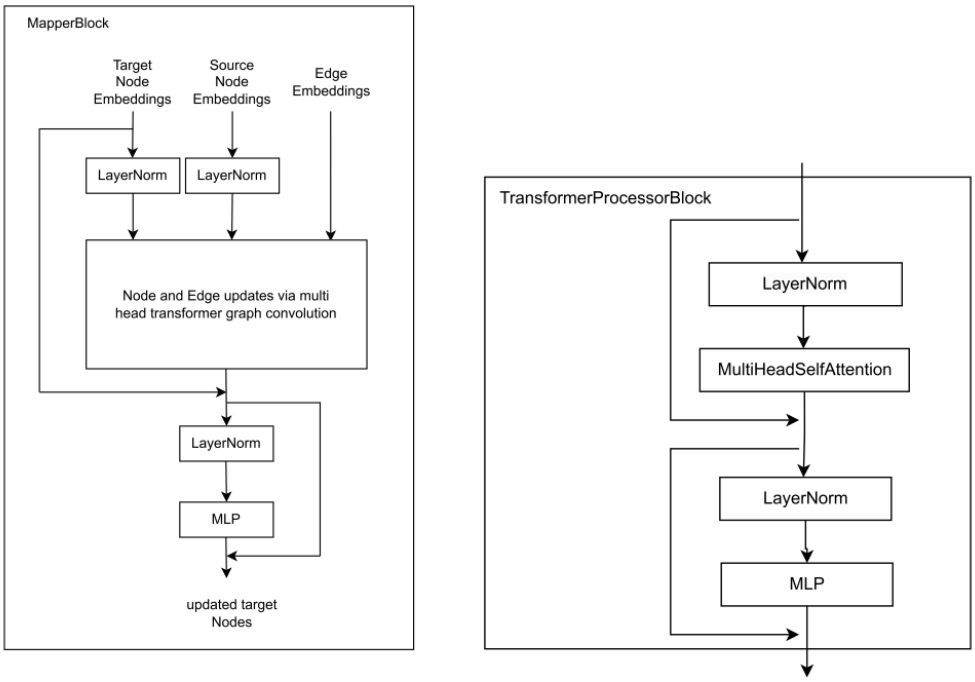


Figure 3.1: AIFS encoder / decoder and processor block schematics: GNN block (left), processor block (right). The GNN block uses a multi-head graph transformer convolution operation to update the nodes and the edges of the processor, whereas the pre-norm transformer block relies on multi-head self-attention. Figure taken from [24].

.

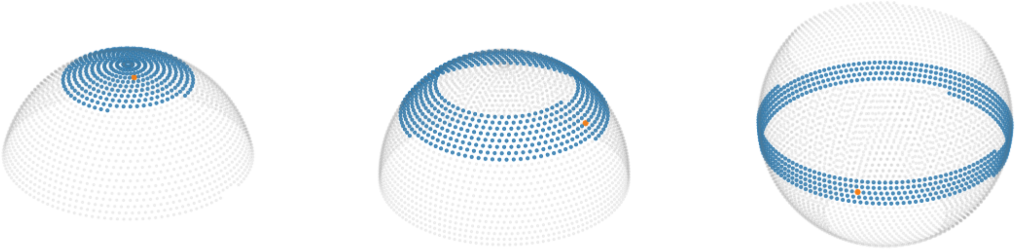


Figure 3.2: Shifted window attention windows (in blue) for different grid points (in red) in the AIFS processor. The range of information transfer within 6 processor layers is given in grey. For visualization purposes, the grid is shown at a lower resolution. Figure taken from [24].

$$\hat{h}_i = \sum_{u \in \mathcal{N}(i)} \alpha_{iu}(v_u + \hat{e}_i u). \tag{3.3}$$

This process is repeated C times for the encoder and decoder to accommodate multi-head attention. For the decoder, the multi-head output is averaged. The source and target nodes are normalized using a LayerNorm, similarly for the output of the transformer graph convolution. Finally, a MLP layer is added to form the GraphTransformer block, visualized in Figure 3.1.

The model eliminates the need for processor edges by adopting pre-norm transformer layers with one-dimensional shifted window attention. Attention is computed along latitude bands of a coarser resolution octahedral Gaussian reduced grid (O96, approximately 1 degree spatial resolution). A visualization of the shifted window attention process can be found in Figure 6.3. Similar to GraphCast, sixteen processor layers facilitate the communication of information across the globe. Additionally, eight learnable features are added to compensate for possible missing fields [24].

3.3 Training

AIFS adheres to the training as mentioned in [24] by pretraining on 6 hour time steps and finetuning on larger training steps up to 72 hours (called *rollout steps*). This finetuning is performed by re-initializing the model from the output prediction in an auto-regressive manner. The model is finetuned using the loss calculated over multiple rollout steps (*rollout training*). The predicted values used as input are the *prognostic* variables, the excluded variables (total precipitation and convective precipitation) are called *diagnostic* variables. The *forcing* variables, used only as inputs, are appended to the prognostic variables to provide the complete future state initialization. The variables are normalized using mean-max normalization, transforming the data distribution to have a mean of zero and unit variance. Exceptions are some of the forcing variables: *sdor*, *slor* and z, which are max-normalized, while the remaining forcing variables (see Table 4.4) are left unnormalized.

The latitude-weighted MSE loss function described in Chapter 2 is enhanced by adding variable weighting and pressure-level weighting to balance the data points in the loss function. Variable weighting is applied in multi-variable optimization to account for the relative importance of each prognostic variable. These weightings are empirically tuned hyperparameters, with the optimal values provided in Table 3.2. The diagnostic variables are assigned lower weightings, as they should be induced from the prognostic variables to ultimately be evaluated on the test set. Upper-atmosphere variables are weighted less than lower-atmosphere variables, reflecting the meteorological preference to have improved predictions at high pressure levels. This is called pressure level scaling, AIFS does so linearly starting using a weight of $0.001 * pressure$.

The learning rate is controlled by using a cosine learning rate scheduler with warm restarts. The learning rate is increased during a *warmup period* to ensure proper initialization of the model. After reaching the maximum value η_0, the learning rate is decreased as follows:

$$\eta_t = \eta_T + \frac{\eta_0 - \eta_T}{2}(1 + \cos(\pi t/T)), \tag{3.4}$$

where η_t is the learning rate at epoch $t \in [0, T]$[27]. In AIFS, the initial (maximum) learning rate η_0 is set to 0.625×10^{-4}, the minimum (final) learning rate η_T is set to 3×10^{-7} and the total number of iterations is set to 300000. AIFS is trained on 64 Nvidia A100 GPUs for a total duration of one week.

Table 3.2: Variable weighting in the latitude-weighted MSE loss function used by AIFS [22]. The weightings are hyperparameters tuned on the validation loss. The variables not mentioned in this table are set to a default of one. For a list of variable abbreviations see Table 4.1, 4.2, 4.3 and 4.4.

variable name	weighting	variable name	weighting
q	0.6	sp	10
t	6	10u	0.1
u	0.8	10v	0.1
v	0.5	2d	0.5
w	0.001	tp	0.025
z	12	cp	0.0025

3.4 Results

[24] compared the AIFS to the IFS by evaluating both on the test set, using the operational ECMWF analyses - from which the models are initialized - as ground truth. For each variable and pressure level, the ACC, RMSE and forecast activity are given. The forecast activity is a measure for the smoothness of a forecast, where lower forecast activity indicates a more blurred forecast. The mathematical definition of forecast activity of variable j at lead time τ is given as follows:

$$SDAF(j,t) = \sqrt{\overline{((x_{j,i}^{t_0+\tau} - C_{j,i}^{t_0+\tau}) - \overline{(x_{j,i}^{t_0+\tau} - C_{j,i}^{t_0+\tau})^2}}}. \tag{3.5}$$

where $x_{j,i}^{t_0+\tau}$ is the forecast prediction of variable j at grid point i and time $t_0 + \tau$, $C_{j,i}^{t_0+\tau}$ is the climatological mean of variable j at grid point i and time $t_0 + \tau$ and the bar indicates averaging the variable over grid point $i \in D$ with latitude weighting as in the RMSE and ACC described in Chapter 2.

The results of [24] are summarized in a scorecard given in Figure 3.4. AIFS significantly outperforms IFS for higher pressure levels on ccaf/SEEPS (precipitation skill scores) and RMSE. Lower pressure levels show worse performance, as also observed by Lam et al. [22]. This is likely influenced by the introduction of the pressure level scaler. It is noticable how AIFS gains advantage over IFS for longer lead times, whereas IFS outperforms AIFS for the first day.

3.5 Stretched grid model

This research was done in collaboration with MET Norway, who developed a stretched grid AIFS model (SG-AIFS) over the Nordics. The model is trained on ERA5 and a reanalysis of the Met-CoOp Ensemble Prediction System (MEPS), their operational HARMONIE-AROME NWP model.

The MEPS dataset spans 3 years (2020-2023), thus a transfer learning framework is introduced. In transfer learning, the limited area model utilizes the knowledge acquired from previously trained models by initializing its weights. This approach not only addresses the challenge of varying data availability, but also reduces the overall training time. Moreover, transfer learning improves performance by allowing a smoother transition to intermediate downscaling steps. However, transfer learning comes with two challenges: catastrophic forgetting and the prevalence of trainable parameters. Catastrophic forgetting occurs when parameters are overwritten during finetuning. [29] mitigates this effect by adjusting the learning rate during finetuning, as well as decreasing the number of training steps. The second challenge is the presence of trainable parameters. The problem particularly arises during the transferal of GNN parameters, when the graph structure is different across the models. Since the trainable parameters are inherently tied to the graph structure, omission is necessary during both pretraining and finetuning. [29] suggests this omission to have minimal influence on model performance.

Four finetuning steps are performed, increasing resolution at the first three steps (see Figure 3.3). First, the model is pre-trained on 1 degree ERA5 reanalysis data, after which a refinement step to 0.25 degree ERA5 data is performed. Subsequently, the ERA5 0.25 degree dataset is integrated with a 2.5 km regional MEPS dataset with a training period of 3.3 years. Finally, training is performed for four prediction time steps of rollout. [29] demonstrated better performance than MEPS when evaluating on surface observations, particularly for temperature and wind speed, although their model showed an underestimation of extreme values.

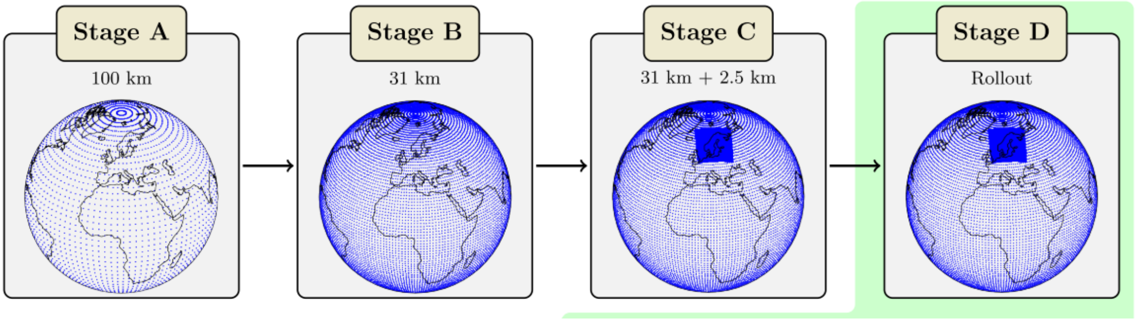


Figure 3.3: Stretched grid model training by Nipen et al. [29] follows a four-stage procedure. First, the DDM is pre-trained on 43 years of ERA5 data with a global resolution of 100 km (stage A) and 31 km (stage B). In stage C, the 31 km global IFS dataset is combined with the 2.5 km regional MEPS dataset with a training period of 3.3 years. Finally, the model is fine-tuned by auto-regressive rollout training over four prediction time steps (stage D). Figure taken from [29].

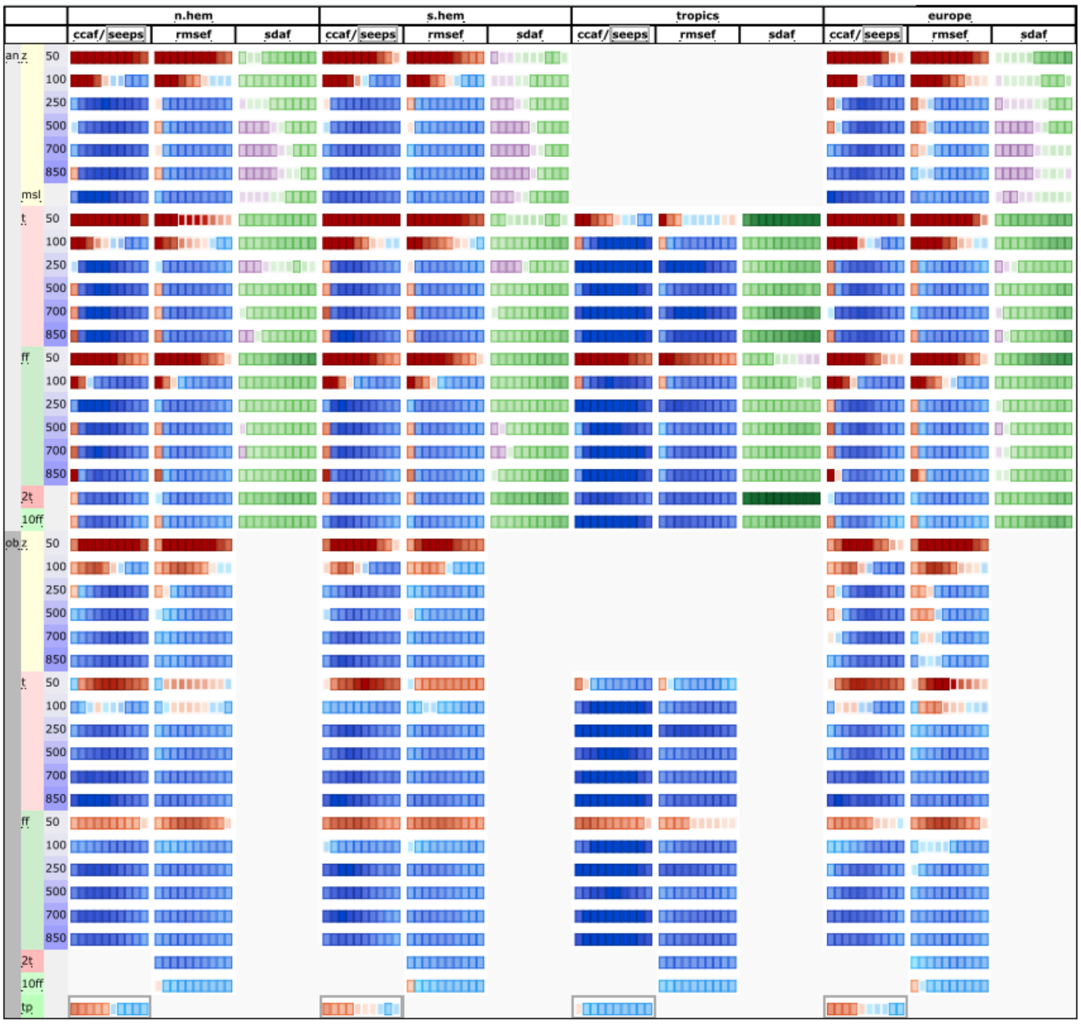


Figure 3.4: Scorecard comparing forecast scores of AIFS versus IFS (for 2022). Forecasts are initialised on 00 and 12 UTC. Shown are relative score changes as function of lead time (day 1 to 10) for northern extra-tropics (n.hem), southern extra-tropics (s.hem), tropics and Europe. Blue colours mark score improvements and red colours score degradations. Purple colours indicate an increased in standard deviation of forecast anomaly, while green colours indicate a reduction. Framed rectangles indicate 95% significance level. Variables are geopotential (z), temperature (t), wind speed (ff), mean sea level pressure (msl), 2 m temperature (2t), 10 m wind speed (10ff) and 24 hr total precipitation (tp). Numbers behind variable abbreviations indicate variables on pressure levels (e.g., 500 hPa), and suffix indicates verification against IFS NWP analyses (an) or radiosonde and SYNOP observations (ob). Scores shown are anomaly correlation (ccaf), SEEPS (seeps, for precipitation), RMSE (rmsef) and standard deviation of forecast anomaly (sdaf, see text for more explanation). Figure taken from [23].

Chapter 4

Data description and preprocessing

Over the past 50 years, an increasing amount of meteorological data can be gathered from satellites and ground observations, among other observation types. In data assimilation, NWP model predictions are combined with observations in an optimal manner in order to provide a regular and accurate representation of the state of the atmosphere in both space and time [3] [20]. In this research project, reanalysis and reforecast products are combined: ERA5 data and DOWA data, respectively.

4.1 ERA5

ERA5 is the 5th generation of reanalysis data provided by ECMWF. The model used is the Integrated Forecasting System (IFS) Cycle 41r2, from which reanalyses were made starting from 1979. The assimilation is done using the novel 4D-Var data assimilation, optimizing the meteorological representation of the initial state in the three dimensions of space as well as optimizing over a time window. Moreover, the reanalysis couples land, sea and atmospheric modeling to form a complete representation. [3] [20]. In this project, the ERA5 data and the variables used are equivalent to the AIFS model variables (see Table 4.1, 4.2 and 4.3). Further details are described in Section 3.

4.2 DOWA

The DOWA dataset is a reforecast dataset, consisting of 11 years of ERA5-HARMONIE forecasts (2008-2017). This reforecast was saved on the domain as depicted in Figure 4.1, which has 789 by 789 grid points corresponding to a spatial resolution of about 2.5 kilometers. It is saved in daily files with 3-hourly forecasts, where the forecast is initialized every 3 hours. DOWA is saved on the ECFS file system, a High-Performance Storage System (HPSS of ECMWF). Files are saved on tapes and have to be explicitly retrieved onto a local server. With a 1-hour temporal resolution and a spatial resolution of 2.5 km on 65 pressure levels, the retrieved dataset has a total size of about 50TB. Due to the storage space policy of ATOS, files are removed after 30 days. The initial training set consisted of the years 2008-2009 and 2013-2015. The year 2016 was chosen as evaluation year and 2017 as testing year. After the adjustments described below are processed, the data size is reduced to 0.493TB.

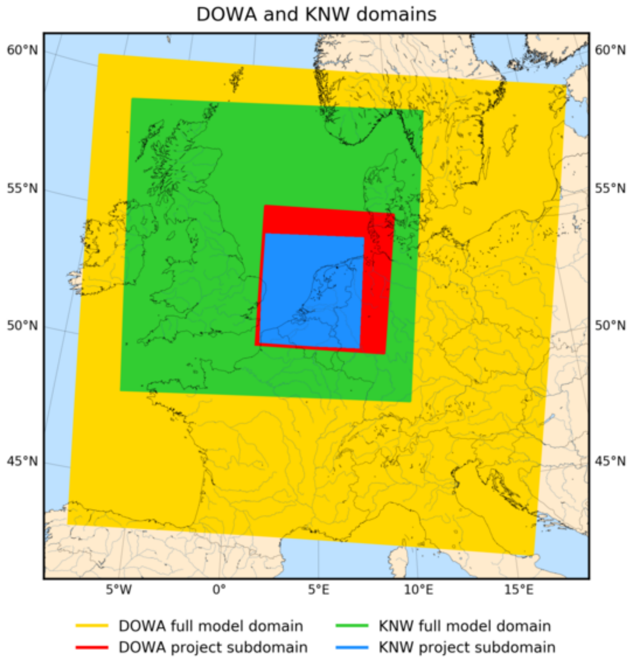


Figure 4.1: ERA5-HARMONIE domain (yellow) of 789x789 points and DOWA-subdomain of 217x234 points (red). ERA-Interim-HARMONIE domain of 500x500 points (green) and KNW-subdomain of 170x188 points (blue). Figure taken from [39].

4.2.1 Preprocessing

Although the DOWA dataset contains most of the ERA5 re-analysis variables used by AIFS, many adjustments had to be made to combine the DOWA and ERA5 datasets.

Firstly, some surface variables are missing in DOWA. The dewpoint temperature at 2 meter height, the total column water, and the convective precipitation are not present in DOWA. The dewpoint temperature at 2 meter height d_{2m} can be calculated from the relative humidity at 2 meter height RH_{2m} and the temperature at 2 meter height T_{2m}, but the relative humidity at 2m height is not present in the dataset. Thus, the relative humidity is computed from the specific humidity using the following equations [25]:

$$RH_{2m} = 100\frac{w}{w_s}, \tag{4.1}$$

where $w_s = 0.622 * \frac{e_s}{P_{2m}-e_s}$ and $w \approx q_2 m$. Here, q_{2m} is the specific humidity at 2m height, P_{2m} is the pressure at 2m height and e_s is the saturation vapor pressure, defined as:

$$e_s = 610.94 \times e^{17.67\frac{T_{2m}-273.15}{T_{2m}-29.65}}. \tag{4.2}$$

Then, d_{2m} can be calculated using T_{2m} as follows [25]:

$$d_{2m} = T_{2m} - ((100 - RH_{2m})/5). \tag{4.3}$$

The total column water and convective precipitation could not be computed from the available variables and are consequently omitted from both datasets. As a consequence, the dataset contains insufficient hydrological variables to evaluate precipitation results (cp, tp in AIFS) in a meaningful way. Incorporating proper precipitation data is left to future research.

Secondly, the atmospheric variables in DOWA are saved on model levels, instead of pressure levels. Model levels follow the shape of the terrain for lower pressure levels, and consequently become closer to pressure levels higher in the atmosphere. The pressure between two levels is constant, and defined in so-called half-levels, depending on the pressure at surface level ps [15]:

$$p_{k+1/2}(x,y) = A_{k+1/2}(x,y) + B_{k+1/2}(x,y) * ps(x,y) \qquad \forall(x,y) \in R, \tag{4.4}$$

where R is the two-dimensional region where DOWA is defined. The coefficients $A_{k+1/2}$ and $B_{k+1/2}$ are fixed, such that the pressure at the lowest level (65) is equivalent to the surface pressure, and the pressure at the highest atmospheric level is zero. DOWA is saved on HARMONIE-AROME's 65 pressure levels. Thus, to convert from model levels to the pressure levels defined by ERA5, first the pressure is calculated at each spatial point from A and B, then each spatial point is interpolated vertically to the 13 desired pressure levels (Table 3.1). Since this is computationally expensive in Python, we use the Python package `numba` to speed up computation. A challenge with this interpolation strategy is the orography of the domain. High pressure levels (such as 1000 hPa) are not present over mountainous regions, such as the Alps. In ERA5 and MEPS, this is addressed by extrapolating to these regions while completely disregarding the orography. However, high-quality extrapolation is computationally expensive and adds little value to the outcome of the model. Therefore, we decide to set these values to `NaN`. During training, these `NaN`s are imputed with the mean or minimum of each corresponding variable, depending on the normalization. As a consequence, after normalization the missing values will be zero and the model is able to identify these regions effortlessly, eliminating any impact on training. After training, the imputation is reversed to provide the proper predictions.

Thirdly, DOWA is re-gridded using a Lambert conformal conic projection instead of a latitude/longitude grid. Fortunately, the structure of GNNs is very adaptable to grid types, meaning that no spatial interpolation of DOWA is necessary. However, the u and v components of the wind are provided relative to the grid and therefore require re-projection of the wind directions relative to the latitude/longitude grid. We adopt the code provided by the Norwegian Meteorological Institute (MET Norway) using the package `pyproj` and taking into account the change in windspeed that occurs due to differently sized gridboxes. Moreover, since the vertical wind velocity w is given in a hydrostatic manner (in meters per second), it should be converted to the pressure velocity ω in Pascal per second as in ERA5. This can be done using the temperature T in Kelvin and the pressure p in Pascal, according to the following formula:

$$\omega = -wgp/(TR_d), \tag{4.5}$$

where g is the gravitational acceleration and R_d is the specific gas constant.

Lastly, the forcing variables (Table 4.4) are calculated and the required statistical attributes (sums, squares, minimum and maximum) are added. We selected 6-hourly time steps to match the ERA5 temporal resolution (meaning the dataset fully consists of +3h reforecasts) and the resulting dataset is added to a Zarr archive matching the metadata and structure of ERA5. The code was based on the Zarr converting package provided by MET Norway. An overview of the comparison of ERA5 and DOWA variable names can be seen in Tables 4.1, 4.2, 4.3 and 4.4. Note that abbreviations for the statistical attributes are not given, since these are calculated from the data for fast accessibility during training. Due to long pre-processing times, 7 of the 10 available years were converted when starting the high-resolution training.

Table 4.1: Atmospheric variables.

Variable name	**AIFS/ERA5 code**	**DOWA code**
Geopotential	z	phi
U component of the wind	u	ua
V component of the wind	v	va
Vertical velocity	w	w
Specific humidity	q	hus
Temperature	t	ta

Table 4.2: Surface variables.

Variable name	**AIFS/ERA5 code**	**DOWA code**
Surface pressure	sp	ps
Mean sea-level pressure	msl	psl
Sea surface temperature	sst	sst
10m U component of wind	10u	uas
10m V component of wind	10v	vas
2-meter temperature	2t	tas
Skin temperature	skt	ts
2-meter dewpoint temperature	2d	-
Total column water	tcw	-

Table 4.3: Diagnostic variables.

Variable name	**AIFS/ERA5 code**	**DOWA code**
Total precipitation	tp	prrain
Convective precipitation	cp	-

Table 4.4: Forcing variables.

Variable name	**AIFS/ERA5 code**	**DOWA code**
Land-sea mask	lsm	sftof
Orography	z	Orog
STD of Orography	sdor	-
Slope of Orography	slor	-
Insolation	Insolation	Insolation
Sine of latitude	sin_latitude	sin_latitude
Sine of longitude	sin_longitude	sin_longitude
Cosine of the Julian day	cos_julian_day	cos_julian_day
Sine of the Julian day	sin_julian_day	sin_julian_day
Sine of the local time	sin_local_time	sin_local_time
Cosine of the local time	cos_local_time	cos_local_time

Chapter 5

Methodology

While advances in global medium-range weather prediction are numerous and promising, further improvements are needed in the field of high-resolution weather prediction. This MSc thesis seeks to build on recent developments in collaboration with MET Norway [29], aiming to effectively apply the stretched grid approach to increase resolution for regional forecasts. Fully utilizing the data driven model developed by [29] is not feasible, due to the memory and computational requirements. Nevertheless, it shows that transfer learning has the potential to optimize data utilization whilst accelerating individual model training time. In this section, we describe the stretched grid (SG-AIFS) experiment setup and explain how we expand on the current developments by exploring the properties of this model.

A stretched hidden grid is implemented within the AIFS framework, enabling the use of additional regional high-resolution data. Initially, the vanilla GNN version of AIFS is considered, allowing us to directly compare the performance of the GNN to the Transformer. The hidden mesh structure of the GNN follows an iteratively refined icosahedron similar to GraphCast, but the triangles overlapping with the area of interest are refined multiple times to keep the number of neighbours constant for the finest level of the multi-mesh. This results in a more gradual boundary and incorporates global data directly during training. Two configurations arise to combine the different data sources: concatenating overlapping global and regional data points ("concatenation") or excluding global data points within the limited area domain ("cutout"). The latter "cutout" approach was elected to avoid data continuity issues that could arise from combining inherently different reanalysis and reforecast datasets on the same domain. Additionally, deviating from the implementation of the stretched grid by [29], we iterate over the regional data points, refining any triangles that contain such a data point. On the other hand, [29] concatenate two graphs by masking out nodes and edges within the region of interest, resulting in minor connectivity differences. Additionally, the available preprocessed DOWA data spans seven years as opposed to the three years of MEPS reanalysis. DOWA's extended temporal range could positively impact performance on more recent years (as noted by [22]).

5.1 ERA5 experiments

Due to large training and data preprocessing times associated with the DOWA dataset, a lower resolution stretched grid framework was tested to experiment with the configuration of the model. For the initial experiments, ERA5 data are used on a global scale as well as on a regional scale, albeit at different resolutions. The N320 octahedral grid at about 0.25 degree resolution is extracted over the regional domain. On the remainder of the globe, o96 Gaussian grid data are provided at about 1 degree resolution. The training data range from 1979 to 2020,

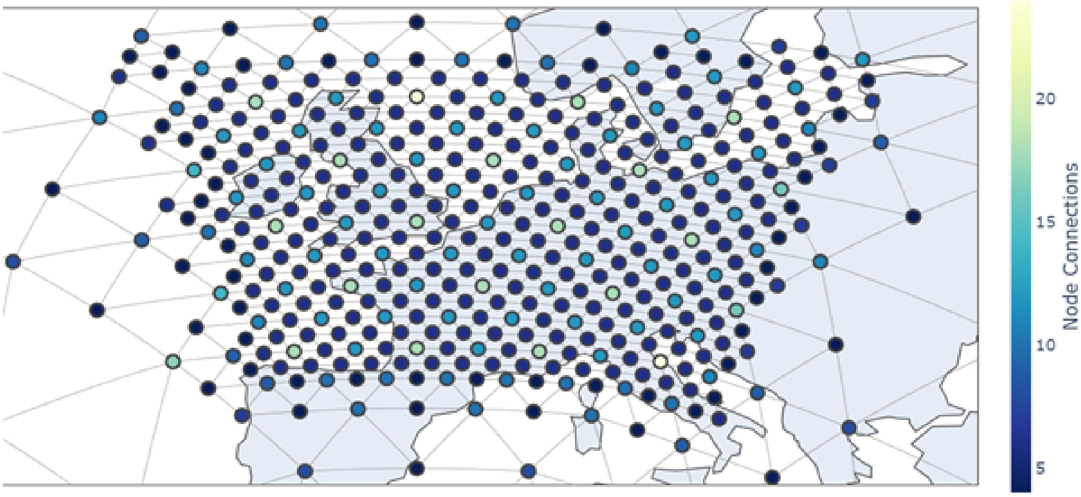


Figure 5.1: Visualization of the stretched hidden grid, with the refined area corresponding to the HARMONIE-AROME/DOWA full model domain. Each of the triangles of the hidden grid are split into four triangles and projected to the unit sphere.

with 2021 as validation set and 2022 as testing set. Training is done using an Adam optimizer on a single NVIDIA A200 GPU for 150 epochs on the European Weather Cloud (EWC). The batch size is set to 2, with 4000 batches per epoch. For the validation and testing, a batch size of 4 is used with a maximum number of batches of 700. The number of channels is set to 256, and the learning rate is scaled in each epoch, starting from 0.00000619 and increasing to 0.0000624 over the course of 10 epochs before decreasing to $3 * 10^{-7}$ according to the cosine learning rate scheduler. Incorporating the high-resolution data requires alteration of the latitude-weighted loss function. Typically, latitude-based weighting is applied, assuming an equal distribution of points across each latitude band [17]. To compensate for the imbalance introduced by additional high-resolution data, we combined a latitude weighted global loss with an unweighted high-resolution loss. Each high-resolution data point is weighted equally to adhere to the total weighting of the global area summing to equal the amount of data points. As a result, for the ERA5 experiments all grid boxes in the DOWA domain are assumed to have approximately the same size, amounting to a total weight contribution of the high-resolution data of about 10% to the total loss.

We assume that the ERA5 experiment results can be consequently transferred to a higher-resolution model. Three of the research questions posed in Chapter 1 are addressed, namely 1) the influence of increasing resolution in the processor hidden grid 2) the performance of the Transformer model and 3) the influence of including the rollout step described in [22], [23] and [29]. Concerning the first research question (Experiment 1), the adaptation of the hidden grid structure will be analyzed, specifically the difference in resolution of the processor grid. Two stretched grids are developed: one with a global refinement level of 4 and a local refinement level of 6, later referenced as having resolution 4 (about 4 longitudinal degrees). The second grid has a global refinement of 5 and a local refinement of 7 (referred to as resolution 5, about 2 longitudinal degrees). To test if the adaptation of the grid changed the training results, we connect the hidden grid to the ERA5 data and run it with the same parameters as for AIFS. The models are trained to optimize on 6 hour lead time, initially excluding rollout training.

Addressing the second research question (Experiment 2), the GNN processor will be compared with the Transformer processor. The current AIFS model has deviated from using a GNN in part of the processing layers, opting instead for using attention mechanisms. This development is based on promising results in the literature, where Transformer models have been applied successfully to global weather data (see Appendix A for background on applying Transformers as DLWP models). This has the potential to increase model speed and eliminate the need for predetermined edges. We experiment with implementing this adaptation for a limited area model. Since AIFS implements the Transformer in a one dimensional sliding window attention operation, complications arise when adapting this process to a combined hidden mesh. The decision is made for a simple data appendment, resulting in a reasonable model considering the large window size. Since the Transformer processor should be on a Gaussian grid (following Lang et al [24])], resolution N32 is used on the global scale, whereas on a local scale resolution N80 is implemented. However, for future implementation we anticipate that a two dimensional attention mechanism will be more suitable. Our model follows the approach of AIFS in using a Gaussian grid for the global ERA5 data. A visualization of this grid can be seen in Figure 5.1. The Gaussian grid is refined locally by increasing the hidden grid resolution twice. Note that the multi-mesh allows nodes with local connectivity and global connectivity to be mixed in order to provide a better latent representation of the state of the atmosphere. Lastly, for the third research question (Experiment 3), the impact of rollout training on the limited area is investigated. We compare the results of the rollout model to the best performing model to the resolution 5 hidden grid model to investigate the influence of including an autoregressive rollout finetuning phase.

5.2 High-resolution experiments

After testing the stretched grid on the lower resolution ERA5 data, we introduce the DOWA dataset, described in Chapter 4. Due to time and resource constraints, we adapt only Stage C of the framework described by [29]. Stage C is adapted to have 100km resolution globally, instead of 31km, without applying transfer learning. We implement some hyperparameter adaptations to improve learning performance, and train a graph-transformer model in the new Anemoi framework, developed by ECMWF. Although we planned on implementing the stretched grid using the local DOWA data, due to slow retrieval speeds we were not able to obtain the full dataset initially. We decided to start experiments when 7 of the 10 available years were converted. The initial model was trained on 5 years of data (2008, 2009 and 2013-2015), with validation year 2016 and testing year 2017, based on the data available after pre-processing. Additionally, preliminary results from both KNMI and MET Norway indicate the potential performance benefits of increasing the learning rate and the number of channels. The maximum learning rate is increased eight-fold, to a rate of $5 * 10^{-5}$, effectively reducing the training time by half. The number of channels is raised from 256 to 512, with the reasoning that the large data volume available in this field elicits increasing the parameter space.

Besides, adjustments were made to the model connectivity. Previously, a fully connected encoder was deemed necessary for optimal information flow across the network. However, preliminary results suggest that a smaller connectivity minimally impacts WMSE performance. It appears that local data points are of highest importance during encoding and disconnected nodes exert limited influence, allowing the number of neighbouring connections to a grid point to be reduced to 10. This adjustment conserves the GPU memory necessary for increasing the number of channels without compromising model efficiency. To further alleviate GPU memory constraints, he model is distributed over 4 GPUs using PyTorch's DDP (Distributed Data Parallel) framework. Moreover, the batch size was reduced to 1 during training and validation, doubling the available GPU memory. To make a fair comparison with transfer-learning models such as the model by [29], trainable parameters are omitted from the model. The model is trained for 150 epochs on 16 NVIDIA GPUs at the ECMWF HPC called ATOS.

These high-resolution experiments aim to answer three of the research questions from Chapter 1: 4) The impact of training on 2.5 km resolution data 5) the effectiveness of the model in capturing extreme events and 6) the performance of the model compared to the HA model. Research question 4 is answered by evaluating the high-resolution model described above on the test year 2017 and examining the RMSE and power spectra plots. Research questions 5) and 6) will be addressed using a case study from the 22nd and 23rd of February, containing a storm with moderate impact on the Netherlands. For this case study, the pressure and wind speed of the SG-AIFS model predictions will be compared to the HA model predictions, allowing us to infer information on the performance of the model on extreme events. These experiments collectively form a comprehensive framework for analyzing the strengths and limitations of the SG-AIFS model, which will be assessed in the following chapter through both quantitative and qualitative performance analyses.

Chapter 6

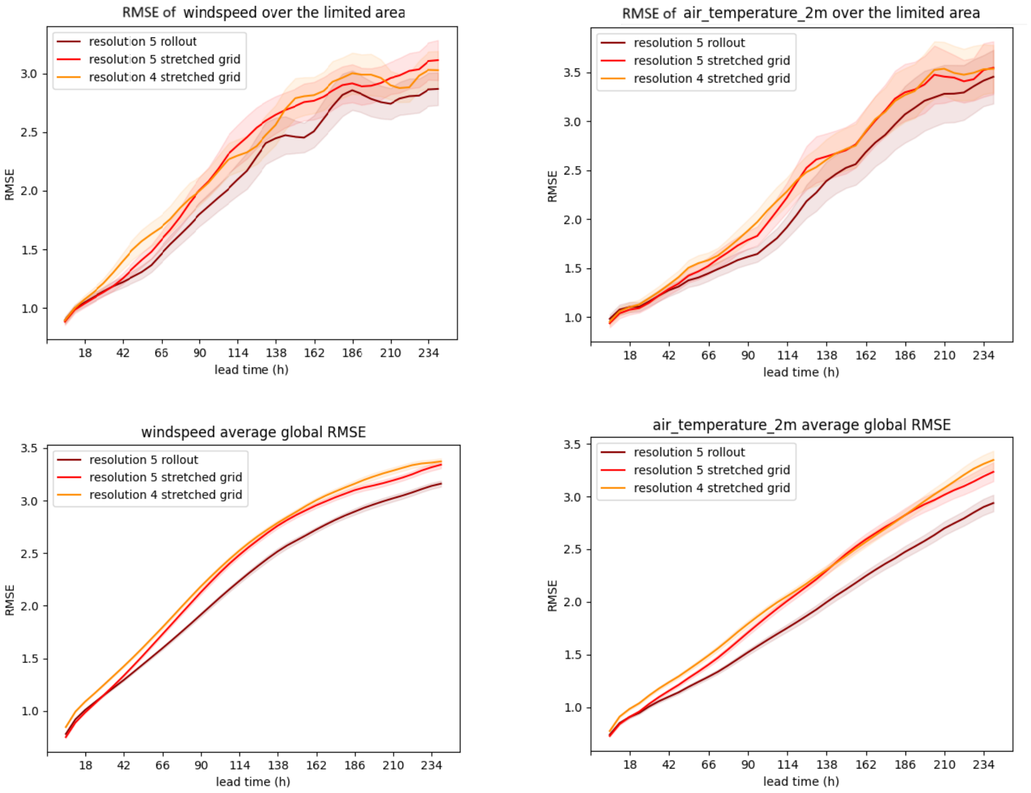
Results

This chapter presents the results of the experiments described in Chapter 5. First, convergence results of the ERA5 experiments are analyzed, examining convergence speed and final training and evaluation loss scores. In addition, inference results for the ERA5 experiments are provided, including Root Mean Squared Error (RMSE) evaluation at extended lead times. Eighteen initial dates are selected from the test set (see Appendix B for a list of testing dates), equally distributed throughout the year to account for seasonal variability. For the four 6-hour time steps included in these dates, autoregressive rollout predictions for up to 40 time steps (10 days) are initialized and the results are averaged over the selected 18 days to obtain the final averaged RMSE values. Besides, qualitative analyses are presented using visualizations of the best-performing model's predictions for January 2, 2022, and power spectrum analysis is conducted on this date to diagnose blurring. For the high-resolution experiments, visual examination together with RMSE plots and power spectrum analysis is utilized to provide an overview of the model performance. The impact of missing values due to the conversion from model to pressure levels described in Chapter 4 is inspected visually. Finally, the case study described in Chapter 5 is conducted and visualizations of the air pressure at sea level and wind speed are examined to assess the performance of the SG-AIFS model for an extreme weather event compared to the HA model.

6.1 ERA5 experiments

6.1.1 Quantitative results

In the first experiment, we compare the SG-AIFS model with a global grid resolution of 4 and local grid resolution of 6 against the same configuration, but with a global grid resolution 5 and local grid resolution of 7. Convergence results for this experiment are presented in Figure 6.2. SG-AIFS with global resolution 5 exhibits faster convergence, achieving a lower final loss. This can be attributed to the expanded parameter space facilitating accelerated adaptation to the training samples. Following training, model performance is assessed through performing several consecutive evaluation rollout steps on selected dates from the test set, allowing for an analysis of prediction accuracy at extended lead times. The evaluation focuses on two variables: the temperature at 2 meter height in Kelvin (K) and the wind speed in meter per second (m/s) at 10 meter height. The RMSE as a function of lead time, evaluated on the local domain and the global domain, is shown in Figure 6.1. Both evaluation variables show a near-linear increase of RMSE with lead time, which is expected given that predictability of the atmosphere diminishes with lead time. Although resolution 5 significantly lowers the global RMSE for lead times up to 114 hours, the difference in performance between the hidden grid resolutions at longer lead


RMSE of windspeed over the limited area

RMSE of air_temperature_2m over the limited area

windspeed average global RMSE

air_temperature_2m average global RMSE

Figure 6.1: SG-AIFS averaged RMSE of the air temperature (K) at 2 meter height and the wind speed at 10 meter height, evaluated on 18x4 cases of the test set (Appendix B) for lead times up to 10 days. The colored bands represent the 95% confidence intervals. Increasing the hidden grid refinement to resolution 5 improves RMSE values for short lead times. However, performance is similar or worse for longer lead times. Rollout training substantially lowers RMSE for all lead times.

times is minimal. This indicates that a refinement level of 4 could suffice for data at this resolution. Computational resources (in particular GPU memory) can be saved by using a lower resolution hidden grid. However, further experimentation is required to determine how these results generalize to higher resolution datasets such as DOWA. On the regional domain, the SG-AIFS model with resolution 5 demonstrates improved short lead time performance, however this effect may be influenced by fluctuations in RMSE associated with a smaller sample size. Indeed, the 95% confidence intervals suggest that the difference lacks statistical significance. On the other hand, the short lead time global performance increase may advocate for additional latent space resolution, although this effect diminishes over time.

Prediction accuracy at extended lead times can be enhanced by finetuning on longer lead times (rollout training). In the experiment, the SG-AIFS model with resolution 5 is finetuned for up to 12 rollout steps (3 days), as shown in Figure 6.1. Rollout training significantly improves the performance for short and long lead times for both models, in alignment with the results found by [22], [24] and [29], highlighting the importance of finetuning. Rollout training could substantially decrease temperature RMSE up to 0.5 degree Celsius temperature and wind speed RMSE up to 0.5 m/s. On the other hand, high impact of rollout training may indicate the necessity for further training, although consistency in RMSE performance strongly indicates

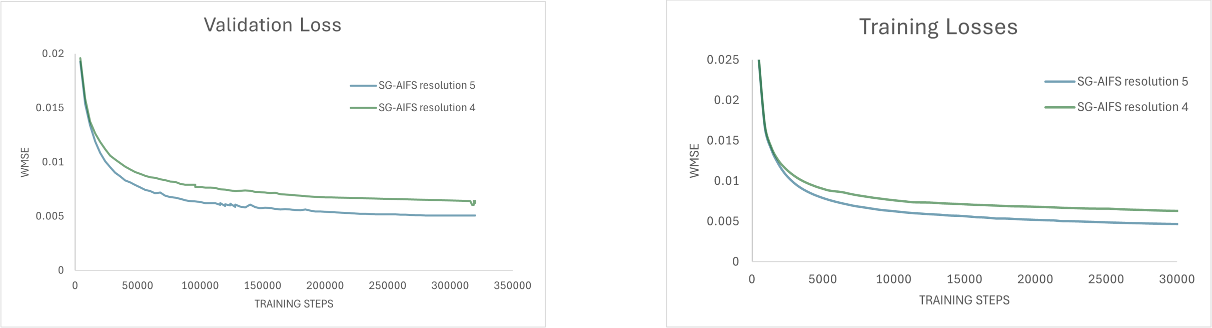


Figure 6.2: Influence of different hidden grid resolution on model convergence. The model with hidden grid resolution 4 has a limited area resolution of 6. The model with hidden grid resolution 5 has a limited area resolution of 7. Higher resolution results in faster convergence for both validation loss as well as training loss.

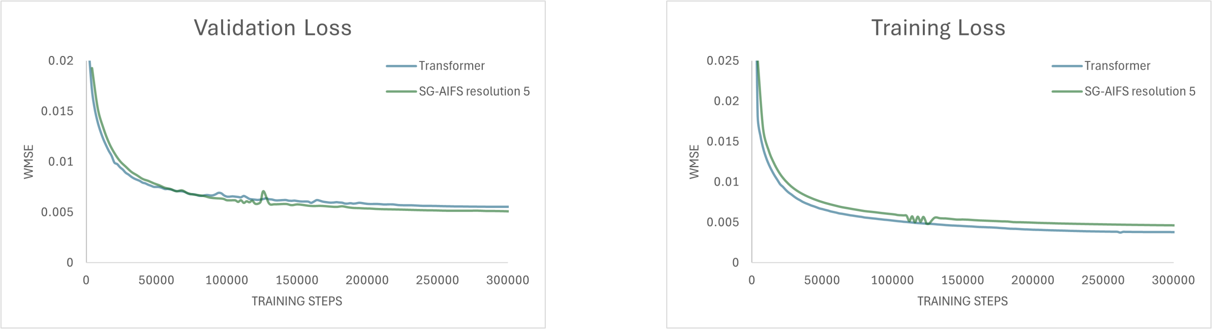


Figure 6.3: Comparison of the WMSE between the transformer model and the SG-AIFS model. Including the transformer architecture appears to accelerate training convergence, although resulting in a higher final validation loss.

convergence (Fig 6.1).

The second experiment compares the Transformer processor architecture to the GNN-based stretched-grid model. The results of this experiment are presented in Figure 6.3. The inclusion of the Transformer processor leads to faster convergence relative to the GNN model, with the attention mechanism correctly identifying relationships between nodes in fewer training steps than the GNN. However, the GNN processor results in an improved final validation loss, while the Transformer displays poor generalization in comparison. Given the Transformer's higher number of parameters - with 105 million parameters compared to the GNN's 17.6 million - it is suspected that the model is overfitting on the training data and thereby reducing its generalization capability on the validation data. Moreover, the high number of parameters causes the iterations per second to drop, yielding minimally improved overall training time.

Examining the performance on the test set reveals a relatively high RMSE for the selected surface variables. Additional RMSE figures for lower-pressure-level variables are provided in Appendix C.2. While RMSE values for the air temperature (in Kelvin) at 150 hPa display strongly improved performance, RMSE values for variables at lower atmospheric levels reflect the opposite. This contradiction suggests that the pressure level performance of the Transformer is imbalanced, indicating the necessity of adjustments to improve performance at higher pres-

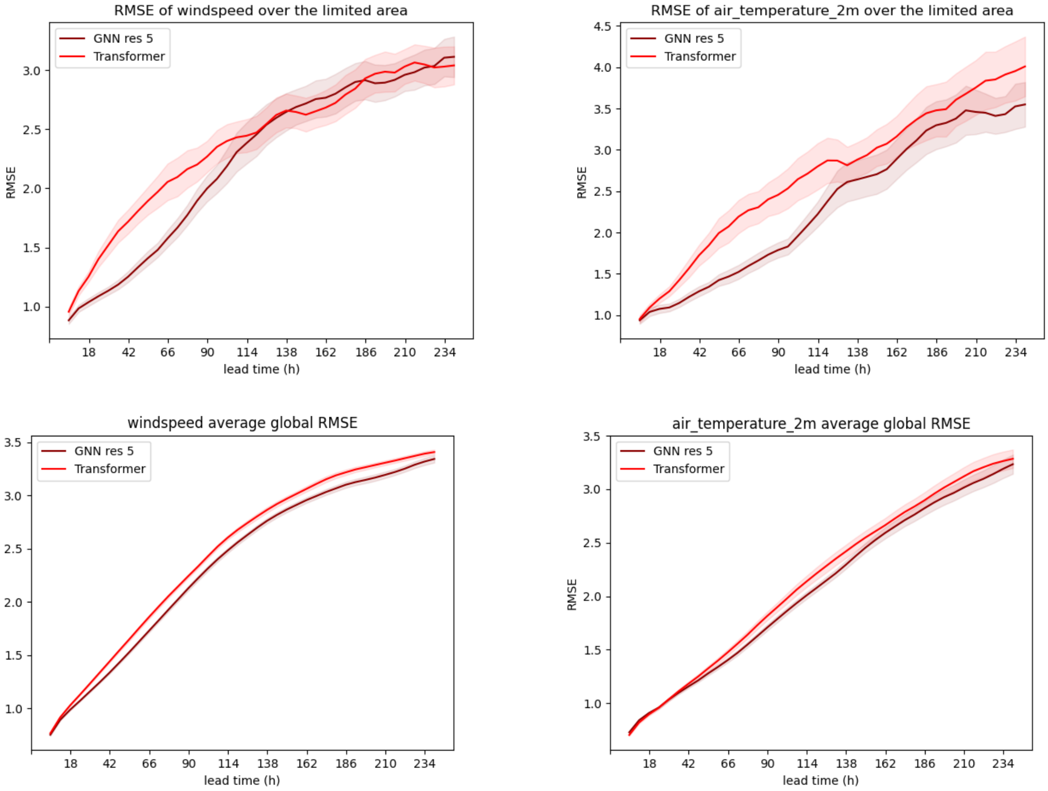


Figure 6.4: SG-AIFS GNN processor compared to the Transformer processor, averaged RMSE of the air temperature (K) at 2 meter height and the wind speed (m/s) at 10 meter height, evaluated on 18x4 cases of the test set (Appendix B) for lead times up to 10 days. The colored bands represent the 95% confidence intervals. The Transformer model shows decreased performance compared to the GNN over these surface variables.

sure levels. Furthermore, the plots in Appendix C.2 reveal certain variables that have good performance globally failing to predict local values accurately, particularly at short lead times. This may indicate an imbalance in the weighting of the high-resolution data, which could be attributed to the single dimensional processor structure as mentioned in Chapter 5. In conclusion, preliminary results show the necessity for further tuning and developing the Transformer framework before similar RMSE performance could be achieved in fewer training steps. Until proper pressure level weighting is implemented in the AIFS framework, the GNN will continue to outperform the Transformer on surface variables at both the global and regional domain.

6.1.2 Qualitative results

Qualitative analysis reveals reasonable predictions for short lead times. In Figure 6.5 visualizations of the temperature at 2 meter height in Kelvin (K) and the windspeed in meter per second (m/s) at 10 meter height are shown. In general, we see that surface variables and variables at higher pressure levels show better performance than upper air variables. This is comparable to GraphCast and AIFS model results ([22], [24]). For longer lead times the model tends towards the mean, resulting in blurred forecasts. Smoothing is a prevalent issue in deterministic deep learning for weather prediction (DLWP), since the mean squared error loss (MSE loss) tends to favour predictions towards the mean of the distribution [40].

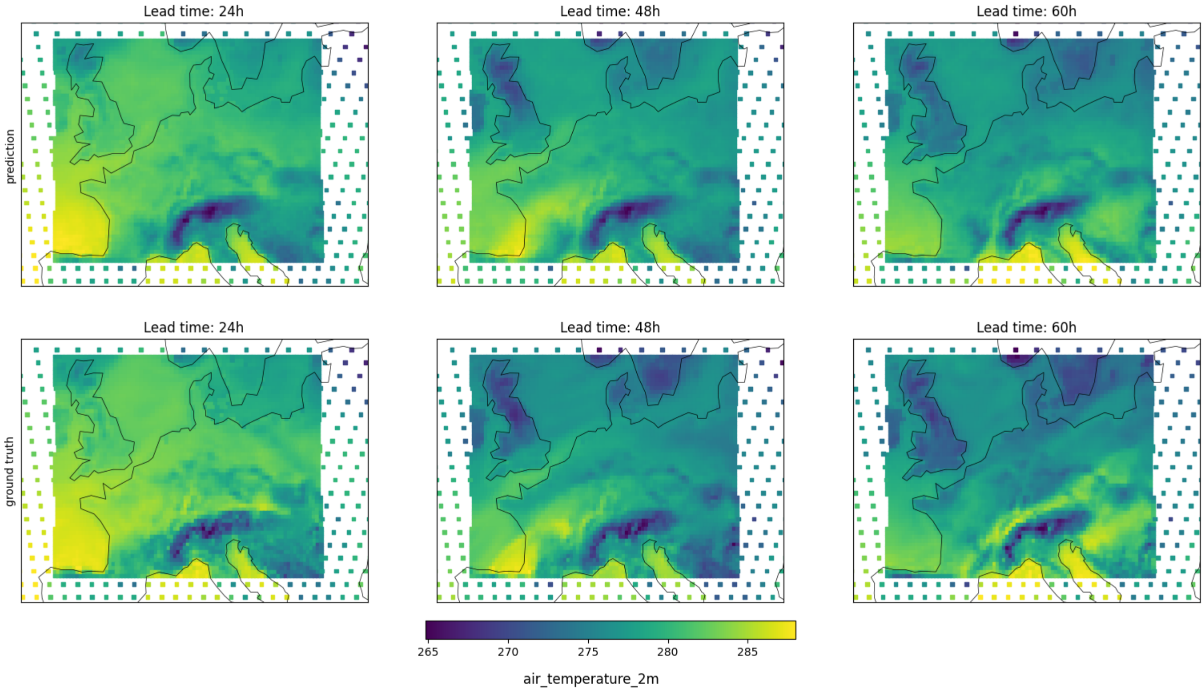
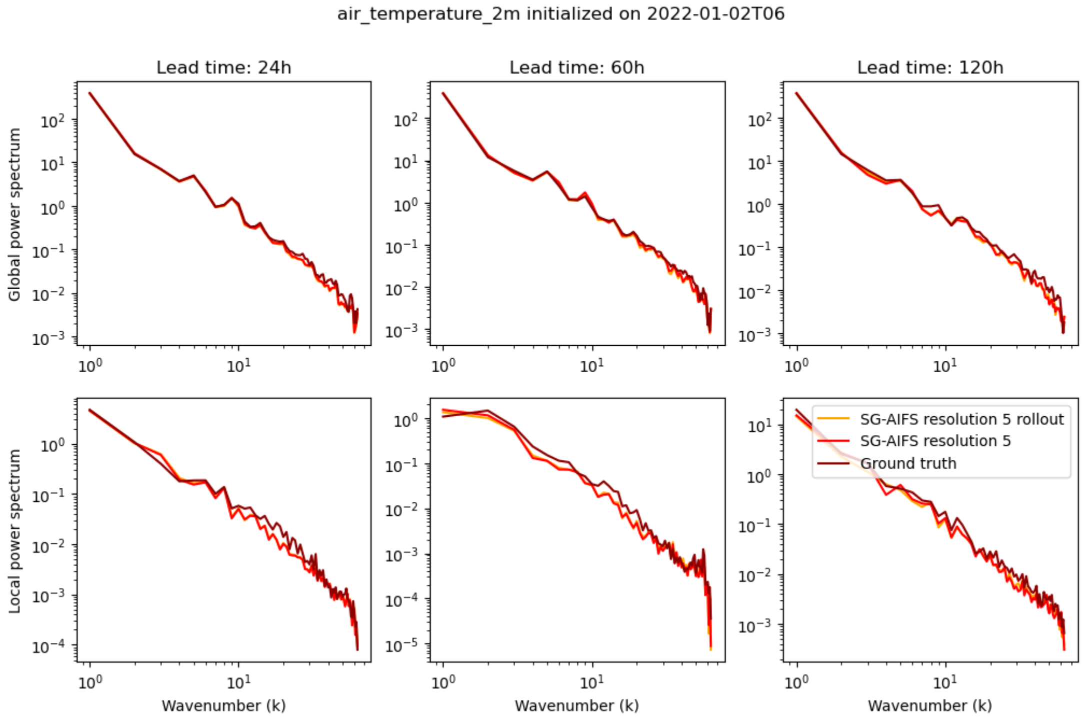
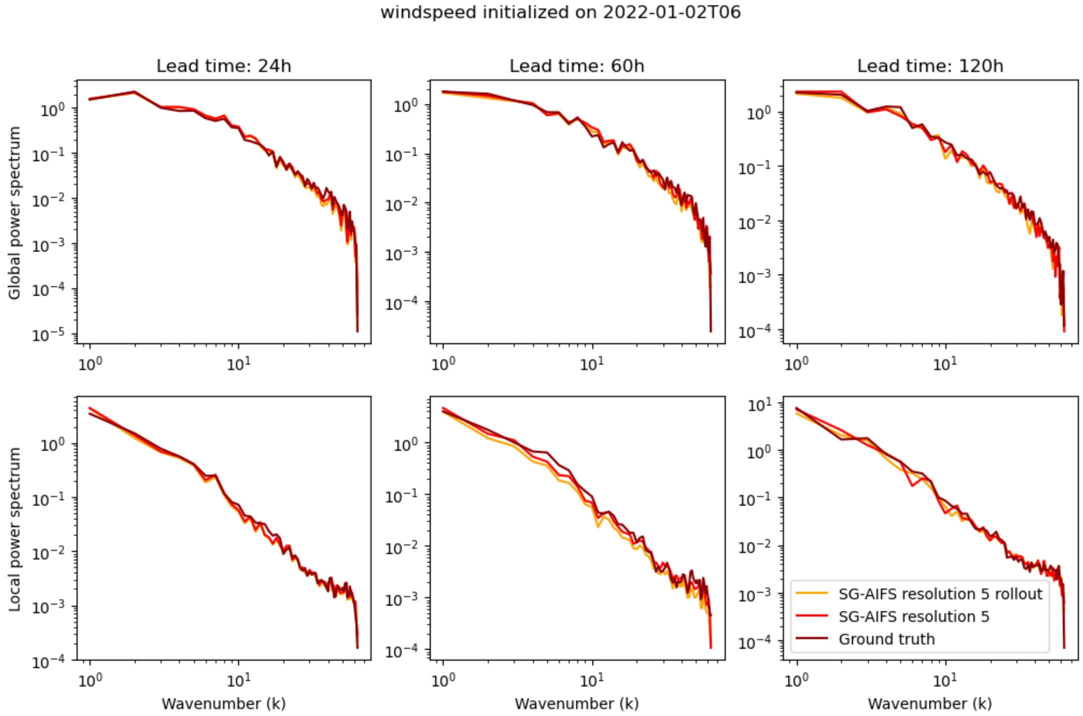


Figure 6.5: SG-AIFS with hidden grid resolution 5 (7) with rollout evaluated on the test set. The first and second row of the figure show the SG-AIFS forecast and ground truth, respectively, of the air temperature (K) at 2 meter height. The forecast is initialized from 2022-01-02T06. The model learns to predict on a high-resolution data grid, although at longer lead times the quality of the forecast decreases.



(a) Power spectra of the air temperature (K) at 2 meter height for a single case from the test set (2022-01-02 at 06UTC). The top row displays the global power spectra for lead times 24h, 120h and 180h, the bottom row shows the regional power spectra for lead times 24h, 120h and 180h.



(b) Power spectra of the wind speed (m/s) at 10 meter height for a single case from the test set (2022-01-02 at 06UTC). The top row displays the global power spectra for lead times 24h, 120h and 180h, the bottom row shows the regional power spectra for lead times 24h, 120h and 180h.

To diagnose smoothing in the model, a power spectrum analysis is performed. Power spectrum analysis is based on spherical Fourier transforms, stating that any real square-integrable function $f(\theta, \phi)$, defined on the unit sphere, can be expressed as a series of spherical harmonic functions [10]:

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} f_l^m Y_l^m(\theta, \phi), \tag{6.1}$$

where Y_l^m is a complex spherical harmonic function, given as

$$Y_l^m(\theta, \phi) = e^{im\phi} P_l^{-m}(\cos(\theta)), \tag{6.2}$$

a solution to the Legendre equation $\nabla Y = 0$. Here, m is the order, and l is the degree of the Legendre polynomial P_l^{-m}. The complex spherical harmonic coefficients f_l^m can then be calculated as follows:

$$f_l^m = \frac{1}{4\pi} \int_{\omega} f(\theta, \phi) Y_l^{m*}(\theta, \phi) d\Omega, \tag{6.3}$$

where $d\Omega$ is the differential surface area on the unit sphere and the asterisk denotes complex conjugation. These coefficients indicate the contribution of each spherical harmonic function to the function f. A power spectrum S for a function f at degree l is then defined as [10]:

$$S_{ff}(l) = \sum_{m=-l}^{l} |f_l^m|^2. \tag{6.4}$$

The power spectrum allows us to analyze how well the model performs at different scales (wavelengths). The example power spectra for the air temperature at 2 meter height and the wind speed at 10 meter height on 2022-01-02 at 06UTC can be seen in Figure 6.6a and Figure 6.10b, respectively. For longer lead times we observe minimal decrease in the spectral power of high wavelengths, corresponding to an under-representation of small-scale features of the model. Rollout training does not appear to decrease blurring, but instead shows similar loss of small spatial features for both the wind speed and temperature. This discrepancy can be viewed as a consequence of the small relative difference and the low regional data resolution. When moving to higher resolution regional data, these differences are expected to increase.

6.2 High-resolution experiments

Considering the results presented above, the initial high-resolution model is trained using a Graph-Transformer architecture, with a global resolution of 5 and a local resolution of 9, aligning with the increased DOWA dataset resolution. As described in Section 5, Stage C of [29] is adapted to 1 degree resolution globally, without using transfer learning. Qualitative and quantitative results are presented, as well as an evaluation of the model using a case study. Note that the ground truth data consists of +3h reforecast DOWA data.

6.2.1 Example forecasts and RMSE

Example forecasts for the air temperature at a height of 2 meters are presented in Figure 6.7 and Figure 6.8. The model captures some temperature features even at extended forecast lead times. Diurnal cycles seem to be somewhat represented. However, a large overestimating and underestimation of temperature can be observed (up to 30 degree Celcius globally), indicating a

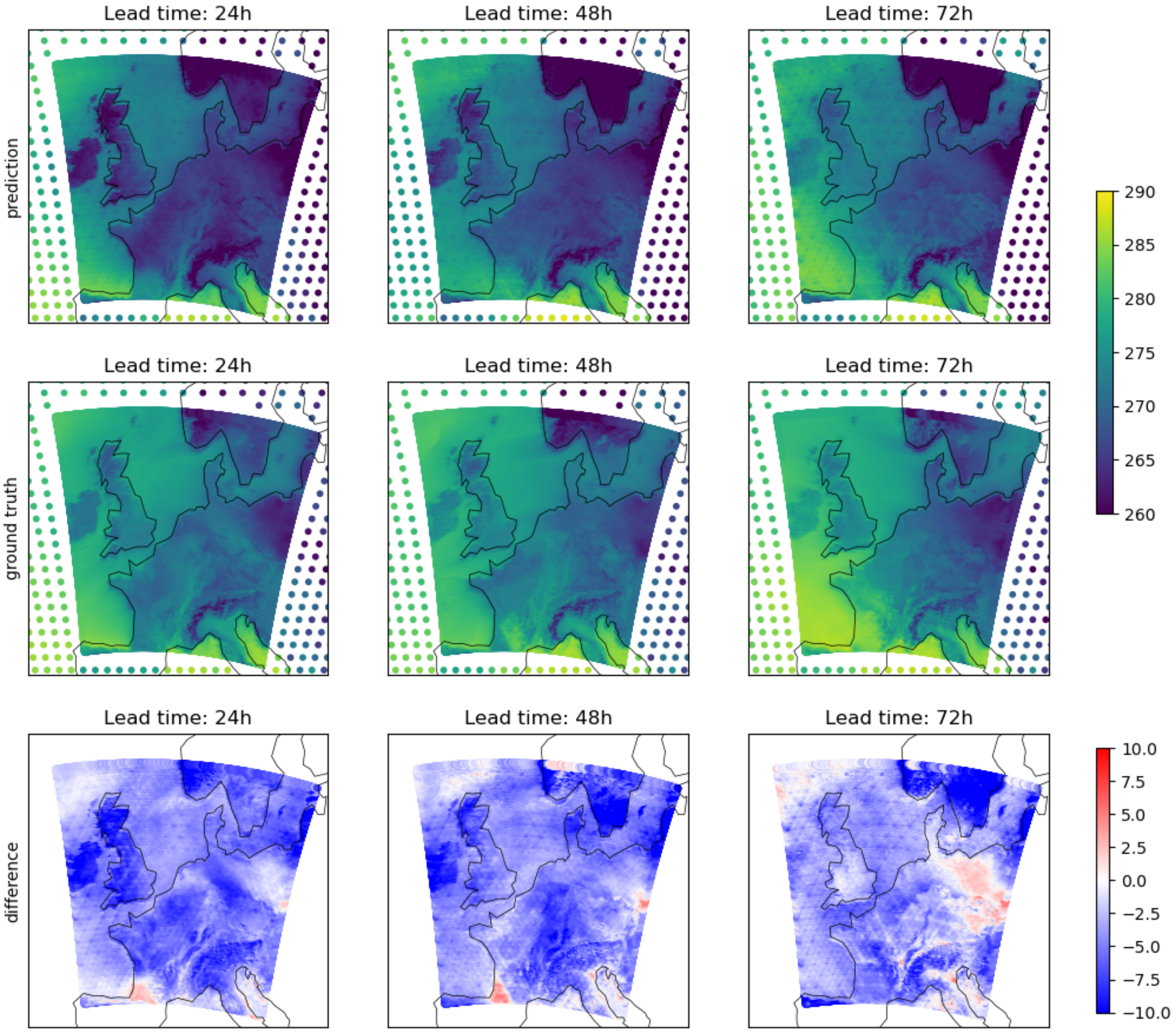


Figure 6.7: SG-AIFS with hidden grid resolution 5 (9) trained on ERA5 globally and DOWA locally, initialized from 2017-02-10T06. The first and second row of the figure show the SG-AIFS forecast and ground truth, respectively, of the air temperature (K) at 2 meter height, and the bottow row shows the air temperature difference between the ground truth and the forecast. The model learns to predict on a high-resolution data grid, although longer lead times reveal underestimated air temperature values and the difference plots show hidden grid artefacts.

deviation away from the mean. The model encounters challenges in extracting detailed features, with visible artifacts reflecting the structure of the latent space, particularly noticeable in the predictions of the wind speed at a height of 10 meters (see Appendix C.1, Figure C.1) and in the difference plots in Figure 6.7 and 6.8. The global wind speed at a height of 10 meters (Figure C.2) highlights the model capability to capture large-scale atmospheric patterns. At 60-hour lead times, some high-level features remain reasonably accurate. Similarly, global temperature forecasts (Figure 6.8) exhibit some accuracy for extended lead times. As anticipated, predictions show incremental smoothing over time, which can be attributed to the deterministic training of the model towards the WMSE loss, resulting in the model's diminished ability to capture finer-scale patterns. Moreover, at longer lead times the forecast error accumulates rapidly and underestimation of extremes becomes more prevalent. This inclination is particularly evident when plotting the RMSE as a function of lead time, as can be seen in Figure 6.9. The RMSE rises steeply from 6h until 42 h, after which the error increases less. Notably, the RMSE of the temperature at 2 meter doubles compared to the corresponding RMSE during the low-resolution experiments, whereas the RMSE for the wind speed remains of a similar magnitude. This discrepancy may suggest overfitting, given the large parameter space and the limited available training data. The observed artifacts matching the grid lines in the hidden grid support this hypothesis.

6.2.2 Power spectra

Similarly to Section 6.1.2, blurring in the model is diagnosed using power spectrum analysis, allowing us to analyze the performance of the model at different wavelengths. The power spectra for the air temperature at 2 meter height and the wind speed at 10 meter height can be seen in Figure 6.10b and Figure 6.10a, respectively. Wind speed global power spectra show a slight underestimation of power across all wavelengths, also for lead times with poor predictive skill (lead time +120h). On the regional domain a significant loss of power is observed across all wavelengths. Already after 24 hours, smoothing is observed for both the wind speed at 10 meter height and the temperature at 2 meter height. This is in accordance with the limitations of deterministic data-driven models found by Bonavita et al. [6]. Since smoothing and underestimation of extreme values appear to be especially prevalent over the DOWA domain, the necessity of moving away from the Mean Squared Error loss and towards probabilistic modeling is even more important for limited area modeling. Deterministic models fail to provide a realistic representation of the chaotic nature of the atmosphere and tend towards the mean of the distribution. The uncertainty is better quantified in probabilistic models such as GenCast, a diffusion model introduced by [33]. On the other hand, the temperature power spectra display higher power than the ground truth. This reflects the overestimation and underestimation of temperature values seen in Figure 6.7 and 6.8, of which the exact cause remains to be investigated.

6.2.3 Missing value prediction

As described in Chapter 4, the preprocesssed DOWA dataset contains high pressure level variables with missing values dependent on the pressure and the orography. Since these missing values are imputed with the average of each field so that the model can easily predict these values, the assessment of the SG-AIFS model prediction of these values is necessary. We evaluate the missing values of the temperature at 1000 hPa, the pressure level with on average the most imputed values. The predicted values and ground truth values are displayed in Figure 6.11. Over extended lead times, an increasing overestimation of the missing values' magnitude is observed, possibly due to the influence of other fields on the temperature predictions. Furthermore, after 60 hours lower pressure levels cause the amount of missing values to noticeably increase (for

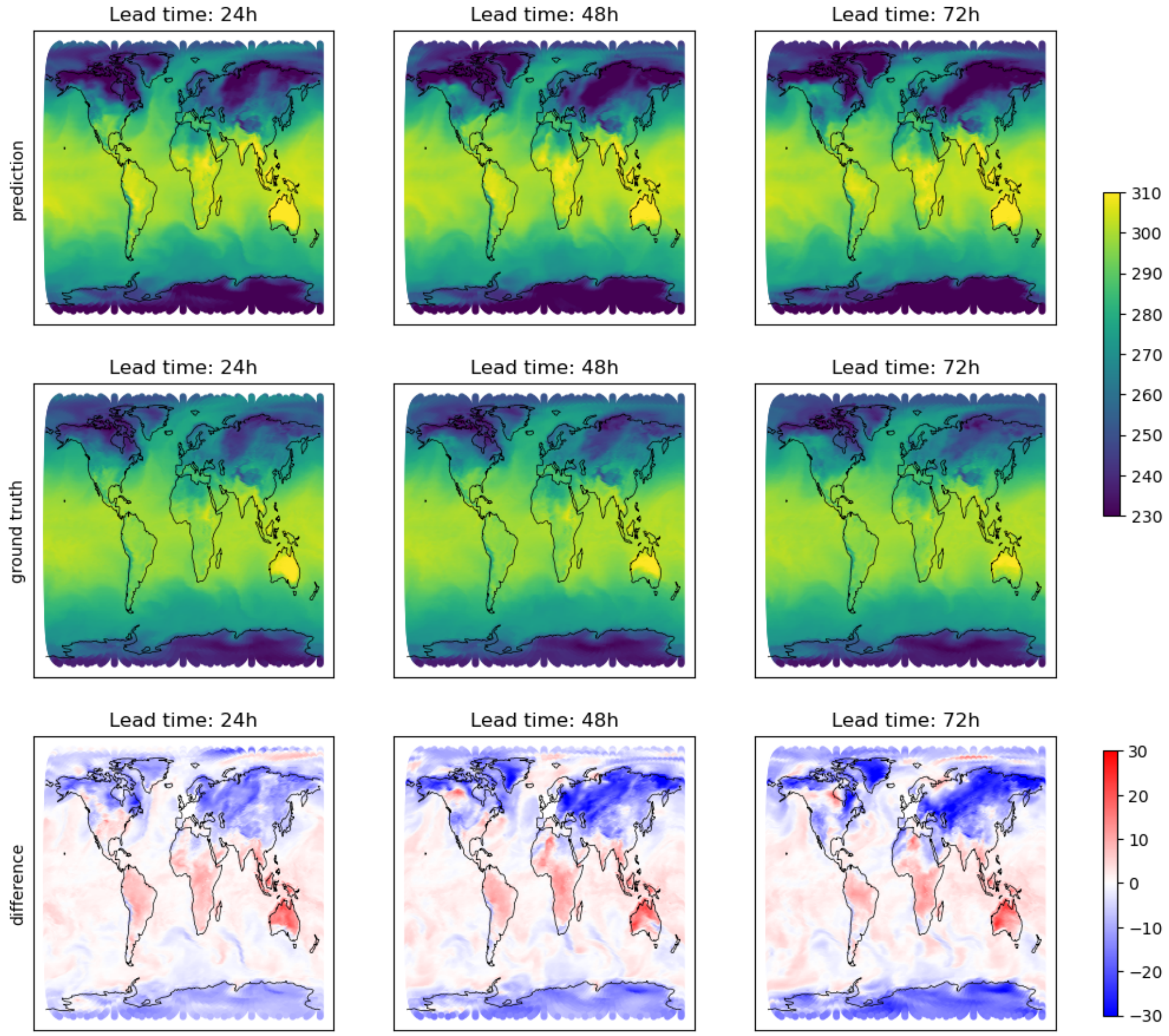


Figure 6.8: SG-AIFS with hidden grid resolution 5 (9) trained on ERA5 globally and DOWA locally, initialized from 2017-02-10T06. The first and second row of the figure show the SG-AIFS prediction and ground truth of the air temperature (K) at 2 meter height. The bottom row shows the air temperature difference between the ground truth and the forecast.

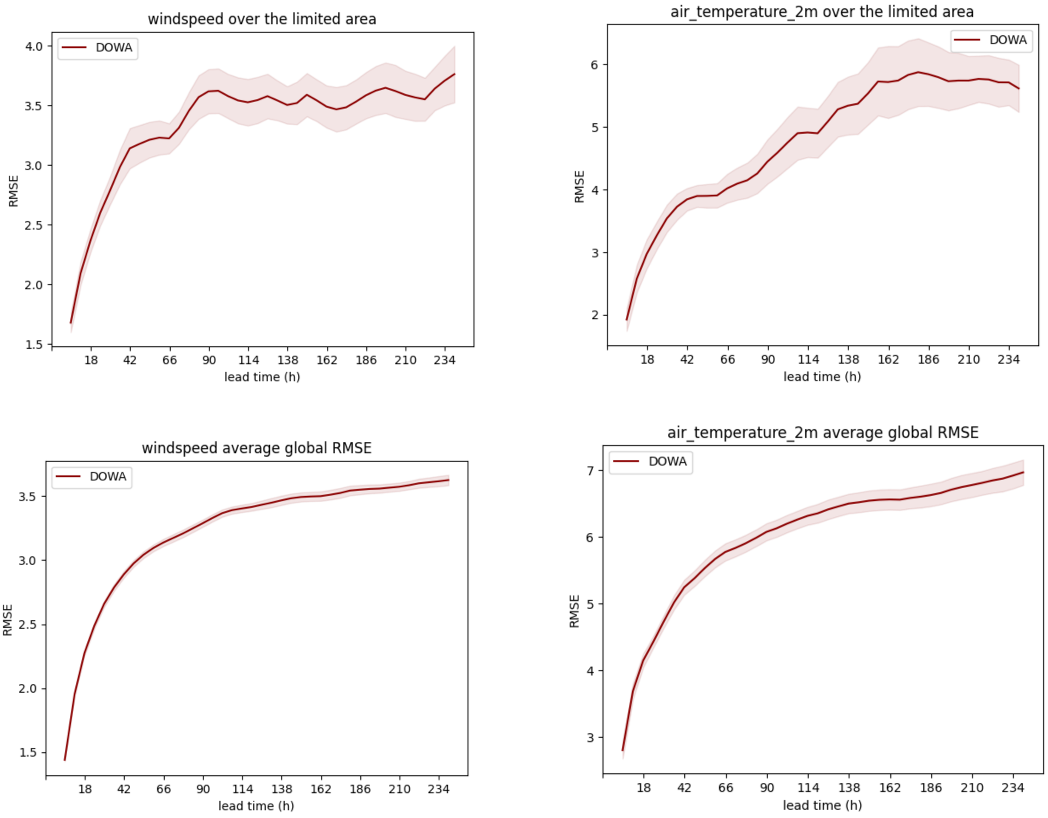


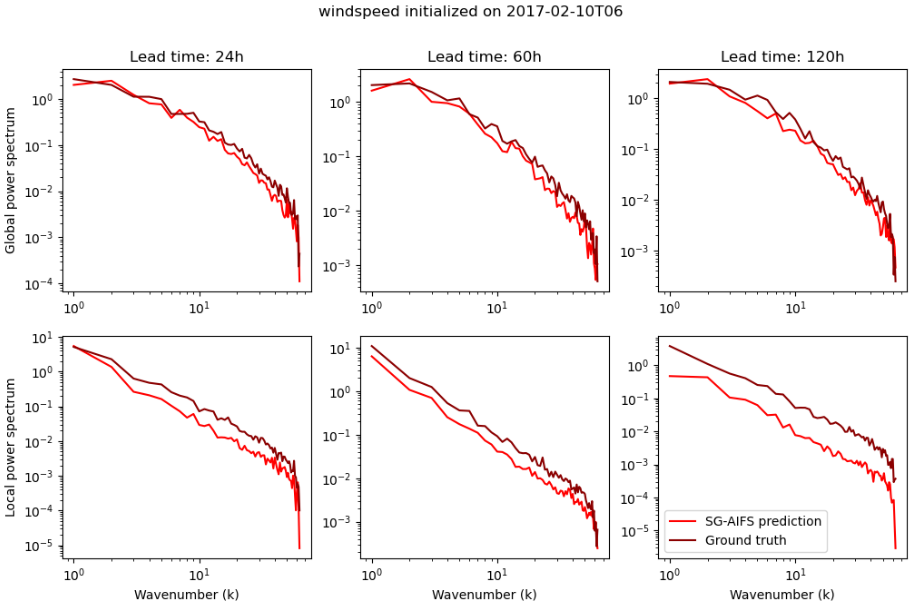
Figure 6.9: SG-AIFS trained on 5 years of ERA5 and DOWA data. The figure shows the averaged RMSE of the air temperature (K) at 2 meter height and the wind speed (m/s) at 10 meter height, evaluated on 18x4 cases of the test set (Appendix B) for lead times up to 10 days. The colored bands represent the 95% confidence intervals. RMSE values increase rapidly from +6h onwards.

example in the west of France). The SG-AIFS model fails to properly capture this phenomenon, and instead assumes a constant number of missing values. This indicates that the model has not properly learned to associate the decrease in pressure with the increase in missing values.

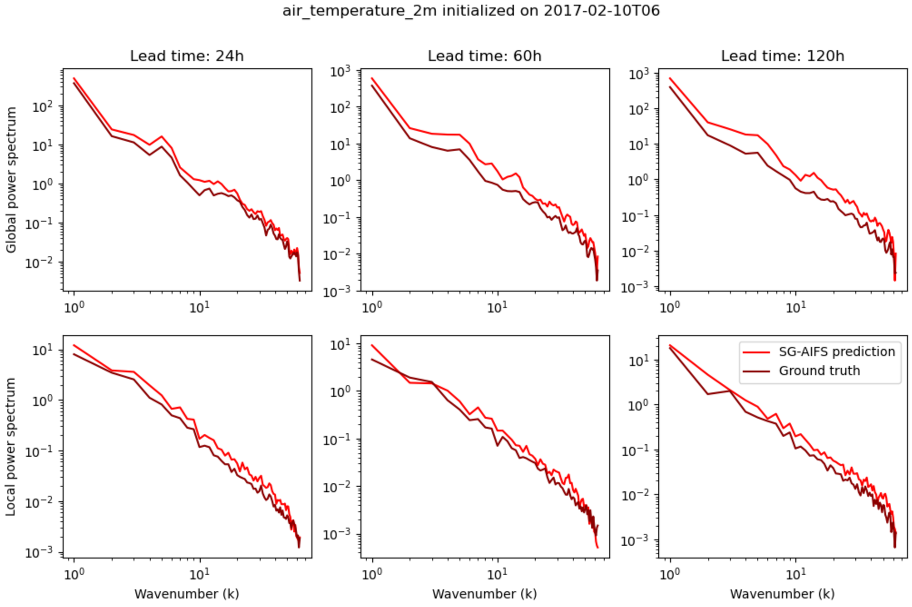
Both of these aspects, combined with possible errors resulting from the linear interpolation used to convert the DOWA model levels to ERA5 pressure levels, emphasize the need for a different approach when dealing with missing values, such as introducing a separate encoder and decoder.

6.2.4 Case study

One of KNMI's core tasks is extreme weather prediction, issuing warnings for high-impact weather events. In order to compare our model to the operational HA forecast and to evaluate the model performance on an extreme weather event, a case study from the test year 2017 was selected, containing a storm with moderate impact on the Netherlands. On the 22nd of February 2017 a low pressure system with a core pressure of 985hPa developed over Ireland. The low pressure system deepened until it arrived above the North Sea, before continuing eastwards. The passage of this low pressure system caused extreme wind gusts in a small area. On the 23rd of February the wind speed increased, reaching a peak in the west and the southwest of the



(a) Power spectra for a single case from the test set (2017-02-10 at 06UTC) of the wind speed (m/s) at 10 meter height for rollout times 24h, 60h and 120h. The top row displays the global power spectra for lead times 24h, 60h and 120h, the bottom row shows the regional power spectra for lead times 24h, 60h and 120h.



(b) Power spectra for a single case from the test set (2017-02-10 at 06UTC) of the air temperature (K) at 2 meter height for rollout times 24h, 60h and 120h. The top row displays the global power spectra for lead times 24h, 60h and 120h, the bottom row shows the regional power spectra for lead times 24h, 60h and 120h.

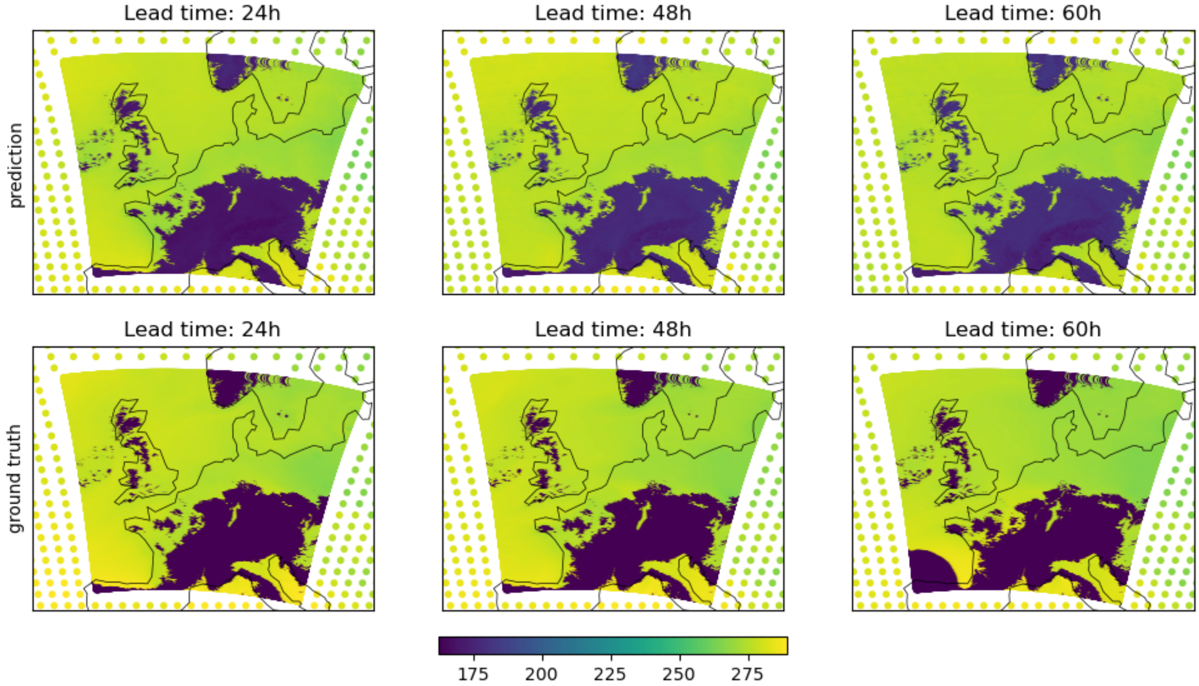


Figure 6.11: Missing values of the air temperature (K) at 1000 hPa on the 10th of February 2017 at 06UTC. The magnitude of the missing values is slightly overestimated, and the increase in missing values after 60 hours due to decreasing pressure is not properly captured.

Netherlands at 12:00CEST, between 16:00CEST and 18:00CEST; and between 20:00CEST and 21:00CEST. Wind gusts up to 115km/h were measured along the west coast, causing disruptions in traffic due to falling trees. Lower wind speeds occurred in the north, where the centre of the low pressure system passed over. From the 22nd of February 2017 02:01 CEST until the 23rd of February 23:26CEST KNMI issued warnings for very severe wind gusts of 100-120km/h for North-Holland, South-Holland, Zeeland, Utrecht and North-Brabant. In the rest of the country warnings were issued for severe wind gusts of 80-90km/h. The warning was issued until the 23rd of February 2017 [38].

We analyze the air pressure at sea level and the wind speed predictions of the HA cycle 40 forecast compared to the SG-AIFS forecast, initializing both forecasts from the 22nd of February 2017 at 00:00UTC. The air pressure at sea level on the 22nd and 23rd of February can be seen in Figure 6.12. In terms of pressure, SG-AIFS performs similarly to HA for lead times +12h and +24h. For lead time +48h, HA still outperforms SG-AIFS in terms of location and intensity of the pressure. SG-AIFS shows overestimation of the pressure in the southeast, and underestimation in the north, whereas the HA predictions remain consistent through longer lead times. When comparing the wind speed predictions of the storm on the 23rd (see Figure 6.13), it can be noticed that although SG-AIFS predicts the approximate location of high and low wind speeds, detailed featured are missing. Furthermore, for extended lead times cross-shaped artefacts appear which we have not seen for the air pressure at sea level, but we have seen for other wind speed predictions (Figure C.1). On the other hand, the HA predictions remain closer to the ground truth and do not present any physical inconsistencies. We can conclude that more developments of the SG-AIFS model are needed to obtain the same or higher accuracy as the HA model.

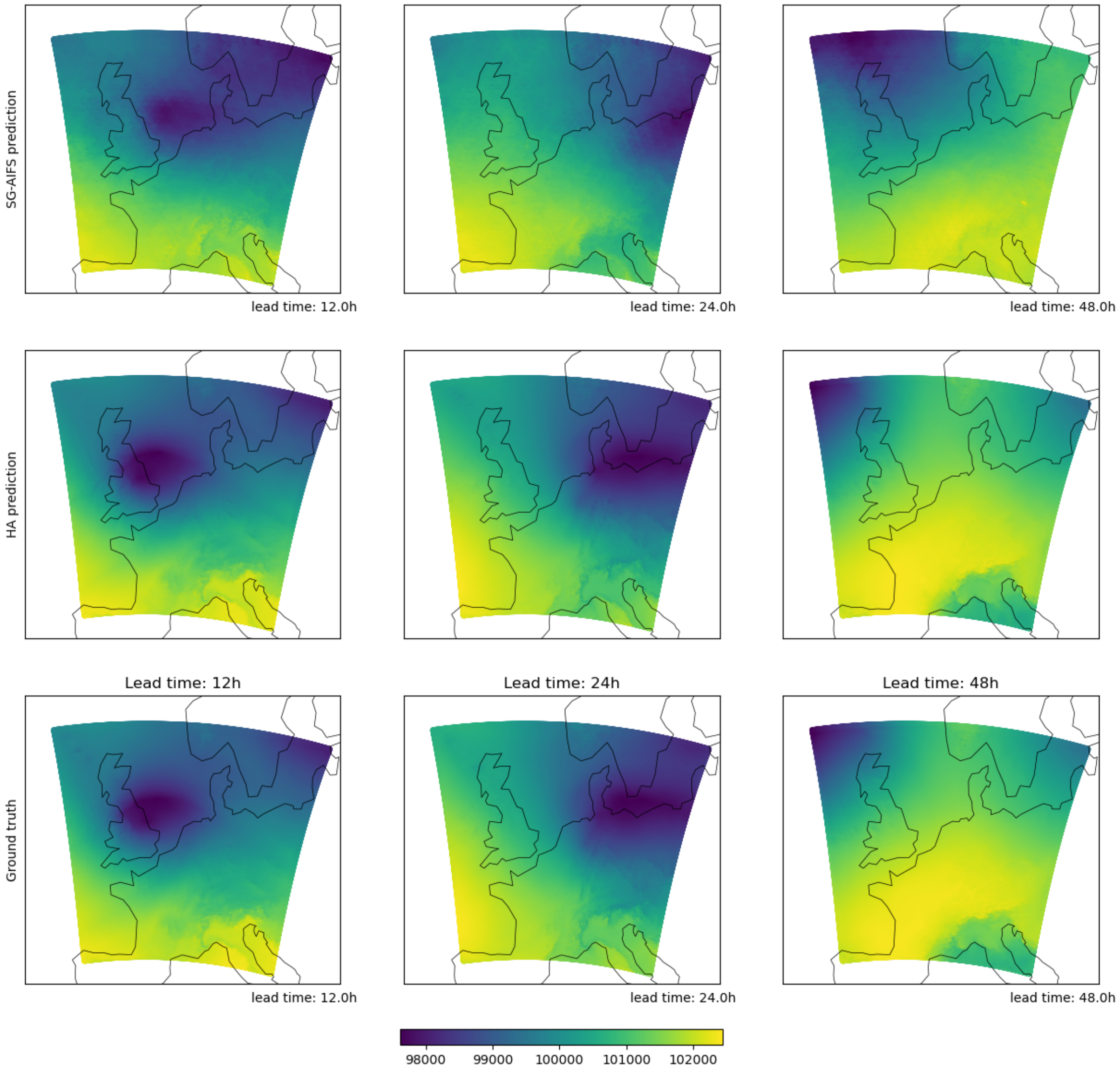


Figure 6.12: Air pressure (in Pa) at sea level on the 22nd and 23rd of February initialized from 22-02-2017 00:00UTC. The top row shows the SG-AIFS forecasts, the middle row shows the HA forecasts and the bottom row shows the ground truth.

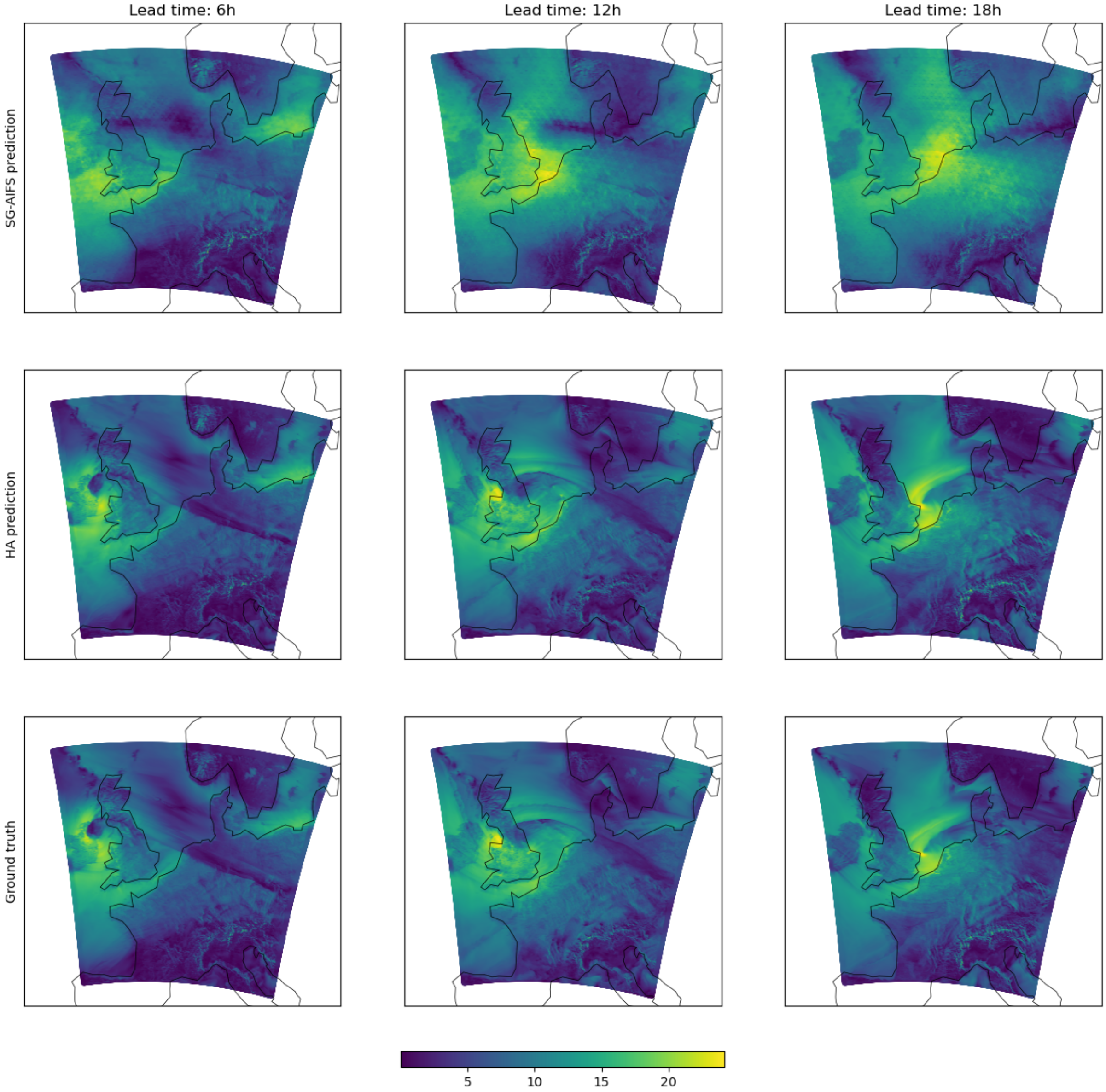


Figure 6.13: Wind speed (m/s) at 10 meter height on the 23rd of February 2017 initialized from 23-02-2017 00:00UTC, showing high wind speeds passing over Ireland and the UK towards the Netherlands. The top row shows the SG-AIFS forecasts, the middle row shows the HA forecasts and the bottom row shows the ground truth.

Chapter 7

Model limitations

In this section we present the limitations of DLWP models in general, as well as discussing shortcomings of the SG-AIFS model in relation to the results from Chapter 6. The limitations can be categorized into six main areas:

1. DLWP models

Some limitations of global DLWP models have been well-documented [34] [30] [7]. Data-driven models are dependent on the quality of the reanalyses and the data availability. DLWP models appear to learn the biases inherited from the reanalyses. Reanalyses depend on the quality of NWP models, in particular their data assimilation scheme and the quality of observations. Similarly to NWP models, initializing DLWP models directly from operational analysis or even finetuning it on operational real-time data (like in [23]), would improve the overall forecasting performance of DLWP models, since the performance of DLWP also depends on the quality of the training data, i.e. quality of initial conditions. Besides, as mentioned before, deterministic models trained to minimize the MSE tend to show blurred predictions, although this blurring effect has not been shown to increase significantly with lead time. Therefore, deterministic DLWP models tend to struggle with predicting extreme values. This is a result of the training objective, which favors predictions towards the mean (as shown by [40]) and thus performance for the tail of the distribution is limited. Given the societal impact of extreme weather, further improvement of the extreme weather prediction skill of DLWP models is required. Another aspect of DLWP models is that they are not physically constrained and might show nonphysical behaviors both inside and outside the learned data distribution. Careful supervision is required to guarantee physical properties are preserved. Hybrid NWP-DLWP models are able to impose these constraints. Finally, although some (aggregated) precipitation results have been documented [29][23][33] using precipitation as diagnostic variable, many global models -and the model presented in this MSc thesis- provide no precipitation input data for medium-range forecasts. This is partially due to the limited data quality of ERA5, since convection is not resolved.

2. The AIFS model

In this paragraph we discuss the limitations of the chosen AIFS model and the stretched grid approach. Due to the structure of the AIFS model and the dataloader, the data in- and output is limited to a specific data structure (Zarr) and type of variables (following ERA5 conventions). Furthermore, the model contains many - largely untuned -hyperparameter settings that are expensive to be optimized. Moreover, as noted by [29], the original AIFS model contains additional fields (trainable parameters) that are dependent on the grid structure, inhibiting direct transfer learning from AIFS. Furthermore, the input and output variables of each model must align precisely, as there are one encoder and one decoder available presently. This makes transfer learning less efficient, since a new model has to be trained for different applications. Due to these constraints, as well as limited time, this MSc thesis does not evaluate the added value that transfer learning might bring to model performance, although this was demonstrated already by [29].

3. ERA5 experiments

The principal drawbacks identified in the ERA5 experiment setup concern the size of the model and the data resolutions. Given that the SG-AIFS models were trained on regional data at a coarse resolution, it remains uncertain how indicative these results are for the performance of the high-resolution stretched grid model using the DOWA data. Furthermore, the model size was limited due to the single GPU memory restriction, and the question remains whether the models were fully optimized after training for 150 epochs, as the WMSE training loss displayed a small yet persistent decrease. In addition, the Transformer model exhibited signs of overfitting during convergence, along with an observed imbalance in pressure level performance.

4. High-resolution experiments

Identifying the shortcomings of the high-resolution experiments described in this report, the most significant obstacle was the data availability. Long data transfer times and data pre-processing times caused significant delay, attributed to the strict AIFS model input requirements and to the large initial size of the dataset of 50TB. The training data is a +3h reforecast and thus contains inaccuracies. The data size was restricted to 5 years, which is reflected in the model results. The high-resolution model exhibits two mayor inaccuracies: low predictive power beyond 6-hour lead times and the presence of artefacts reflecting the shape of the hidden grid. We suspect both phenomena can be attributed to overfitting, because of the steep incline of the RMSE curve and loss of power appearing across all wavelengths in longer lead time power spectra. There are three factors that could contribute to this overfitting: the model size, data size and the training regime. The model size was optimized with the assumption of utilizing 40 years of ERA5 data, as model performance generally improves with an increase in parameters given a sufficiently large dataset. Therefore, the number of channels was doubled. However, since the dataset was limited due to the factors described above, the risk of overfitting was greatly increased. Additionally, due to the large model size, we were unable to perform rollout training, further limiting performance at longer lead times. Finally, the exclusion of trainable parameters could have further degraded performance by potentially omitting essential field information. Concerning the hidden grid artefacts appearing in some of the longer lead time predictions: similar artefacts have been observed by [31]. The authors of [31] argue that these artefacts are caused by a lack of spatial connections over a longer range. Since message passing steps occur at all refinements of the hidden grid simultaneously, the decoder is not able to homogenize information from nodes at different locations of the hidden grid. Therefore, the artefacts could be associated with limited computational resources, since the global resolution remained at 1 degree latitude-longitude, causing an increased gap in resolution between the global and the regional (2.5 km resolution) dataset.

5. Missing values

Another important challenge identified in this thesis is the handling of missing values in the conversion from model to pressure levels, which are influenced by both pressure and orographic variations, as described in section 6.2.3. To enable model predictions despite these gaps, missing values were imputed with the average of each field. However, a visual examination of the model output after predicting the imputed values -specifically for the temperature at 1000 hPa- revealed that the model insufficiently captured the magnitude and temporal changes of the missing values.

6. Model evaluation

Regarding the model evaluation, the RMSE is known to provide limited evidence of the skill of DL models. Since the training loss is designed to optimize this metric, forecast smoothing can remain unnoticed. Furthermore, the eighteen selected dates might have been insufficient to provide a comprehensive analysis, particularly given the presence of temporal correlations between different run hours within the same day. Qualitative analysis reveals that better metrics are required to evaluate factors such as smoothing and variable dependence. Besides, the RMSE was computed with respect to the ground truth, hence an evaluation against station observations as in [29] is missing. Since the ground truth reanalysis is based on the HA model predictions, comparing the ground truth with HA predictions in the verification results in a biased comparison. Regarding the confidence intervals presented in this paper, time independence is assumed which is a requirement that is generally not met for time series predictions, as these are highly correlated. Besides, for the power spectra equal coverage of the globe is assumed, which is an assumption that does not apply to limited area evaluation.

Chapter 8

Conclusion and outlook

This MSc thesis explores the adaptation of the Artificial Intelligence/Integrated Forecasting System (AIFS) for high-resolution limited area weather prediction. Addressing the limitations of existing models, we implement a framework following a similar approach as in [29], based on the AIFS model developed by ECMWF: a stretched hidden grid AIFS (SG-AIFS). This framework leverages a locally refined multi-mesh hidden grid to enhance information flow, especially for extended lead times. Our findings reveal several important insights: using global and regional ERA5 data, we observed that while increasing the hidden-grid resolution accelerates training time by expanding the parameter space, it has a marginal effect on the performance of the SG-AIFS model beyond a 90-hour lead time. However, finetuning the SG-AIFS model on longer rollout steps proved effective in reducing RMSE values across all lead times, with a notable improvement in global RMSE as lead time increases. Additionally, we extended the AIFS model processor to incorporate Transformer layers for prediction on a high-resolution regional domain, and compared its performance with that of the original GNN-based AIFS. Preliminary results indicate an pressure level imbalance in the Transformer model, degrading the performance of surface variables.

A high-resolution model was trained using the AIFS Graph-Transformer on 5 years of DOWA +3h reforecast data at a 2.5 km resolution over Western Europe, integrating global information from the ERA5 dataset at 1-degree resolution. This model demonstrates the ability to produce medium-range temperature forecasts at 2.5km resolution, effectively leveraging global information at 1 degree resolution for 6-hour lead times. For the 10-meter wind speed, the model provides 6h predictions with reasonable accuracy, although it lacks detailed features and longer lead times showed artefacts similar to those identified in [31]. Power spectrum analysis provided crucial insights into the SG-AIFS model's performance across different spatial scales for both 2-meter air temperature and 10-meter wind speed. Further analysis with DOWA regional data confirmed these findings, as significant smoothing effects became apparent for both the 2-meter temperature and 10-meter wind speed predictions. In contrast to global power spectra, where spectral power remained consistent even at extended lead times (+120h), regional power spectra revealed a marked loss across all wavelengths, reinforcing the limitations of deterministic models for capturing the high spatial variability of localized phenomena. This aligns with findings by [6] that deterministic data-driven models often smooth extreme values as a result of training to minimize the mean squared error (MSE).

Finally, the SG-AIFS model performance in predicting an extreme weather event was investigated. A case study was conducted, comparing the SG-AIFS model with the HARMONIE-AROME (HA) forecasts for a storm that occurred on the 22nd and 23rd of February 2017.

Although the SG-AIFS model predicted the storm's pressure and location with reasonable accuracy, the model consistently failed to capture finer-scale features. These findings demonstrate that while the SG-AIFS model shows promise in forecasting extreme weather phenomena, it currently does not match the operational accuracy of the HA model. Nonetheless, the potential of SG-AIFS and similar models for future development has become evident. As deep learning models for meteorology continue to evolve, the SG-AIFS framework could be of high relevance for the prospective operational usage of deep learning based meteorological models by national meteorological institutions.

For future research, numerous promising directions can be explored. Future work should focus on advancing the data integration of the model, exploring probabilistic forecasting approaches, and improving model evaluation. First and foremost, to elevate the SG-AIFS model to operational standards it is essential to improve the representation of fine-scale features. Having identified the three main contributors to the issue in Chapter 7, namely the model size, data size and the training regime, we recommend several directions of research. A straightforward approach to addressing the model size would be to re-train it with a reduced number of channels, which would also help confirm whether overfitting is indeed the underlying issue. Secondly, to increase the available data several possibilities arise. Leveraging transfer learning to incorporate additional ERA5 data could significantly improve results, as seen in [29]. Transfer learning would not only increase the number of usable years, but also allow us to incrementally incorporate different types of datasets. Possible candidates for incorporating additional data are the remaining DOWA years (2010-2012), the UWC-West reforecasting dataset (2km resolution; 2020-2023) and the CERRA dataset (Copernicus European Regional ReAnalysis, 1984-2021, 5.5km resolution [14]). Further study is required to investigate the impact of omitting trainable parameters necessary for transfer learning. Another experiment regarding transfer learning would be to investigate the influence of the temporal size of the regional dataset on the effectiveness of transfer learning. This investigation could help optimize the number of years required, thereby reducing the pre-processing effort needed for incorporating additional datasets across Europe. Transfer learning has the potential to resolve the issues related to the training regime, allowing us to transfer from a 1 degree resolution global grid to our current stretched-grid setup. Given additional computational resources, an increase in global resolution (as in [23] and [29]) or intermediate European datasets such as the CERRA dataset [14] could smoothen the transition and allow for more efficient information transfer within the model. Another limitation of this model is the lack of precipitation and poor humidity predictions. Including rain-gauge adjusted radar data [32] in the model could lead to high-quality precipitation forecasts. Ultimately, we believe that investigating the influence of transfer learning in the SG-AIFS model will be pivotal for improving this model.

Regarding the Transformer model, we recommend fine-tuning of the pressure level scaler to equalize performance across variables or even further prioritize surface variables. Other parameters of the SG-AIFS model that need to be tuned are the learning rate, the number of channels, and number of processor layers. The influence of rollout in the high-resolution model remains to be investigated, as well as the influence of eliminating trainable parameters. Since the number of rollout steps is limited by the size of the model, a trade-off has to be found between the benefit of increasing the number of parameters and the value of additional rollout steps. The rollout finetuning step could be made more efficient by eliminating the encoding and decoding in between steps. Additionally, the influence of taking larger rollout steps at once could be investigated.

To improve the data processing, more flexibility in the encoder and decoder would allow for extending the data sources and moving towards a more foundational weather model with fewer pre-processing steps. A potential solution could involve the use of separate encoder and decoder components specifically designed for different data sources, allowing the model to learn the transformations between variables instead of computing them manually. Alternatively, running AIFS on model levels, with ERA5 data retrieved at the model level, may improve prediction quality by incorporating level definitions that account for orography. Addressing these aspects would likely enhance the model's performance and accuracy in high-resolution forecasting applications. Alternatively, different imputation methods could be investigated to manage missing data effectively.

Ultimately, developing ensemble forecasting for DLWP is a crucial step to improve (extreme) weather prediction, mitigate blurring effects and providing the uncertainty information necessary for decision-making [33][7]. The results presented in this MSc thesis further highlight the need to move away from the Mean Squared Error loss for high-resolution, limited-area modeling. Probabilistic approaches, such as GenCast (a diffusion model proposed by Price et al. [33]), may provide a more accurate quantification of forecast uncertainty, offering a better representation of extreme events by capturing the full distribution of possible atmospheric states rather than predicting forecasts deterministically. In general, for future research diffusion models, variational auto-encoders and score-based uncertainty quantification are recommended to deal with the chaotic nature of the atmosphere. Such advancements will be essential for enhancing the reliability and accuracy of high-resolution weather forecasting models in operational contexts. Besides, alternative metrics could evaluate whether known physical properties are maintained. For regional analysis, Euclidean spectral analysis instead of spectral analysis based on spherical harmonics, might prove more suitable in terms of evaluation of smoothing. Proper confidence intervals would have to be computed, taking the temporal correlation of time series into account. Finally, future verification should consider the full testing set to provide an improved representation of forecast skill. The advancements described above will be imperative for the prospective operational deployment of deep learning-based meteorological models by national meteorological institutions, thereby enhancing the speed and precision of weather forecasting systems.

Bibliography

[1] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 6 2018. `doi:10.48550/arXiv.1806.01261`.

[2] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature 2015 525:7567*, 525:47–55, 9 2015. `doi:10.1038/NATURE14956`.

[3] Bill Bell, Hans Hersbach, Adrian Simmons, Paul Berrisford, Per Dahlgren, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Raluca Radu, Dinand Schepers, Cornel Soci, Sebastien Villaume, Jean Raymond Bidlot, Leo Haimberger, Jack Woollen, Carlo Buontempo, and Jean Noël Thépaut. The era5 global reanalysis: Preliminary extension to 1950. *Quarterly Journal of the Royal Meteorological Society*, 147:4186–4227, 10 2021. `doi:10.1002/QJ.4174`.

[4] Lisa Bengtsson, Ulf Andrae, Trygve Aspelien, Yurii Batrak, Javier Calvo, Wim de Rooy, Emily Gleeson, Bent Hansen-Sass, Mariken Homleid, Mariano Hortal, Karl Ivar Ivarsson, Geert Lenderink, Sami Niemelä, Kristian Pagh Nielsen, Jeanette Onvlee, Laura Rontu, Patrick Samuelsson, Daniel Santos Muñoz, Alvaro Subias, Sander Tijm, Velle Toll, Xiaohua Yang, and Morten Ødegaard Køltzow. The harmonie–arome model configuration in the aladin–hirlam nwp system. *Monthly Weather Review*, 145:1919–1935, 5 2017. `doi:10.1175/MWR-D-16-0417.1`.

[5] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast. 11 2022. `doi:10.48550/arXiv.2211.02556`.

[6] Massimo Bonavita. On some limitations of data-driven weather forecasting models. 9 2023. URL: `https://arxiv.org/abs/2309.08473v2`.

[7] Zied Ben Bouallègue, Mariana C A Clare, Linus Magnusson, Estibaliz Gascón, Michael Maier-Gerber, Martin Janoušek, Mark Rodwell, Florian Pinault, Jesper S Dramsch, Simon T K Lang, Baudouin Raoult, Florence Rabier, Matthieu Chevallier, Irina Sandu, Peter Dueben, Matthew Chantry, and Florian Pappenberger ECMWF. The rise of data-driven weather forecasting. 7 2023. `doi:10.48550/arXiv.2307.10128`.

[8] Kang Chen, Tao Han, Junchao Gong, Lei Bai, Fenghua Ling, Jing-Jia Luo, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, Yuanzheng Ci, Bin Li, Xiaokang Yang, Wanli Ouyang, Equal Contributions, and Project Lead. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. 4 2023. `doi:10.48550/arXiv.2304.02948`.

[9] Lei Chen, Xiaohui Zhong, Feng Zhang, Yuan Cheng, Yinghui Xu, Yuan Qi, and Hao Li. Fuxi: A cascade machine learning forecasting system for 15-day global weather forecast. *npj Climate and Atmospheric Science*, 6 2023. `doi:10.1038/s41612-023-00512-1`.

[10] F. A. Dahlen and Jeroen Tromp. *Theoretical global seismology.* Princeton University Press, Princeton, New Jersey, 1998.

[11] Paul Dando. Gaussian grids - forecast user - ecmwf confluence wiki. last updated: 04/04/2016. URL: `https://confluence.ecmwf.int/display/FCST/Gaussian+grids`.

[12] Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning.* Cambridge University Press, 2020.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR 2021 - 9th International Conference on Learning Representations*, 10 2020. `doi:10.48550/arXiv.2010.11929`.

[14] ECMWF. Copernicus regional reanalysis for europe (cerra) — copernicus. Accessed: 10/11/2024. URL: `https://climate.copernicus.eu/copernicus-regional-reanalysis-europe-cerra`.

[15] ECMWF. *IFS Documentation CY48R1 - Part III: Dynamics and Numerical Procedures*, pages 6–8. Number 3. ECMWF, 06 2023. `doi:10.21957/26f0ad3473`.

[16] Mohammad Hemmat Esfe, S. Ali Eftekhari, Maboud Hekmatifar, and Davood Toghraie. A well-trained artificial neural network for predicting the rheological behavior of mwcnt–al2o3 (30–70 *Scientific Reports 2021 11:1*, 11:1–11, 8 2021. URL: `https://www.nature.com/articles/s41598-021-96808-4`, `doi:10.1038/s41598-021-96808-4`.

[17] Hans Gleisner. Latitudinal binning and area-weighted averaging of irregularly distributed radio occultation data, 4 2011. URL: `https://rom-saf.eumetsat.int/rsr.php`.

[18] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, Cambridge, MA, USA, 2016. `http://www.deeplearningbook.org`.

[19] Guymonahan. Basic overviews on convolutional neural networks, Jul 2021. URL: `https://guymonahan.medium.com/basic-overviews-on-convolutional-neural-networks-f205180116be`.

[20] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean Noël Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146:1999–2049, 7 2020. `doi:10.1002/QJ.3803`.

[21] M. Hortal and Adrian Simmons. Use of reduced gaussian grids in spectral models. *ECMWF Technical Memoranda*, (168):31, 06 1990. `doi:10.21957/v413vy4fg`.

[22] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023. `doi:10.1126/science.adi2336`.

[23] S. Lang, M. Alexe, M. Chantry, J. Dramsch, F. Pinault, B. Raoult, Z. Ben Bouallègue, M. Clare, C. Lessig, L. Magnusson, and A. N. Prieto. Aifs: A new ecmwf forecasting system. *ECMWF Newsletter*, pages 4–5, 01 2024. `doi:10.21957/1a8466ec2f`.

[24] Simon Lang, Mihai Alexe, Matthew Chantry, Jesper Dramsch, Florian Pinault, Baudouin Raoult, Mariana C A Clare, Christian Lessig, Michael Maier-Gerber, Linus Magnusson, Zied Ben Bouallèggue, Ana Prieto, Nemesio Peter, D Dueben, Andrew Brown, Florian Pappenberger, and Florence Rabier. Aifs - ecmwf's data-driven forecasting system. 6 2024. `doi:10.48550/arXiv.2406.01465`.

[25] Mark G. Lawrence. The relationship between relative humidity and the dewpoint temperature in moist air: A simple conversion and applications. *Bulletin of the American Meteorological Society*, 86:225–234, 2 2005. `doi:10.1175/BAMS-86-2-225`.

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE International Conference on Computer Vision*, pages 9992–10002, 2021. `doi:10.1109/ICCV48922.2021.00986`.

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 8 2016. `doi:10.48550/arXiv.1608.03983`.

[28] Inneke Mayachita. Getting the intuition of graph neural networks, 5 2020. `https://medium.com/analytics-vidhya/getting-the-intuition-of-graph-neural-networks-a30a2c34280d` [Accessed: 12-02-2024].

[29] Thomas Nils Nipen, Håvard Homleid Haugen, Magnus Sikora Ingstad, Even Marius Nordhagen, Aram Farhad Shafiq Salihi, Paulina Tedesco, Ivar Ambjørn Seierstad, Jørn Kristiansen, Simon Lang, Mihai Alexe, Jesper Dramsch, Baudouin Raoult, Gert Mertes, and Matthew Chantry. Regional data-driven weather modeling with a global stretched-grid. 9 2024. `doi:10.48550/arXiv.2409.02891`.

[30] Leonardo Olivetti and Gabriele Messori. Advances and prospects of deep learning for medium-range extreme weather forecasting. *Geoscientific Model Development*, 17:2347–2358, 3 2024. `doi:10.5194/GMD-17-2347-2024`.

[31] Joel Oskarsson, Tomas Landelius, and Fredrik Lindsten. Graph-based neural weather prediction for limited area modeling. 9 2023. `doi:10.48550/arXiv.2309.17370`.

[32] A. Overeem, T. A. Buishand, I. Holleman, and R. Uijlenhoet. Extreme value modeling of areal rainfall from weather radar. *Water Resources Research*, 46:W09514, 2010. URL: `https://research.wur.nl/en/publications/extreme-value-modeling-of-areal-rainfall-from-weather-radar`, `doi:10.1029/2009WR008517`.

[33] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, Matthew Willson, and Google Deepmind. Gencast: Diffusion-based ensemble forecasting for medium-range weather. 12 2023. URL: `https://arxiv.org/abs/2312.15796v2`.

[34] Xiaoli Ren, Xiaoyong Li, Kaijun Ren, Junqiang Song, Zichen Xu, Kefeng Deng, and Xiang Wang. Deep learning-based weather prediction: A survey. *Big Data Research*, 23:100178, 2 2021. `doi:10.1016/J.BDR.2020.100178`.

[35] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20:61–80, 1 2009. `doi:10.1109/TNN.2008.2005605`.

[36] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *IJCAI International Joint Conference on Artificial Intelligence*, 2:1548–1554, 8 2021. URL: `https://www.ijcai.org/proceedings/2021/214`, `doi:10.24963/IJCAI.2021/214`.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-December:5999–6009, 6 2017. `doi:10.48550/arXiv.1706.03762`.

[38] Sluijter KNMI Weer-en Klimaatdiensten. Knmi - zeer zware windstoten 23 februari 2017, 2 2017. Accessed: 10/11/2024. URL: `https://www.knmi.nl/kennis-en-datacentrum/achtergrond/zeer-zware-windstoten-23-februari-2017`.

[39] L Wijnant, B Van Ulft, B Van Stratum, J Barkmeijer, J Onvlee, C De Valk, S Knoop, S Kok, G J Marseille, H Klein Baltink, and A Stepek. The dutch offshore wind atlas (dowa): description of the dataset. Technical report, KNMI, 2019. URL: `https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://cdn.knmi.nl/knmi/pdf/bibliotheek/knmipubTR/TR380.pdf&ved=2ahUKEwjB3ZK3-taJAxX-xQIHHbVxFWYQFnoECBkQAQ&usg=AOvVaw0eQSYcHU2HLgLbOlpk5mOR`.

[40] Wanghan Xu, Kang Chen, Tao Han, Hao Chen, Wanli Ouyang, and Lei Bai. Extremecast: Boosting extreme value prediction for global weather forecast. 2 2024. `doi:10.48550/arXiv.2402.01295`.

Appendix A

Transformer models

This appendix provides some additional information on Transformer models, also referred to as Transformers. These represent a second class of data-driven models used in data-driven medium-range global weather forecasting. Originally introduced by Vaswani et al. [37], Transformers serve as an alternative to recurrent models for machine translation, to pass important information over a longer range. With their Vision Transformer (ViT), Dosovitskiy et al. extended the Transformers framework to image recognition tasks. ViTs have shown state-of-the-art results while being more computationally efficient than CNNs [13]. However, they do not scale well with image resolution, giving rise to an adapted model called Swin Transformer. Swin Transformer uses shifted windows in each Self-Attention layer to partition the image into smaller segments with the same parameters. This model achieves faster results for the training and prediction of higher-resolution images [26].

A.1 Mathematical framework

The basis of Transformer models is the Self-Attention operation (also called *qkv self-attention*). The operation consists of three weight matrices corresponding to the so-called *queries*, *keys* and *values* (W^Q, W^K and W^v), such that $q = W^Q x$, $k = W^K x$ and $v = W^V x$. Then, the *attention matrix* $A \in \mathbb{R}^{N \times N}$ is computed as follows:

$$A = SoftMax(qk^T/\sqrt{D}) \tag{A.1}$$

where D is the matrix containing the query/key feature dimensionality. The attention matrix A aims to capture the relationships between the components of a given input sequence $x \in \mathbb{R}^N$. It consists of weights that should be learned, where each element $a_{i,j} \in A$ corresponds to the similarity between a query q_i and a key k_j. The softmax operation normalizes the matrix into a probability distribution, with each row computed as follows:

$$A_i = \frac{e^{q_i k_i/\sqrt{d_i}}}{\sum_j e^{q_j k_j/\sqrt{d_j}}} \tag{A.2}$$

where d_j corresponds to the dimensionality of the key vectors. Based on this matrix, the output sequence is $\mathit{Attention}(x) = Av$, where v is called the value vector. In Swin Transformers, a relative positional bias B is added, which learns the distances between the patches as extra parameters in the model. This results in the final definition:

$$\mathit{Attention}(q, k, v) = SoftMax(qk^t/\sqrt{D} + B)v \tag{A.3}$$

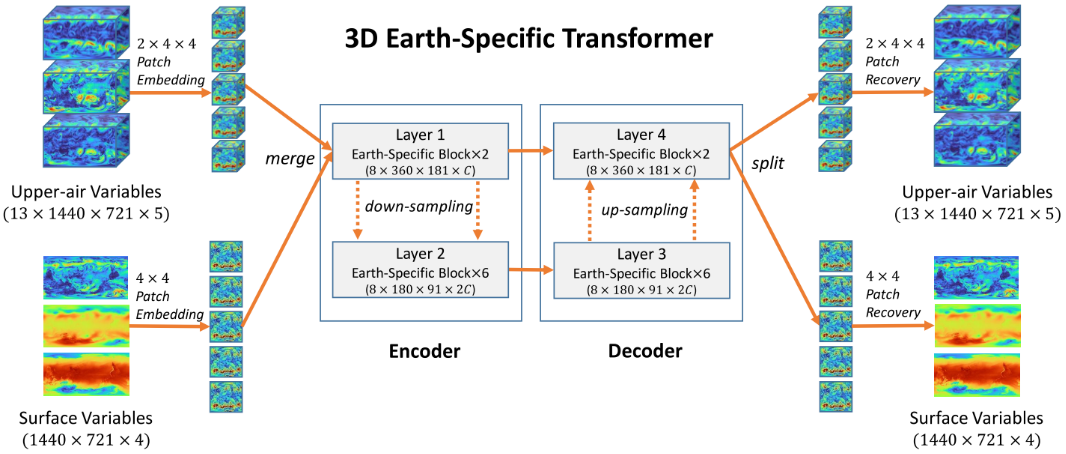


Figure A.1: PanguWeather: An overview of the 3D Earth-specific transformer architecture. Based on the standard encode-decoder design, we (i) adjust the shifted-window mechanism and (ii) apply an Earth-specific positional bias. Figure taken from [5].

A.2 Pangu-Weather

Pangu-Weather was developed by Bi et al. [5], and is considered to be the first deep learning based model that outperformed state of the art medium-range global deterministic NWP methods in terms of accuracy. Its novelty lies in training a Swin Transformer with a 3D Earth-specific bias, and using a combination of different forecast lead times to mitigate the effect of cumulative forecasting errors.

Pangu-Weather uses an encoder-decoder architecture, similar to GraphCast. However, the encoder and decoder consist of a simple downsampling and upsampling in resolution using a Vision Transformer block specifically adjusted to match the Earth's geometry. Moreover, instead of having a linear or textual output as in classical (Vision) Transformers, the model is *generative*, such that the information is decoded back into the same shape. The model architecture of Pangu-Weather is visualized in Figure A.1.

The Earth-specific positional bias B_{ESP} adds a positional bias based on the absolute coordinate of each window. The bias matrix contains of submatrices corresponding to the specific latitudes and pressure levels (the longitude is assumed to have the same bias). When the attention is computed between two units in the same window, the bias can be found in the three-dimensional submatrix of that window. It contains the bias of the intra-window coordinates (h_1, ϕ_1, λ_1) and (h_2, ϕ_2, λ_2) at position $(h_1 + h_2 \times W_{pl}, \phi_1 + \phi_2 \times W_{lat}, \lambda_1 - \lambda_2 + W_{lon} - 1)$, where $W_{pl} \times W_{lat} \times W_{lon}$ would be the size of the window. Even though the use of this prior increases the number of parameters, the performance remains unaffected.

A.2.1 Training details

The authors of Pangu-Weather introduce hierarchical temporal aggregation, where four different models are trained for different lead times (1 hours, 3 hours, 6 hours and 24 hours) and predictions are made for a specific hour using the least amount of aggregated predictions. The four models are trained individually on 192 NVIDIA Tesla-V100 GPUs for 16 days. The training/testing data used is similar to GraphCast (0.25 degree resolution), but the pressure levels are downsampled to 13.

A.2.2 Results

The authors of Pangu-Weather do not compare their model to HRES, but rather to an individual ensemble member of the operational ECMWF ENS by using the model output, started from the unperturbed initial conditions. Pangu-Weather shows consistent improvement over the operational IFS in all forecast times and all variables, with increasing improvement over longer lead times.

A.3 FuXi

FuXi is a Transformer based model proposed by Chen et al. [9]. With their model, they not only extended the prediction lead time to 15 days, but also introduced an ensemble version of their model to provide probabilistic forecasts. The basis model consists of a convolutional 3D embedding, which splits the data into vectorized blocks. Besides, they propose a Swin Transformer V2, by using post-normalization and changing the original self-attention to a scaled cosine attention:

$$Attention(q, k, v) = (\cos(q, k)/\tau + B)v, \tag{A.4}$$

where τ is a learnable scalar, which is not shared across heads and layers. The choice of the cosine attention is motivated by the natural normalization that this function provides. Otherwise, the model again has an encoder-decoder structure, using 48 Transformer blocks (with skip connections) during processing.

They implemented these adaptations (the convolutional embedding and the Swin Transformer v2) and train three different models: FuXi-Short optimizing for 6 hour forecasts, FuXi-Medium for 5-10 days, and FuXi-Long for 10-15 days. The models are cascaded, similarly to PanguWeather, to produce the final forecast. Additionally, ensemble forecasts are generated from the cascaded models, by introducing random noise to perturb the initial conditions and introducing Monte Carlo dropout during inference to perturb the model parameters.

The models are trained on 8 Nvidia A100 GPUs for 30 hours. The training data consists of ERA5 data as similar to AIFS (13 pressure levels, 0.25 degree resolution). The results show comparable performance of the ensemble mean to the ECMWF ENS mean in 15-day forecasts at a 6 hour temporal resolution. FuXi is the first data-driven model reaching similar performance to the ECMWF ENS mean, which is valuable in some applications.

A.4 FengWu

In April 2023, Chen et al. [8] introduced FengWu, which treats the problem from a multi-task perspective, since the general problem objective is to predict many variables simultaneously. The weather states are separated to extract features independently during the encoding, and only fuses them later during processing. Separate decoders are defined for each feature to predict the output weather state. They also adapt the loss function to allow for multi-task learning, by defining a Gaussian probablistic model for each predictand (variable and grid cell), minimizing the maximum likelihood.

Furthermore, FengWu uses a *replay buffer*, to replay data from earlier predictions (e.g. used as input). This buffer consists of the last N predictions, which are sampled with a certain probability, resulting in better results for longer lead times. This is an alternative to the hierarchical temporal aggregation used by Pangu-Weather or the autoregressive training stage used by GraphCast.

FengWu exceeds GraphCast in performance for 80% of the prediction variables, as well as extending prediction to longer lead times (10.75 days). The authors attribute these improvements to the adaptations described above.

Appendix B

List of testing dates

Table B.1: Testing dates for the tuning experiments and high-resolution SG-AIFS model trained on 5 years of ERA5 and DOWA data described in Chapter 6. The dates are selected to ensure evenly distributed evaluation across the year. For each date 4 runs have been performed, starting at 00, 06, 12 and 18UTC.

ERA5 experiments
2022-01-02
2022-01-22
2022-02-11
2022-03-03
2022-03-23
2022-04-12
2022-05-02
2022-05-22
2022-06-11
2022-07-01
2022-07-22
2022-08-10
2022-08-30
2022-09-19
2022-10-09
2022-10-29
2022-11-18
2022-12-08

High-resolution model
2017-01-01
2017-01-21
2017-02-10
2017-03-02
2017-03-23
2017-04-11
2017-05-01
2017-05-21
2017-06-10
2017-06-30
2017-07-20
2017-08-09
2017-08-29
2017-09-18
2017-10-08
2017-10-28
2017-11-17
2017-12-07

Appendix C

Additional results

C.1 SG-AIFS additional results

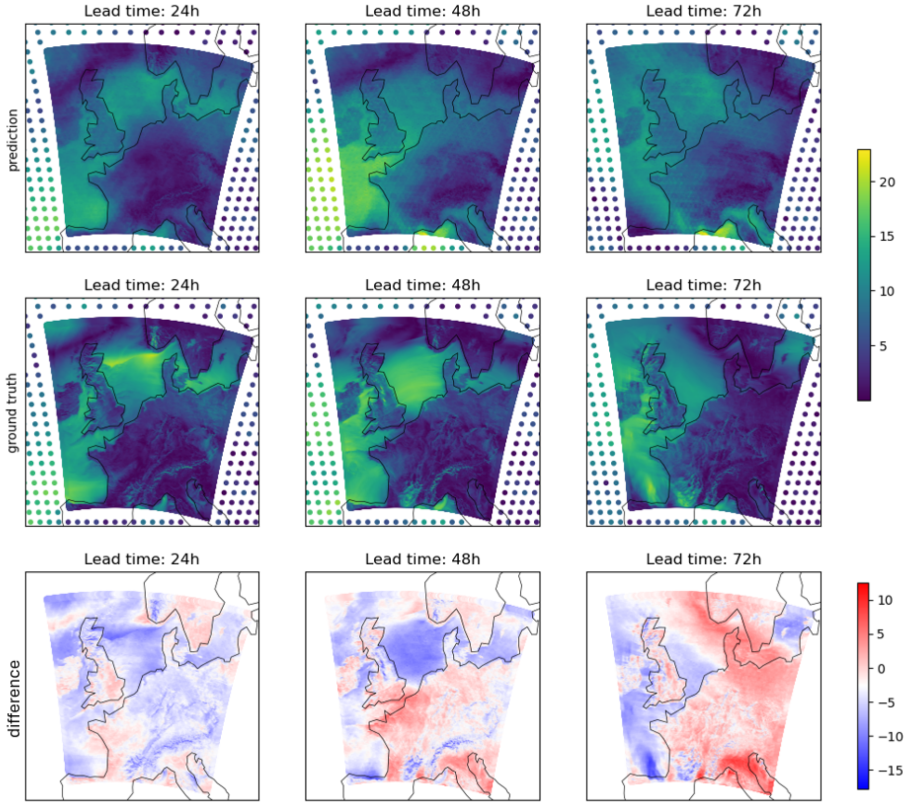


Figure C.1: SG-AIFS with hidden grid resolution 5 (9) trained on ERA5 globally and DOWA locally initialized from 2017-02-10T06. The first and second row of the figure show the SG-AIFS forecast and ground truth, respectively, of the wind speed (m/s) at 10 meter height over western Europe. The bottom row shows the wind speed difference between the ground truth and the forecast. The model displays smoothing and hidden grid artefacts on the regional domain, with wind speed differences of up to 10 m/s.

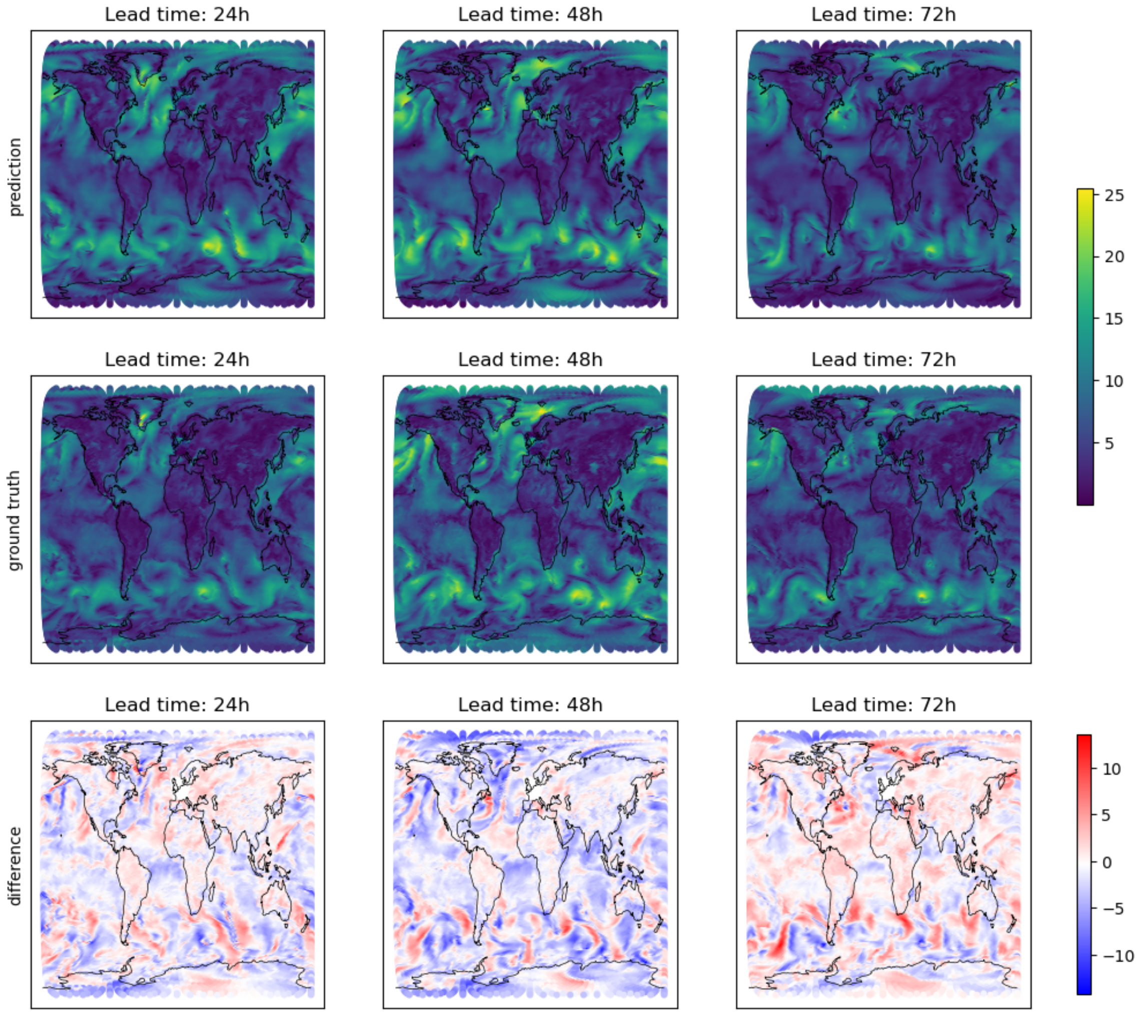


Figure C.2: SG-AIFS with hidden grid resolution 5 (9) trained on ERA5 globally and DOWA locally, initialized from 2017-02-10T06. The first and second row of the figure show the SG-AIFS prediction and ground truth, respectively, of the global wind speed (m/s) at 10 meter height. The bottom row shows the wind speed difference between the ground truth and the forecast. The model captures the intricate global patterns well, although longer lead times display smoothing.

C.2 Transformer additional results

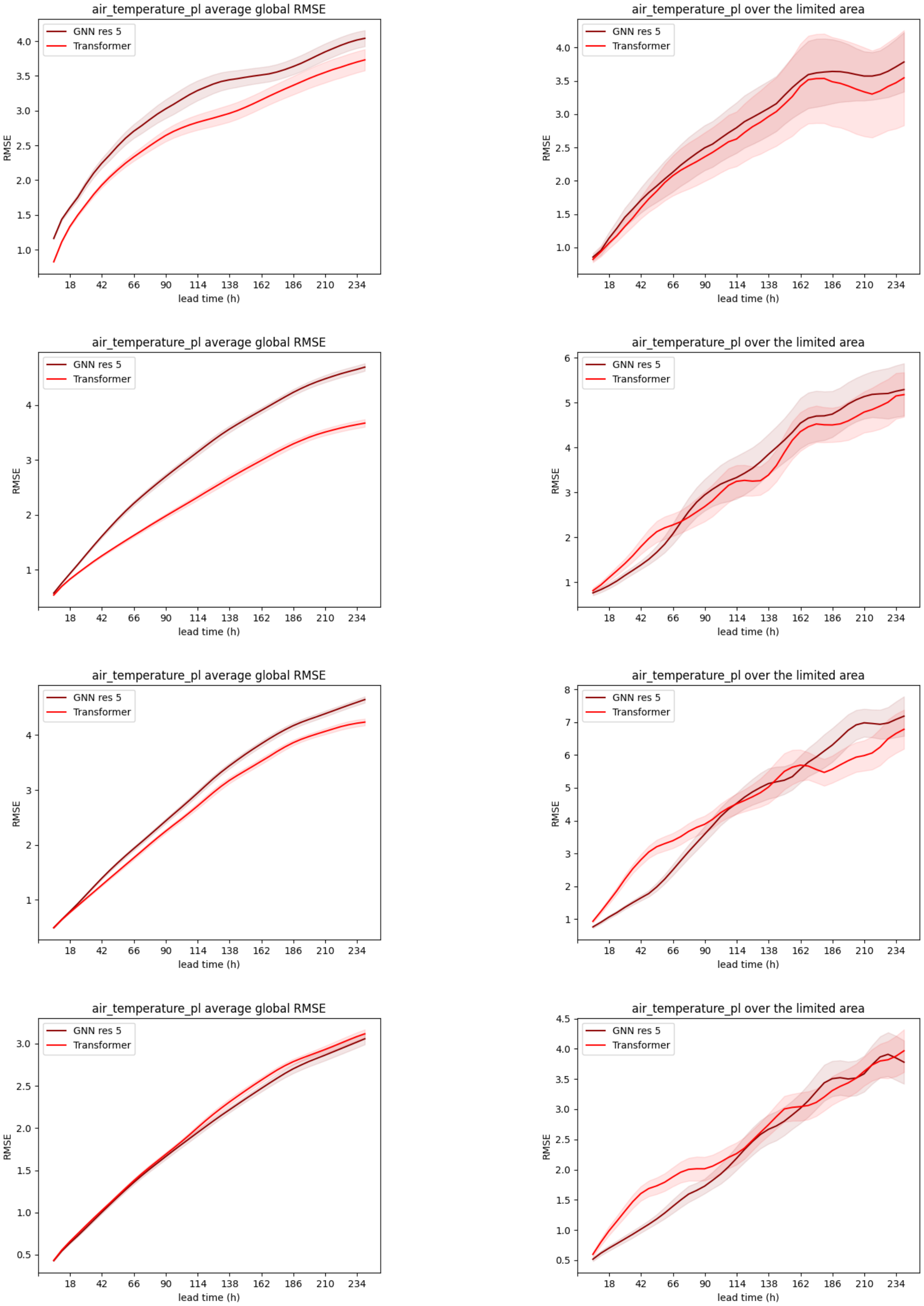


Figure C.3: SG-AIFS GNN processor compared to the Transformer processor, averaged RMSE for the air temperature at 50, 150, 200 and 300 hPa, evaluated on 18x4 cases of the test set (Appendix B) for lead times up to 10 days. The colored bands represent the 95% confidence intervals. The Transformer model shows decreased performance compared to the GNN for higher pressure levels.

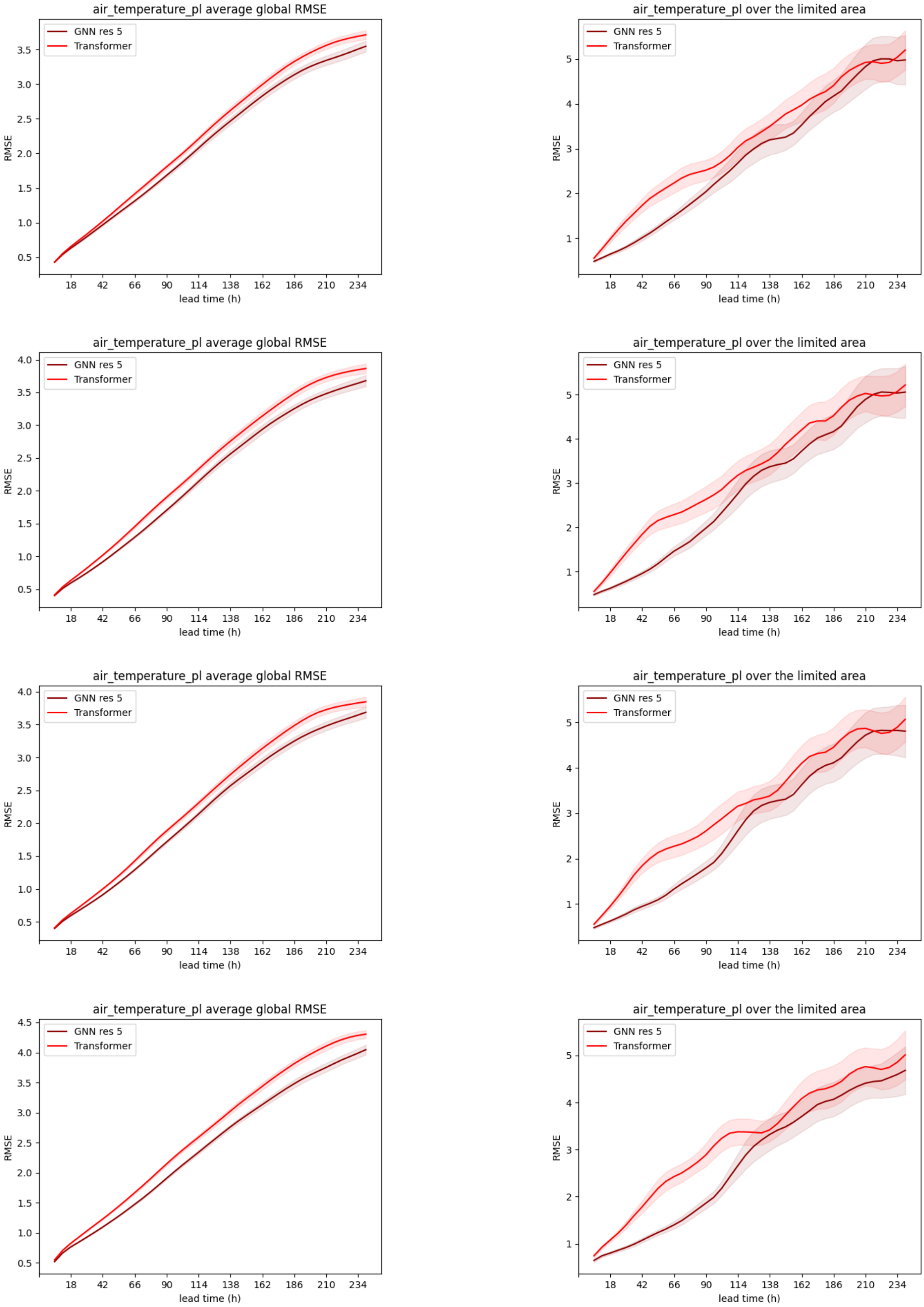


Figure C.4: SG-AIFS GNN processor compared to the Transformer processor, averaged RMSE for the air temperature at 400, 500, 700 and 850 hPa, evaluated on 8x4 cases of the test set (Appendix B) for lead times up to 10 days. The colored bands represent the 95% confidence intervals. The Transformer model shows decreased performance compared to the GNN for higher pressure levels.