



**Documenting and improving machine learning-based weather prediction techniques for
medium range weather forecasts**

Jacob Davies

Professor Jizheng Wan (supervisor)

School of Computer Science
University of Birmingham

Abstract

MLWP (Machine Learning Weather Prediction) has very recently proven to be more accurate than traditional NWP (Numerical Weather Prediction) in medium-range weather forecasting and has increased the lead time for accurate weather forecasts; moreover the cost of inference is several orders of magnitude cheaper than traditional methods. However, major weather services are yet to field a homegrown system, and are struggling to develop a MLWP system of their own.

Despite the fact that private companies like Google and Huawei have been able to create and improve new models; the Met-Office, ECMWF, NOAA, and other weather services have not been able to produce a model that is remotely comparable despite the fact that there is a huge monetary incentive for them to do so. There are huge problems in developing a new model stemming heavily from the lack of information on how different models have been developed.

This paper evaluates how similar systems to those used in landmark papers can be implemented. Reviewed training parameters and how they are best optimised, and compared different models which have the same amount of optimisation. Using these findings this paper offers a conclusive comparison of MLWP models, and gives actionable evidenced suggestions that pave the way for researchers to create future models by implementing genuine improvements.

Acknowledgements

Firstly I would like to thank my supervisor, professor Jizheng Wan, for the time and guidance he has dedicated to this project. Our weekly meetings and all the feedback to get my work into a professional document made this project possible and guided every part without exception. I would not have been able to do this without him.

I'd also like to thank professor Lee Chapman for his willingness to meet with me and the help in understanding the meteorology side of this project. His support was invaluable to this project.

I'd also like to thank the University of Birmingham for access to BEAR, it made this project possible and it has been put to great use.

Abstract.....	2
Acknowledgements.....	3
1 Introduction.....	5
2 Background.....	6
2.1 Forecast Types.....	6
2.2 System Analysis.....	7
2.3 How Predictions Are Presented.....	8
2.4 The effects of MLWP.....	10
3 Literature Review.....	11
3.1 Evaluating the quality of weather predictions.....	11
3.2 FourCastNet.....	17
3.3 Pangu-Weather.....	18
3.4 GraphCast.....	21
3.5 GenCast.....	25
3.6 Convolutional neural networks.....	28
3.7 Attention based transformer systems.....	30
4 Methodology.....	32
4.1 Data and pre-processing.....	32
4.2 Models and training.....	33
4.3 Testing.....	34
5 Implementation.....	35
5.1 Traditional NN model.....	35
5.2 Simple CNN Implementation.....	40
5.3 Patch Embedding Transformer Based.....	45
5.4 Graphical Neural Network.....	49
6 Professional, legal, social, and ethical Issues.....	55
6.1 Professional and legal issues.....	55
6.2 Social and Ethical issues.....	55
7 Evaluation.....	57
7.1 Results comparison.....	57
7.2 Results and recommendations.....	60
8 Conclusion.....	61
Research proposal.....	61
References.....	62
Appendices.....	67
A.....	67
B.....	69
C.....	71
Figures and Tables.....	72

fl Introduction

Millions of people every year die from weather related causes [33], millions more lives are saved because of planning and measures put in place to protect them from extreme weather. Any measures to protect people are reliant on timely and accurate weather predictions.

Fortunately weather prediction has improved dramatically over the last 75 years [27] and consequently the number of high death toll weather events has reduced drastically [34]. Unfortunately improving the accuracy of weather predictions is immensely difficult. Better models, dramatically increased data collection, hardware improvements, and drastically increased compute spending now only offers marginal improvements.

MLWP offers a new approach, allowing improvements in accuracy with a drastic decrease in computing power; weather agencies and meteorologists want to be able to use MLWP to improve predictions further, but have yet to produce or improve models.

The problem: it is very difficult to make an informed decision on how to go forward with new developments, or give a research proposal with good reasoning about how said research will result in improvements. This is due to the numerous models, different amounts of training for each model, lack of documentation on what is important to tune, and no guidance on how to improve these models.

The solution to this problem is to evaluate on equal standing the performance of different model architectures. Then for each model evaluate which parameters are important for the quality of the results, and therefore how the model could be improved.

This paper achieves this by:

- Reviewing the background information needed to understand the basics of weather prediction.
- Reviewing literature on the meteorological principles of weather prediction, current MLPW papers and models, and the effective implementation of ML architectures.
- Defining a methodology for making, training, and testing MLWP models.
- Implementation and testing of several different MLWP architectures.
- Evaluating the results from the tests of the MLWP architectures, and what that tells us.
- Concluding with a suggestion for the direction of future research.

2 Background

While weather forecasts are something used by almost everyone, the inner workings are not widely understood. This section gives an introduction to how forecasts work, and the acronyms and technical vocabulary used while discussing weather forecasting.

2.1 Forecast Types

This section explains briefly how the global forecasting system works and the steps that go into creating a weather forecast. There are many subcategories and crossover between forecast types and there are big differences between different forecasts within categories, but broadly they are categorized as shown below. While the categories are typically used for certain ranges of lead time, they are defined not by how far into the future they look but how they work and what the aim of the model is. What are the different types of forecast?

Short-range

Forecasts of typically less than 2 or 3 days are known as short-range forecasts. They are categorised by not needing to simulate the entire earth, only simulating an area local to the forecast area. They typically focus on surface-level weather and things like precipitation and precipitation type. They typically have a high resolution both temporarily and positionally. They also focus a lot on data processing of the predictions and ‘forecast communication’.

Medium-range

Medium-range forecasts are typically 3-10 days, characterised by being ‘global models’ needing to simulate the whole globe. While they have improved significantly and continue to do so they struggle to accurately predict weather beyond 7 days. They are typically updated every 6-12 hours depending on lead time. [35]

Long-range

Long-range forecasts are probabilistic forecasts of longer than 10 days. Precise predictions cannot be accurately made at this time scale so they focus heavily on the level of uncertainty and make broad predictions. They often focus on oceanographic predictions as they are more consistent and can be predicted with a very long lead time.

Other Forecasts

There are also forecasts for oceanographic changes and space weather, non-atmospheric forecasts are outside the scope of this research.

2.2 System Analysis

While no two systems are identical, there are strong similarities between them. The system used by the ECMWF's HRES is detailed below, other systems are very similar.

Gather data

Firstly data is gathered from across the world from a variety of sources: satellites, aircraft, ships, buoys, weather balloons, weather radar, atmospheric radar, dropsnodes and even GPS, among other systems. All this data needs to be: measured, collected, and sent to a central location.

Assimilate the data

To make predictions models require a complete image of the world at a fixed point in time as a starting point. It is not feasible to measure every point on Earth at several dozen altitudes simultaneously. So, the incredibly difficult job of data assimilation is to take this flawed, limited data and output a complete image of the Earth. (See Appendix A 1 for a more detailed explanation).

Prediction

From this initial starting condition, a future prediction of the weather is made. The weather model takes in the starting state and does mathematical calculations to produce the predicted state in one time step. It then takes the output state and uses it as an input for the next time step, repeating until it has predicted the desired lead time. (See Appendix A 2 for a more detailed explanation).

Post-processing

Post-processing takes in the final weather forecast and applies data processing techniques (a small amount of statistics and, or machine learning) to bias correct. This allows users to quickly figure out what they want to know i.e. 'What's the chance it rains here on Thursday?'.

2.3 How Predictions Are Presented

When weather forecasts are presented to an end user (on a mobile app or television), the information that the end user cares about is given in a ‘user-friendly’ form (such as ‘the likelihood of rain’ or ‘the temperature’. However, the results given in papers and model outputs are less intelligible: “T850 = 283.65” for example, what does this mean?

Altitudes and Pressure Levels

Until very recently there was no cheap way of getting accurate altitude measurements. These either required an expensive GPS receiver or a radar altimeter. Therefore to reliably measure the atmospheric conditions weather balloons and aircraft would use the outside pressure measurement. This data can be gathered easily: look at a pressure reading of the desired level, and use the other measurements taken at the same time.

Pressure is usually measured in hPa (hectopascals) or in millibars; these are the same units. There are several different altitudes important to weather prediction: sea-level boundary level and steering level. Sea-level pressure is typically a little over 1000 hPa. 850 hPa slightly above the ‘boundary layer’ and is the lowest level where the temperature is practically unaffected by the day-night cycle. 500hPa is known as the steering level, and the winds at this altitude are usually the same direction and speed as the underlying weather systems. (The University of Birmingham is usually around 985 hPa [32])

Measurements and Their Units

There are only a handful of measurements and units used by MLWP models. Here is a quick overview:

location	Term	Unit	Explanation
Surface	T2M	K (Kelvin)	The surface air temperature: the air temperature at 2m above the ground in the shade measured in Kelvin.
	U10M	MS ⁻¹	The eastward component of wind at 10m: eastward wind speed 10m off the ground in meters per second, if the wind is going west this is negative.
	V10M	MS ⁻¹	The northward component of wind at 10m: northward wind speed 10m off the ground in meters per second, if the wind is going south this is negative.

	SP	Pa	Surface pressure: the pressure on the surface in pascals
	MSLP	Pa	Mean sea level pressure: the pressure at sea level
Atmospheric	T	K	Temperature: air temperature in Kelvin
	U	MS^{-1}	The eastward component of wind: eastward wind speed in meters per second, if the wind is going west this is negative.
	V	MS^{-1}	The northward component of wind: northward wind speed in meters per second; if the wind is going south, this is negative
	Q	gKg^{-1} or KgKg^{-1}	Specific humidity: The amount of moisture in 1 kg of air. Measured in grams of water per Kilogram of air or Kilograms of water per kilogram of air
	RH	Proportion	Relative humidity: the proportion of water in the air compared to the maximum amount of water the air could carry given its temperature and pressure. Air can be ‘supersaturated’ and have a humidity above 100%
	Z	M^2S^{-2}	Geopotential: The gravitational potential energy of a unit mass at a given height compared to sea level. Geopotential height (height above sea level) is geopotential divided by g (9.80665 ms^{-2}) This can be a negative value if the sea level pressure is less than the pressure of the level being measured. In a system where levels are fixed at a pressure, the height of the level is an analogue for pressure.

(Table 1, [35]): weather forecast measurements and their units

2.4 The effects of MLWP

A detailed analysis of MLWP and its effects is given in the literature review. However a brief introduction here will help to show the importance and reasoning behind this analysis.

Weather prediction has been an obvious target for machine learning. It has a huge amount of well-labelled, free data that has no gaps, and there are no problems with data protection due to its dispersonal nature. Despite this, up until late 2022 it was pessimistically predicted to be unviable until the early 2030's [9].

In 2022 Huawei came out with an MLWP model that had similar performance to one of the modern NWP models used, but required 99.999% less computing power [1]. Since then Google has shown further improvements with their latest model being better than NWP models, with much lower computational demands [6].

MLWP has only changed the prediction step for medium-term weather prediction, becoming a component that can be used to produce the medium-term weather predictions that allow prediction of storms and other large weather events.

MLWP is unlikely to fully replace NWP for many reasons. The primary issue being that weather prediction is an incredibly important problem that affects people's lives, and costs hundreds of billions of dollars each year. Weather services, and other actors that are informing their decisions based on weather forecasts, have a high threshold for reliability, and require a long track record of reliability before changing how they are informed.

3 Literature Review

This literature review analyzes and explains the information needed to create MLWP models, it is in 3 parts:

- Meteorological principles of weather prediction. How are weather models evaluated and why are the tests scientifically valid? (3.1)
- Current MLPW papers and models. From the papers that have been released- what developments have been made, why they are significant, and what is the performance? (3.2 - 3.5)
- Effective implementation of ML architectures. What do different ML architectures do, how do they work, and what makes them work well? (3.6 - 3.7)

It is important to consider not only academic papers and the well documented and well understood facts, but also the opinions of academics and researchers who are knowledgeable in this field and who have worked on these papers. By looking at what is said informally it is possible to gain a fuller picture, especially when there is little or no information on a given topic.

The following overviews are built on academic papers and informal commentary from researchers, offering a more complete understanding of the decisions made in papers that could not be achieved through analysis of the papers alone.

3.1 Evaluating the quality of weather predictions

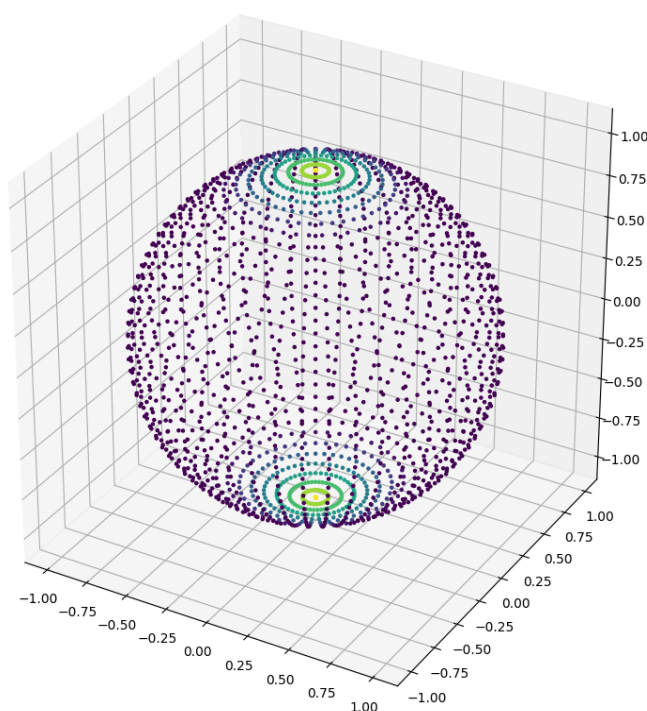
In order to improve weather prediction models and compare them to one another, a system of metrics is needed to evaluate the quality of weather predictions. Such a system would allow us to quantify the quality of the predictions made by the model. Due to the importance of this task, well established, widely used, universal systems have been chosen. Although these metrics are widely used, it is not inherently obvious why they are relevant. The following explanations of quality measures are all given from the perspective of medium-range forecasts, other forecasts have other objectives and as such utilize a different array of tests. Professor Lee Chapman a Professor of Climate Resilience & Met Office Joint Chair at the University of Birmingham has been extremely helpful in helping me navigate this, and my hope is that the following explanations are explanatory and comprehensive.

The primary way that model quality (referred to as ‘skill’) is measured is through a latitude-weighted measurement of ACC and RMSE of a subset of parameters from the prediction. This gives numerical values that show to what extent a model's predictions are better than others. The purpose of these parameters will be discussed later.

Latitude-weighting

To get a fair RMSE or ACC value, adjustment for latitude is required. As latitudes move towards the poles, the distance between lines on a longitudinal grid decreases and becomes more dense; as such, we have to weight the values depending on latitude; otherwise, performance changes in polar regions would be dramatically over-represented. To adjust, the cosine of the latitude (in radians) is weighted.

$$P = \frac{\sum_{ij} P_{ij} \cdot \cos(\phi_i)}{\cos(\phi_i)}$$



(Figure 1, original): The diagram shows points at 2.5° longitude and latitude on a sphere with colour representing distance to the next closest point. (assumes perfectly spherical). Figure made by me and taken from “/FigureMaker/points_on_sphere.ipynb”

Parameter choice

When considering parameters to compare, there are two that are used universally by weather services, NGO's, businesses, and academics in all comparisons: Z500 and T850. But why these measurements at these levels?

Z500

Z500 is the geopotential height of the point of the atmosphere where the pressure is 500hPa, but why geopotential and why at this level?

Why geopotential?

Geopotential height is proportional to height above sea level. By having a measurement system grounded on pressure levels, the geopotential height is a derivative of pressure in a region, as pressure lowers, the point in the atmosphere of a given pressure level moves closer to sea level, and vice versa. The pressure gradient of the atmosphere plays an extremely important role in atmospheric dynamics [36] and is the key driver in storm dynamics and the movement of weather systems. Measuring other factors like wind speed is problematic; they are derivatives of the underlying causes of weather, and they are unstable. This might give positive feedback on models that predict outputs that are more similar to the mean - a problem since the most important weather happens in the most extreme conditions [37].

Why 500hPa?

500hPa is the middle of the troposphere (the lowest layer of the atmosphere where weather systems that affect the surface happen); as such, it is a good level for the measurement of weather systems. It's also at a level where large-scale weather systems, which have the largest effect on our lives (cyclones, storms, jet streams), are well developed. 500hPa is often referred to as the "steering level" [19]; this is because weather systems move in the same direction as the wind at this level.

The reason we don't use a higher or lower altitude is that, higher altitudes are less dense and, as such, have much faster winds that are less correlated with the direction of weather patterns and are less changeable; lower altitudes are more heavily affected by the surface and surface conditions, slowing wind speeds and changing characteristics with the diurnal cycle.

T850

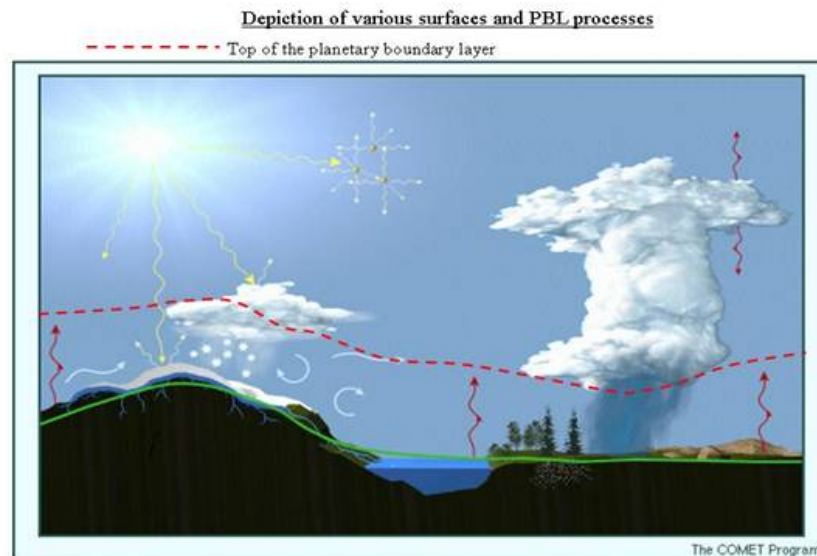
Why temperature?

The air temperature at 850 hPa is important for accurate differentiation between different air masses so that warm and cold fronts can be identified. It's important to know the temperature of a weather system near the surface, as the behaviour of weather systems is often temperature-dependent.

Why 850hPa?

The problem with temperature is that near the surface it is affected by diurnal solar conditions more than atmospheric conditions [378]. 850hPa is just above the ABL (atmospheric boundary layer, see diagram below) and in the middle of near-surface weather phenomena. It is high

enough that it is not majorly adversely affected by diurnal conditions but low enough to capture atmospheric conditions, making it an important measurement for identifying different air masses.



(Figure 2, [20]):Diagram of the atmospheric boundary layer. The red dotted line shows the altitude of the atmospheric boundary, red arrows show heat radiation, and yellow arrows show solar effects. Source: Pennsylvania State University, College of Earth and Mineral Sciences [20].

Anomaly Correlation Coefficient?

ACC, Anomaly Correlation Coefficient, is a measurement of the skill of a weather forecast by comparing the difference between forecast, observation, and climatological mean. For example if the weather is unexpectedly 7° warmer than usual, the ACC would measure the ability of models to predict such an anomaly. As such an ACC of 1 would be measured from predicting 7°, an ACC of 0 would be measured from predicting the average, and an ACC <0 would result from predicting a lower temperature.

The equation for ACC:

F= forecast, O = observation, C = climatological mean, F' = forecast anomaly = F - C, O' = observational anomaly = O - C.

$$ACC = \frac{\Sigma (F' \cdot O')}{\sqrt{\Sigma (F'^2) \cdot \Sigma (O'^2)}}$$

ACC is essentially a Person correlation, while it would be possible to combine ACC of different measured values to get a total ACC, this is not useful as different parameters are important for different applications and some variables are more easily predicted.

The correct selection of climatological mean is debated, as the climate is changing more and more rapidly it becomes more difficult to produce a climatological mean that is representative of current expected values.

Secondary measurements of quality

In some papers, secondary measurements of quality are used to compare two models. Examples of this are: storm tracking [1, 2, 6] and target lists [2, 6]. However, in almost all cases, these secondary measurements produce results that have been an inaccurate representation of the quality of the predictions or in some cases have suggested that a model is better while in reality it was not. There is a reason that these measurements are not used by traditional organizations, and have not been adopted outside their original papers. The problematic use of these measurements has been the primary source of criticism of the development of new MLWP models in Google and Huawei's papers.

These measurements are inadequate and problematic, the criticisms that academics and professionals have levied against them are documented below.

Suggestions of foul play

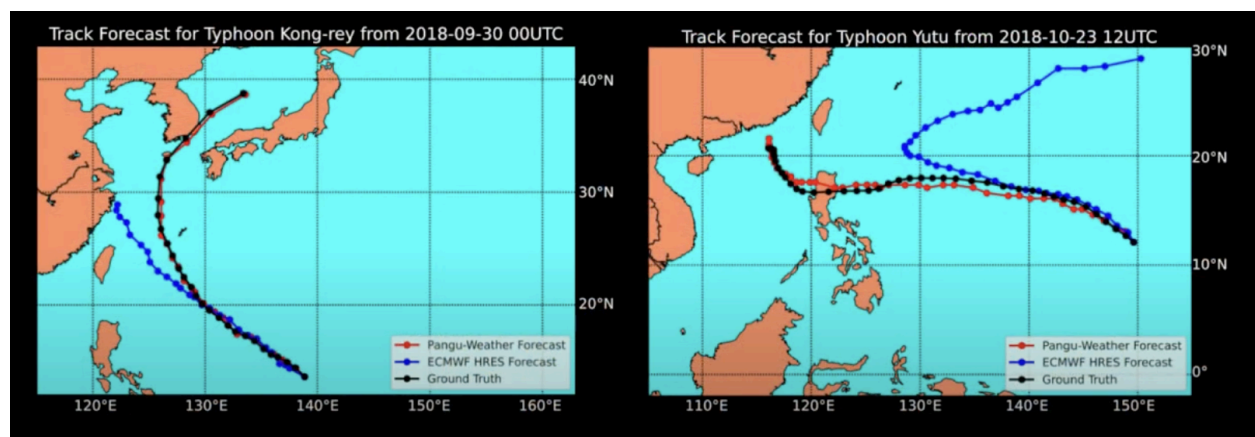
When media have reported on new MLWP models [21], they have heavily or solely used these metrics. While they may be trying to put the model improvements in more understandable terms, these metrics also make the model look more impressive. This has attracted publicity to these papers and the companies that developed them, however it has also attracted criticism, suggesting that they may be purposefully used in the papers to increase media attention, at the cost of academic rigor.

Storm tracking

Storm tracking is an obvious choice when looking at how to measure the performance of a model's predictions. This is done by tracking the position of the center of the eye of a storm and measuring the distance between the predicted and real paths. This is already used when trying to decide on the best systems for predicting hurricanes [22], however, it would be problematic for a model trained on all but the last 4 years worth of data as the testing data would only have a very small sample of storms.

Despite this, it has been used by both Pangu-Weather and GraphCast, both suggesting that they are an improvement over traditional models. However, since the release of these models,

predictions have been essentially equivalent to the latest models, because of this it has been suggested that the examples used for the data were cherry picked.



(figure 3, [9]): Taken from slide in Professor Richard Turner's lecture (53:41)[9] which was taken from the pangu-wether paper [1].

“In the comparison they did here to an ECMWF product, the ECMWF product went wrong in these two cases, but from the people I talked to on the inside, they say that these are probably cherry-picked. In general, actually when you do the comparison fairly, the two systems are about as good on hurricanes and typhoons” - Professor Richard Turner [9]

Target sets

Another thing seen in only Google's papers is that their model beats traditional methods on n% of targets. Other than the caveats of it not actually being the best traditional model (explanation in 3.4), the targets themselves present another problem. What is being referred to as a ‘target’ is an average RMSE. If a model is 0.01% better on 80% of targets, the target set would report that the model beats a model on 80% of targets, suggesting that it is significantly better when, in reality, only a marginal improvement has been achieved.

Conclusion on testing criteria

Due to the fact that weather model variables are so tightly linked, improving one, such as Z500 or T850, is incredibly difficult without also improving the others, and improving one variable usually leads to universal improvements across the board, as the whole system is so interdependent. For this reason the same systems recommended by academics and institutions that are already well established in this field will be used: measuring a latitude-weighted Z500 and T850, RMSE and ACC, over 1-10 days lead time, using 4 years and 10 months worth of test data. One metric is an average of over 218 million measurements, in total the comparison will consider over 875 million measurements per model. If a model is better across these metrics, it is a safe assertion that it is a better model. (see appendix B 3)

3.2 FourCastNet

In March or 2022 Nvidia released FourCastNet [3] (short for Fourier Forecasting Neural Network). The main focus of this project was not weather forecasting, but a demonstration of how GPU acceleration can speed up otherwise hugely expensive CPU-intensive problems (and so sell more GPUs). In this capacity it did succeed as one GPU was able to create a 7-day forecast in a fraction of a second ‘five orders-of-magnitude faster than NWP’ something that in some cases is computed on 10,000s of CPU’s [39, 40]. However it did not succeed as a weather prediction model, dramatically underperforming HRES with an RMSE of 5-day Z500 forecast using 484.5, which is much worse than the 333.7 of HRES [3](see 2.1 for details on why this accuracy measure is used as a measure of skill for weather models).

The model was more of a marketing ‘stunt’ than an attempt at a precise and accurate weather prediction model so less scrutiny of this paper is necessary, although it is arguably the genesis of large MLWP models and there is still some value in looking at it. The main reason that it was not designed to be an accurate weather forecast model is that, at the time, effective MLWP was thought to be a long way away, as such it was decided to focus on the modelling of atmospheric rivers, AR’s. These are ‘rivers’ in the sky that transport huge amounts of water vapour from the oceans to land [41, 42], and are crucial for agriculture but can be responsible for floods and droughts. As such, the model used a smaller set of parameters that were more important for modelling AR’s. This allowed them to train in a reasonable time given their limited resources.

Vertical Level	Variables
Surface	U10, V10, T2m, SP, MSLP
1000hPa	U, V, Z
850hPa	T, U, V, Z, RH
500hPa	T, U, V, Z, RH
50hPa	Z
Integrated	TCWV

(Table 2, [3]) (page 7, Table 1). Note: The ‘Integrated: TCWV’ refers to the total water vapour in the air column and is used instead of Q to better quantify the total water content that could be carried out without having to model q for many levels.

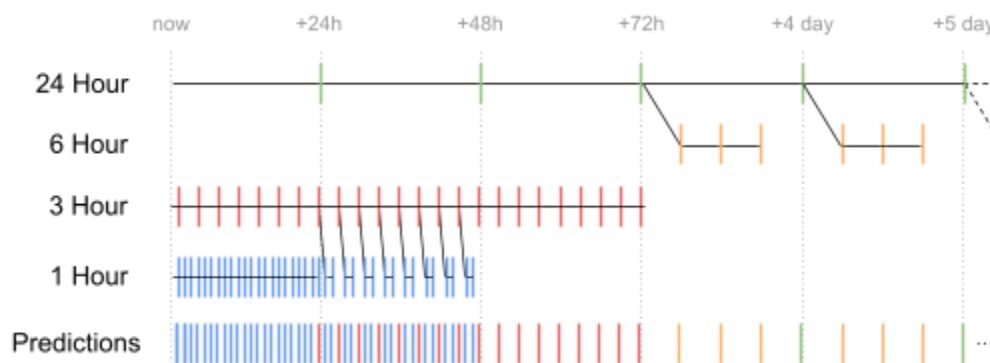
The model was good marketing for Nvidia, as it enabled them to achieve their initial objectives, but the shortcuts taken and the lack of improvements, limited its usefulness in predicting weather.

3.3 Pangu-Weather

The Pangu-Weather paper was hugely influential. Not only was it the first model to challenge NWP and the idea that MLWP was 10 years away; adding to its significance, it was also the first Chinese paper to be published in the journal Nature. Pangu-Weather was the first MLWP model that had high enough accuracy and skill to be comparable with the state-of-the-art deterministic NWP models, specifically the ECMWF's HRES [1]. Unlike FourCastNet the objective was to create a model that could accurately predict the future state of the atmosphere; therefore a full 3D representation of the atmosphere was used.

Pangu-Weather uses a transformer system in which the atmospheric layers are embedded using 3D patch embedding and are merged with a 2D patch-embedded surface layer. This is passed into several transformers that compute the next weatherstate. Like traditional models, this is done recursively to predict weather out to a desired lead time.

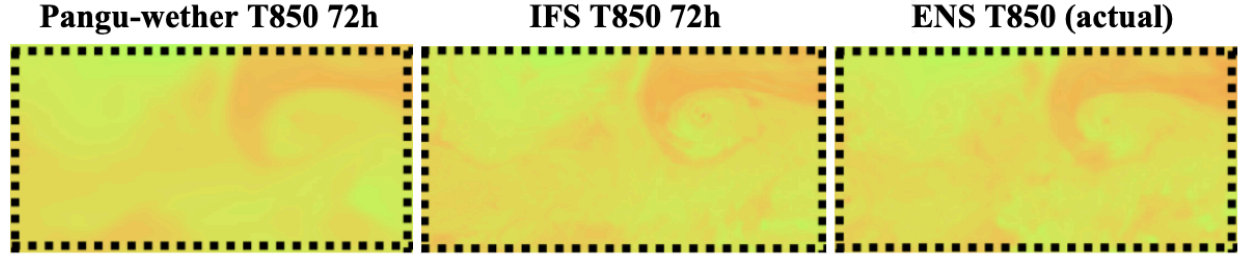
The predictions are based on ERA5 reanalysis with a resolution of 0.25° , in the paper 4 models were trained to predict in 1, 3, 6, or 24 hour time steps. It was found that the longer the time step the less errors accumulated and the more accurate the long term results were. However the shorter 1-3 hour models were much better at modeling higher frequency phenomena and were more accurate in shorter timescales. As a result a system to use a hybrid approach to predict the weather using multiple models was proposed. Such a system would see short term forecasts (up to 72 hours) being predicted with a hybrid system of the 3 hour and 1 hour models. Later forecasts are predicted using the longer models, with a consistent model used when it is both available, and the highest accuracy, then hybrid models used for irregular time steps where the model is worse if used alone. See diagram for an example prediction schedule:



(Figure 4, original) Diagram showing how different hybrid models can be used to produce accurate high frequency weather predictions while maintaining high accuracy and reducing error accumulation. A similar system is in place at the ECMWF [25] (I believe it just uses 24 and 6 hours in a hybrid system, but information is sparse and it may be a pure 6h system or a combination system)

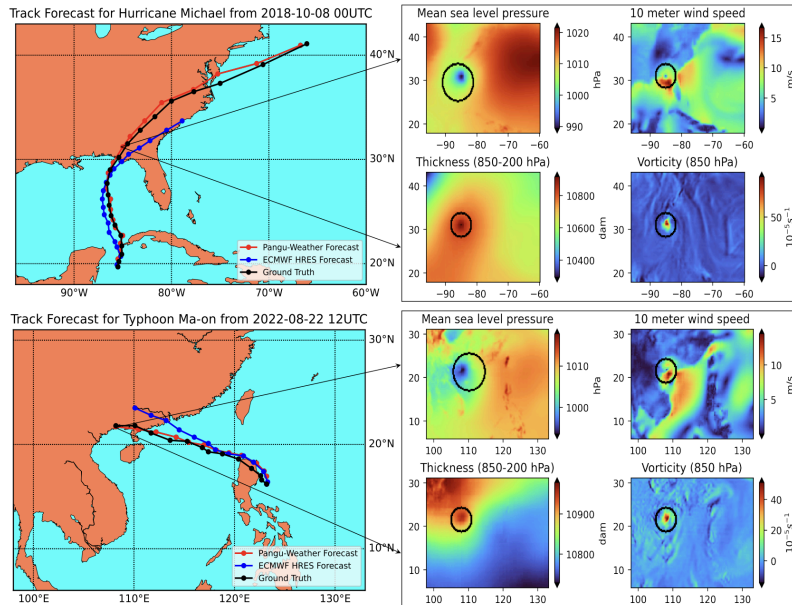
The outputs of the model, especially after many time steps, are clearly smoothing, this is expected from a machine learning model. Having high-frequency weather patterns smoothed out

and only having larger lower-frequency weather patterns modeled (see figure below) was expected from an ML model, however, the model is relatively unaffected by the inputs being smoothed and is able to continue to a reasonable level of accuracy.



(figure 5, [1]) Left: Pangu-Weather 3-hour T 850 120°W -180° 0°- 30°N 72 hour prediction, Center: IFS T850 120°W -180° 0°- 30°N 72 hour prediction, Right T850 120°W -180° 0°- 30°N ERA5 ground-truth. All from [1] (page 11, Figure:7)

It was also expected that any model would struggle to accurately model extreme weather like hurricanes accurately, due to the rarity of these events in the training data and ML's tendency to underestimate extreme values. However the paper claims that this is not an issue, and it is able to accurately model extreme weather events (see figure below). Since its release, as mentioned in 2.1, it has only beaten deterministic NWP about 50% of the time and it is believed that these are cherry-picked examples [9]. However matching NWP is still unexpected and impressive, showing that this model is skillful and able to predict to a good accuracy. Note: the ECMWF 'storm eye' tracking does not work well with the Pangu-Weather model, so to evaluate accuracy a CV system was developed to plot the centerpoint of the storm eyes for Pangu-Weather predictions, it's possible that this has some effect on the results.



(Figure 6, [1]) (page 15, figure 11) Left: the tracking of cyclone eyes for Hurricane Michael (2018-13) and Typhoon Ma-on (2022-09) by Pangu-Weather and ECMWF-HRES, with a comparison to the ground truth (by IBTrACS). Right: an illustration of the tracking process, where we use Pangu-Weather as an example. It locates cyclone eye by checking four variables (from forecast results), namely, mean sea level pressure, 10m wind speed, thickness between 850hPa and 200hPa, and vorticity of 850hPa). The displayed figures correspond to the forecast of these variables at a forecast time of 72 hours, and the forecast of Cyclone Eye is indicated using the tail of arrows.

While the model's inaccuracies in cyclone parameters has not affected its accuracy in prediction of cyclone trajectories it does mean that it is underestimating a storm's intensity “*All of the machine learning models underestimate the peak amplitude of winds associated with the storm*” [24]. This is a major problem that will have to be addressed in order to move away from NWP models to MLWP models - knowing how bad a storm is going to be is as important as knowing where a storm will be.

The paper goes on to show that it outperforms INS (HRES) which is ECMWF's foremost deterministic model, but it does not at any point, compare to ENS which is the actual state of the art. The reasoning given is that being a deterministic model it can only be fairly compared to another deterministic model. And they are unable to make it probabilistic through an ensemble without reducing accuracy “*Pangu-Weather is based on deep neural networks that contains hundreds of millions of parameters, unlike conventional NWP models, that contain much fewer yet physically meaningful parameters. In addition, the learned parameters are highly sensitive to a few random factors during the training procedure (e.g., random seeds, data sampling strategies, etc.) and thus are difficult to be perturbed for specific purposes.*” [1] (page 18, 4.3 ensemble forecasts). When compared to ENS, it is clear that the state-of-the-art NWP models are still superior.

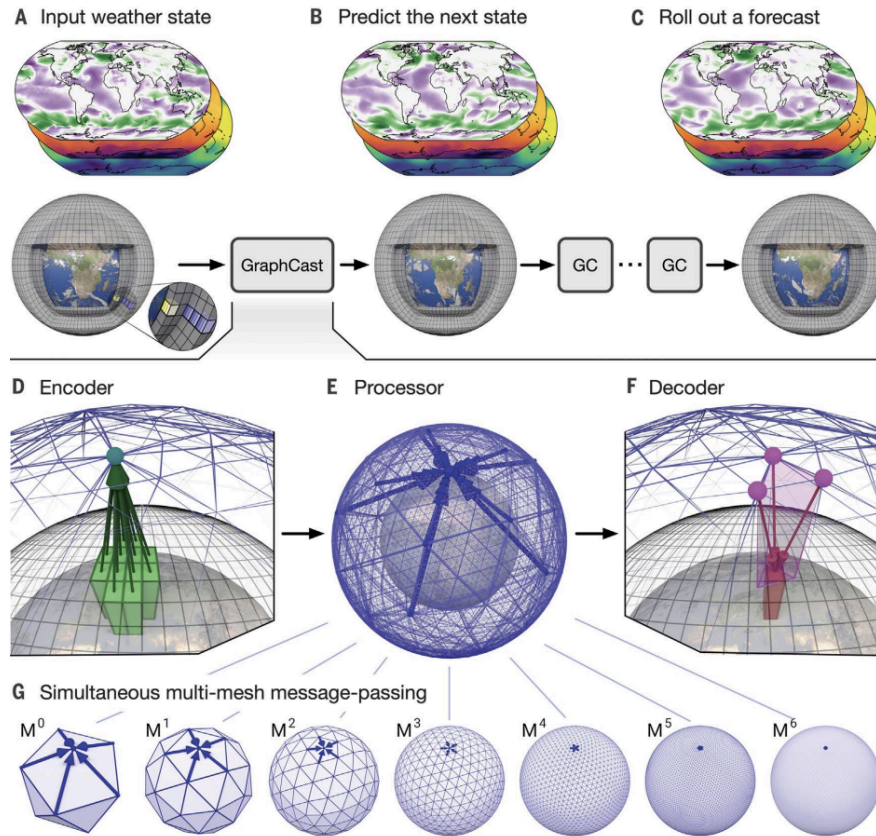
3.4 GraphCast

When Google’s GraphCast paper was released it was groundbreakingly influential, producing the first model to claim to be better than traditional NWP, a feat believed to be a decade away just a few months prior. The models used ERA5 0.25° data, modeling the atmosphere at 37 pressure levels and used a novel graph neural network on a multi-mesh encoding of the atmosphere [2]. This allowed cheap processing and let the model consider nearer, smaller, high-frequency weather patterns while also processing larger weather patterns that were further away.

GraphCast system description

The GraphCast system uses a deep graphical neural network sitting on top of a ‘multimesh’ where the GNN uses data passed along the edges of the mesh as it would in a graph neural network.

For each point on the graph, areas on the input data that are nearest to the point on the graph are encoded for two time steps worth of data. The GNN then predicts the conditions at that point in 6 hours given the conditions at the point and the 5 or 6 adjacent points for each graph that the point is on. This happens simultaneously for all points, then all of these predicted points are decoded, to produce a snapshot of the earth in 6 hours time, this is run autoregressively to predict out to a specific prediction time. Below is a diagram of the system from the paper



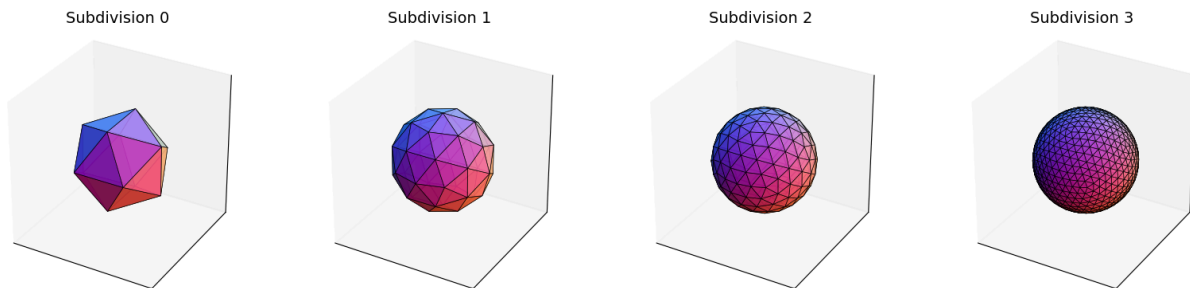
(Figure 7, [2]), description paraphrased from paper: (A) The input weather states are 0.25° latitude-longitude grid comprising five surface variables, and six atmospheric variables that are at 37 pressure levels. (B) GraphCast predicts the next state of the weather on the grid. (C) A forecast is made by iteratively applying GraphCast to each previous predicted state that represents the weather prediction at successive lead times. (D) The encoder component of the GraphCast architecture maps local regions of the input into nodes of the multimesh graph representation. (E) The processor updates each multimesh node using learned message-passing. (F) The decoder component maps the processed multimesh features back onto the grid representation. (G) The multimesh is derived from icosahedral meshes of increasing resolution, which has uniform resolution across the globe. The learned message-passing over the different meshes edges happens simultaneously, so that each node is updated by all of its incoming edges.

Importance of a multimesh

The concept of a multimesh is the defining feature of this paper, what is a ‘multimesh’, how is it created, and how is it used?

The creation of a multimesh

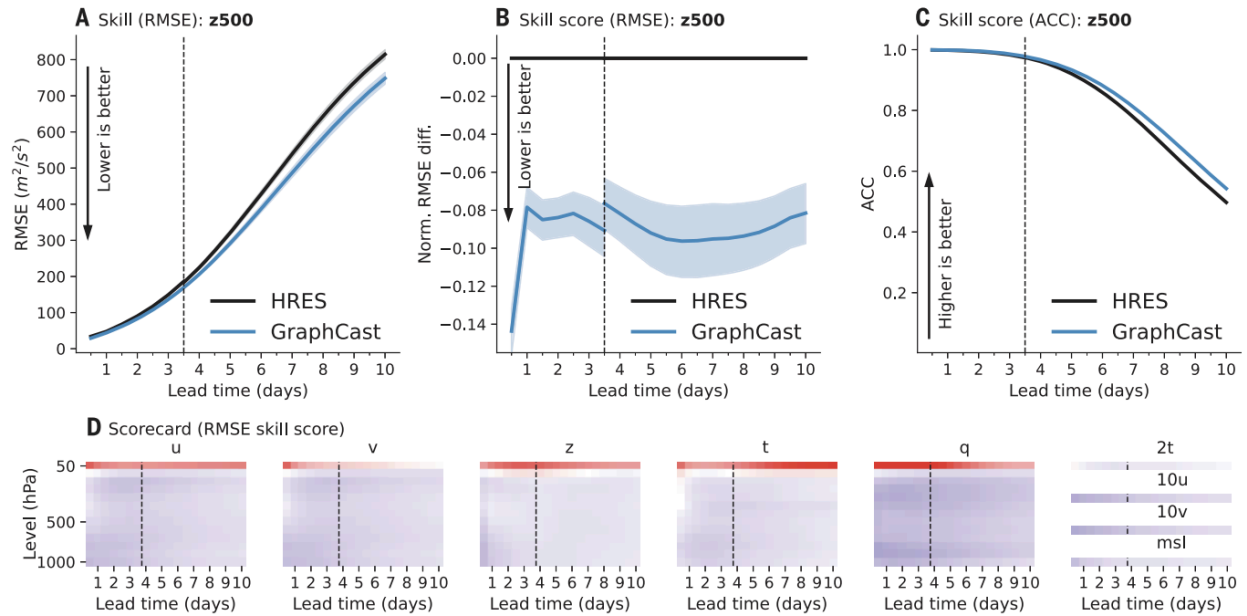
A multimesh is simply a set of multiple meshes, the points that remain in the same place across meshes are shared and are considered the same. The meshes are subdivided icosahedron from 0 to 6 subdivisions, a subdivided icosahedron is created by adding a vertex on the centerpoint of each edge, adding edges to connect the vertices by constructing a triangle on each face that is $\frac{1}{3}$ the area of the original face and rotated 180° , the new vertices are normalized to be a consistent distance from the origin at the centre of the icosahedron. The subdivision algorithm is repeated until a desired number of subdivisions is achieved. The diagram below shows how an icosahedron can be subdivided repeatedly.



(Figure 8, original) from: ‘visualisations/subdivided_Icosohedrons.ipynb’ shows how an icosahedron can be iteratively subdivided.

The meshes in the paper are constructed by first subdividing an icosahedron as described above, and then assigning values to the vertices; each vertex has 227 values (6 for each atmospheric level and 5 for the surface). As such, the points on a given mesh represent the weather in the area closest to them. (These areas are representative of an (n,0) Goldberg polyhedra [26])

Results and undiscussed issues

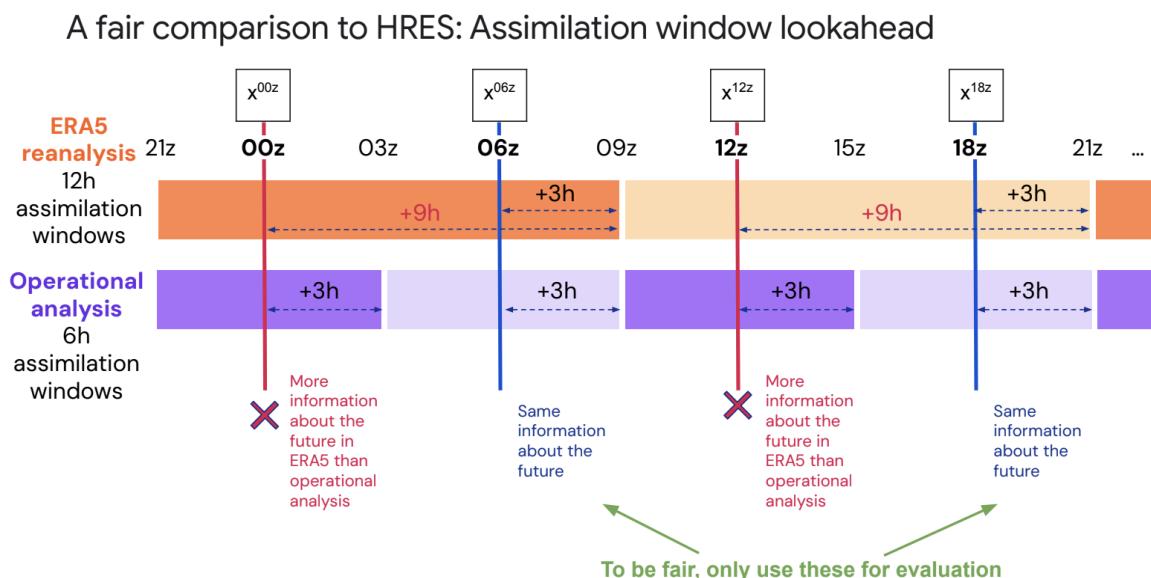


(Figure 9, [2]) (page 3): shows Z500 RMSE and ACC between the ECMWF's HRES and Google's GraphCast

There are a few ways in which there is a systematic advantage in the measurement system; however, while the authors tried to adjust to give a fair comparison it is impossible to completely remove them. Firstly the predictions made by GraphCast and HRES are not having the RMSE calculated in comparison to the same ground truth. If GraphCast was measured against the HRES operational analysis or HRES was measured against ERA5 reanalysis, then there would be an unfair additional error as both systems are different. Therefore GraphCast is compared to ERA5 reanalysis (which it was trained on) and HRES is compared to Operational analysis (which it is designed to run on). As the Operational analysis has a smaller assimilation window and therefore less data, it is noisier and has larger errors artificially inflating the RMSE of HRES. (See appendix A 1 for data assimilation).

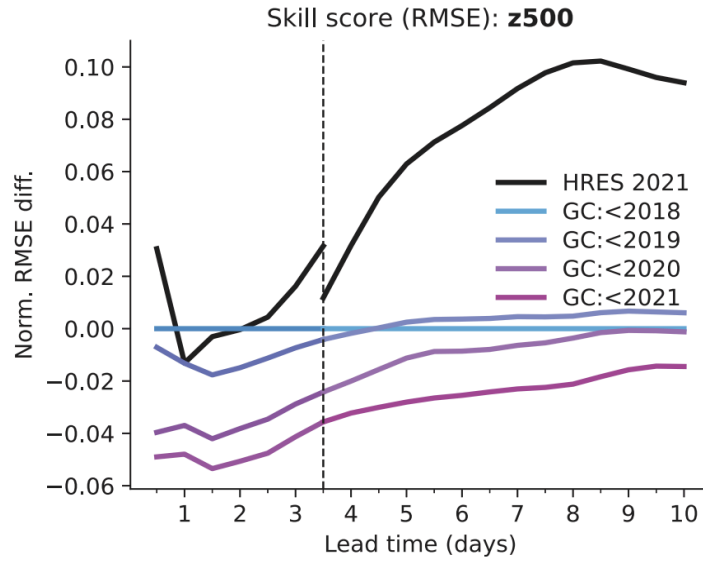
Another issue comes again from the differences between operational analysis for data assimilation and reanalysis (See appendix A 1). Since ERA5 reanalysis has a 12-hour data assimilation windows and GraphCast makes 6-hour predictions, the predictions are made either 3 or 9 hours into an assimilation window, but the predictions made 3 hours into a window have the advantage of making 6 hour predictions where the input data is calculated with measurements taken up to 9 hours into the future; whereas when making a prediction 9 hours into the window the input is only based on data up to 3 hours ahead, forcing the model to have to predict 'blind'. In contrast, HRES, running on operational analysis data, always has to predict across an assimilation window. To control for this advantage where possible (up to 3.5 days), a fair

comparison should be made by only comparing comparisons across an assimilation window (at 06 and 18 UTC) [8].



(Figure 10, [8]) (slide 23): shows how 6-hour predictions are unfairly advantageous for ERA5 based systems when made at 0 and 12 UTC due to the fact the base snapshot will be based off of information taken at or after the first prediction.

It is also notable that GraphCast improves measurably when it is also ‘fine-tuned’ on more recent data; this may be due to climate change or due to there being better sensing in modern-day systems and different underlying data. “We compared their performances to HRES on 2021 test data. [figure below] shows the skill scores of the four variants and HRES for Z500. We found that while GraphCast’s performance when trained up to before 2018 is still competitive with HRES in 2021, training it up to before 2021 further improves its skill scores. We speculate that this recency effect allows recent weather trends to be captured to improve accuracy. This shows that GraphCast’s performance can be improved by retraining on more recent data” - Remi Lam et al [2].



(Figure 11, [2]) (page 5): shows the z500 RMSE for HRES and several versions of graphcast trained on data up until different years (normalized to Graphcast trained until 2018 (used for all other comparisons)).

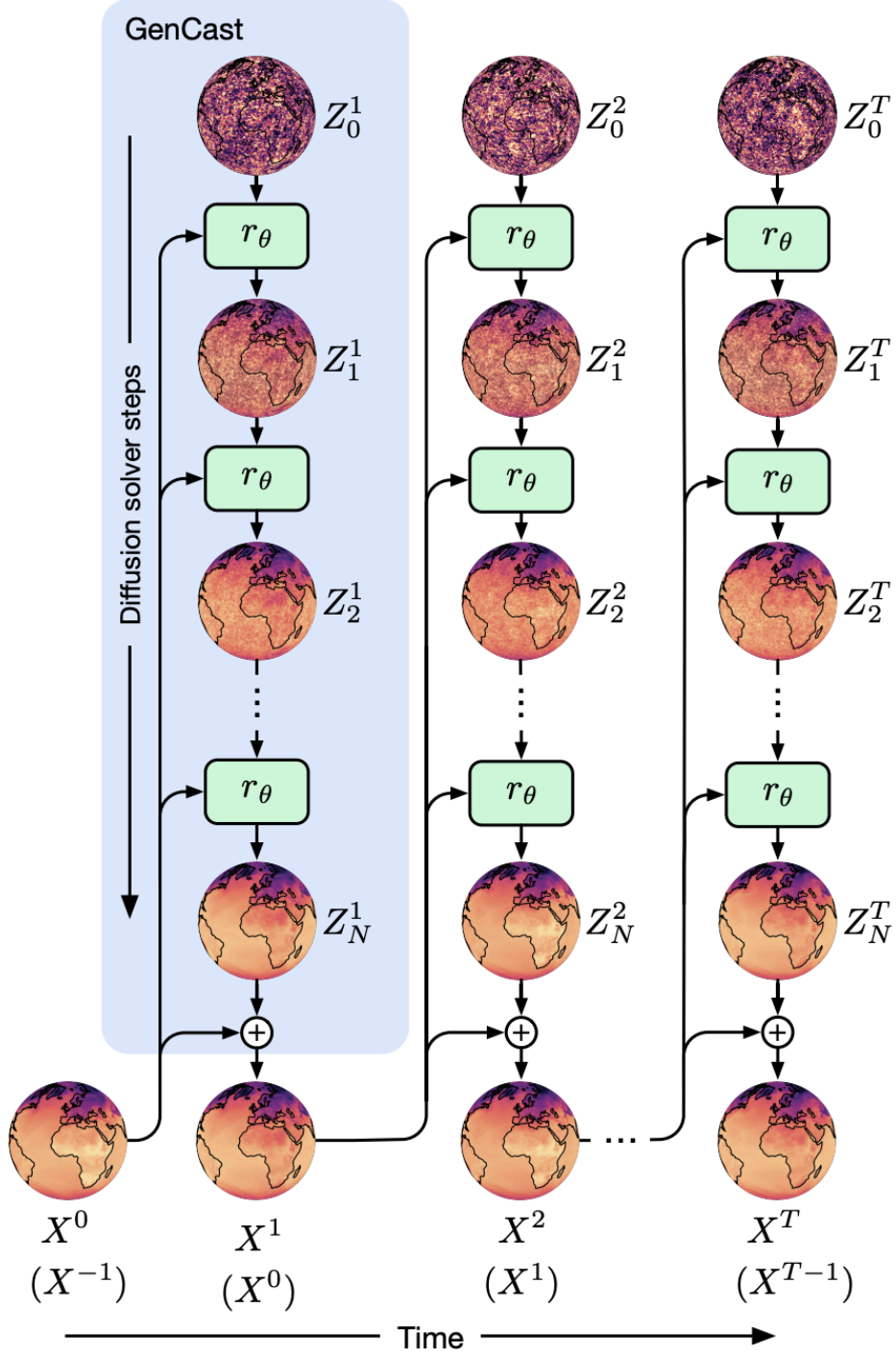
3.5 GenCast

GenCast [6] is a model developed by Google as a successor to GraphCast. Exact details of how the model works are not known, only a basic outline, however from what is discussed in the paper broad details of the predictions can be deduced.

GenCast system details

It is known that the model uses probabilistic de-noising diffusion to perform 12-hour predictions based on an encoding of the starting weather state. It can be run iteratively with the previous prediction running as the new input to predict weather out to a specific lead time. The system is based on ERA5 reanalysis with a 0.25° resolution and uses 6 surface and 6 atmospheric variables at 13 levels in the atmosphere. The exact details of encoding for the system is unknown, although we do know it uses a subdivided icosahedron point encoding system.

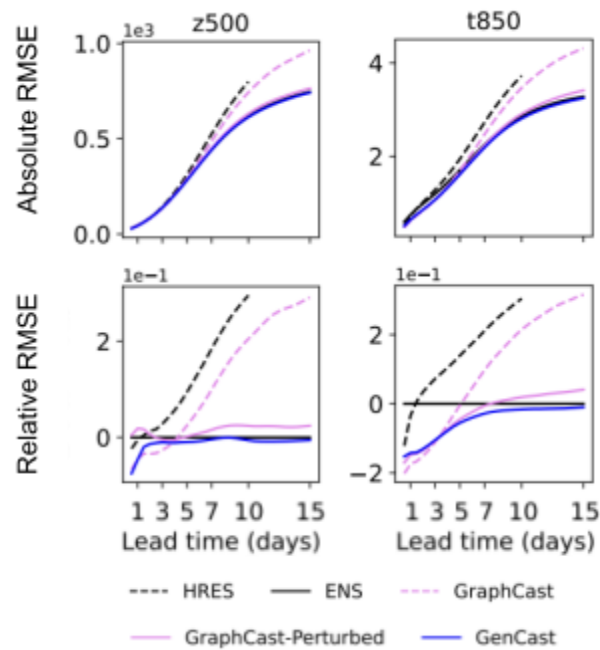
The GenCast system is able to make an ensemble forecast, which is a large contributor to the increased performance of traditional NWP models. It has been difficult to reproduce in MLWP models successfully because the models are very sensitive to input data noise and adding additional noise reduces the accuracy of predictions more than the ensemble increases accuracy. Ensembles are, however, easy to implement in a PDD model as different noise samples can be used to produce similar but different results. However this has the side effect of not allowing the representation of uncertainties in the input data across the ensemble output, as changes in this would result in the same issues as earlier MLWP systems.



(Figure 12, [6]) (page 2) annotation from same source: Schematic of how GenCast produces a forecast. The blue box shows how conditioning inputs, (X^0, X^{-1}) , and an initial noise sample, Z_0^1 , are refined by the neural network refinement function, r_θ (green box), which is parameterised by θ . The resulting Z_1^1 is the first refined candidate state, and this process repeats N times. The final Z_N^1 is then added as a residual to X_0 to produce the weather state at the next time step, X^1 . This process then repeats, autoregressive, $T = 30$ times, conditioning on (X^1, X^{-1}) and using a new initial noise sample Z_0^t at each step to produce the full weather trajectory sample. Each trajectory generated via independent $Z^{1:T}_0$ noise samples represents a sample from $p(X^{1:T} | X^0, X^{-1})$

Results

The results discussed in the GenCast paper not only suggest that GenCast is better than the previous GraphCast system, but that GenCast is better than even the best ensemble systems, beating ENS. This is impressive, it's also impressive that they have managed to create a 'graphcast-perturbed' that produces an ensemble that improves the performance of an ML model through ensemble. While ENS does have a higher resolution and the improvement over ENS is only small over its useful time period, the data would suggest that GenCast is the best medium range weather prediction model.

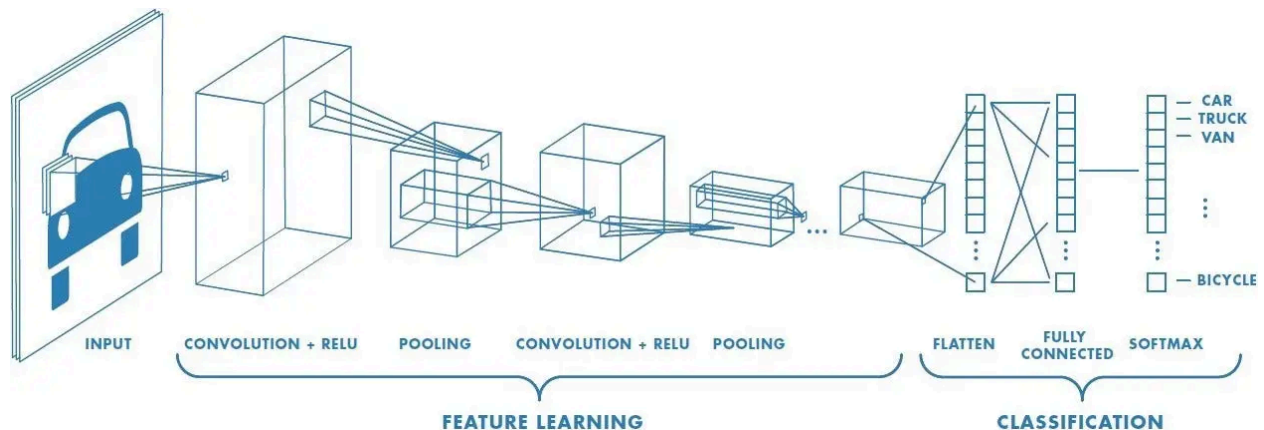


(Figure 13, [6]) shows RMSE of Z500 and T850 of HRES, ENS, GraphCast, GraphCast-pterbed (ensemble graphcast), and gen cast, relative RMSE is compared to ENS.

GenCast is available from Google with payment, and the ECMWF is trialing its use to evaluate using it in a 'Multi model ensemble'. There is a possibility that it may be made open source in the future.

3.6 Convolutional neural networks

CNN's or Convolutional Neural Networks are a type of neural network that uses convolutional layers to extract features from the input data, and uses said feature extraction to give an output. Typically they are used for 2D image processing, but the underlying concept can be applied to any number of dimensions. While CNN's are widely used in classification, the feature learning can be applied to a much broader set of features.



(Figure 14 [30]) Shows A CNN used for classification. Feature learning system is shown extracting features from an image of a car, with a final classification step shown at the end.

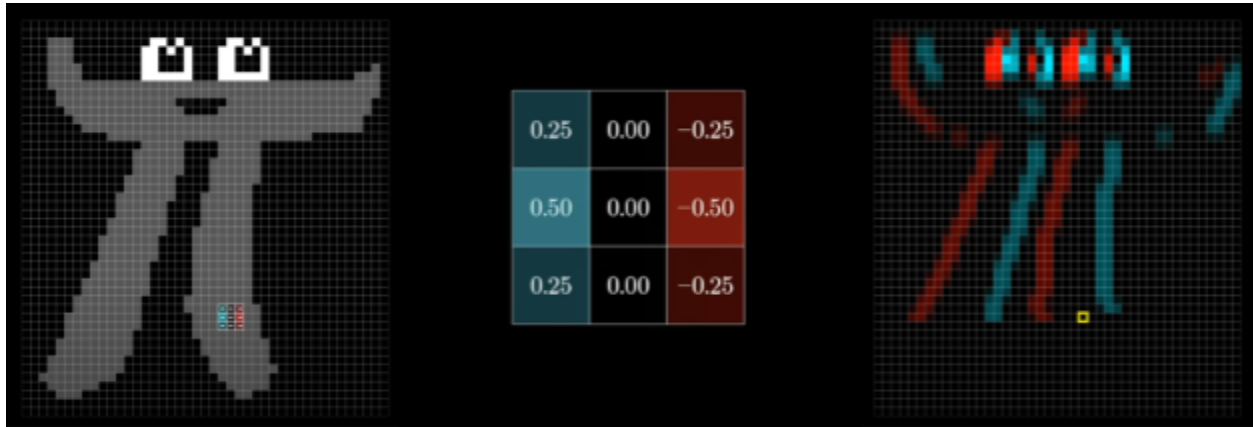
This section gives an overview of CNN's, broken down into:

- Functionality of CNN's, how CNN's function and why they work.
- Hyperparameters and design considerations, how to make a good CNN.
- Existing use in weather forecasting, how CNN's are already used in weather prediction.

Functionality of CNN's

CNNs work by automatically learning spatial patterns in data through convolutional operations. An example of this would be identifying a weather front, or a changing air mass.

Convolutional layers apply kernels to a data set, these kernels scan across the axis of the data set, applying a filter to produce resulting pool layers. These pool layers represent the prevalence of the output of a filter at different points in the data. Filters are learnable and are trained to describe important features.



(Figure 15, [29]), shows how vertical Line features can be extracted from an image. Left: input image (Pi creature), center: filter that extracts the gradient change horizontally (returns negative values for vertical lines that get brighter towards the right, and vice versa), right: the output of this filter mid processing.

This then produces a set of outputs that are pooled, these pooled outputs are essentially images that represent the presence of different learned features, the same steps can be run again to get an analysis of the features. Finally the system has a pooling layer, this can be used to effectively process an image, the exact specifications of this last step is dependent on the use case.

Hyperparameters and design considerations

While creating a CNN there are four main things you can adjust to alter the behaviour of a model with use case: kernel size, number of filters, depth, and padding. This subsection will be looking at each one individually.

Kernel size

The size of the kernel represents the area in which filters are applied to an underlying image. It is important to have a kernel that is large enough to see features to be extracted, and small enough to not overly smooth features. In papers and documentation a kernel size between 3x3 and 5x5 is common but different size kernels can be used depending on the use case.

Number of filters

Increasing the number of filters allows the extraction of more spatial features and also the extraction of more complex features in future steps. In papers and documentation the number of filters used varies with typical values between 32 and 128.

Depth

The more layers in a system the more abstract features and patterns can be extracted and learned. However, as in other NNs, increasing the model size without an increase in data can result in overfitting.

Padding

The edges of an input can be problematic, especially with large kernel sizes compared to the input. As such, a decision needs to be made on how to deal with the edges of an image and how they can be padded.

Existing use in weather forecasting

CNN's have been recently used in MLWP as discussed in 3.2, however, they have been used in short range forecasts, especially 'nowcasts' for longer [43]. Having a local model that is easy to make is adventitious as it allows for easy bias adjustment for specific population centers, very short range forecasts can rely on high resolution frequent weather data, which is ideal for this use case.

3.7 Attention based transformer systems

Transformers were first proposed in the paper 'Attention is all you need' [31], while originally proposed for language processing, and being most visibly used in LLMs, they have found other uses. Transformers work on 'self attention', the mechanism allows relationships between all parts of an input to be modeled, allowing a system to model local and long range dependencies in data.

Functionality of transformer systems

Transformer models work by calculating the attention scores between each part of the input and every other part. This lets the model adjust for how much different regions should affect the final output.

Attention scores represent how much focus the model should place on each part of the input when generating an output, based on the relevance between query and key vectors (computed as a dot product). A query vector is a vector that represents the current position where information is being gathered by the model, and the key vector represents where more information may be found. Value vectors represent the information at a given position, attention scores are used to weight the output value, this outputs the most important information.

The global nature of attention imposes no limit on how spatial relationships and features may be found, and allows a model to find these patterns no matter how widespread they may be. As there are likely multiple features in the underlying data, multiple 'heads' may be used to represent this.

Unlike CNNs transformers don't have any built in understanding of the structure of the input data, input data is treated as a set. This issue is resolved by having positional embeddings for each input vector; these encode the position of the vector, allowing the model to understand the positional relationships between vectors.

Hyperparameters and design considerations

While creating a transformer model there are several things that can be adjusted to alter the behaviour of a model, this subsection will be looking at these individually.

Input size

The input size of a transformer system is denoted by the number of vectors in the input and the dimensionality of each vector.

The number of vectors in an input is determined by the size of the context window. A large context window allows for a large input to be considered, but means that the model may be difficult to train and expensive to infer.

While dimensionality is sometimes pre-determined by the dataset, if you have control over how vectors are made you may have control over this parameter. Large vectors contain much more information but are harder to compute.

Embedding dimension

The dimensionality of the vectors being fed into the transformer layers depends on model size and input complexity, typically between 128 and 512 dimensions. This parameter needs to be tuned keeping it low to create an efficient model without cutting too many dimensions and resulting in a loss of accuracy.

Depth

The number of layers or depth of a transformer. More layers allows for the modeling of more complex relationships but at the detriment of training time. Unnecessarily increasing depth results in a model that is worse given the same training.

Attention heads

The number of attention heads dictates the number of relationships that can be modeled, a small number of heads may result in poor performance as heads try to represent multiple relationships, but a large number of heads impairs training and may make inference unnecessarily expensive.

Positional encoding

As mentioned transformers operate on a set as an input, and don't have any built in understanding of the structure or position of inputs. Positional encoding may represent where in

a text a token is, or it may represent where on earth a meteorological measurement has come from, the decision on how to encode this and what to encode is important in training.

4 Methodology

A defined methodology, to accurately and fairly train and compare any MLWP models, was developed. In order to do so a fair and common methodology has been described in the following section. The section focuses on 3 parts of the methodology:

- Data: For a fair comparison, a consistent good source of data is needed. This data needs to be consistent so there are no advantages or disadvantages to different models; also the data must be of high quality and appropriate for the needs of the models.
- Models: All parameters needed must be made and trained with optimal parameters and conditions, so that any comparisons made between models are not tainted by nonoptimal hyperparameters.
- Testing: For a fair comparison between models the same metrics need to be measured, these metrics need to be measured in the same way to allow them to be an equal comparison.

4.1 Data and pre-processing

The data used by any models needs to be sourced and processed before it can be used to train any models, furthermore the data must be adequate for the constraints of the models.

Data source

To construct a snapshot of the Earth's atmosphere, first assimilated data is needed, the data must meet the following criteria:

- Global data. Medium-range models are global models and as such are reliant on global data.
- All parameters needed. The data must contain the data for the following parameters: surface pressure, surface temperature, geopotential height, air temperature, air humidity, and wind speed and direction.
- Large historic backlog. 20 years worth of data is needed (15 for training and 5 for testing).
- Free and easy accessibility. The data must be easily accessible (to an academic), more specifically the data defined above must be accessible with minimal processing.

Both the ECMWF and NOAA offer products that fit the specifications above; ERA5 and GDAS. ERA5 is difficult to source in a resolution that is appropriate for the constraints of training placed

upon the models being trained in this paper, as such the NCEP Global Data Assimilation System GDAS [16] has been used. (The NCEP is a part of the National Weather Service which is a part of NOAA, GDAS is the NCEP's twice-daily global analysis), it is produced at a 2.5° resolution at 12 pressure levels 1000, 850, 700, 500, 400, 300, 250, 200, 150, 100, 70, 50 hPa.

Pre-processing

From this data, 7 channels of data were used, 5 for atmospheric levels (air temperature, northward wind speed, eastward wind speed, relative humidity) and 2 for surface (sea level pressure, surface potential temperature). These are supplied in files containing a year's worth of data, for each parameter. The file contains 1 row for each longitude and latitude recorded, for each day in the year, also for atmospheric levels they are broken up by level. This was consolidated into 1 row for every longitude and latitude for each day, with each row containing all data at 6 levels (1000, 850, 500, 250, 50) for each location. These files are referred to as 'snapshot_yyyy.parquets'. Parquets have been used because they have reasonable compression, they work well with large file sizes, keeping file sizes small, and I have professional experience of using them.

4.2 Models and training

In order for models to be fairly compared they must first be created fairly, and trained. This involves ensuring that the creation of the models is done so that each model has the best hyper parameters for its objectives, and ensuring each model receives equal computation for training.

Hyperparameter tuning

The number of layers, layer size, kernel size, number of heads, are the main parameters needed among others to be adjusted to create an accurate and precise model. In order to do this fairly, so all models have selected starting parameters based on what is believed to be good practice, this information is taken from a variety of sources and pieces of documentation. This is then elaborated on by testing of different parameters. This is different for each model and information about this process is elaborated on in each subsection of section 5.

Training

For a fair comparison all models must be trained on the same hardware for the same time. The models will be trained on JupiterLab instances on Python servers Hosted on the BlueBEAR cluster, provided through the BlueBEAR OnDemand service. This service has been generously provided by the University of Birmingham. (Due to issues with this service some models were eventually trained on an Identical Jupiter lab Azure system)

The system parameters of the instances used for training are consistent and described below:

Component/Software	Specifications
CPU	10 threads ('cores'), from an Ice lake (10th generation) Intel xenon platinum processor
Memory	40GB, (4 GB are allocated per thread)
GPU	1 Nvidia ampere A30 (24GB VRAM)
Server kernel	Python 3.11.3 (BEAR apps version: 2023a)

(Table 3, original)

While ensuring that all models have the same training time, this timing is based only on the time to train a model, not the time to load or pre-process model targets or inputs. This allows for control of hardware limitations and any inefficiencies in code that would not exist for other researchers, or would be insignificant given a long enough training schedule. All models received a maximum of 1 hour of training.

4.3 Testing

In order to give a fair and equal test it is imperative to use a testing methodology that fairly tests statistically significant parameters of the predictions. This is achieved by using a testing method that is universally accepted and has been used consistently for a prolonged period, and has stood up to scientific criticism.

A latitude-weighted Z500 and T850, RMSE and ACC, over 1-10 days lead time, using 4 years and 10 months worth of test data was used. The climatological mean used for the ACC is the climatological mean of the test data.

Latitude adjustment ensures that measurements evenly represent global performance. Z500 and T850 are critically important to predicting the path and behaviour of storms; this system is universally recognized as the standard benchmark for evaluating medium-range weather forecasts.

(Further reasoning behind this is discussed in 3.1)

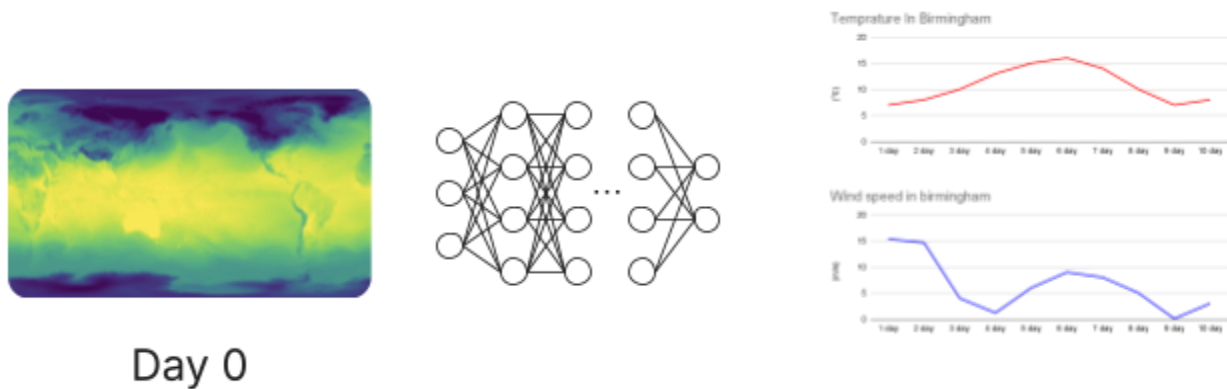
5 Implementation

5.1 Traditional NN model

Connected neural networks could be used to predict the weather, similar systems are used in a specific type of short-range weather forecasts called ‘nowcasts’, these are very short range locationally precise weather forecasts.

NN system overview

Conceptually the simplest system by which an NN could forecast the weather is by taking a snapshot of the state of the Earth's atmosphere and outputting a prediction of the weather at a specific location over the next number of days (see diagram below).



(Figure 16, original) Diagram of how a NN-based system works, A snapshot of the world weather is shown with the caption ‘Day 0’ then a NN diagram is shown then two graphs titled ‘temperature in Birmingham’ and ‘wind speed in Birmingham’.

How targets are created

Targets are produced in a preprocessing step, additional from the preprocessing of the data. For each date target files contain the data of the weather in said location for several following days. The data contained includes: air temperature, northward wind speed, eastward wind speed, relative humidity, sea level pressure, and surface potential temperature; in one location for several multiple days. The best weather models in the world struggle predicting beyond 7 days with reasonable accuracy so after testing 21 days of lead time, the decision has been made to use 10 days of lead time, for 4 locations.



(Figure 17, original) Diagram showing the position of targets for a european trained NN. Closest points to: Birmingham, London, Paris, and Berlin on a 2.5° grid.

The model and training

How the NN works

The NN is very basic, it has a huge input layer that contains the data from a 1-day snapshot, it contains several hidden layers and an output layer that contains values for all model variables at 4 locations for several days.

Determining the best dimensionality for the neural network was difficult as the size of the input was huge meaning that small layers would reduce accuracy, and large layers would quickly explode in model size making the model slow or nearly impossible to train. Guidance on optimizing NN is to use trial and error. A balance was found by having a large first layer, followed by several mid-sized layers.

A larger model would improve performance, however the longer training requirements and the limited data would eventually curb the benefits. Given the speed of training and the training time-frame the model performance is very likely to be data limited. The hyper parameters of the model are that a model be able to reach a similar plato to larger models, and that the model be small enough to be quickly and cheaply inferred.

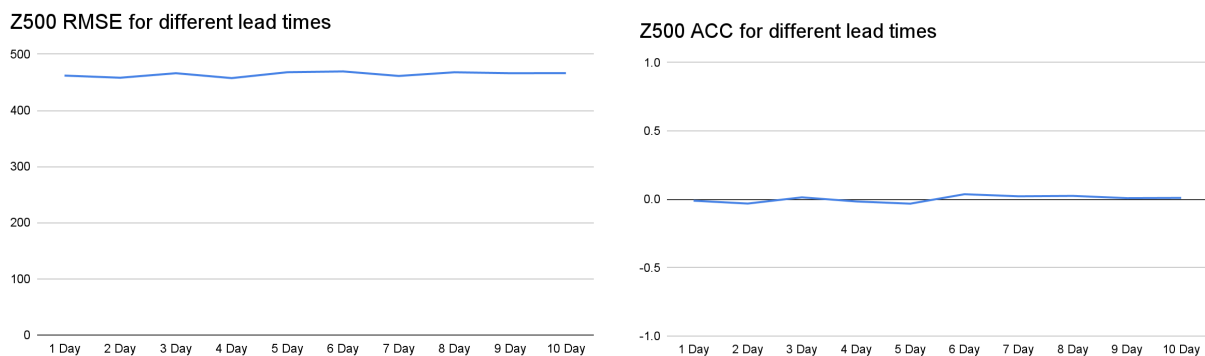
The chosen hyperparameters for model layers are as follows:

- Input layer
 - size 329472 (32 parameters at 10298 points)
- Layer 1
 - Fully connected
 - Size 4096
 - ReLU activation
- Layers 2 - 5
 - Fully connected
 - size 1024
 - ReLU activation
- Output layer
 - Size 2688 (model is actually trained on 21 days worth of data: 32 parameters at 4 points for 21 days)

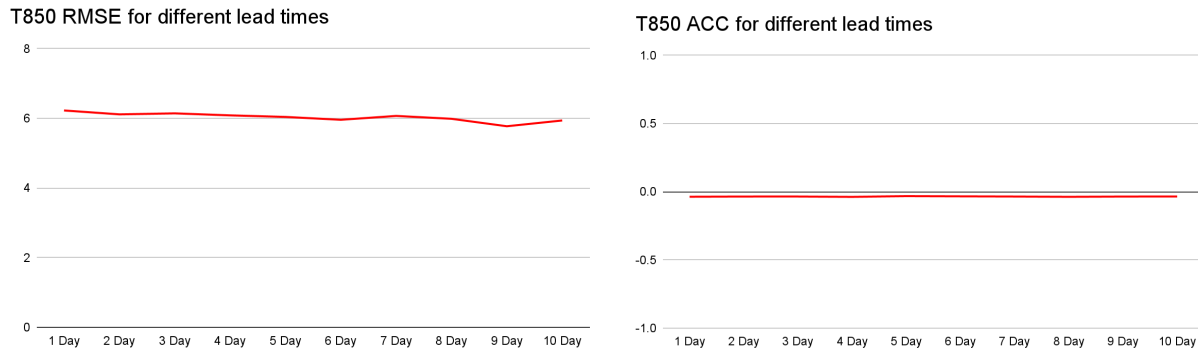
Results

After training, the models were tested on data from 01/02/2010 - 30/11/2015 below are the results for the RMSE and ACC for Z500 and T850 as described in the methodology.

Note: latitudinal adjustment would unfairly inflate the performance of predictions near the equator as the positions for these predictions is determined by the location of the desired forecast. Therefore latitude adjustment is not used in the following metrics.



(Figure 18, original) Charts showing: RMSE for Z500 up to 10 days lead time averaged over 4 years and 10 months, ACC for Z500 up to 10 days lead time averaged over 4 years and 10 months.



(Figure 19, original) Charts showing: RMSE for T850 up to 10 days lead time averaged over 4 years and 10 months, ACC for T850 up to 10 days lead time averaged over 4 years and 10 months.

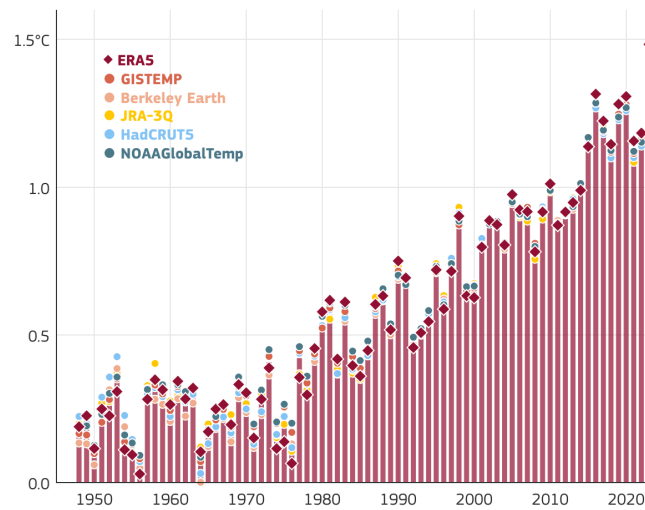
The above data is at first unexpected, the graph seems very flat. This is unfortunately due to the fact that the early predictions are so bad they are essentially guesses, and the later predictions are also just guesses. It's a similar story with the T850 RMSE with predictions being so bad they sometimes get better over time.

The ACC results confirm the predictions that; an ACC of 0 is equivalent to predicting the climatological average, and a negative ACC means that a prediction is worse than guessing the average. This negative performance would suggest that there is a bias in the underlying data; this has been seen in previous papers. With it being posited that this may be due to climate change, or a change in the characteristics of the assimilated data due to more measurements going into the data assimilation.

The underlying data has a low spatial and temporal resolution, and as such is unlikely to change significantly with more measurements. The results will be influenced by the average yearly temperature of the data set used. The average yearly temperature in the training data (1995-2009) was 0.6° to 0.9° above pre industrial levels and the average yearly temperature in the testing data was 0.8° to 1.2° above pre industrial levels [28] (see below diagram), a system that learned in training to predict the average temperature would predict 0.25° - 0.3° lower than the average in the training data.

Annual global surface temperature

Annual average, since 1948. Vertical bars represent the average of available datasets. Increase above 1850–1900 reference (pre-industrial)



Data: ERA5 (C3S/ECMWF), JRA-3Q (JMA), Berkeley Earth, GISTEMPv4 (NASA), HadCRUT5 (Met Office Hadley Centre) and NOAA GlobalTempv6 (NOAA) - Credit: C3S/ECMWF



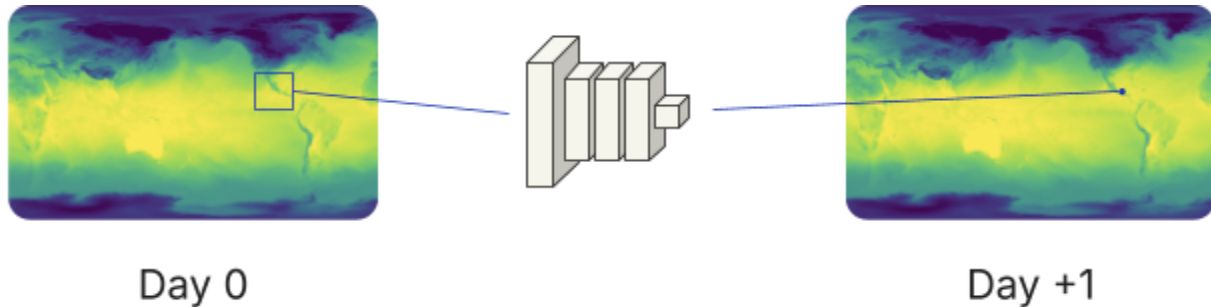
(Figure 20, [28]) figure shows average temperature above pre industrial levels from 1948 to 2023 from 6 meteorological datasets, ERA5, GISTEMP, Berkeley earth, JRA-3Q, HadCRUT5, NOAA GlobalTemp

Conclusion

A traditional NN is not a feasible model for weather prediction. It is incredibly difficult to come up with a model that can accurately predict the weather using this architecture. It may be more feasible to use a smaller part of the Earth and predict the weather over a shorter time frame, using a similar system to make short term predictions. This is already used and is called 'nowcasting'. This model should only be used as a benchmark and not be a serious consideration for weather predictions.

5.2 Simple CNN Implementation

The next system is a convoluted neural network. The system aims to predict a snapshot given a snapshot of the previous day. Despite the fact that the data is represented as rows in a large table it is essentially just a very colourful image (many colour channels). Since CNN have been shown to be very good in image processing it was probable that they would also be good at this task. All other systems described in literature, use either image processing ML models or are heavily based on them, except for Google's GraphCast[8].



(Figure 21, original) The diagram shows: a snapshot of the world with the caption 'Day 0', a small square on that snapshot that points to a CNN, and a line from the CNN to a point on another snapshot captioned 'Day +1'

Data pre-processing

The system uses the same 'snapshot' parquet files described in the methodology.

The model and training

The model consists of a convoluted neural network of size 7x7 that represents a window of 17.5° longitude and latitude. The CFL condition suggests that this window of approximately 1,942 Km latitude, and similar but varying longitudinal width can predict a weather front moving at up to 40 km/h (see appendix C 1). In NWP the CFL condition describe a maximum wind speed, but as the system is considering cells further away. it is possible to model a much higher wind speed, this is because high wind speeds are usually not omnidirectional and it is easy to account for this as it is not a strictly cellular system.

Its important to understand that during tuning it was found the models hyperparameters other than kernel size were only marginally affecting the quality of the outputs when they were producing a very small size model; as such, finding a kernel size that was large enough to model fast moving weather systems, but not large enough to impair the main criteria for testing.

Following the guidance given in tensor flow documentation and tutorials, combined with knowledge of the limitations of weather predictions, provided a starting point. With further

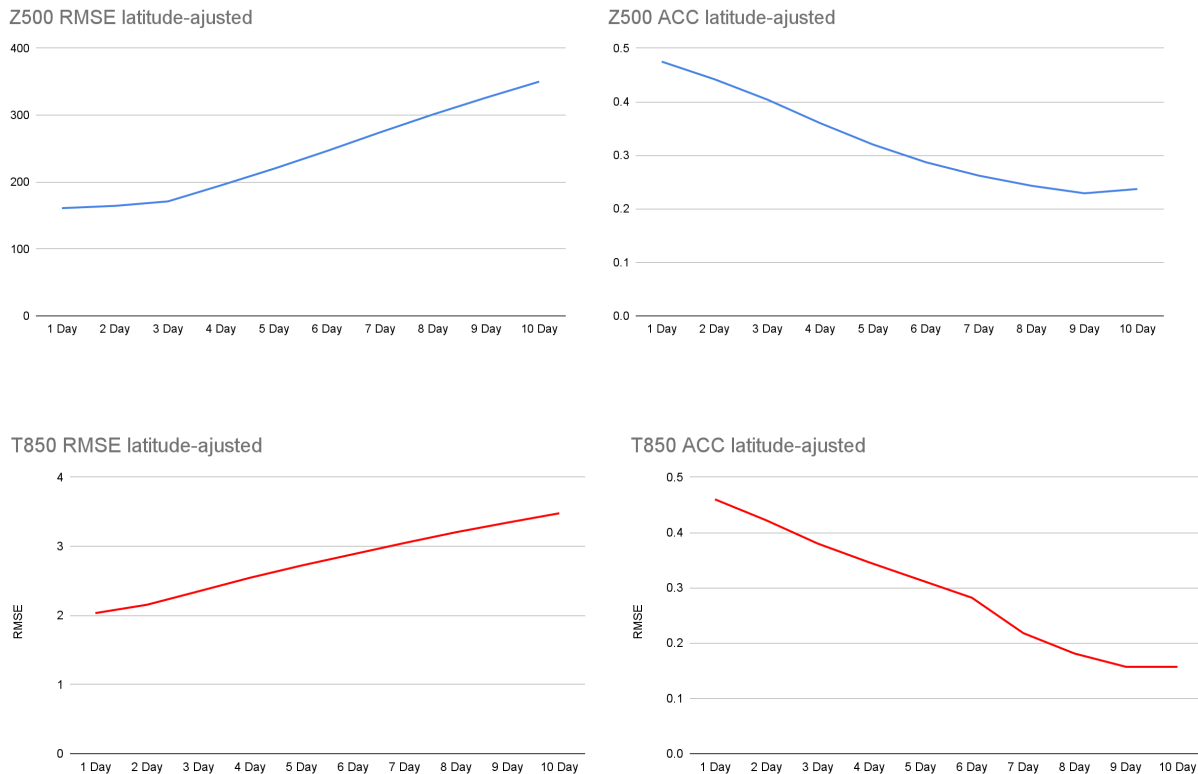
tuning and testing of the parameters of the model to find a set of hyperparameters was found that work well with the given limitations.

Model parameters

The exact parameters of the model layers are:

- Input layer:
 - 7x7 points, 25 values per point (1225)
- Layer 1:
 - 2D convolutional, 32 filters, 7x7 kernel size, ReLU activation, Same Padding
- Layer 2:
 - 2D convolutional, 64 filters, 7x7 kernel size, ReLU activation, Same Padding
- Layer 3:
 - 2D convolutional, 128 filters, 7x7 kernel size, ReLU activation, Same Padding
- Layer 4:
 - Transposed 2D convolutional Layer, 64 filters, 7x7 kernel size, ReLU activation, Same Padding
- Layer 5:
 - Transposed 2D convolutional Layer, 32 filters, 7x7 kernel size, ReLU activation, Same Padding
- Layer 6:
 - Transposed 2D convolutional Layer, (num_fetures) 32 filters, 7x7 kernel size, ReLU activation, Same Padding
- Output layer:
 - 1 Point, 25 values

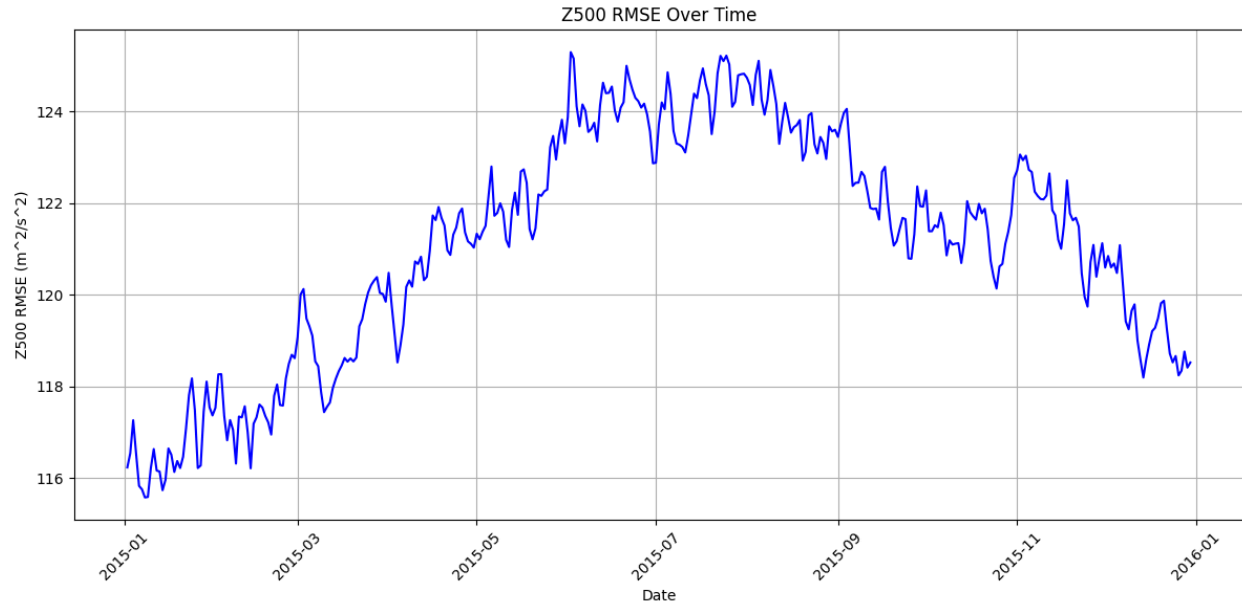
Results



(Figure 22, original) Charts showing: RMSE for Z500 up to 10 days lead time averaged over 4 years and 10 months, ACC for Z500 up to 10 days lead time averaged over 4 years and 10 months, RMSE for T850 up to 10 days lead time averaged over 4 years and 10 months, ACC for T850 up to 10 days lead time averaged over 4 years and 10 months.

These results show a huge improvement over the previous model, with a reasonable Z500 and T850 RMSE being given for the first few days, and demonstrating clear skill in its predictions all the way out to the maximum 10 day lead time.

While evaluating this model a metric observed for validation was the error by day the of year. While this metric is used to validate that accuracy metrics are correct, it tells an interesting story.



(Figure 23, original) Figure showing Z500 RMSE by day of year in 2015

There is clearly a trend of higher RMSE in the middle of the calendar year. As the southern hemisphere has significantly more ocean than the northern hemisphere, and oceans take much longer to change temperatures, the global average atmospheric temperatures are determined by the season in the northern hemisphere, despite the southern hemisphere being in the opposite season. This may suggest that the temperatures in the training data were lower than the temperatures in the testing data [28], and the error experienced is worse with higher than average temperatures.

The results are better than expected. Papers describing a similar approach, but using larger models, with more training, and significantly more data showed that even these models struggled [3]. It is possible that this is due to the fact that it's easy to train a mediocre model but difficult to train a good one. Comparison of these findings with other models will be considered in the conclusion.

The largest limitation for the skill of the predictions produced by this model would be the kernel size, looking at a larger window would have a marginal effect on the quality of predictions however it would require more data to get to that increased level of quality. Short of waiting 40 years for more data to be produced, the way to increase training data would be to shorten the lead time of one time step, thus increasing the number of time steps that can be used for training. However from CFL conditions we know that a decreased timestep time means the distance needed to be evaluated reduces, so we are back where we began with a shorter time step. This again could be improved by a shorter time step, this process could be continued until only adjacent cells are being considered.

This would be equivalent to a NWP system where the PDEs are replaced with an NN. while there might be some problems with the PDEs used in these models and an NN that has similar performance may be found, the point of MLWP is to improve weather prediction beyond what is currently possible and this seems like a step in the wrong direction. The quality of predictions could be increased by using a higher resolution dataset.

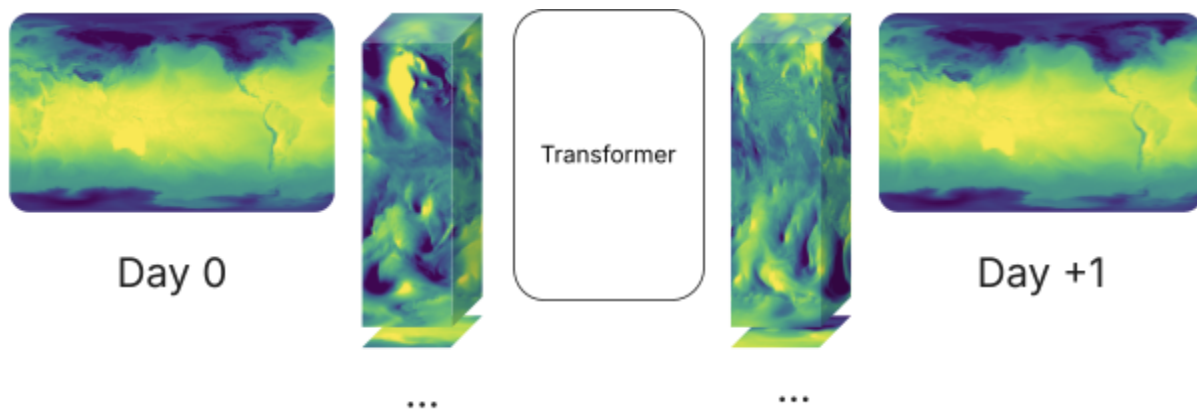
Conclusion

This architecture is a feasible model for some weather prediction types so shows some promise. Continual logical improvements would result in a system where the PDE's of NWP are replaced with ML, likely resulting in poorer performance for little or no computational improvement. Improving the model from the current state would only achieve trivial improvements so it is not obvious how this system would be better than NWP.

This video shows a smaller CNN I trained running: <https://youtu.be/wHGRG7PzSQc>

5.3 Patch Embedding Transformer Based

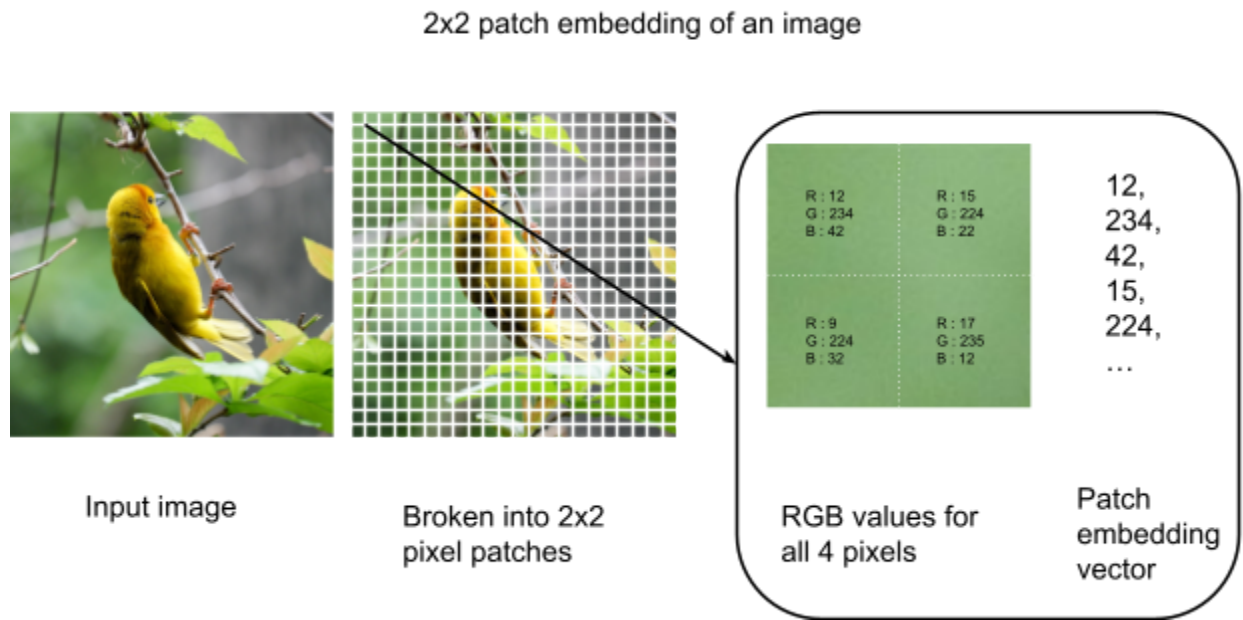
Patch embedding is a system used for image processing with transformers, this is an obvious choice if using transformers for MLWP. Patch embedding has been used in image processing, and Earth's weather as described in a gridded dataset is equivalent to a multi channel image. Typically the channels in an image represent red, green, and blue, there is little difference for them to represent the variables being modeled in a graph transformer instead of colours. Researchers at Huawei created the first MLWP model to beat traditional NWP models using this method. Diagram below shows a basic system outline;



(Figure 24, original) The diagram shows: A snapshot of the Earth's weather with the caption 'Day 0', A 3d cube with a 2d plane beneath showing the patch embedding of a part of the atmosphere and an epsilon beneath, a transformer, Another patch embedding, and a snapshot of the Earth's weather with the caption 'Day +1'

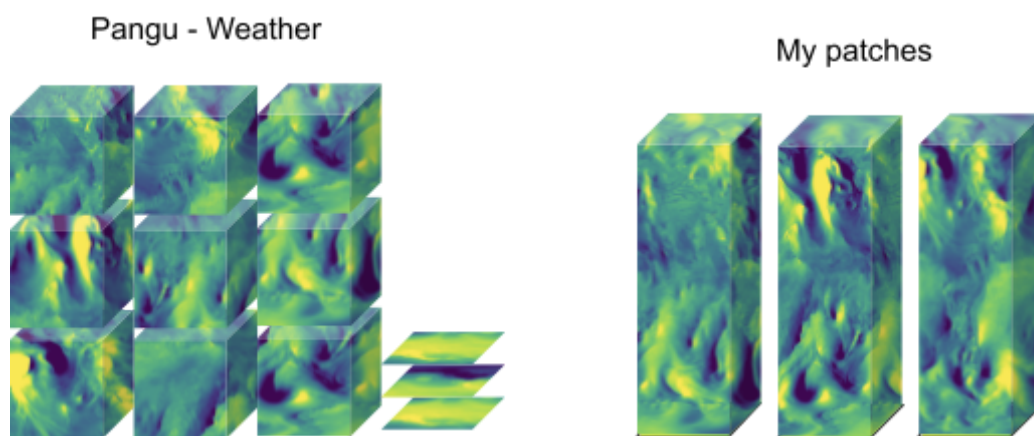
Patch embedding is a system of converting a gridded piece of data (typically an image), into a list of vectors. A 1x1 patch embedding would produce a list of vectors, with each dimension in a vector corresponding directly to a variable in the data point; a 1x2 patch embedding would result in a list of vectors where each vector is the concatenation of every set of 2 vectors from previously. As you scale up the size of the patches you get less vectors but the vectors are larger.

The diagram below shows how patch embedding works:



(Figure 25, original) Diagram showing: an input image, this image is broken down into lots of small patches of size 2x2 pixels (as this is a 2x2 patch embedding), for each pixel in the patch the RGB value is taken in order and these are used to construct the vector; repeat for all patches to get all embeddings. Diagram by me, source image [18].

The system employed by Huawei uses a normal 2D patch-embedding system for the surface variables using 4x4 patches ($1^\circ \times 1^\circ$ as the data is at a 0.25° resolution) and an additional system for atmospheric variables that uses a 2x4x4 patch embedding, these are merged to make the input for the transformer. The model trained here differs from the Huawei system, in that it embeds the whole atmosphere and surface in one, as opposed to doing them separately and then merging them. This simplifies the model and makes training slightly easier.



(Figure 26, original) Diagram showing Pangu Wether's combination of 3D and 2D patch embedding vs my version of a 2D patch embedding of 3D parameters though having all parameters for a given longitude and latitude in the same patch.

Model parameters

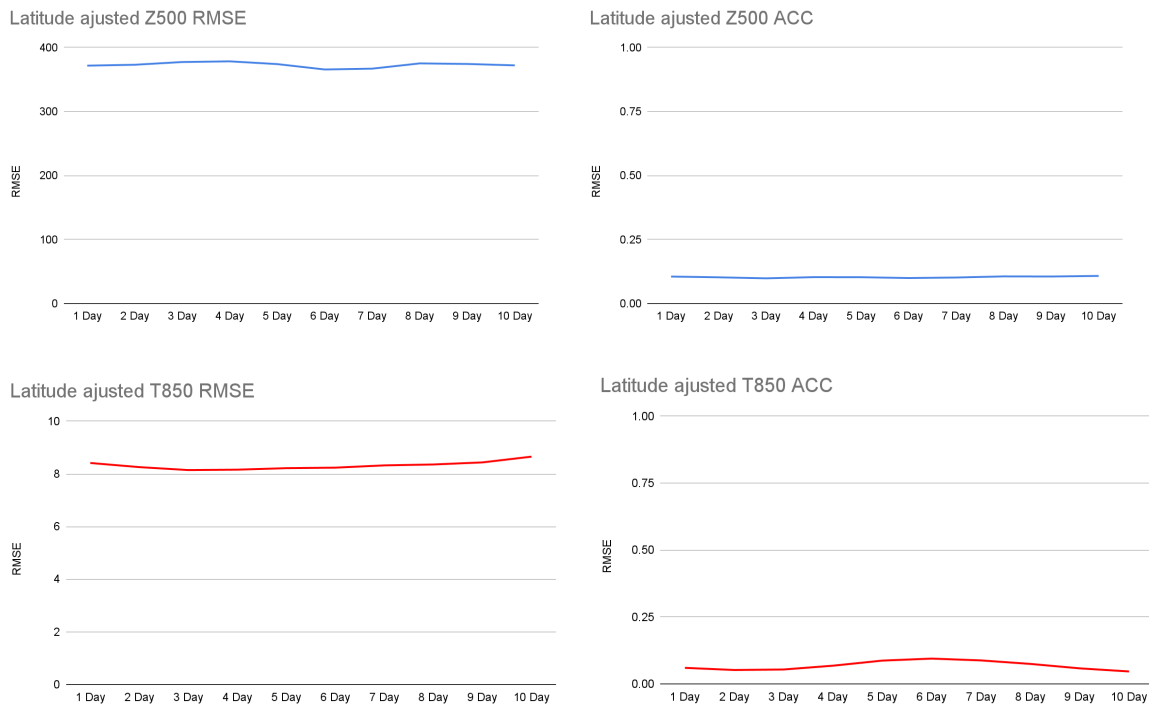
Fine tuning of the model has not had a drastic effect on the outputs except with very low values (for dimensionality, number of heads, layers, etc). The transformer parameters are inspired by other image to image patch embedding transformers.

Pytorch was used for this implementation and not TF as it was found that there is more useful information on vision transformers implemented in Pytorch over tensor flow.

Unfortunately this model is very difficult to train, the limitations forced the choice between 1x1 patches and small model size, or 2x2 patches and an even smaller model. The 1x1 model had slightly better performance so it was the chosen model. The exact parameters of the model is as described below:

- Input Layer:
 - 29 dimensional patches, 1x1 patches with each point having 25 features and
- Embedding layer:
 - Produces 32 dimensional embeddings of each patch
- Transformer encoder:
 - 1 layer, with: 32 dimensions, 2 heads
- Output Layer
 - Takes in 32 dimensions and decodes into 29 dimensions of the model

Results



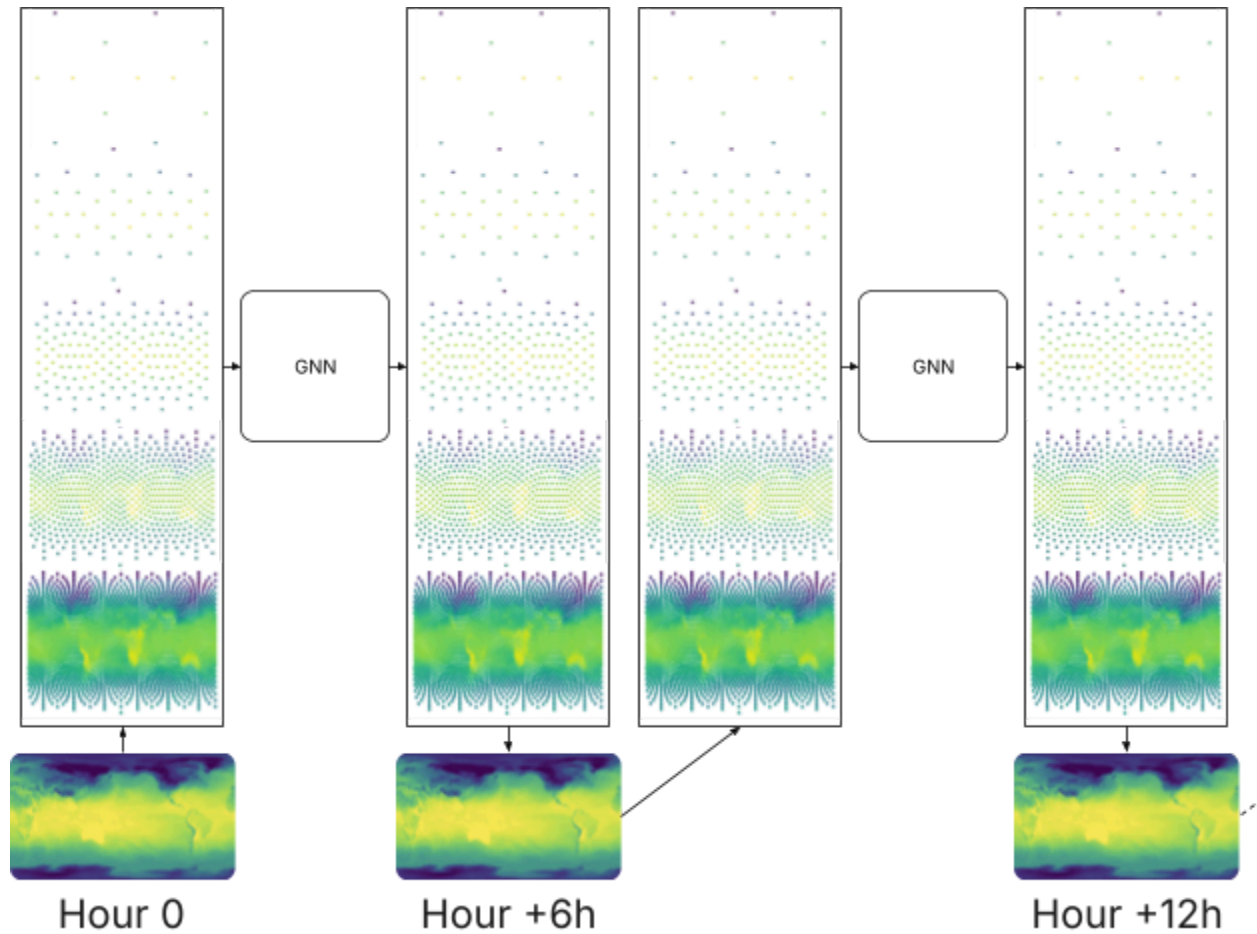
(Figure 27, original) Charts showing: RMSE for Z500 up to 10 days lead time averaged over 4 years and 10 months, ACC for Z500 up to 10 days lead time averaged over 4 years and 10 months, RMSE for Z500 up to 10 days lead time averaged over 4 years and 10 months, ACC for Z500 up to 10 days lead time averaged over 4 years and 10 months.

The results produced by this model are poor, the performance is unacceptable showing consistently high RMSE and consistently low ACC. The results for Z500 are not as bad as those for temperature, and there is a clear attempt to predict it accurately. However the initial hit is quite bad and it's unable to improve.

More training alone would not improve this model. Size constraints are problematic and would require a larger model before any improvement. The model would likely improve with the speed increases gained from having patches that are larger than 1x1, but the increased input vector sizes would increase the memory demands dramatically.

5.4 Graphical Neural Network

The GNN system described in this section is loosely based on GraphCast. As a quick recap the GraphCast paper analysed in 3.4 describes a system of weather prediction wherein a snapshot is encoded onto multiple meshes. These meshes are used as inputs for a graphical neural network where each point takes in the data from itself and adjacent points and is imputed into a deep graphical neural network where the data for that point in 6 hours time is predicted. This is then decoded into a snapshot of a 6 hour prediction, then repeated for a desired lead time.



(Figure 28, original) Diagram of the graphcast system, showing a snapshot of hour 0, the snapshot is encoded into a number of meshes of increasing subdivisions, the meshes are parsed into a GNN, a number of different meshes are outputted, the meshes are decoded into a snapshot of 'Hour +6h'. This system is repeated from snapshot 'hour +6h' to snapshot 'hour +12h'

Rationale of GraphCast design decisions and how they can be adapted for the needs of this model

There are no similar or comparable approaches to this unique system, therefore analyzing the design decisions taken throughout is paramount. The GraphCast system described by Google is an impressive idea that is clearly functional. While an open source model is available and information about the final decisions are known, the reasoning behind why these decisions were made is not known. Some of the time a contradictory opinion to the reason is given, and other times there is no rational opinion. What can be learnt from secondary sources closer to the development needs to be investigated. As well as what can be derived from current understanding of the fundamental limitations of weather prediction calculations and what can be found from my own experimentation.

Single mesh system

In the paper it is stated that “*The processor uses 16 unshared GNN layers to perform learned message-passing on the multimesh, enabling efficient local and long-range information propagation with few message passing steps.*”[2] (page 2). This implies that the primary advantage of a multimesh is its ability to communicate large and small scale features. However that is contradicted by one of the authors of the paper, Ferran Allet in his talk on the development of GraphCast [8] where he shows that the multimesh sees only a small improvement on pressure (geopotential and mean sea level pressure) and actually shows worse performance in higher-level values.

This slight improvement is likely due to the fact the model with all meshes has double the training data. Higher frequency systems like wind and temperature are still able to be learned to a near equivalent quality despite the decrease in data, but the lowest frequency variables (pressure and pressure derivatives like geopotential) see a slight increase from having more training data.

This is not surprising given an understanding of the fundamental principles of weather prediction. Since 1928 the CFL condition has been known to be the limiting factor for the relationship between the time-step length and fastest moving phenomena [23]. This is a mathematically proven limitation and it shouldn't matter if the model predictions are coming from partial differential equations or ML models; nor whether the measurements are being passed to adjacent cells in a voxel grid geometry, or adjacent vertices on a graph.

A storm moving at 40 MPH, will change position by 40 miles for every 1 hour of prediction time, if you look less than 40 miles you're not going to be able to model correctly as you can't 'see' the storm, looking further is unnecessary. (Note: a storm's wind speed is different to how fast a storm is moving).

Mesh size

There is no discussion in the GraphCast paper on the reasoning behind the size of the mesh other than one sentence: “*The multimesh contains the 40,962 nodes from the highest-resolution mesh (which is roughly 1/25 the number of latitude-longitude grid points at 0.25°)*” [2] (page 2).

However the research carried out in this study has found the main limiting factor and the relationship is the number of subdivisions in the most subdivided graph and the resolution of the dataset that the graph is sampling from.

The mesh encoding of the GraphCast system is for 7 meshes from 0 to 6 subdivisions; 6 subdivisions are possible as they are using a 0.25° dataset. However, using a GNN system with a 2.5° dataset, is limited to a lower number of subdivisions.

The limiting factor with subdivisions is the equator (as described in 2.1, points are more dense toward the poles). If there are more points along the equator on the graph than on the input data, then there are some points which cannot be encoded with accurate information. There is a further limit that if there are many points along the equator, and they are sampling from drastically different numbers of points in the data. The models can be highly sensitive to the different quality of encoding between points and can decrease the quality of predictions.

The Google system uses 256 equatorial points sampling from 1439 longitudinal points (See appendix B 1, 2), this resource has only used a dataset with 143 longitudes so will be limited to 3 subdivides. Using 4 subdivides creates large differences in the number of points sampled.

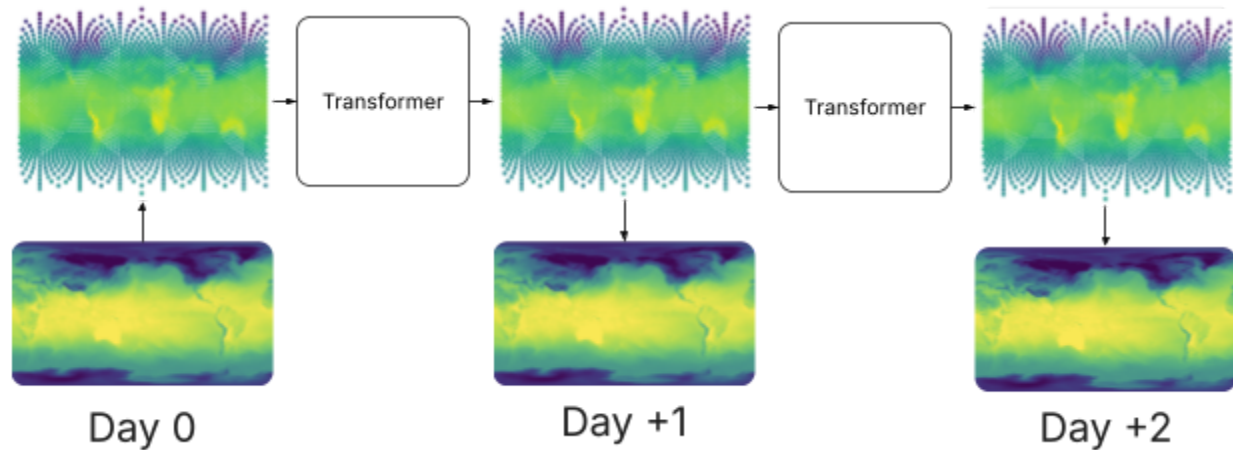
Time step change

As with all the models, a once daily 24 hour timestep dataset is being used. This gives predictions in 24 hour timesteps.

Direct mesh-to-mesh auto recursion

In the GraphCast paper, each mesh set is decoded back into a snapshot before being encoded again into a mesh; this adds error to the system, however, it is necessary in a multi-mesh system. By using a single mesh in the GNN model this step can be skipped and the output mesh can be used as an input mesh for the next step.

See the diagram below for a reference:



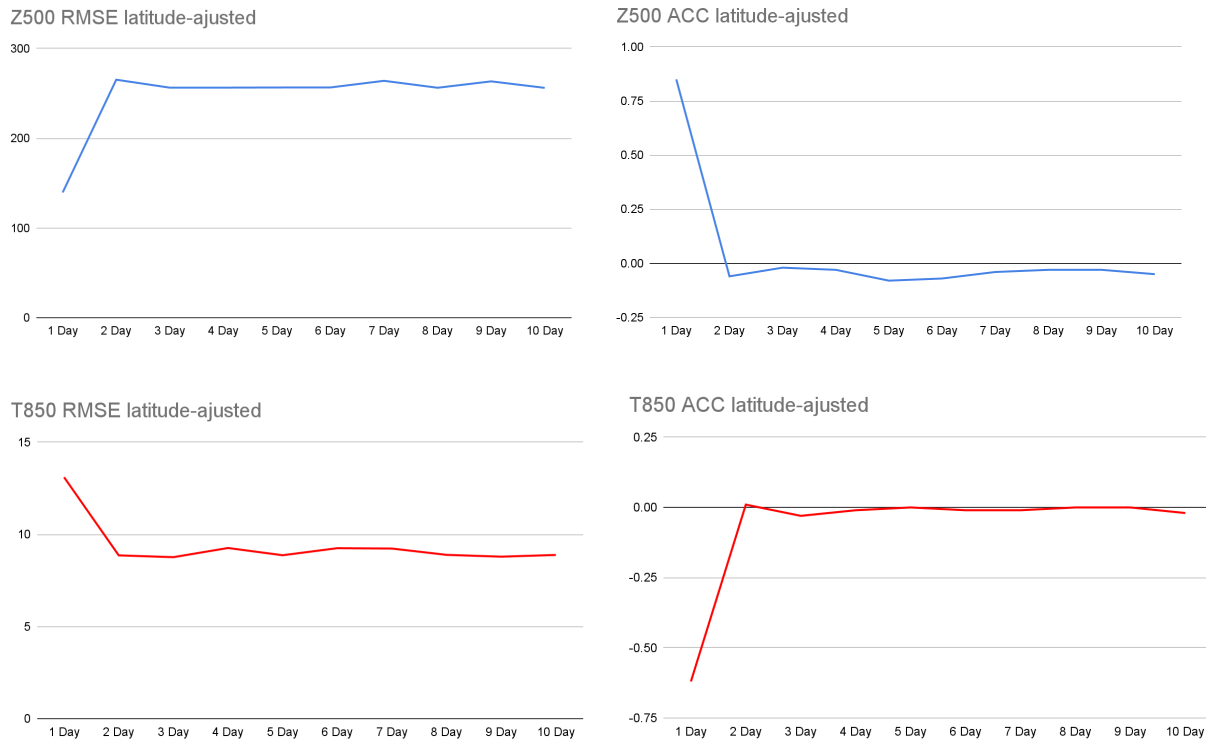
(Figure 29, original) Diagram showing the path of the data in my version of the graphics neural network transformer system, a snapshot of 'Day 0' becomes a graphical embedding of itself, this embedding is passed into a transformer which outputs a different graphical embedding, which is then converted into a snapshot of 'day+1'

Model parameters

The model design is described as:

- Input layer:
 - Shape (7, 25). Up to 7, 25 dimensional vectors. The transformer can accept 7 points (6 adjacent and central point) or 6 points (5 adjacent and target)
- Layer 1:
 - Multy head attention layer: 4 heads, 16 key dimensions
- Layer 2:
 - Layer normalization: epsilon = 1e-6
- Layer 3:
 - Dense layer: 64 units, ReLU
- Layer 5:
 - Dropout layer: 0.1 dropout rate
- Output layer:
 - Dense layer: 25 dimension vector

Results



(Figure 30, original) Charts showing: RMSE for Z500 up to 10 days lead time averaged over 4 years and 10 months, ACC for Z500 up to 10 days lead time averaged over 4 years and 10 months, RMSE for Z500 up to 10 days lead time averaged over 4 years and 10 months, ACC for Z500 up to 10 days lead time averaged over 4 years and 10 months.

The results for Z500 and T850 show two different stories. The initial z500 prediction is very good, however it soon devolves, giving very poor accuracy. The initial acc value of over nearly 0.8 would be considered highly accurate and is comparable to the accuracy of a 5 day forecast in 2000, all acc values after this are negative. This would suggest, given the accuracy of the information, the model is able to predict Z500 accurately, but that the output of other variables may be less accurate, as the model is likely relying on other variables that are not being modeled accurately. This problem has been seen in previous instances, although not to this extent. The solution to this problem is to spend time fine tuning at the end of training using model outputs as the inputs. If this change were to be implemented, it is likely that the model would learn to distrust atmospheric variables that have high error rates and trust the better predicted pressure variables.

T850 is the opposite, with an initial poor prediction following predictions data then becomes average. The reason that this model would be struggling to predict temperature is, that unlike pressure, it is much higher frequency data and the downsampling required to mesh this problem

has exacerbated it (see diagram below). This would result in the data on temperature being less correlated with other variables (as it is not representative of the weather) and would result in poor performance. It is interesting that it improves after the first time step, suggesting that a randomised input is more likely to predict the mean and archives a better result than a good input predicting values incorrectly.



(Figure 31, original) Figure showing two temperature maps, one on a gridded system and one on a mesh system and how a cell of warm air is not visible on a gridded system.

Conclusion

This system is very good at predicting low frequency weather events given accurate input data, the best of all models tested so far. However it has extremely poor performance at higher frequency weather systems, this pollutes the data and means that no feasible predictions past 1 day can be made, for any of the variables. Moving forward, using a finer mesh, and fine tuning with previous predictions as inputs will mitigate these issues and improve performance.

6 Professional, legal, social, and ethical Issues

6.1 Professional and legal issues

All of the data used for the training came from one source [16] the NOAA GDA system NCEP/NCAR the data can be used freely by anyone, the only caveat given:

“If you acquire NCEP Global Data Assimilation System GDAS data products from PSL, we ask that you acknowledge us in your use of the data. This may be done by including text such as NCEP Global Data Assimilation System GDAS data provided by the NOAA PSL, Boulder, Colorado, USA, from their website at <https://psl.noaa.gov> in any documents or publications using these data. We would also appreciate receiving a copy of the relevant publications. This will help PSL to justify keeping the data freely available online in the future. Thank you!”- NOAA [16]

As I have been diligent in explaining where I got my data, crediting NOAA, saying precisely what data I used, and what I did, I have reached all legal thresholds for the use of the data. All other libraries I used have been open source and free to use.

6.2 Social and Ethical issues

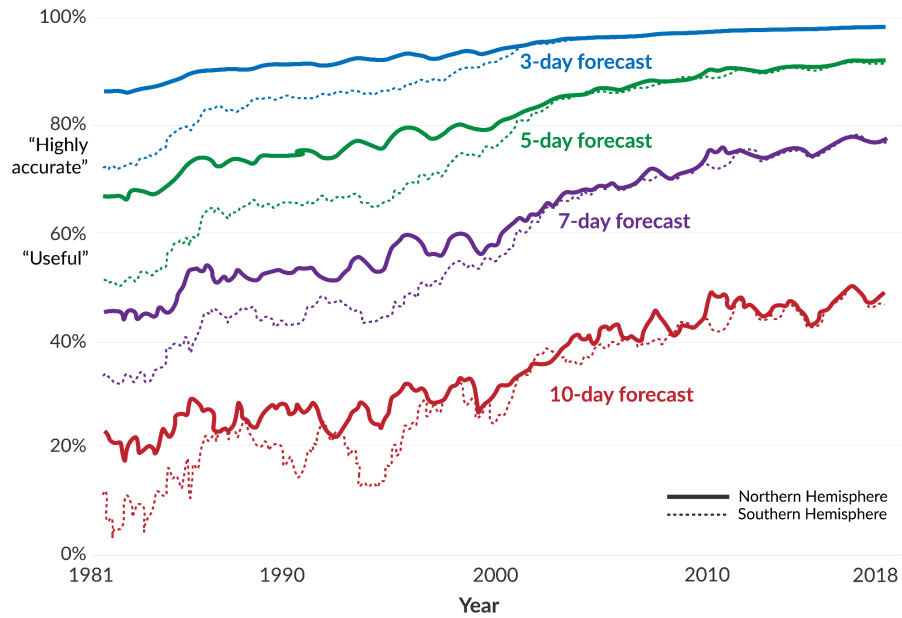
From 2000 to 2019, approximately 5 million people per year died as a result of extreme temperatures alone [33]. This number does not include the additional deaths caused by extreme weather events such as floods, hurricanes, and droughts. As climate change worsens, the frequency and severity of such events are expected to increase.

Historically there has been a bias in weather prediction with poorer areas receiving worse, less accurate predictions (see diagram below). This gap has been closed in recent years with more global sensing solutions and global models becoming the ‘go to’ for medium range weather forecasting. I have continued the leveling of these models by creating a global model that optimizes for all areas on Earth equally.

The accuracy of weather forecasts has improved

Accuracy is measured as the difference between the forecast and subsequent weather. This is based on the '500 hPa geopotential height' which is a common meteorological metric used to measure air pressure.

Our World
in Data



Source: European Centre for Medium-Range Weather Forecasts (ECMWF).

Licensed under CC-BY by the author Hannah Ritchie.

(Figure 32, [27]) Showing disparity between accuracy of wether predictions in the northern hemisphere and wether prediction in the southern hemisphere and the disparity over time.

7 Evaluation

This evaluation considers the results of the models and overall findings of this research by:

- Looking at the results of different models and how they compare to one another
- Looking at the reasons behind these results
- Giving reasons on why a certain system would be useful going forward and what would need to be done to improve it

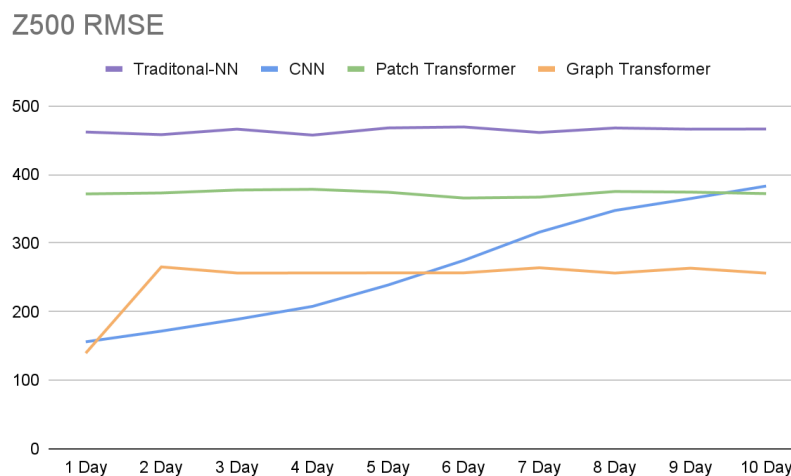
7.1 Results comparison

This subsection analyzes how the results from testing metrics compare between models. The reasoning behind the results are given in the following subsection.

Z500 RMSE

Data for the Z500 RMSE from a latitude adjusted RMSE averaged over 4 years and 10 months of weather data.

Chart showing data plotted:



(Figure 33, original) Chart shows the Z500 RMSE of the four models in this paper, over 10 days of lead time.

Takeaways from Z500 RMSE:

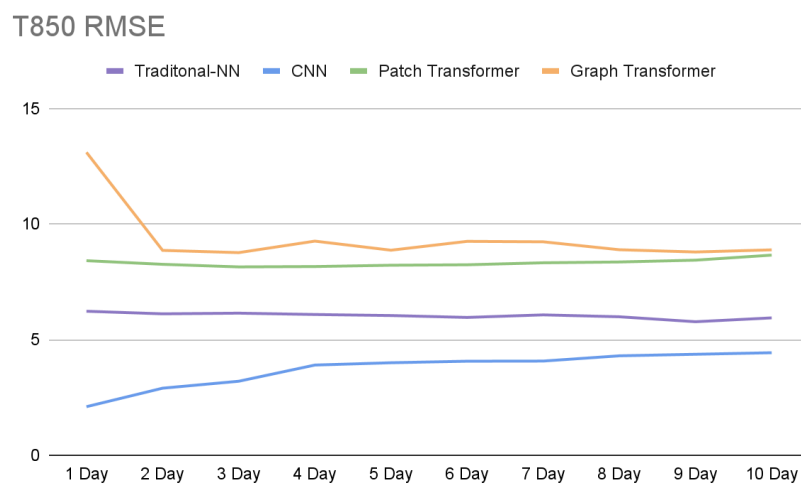
To begin with the graph transformer has a good RMSE, CNN has a similar but slightly worse RMSE, Traditional-NN and Patch transformer have bad RMSE. On the second day the RMSE for the Graph transformer jumps to a poor RMSE which is not as bad as Traditional-NN and Patch transformer, this RMSE is better as the more coarse grid averages values and means less

extreme high error values are produced, Traditional-NN and Patch transformer continue with bad RMSE, CNN sees a slight increase in RMSE but is still considered good. Over days 3-10 Traditional-NN, Patch transformer, and graph transformer continue to be bad, CNN reduces in quality moving toward a similar quality around day 10.

T850 RMSE

Data for the T850 RMSE from a latitude adjusted RMSE averaged over 4 years and 10 months of weather data.

Chart showing data plotted:



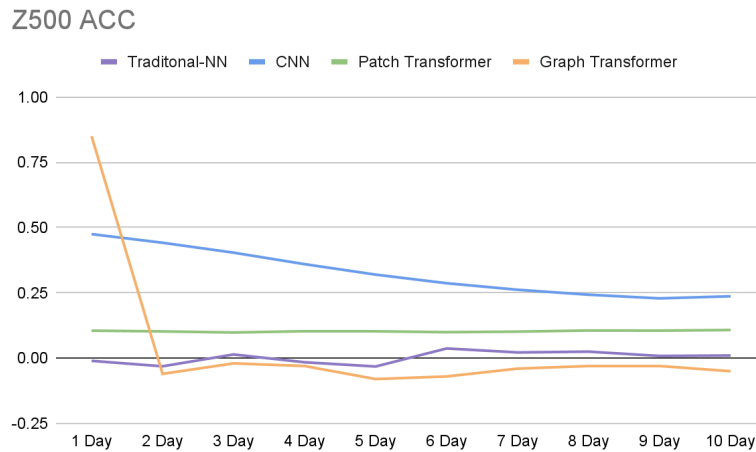
(Figure 34, original) Chart shows the T850 RMSE of the four models in this paper; over 10 days of lead time.

CNN has good RMSE which gradually gets worse over time, Traditional-NN and Patch transformer have poor RMSE which remains poor. The graph transformer starts with a very poor RMSE and reduces to a poor RMSE from day 2 onward.

Z500 ACC

Data for the Z500 ACC from a latitude adjusted RMSE averaged over 4 years and 10 months of weather data.

Chart showing data plotted:



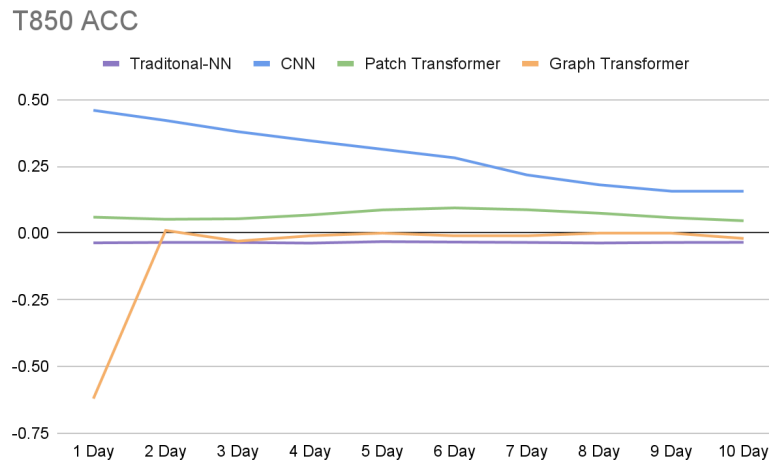
(Figure 35, original) Chart shows the Z500 ACC of the four models in this paper, over 10 days of lead time.

Graph transformer has the highest ACC of any model at the beginning, the 1 day ACC would be considered “highly accurate” however all subsequent ACC values are very poor, CNN starts with a reasonable ACC and sees a slow reduction over time. Patch transformer and Traditional NN are poor throughout.

T850 ACC

Data for the T850 ACC from a latitude adjusted RMSE averaged over 4 years and 10 months of weather data.

Chart showing data plotted:



(Figure 36, original) Chart shows the T850 ACC of the four models in this paper, over 10 days of lead time.

CNN starts with a reasonable ACC and reduces in quality over time. Graph transformer starts with an extremely poor ACC on day 1 and reduces to average afterward, Patch transformer and

Traditional NN are poor throughout.

Conclusion

These results show that the CNN is the only reasonable system to use; the Graph transformer can be good in the short term but gets universally poor after that, the other two are universally poor throughout.

7.2 Results and recommendations

These results may not accurately reflect the true performance of the underlying architectures when deployed at scale; particularly the two transformer base systems, which struggled and produced poor predictions. Unlike their performance seen in landmark papers, it was clear during development that the models were not going to produce the full performance they were capable of, due to the limitations placed on them to give a fair comparison.

The patch based transformer was too small, and the model and its training underfit the data, this would not be a problem for future researchers. When this model successfully used a much larger model size with training, it was also trained for 700,000 hours which is 700,000 times longer than this model was trained for.

A similar issue was encountered with the coarse nature of the mesh that was used by the graph based transformer. The resolution of the mesh could not be made any higher, given the resolution of the underlying data set that it was encoding. The model was good at modelling large low frequency weather changes given good input data, it struggled at modelling high frequency weather patterns; polluting the output and preventing the model from continuing to accurately predict the lower frequency weather patterns. It is impressive that it is still able to predict low frequency weather patterns like pressure despite poor performance with high frequency patterns like temperature and wind.

The model recommendation based on performance would logically be CNNs, it is true that the comparative performance across all lead times is impressive but it is likely that the model will struggle to improve much further. Instead it is recommended that future developments focus on a machine learning architecture built upon a mesh, with the caveat that the underlying mesh is of a high resolution. It is imperative that the area of the mesh considered for the prediction of a given point relates to the length of a times step considering CFL conditions.

8 Conclusion

It is very difficult to make an informed decision on how to go forward with new developments, or give a research proposal with good reasoning about how the research will result in improvements; because there are numerous models, different amounts of training for each model, lack of documentation on what is important to tune, and no guidance on how to improve these models.

This paper has helped solve this issue by: evaluating on equal standing the performance of different model architectures, evaluating which parameters are important for the quality of the results, and giving advice on how models could be improved.

This paper has:

- Reviewed the background information needed to understand the basics of weather prediction.
- Reviewed literature on the meteorological principles of weather prediction, current MLPW papers and models, and the effective implementation of ML architectures.
- Defined a methodology for making, training, and testing MLPW models.
- Implemented and tested several different MLPW architectures.
- Evaluated the results from the tests of the MLPW architectures, and made conclusions.

Finally to conclude a brief recommendation on the future direction of research in MLPW:

Research proposal

Future research should be focused on using larger transformer systems, on a dataset that adjusts for global curvature and the scaling issues by being mesh based; a large model may require more hardware to train and infer but this would still be small compared to NWP requirements.

To alleviate issues with mesh resolution, the mesh should be created from data assimilation not encoded from a gridded dataset, this will allow a high resolution mesh to be accurately made, which is important for modeling high frequency weather systems.

The length of a time step and the area considered to predict a point should be related by CFL conditions; furthermore a long time step of 12 or 24 hours should be used, as longer time steps have been shown to improve accuracy over the time frames important for medium-range forecasts.

References

- [1] Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X. and Tian, Q., 2023. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970), pp.533-538.
- [2] Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W. and Merose, A., 2023. Learning skillful medium-range global weather forecasting. *Science*, 382(6677), pp.1416-1421.
- [3] Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K. and Hassanzadeh, P., 2022. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*.
- [4] Chen, L., Zhong, X., Zhang, F., Cheng, Y., Xu, Y., Qi, Y. and Li, H., 2023. FuXi: A cascade machine learning forecasting system for 15-day global weather forecast. *npj Climate and Atmospheric Science*, 6(1), p.190.
- [5] Chen, K., Han, T., Gong, J., Bai, L., Ling, F., Luo, J.J., Chen, X., Ma, L., Zhang, T., Su, R. and Ci, Y., 2023. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*.
- [6] Price, I., Sanchez-Gonzalez, A., Alet, F., Andersson, T.R., El-Kadi, A., Masters, D., Ewalds, T., Stott, J., Mohamed, S., Battaglia, P. and Lam, R., 2023. Gencast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796*.
- [7] Lorenz, Edward. "Predictability: Does the flap of a butterfly's wing in Brazil set off a tornado in Texas?." (1972).
- [8] Alet, F., 2025. GraphCast: Learning Skillful Medium-Range Global Weather Forecasting. Presented at Google DeepMind Modeling Talks.
- Available at:
<https://sites.google.com/modelingtalks.org/entry/graphcast-learning-skillful-medium-range-global-weather-forecasting>
- [9] Cambridge Philosophical Society, 2025. Professor Richard Turner - The Quiet AI Revolution in Weather Forecasting.
- [lecture] Recoding available at: <https://www.youtube.com/watch?v=JGn18WH0d6s&t=1616s>

[10] Frnda, J., Durica, M., Rozhon, J., Vojtekova, M., Nedoma, J. and Martinek, R., 2022. ECMWF short-term prediction accuracy improvement by deep learning. Scientific Reports, 12(1), p.7898.

[11] Cheon, M., Choi, Y.H., Kang, S.Y., Choi, Y., Lee, J.G. and Kang, D., 2024. Karina: An efficient deep learning model for global weather forecast. arXiv preprint arXiv:2403.10555.

[12] Jouppi, N., Kurian, G., Li, S., Ma, P., Nagarajan, R., Nai, L., Patil, N., Subramanian, S., Swing, A., Towles, B. and Young, C., 2023, June. Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In Proceedings of the 50th Annual International Symposium on Computer Architecture (pp. 1-14).

[13] Lambda Labs, 2025. NVIDIA A100 vs V100 Benchmarks.

Available at: <https://lambdalabs.com/blog/nvidia-a100-vs-v100-benchmarks>

[14] Simmons, A, 1995. The skill of 500hPa height forecasts, the European Centre for medium range weather forecasts.

Available at:

<https://www.ecmwf.int/sites/default/files/elibrary/1995/12230-skill-500-hpa-height-forecasts.pdf>

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. (2017) ‘Attention is all you need’, Advances in Neural Information Processing Systems, 30, pp. 5998–6008.

[16] NOAA Physical Sciences Laboratory (n.d.) NCEP/NCAR Reanalysis 1.

Available at: <https://psl.noaa.gov/data/gridded/data.ncep.html>

[17] European Centre for Medium-Range Weather Forecasts (ECMWF), 2025. ECMWF Reanalysis v5 (ERA5).

Available at: <https://www.ecmwf.int/en/forecasts/dataset/ecmwf-reanalysis-v5>

[18] Carey Rose, 2017. Fujifilm X-A3 sample gallery.

Available at: <https://www.dpreview.com/samples/3925134721/fujifilm-x-a3-sample-gallery>

[19] ACMAD (African Centre of Meteorological Applications for Development), 2025. Weekly Weather Forecast – 500 hPa steering level explanation.

Available at: <https://acmad.org/index.php/weekly-weather-forecast/>

[20] Pennsylvania State University. (n.d.). Geostrophic Wind.

Available at: <https://www.e-education.psu.edu/meteo300/node/711>

[21] Peel, M. (2024) ‘Google DeepMind hits new milestone in AI weather forecasting’, Financial Times.

Available at: <https://www.ft.com/content/4a2f0a5c-d23d-4f6a-9c89-32c1d76be19e>

[22] Hazelton, A., Gao, K., Bender, M., Cowan, L., Alaka Jr, G.J., Kaltenbaugh, A., Gramer, L., Zhang, X., Harris, L., Marchok, T. and Morin, M., 2022. Performance of 2020 real-time Atlantic hurricane forecasts from high-resolution global-nested hurricane models: HAFS-globalnest and GFDL T-SHiELD. *Weather and Forecasting*, 37(1), pp.143-161.

[23] Courant, R., Friedrichs, K. and Lewy, H., 1928. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische annalen*, 100(1), pp.32-74.

English translation: <https://web.stanford.edu/class/cme324/classics/courant-friedrichs-lewy.pdf>

[24] Charlton-Perez, A.J., Dacre, H.F., Driscoll, S., Gray, S.L., Harvey, B., Harvey, N.J., Hunt, K.M., Lee, R.W., Swaminathan, R., Vandaele, R. and Volonté, A., 2024. Do AI models produce better weather forecasts than physics-based models? A quantitative evaluation case study of Storm Ciarán. *npj Climate and Atmospheric Science*, 7(1), p.93.

[25] European Centre for Medium-Range Weather Forecasts (2025) ‘Pangu Medium-Range Temperature Chart’

Available at:

https://charts.ecmwf.int/products/pangu_medium-t-z?base_time=202504010000&level=1000&projection=opencharts_europe&valid_time=202504080000

[26] Goldberg, M., 1937. A class of multi-symmetric polyhedra. *Tohoku Mathematical Journal*, First Series, 43, pp.104-108.

[27] Ritchie, H., 2024. Weather forecasts have become much more accurate; we now need to make them available to everyone. Our World in Data.

Available at: <https://ourworldindata.org/weather-forecasts>

[28] Copernicus Climate Change Service (2024) Why do we keep talking about 1.5°C and 2°C above pre-industrial era?

Available at:

<https://climate.copernicus.eu/why-do-we-keep-talking-about-15degc-and-2degc-above-pre-industrial-era>

[29] 3Blue1Brown (2022) But what is a convolution?

Available at: <https://www.youtube.com/watch?v=KuXjwB4LzSA>

Code: https://github.com/3b1b/videos/tree/master/_2022/convolutions

[30] Saha, S. (2018) A comprehensive guide to Convolutional Neural Networks — the ELI5 way. Saturn Cloud.

Available at:

<https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>

[31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

[32] University of Birmingham weather station (live feed)

Available at: <https://app.konectgds.com/kiosk/c8c55446-9283-4acc-94c1-4a2ff81b6186>

[33] Gasparrini, A., Guo, Y., Hashizume, M., Kinney, P.L., Petkova, E.P., Lavigne, E., Zanobetti, A., Schwartz, J.D. and Armstrong, B., 2015. Mortality risk attributable to high and low ambient temperature: a multicountry observational study. The Lancet, 386(9991), pp.369–375.

[34] Ritchie, H. and Roser, M., 2021. Natural disasters. Our World in Data.

Available at: <https://ourworldindata.org/natural-disasters>

[35] ECMWF, 2023. Weather parameters used in forecasts. European Centre for Medium-Range Weather Forecasts.

Available at: <https://www.ecmwf.int/en/forecasts/documentation-and-support/parameter-list>

[36] ECMWF, 2023. Forecast verification: skill scores. European Centre for Medium-Range Weather Forecasts.

Available at:

<https://www.ecmwf.int/en/forecasts/documentation-and-support/forecast-verification/skill-scores>

[37] NASA, 2023. Extreme Weather and Climate Change. NASA Science.

Available at: <https://science.nasa.gov/climate-change/extreme-weather/>

[38] OpenSnow, 2024. Diurnal Heating, Explained.

Available at: <https://opensnow.com/news/post/diurnal-heating-explained>

[39] Dueben, P.D. and Bauer, P., 2018. Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10), pp.3999-4009.

[40] ECMWF, 2022. High-performance computing facility. European Centre for Medium-Range Weather Forecasts.

Available at:

<https://www.ecmwf.int/en/computing/our-facilities/high-performance-computing-facility>

[41] Ralph, F.M., Dettinger, M.D., Cairns, M.M., Galarneau, T.J. and Eylander, J., 2018. Defining “atmospheric river”: How the Glossary of Meteorology helped resolve a debate. *Bulletin of the American Meteorological Society*, 99(4), pp.837-839.

[42] Guan, B., Molotch, N.P., Waliser, D.E., Fetzner, E.J. and Neiman, P.J., 2013. The 2010/2011 snow season in California's Sierra Nevada: Role of atmospheric rivers and modes of large-scale variability. *Water Resources Research*, 49(10), pp.6731-6743.

[43] Kim, D.K., Suezawa, T., Mega, T., Kikuchi, H., Yoshikawa, E., Baron, P. and Ushio, T., 2021. Improving precipitation nowcasting using a three-dimensional convolutional neural network model from Multi Parameter Phased Array Weather Radar observations. *Atmospheric Research*, 262, p.105774.

Appendices

A

1

‘Operational data assimilation’ is the technique used to assimilate data in real-time. It is similar but not identical to ‘Data reassimilation’ which can use data that takes longer between measurement and collection i.e. satellites and ‘citizen weather volunteers’. Data reassimilation datasets like the ECMWF’s ERA5 (ECMWF Reassimilation v5) are used to train MLWP models.

The goal of data assimilation is to take in data from an ‘assimilation window’ (6 hours for Operational data assimilation and 12 hours for ERA5) and output a continuous weather forecast that has minimal variation from observed values. The ECMWF ‘4D-Var’ does this by taking a short-range forecast from the last assimilation window and then systematically adjusting the forecast to minimize differences between the forecast and the measurements, when this is done the forecast is treated as ‘ground truth’ and we can use the model to create initial conditions for any forecast we want to make that starts in the assimilation window.

There is also, on the state-of-the-art systems, an additional step to estimate the accuracy of this model, an ‘ensemble’ of 51 lower-resolution data assimilations are run 1 with the same input data and 50 with noise introduced to represent inaccuracy in the readings. The differences between these 51 simulations are mapped to plot the areas where there are larger uncertainties in the predictions. This map can then be used to add noise to the high-resolution assimilation that is representative of the actual uncertainty.

Data assimilation represents around half of the computation that goes into modern weather forecasting. MLWP papers brag about 10,000x 100,000x computational savings; however, this is not true as they still rely on this as of yet still numerically calculated system.

It's also worth noting that because MLWP models are trained on reassimilation datasets like ERA5 that have a 12-hour assimilation window and initial conditions pulled from this context window can rely on data that would not be recorded until 12 hours in the future the models can ‘learn’ and reverse engineer the assimilation algorithms, because of this it is important to make predictions that cross an assimilation window. Google’s Graphcast compared 6-hour predictions in the last 3 hours of an assimilation window to mirror the ECMWF’s HRES, which takes its initial conditions from the middle of a 6-hour assimilation window.

NWP weather forecasts are programmed to predict the state of the atmosphere at time $t+\Delta t$ from the state of the atmosphere at time t . They can then use the prediction from $t+\Delta t$ to predict the state at time $t+2\Delta t$ and repeat until they have predicted out to the desired lead time. In the case of ECMWF's HRES the time step Δt is 12 minutes. (The time step is usually much longer for MLWP being a few hours). This produces a deterministic forecast that can only tell us what it thinks the weather is going to be, it doesn't comment on the likelihood of something happening or give us any indication of other possibilities, this has to be extrapolated in the data processing step. Deterministic MLWP compare themselves to this to make their claims about being better than traditional methods but this is not the state of the art. This is the best prediction for the first (~ 1) day. But for longer time scales 'ensemble' forecasts are used.

Ensemble forecasts are a pretty simple idea on paper, run multiple models from ever so slightly different initial conditions to get a range of possible outcomes. The ECMWF's ERA system does this by running 51 prediction models analogous to HRES one as a baseline on the Initial starting conditions and 50 more with slightly different initial conditions and slightly modified model physics. The initial conditions are altered by applying Gaussian noise to the high-resolution initial conditions from the data assimilation based on the uncertainty map created by the data assimilation ensemble. This would introduce more noise over sparsely measured areas i.e. Pacific Ocean, Southern Ocean, and Antarctica and less over Europe and the US which have dense high-quality measurements.

B

1

Equation for number of points along the equator in a subdivided icosahedron.

N = number of points, S = number of subdivisions

$$N = 4 \cdot 2^S$$

Equation for number of vertices in a subdivided icosahedron.

N = number of points, S = number of subdivisions

$$N = 10S^2 + 2$$

2

Equation for number of longitudes on a coordinate grid by grid resolution

N = number of longitudes, R = resolution

$$N = 360/R - 1$$

Equation for number of latitudes on a coordinate grid by grid resolution

N = number of latitudes, R = resolution

$$N = 180/R$$

Examples of Resolutions for Different Systems:

- ERA5 (Used by Pangu-weather, Graphcast, and Gencast) **0.25°**
- HRES (Latest ECMWF) **0.1°**
- NCEP GDAS (used in all my models) **2.5°**

3

Number of measurements:

- 4 years 10 months = 2128 days tested
- 2.5° resolution = $143 \times 72 = 10286$ points tested per day
- 4 measurements (Z500 / T850) (RMSE / ACC) measured
- 10 days lead time

$$2128 \times 10286 \times 4 \times 10 = 875544320$$

C

1

CFL condition limitations for multiple dimensions where:

C = characteristic wave speed (for us this is speed of a weather front), Δx = spatial resolution (distance between simulate points), Δt = time step

$$\Delta t = \frac{1}{c} \left(\sum_{i=1}^d \frac{1}{(\Delta x_i)^2} \right)^{-1/2}$$

For evenly spaced spatial resolution it can be simplified to

$$\Delta t = \frac{\Delta x}{c\sqrt{d}}$$

Figures and Tables

Figure / Table number	Description	Origin and sources
Figure 1	Points on a sphere on a lon/lat grid with colour representing distance to the next closest point.	Original, made with: /FigureMaker/points_on_sphere.ipynb
Figure 2	Diagram of the atmospheric boundary layer.	From Pennsylvania State University, College of Earth and Mineral Sciences [20]
Figure 3	Storm tracking of Pangu-weather.	Slide from [9] (screenshots of [1])
Figure 4	Time schedule of hybrid weather forecast.	Original, made with google sheets drawing tool
Figure 5	Pangu-wether vs IFS vs Ground truth.	All three taken from [1]
Figure 6	Pangu-Weather storm tracking.	From [1]
Figure 7	GraphCast system diagram.	From [2]
Figure 8	Subdivided icosahedrons (0-3 subdivisions).	Original, made with: visualisations/subdivided_Icosohedrons.ipynb
Figure 9	GraphCast vs HRES performance.	From [2]
Figure 10	GraphCast prediction schedule vs assimilation windows.	From [8]
Figure 11	GraphCast performance increases with later training.	From [2]
Figure 12	GenCast system diagram.	From [6]
Figure 13	RMSE of z500 and t850 gencast and other models	All four taken from [6]
Figure 14	CNN diagram	From [30]
Figure 15	A convolution being applied	From [29]

Figure 16	Traditional-NN system diagram	Original, made with google sheets drawing tool, ‘snapshot’ screenshot of ERA5 visualization
Figure 17	Points of prediction	Original, made with google sheets drawing tool, earth screenshot from google earth
Figure 18/19	Traditional-NN - Z500, T850: RMSE and ACC	Original
Figure 20	Average global temperature above pre industrial levels	From [28]
Figure 21	CNN system diagram	Original, made with google sheets drawing tool, ‘snapshot’ screenshot of ERA5 visualization
Figure 22	CNN - Z500, T850: RMSE and ACC	Original
Figure 23	1 day z500 RMSE by date in year (2015)	Original
Figure 24	Patch embedding transformer system diagram	Original, made with google sheets drawing tool, ‘snapshot’ and ‘patch’ screenshot of ERA5 visualization
Figure 25	Patch embedding example	Original, made with google sheets drawing tool, bird image from [18]
Figure 26	Patch embedding example implementation	Original, made with google sheets drawing tool, ‘patch’ screenshot of ERA5 visualization
Figure 27	Patch embedding transformer - RMSE and ACC	Original
Figure 28	Graphcast system diagram	Original, made with google sheets drawing tool, ‘snapshot’ screenshot of ERA5 visualization
Figure 29	Graph transformer system diagram	Original, made with google sheets drawing tool, ‘snapshot’ screenshot of ERA5 visualization
Figure 30	Graph transformer - Z500, T850: RMSE and ACC	Original

Figure 31	Grided vs meshed temperature data	Original, made with google sheets drawing tool
Figure 32	ACC of predictions of weather forecasts over time	From [27]
Figure 33	Z500 RMSE for all 4 models	Original
Figure 34	T850 RMSE for all 4 models	Original
Figure 35	Z500 ACC for all 4 models	Original
Figure 36	T850 ACC for all 4 models	Original
Table 1	weather forecast measurements and their units.	From [35]
Table 2	Variables and their levels used in FourCastNet.	From [3]
Table 3	Specs for training	Original