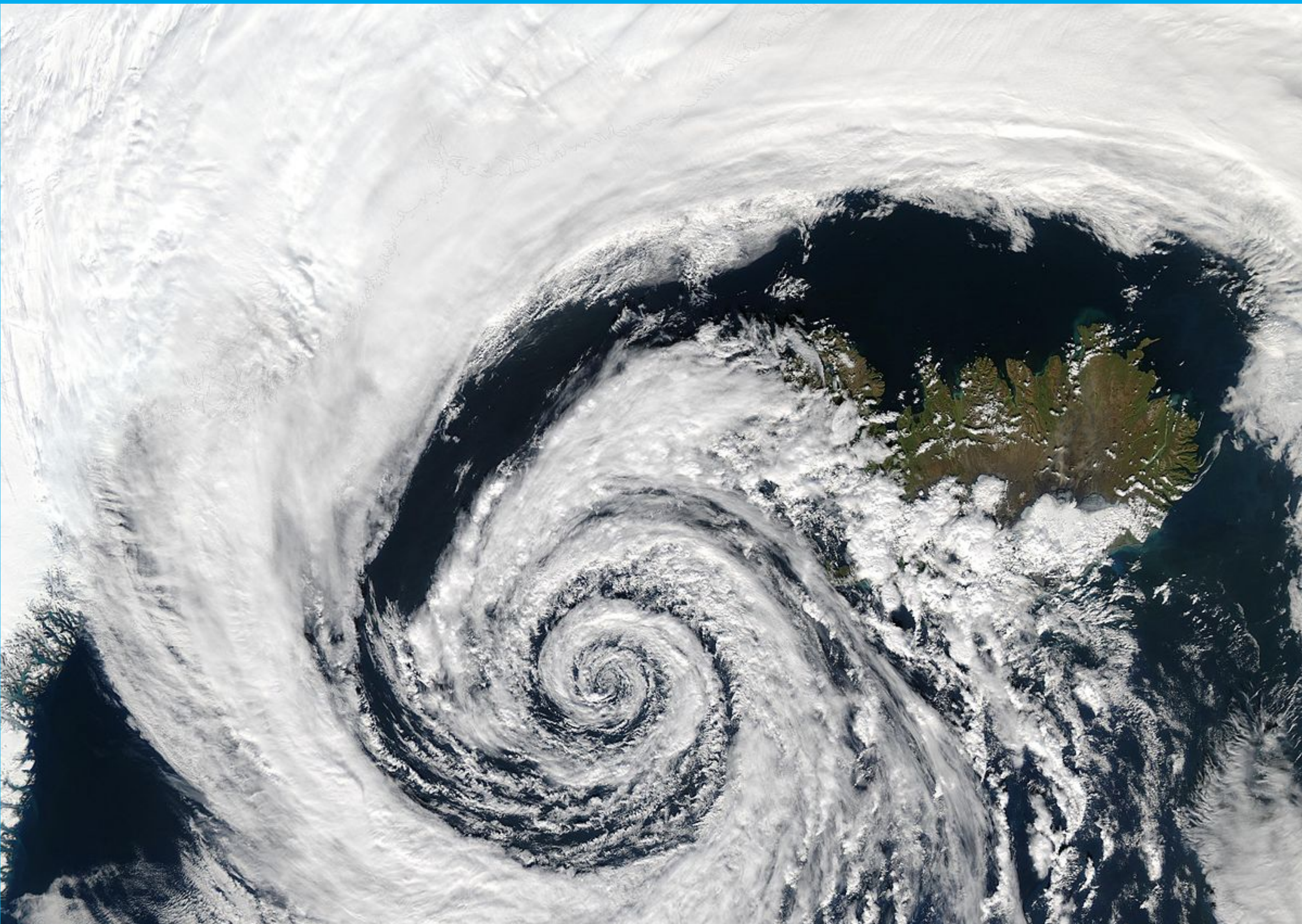
Creation of new Extra-Tropical Cyclone fields in the North Atlantic with Generative Adversarial Networks

A deep learning framework to generate new synthetic atmospheric variables fields from the learned original sample data distribution

Filippo Dainelli

MSc Civil Engineering - Environmental Engineering Science track



Creation of new Extra-Tropical Cyclone fields in the North Atlantic with Generative Adversarial Networks

A deep learning framework to generate new synthetic atmospheric variables fields from the learned original sample data distribution

by

Filippo Dainelli

to obtain the degree of Master of Science in Civil Engineering
at the Delft University of Technology,
to be defended publicly on 17th December 2020 i.

Student number:	4771435	
Project duration:	February, 2020 – December, 2020	
Committee:	Dr. Riccardo Taormina (Chair)	TU Delft
	Dr. ir. Jeremy Bricker	TU Delft
	Dr. Franziska Glassmeier	TU Delft
	Dr. Sofia Caires	Deltares

Contents

Abstract

Extra-Tropical Cyclones (ETCs) are major storm system ruling and influencing the atmospheric structure at mid-latitudes. These events are usually characterized by strong winds and heavy precipitation and cause considerable storm surges with threatening wave systems for coastal regions. The possibility to simulate these storms or to increase the amount of significant data available is crucial to optimize risk assessment and risk management for construction projects and territorial plans which might get damaged by events of this kind. The project addresses the possibility to learn the distribution of cyclones atmospheric fields of pressure, wind and precipitation in the North Atlantic by training a Generative Adversarial Network (GAN). The ETCs tracks are extracted from the ERA5 reanalysis dataset in the domain with boundaries 0°-90°N, 70°W-20°E and period going from 1^{st} January 1979 to 1^{st} January 2020. A GAN tries to learn the distribution of a training set based on a game theoretic scenario where two network competes against each other, the generator and the discriminator. The former is trained to generate new examples which are plausible and similar to the real ones by having as input a vector of random Gaussian values. The random vectors domain is called latent space. The latter learns to distinguish whether an example is coming from the dataset distribution or not. The competition set by the game scenario makes the network improve until the counterfeits are indistinguishable form the original. The generative models trained on the ETCs dataset are validated to understand if they are able to generate new samples of fields presenting similar atmospheric characteristics to those of the original dataset. To train the GAN two different loss function are considered, the Wasserstein distance and the Cramèr distance. The Cramèr Gan (CGAN) shows better performance in representing the distribution of the atmospheric fields, generating images that on average look similar to the original ones. The Wasserstein GAN (WGAN) behaviour shows poor performance in representing the precipitation in general, but it is able to similarly reproduce the values distribution for what concerns pressure and wind. The images generated by the WGAN have many differences compared to the original ones and are very blurry with particular data structures that looks like artefacts built by the network. The atmospheric structure of the images generated by the CGAN is investigated by considering 4 cyclones as case study and comparing the frames of their tracks to those of synthetic tracks generated by linear interpolation. The linear interpolation is performed between the random vectors generating the most similar images to the initial and final snapshot of the original track. The interpolated images show interesting features in the similarity with the original track, which suggest that the network has learned a representation of the ETCs fields that is promising for further investigation.

Preface

This report is the result of my master thesis project and concludes my Master of Science in Environmental Engineering studies at the Delft University of Technology in Delft, The Netherlands. This work was developed together with the Hydraulic department of Deltares.

2020 has been a difficult year for everyone but I must admit that I consider myself very lucky for the opportunity of working on this project. During the last 10 months I expanded my knowledge of cyclones events, mid-latitudes atmospheric regime and deep learning methods. After my Thesis I find myself with a new skill set in the artificial intelligence sector, which is relevant for my background and career.
Of course I could not face such a project trusting only myself and the help of the people who surrounded me was crucial for my success.
I want to thank Sofia and Jeremy for the opportunity to work on this topic, the freedom to choose the path I preferred and the precious advice. I want to thank Franziska for the enthusiasm and the ability to point out crucial elements of discussion in my work. I am grateful to Riccardo for the time, the resources and the passion he put in the project. You assisted me with precious clarifications and help, making me able to truly understand the methods I have applied.
My thesis, master degree and experience in Delft would have never been possible without the love of my family and friends. Mamma, Babbo I have your support since October 1996, you have always pushed me to give the best of me in a life full of opportunities. Fede, Pancho thank you for your admiration and encouragement, you are my first supporters. Cecilia, Elena, Giorgio, Giulio, Henar, Luca, Luigi, Nicolò, Stefano, Vlad thank you for the good moments, the happiness and the motivation you shared with me while I was working on this project.

Filippo Dainelli
Delft, December 2020

Introduction

Extra-tropical Cyclones (ETCs) are main weather features of the mid-latitudes, where they play an important role in the hydrological cycle (Cotton and Anthes (2011)). The warm conveyor belts associated with these cyclones, which are airstreams of warm and moist air rising from the boundary layer to the troposphere along the cyclone's warm front, produce around half of the precipitation during winter in middle and high latitudes (Eckhardt et al. (2004)). Coastal regions in the northern and mid-Atlantic can experience more than 20 extra-tropical storms every year (Service (2019)). Because of their spatial extension and the variability of their characteristics in space and time a variety of clouds and mesoscale precipitation systems can be found within these storms. In fact, ETCs also impact the global energy balance and may also produce a wide range of severe weather (Cotton and Anthes (2011)). Often their passage is associated with strong winds and temperature changes (Ulbrich et al. (2009)). Extra-tropical storm intensity and size can vary significantly, they can bring calm weather with moderate and scattered rain and surface winds of 16-32 kilometers per hour, or they can bring cold temperatures and severe weather with torrential rain and wind exceeding 120 kilometers per hour (Service (2019)).

Intense mid-latitudes storms represent a considerable threat for infrastructures, services and inhabitants of the middle and high latitudes. Not only for the high speed winds or the heavy rainfall they transport but also for their influence on oceans' phenomena. In fact, consequences of ETCs could be wave systems with serious wave height and energy transport and/or storm surges. Especially the shorelines are exposed to this natural hazards. The most extreme single-basin floods are known to be caused by ETCs, but also multi-basin flooding has been shown to be linked with this mid-latitudes systems (De Luca et al. (2017), Lavers et al. (2011)).
The chance to perform simulations of ETCs would be very helpful both for forecasting events which could impact production and population both for extreme events analysis and risk assessment for design and planning. Thanks to reanalysis data and algorithms of identification and tracking for ETCs nowadays decent datasets can be utilized, especially for the North Atlantic. However to increase the goodness of estimation and to reduce the errors, the possibility to have a bigger dataset is very tempting. Generating synthetic cyclones' fields using a simulation tool will rather increase the amount of data available. Methods to achieve this are lacking in literature, especially for the extreme complexity of representing the mid-latitudes cyclones. In the scientific literature there are many more examples and studies on simulation of Tropical Cyclones (TCs) rather then ETCs. As a matter of fact examples of ETCs' simulation are rare to find and they are always circumscribed to specific region around the globe. That is because ETCs are extremely complicated to represent in a parametric simulation and the reasons are different. They are much more common features of the atmosphere than the TCs so because of their higher frequency of appearance they are difficult to separate as a specific event from the climatological evolution of the atmosphere. They can range considerably in size, having the smallest cyclones with diameters of about 100 kilometers and the biggest with diameters higher than 1000 kilometers. Their shape and their structure can be diverse, also because they are often asymmetric. Their velocity of propagation can vary significantly from an event to another but also during a cyclone's life cycle. The weather fronts involved with ETCs brings other complications in their representations. In fact, event if a general structure of the fronts can be described, the habitual presence of two different kinds of fronts (warm and cold) which usually evolve into an occluded front, involves different weather conditions and brings instability to the atmospheric state. Even though is known which are the fronts involved is difficult to locate them and predict their evolution along with the rainfall connected to this atmospheric features. So, the variability of cyclones' characteristics along their evolution makes it complicated to simulate their development in space and time knowing the initial state conditions. In addition, the variability in their structural features and in the atmospheric state related to them makes it complicated to represent them using a set of summary parameters.
Because of the previously listed characteristics, opting for a parameterized representations of ETCs involves different problems and finding a set of physical meaning parameters seems particularly difficult for such an event. Opting for a method which does not consider a set of parameter defined a priori to represent and generate mid-latitudes storms fields looks like a more promising and easier approach to model such phenomena. Instead of having an engineering approach where is the modeler who defines a representation for

the atmospheric events, let the model learn by itself the structure underlying ETCs.

The intuition behind this project is to use a method able itself to learn the representation of such a complex class of events instead of having someone to codify it. For this reason, the representation-learning ability of deep learning methods is explored. Deep learning methods are believed to be able to learn the representation of the ETCs' related fields of pressure, wind and precipitation. The data used in the project are extracted from the ERA5 reanalysis dataset of the European Centre for Medium-Range Weather Forecasts (ECMWF) and the domain considered covers most of the North Atlantic and the European continent. Because of the extension of the domain and the structure of the data the atmospheric variables fields can be treated as images. Each grid point which is part of the domain is a pixel containing the values of the atmospheric variables at that location. The images might have a singular channel if one variable at time is considered in the representation, or they might present more than one channel if more than one variable is considered. We will refer to the first case as single-field representation and to the latter as multi-field representation. In the last five years a particular framework of deep learning has been very successful with image generation, the Generative Adversarial Network (GAN) (Goodfellow et al. (2014)). This class of network suits the application because given a training set, the algorithm learns to generate new data that have similar statistics to the data in the training set. GANs are generative methods able to learn the statistical distribution underlying the training data. Once they have learned it, they can generate a new sample of data which is different from the original one but comes from the same distribution. The purpose of this project is to verify if a GAN, trained on a sufficient number of storms fields, is able to reproduce new fields that have never occurred in the past but present an atmospheric structure similar to the originals. By consequence the research objective of the thesis is the following:

"Train a Generative Adversarial Network to generate synthetic fields of pressure, wind and precipitation for Extra-Tropical Cyclones events in the North Atlantic."

The learning framework of a GAN relies on and adversarial competition between two networks, a generator and a discriminator. The discriminator is trained to distinguish a data coming from the training sample (real) and a data coming from the generated one (fake). The generator learns to make samples which deceive the discriminator. The training progresses until the discriminator is not able to distinguish a real sample from a generated one anymore (Goodfellow et al. (2016)). The generator creates the synthetic data example starting from an input vector containing multiple random values extracted from a Gaussian distribution, the space from which the random noise is extracted is called latent space. Since the data are images is convenient to choose a Convolutional Neural Network (CNN) as architecture of the two networks (LeCun et al. (2015)).
Since their introduction in 2014 by Goodfellow et al. (2014) new architectures and frameworks of training have been introduced to improve the quality of images generation with GANs. Different studies have focused especially on the creation of new images of human faces, so representing people which actually do not exist, and the results obtained are very impressive (Karras et al. (2017), Karras et al. (2018), Karras et al. (2020), Liu and Tuzel (2016), Radford et al. (2015)). GANs have been also adopted to perform deep learning application concerning videos both for recognition tasks (e.g. action classification) and video generation tasks (e.g. future prediction). GAN models are able to generate frames of videos which have fairly realistic dynamics and motions and can capture the videos complexity and diversity (Clark et al. (2019), Vondrick et al. (2016)). However, GAN suffer from instability during training, mode collapse and they have difficulties in the generation of high resolution images. To try to avoid this issues in this application a particular architecture is considered, the Progressive Growing GAN (PGGAN) (Karras et al. (2017)). This recent method relies on training the network by increasing the resolution step by step, so the first iterations along the training data are done with low resolution and as the training continues the images resolution increases. In this way the network is able to first learn large scale structure and after shifts to low scale details, avoiding to learn all features simultaneously.

GANs' breakthroughs with images and frames generation has found a fertile environment to develop also in the meteorological forecasting field. Because of its ability to learn the distribution underlying the training data, its ability to handle images and the promising results in predicting the following frame given a previous states GANs have been applied for different purposes in atmospheric science problems. Cheng et al. (2020) applied GANs to eliminate artifacts in the down-scaling process from high-resolution climate prediction. Schmidt et al. (2020) adopted the GAN architecture to generate cloud reflectance fields conditioned on large scale meteorological variables. Bihlo (2020) trained a GAN to predict fields of atmospheric variables for the next 24 hours over Europe. Ruttgers et al. (2019) trained a GAN to predict track of typhoons using satellite images and meteorological data from a reanalysis database. Considering the development of GAN architectures and the training frameworks arose in recent years the purpose of this project represents just the first step in the direction of a more powerful tool. Starting from the resolution of the images used for the training up until implementing a deeper architectures for ETCs forecasting. In fact, the images considered in the project have 64x64 resolution, mainly because of training stability and lack of time to progress on higher resolutions for a thesis work. The model proposed tries to generate new fields of storms in the North Atlantic, but what about generating new tracks? That is even more interesting. However the problem complicates further because of the entry of the temporal aspect of the event and because of data availability. In fact, it is important to always keep in mind that this kind of networks necessitates of very large training dataset to output relevant results. For example, in Ruttgers et al. (2019) they used a total of 5'000'000 images. Another possible step is to combine a GAN model with another network to produce forecast of ETCs, which is similar to the work of Liu and Lee (2020). The authors considers the problem of weather forecasting as a video frame prediction problem and implement a model that utilises the Conditional GAN (CGAN) and the Convolutional LSTM (Conv-LSTM) to produce weather forecasts and handle the uncertainty in future frame prediction. A similar framework involving the GAN trained in this project could be organized for cyclones prediction, where a predictive learning module such a Conv-LSTM is coupled with the PGGAN to try to solve the difficulties that both the networks have in handling meteorological data. Training the GAN on ETCs tracks by implementing sequential modelling is a very interesting and valid development of the project. However it falls outside the time and the workload for a Thesis proposal. The focus of the project is predominantly oriented on analyzing the GAN representation ability of ETCs considering both the statistical distribution of the variables and the the atmospheric structure. A positive validation according to this aspects is very promising for further development of the network in learning the cyclone track evolution.

The project aims to address 4 main research questions, considering the influence of multi-channel images on the network training and the relationship between the latent space and the generated fields structure. Multi-field representation is investigated to understand if processing information concurrently helps the network to learn better the atmospheric structure, in this case the network is trained on multi-channel images where every channel corresponds to one of the variables. The latent space is examined to verify if there relationship between vectors and images generation is stable and meaningful for what concern the atmospheric structure. A stable relationship means that latent space vectors belonging to the same region show similar storm situations and that by moving coherently in the latent space physically meaningful situation are generated. The latter means that by going from a vector generating a known situation to another vector generating a known storm conditions it is expected that the intermediate vectors generate physically meaning fields with conditions related to the known ones. The main research questions that will be addressed are:

1. Is it possible to generate meaningful ETCs' atmospheric variable fields using a Progressive Growing GAN?
2. Can the combination of different variable fields improve the PGGAN learning of the atmospheric structure during ETCs?
3. How to verify consistency in the generated synthetic ETC fields? Is similarity in statistical distribution a good indicator? Is it possible to evaluate the atmospheric physical structure with quantitative metrics or just by qualitative comments?
4. How are latent space values influencing the generation of ETC fields? Is the relationship between latent space vectors and image generation stable and respecting the physical structure of the data?

This research is performed as graduation project for the Master in Civil Engineering at Delft University of Technology in cooperation with Deltares. Before moving directly straight to the project implementation, in chapter 1 are reported the necessary literature and theory background to better understand the project. It involves ETCs representation, structure and tracking, the activity of such events in the North Atlantic and a review of the basic concepts of deep learning, together with a more accurate explanation of the networks involved in this project. In chapter 2 the dataset utilized to train the network and the domain are presented, while in the following chapter 3, the architecture of the network and the training framework are discussed. In chapter 4 are shown the results obtained from the generator models trained on different atmospheric variables. The section comprehends also comments on the methods utilized in the project and how they have influenced the results, considering also previous work of GANs on images, and recommendations for future developments involving GAN and ETCs fields generation.

1

Literature study and theoretical background

ETCs are dominant weather systems of the mid-latitudes, along their paths they produce strong winds, precipitation and temperature changes. Information on cyclones' characteristics and paths are important to characterize the climate of a certain region but also to understand the variations of the local weather (Ulbrich et al. (2009)).
ETCs play a central role in the transport of heat and moisture from the tropics region to the polar regions. In fact, changes in frequency and in intensity for this kind of phenomena has a big impact on the climate of mid-latitudes (Bartholy et al. (2006)). To give and idea of their influence, according to Fraedrich et al. (1986), more than two-thirds of the precipitation during winter over Europe originate from the frontal systems of less than 15 cyclones. They can be considered a key resource for water supply in the sub-tropics region, but also they are linked with many natural hazard occurring at middle and high latitudes (wind damage, precipitation related flooding, storm surges and marine storminess) (Neu et al. (2013)). Thus, an analysis of cyclones activity and the possibility of evaluating their impacts on a certain region has a crucial value for risk assessment.

Machine learning has become increasingly utilized in the last two decades and the results achieved by its methods for a vast variety of problems are stunning. In the modern society machine learning is involved in different applications: web searches, recommendations on e-commerce websites, content filtering on social networks, identification of objects in images, transcription of speech into text and many others. A lot of these applications uses a class of techniques known as deep learning. Deep learning methods are a class of representation-learning models which decompose the problem of representation in multiple simpler levels, this is done by concatenating simple but non-linear modules where each is able to bring the representation from one level to a higher, more abstract one (LeCun et al. (2015)).

This chapter contains the necessary background information and the literature review for this MSc project. ETCs' definition and cycle of formation will be first discussed, together with the conceptual models used to describe such phenomena. Section 1.2 presents the methods adopted to identify and characterise ETCs. In section 1.3, the climatology and cyclones activity trends in the North Atlantic will be reviewed. In section 1.4, the structure of a deep learning algorithm is discussed. The section focuses on Convolutional Neural Networks and Generative Adversarial Networks, which are the two architectures employed in the project. To give a better idea on what is the advancement of neural network application in the meteorological fields some interesting examples will be discussed in section 1.5. Because cyclones identification and tracking algorithm were considered to create the ETCs dataset but not applied in the end, a review of this methods can be found in the Appendix.

1.1. Extra-tropical cyclone definition, formation and evolution

With the term extra-tropical cyclone a type of storm system is identified, these storms are formed in middle or high latitudes, in regions of large horizontal temperature variations called frontal zones (editors of Encyclopaedia Britannica (2020)). ETCs often contain a cold front that extends toward the equator for hundred miles (Merriam-Webster (2020)).

The middle-latitude atmosphere is characterized by large-scale horizontal temperature gradients and by westerly winds that normally increase with height throughout the troposphere. The gradients are driven inexorably by differential radiative heating between high and low latitudes while the winds are a consequence of the thermal wind relationship (Cotton and Anthes (2011)). Most of the time the poleward decrease of temperature is not uniform, instead it is usually concentrated in tight baroclinic zones or fronts. ETCs develop in these baroclinic zones because the polar fronts become unstable to wavelike perturbations (Charney (1947), Cotton and Anthes (2011)).

Extensive studies of ETCs over the past seventy years have led to a definition of different conceptual models that describe the structure and evolution of this atmospheric phenomenon (Semple (2003)). The structure and evolution was first described by the Norwegian cyclone model, which characterize the storm as viewed from the surface (Bjerknes and Solberg (1922)). Because of some inconsistencies between observations and the Norwegian cyclone model during the years the model has been refined different times and a new conceptual model was developed by Shapiro and Keyser (1990). Taking into account surface observations, satellite imagery, radar data and model output Browning and Roberts (1990) also described the structural evolution of an extra-tropical cyclone. They identified and described the evolution of cyclonic flows, which are the warm conveyor belt, the cold conveyor belt and the dry intrusion. However, different analysis of the structure and evolution of ETCs suggests that there is no universal cycle of ETCs but some general characteristics can be identified (Ulbrich et al. (2009)). Dacre et al. (2012) tries to identify them by introducing a tool which explores average characteristics of a cyclones' sample (section 1.1.4).

The initiation or strengthening of cyclonic circulation is described by the term *cyclogenesis*, while the broader evolution of the cyclone (its formation, deepening, occluding and filling up) is collectively referred to as its *life cycle* (Semple (2003)). The first step in describing a cyclonic depressions once it has been recognized is to identify inside its boundaries the frontal regions across which temperature, winds, cloud and precipitation are observed to vary. A front can be defined as a boundary region between two air masses with contrasting thermal and humidity characteristics (Semple (2003)).

Historically the study of mid-latitudes cyclones has been done following two different ways. *Single-body cyclogenesis* if it is assumed that cyclones develop from a single element in the synoptic scale flow, like a disturbance in the baroclinic region (section 1.1.1 and 1.1.2). *Multi-body cyclogenesis* assumes that cyclones are a consequence of the interactions between different disturbances in the synoptic scale flow (section 1.1.3) (Semple (2003)).

Independently from the model the advancement in technology in satellite imagery have made possible to identify cloud signature of the cyclogenesis (Semple (2003)). Cloud development in an ETCs starts with a band of dense and stratiform clouds mainly located on the cold size of the surface frotns (Fig. 1.1, stage 1). This frontal clouds are produced by the ascending air coming from the warm side of the front. North of the warm front the precipitation is widespread. With the cyclone's development the cloud layer expands to the west and southwest of the surface low centre (Fig. 1.1, stage 2). Together with the stratiform layer surrounding the fronts they form the head of the comma cloud (Fig. 1.1, stages 1 and 2). [h]The extension towards west and southwest continues with the development of the cyclone and produce shallow stratiform cloud within the boundary layer (Fig. 1.1, stage 3). Toghether with it a cloud-free region develops an intrude the cloud-head region. Finally the entire system becomes a large swirl of cloud surrounding the low centre (Carlson (1980), Semple (2003)).

1.1.1. Norwegian Cyclone model

The Norwegian model was developed almost 100 years ago and still remains useful as a simplified conceptual model, it is usually referred as the classical cyclone model (Cotton and Anthes (2011)).

Examining weather data from archive of the meteorological institutes in Norway, Sweden and Denmark Bjerknes (1919) highlights the characteristic features of the surface flow of a moving cyclone and of the Norwegian model. Every shifting cyclone has two lines of convergence which are distinguished by thermal properties. The first one lies on the right side of the cyclone's path and comes in to the centre from the front. The second one comes in from the rear of the cyclone and lies on the left side of the cyclone's path. They are respectively called *steering line* and *squall line* (Bjerknes (1919)). In Figure 1.2 a schematic representation of the Norwegian model and its flow structure is shown.

The two lines, because at the border of different air masses are intimately related to the distribution of temperature. The passage of the steering line will cause an increase of temperature, having warm air to the left (warm front). The passage of the squall line will cause a decrease of temperature, having cold air to the left

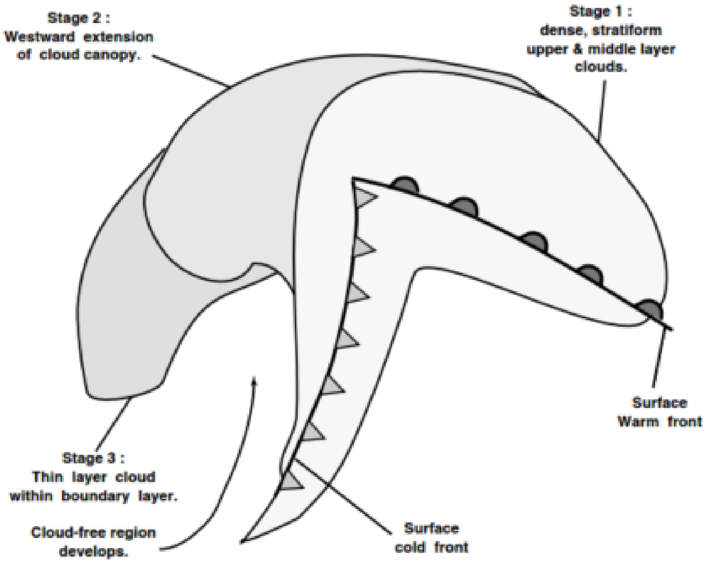


Figure 1.1: Evolution of cloud distribution in a developing cyclone (Semple (2003)).

(cold front) (Bjerknes (1919)).
Considering Figure 1.2 the characteristics features at the surface of a moving cyclone are: (a) a flow towards the centre of the cyclone following a spiral direction; (b) two lines of convergence which corresponds to a cold and a warm front; and (c) an air mass south of the cyclone centre called warm sector, because of the higher temperatures in respect of other cyclone's region (Volkert (1999)).

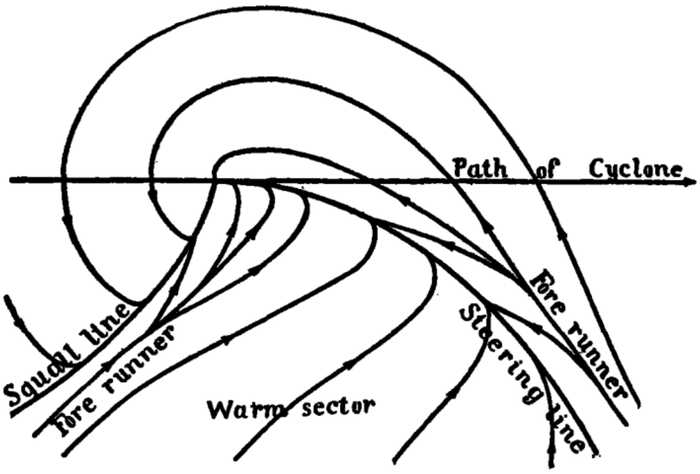


Figure 1.2: Lines of flow in a moving cyclone (Bjerknes (1919)).

Further development of the Norwegian model results in the evolution structure shown in Figure 1.3. Step I represents the development of the frontal wave at its beginning phase due to wavelike perturbations in the baroclinic region. Step II represents the development of the two fronts and the warm sector until the cyclone reaches its mature cyclogenesis phase (Step III) where the central pressure is at its minimum and the pressure gradient is the most steep. Step IV is the final frontal occlusion phase, north of the occluded front a seclusion of warm air from the warm sector can be find which remains trapped during the occlusion process (Cotton and Anthes (2011)).
The Norwegian model also includes a description of the vertical structure of the warm and cold front and the

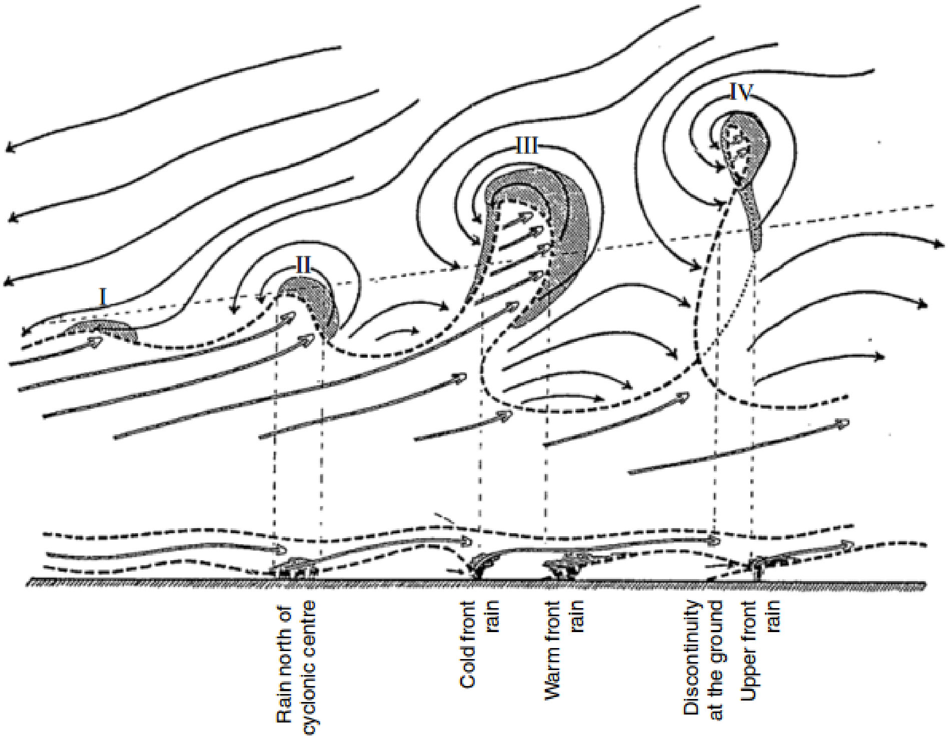


Figure 1.3: The Norwegian frontal cyclone model (Bjerknes and Solberg (1922)).

precipitation and the clouds associated with them (Fig. 1.4) (Cotton and Anthes (2011)). From the observations a practical rule was obtained, however is not always exact. The areas of small deflection of the wind from the gradient are generally areas of fast upward movement, the area of great deflection are generally areas of slow upward or even down ward movement (Bjerknes (1919)). The smallest deflections (fast upward movements) are found in the warm sector, while outside of it there are only slow upward movements or even descending in the rear of the cyclone. As all the most considerable part of the ascending motions is originating from the ground in the warm sector, cloudiness and precipitation are mainly involved there (Bjerknes (1919)). The ascension of the warm air will extend precipitation and cloud areas beyond the steering line, like the warm front representation in Figure 1.4. The rainfall over the cold area behind the squall line comes from the warm air lifted by the advancing cold wedge. The ascending motion will be rather strong, but its effect will not be perceived far behind the line, since there are no higher clouds occurring the squall surface is expected not to reach higher strata (Bjerknes (1919)).

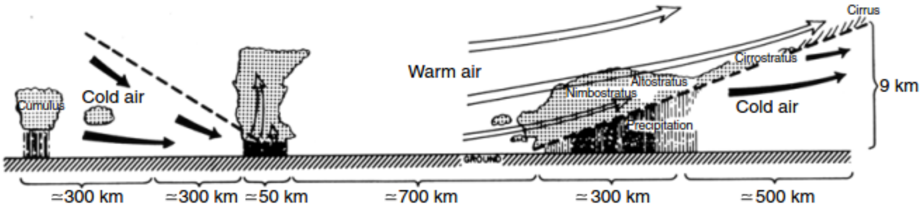


Figure 1.4: Vertical cross section through an idealized mid-latitude cyclone, according to the Norwegian cyclone model (Houze Jr. and Hobbs (1982)).

1.1.2. Shapiro-Keyser model

Observational studies after Bjerknes and Solberg (1922) indicated that the cyclone and the frontal structures were not always continuous from the surface to the upper atmosphere and that different dynamics were involved from upper- and lower-level fronts (Cotton and Anthes (2011)). Shapiro and Keyser (1990) proposed modifications to the Norwegian Cyclone model. This model takes the single-body approach and defines four basic phases of a cyclone in its life cycle (Semple (2003)). The ETCs starts developing from a a continuous and broad baroclinic front (Fig. 1.5I). As the cyclone develops in the vicinity of the centre there is a frontal fracture, and the cold front separates from the warm front and enters into the warm sector air (Fig. 1.5II). While the cold front moves eastward into the warm sector, the warm front develops in the west direction. This extension of the warm front starts to wrap around to the rear of the cyclone driven by the cold, northerly flow and forms the *bent-back* front. This front has the structure of a warm front and not that of an occluded front as in the Norwegian model. To indicate the cold front that is oriented perpendicular to the bent-back extension of the warm front the term *T-bone* is used (Fig. 1.5III). In the phase of maximum intensity the warm-core frontal seclusion is formed. The cold front moves further east of the cyclone centre and together with the bent-back warm front start encircling the low and trap relatively warm air at the centre of the cyclone. This seclusion is formed within the polar air and does not include air originating from the warm sector, unlike the Norwegian model suggests (Fig. 1.5IV) (Cotton and Anthes (2011), Semple (2003)).

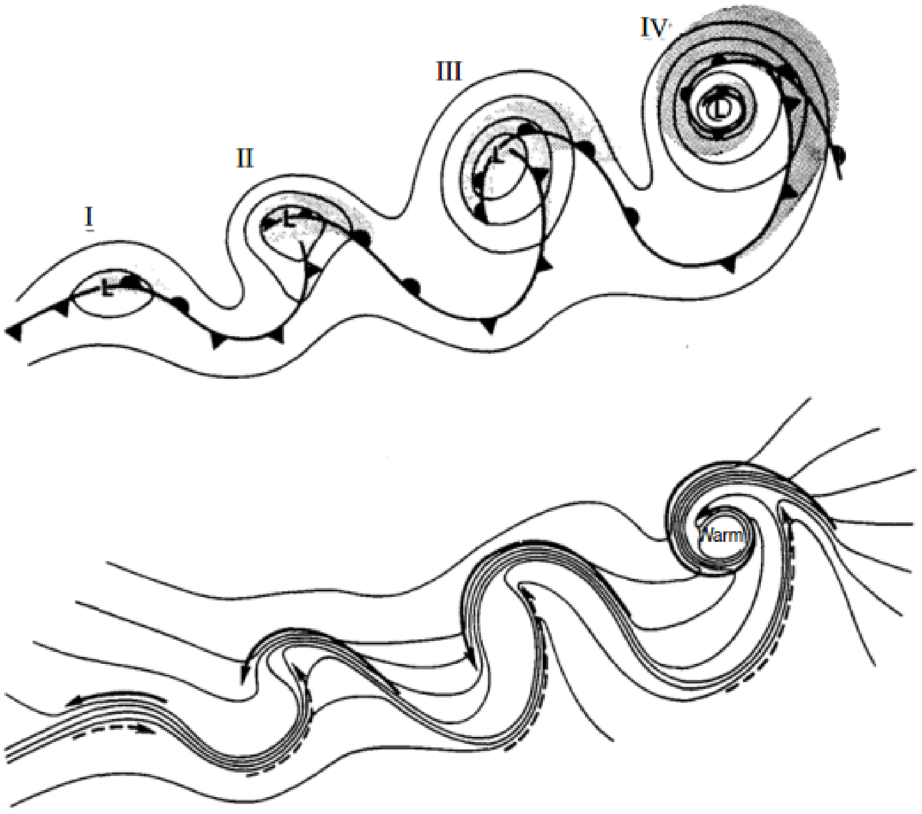


Figure 1.5: The life cycle of the marine extratropical cyclone. Upper: sea-level pressure, solid lines; front, bold lines; and cloud signature, shaded. Lower: temperature, solid lines; cold and warm air currents, solid and dashed arrows, respectively (Shapiro and Keyser (1990)).

Despite the model proposed by Shapiro and Keyser (1990) could be considered as an improvement of the Norwegian Cyclone model formulated by Bjerknes and Solberg (1922), different studies have highlight how depending on the characteristics of the storm and the location one models could be more appropriate than the other. Also, the best choice seems to be that of taking into account both model if the development of ETCs is examined (Cotton and Anthes (2011)).

1.1.3. The Conveyor Belt Model

Single-body approaches usually miss some of the crucial aspects in a cyclone's development and structure, while it is considered that multi-body approach give a more complete picture of these system (Semple (2003)). In the conveyor belt model the purpose is to relate the mesoscale clouds and precipitation features observed in ETCs to commonly observed airstreams or conveyor belts in these storms (Cotton and Anthes (2011)). The term conveyor belt defines the large-scale motion of air along an isentropic surfaces, which is a surface of constant potential temperature where air ascends/descends adiabatically (Semple (2003)). In an extra-tropical cyclone there are two kind of conveyor belt classified according to the relative temperature of the air flowing within them: the Warm Conveyor Belt (WCB) and the Cold Conveyor Belt (CCB) (Fig. 1.6). Browning and Roberts (1990) define the first one as a strong, rather well defined and often main rain-producing flow of air with high wet-bulb potential temperature that advances polewards ahead of the cold front within mid-latitude. The latter is defined as air ahead of and beneath the warm front which, relative to the advancing system, flows rapidly rearwards around the poleward side of the low centre (Browning and Roberts (1990)).

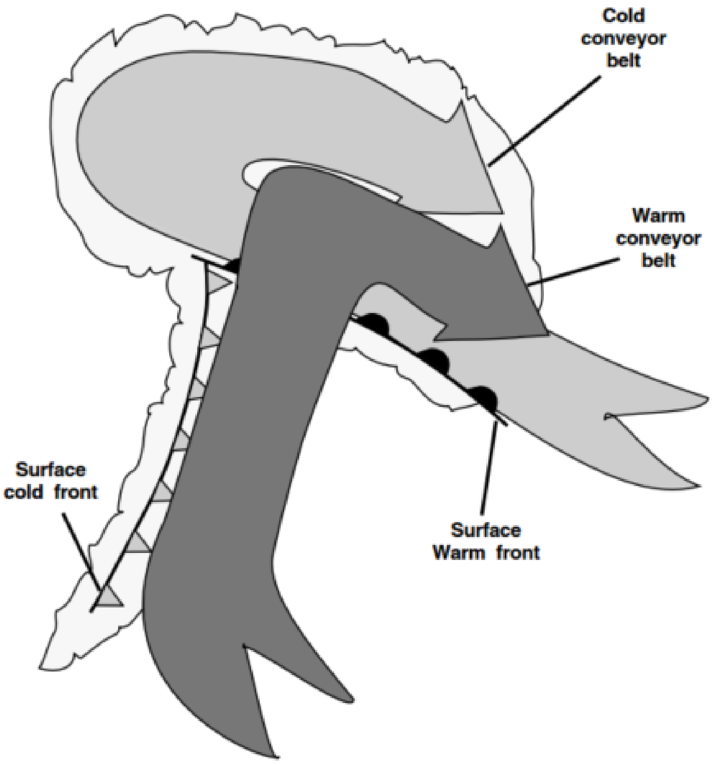


Figure 1.6: Two major airflows in a cyclonic system: the warm and cold conveyor belt (Semple (2003)).

The warm conveyor belt originates at low-levels in the southeast region of the storm and flows northward and westward, once it reaches the warm front it rises above it. If the the air in the WCB eventually reaches the upper troposphere, it turn antiyclonically to the northeast direction (Carlson (1980), Semple (2003)). If the air inside the WCB in convectively unstable the lifting will produce convective clouds, precipitation systems like rain-bands and severe thunderstorms. If the air inside the WCB is completely stable the lifting will lead to extensive layers of stratiform clouds, including nimbostratus (Cotton and Anthes (2011)). Most of the time the air is unstable in the low-levels in the warm sector and stable in the upper-levels of the cool sector north of the warm front. For this reason, in the warm sector is most probable to find convective clouds and precipitation systems while in the cold sector stratiform clouds and precipitation are more likely to happen (Cotton and Anthes (2011)).

The cold conveyor belt originates at low-levels in the northeast region ahead of a warm front of a developing cyclone and flows westward underneath the WCB travelling poleward above it. The CCB then rises following a lower isentropic surface than that of WCB and emerges north of the surface low. If it ascends enough it usually turns anticyclonically and flows parallel to the WCB (Carlson (1980), Semple (2003)). The temperature and the humidity of the CCB can significantly impact the amount and kind of precipitation reaching the surface. If the air in the CCB is relatively dry the precipitation reaching the surface will be reduced because of evaporation and/or sublimation of hydrometeors. Instead if the air in the CCB is relatively humid the amount of precipitation reaching the surface will be higher. Considering the temperature, if it is below freezing hail may be expected at the surface (Cotton and Anthes (2011)). The cloud produced by the ascension of the CCB are mainly confined to low and medium levels, below the cloud band associated with the WCB (Semple (2003)).

There is a third airstream being part of the conveyor belt model, the Dry Intrusion. It is a stream of air from the upper troposphere and lower stratosphere which, after earlier descent, approaches the centre of the cyclone as a well defined reascending dry flow with low wet-bulb potential temperature (Browning and Roberts (1990)). The northern part of the dry intrusion separates from the descending flow and turns northeastward to flow next to the left edge of the WCB. Despite this air ascends over the warm front it is usually cloud free due to its very low level of humidity. The southern part continues to descend warm and dry, flowing with a southwestward direction a it approaches the surface (Cotton and Anthes (2011)). In Figure 1.7 a representation of the three air stream acting in a cyclone system is represented.

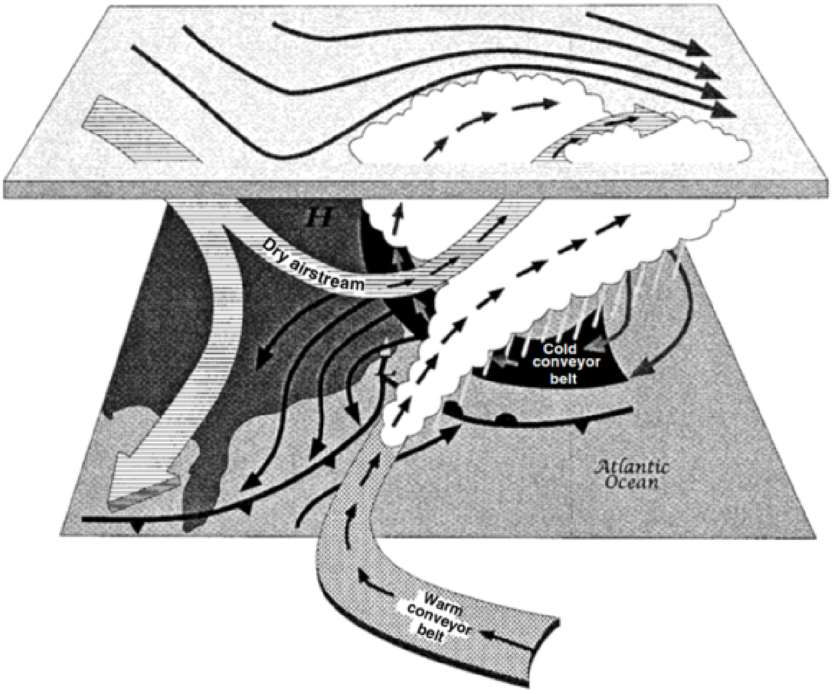


Figure 1.7: The conveyor belt model of airflow through northeast US cyclones (Cotton and Anthes (2011)).

1.1.4. A general structure for ETCs

Dacre et al. (2012) introduce the use of a tool to illustrate cyclone structure and characteristics. This tool has been created to explore the mean structure and evolution of a sample of 200 cyclones in the North Atlantic from 1989 to 2009. The authors summarise a general evolution of the low-level cyclone structure and of the vertical cyclone structure using the cyclone sample as reference.
Figure 1.8 shows the generic evolution of low-level fields throughout the cyclone life cycle. In the early stage of the life cycle, so when the cyclone is developing, the low-level temperature wave (θ_e) is amplifying between the cold and the warm fronts, the air is moist and warm and the region is called the warm sector (Figs. 1.8a and b). Meanwhile a closed isobar develops and the central pressure falls, as a consequence cyclonic circulation and system-relative wind speeds around the centre increase (Fig. 1.8b). At the time of maximum intensity, the central pressure is at its minimum and the steep pressure gradient causes the peak of wind speeds (Fig. 1.8c).

In the decaying stage the central pressure increases while the pressure gradients diminish, leading to lower wind speeds; the fronts weaken and the warm front wrap around the cyclone centre (Fig. 1.8d). Between 24 and 48 hours after maximum intensity, the central pressure continues to rise and the frontal gradients weaken further (Fig. 1.8e) (Dacre et al. (2012)).

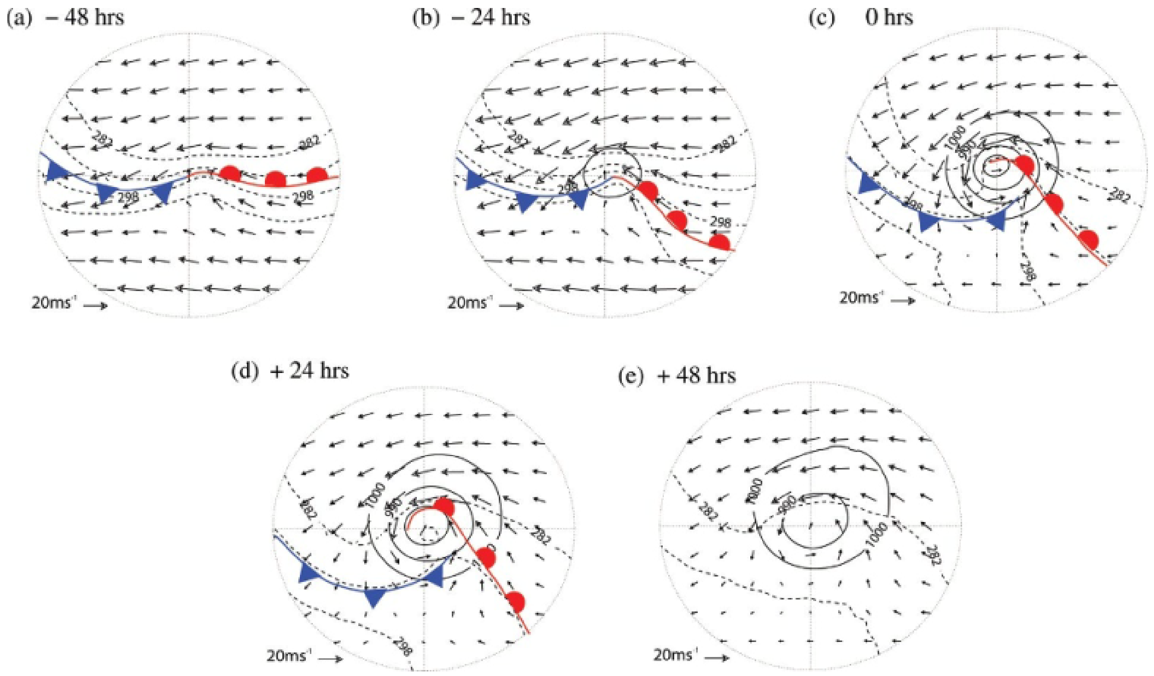


Figure 1.8: Horizontal composites from the extra-tropical cyclone atlas show the position of 925-hPa fronts, closed MSLP contours (solid lines), 925-hPa system-relative wind vectors, and 925-hPa equivalent potential temperature (θ_e, dashed lines at 282, 290 and 298 K) at (a) 48 and (b) 24 h before maximum intensity; at the time of maximum intensity; and (d) 24 and (e) 48 h after maximum intensity (Dacre et al. (2012)).

Figure 1.9 shows the generic evolution of the lower- and upper-level fields throughout the developing stage of the cyclone life cycle. In the early stage of the life cycle, a weak low-level temperature wave is seen to form downstream of an upper-level shortwave depression (Fig. 1.9a). As the cyclone develops at the surface (Figs. 1.9b,c), the thickness of the atmospheric layer between 1000 and 500 hPa is reduced by the advection to the west of cold air. Also the upper-level trough is deepened. In other direction, warm air advection to the east of the surface cyclone increases the layer 1000-500 hPa and intensifies the upper-level high pressure. Differential temperature advection to the west and the east of the surface cyclone amplifies the upper-level wave. Positive differential vorticity advection downstream of the upper-level trough forces ascent above the surface cyclone. This ascent causes a lower-tropospheric vortex which intensifies the surface cyclone. An increase in low-level wind speeds amplifies the low-level temperature wave, which amplifies the upper-level wave. Therefore, a positive feedback occurs between the processes at upper- and lower-levels. As the cyclone develops, the upper-level low-pressure system moves toward the surface low, until is located directly above it in the mature stage. This alignment produces a vertically stacked cold-core system which does not longer have a positive feedback between upper- and lower-levels. Then the cyclone decays (Dacre et al. (2012)).

1.2. ETCs identification and tracking

In the last three decades many numerical algorithm have been developed to objectively identify and track cyclones from digital data (Ulbrich et al. (2009)). However, since there is a great complexity behind extra-tropical cyclone's definition and characteristics, the application of different algorithms may give similar results considering certain aspects but also returns a lot of differences considering others. So, the selection of a particular method could really affect the conclusions on the results (Neu et al. (2013)).

The identification and tracking of ETCs might be considered a simple task, but is actually very challenging. The characteristics of tropical cyclones, which make them relatively easy to track, can be taken as a reference

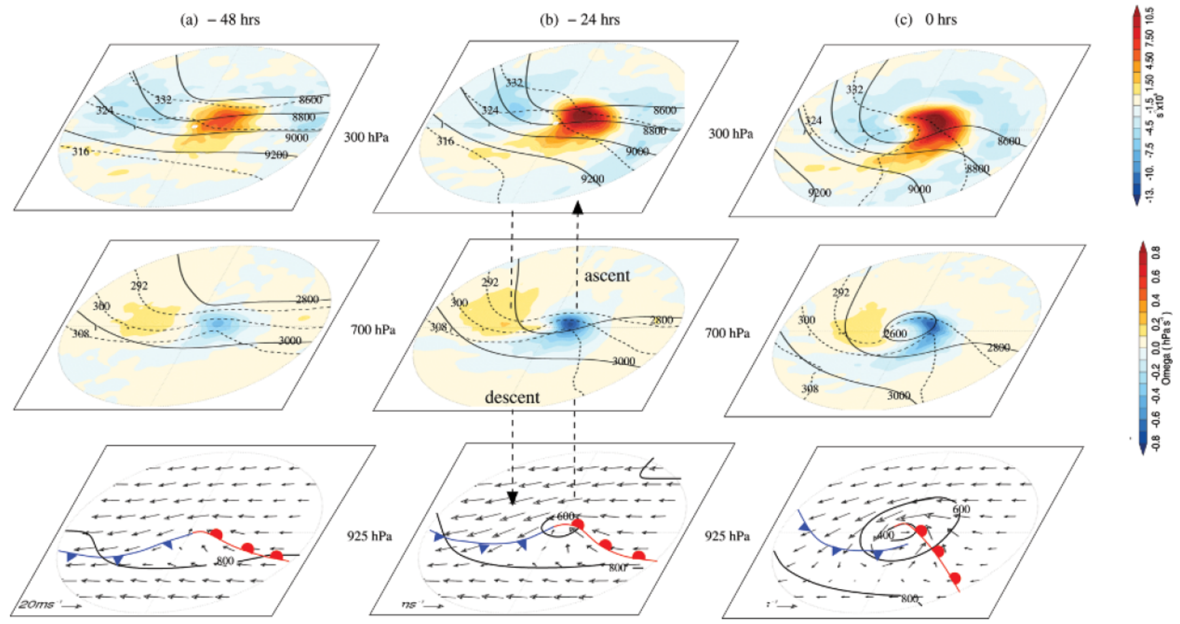


Figure 1.9: Horinzontal composites at (a) 48 and (b) 24 h before the time of maximum intensity, and (c) at the time of maximum intensity. Bottom row: 925-hPa geopotential height (solid lines at 400, 600, and 800 m). Middle row: 700-hPa geopotential height (solid lines at 2800 and 3000 m); equivalent potential temperature (θ_e, dashed lines at 292, 300, and 308 K) and vertical velocity (Omega, filled). Top row: 300-hPa geopotential height (solid lines at 8600, 8800, 9000, and 9200 m); $theta_e$ (dashed lines at 316, 324, and 332 K) and divergence (filled) (Dacre et al. (2012)).

to make a comparison with mid-latitude storms. Tropical storms occur rarely, move slow, are symmetric and have a clear structure. Extra-tropical storms are much more common, are often asymmetric, can vary a lot in size (diameters ranging from 100 to over 1000 km) and also in the value of the translational velocities. Also ETCs may form at different synoptic situations, from lower-tropospheric levels to extended depth. Another complications is due to the possibility that two cyclones can merge into a singular one and that one cyclone can split into two different ones (Neu et al. (2013)).

The quality of the datasets plays an important role in the tracking of cyclones. In fact, different resolutions bring diverse problems. Tracking in high spatial resolutions results difficult since the number of cyclones located is high, while tracking in low temporal resolutions makes the search areas that have to be scanned larger (Blender et al. (1997)). However if the full life cycle of cyclones wants to be capture that even the cyclonic windstorms are traced high resolution is essential (Pinto et al. (2005)).

Tracking methods are performed taking into account different atmospheric variables. The biggest difference in the algorithm is linked to choosing the mean sea level pressure (MSLP) or the 1000 hPa geopotential-height (z1000) and lower-tropospheric vorticity a basic metric identification for tracking. Selecting one in respect of the other denotes the different purpose on which one might focus, the MSLP is linked to the mass field of the cyclone and gives a better representation of the low-frequency scale, while the vorticity is linked to the wind field and has more information regarding the high frequency scale (Hodges et al. (2003)). Choosing an atmospheric variable rather than the other can lead to different estimated positions of the cyclone centre, since the centre based on vorticity in a westerly flow can be found a few hundred kilometers south of the related minimum in the pressure field (Sinclair (1994)). The choice of the atmospheric variable depend also from the resolution adopted in the analysis. For example, Blender et al. (1997) do not consider maxima in the vorticity because the structure of the field is too detailed and noisy in high-resolution data. However, the vorticity field becomes useful when considering lower resolution (Hodges (1994), König et al. (1993),). Methods considering MSLP and z1000 minima have good performance when the cyclone has reached the stage of maturity but in general fail to locate the cyclones at early stage of their life cycle, that is because flat eaves and pressure minima might not be always linked togheter (Dacre and Gray (2009)).

In different studies (Bartholy et al. (2006), Blender et al. (1997), Blender and Schubert (2000), Geng and Sugi (2001), Jung et al. (2006), König et al. (1993), Sickmöller et al. (2000)) the tracking procedure begins with the identification of cyclones' positions trough the analysis of local minima in the MSLP fiels or in the 1000 hPa geopotential-height reference field. The pressure values in the point is compared with those of the surrounding 8 neighbours to determine whether or not the point considered is a local minimum. Another step of the procedure is an additional condition on this minimum as a measure of intensity. It is done by examining the averaged pressure difference between the central grid point and the surrounding ones. In other words the local minimum in pressure will be a cyclone's position only if the pressure gradient is larger than a certain threshold in its neighbourhood. The threshold value and the points considered as surroundings change depending on the study considered. Geng and Sugi (2001) are expecting the averaged pressure difference between the central grid point and the eight neighbours to be larger than 0.3 hPa. König et al. (1993) expect to have a closed chain of grid points surrounding the central one with a geopotential height larger than 20 gpm among the 24 neighbouring and surrounding points (Fig. 1.10). Blender et al. (1997) state that the requirement of a closed isobar in a synoptic region would exclude the weak phenomena, like the on used in König et al. (1993). However they do not state the values of the gradient or of the difference in pressure between the central points and its surroundings. Sickmöller et al. (2000) identify a value of 5 m (1000 km)$^{-1}$ as threshold value in the mean z1000 gradient in a synoptic neighbourhood, of which the extension is not specified since the value of the gradient is given. Bartholy et al. (2006) criterion expect the pressure gradient to be greater than 0.07 hPa (100 km)$^{-1}$ for all directions. Blender and Schubert (2000) take as threshold a larger value, in fact the minimum value for the mean gradient of z1000 is 100 m (1000 km)$^{-1}$ (12 hPa (1000 km)$^{-1}$) in a synoptic neighbourhood. In the study it is specified that this value restricts the analysis to intense ETCs. In each studies previously commented is also highlighted how a threshold on the orography is also needed, the topographic height at the grid point identified as a local minimum must be lower than 1000 m (1500 m in Geng and Sugi (2001)). If also higher altitudes are considered, the identification of local minimum in the pressure field could be misleading.

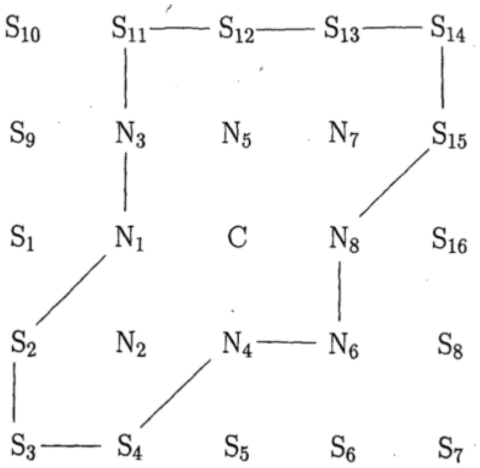


Figure 1.10: Example of identification of a cyclone at grid point C. Position of neighbour (N_i) and surrounding (S_i) points, which form a closed chain, where the difference of geopotential height exceeds the margin (König et al. (1993)).

The identification of the cyclones' positions is performed at each time step of the dataset available. If it is the first time step, all the cyclone centres found are considered to be a newly generated cyclone. Otherwise, a procedure to track whether the cyclone centre is just one of those already present at the previous time step that has moved or if it is a newly generated cyclone. The tracking in most of the studies previously cited (Bartholy et al. (2006), Blender et al. (1997), Blender and Schubert (2000), Geng and Sugi (2001), Jung et al. (2006), Sickmöller et al. (2000)) is done by applying a nearest neighbour search in the z1000 field considering the positions at the previous time step. A threshold is also taken into account to limit the area of search where the cyclone can travel. Blender et al. (1997) do not assume a preferred direction and speed of propagation for the cyclones and assumes a maximum displacement velocities of 80 km h^{-1} to restrict the range of motion. Sickmöller et al. (2000) limit the search within a distance of 600 km independently of the latitude. It also considers other aspects to validate the track of a cyclone, it requires a minimum lifetime of 3 days and that a minimum pressure gradient of 20 m (1000 km)$^{-1}$ is exceeded at least once during the life cycle. Jung et al. (2006) do not specify which value are assumed for the threshold but it states how the tracks identification proceeds following a three steps analysis on cyclone propagation analysis, sorting of the crossing trajectories and separate analysis of the stationary cyclones. In Bartholy et al. (2006) two subsequent potential cyclone centres belong to the same cyclone track if their geographical distance is less than 900 km and if the difference in sea-level pressure between the two is less than 5 hPa in absolute value. Blender and Schubert (2000) limit the extend range up to a width which corresponds to a maximum propagation speed of 100 km h^{-1}. Geng and Sugi (2001) explain better how the nearest neighbour search algorithm works. It starts by enumerating the cyclone centres to identify them, if it is the first time step of the data sets all the cyclone identified are considered newly generated and a new number is assigned to each one of them. If it is not the first time step the cyclone tracking procedure is applied. The procedure is performed by checking the cyclone centres detected in the previous time step following three criteria:

1. The cyclone centre n in the present time step t is the one closest to the cyclone centre n' at time step t - 1
2. The distance between the two cyclone centres is smaller than a threshold value of 600 km
3. The direction of movement of the cyclone is satisfying a tolerance limit which is a function of the 700-hPa steering flow and the geographical location

If all the criteria are met it is straightforward that the cyclone centre n at time step t come from the cyclone centre n' at time step t - 1. Then to the cyclone centre n will have the same number of n', $c = c'$. If either criterion 2) or 3) is not satisfied, then the cyclone centre n is considered to be newly generated and it receives a new number c.

König et al. (1993) use a different procedure to track the cyclone centres which is performed by specifying search regions around the local minima and checking in the next time step if these regions have a contact or a overlap. Each search region is formed by the grid point where the cyclone centre has been identified and the left and right neighbours. Each cyclone centres at time t gets an ordinal number. If a the same time step two region are overlapping or adjacent both regions are combined in one with same ordinal number and as cyclone centre the grid point with lowest geopotential height. At time $t + \Delta t$ a search region can be associated to one at time t if the spatial difference between the two regions is equal or less than the grid size, either in meridional or longitudinal direction. If this is the case it receives the same ordinal number of the region at time t, otherwise a new number is assigned to the region. Once all the time series has been analyzed the trajectories of the cyclone can be identified by connecting all the centres at the different time. At last only the trajectories lasting more than 24 hours are kept.

1.3. Cyclone Activity in the North Atlantic

Cyclone represents a primary mechanism of the transport of heat and moisture towards the poles. A systematic change in the intensity/frequency of cyclone activity or in the geographical location will cause anomalies on the regional climate. Due to the link between them a shift in the preferred regions of cyclone activity will determine deviation in the planetary scale-flow (Wang et al. (2006)).

Dacre and Gray (2009) report a thorough analysis of the spatial distribution and the characteristics of the cyclone track in the North Atlantic. In the study cyclone track density is defined as the number of cyclones passing through a certain area in a certain time, and genesis density is defined as the number of cyclones that generates in a given area in a given time. The authors identify the North Atlantic storm track as the region of cyclone track density plotted in Figure 1.11a. There is an area of maximum density extending from the east coast of North America into the North Atlantic. In the region of interest 3 regions of maximum genesis are found, one east of the Rocky Mountains, one off the east coast of North America and one southeast of Greenland. While in the east Atlantic and in the northern Mediterranean weaker genesis density are found (Fig. 1.11b).

Moisture and friction are two crucial factor for the developing of a cyclone, in fact most of the cyclones that grow are generated over the sea, where the source of moisture is bigger an the friction over the surface is lower. Cyclones which originate off the coast of North America, in the West Atlantic, follow a path with a northeastward direction and decay at the high latitude of the Atlantic Ocean, also west of Greenland. A big part of the cyclones reaching western Europe is generated in the east Atlantic region, this area also corresponds to a region of decay for west Atlantic cyclones. Because of this circumstance it is hypothesized that east Atlantic cyclones forms on the trailing fronts of decaying west Atlantic cyclones. This supposition is supported by the fact that east Atlantic cyclones have anomalously low MSLP, due to their proximity to a pre-existing cyclone,

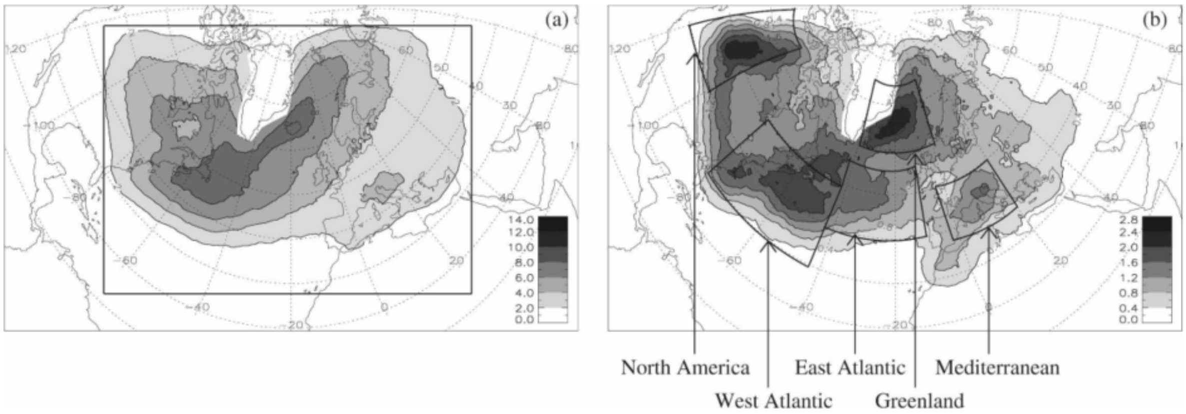


Figure 1.11: (a) Track density over the North Atlantic, the contours line are drawn every 2 cyclones (10^6 km^2)$^{-1}$ month^{-1} and (b) genesis density, the contours are every 0.4 cyclones (10^6 km^2)$^{-1}$ month^{-1} (Dacre and Gray (2009)).

and high relative vorticity in the genesis point, which is usually associated with the fronts. Figure 1.12 shows the principal cyclone track in the North Atlantic overlaying the track density (Dacre and Gray (2009)).

In Wang et al. (2006) the authors comment the changes in ETCs' climatology based on two different datasets, the 40-yr ECMWF Re-Analysis (ERA-40) and the NCEP-NCAR reanalysis (NNR) for 1958-2001. For the regions of northern Europe and eastern North America the trends found in the two different datasets are in agreement with each other. Considering the North Atlantic the biggest changes are seen in winter (JFM), where cyclone activity has increase over high-latitudes and decreased over the mid-latitudes. Together with the changes in the cyclone activity also the storm track has moved, extending southeastward toward the North Sea while reducing a little bit in northward direction over the northeastern Atlantic. The increase at high latitudes during winter is mainly associated with strong cyclone while the decrease at mid-latitudes is associated with every kind of cyclone intensity. During other periods, AMJ and JAS, the changes in cyclone activity are generally small and insignificant both at high and middle latitudes. However in summer (JAS) the cyclone tracks have a longer life cycle in the recent decades rather than in the early ones.

Geng and Sugi (2001) show that during the past 40 years in the northern North Atlantic there has been a significant intensification of the cyclone activity. In fact around Greenland and the Icelandic regions a gradual increase of cyclone density has been found, while over the eastern coast of the Atlantic and western Europe a gradual decrease in cyclone density has happened. In accordance with these results other variables concerning cyclone intensity increases over the northern part of the North Atlantic storm track. These are the cyclone deepening rate which became stronger and the cyclone moving speed which became faster. Together with this intensification the cyclone activity along the 40 years previous to 2000 reveal a northeastward trend of extension for the North Atlantic storm track, so toward the end of the storm track. So generally from the result it can be stated that more cyclone with stronger central pressure gradient, faster moving speed and stronger deepening rate are observed throughout the 40 years. North Atlantic storm track is also influenced by El Niño southern oscillation (ENSO), in the east coast of North America a significant increase in cyclone activity has been found during El Niño, while a decrease has been found during La Niña (Eichler and Higgins (2006), Hirsch et al. (2001)).

1.4. Deep Learning

Deep Learning are Artificial Intelligence (AI) techniques that try to reproduce the works of human brain in processing data and in the creation of patterns between inputs and outputs. Deep learning models are able to learn representations of data with multiple levels of abstraction (LeCun et al. (2015)). Deep Learning is a more specific part of machine learning methods which uses multiple layers in the construction of a Neural Network (NN), that is why the adjective deep is utilized. In particular, adopting multiple layers in a network allows to progressively extract higher level features from raw data.

Deep learning has obtained large success in the recent years because of its application in different fields of knowledge and business such as computer vision, speech recognition, natural language processing, audio recognition and many others (Hinton et al. (2012), Krizhevsky et al. (2012), Lawrence et al. (1997), Sutskever

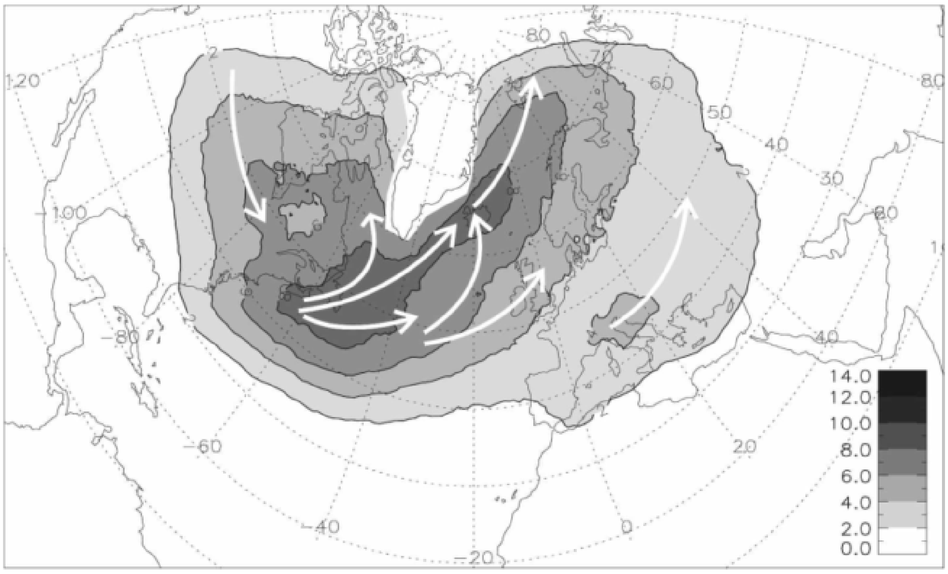


Figure 1.12: Schematic of cyclone track paths plotted over track density. Contours line are drawn every 2 cyclones (10^6 km^2)$^{-1}$ month^{-1} (Dacre and Gray (2009)).

et al. (2011), Sutskever et al. (2014), Taigman et al. (2014),Tompson et al. (2015)).

Deep learning big success is due especially to its ability to exploit the property of compositional hierarchies in different natural signals and solving the problem of learning a representation by expressing it in terms of other simpler ones. This method allows to realize higher-level features by composition of lower-level ones (LeCun et al. (2015)). For example, a deep learning algorithm is able to represent an image of a person or an object by combining simpler concepts that defines it, such as corners or contours (Goodfellow et al. (2016)), see Figure 1.13. Each layer of the deep neural network learns one of these simpler representations, and by putting them all together they are able to reproduce the complex representation which was the aim of the algorithm. These execution of sequential instruction offers great power to the network because later layers can refer back to the results of earlier instructions.
As general summary definition it can be stated that deep learning is a set of particular algorithm of machine learning which is able to learn representation of the world as a nested hierarchy of concepts (Goodfellow et al. (2016)). The general architecture is a multi-layer composition of simple modules, all of which contributes to calculate an input-output mappings and are subjects to learning (LeCun et al. (2015)).

Deep learning algorithm will be first introduce with a summary description of the basic unit, the general structure and supervised learning. In the following sections, Convolutional Neural Networks and Generative Adversarial Networks will be better described. In the final part some relevant examples of Neural Network application in the meteorology field will be discussed.

1.4.1. Artificial Neuron

The basic unit composing a Neural Network is called artificial neuron. The functioning of an artificial neuron is inspired by biological (or natural) neurons. So the artificial neuron receive information through inputs, like the natural one receive signals through synapses. Each input is multiplied by a weight, which determines the strength of the signal. These terms are then combined together to determine the activation of the neuron. The output of the neurons is computed by a specific function, which sometimes is in dependence of a certain threshold (Gershenson (2003)).
Mathematically a neuron is an operator which computes a linear function of the form $z = W^T x + b$ followed by an activation function $g()$ which is usually nonlinear like a sigmoid (e.g. logistic function), a hyperbolic tangent (tanh) or a rectified linear unit (ReLU). The latter is a function defined as the positive part of its argument $f(x) = max(0, x)$, it has been increasingly used in the last years because it yields superior performances. In Figure 1.14 is shown how this is computed, a weighted (w_i) average is performed on the inputs of the neu-

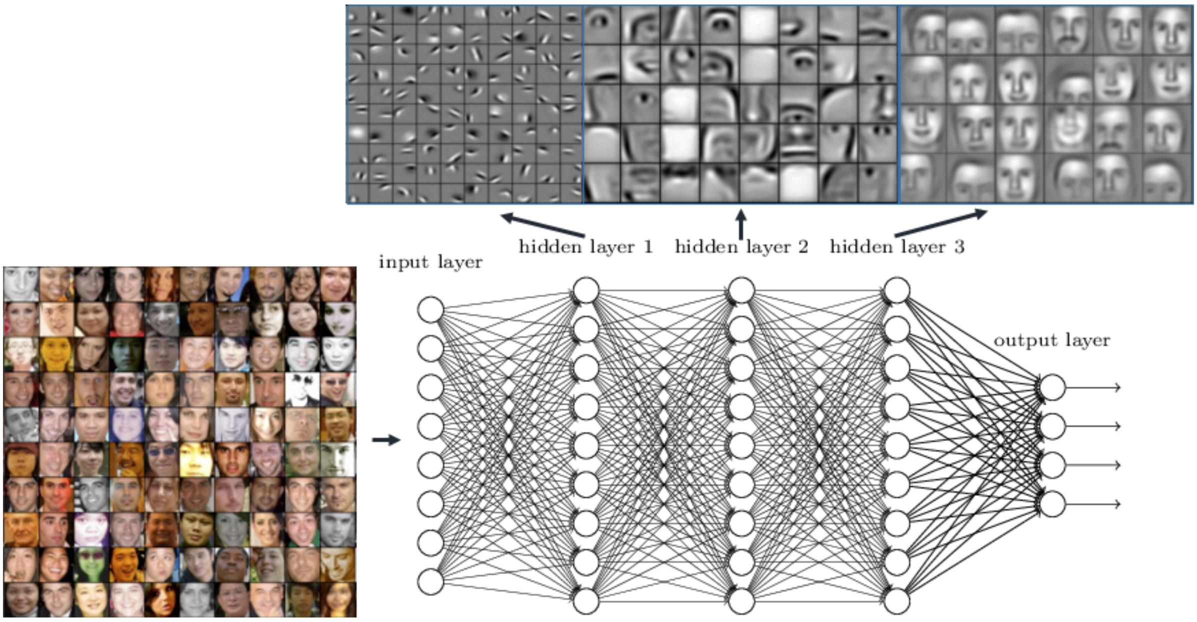


Figure 1.13: Deep neural network learns hierarchical feature representation for human faces. In the first hidden layer edges and small features of the faces are learned. In the second hidden layer the composition of different part of the face is learned, like nose and mouth or eyebrow together. In the third hidden layer the different part composing a human face are put together.

ron (x_i), after it a bias is added (b). The result of this operation becomes the input of the activation function which returns the output of the neuron. The activation function is used to introduce non-linearity in the modeling capabilities of the network. Deep networks are built of multiple layers of neurons.

In Figure 1.15 a general structure of Deep NN is represented. There is a neuron in the input layer for each input that is given to the network. In the graph only two hidden layers are represented but there could be many more, each neuron in the hidden layers is connected with all the neurons in the previous layer and in the following layer. All the values that are passed from a layer l to the layer $l+1$ are grouped into a vector called activation of the l^{th}'s layer. The output layer contains the neurons which compute the output of the network, usually the activation function in these neurons is different from the one adopted in the other layers and depends on the task of the network. In the case of a classification of images, like determining if in the image a cat is present, the input will be the values on the three different channels of each pixel in the RGB image. The output will be 0 or 1 depending on the eventual presence of a cat in the picture. The operation of passing through the layers of the network, where a set of units computes the output by applying an activation function to the weighted sum of the inputs from the previous layer, is called forward propagation (LeCun et al. (2015)).

1.4.2. Supervised learning and Gradient Descent

Supervised learning consists of observing a dataset of input features $\mathbf{x}$ which are associated to a label or target value $\mathbf{y}$. The algorithm learns to predict $\mathbf{y}$ knowing $\mathbf{x}$, usually by estimating the conditional probability $p(\mathbf{y}|\mathbf{x})$. The term supervised comes from the fact that the learning is performed by providing the the deep learning system a target value $\mathbf{y}$ which is already determined by an instructor (Goodfellow et al. (2016)). Many deep learning applications have a feed-forward Neural Network architecture which is trained to represent a mapping from a fixed-size input to a fixed-size output (LeCun et al. (2015)).
To train the network a learning algorithm is applied, which first computes an objective/cost function that measures the error between the output values computed through the forward propagation and the desired values determined by the target features. Then it adjusts the internal parameters of the network to reduce this error (LeCun et al. (2015)). The weights/parameters are properly adjusted by computing the vector of gradients and applying the algorithm of gradient descent (Ruder (2016)). The gradient is the partial derivative of the cost function with respect to each parameter, which is computed using the chain rule of derivation. This "propagation" of the error from the output layer towards the input layer is known as "back-propagation". Af-

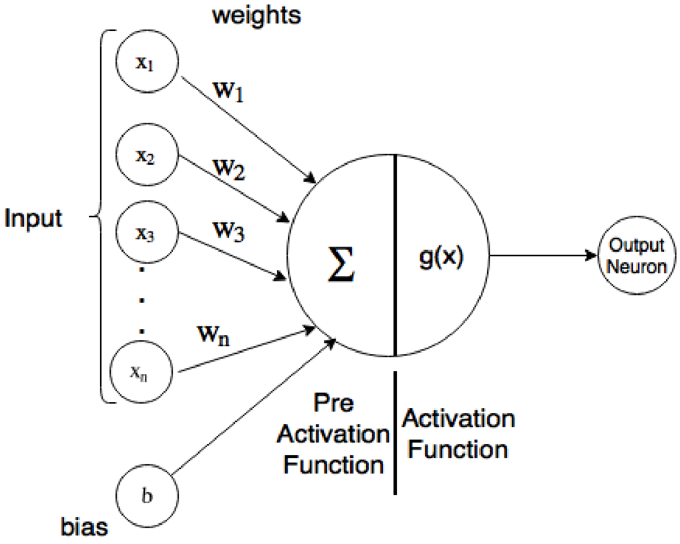


Figure 1.14: Structure of a Neuron, the basic unit of a Neural Network.

ter propagating the gradients, the weights are adjusted via gradient descent, which consists in minimizing the objective function by updating the parameters in the opposite direction of the derivative of the cost function with respect to the parameters (LeCun et al. (2015), Ruder (2016)). This procedure is recursively repeated until the objective function stops decreasing. Because computing the firs-order partial derivatives with respect to the parameters is as much complex as evaluating the function, if the function is differentiable with respect to its parameters, gradient descent is a relatively efficient optimization algorithm(Kingma and Ba (2017)).
In this operation the learning rate α plays a crucial role, because it determines the size of the steps taken to update the parameter and also controls the speed at which the model adapts to the problem. A too small value of α can cause the training to get stuck in the objective function space, while a large value can bring the model to converge very rapidly and learn a sub-optimal solution (Brownlee (2019c)).
To summarize a learning algorithm works through different steps:

- **Forward propagation**: Given the input x of a training example the expected value $\hat{y}$ is computed going trough the network operations.
- **Error computation**: the error for the training example is computed by the loss function and the objective function is updated.
- **Backward propagation**: The derivatives of the cost function with respect to the different variables present in the network (activations, weights and biases) are computed.
- **Parameters update**: Weights and biases values are updated through gradient descent according to the following formulas:

$$W^{(t+1)} = W^{(t)} - \alpha \nabla_W L^{(t)} \tag{1.1}$$

$$b^{(t+1)} = b^{(t)} - \alpha \nabla_b L^{(t)} \tag{1.2}$$

Depending on the amount of data that wants to be used to compute the gradient of the objective function there are three different variants of gradient descent (GD):

- **Batch Gradient Descent**: performs a parameters update considering the entire training set.
- **Stochastic Gradient Descent (SGD)**: computes the gradient of the cost function with respect to the parameters for each input/output training couple.
- **Mini-batch Gradient Descent**: is half way between the two previous method since it performs an update for every mini-batch of n training examples.

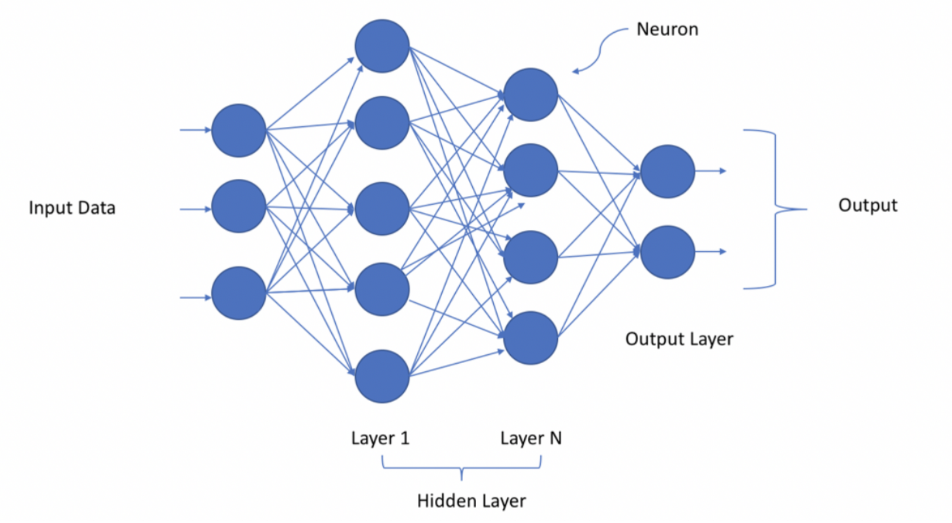


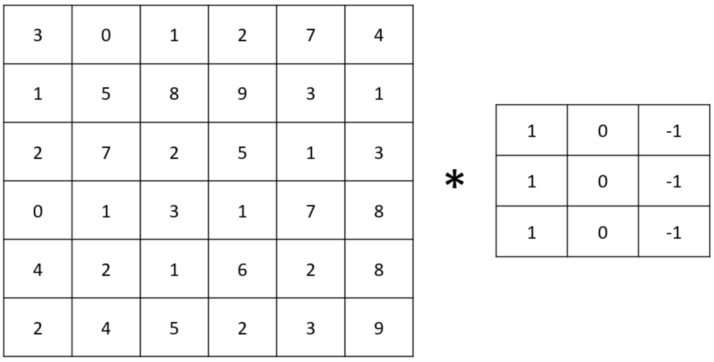
Figure 1.15: General structure of a Deep Neural Network.

Batch gradient descent needs to compute the gradients for the whole data-set before performing an update. It also does redundant computations since it recomputes gradients for similar examples before each parameter update. Stochastic gradient descent is usually much faster because it avoids the redundancy of batch gradient descent by doing one update at a time. The frequent updates of SGD cause the objective function to fluctuate heavily. SGD's fluctuations enable the algorithm to jump to new and potentially better local minima, while considering batch GD the algorithm converges, therefore it can guarantee a more stable convergence of the objective function (Ruder (2016)).
After the optimization algorithm as minimized the error on the training samples, the network performance must be evaluated on a different set of examples called test set. The system must be tested to measure the ability of the machine to generalize and to produce sensible answers on new inputs that it has never seen before (LeCun et al. (2015)).

1.4.3. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a special kind of neural network for processing data that has a known grid-like topology, such as images (Goodfellow et al. (2016)). They are designed to process the data that come in the form of multiple channels. The typical architecture of a CNN is structured as a series of stages. The first stages are usually composed by two types of layers: convolutional and pooling. The last stages are usually fully connected layers like the one described in section 1.4.1.
In a convolutional layer the units are organized in feature maps, each unit is connected to local regions in the feature maps of the previous layer thanks to a set of weights called filter. All the units in a feature maps share the same filter. The result of the weighted sum and the filtering operation is then passed through an activation function. Mathematically, the filtering operation which is performed in a convolutional layer is a discrete convolution. Convolution is a linear operation which implies a particular kind of matrix multiplication between an input matrix and a filter. The output matrix resulting from this operation will have smaller dimension than the input ones (LeCun et al. (2015)). To understand better how convolution works is better to go through an example. As example a two dimensional convolution of 6 by 6 matrix with a 3 by 3 filter is considered. The operation is represented in Fig. 1.16, the '*' symbol denotes the convolution operation.

The resulting matrix of this operation will be a 4 by 4 matrix for reason that will be clear later. The convolution works as follow: to compute each element of the the output matrix the filter is applied on the matrix using a window of the same dimension of the filter, starting from the top left corner and proceeding one step forward at each new element of the output matrix. So the first element's window will be the 3 by 3 sub-matrix in the top left corner (the orange box in Fig. 1.17). To compute the element of the output matrix a element wise multiplication between the filter and the matrix window is performed, after that all the elements are summed. So in case of the first element of the output matrix we will have $(3*1)+(1*1)+(2*1)+(0*0)+(5*0)+(7*0)+(1*(-1))+(8*(-1))+(2*(-1)) = -5$.



$$\begin{bmatrix} 3 & 0 & 1 & 2 & 7 & 4 \\ 1 & 5 & 8 & 9 & 3 & 1 \\ 2 & 7 & 2 & 5 & 1 & 3 \\ 0 & 1 & 3 & 1 & 7 & 8 \\ 4 & 2 & 1 & 6 & 2 & 8 \\ 2 & 4 & 5 & 2 & 3 & 9 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Figure 1.16: Convolution operation on a 6 by 6 matrix using a 3 by 3 filter.

Figure 1.17 shows how the convolution operation proceed for the first 5 output elements. Figure 1.18 shows the final result of the convolution operation in example.

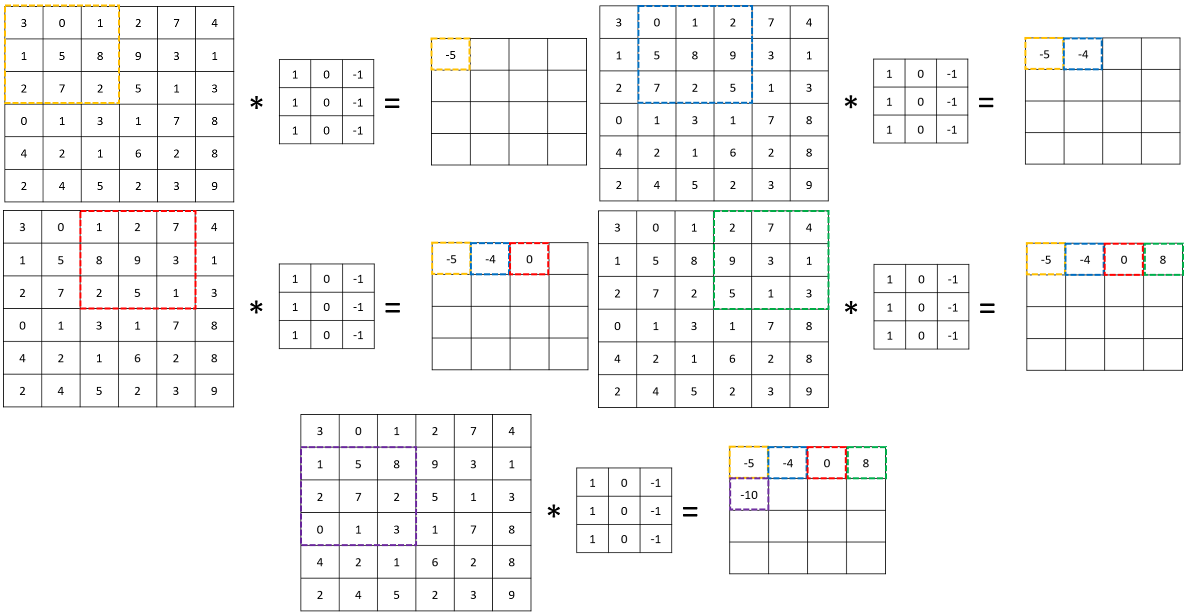


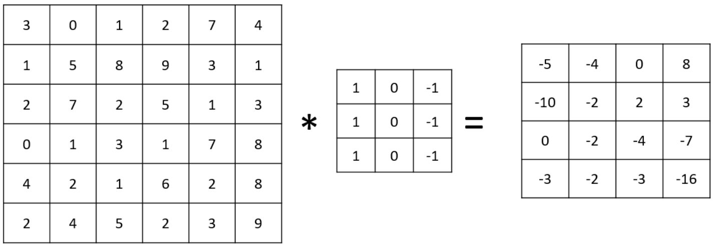
Figure 1.17: Windows selection for the first 5 elements of the output matrix of the convolution operation.

Convolutional layer role is to detect local conjunctions of features from the previous layer (LeCun et al. (2015)).

Pooling layers reduce the spatial resolution of the input by computing a summary statistic over a local spatial region. So, instead of having a filter performing a convolution operation there is a filter computing a statistic (average, min, max, etc.) across a window determined by the filter size. Pooling layers main motivation of usage is that they promote invariance to local input transformations since their outputs are invariant to spatial location within the pooling region (Tompson et al. (2015)). The role of the pooling layers is to semantically merge similar features into one, reduce the size of the representation and speed the computation (LeCun et al. (2015)).

In a ConvNet the values in the filter cells are the parameters that the network will learn, in fact back-propagation gradients through a CNN is as simple as through a regular network, making all the weights in all the filters trainable (LeCun et al. (2015)). These weights will determine, for example, the kind of features that a layer will learn from images like edges, corners or particular shapes. Obviously these features will be simpler, such as vertical edges, in the early layers and more complex aspects, such as objects, in the later layers.

Convolution layers are always more convenient to use in respect of traditional layers when working of grid-like topology data. In fact, compared to a standard feed-forward network with similarly-sized layers, Con-



$$\begin{bmatrix} 3 & 0 & 1 & 2 & 7 & 4 \\ 1 & 5 & 8 & 9 & 3 & 1 \\ 2 & 7 & 2 & 5 & 1 & 3 \\ 0 & 1 & 3 & 1 & 7 & 8 \\ 4 & 2 & 1 & 6 & 2 & 8 \\ 2 & 4 & 5 & 2 & 3 & 9 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} -5 & -4 & 0 & 8 \\ -10 & -2 & 2 & 3 \\ 0 & -2 & -4 & -7 \\ -3 & -2 & -3 & -16 \end{bmatrix}$$

Figure 1.18: Final result of the convolution operation on a 6 by 6 matrix using a 3 by 3 filter.

vNets have much fewer connections and parameters to learn, so they are easier to train (Krizhevsky et al. (2012)). The are 3 main reasons behind ConvNets taking advantage of the properties of natural signals forming a regular grid(LeCun et al. (2015)):

- **Parameter sharing**: a feature detector in a layer (such as a vertical edge detector) that is useful in one part of the image is most probably useful also in other parts of the image;
- **Sparsity of connections**: in each layer, each output value depends only on a small number of inputs;
- **Hierarchical features**: the network progressively builds more complex features from simple ones exploiting the compositional nature of input data.

1.4.4. Generative Adversarial Networks

Generative Adversarial Network (GAN) is a deep learning method to make a generative model learn a complex data distributions from samples (Lin et al. (2018)). GANs are based on a game-theoretic scenario in which the generator network must compete against ad adversary, the discriminator. So, the training of the generative model happens through an adversarial competition between two sub-models: the generator and the discriminator. The first one is trained to generate new examples which are plausible and similar to the real ones, having a vector of random values as input. The second one is trained to classify the examples as real or generated (Gulrajani et al. (2017), Li et al. (2017)), see Figure 1.19. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine articles (Goodfellow et al. (2014)). The generator is trained by the discriminator who acts like an adaptable loss function and teaches the features it learns from the training data to the generator through back-propagation (Gagne II et al. (2020)). At the end of the training what interests is the generative model, which is assumed to be able to generate new examples from the same data distribution of the training samples.

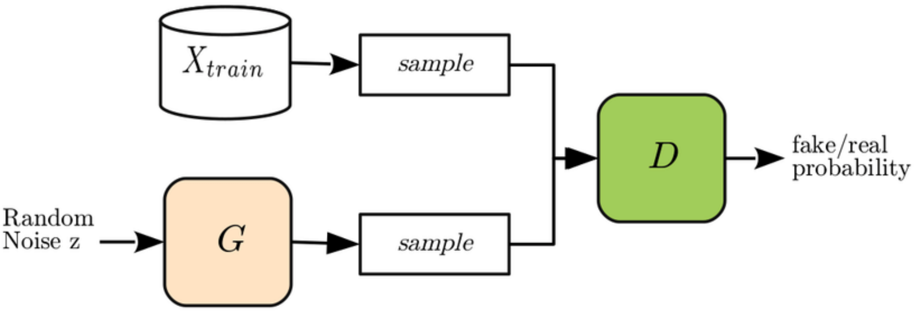


Figure 1.19: GAN training framework. The generator (G) creates a sample of new examples having random noise as input. The discriminator (D) classifies the examples it receives as input between reals and fakes.

The Generative Adversarial Network is a very powerful method for fitting implicit generative models, able to learn the probability distribution of the training dataset (Kodali et al. (2017)). However the training of this framework is very challenging because it consists of searching the Nash equilibrium of a non-convex game in a high dimensional parameters space (Chen et al. (2019), Salimans et al. (2016)). In fact, this process is very unstable and chaotic being highly dependent from the parameters initialization and it also often results in mode collapse, divergence, cyclic behaviour and vanishing gradients (Chen et al. (2019), Kodali et al. (2017), Li et al. (2017), Radford et al. (2015)).

The generator network produces samples $x = g(z;\theta^{(g)})$. z is a set of random values vectors extracted from a Gaussian distribution, the domain containing these vectors is called latent space. The vectors of the latent space are the input random noise of the model to generate new examples. The discriminator emits a probability value given by $d(x;\theta^{(d)})$. The learning problem for a GAN is usually formulated as a zero-sum game, in which a function $v(\theta^{(g)},\theta^{(d)})$ determines the payoff of the discriminator. The generator receives $-v(\theta^{(g)},\theta^{(d)})$ as its own payoff. During learning, each component or player attempts to maximize its own payoff (Goodfellow et al. (2016)).

The advantages of GANs are primarily computational in respect of other generative methods but they can also represent very sharp, even degenerate distributions. GANs are deep learning alternatives to traditional generative models based on maximum likelihood estimation. Their advantage lies in the approximation of complex data distribution and taking advantage of the piece-wise linear units in a generative framework (Goodfellow et al. (2014)).
GANs are having a lot of success among generative methods in the recent years, especially for human faces images generation. The resolution and the quality of the pictures produced by the generator have seen rapid improvement recently (Figure 1.20), thanks to different studies which have implemented new architectures or methods of training for this kind of network. The images generated by one of the most advanced GAN architecture are extremely detailed and realistic that humans struggle to consistently identify faces of real persons from fake ones (check the website `https://thispersondoesnotexist.com/`).

Figure 1.20: The improvement in the generation of human faces images in the recent years, each image is generated by a different GAN architecture. 2014: Goodfellow et al. (2014), 2015: Radford et al. (2015), 2016: Liu and Tuzel (2016), 2017: Karras et al. (2017), 2018: Karras et al. (2018).

In this project the Progressive Growing GAN (PGGAN) developed in Karras et al. (2017) has been adopted, which will be further described in section 3.1. Although this network no longer represents the state-of-the-art (Karras et al. (2018)), its capabilities are sufficient for the resolution of the images considered in this thesis project.

1.5. Neural Network application in Meteorology

Weather prediction is traditionally done by solving numerically the partial differential equations that are believed to govern the atmosphere-ocean system. The differential equations are solved with advanced models, which improvement in the forecast is strictly related to the increase of available computing resources. Not only they are computationally expensive but numerical weather models require advanced physical parameterization schemes for unresolved processes which have to be adjusted every time the resolution is increased (Bihlo (2020)). Recently, with the improvement of deep learning frameworks different studies have tried to implement Neural Networks based methods to address the challenging problem of weather forecasting (Liu and Lee (2020)).

1.5.1. Tropical Cyclone forecast

Considering the forecast of hurricane tracks and intensity there are already different studies using Neural Network methods. In Giffard-Roisin et al. (2018) the authors propose a neural network architecture which considers past trajectory of the cyclones and reanalysis atmospheric wind images to estimate the future displacement of each storm at each time step and depressions. The neural network architecture is composed by a two layers NN, which is able to determine the next 6 hours track knowing the past track of a typhoon, and a Convolutional Neural Network (CNN) to reproduce the wind field associated to the cyclone.

In Ali et al. (2007) the authors estimates the future positions in 24 hours for tropical cyclones in the North Indian Ocean using a NN. To do so they just use the past 12 hours locations at time intervals of 6 hours. The mean error they obtain in the distance between the actual and the predicted positions is 137.5 km. The best forecast is for the positions after 12 hours and the model accuracy decreases beyond 24 hours forecast.

In Baik and Paek (2000) a neural network is used as model for predicting tropical cyclone intensity changes in 72 hours with a time step of 12 hours in the western North Pacific. More in particular in this study 4 different kind of predictions models are built to compare the difference in performance between a neural network and a multiple linear regression model. The 4 models are a neural network with climatology and persistence predictors (N-CP), a multiple linear regression model with climatology and persistence predictors (R-CP), a neural network with climatology, persistence and synoptic predictors (N-CPS) and a multiple linear regression model with climatology, persistence and synoptic predictors (R-CPS). Between all the models N-CPS present the best performance in predicting tropical cyclone intensity. The authors also evidence a tendency of the estimation error to decrease with the increase of the number of units in hidden layers.

Loridan and Crompton (2017) discuss the possibility of using machine learning algorithms as an alternative to parametric models in the representation of TC wind fields. The methodology proposed on the wind field data is as follows: a Principal Component Analysis (PCA) is performed on the training data which produces a set of Principal Components (PCs) and corresponding weights with which is possible to reconstruct the TC wind fields; after that supervised learning algorithms are applied that can learn to predict the TC wind fields by reconstructing them using the PC weights and a set of input features.

In Ruttgers et al. (2019) a framework to predict typhoons tracks using a GAN with satellite images as inputs is proposed. The neural network is constructed to produce a 6-hour-advance track for a typhoon that was not present in the training set. The input set employed by the GAN are satellite cloud images together with the location of the typhoon center, no auxiliar information regarding moving direction or speeds at the typhoon center are considered. The model successfully reproduce the 6-hour image track and is able to identify the future location of the typhoon center together with the deformed cloud structures. The authors underline how the predictions are significantly improved by including the wind velocity fields together with the satellite images.

1.5.2. Precipitation Nowcasting

Another interesting fields of meteorology where neural networks have been applied is precipitation nowcasting, which has long been an important problem in the field of weather forecasting. Precipitation nowcasting consists in generating prediction of rainfall intensity in a local region over a short period time, which usually is between 0 to 6 hours (Shi et al. (2015)).

Precipitation nowcasting can be treated as spatio-temporal sequence forecasting problem where both the inputs and the outputs of the model are sequential data. In Shi et al. (2015) precipitation nowcasting is modelled with a sequence-to-sequence learning framework, a Convolutional Long Short-Term Memory (ConvLSTM). LSTMs are units which can be implemented in a Recurrent Neural Network (RNNs) to replace a traditional neuron. RNN is a specific deep learning framework which works with sequences of data as input, output or both of them. In the architecture of a RNN the nodes/neurons are connected in a directed graph so that the network can learn the temporal dynamic behaviour of the data (Sherstinsky (2020)). LSTM units are having a successful implementation because they are able to store information regarding values in the input sequence and relate it with the following values in the sequence. This enables the network to better learn the connection in the sequential data and how the past, the first values of the sequence, influences the future, the last values of the sequence (Hochreiter and Schmidhuber (1997)). The ConvLSTM network applied by Shi et al. (2015) works with sequence of past radar maps images as input and a sequence of fixed number of future radar maps as output. The model proposed in based on the sequence-to-sequence general framework introduced in Sutskever et al. (2014). This pioneering LSTM encoder-decoder adopts temporally concatenated fully connected LSTM layers (FC-LSTM) which are demonstrated to perform worse than the convolutional LSTM layers because they do not take into account the spatial correlation in the images. ConvLSTM shows to better capture spatio-temporal correlations and consistently outperforms not only the FC-LSTM but also state-of-the-art operational ROVER algorithm for precipitation nowcasting. It is also observed that ConvLSTM is able to handle well boundary conditions, like a sudden agglomeration of clouds appearing at the boundary, if it has seen them during training (Shi et al. (2015)). ConvLSTM has been built as an encoding-forecasting framework for precipitation, but it can be applied to more general spatio-temporal sequence forecasting problem.

In Shi et al. (2017) the authors go beyond ConvLSTM and propose TrajectoryGRU (TrajGRU) model which can learn the location-variant structure for recurrent connections. In fact, considering atmospheric satellite images the correlation between local motion patterns, like rotation and scaling, in consecutive frames will be different for different locations and times. Using a ConvLSTM is inefficient because the filter used are location-invariant while the relationship is location-variant. TrajGRU architecture is built with a sub-network that outputs the state-to-state connection structures before state transitions. TrajGRU present the best overall performance among the other methods for precipitation nowcasting (FC-LSTM, ConvLSTM and ROVER algorithm) (Shi et al. (2017)).

1.5.3. Weather Forecasting

In Bihlo (2020) a conditional generative adversarial network (cGAN) is trained to predict the geopotential height of the 500 hPa pressure level, the two-meter temperature and the total precipitation for the next 24 hours over Europe. The author wants to investigate how well the network is able to learn the distribution underlying the atmospheric system. That is investigating if the cGANs have the ability to learn the physics related to the meteorological data. He also wants to quantify the uncertainty in weather forecasts using ensemble predictions and machine learning. The cGAN model is trained using 4 years of ERA5 reanalysis data from 2015 to 2018 (Hersbach et al. (2020)). The problem of weather forecasting can be interpreted as a video prediction from the machine learning perspective, the past meteorological fields are the input frames and the future meteorological fields are the output frames (Bihlo (2020)). The forecasts produced for the meteorological fields in 2019 by the machine learning framework show a good qualitative and quantitative agreement with the true reanalysis data. This is valid for the two-meter temperature and for the geopotential height, while there is mostly disagreement for the total precipitation. The fail of representing rainfall correctly is probably related with the low resolution of the ERA5 dataset, in fact precipitation is mostly localized and convective precipitation in particular happens at scale which are definitely smaller than the one of ERA5. For the model is difficult to learn meaningful signals from rainfall data. The results obtained for the other parameters show that the architecture is able to learn some of the basic physical processes governing the atmosphere and to quantify the uncertainty behind the evolution of the atmospheric system (Bihlo (2020)).

Liu and Lee (2020) also address the problem of weather forecasting, they do it by introducing a machine learning architecture which uses a conditional GAN coupled with a Convolutional LSTM, which they also define as predictive learning module. The authors justify the coupling of the two models to overcome the difficulties that the two methods have shown in other applications. In fact, ConvLSTM generally produces blurry images, and the effect worsen as the time step moves forward. GAN methods instead are able to generate realistic looking video frames but they fails in representing the actual atmospheric movement with local variations and patterns (Liu and Lee (2020)). The ConvLSTM models the meteorological patterns of the atmosphere and the conditional GAN works to map back to non-blurry imagery distributions the predictions generated by the predictive learning. The model is able to generate correctly weather forecast predictions and to avoid that the images produced remains in the original data distribution without showing a blurry nature (Liu and Lee (2020)).

2

Data acquisition and domain definition

The values of the atmospheric variables of interest are collected from the ERA5 dataset. The dataset is the latest reanalysis of the European Centre for Medium-Range Weather Forecasts (EMCWF), it incorporates a detailed global record of atmospheric, ocean waves and land surface variables and parameters. The reanalysis covers the period going from 1979 until present time, for the project a time span of 40 years has been considered (1^{st} January 1979 - 1^{st} January 2020). The datasets presents a finer resolution with respect to the previous ECMWF reanalyses, with a spatial resolution of 31 km and hourly availability (Hersbach et al. (2020)). The data are downloaded through the Python based CDS API in netCDF format. The atmospheric variables collected are mean sea level pressure, 10 metre U wind component, 10 metre V wind component and mean total precipitation rate. They are reported, together with their units, in Table 2.1. During the creation of the cyclones dataset pressure and mean total precipitation rate were converted from Pa and $\frac{kg}{m^2 s}$ to hPa and $\frac{mm}{h}$.

Variable	Unit
Mean sea level pressure	Pa
10 metre U wind component	$\frac{m}{s}$
10 metre V wind component	$\frac{m}{s}$
Mean total precipitation rate	$\frac{kg}{m^2 s}$

Table 2.1: Atmospheric variables considered for the fields generation

The domain considered for the atmospheric fields is a squared domain with coordinates of the horizontal boundaries corresponding to 0° and 90°N, and coordinates of the horizontal boundaries corresponding to 70°W and 20°E (Figure 2.1). The choice of a squared domain is determined by the characteristic architecture of the GAN adopted in the project, in fact the images used to train the network are down-sampled and up-sampled different times as it would be more clear in section 3.1. The availability of squared images makes this kind of operations easier to handle and avoids several modification to the network's architecture. Also a rectangular input requires the filters in the convolutional layers to be asymmetrical, so again it requires to modify the architecture of the network accordingly. For what concerns the coordinates of the domain the choice is motivated by the possibility to include all the life cycle of a cyclone occurring in the North Atlantic. Generally, the cyclogenesis occurs along the east coast of North America and the cyclones follow a path that brings them over Europe or even to the high latitudes of the Northern Hemisphere, see Figure 1.11 and 1.12.

2.1. Extra-Tropical Cyclones dataset

The ETCs dataset is created considering the record of events contained in the Extreme Wind Storms Catalogue and in the STORMS Extratropical Cyclone Atlas (Roberts et al. (2014), Dacre et al. (2012)). From this record the start date and the end date of the cyclones were considered to extract the corresponding fields from the ERA5 reanalysis data. The creation of the dataset is performed following two steps: first all the interested fields are extracted from the netCDF monthly files and organized by cyclone track; then each atmospheric field in each track is down-sampled to a 64x64 resolution and normalized considering the historic maximum value and minimum value occurred in the North Atlantic. This two operation are necessary to help the network to learn faster and better, in fact a higher resolution makes more detailed the features that the network has to learn and the availability of data is not enough to guarantee a stable training. Normalization helps stabilize the training because the network deals with small numbers in absolute value. All the fields down-scaled and normalized are stacked together in a single file which will be used to train the network. The total number of fields obtained from this operation is 20'346 for each variable and it has been used to initially train the network. Deep learning methods, and especially GANs, require a big number of examples in the training set to obtain meaningful results. The networks trained with training set containing 20'346 examples have not shown valid results. The scarce number of fields adopted for training is considered to be the cause for the mediocre fields generated by the models.

To increase the amount of data fields representing ETCs available during training the wind speed is considered. In particular, if the 10m zonal wind speed surpasses a threshold of 17 m/s at a specific chosen location of coordinates (30°W, 50°N) (Pinto and Ludwig (2020)), the hour and date are relevant to extract a new storm track. It is believed that when the wind speed surpasses the threshold in the location of interest a cyclone event is happening in the North Atlantic. Therefore, to have more images which most likely contain such a phenomenon, also the fields in the 12 hours of reanalysis data before and after the moment selected considering the zonal wind speed are extracted. The data augmentation leads to an increased number of tracks and fields for each atmospheric variable, respectively 12'228 tracks and 43'313 fields.

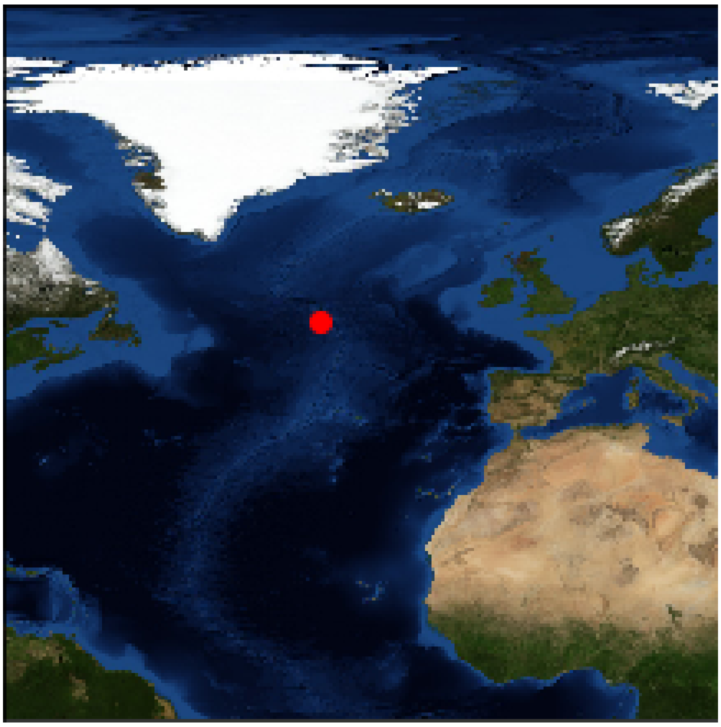


Figure 2.1: The domain chosen for the project, it covers most of the North Atlantic and the European region were ETCs usually make landfall. The red dots indicates the point with coordinates (30°W, 50°N) where the zonal wind speed is considered to extract new storm tracks.

3

Network set up and training

The atmospheric variables fields of pressure, wind and precipitation produced in the ERA5 dateset can be considered as images. In the recent years a particular architecture of Neural Network has become increasingly popular for images and videos generation, the Generative Adversarial Network (Goodfellow et al. (2014), Goodfellow (2017)). This kind of generative method is finding use for different application and it is outperforming other existing techniques, achieving a better quality and resolution of image synthesis and editing (Iizuka et al. (2017), Lin et al. (2018), Radford et al. (2015), Wang et al. (2018)).

Different studies have shown positive performances of GANs in generating convincing images starting from datasets with low variability and low resolution (Denton et al. (2015), Radford et al. (2015)). However, GANs present high difficulty to generate coherent and significant samples for high resolution images, especially from datasets with high variability (Odena et al. (2017)). Until a few years ago, GANs were limited to images of less than 100-pixel square (e.g., 32x32) (Brownlee (2019b)). Generating images with high resolution is complicated for such a network because of a higher level of details in the data. It makes it easier for the discriminator to distinguish the generated images from the real ones. It makes it more difficult for the generator to learn to output images which both have a large structure and fine details (Karras et al. (2017)).

To deal both with the instability during training and the problem of generating high resolution images the Progressive Growing of GANs has been adopted (Karras et al. (2017)). This is a particular method to train a Generative Adversarial Network. It consists of growing progressively both the generator and the discriminator, starting the training with low resolution images and then increasing the details and adding new convolutional layers as the training progresses (Brownlee (2019b)).

To train the network the Wasserstein distance has been adopted, due to its recent success with learning of generative models (Arjovsky et al. (2017), Weng (2019)). However, because of particular trends in the images generated by the Wasserstein PGGAN, another loss function has been considered for training, the Cramèr distance (Bellemare et al. (2017)). As training algorithm for optimizing the network ADAM has been chosen, with $\beta_1 = 0.0$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$.

The PGGAN's training was initially performed on Google Colab NVIDIA Tesla K80 GPU and on Kaggle NVIDIA Tesla K80 GPU. Aferwards SurfSara has granted the access to the accelerator island on Cartesius system, so it was possible to perform the train on their NVIDIA Tesla K40m GPUs nodes.

The PGGAN model has been defined and trained using Keras deep learning API available on Python and running on top of the machine learning platform TensorFlow. The code which defines the model architecture and the training procedure has been adapted from to the one posted by Brownlee (2019d), with some modification for the project specific application. The source code is available at the following Github repository https://github.com/Carmelo-Belo/pggan-etcs.git.

3.1. Progressive Growing Generative Adversarial Network

The Generative Adversarial Network is a particular deep learning technique to make a generative model learn to reproduce samples from a desired data distribution. In particular, the GAN is an effective method to train a Convolutional Neural Network to generate images similar to those in the training data, which is one of the purpose of this project (Brownlee (2019b)). In this framework the generative model, or generator, is trained in competition with an opponent, the discriminator. The discriminator learns to understand if a data sample is coming from the model distribution or the data distribution. The generator learns to produce counterfeits which resemble the most the original data distribution. The adversarial game between the two models optimizes both methods until the samples produced by the generator cannot be distinguished from the original data (Goodfellow et al. (2014)).

GANs are able to produce clear images, even though rather small resolution and low variability must be considered and the training remains unstable and strongly dependent from initialization (Karras et al. (2017)). GANs training on large images also demands a greater space on GPU memory, which is in relatively limited supply compared to main memory. For this reason, the batch size which comprehends the number of images used to update the parameters at each iterations must be reduced to allow the images to fit into memory. However a smaller size of the minibatches further compromise training stability (Brownlee (2019b), Karras et al. (2017)).

Trying to resolve the problem of training stability on larger images Karras et al. (2017) propose a new approach called Progressive Growing GAN (PGGAN). This method consists in progressively growing the number of layers in both the generator and the discriminator during training, starting from easier low-resolution images and adding new layers which develop higher-resolution details as the training progresses (Figure 3.1). Progressive growing of GANs greatly accelerates training and reduces instability at higher resolution (Karras et al. (2017)). The incremental nature of the training brings different benefits. Large-scale structure in the image distribution are first discovered and then the attention is progressively shifted to finer scale details of the images, avoiding to learn all scales simultaneously, which is more complex. By consequence the generation of smaller images is substantially more stable because all the information regarding higher resolution is not considered (Karras et al. (2017), Odena et al. (2017)). Most of the iterations during training are done at lower resolutions and the training time is reduced up to 2-6 times faster at comparable result quality (Karras et al. (2017)).

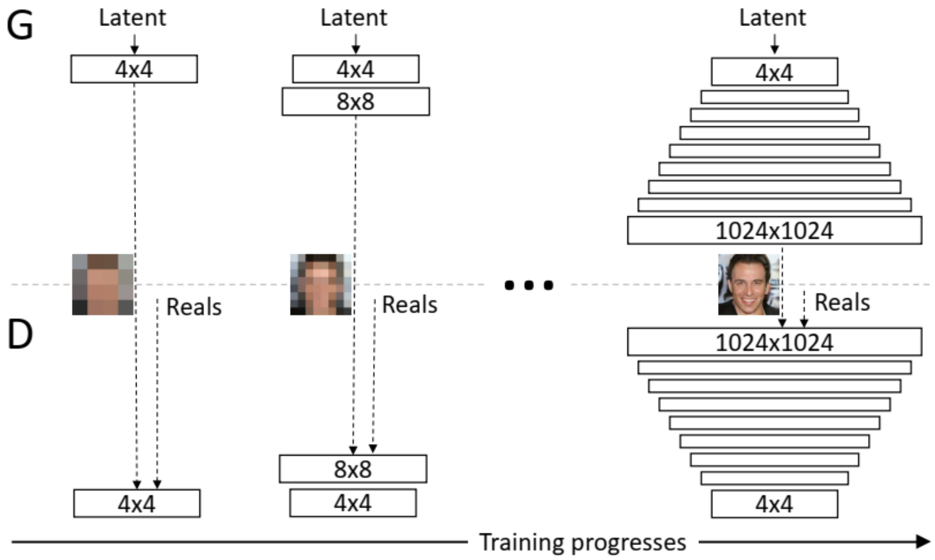


Figure 3.1: Progressive growing of a GAN: it starts at low resolution (4x4) and as the training advances new convolutional layers are added to both the generator (G) and the discriminator (D), so that the spatial resolution of the generated images is increased. All the existing layers remain trainable during the process (Karras et al. (2017)).

3.1.1. Progressive Growing of a GAN

The Progressive Growing of a GAN involves adding new blocks of convolutional layers both to the generator and the discriminator along with the advancement of the training. A new block is added every time the resolution is scaled-up, following the power of 2. It starts from a images resolution of 4x4 and it progressively grown until 1024x1024, the steps are 8x8, 16x16, 32x32, 64x64, 128x128, 256x256, 512x512. The addition of a new block of layers is not done directly but they are faded in smoothly to avoid sudden shocks to the already trained smaller-resolution layers (Karras et al. (2017), Brownlee (2019b)). To phase in a new block of layers a skip connection is utilized to connect the input of the discriminator or the output of the generator to the new block. The influence of the new block of layers on the existing input or output layer is controlled with a weighting, using a parameter α that starts from a very small number or 0 and linearly increases to 1 over training iterations (Brownlee (2019b)). Skip connections are deep architectures features which allow to skip some layer in the neural network and feed the output of one layer as the input of the next layers. Figure 3.2 shows the transition from 16x16 images to 32x32 images.

Considering the generator the output of the lower resolution layer is upsampled using nearest neighbour interpolation to the higher resolution and its output is combined to the output of the new block of layers. The contribution of the already existing upsampled layers is weighted by $(1-\alpha)$ while the contribution of the new layers is weighted by α. For what concerns the discriminator the input image is downsampled to the lower resolution using average pooling so it can be fed to the existing convolutional layers (Brownlee (2019b)). For each growing phase of the training, i.e. each different stage of images resolution, the network is trained in two steps, a faded one where the weighting process is applied and a tuned one where only the higher resolution is considered. All layers remain remain trainable during the all training process for both networks (Karras et al. (2017)).

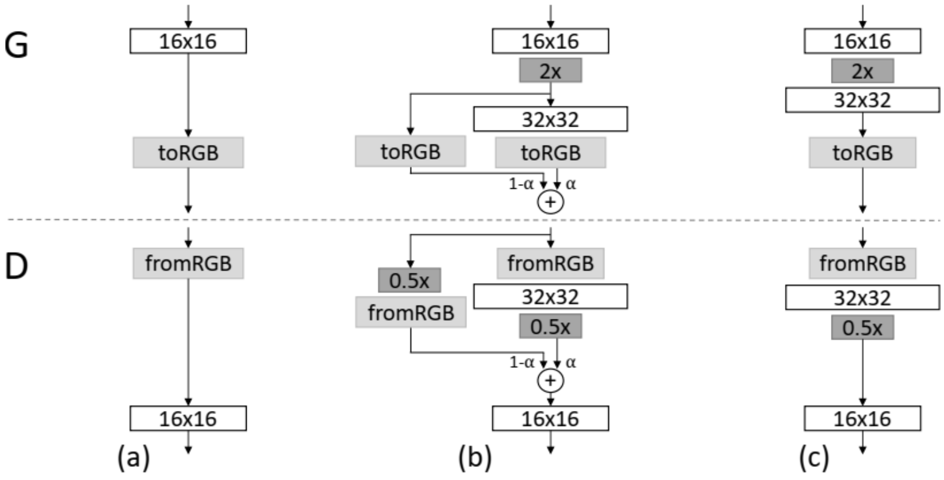


Figure 3.2: Transition from 16x16 images (a) to 32x32 images (c). The transition (b) happens by fading in the new block of layers and happens through the skip connection which is controlled by the weight α that linearly increases from 0 to 1. The toRGB and fromRGB box represents the layer operating that converts feature vectors to RGB colors and vice-versa. The 2x and 0.5x box are respectively a nearest neighbour filtering to upscale the resolution and and average pooling to downscale it (Karras et al. (2017)).

3.1.2. PGGAN architecture

Both the generator and the discriminator are Convolutional Neural Networks. At each step of the progressive growing a block of convolutional layers is added to both of them. It consists of two layers having a filter size of 3x3, the number of filters varies and is specific for each layer. The activation function for the convolutional layer is a Leaky ReLU with the slope of 0.2, except for the last layer which has a linear activation function. Leaky ReLU modifies the ReLU function by allowing small negative values when the input is less than zero. The bias in all the layers is initialized to zero and the model's weights are initialized as random Gaussian using the He weight initialization method (Brownlee (2019a), Kumar (2017)).

The model uses two custom layer, *Minibatch standard deviation* and *Pixel-wise normalization*. The first one is only used in the discriminator output block and its purpose is to determine a statistical summary of the batch of activations. This feature helps the discriminator to better detect fake samples from real samples and also brings the generator to create batches of samples with realistic statistics, since the generator is trained through the discriminator. In this layer there are no new learnable parameters or new hyperparameters, its task is to compute the standard deviation for each pixel in the activation maps over the batch, then it computes the average of these estimates over all pixels to obtain a single value. This single value is then concatenate to all spatial locations to construct a new constant activation map which is appended to the list of activation maps provided as input (Brownlee (2019d), Karras et al. (2017)). Pixel-wise normalization layer is implemented in the generator after each convolutional layer. The feature vector is normalized in each pixel to avoid the scenario where the competition brings the magnitudes to spiral out of control. The activations of the prior layer are then normalized by the L2 norm (Brownlee (2019d), Karras et al. (2017)).

The overall architectures of the two PGGAN's components, considering images with final resolution of 64x64, are contained in Table 6.1 and 6.2 in Appendix 6.

3.2. Network loss function

The choice of the loss function for a neural network is crucial, in fact the definition of the loss function determines the structure/shape of the objective space where the gradients for the weights update are computed. So, the loss function really affects the training procedure and its stability. Wasserstein loss or Earth's Mover (EM) distance has been proved to be one of the most effective loss to be implemented in GAN's training. It significantly increases the training stability and it drastically reduces the mode collapse phenomenon typical of GANs (Arjovsky et al. (2017)). The EM distance is a measure of the gap between two probability distributions, for a continuous probability domain is defined as:

$$W\left(p_r, p_g\right) = \inf_{\gamma \sim \Pi\left(p_r, p_g\right)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|] \tag{3.1}$$

where $\Pi(p_r, pg)$ is the set of all possible joint probability distributions $\gamma(x, y)$ between p_r and pg. Figuratively, the Wasserstein distance is the minimum energy cost of converting and moving mass from the shape of one probability distribution to the shape of the other distribution (Weng (2019)). In other words, is the cost for the optimal transport plan and is quantified by the amount of mass moved multiplied by the moving distance.

The Wasserstein loss function cannot be implemented in Keras as it is described in Arjovsky et al. (2017). The implementation adopted follows the solutions of Brownlee (2019d). The discriminator model is considered as a critic that scores the realness or falseness of a given image and the loss is defined as the average predicted score multiplied by a label, which is 1 for real images and -1 for fake images (Brownlee (2019e)). A smaller Wasserstein distance denotes a similar distribution in the training images and in the generated sample, meaning that both appear similar in appearance and variation (Brownlee (2019d), Karras et al. (2017)).

The results obtained by the PGGAN trained with the Wasserstein distance shows some unwanted artefacts in the generation of ETCs fields, independently from the atmospheric variables considered. For this reason another distance is considered, the Cramèr loss function (Bellemare et al. (2017)). This distance is proposed by the authors of the study as a more valid and improved loss function for GAN training on images. Cramèr distance is characterized by sum invariance and scale sensitivity, the same two properties that made successful Wasserstein distance in generative modelling. Additionally, the Cramèr distance is unbiased to sample gradients (Bellemare et al. (2017)). It is defined as:

$$l_2^2(P, Q) = \int_{-\infty}^{\infty} \left(F_P(x) - F_Q(x)\right)^2 \mathrm{d}x \tag{3.2}$$

The experiments performed by Bellemare et al. (2017) show that the GAN trained with the Cramèr distance leads to an increased diversity in the generated sample and to a more stable learning.

3.3. Network Training

The training of a Generative Adversarial Network consist in a two-player minimax game between the discriminator and the generator, where the two networks are trained in parallel. The generator is represented as a mapping of the input noise variables $\mathbf{z}$ to data space as $G(\mathbf{z},\theta_g)$, where G is a multilayer differentiable function and θ_g are its parameters. Similarly the discriminator is defined as $D(\mathbf{x},\theta_d)$, mapping the input data to a single scalar, D is a multilayer differentiable function and θ_d are the parameter of the discriminator. $D(\mathbf{x})$ represents the probability that $\mathbf{x}$ came from the data rather than the generator's distribution. The purpose of the training is to maximize the probability of D assigning the correct label to both training examples and samples from G and to minimize the $log(1-D(G(\mathbf{z}))$ (Goodfellow et al. (2014)). Mathematically the zero-sum non-cooperative game consist in optimizing the value function $V(G,D)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1-D(G(\mathbf{z})))] \tag{3.3}$$

The training process is performed as a simultaneous Stochastic Gradient Descent, in this project the Adam optimization algorithm has been adopted (Kingma and Ba (2017)). It is an upgraded variant of the SGD which also considers the exponentially weighted moving average of the gradients and of the squared gradients. On each step of gradient descent two mini-batches are sampled: one from the training dataset of $\mathbf{x}$ values, the other one from the latent space of random $\mathbf{z}$ values. The optimization is done simultaneously, updating the weights θ_d and θ_g to reduce the corresponding objective function J_d and J_g defined according to the loss function. The losses computed during training are three. The first one (d1) is the loss of the discriminator with real data utilized, the second one (d2) is the loss of the discriminator with fake data utilized and the third one (g) is the loss of the generator.

The training process is governed by the model hyper-parameters, which are properties that control the behaviour of the training. They can be grouped by the variables which determines the networks structure (e.g. number of layers, number of hidden units) and the variables which determines how the network is trained (e.g. learning rate, number of batches). The hyper-parameters are set before training and progressively adapted in order to optimize the network. Training a deep learning framework is usually a trial and error process. In fact, at the beginning of the operation you do not know which are the value of the hyper-parameters which will guarantee the network to find the best solution of local optimum of the objective function. For this reason you start with initial guess and you proceed by adapting the hyper-parameters values accordingly to the losses computed during the training.

Without considering the variables involved in the network architecture, which is described in section 3.1.2, the hyper-parameters for the training of the PGGAN are:

- **Learning rate:** it controls how much the network's parameters have to change when they are updated in response to the estimated error. The choice of the learning rate is challenging. If the value is too small the time to find an optimal solution will increase and the training risks to get stuck in a not ideal state. Whereas, if the value is too big the gradient descent steps might be too long. This causes the algorithm to not properly explore the cost function space. As a consequence the training converges in a sub-optimal solution or does not converge at all becoming unstable. A classic start for the learning rate values is 0.001.

- **Latent space dimension:** is the dimension of the random values vector which operates as input of the generator. Setting the values of this variable is also challenging. In fact, with a small dimension of the latent space there is the risk that the network is not able to replicate with the same level of detail the variability in the sample distribution. While, with a big dimension the risk is to make the generation of fake data too complex because of the high level of details that is passed as input to the generator.

- **Dimension of batches:** it determines in how many subsets the training set wants to be divided. The weights are updated for each mini-batch of data. Large resolutions needs to have smaller mini-batches due to memory constraints (Karras et al. (2017)). However a high number of mini-batches risks to compromise training stability, introducing more noise in error calculations. In the PGGAN training, because of the different growth phases, there is the need to set a dimension of batches for each resolution step, which in this case are 5. To better handle the tuning a reference number of batches per growth phase was defined, it consists in a vector [16, 16, 16, 8, 8]. It is adjustable by a multiplier which is the only value to modify. This solution result in a more straightforward and clear choice of the number of batches per growth phase.

- **Number of epochs:** an epoch correspond to a an entire iterations through the training dataset. An epoch does not necessarily correspond to a single optimization of the network, in fact in the case of Mini-Batch Gradient Descent the parameters are updated for each mini-batch, so more then once for an epoch. The number of epochs strongly influences the training and it is important to be careful to train the network for as many number of iterations as long as the error keeps decreasing. Similarly to the number of batches a fixed vector equal to [5, 10, 10, 15, 15] and a number of epochs multiplier were defined to better handle the tuning of the hyper-parameter.

The network is trained both for single as well as for combined fields generation. First, GAN performance for single fields is verified, by training a different network for each atmospheric variable separately to obtain 4 distinct field generators. Afterwards, by combining the different fields together as channels of a single image, multi field generators are verified. By seeing more than one field at the same time the network might be able to better learn the atmospheric structure behind the variables considered. The trained models are listed in table 4.1, for both single field representation and different combinations of multi-fields representation Wasserstein and Cramèr distance have been considered as loss function. Unfortunately it was not possible to perform multi-fields representation involving all the 4 atmospheric variables. The limited amount of simulation hours on Cartesius did not make it possible to perform the training and the limited amount of space on memory in Kaggle and Colab could not allow to finish it.

3.3.1. The Adam optimizer

Adam stands for adaptive moment estimation and it is an algorithm for first-order gradient-based optimization of stochastic functions, based on moving averages estimates of lower-order moments. The estimates of first and second moments of the gradients are utilized to computes individual adaptive learning rates (Kingma and Ba (2017)). Computing adaptive learning rates means that Adam estimates individual learning rates for different parameters. The weights update is done accordingly to the following procedure:

$$m_W^{(t+1)} = \beta_1 m_W^{(t)} + \left(1-\beta_1\right) \nabla_W L^{(t)} \tag{3.4}$$

$$m_b^{(t+1)} = \beta_1 m_b^{(t)} + \left(1-\beta_1\right) \nabla_b L^{(t)} \tag{3.5}$$

$$v_W^{(t+1)} = \beta_2 v_W^{(t)} + \left(1-\beta_2\right) \nabla_W \left(L^{(t)}\right)^2 \tag{3.6}$$

$$v_b^{(t+1)} = \beta_2 v_b^{(t)} + \left(1-\beta_2\right) \nabla_b \left(L^{(t)}\right)^2 \tag{3.7}$$

$$\hat{m}_W = \frac{m_W^{(t+1)}}{1-\beta_1^{t+1}} \tag{3.8}$$

$$\hat{m}_b = \frac{m_b^{(t+1)}}{1-\beta_1^{t+1}} \tag{3.9}$$

$$\hat{v}_W = \frac{v_W^{(t+1)}}{1-\beta_2^{t+1}} \tag{3.10}$$

$$\hat{v}_b = \frac{v_b^{(t+1)}}{1-\beta_2^{t+1}} \tag{3.11}$$

$$W^{(t+1)} = W^{(t)} - \alpha \frac{\hat{m}_W}{\sqrt{\hat{v}_W} + \epsilon} \tag{3.12}$$

$$b^{(t+1)} = b^{(t)} - \alpha \frac{\hat{m}_b}{\sqrt{\hat{v}_b} + \epsilon} \tag{3.13}$$

Equation 3.4, 3.5 are the estimates of the exponentially weighted moving averages of the gradients, 3.4 is also called momentum term. $\beta_1 \in [0,1)$ is an hyper-parameter which controls the exponential decay rate of the moving average. In other words β_1 controls the window of previous gradients that are considered in the average ($window = \frac{1}{1-\beta_1}$), if $\beta_1 = 0.9$ the window is equal to 10, so the average considers the last 10 gradients. The momentum term helps smoothing out the update and removing the oscillations in the objective function space that might happen with normal gradient descent. In fact, by taking the average of the gradients the algorithm goes for a more direct path towards the local optimum instead of oscillate on the slope. The momentum term helps accelerate gradients in the right direction and also makes the gradient descent algorithm take larger steps and converge faster. The property to store the gradients for use in the next updates of the parameters cause this.

Equation 3.6, 3.7 are the estimates of the exponentially weighted moving averages of the squared gradients. $\beta_2 \in [0,1)$ controls the average in the same way as β_1 does. The second order moving average operates to normalize the gradients, this normalization balances the step size, decreasing the steps for large gradients to avoid exploding, and increasing the step for small gradients to avoid vanishing.

Equation 3.8, 3.9, 3.10 and 3.11 are bias correction of the estimates. Because the moving averages in Equation 3.4 to 3.7 are initialized to 0, the moment estimates are biased towards 0, in particular during the initial timesteps of the learning and when the decay rates β_1, β_2 are small, that is close to 1 (Kingma and Ba (2017)). This problem is solvable by applying bias correction which makes the computation of the averages more accurate and does not affect it too much when t is large.

Equation 3.12, 3.13 are the parameters update formulas for Adam algorithm. ϵ is a very small term, usually equal to 10^{-8}, to avoid dividing by 0.

3.3.2. Training configuration

The learning of a neural network is characterized by trial & error and by the modeler intuition on how to tune the hyper-parameters accordingly to the networks answers to optimization. In fact, even if it is possible to expect some different behaviour from the network in accordance to an increase or a decrease of specific hyper-parameters, most of the time is difficult to predict how the network will change after an hyper-parameter variation or what could be the best hyper-parameters configuration to optimize the network. Deep neural networks can be interpreted as black box from which is generally very difficult to understand the behaviour and the influence of the structure on the results produced. For this reason training a neural network is challenging and there is not a universal value of number of layers, learning rate, batches dimension, etc., but rather the optimal value of the hyper-parameters changes between applications.

After a consistent number of trials for the training configuration, in respect of the losses values and the images generated by the generator, an optimal set of values for the hyper-parameters has been found. It is the following:

- Learning rate: $\alpha = 10^{-5}$
- Latent space dimension: $z = 128, 256$
- Batches dimension multiplier: $b_m = 8$
- Number of epochs multiplier: $e_m = 2$
- Adam's exponential decay rate 1^{st} moment: $\beta_1 = 0.0$
- Adam's exponential decay rate 2^{nd} moment: $\beta_2 = 0.99$

An interesting aspect of the configuration adopted to train the network is the value of β_1, which generally in deep learning application is equal to 0.9 (Kingma and Ba (2017)). In fact, during the hyper-parameters tuning a considerable improvement in the losses values and in the quality of the generated images is evident in the shift of β_1 from 0.9 to 0.0. $\beta_1 = 0.0$ is also adopted by Karras et al. (2017), denoting that most likely for PGGAN training with Adam optimizer this is the best hyper-parameter configuration.

With $\beta_1 = 0.0$ the exponentially moving average of the first moment is just equal to the gradient at that update step. By consequence the momentum term has no influence on the learning algorithm and the gradient descent in the objective function space. The only presence of the exponentially moving average of the second moments makes the Adam algorithm downgrade to RMSprop learning algorithm, which is part of the adaptive learning rate methods (Bushaev (2018)). Knowing for certain which is the cause that makes the learning algorithm work better without considering the momentum terms is difficult, but intuitively it can be related with the gradient steps in the optimization space. In fact, considering that the momentum term makes the gradient descent take larger steps along the direction of optimization it might be possible that by consequence the training cannot converge to a local minimum. Because of steps that results too large,

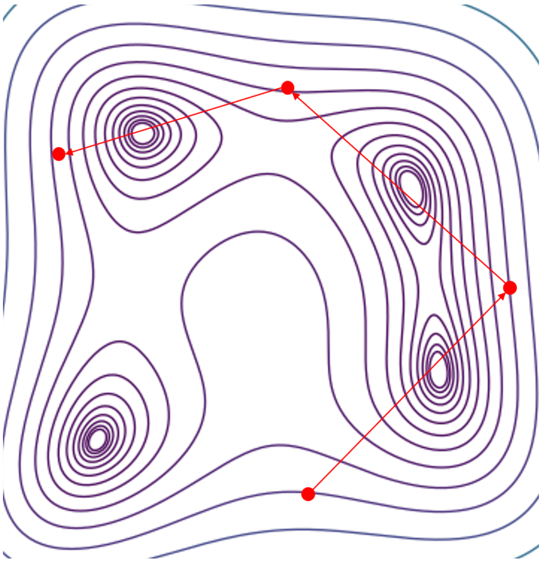


Figure 3.3: Example of a situation where the training algorithm does not converge to a an optima solution in a 2 dimensional space with 4 local minima. Despite gradient descent follows the steepest directions it just ends up on the other side of the optimal points since the steps are way larger than the points' attraction basins. Because of this behaviour the learning algorithm never converges to an optimal solution.

the training algorithm instead of descending in one of the function local minimum surpasses the optimal points and never improves the weights value, a simpler but intuitive example of the situation in a 2D space is illustrated in Fig 3.3.

4

Results and Discussion

The evaluation of the results produced by the generator relies on two important factor: diversity and visual fidelity or quality of the images. Diversity in the images means that the distribution learned by the GAN covers all possible modes present in the original distribution. The extreme case involving diversity is mode collapse, a GANs' structural behaviour which implies the generator to create very similar images with almost no diversification (Bau et al. (2019)). Visual fidelity means that the samples generated by the generator resemble the original one and the structure of the generated images corresponds to the structure of the original images. For what concerns ETCs, visual fidelity involves aspect like location of the regions of low and high pressure and shape of the former, pressure gradient around the points of local minimum which are considered cyclone centers, winds direction and speed in the proximity of the cyclone centers and structure of the mesoscale rain bands. To evaluate the models for each different training setup the generated samples have the same dimension of the original sample if not differently specified. The models considered for evaluation are listed in Table 4.1. Because of the poor model performances the Table does not list any GAN trained on rainfall fields using the Wasserstein loss.

Model ID	Atmopsheric Variables	Loss function
W-P	Pressure	Wasserstein
W-U	Zonal wind	Wasserstein
W-V	Meridional wind	Wasserstein
W-PUV	Pressure, Zonal wind, Meridional wind	Wasserstein
W-PR	Pressure, Precipitation	Wasserstein
W-UV	Zonal wind, Meridional wind	Wasserstein
C-P	Pressure	Cramèr
C-U	Zonal wind	Cramèr
C-V	Meridional wind	Cramèr
C-R	Precipitation	Cramèr
C-PUV	Pressure, Zonal wind, Meridional wind	Cramèr
C-PR	Pressure, Precipitation	Cramèr
C-UV	Zonal wind, Meridional wind	Cramèr

Table 4.1: Trained generator models considered for the discussion and the comments of the results obtained.

Despite their increasing popularity, and their rapid developments concerning architectures and training frameworks, evaluation of GANs' performances is still a difficult task. Different set of measures has been introduced, but there is not yet a factor or an index which is able to take into account strength and limitations of the model and compare it with others (Borji (2018)). There are some measures which are suggested as ad-hoc factor to judge the visual quality of the images and the diversity of generated samples, like Inception score (IS) and Fréchet Inception (FID). However, they rely on an ImageNet-pretrained Inception network, which makes them inconsistent with other datasets like the one involved in this project (Shmelkov et al. (2018)). In fact, it is important to consider that GANs applications on problems of similar nature to the one considered in this project are really a few. In particular there are not example or studies involving ETCs for what is known, but meteorology and weather forecast more in general. This makes it even more complicated because measures of diversity or visual fidelity that have performed well for other kind of applications cannot be directly applied but they rather need to be investigated to assess their consistency and compatibility to the dataset. It is believed that the investigation of a particular performance metric is not needed and goes beyond the purpose of this research. It is believed that the image quality and the variety in the generated samples can still be judged from a qualitative point of view by looking at summary plots of both the generated and the original sample and similarity between them, but also by investigating the latent space representation of the network.

Following the analysis of the results it is progressively clear that the most interesting models to validate are those adopting multi-field representation. In fact, this kind of models are expected to have better performance in respect of single-field representation models because of the physical link between the atmospheric variables. It must be considered that the pressure gradient governs the winds, in particular it is strictly related with the wind speed through the geostrophic wind relation. Similarly, precipitation and pressure are related because rainfall systems are found along the cyclones' fronts. So the combination of different atmospheric variables during learning should improve the network representation thanks to their physical interdependence. This is not the only advantage of adopting multi-field representation, in fact by training the network with different atmospheric variable channels it becomes able to generate fields where the variables are all representing the same situations. To investigate better the links between the variables and if they improve the atmospheric representation, another multi-field model has been trained considering pressure, wind speed and precipitation. The model id is C-PWSR. However, the use of wind speed instead of wind components does not improve the representation with respect to the other models. Graphs the results for model C-PWSR are reported in Appendix 7.

To assess whether or not the model was able to learn the probability distribution of the original sample, the histograms of the original and generated fields are compared. Histograms are not ideal because they just take into account the pixel values independently of their location in the domain and independently of their belonging to the same image. But histograms are good proxies for judging the similarity between the two distributions and the diversity in the values assumed by the atmospheric variables in each pixels, therefore, they are necessary for a first validation of the generated samples and to judge their diversity. However histograms do not provide any information regarding the quality of the generated images.

To judge the visual fidelity of the generated images overall is complicated, in fact judging the fidelity of each GAN generated field by visual inspection of the general atmospheric structure and ETC features is an unfeasible task. Therefore, we proceeded by considering the average field of the sample for each atmospheric variable and comparing it to the original one may give good insight on the goodness of the images' quality. The average field is computed as the mean value in each pixel across all the fields belonging to the generated sample and the original sample. The average fields comparison can help to address some questions regarding the atmospheric structure across the two datasets. Are the regions of low/high pressure located at same longitude and latitude? Do they have equivalent extension? Is rainfall mainly occurring in the latitude band between 30°and 60°where all cyclones pass through? Similarly for higher values of wind speed, are they occurring in the region with higher track density?
To compare the models performance in representing the data structure three similarity/correlation index are computed considering the average fields of each atmospheric variables. They are the Pearson correlation coefficient, the Spearman's rank correlation coefficient and the Structural Similarity Index Measure (SSIM). The Pearson correlation coefficient is a statistical measure to quantify the linear correlation between two variables or datasets, it ranges between -1 and 1. An absolute value of 1 indicates a linear relationship between the two samples, while a value of 0 implies that there is no linear correlation (Benesty et al. (2009)). The Spearman's rank correlation coefficient is a statistical measure to assess monotonic relationship between two variables or datasets, it also ranges between -1 and 1. When each of the samples is a perfect monotone function of the other, Spearman correlation is equal to 1 in absolute value (Sedgwick (2014)). The SSIM is a method to measure the similarity between two images by estimating the absolute errors, differently from mean squared error and peak signal-to-noise ration. The measure considers the structural information of the images, so the idea that pixels in similar locations have strong inter-dependencies. SSIM also ranges between -1 and 1, an absolute value of 1 indicates perfect structural similarity and happens in the case of two identical sets of data. A value of 0 indicates no structural similarity (Zhou Wang et al. (2004)). The average fields generated by the different models highlight a better results for the model adopting the Cramèr distance as loss function of the

network.

There is an inconvenience in considering the average fields, even though the generated and original ones may look alike it does not mean that the values in the pixels of generated fields are occurring with a similar relationship as they are occurring in the original sample. To explore better the visual fidelity of the images 4 specific cyclones are considered as case studies for a field to field comparison with the most similar images from the generated samples, to see if the model is actually able to reproduce similar situation to the one characterizing these cyclones. The storms considered in the analysis are listed in Table 4.2.

Cyclone ID	**Start date**	**End date**	**# of snapshots**
257	14-10-1987 12:00	16-10-1987 06:00	42
441	15-10-1993 11:00	16-10-1993 11:00	25
643	24-12-1999 06:00	27-12-1999 18:00	84
667	29-10-2000 17:00	01-11-2000 00:00	55

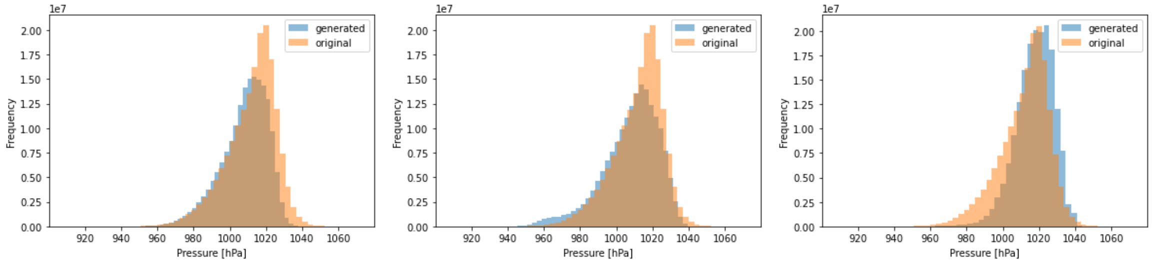
Table 4.2: ETCs considered in the field to field comparison and in the latent space interpolation analysis.

Together with the field to field comparison the latent space is investigated to try to understand how small variations in the random vectors influence the fields generated by the model and if the connection between latent space and images generation is not random. The analysis proceeds according to the method applied in Radford et al. (2015) and it works as follows. Considering the initial and the final frames of each cyclone track, the most similar fields are selected from a generated sample according to the SSIM. For simplicity, let A be the generated field most similar to the initial cyclone snapshot, and B the most similar to the final snapshot. A linear interpolation is performed from the latent space vector generating image A to the vector generating image B. After this procedure a synthetic track is generated from the interpolated vectors, which are the same amount as the original snapshots in the cyclone track. The frames of this new generated track are compared to the ones of the original track by plotting the SSIM evolution in time. To better assess the similarity between the original track and the synthetic track, the average SSIM between the original frames and a sample of 1'000 generated images is also plotted for comparison.
This comparison procedure helps to understand how well the models perform in respect of a random guess (the average SSIM), to analyze better the visual fidelity of the generated images and to investigate if the model has learned the atmospheric structure underlying the fields. By comparing the snapshots in the generated track to the ones in the original track the quality of the images can be judged considering the structure and the trends resemblance between correspondent fields. The network learning can be evaluated by looking at the similarity in the snapshots evolution, in fact if the snapshots in the two tracks present similar features it suggests that the model was able to learn a meaningful mapping of the latent space. Small changes in the vectors are matched by the generated images respecting of the physical structure and evolution of these events. At the same time it is very difficult that the evolution in the interpolated track matches the evolution in the original track because of how the former is produced and because of the data constituting the training set. The frames of the synthetic tracks are obtained from interpolation and the network has trained on ETCs fields generation and not tracks generation.

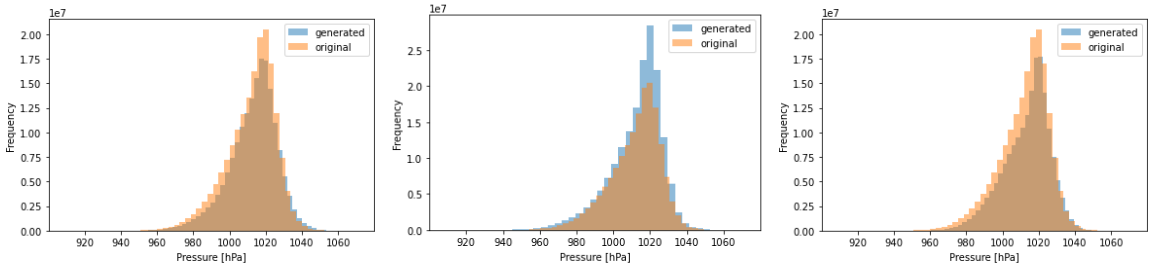
The number of images contained in each generated set adopted for sample distribution comparison, average fields comparison and selection of most similar initial and final snapshots is the same as the original training set, so 43'313 in each generated dataset.

4.1. Sample distribution comparison

The comparison of the histograms of the original and generated dataset gives an insight on the ability of the network to learn the distribution, in particular to understand if the level of variety in the generated sample is high almost as or as much as in the original sample. The histograms are plotted for each atmospheric variable and for each trained model, see Figure 4.1, 4.2, 4.3 and 4.4. All the generated samples of fields for every variable have a unimodal distribution like the original dataset, in some cases the peak in frequencies are located at very similar values, while in a few other the shape of the distribution is equivalent but the range is clearly shifted.
In general, by looking at the histograms, it is evident that the atmospheric variables which is better repre-



(a) W-P (b) W-PUV (c) W-PR



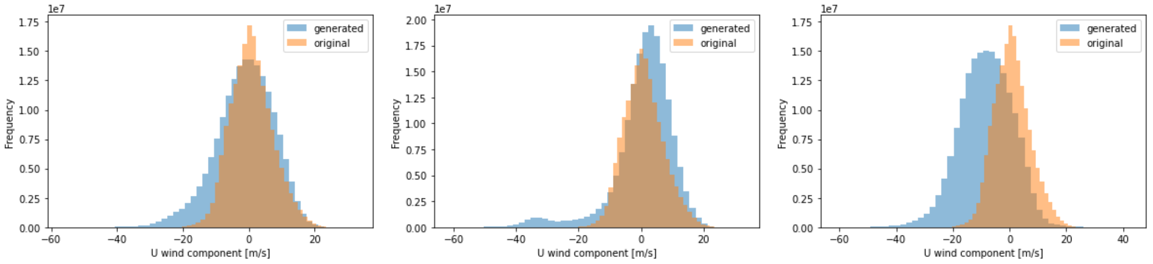
(d) C-P (e) C-PUV (f) C-PR

Figure 4.1: Histograms comparison for pressure between the original sample and the generated sample for the six trained models.

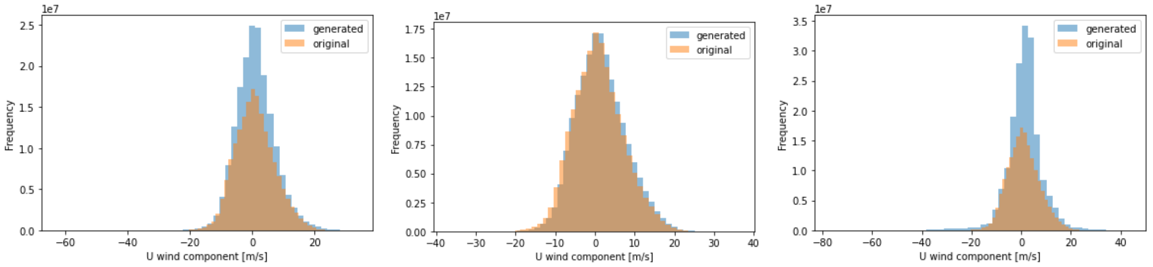
sented across the different models is the pressure (Fig. 4.1). In fact the range of values is usually very similar between generated and original histograms, as well as the shape. It is also clear from the graphs that the distributions generated by the model adopting the Cramèr distance have more similar properties to the original distribution. For example, consider the histograms of pressure and wind components for model W-PUV (Fig. 4.1b, 4.2b, 4.3b), there is a general trend in the histograms of the generated sample of higher frequencies for smaller values which results in bigger left tail for the distribution. This trend is not occurring in the generated sample of model C-PUV (Fig. 4.1e, 4.2e, 4.3e), where instead the histograms look very similar with small differences in the peak frequency.

Considering single-field representation and multi-fields representation it is noticeable that the utilization of more channels for a combined learning is not very helpful to improve the values distribution in the generated samples. There are some cases of multi-fields representation model where the histograms of the atmospheric variables are usually more dissimilar between each other in comparison with those of single-field representation. The most significant examples are the histograms computed from the generated samples of models C-PR (Fig. 4.1f, 4.4c), W-UV (Fig. 4.2c, 4.3c) and C-UV (Fig. 4.2f, 4.3f) for all the atmospheric variables. For pressure and rainfall it is just a matter of discrepancies in frequencies of some values, whereas shape and range of the histograms still look similar. For the wind components the worsening in the histograms similarity is also tangible in the shape, in the range and in the mode location. Especially for what concerns the plots for W-UV it seems like the information of both wind components together decrease the network understanding of the data representation causing a lot of difference between the original histogram and the generated one. This behaviour of the network is unexpected considering that the wind components are strictly correlated since they measure the same atmospheric entity. The additional information regarding pressure increase the quality of the generated sample representation for both wind components, guaranteeing a comparable diversity to the original dataset, see Figure 4.2b, 4.3b, 4.2e and 4.3e.

For what concern rainfall it is clear that the results obtained from the model trained with the Wasserstein distance (Fig. 4.4a) are definitely worse than the results obtained by the Cramèr models. The rainfall generated histogram of W-PR shows considerable frequencies for different negative values of rainfall, which is obviously wrong and not physically possible. Negative values generated by the model are likely to be a consequence of the characteristics of rainfall in the original sample. In fact, because of the domain extension and the characteristics of rainfall events is more frequent to have values equal or very close to zero than larger positive numbers. For this reason and because the network is not aware of natural constraints of precipitation like the impossibility of having the negative values, the model learns the possibility to also have negative values in the variable representation. Negative values are also present in the rainfall histograms for C-P and C-PR model (Fig. 4.4b, 4.4c) but they are very close to 0, which suggests that the model has better learned the occurrences in values for rainfall. C-PR model is the one generating a sample with higher frequency in nil values, closer to



(a) W-U (b) W-PUV (c) W-UV



(d) C-U (e) C-PUV (f) C-UV

Figure 4.2: Histograms comparison for the zonal wind component between the original sample and the generated sample for the six trained models.

the one in the original sample.

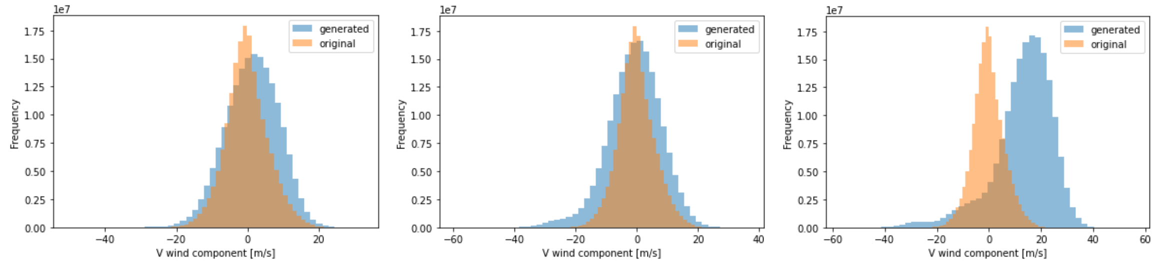
4.2. Average fields comparison

The atmospheric variable average fields are computed both for the original sample and the generated samples as the average in each pixel of the domain across the all dataset. Even though the average fields do not contain information regarding the characteristics of all fields belonging to the sample, they show the average structure of the atmosphere which is useful to understand if in the generated samples there are similar patterns to the original sample. As first analysis the similarity/correlation between the average generated fields and the average original fields for each variables in every model is computed according to the Pearson correlation coefficient, the Spearman correlation coefficient and the SSIM. Figure 4.5, 4.6, 4.7 and 4.8 show the values of the indices for pressure, wind components and precipitation. The variables which presents higher values of similarity/correlation between average fields is pressure, independently of the number of channels in the representation and the loss function considered for training.

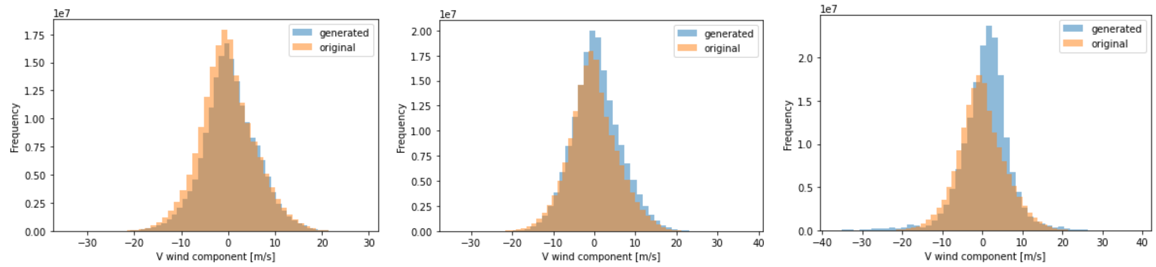
In general, is evident that for the models adopting the Cramèr distance as loss function the similarity between average fields is higher, independently from the variable that is considered. This behaviour is clear for the wind components and rainfall while for what concern pressure there are good values of similarity also considering model W-PR.

Considering the single-field and the multi-fields representation the improvement in similarity between average fields is not always related with the increase of atmospheric variables adopted for training. If the Wasserstein models are considered there is a significant increase in the indices for multi-fields representation only for pressure, while for zonal and meridional wind component the situation is more complex. The SSIM slightly decreases for both wind components in the multi-fields representation as well as Pearson and Spearman coefficients for the zonal wind. While for the meridional wind the correlation coefficients marginally increase. For what concern Cramèr models the only case where the value of the indices increases in the multi-fields representation is for the meridional wind component in model C-PUV. For the other variables there are very equivalent values of correlation for both single-field and multi-fields representation, the only exception is model C-UV where the similarity in the average fields for both wind components is very low compared to the other Cramèr models. The values of the indices for models W-UV and C-UV suggest a poor performance from the models considering the two wind components in their representation, the information regarding both components seem to bring contrast in the network learning. This behaviour is not present when also pressure is considered in the model learning, as in model W-PUV and C-PUV.

Following the bar charts reporting the correlation/similarity indices the plots of the generated average fields

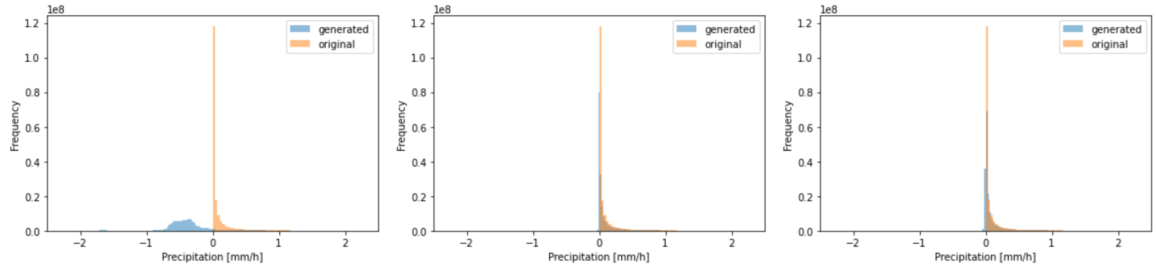


(a) W-V (b) W-PUV (c) W-UV



(d) C-V (e) C-PUV (f) C-UV

Figure 4.3: Histograms comparison for the meridional wind component between the original sample and the generated sample for the six trained models.



(a) W-PR (b) C-R (c) C-PR

Figure 4.4: Histograms comparison for precipitation between the original sample and the generated sample for the six trained models.

for the most significant models are reported in comparison to the original sample of each variable (Fig. 4.9, 4.10, 4.11 and 4.12.

Considering the plots of the average fields for pressure in Figure 4.9 it is interesting to observe that all the models represent the regions of high and low pressure similarly to the original sample average field. In fact, in the central region of the domain the lowest value of pressure are occurring, it respects the truth that most of the cyclone if not all the cyclone tracks are moving across the square with borders 30°N, 60°N, 40°E and 10 °E as reported in Figure 1.12. With respect of a qualitative human eye judgement it can be asserted that the most similar average field is the one obtained by the generated sample of model C-PUV (Fig. 4.9d). The average field produced by model C-PUV is better at representing the extension of the regions of low and high pressure with a similar range of values even if they are generally higher from the ones in the original field. Also the gradient patterns are very similar to those of the original average fields, looking much smoother and continuous with a very close representation of the shift of values in the band comprehended between 0°and 30°N. For what concerns the average field of W-PR model (Fig. 4.9b), the values in each pixel are bigger than the corresponding in the original average fields, in fact even if the region of low and high pressure are truthfully represented the averages in the pixels are really shifted to higher values. This behaviour compromises the representation of the 0°-30°N band which looks like a high pressure region when in the original fields the pressure range is at lower values. The average field of W-PUV model (Fig. 4.9c) present a similar pattern to the original average field considering the location and extension of the central low pressure region. However, it lacks in representing correctly the rest of the domain, the 60°N-90°N band of high pressure contains very high values, the 0°-30°N band has a very irregular gradient and the values are oscillating from very low to very

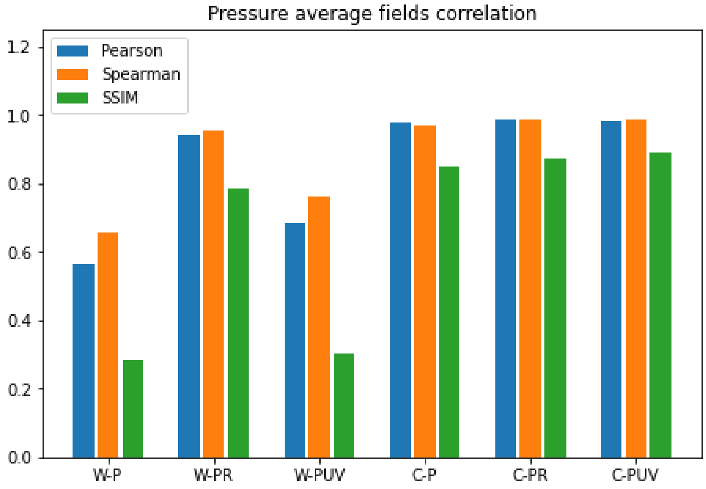


Figure 4.5: Correlation between the original pressure average field and the average fields generated by the different models. The correlation is expressed according to the Pearson correlation coefficient, the Spearman correlation coefficient and the SSIM.

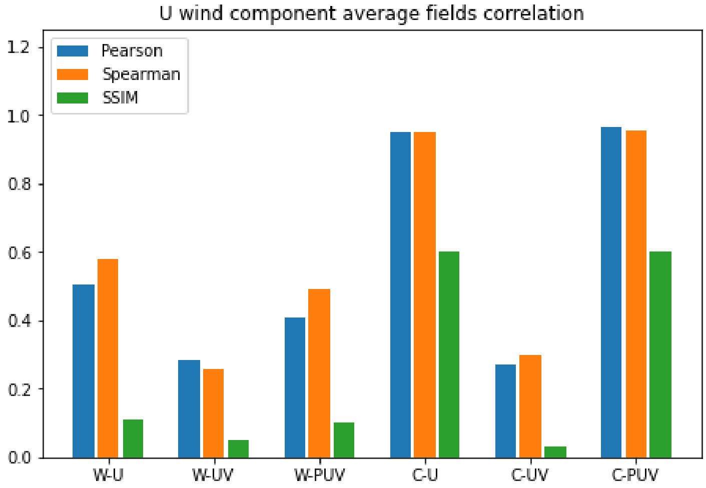


Figure 4.6: Correlation between the original zonal wind component average field and the average fields generated by the different models. The correlation is expressed according to the Pearson correlation coefficient, the Spearman correlation coefficient and the SSIM.

high instead of being varying smoothly and in a smaller range. Not only high and low latitudes are poorly represented but also there is a lot of irregularity in the average field. In fact in different regions of the domain are present vertical or horizontal stripes of values very dissimilar from the ones surrounding them, in particular at the boundaries of the domain. This stripes of different values influence particularly the structure of the average field, which generally has a very irregular gradients in respect of the other average fields and it looks much more blurry and artificial.

For what concerns the average fields plots of the wind components is immediately evident that the most similar are the ones computed from the generate samples of model C-PUV (Fig. 4.10d, 4.11d), as it was for the pressure. The region of high values, located in the central part of the domain, have similar extension and shape for both wind components, with some discrepancy in the magnitude of the values. This is important because it suggests that the network has learned that the central regions of the domain are those where higher wind speed are occurring due to the more frequent passage of storms. The winds gradients are well represented, following similar trends to those appearing in the original average field. The average fields plots of model C-UV confirm the poor performance of a model considering only the two wind components, in fact both Figure 4.10c and 4.11c have nothing in common with the corresponding original plots, looking very chaotic and blurry by showing similar behaviours to Figure 4.9c. Blurriness and artifacts in the field structure

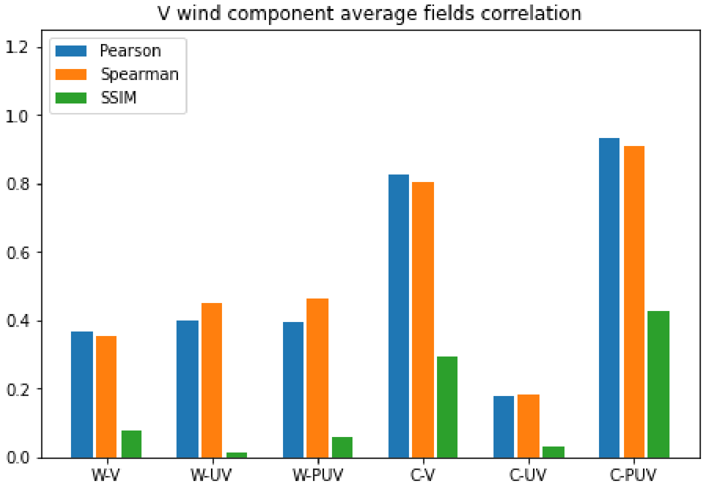


Figure 4.7: Correlation between the original meridional wind component average field and the average fields generated by the different models. The correlation is expressed according to the Pearson correlation coefficient, the Spearman correlation coefficient and the SSIM.

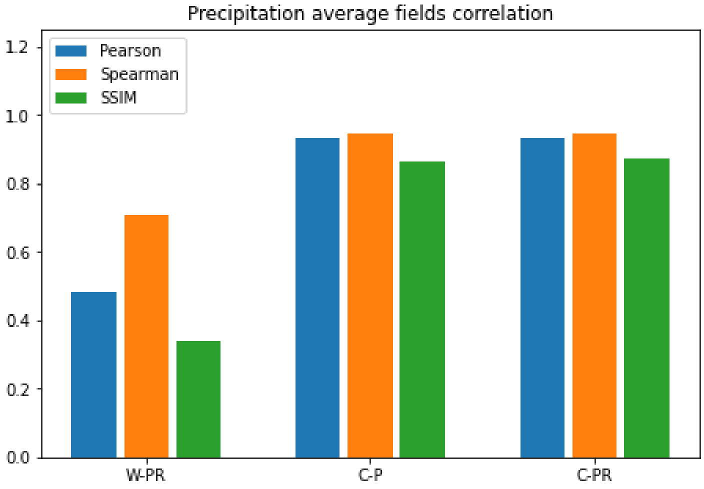
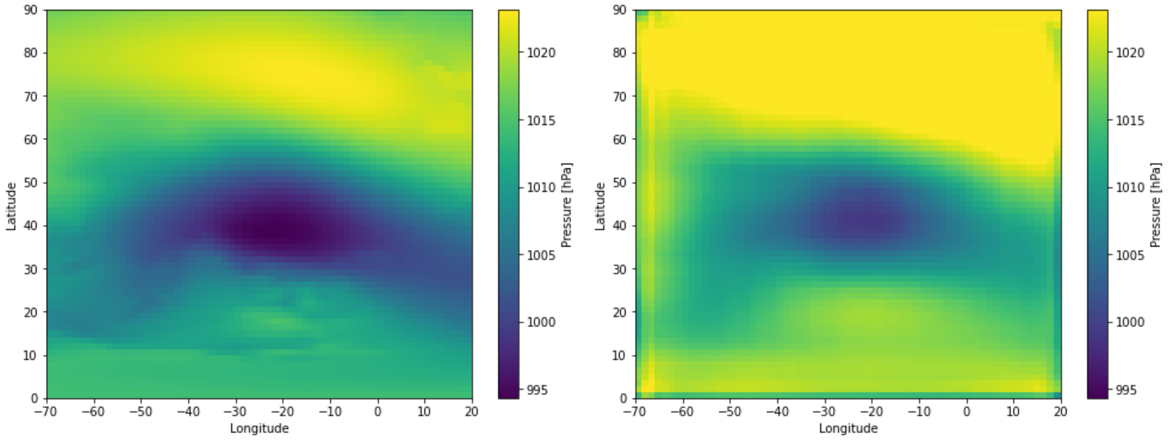


Figure 4.8: Correlation between the original precipitation average field and the average fields generated by the different models. The correlation is expressed according to the Pearson correlation coefficient, the Spearman correlation coefficient and the SSIM.

are also evident in the average fields of model W-PUV (Fig. 4.10b, 4.11b), however in this case in some regions of the domain the values seems to follow a similar structure to the one of the original fields (Fig. 4.10a, 4.11a). In fact, for the meridional wind component the model represents with similar extension and magnitude of the values the region located between 10°N-40°N and 70°E-20°E. Similarly for the zonal wind component the region of high values with sides 40°N, 70°N, 50°E and 10°E has an equivalent extension. However the rest of the domain is poorly represented for both variables.

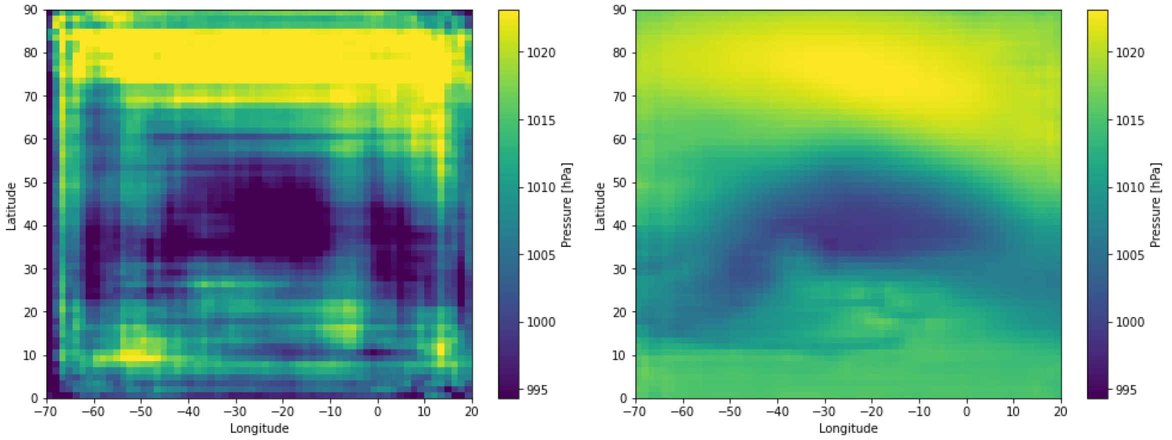
As for the other variables also for precipitation the average fields computed from the generated sample of the model trained with the Cramèr distance looks much more similar to the original average fields than the one obtained from the sample of the Wassertein model. Figure 4.12c shows that model C-PR correctly represent the structure underlying the average rainfall field, in fact the region of higher values have similar shape and extension both in the generated field and in the original field with similar range of values. However the gradient in the generated field is not smooth as in the original one. The generated average fields of model W-PR (Fig. 4.12b) is completely different from the original average field, especially because almost all the value in the domain are negative. The other interesting feature to observe are the very high positive values at the left boundary of the domain an in the top right corner, they are probably caused by the same kind of behaviour typical of the models trained with Wassertein distance.

If the histograms comparison between original samples and generated samples suggests a better represen-



(a) Original

(b) W-PR



(c) W-PUV

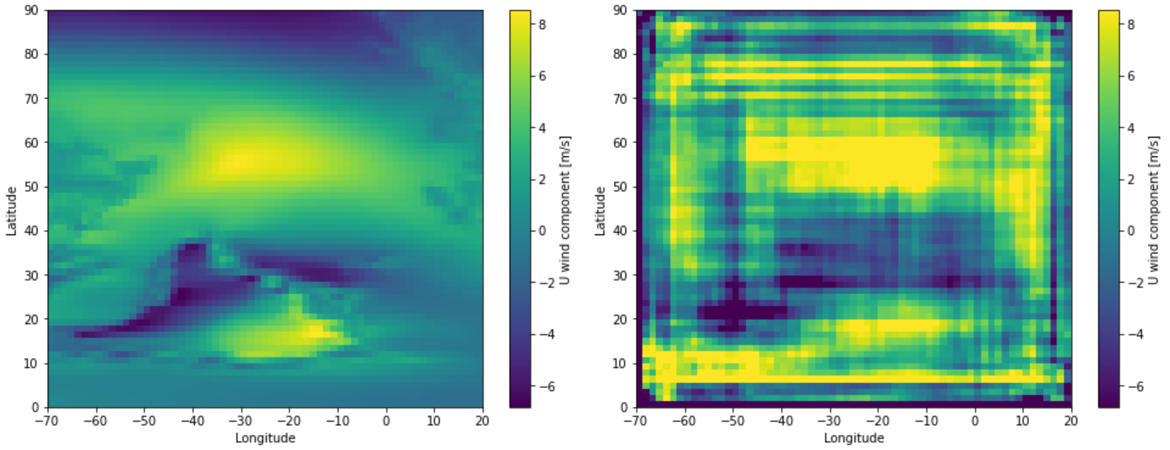
(d) C-PUV

Figure 4.9: Average fields of pressure for the original sample and for the generated samples of significant trained models. In this case the average fields of W-P, C-P and C-PR are not considered because the former is equivalent to W-PUV and the latter to C-PUV.

tation of the data structure by the model adopting the Cramèr distance as loss function, the plots of the average fields confirm this intuition. The average fields of the samples generated with the Cramèr models are much more similar to the original average fields, they correctly represent the extension, the shape and the magnitude of the different regions in the domain. Also the gradients in this field looks more smooth and homogeneous, matching those of the original fields with some small discrepancies especially for precipitation. Generally, pressure seems to be the variable of which the network is able to learn the structure better, in fact its additional information causes a much better representation for the wind components.

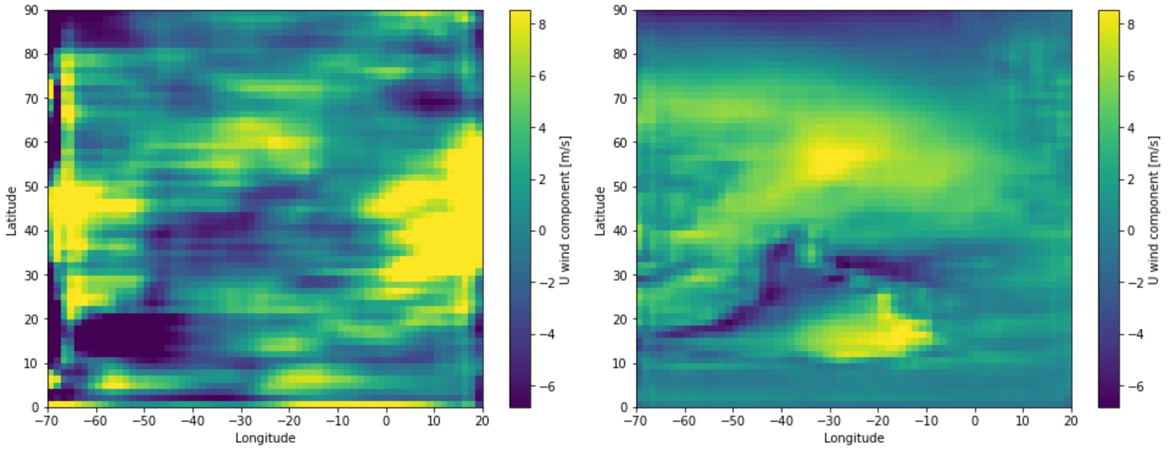
The average fields of the samples generated with the Wasserstein models usually comprehend some characteristics of the original fields' structure but they look more chaotic, with an irregular gradient that is usually evident because of considerable variation in values between close grid points. These generated images are characterized by random horizontal and vertical stripes of very different values from those in the pixel surrounding them. This kind of behaviour makes the average field look more artificial and it occurs in all the images of the generated samples examined. It is believed that the stripes behaviour is a consequence of the network learning using the Wasserstein distance. In fact, the stripes are probably artifacts created by the network to better match the distribution of the atmospheric variables across the fields. The GAN may generate these lines of extreme values to balance the data distribution structure doing training. This behaviour is here shown for the average fields of the generated sample but its characteristics are expandable to almost all WGAN models since it is so clear in different region of the average field and of the generated images. However, there is an interesting exception in model W-PR, in fact it does not show any of the characteristics common to the other Wasserstein models. Nevertheless its outcome is still inferior to the results obtained with C-PR.

From the average fields comparison it emerges that for the Cramèr models there is a small improvement in



(a) Original

(b) W-PUV



(c) C-UV

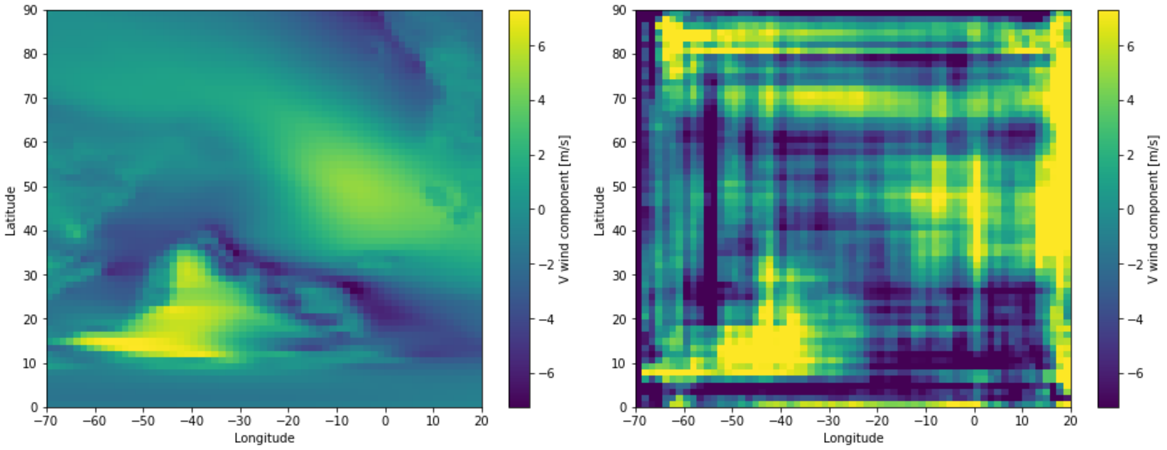
(d) C-PUV

Figure 4.10: Average fields of zonal wind component for the original sample and for the generated samples of significant trained models. In this case the average fields of W-U and C-U are not considered because the former is equivalent to W-PUV and the latter to C-PUV. To show the poor representation of combined wind fields only C-UV is considered.

the representation learned by the network if multi-field learning is considered instead of single-field, with the exception of C-UV that is instead worse of the wind components single-field representation. For the Wasserstein models similar conclusion are valid but the case of W-PR might be highlighted, in fact it seems that the additional information regarding rainfall is helpful for a better learning of the pressure fields structure. This behaviour makes sense considering the physical connection between the two variables.

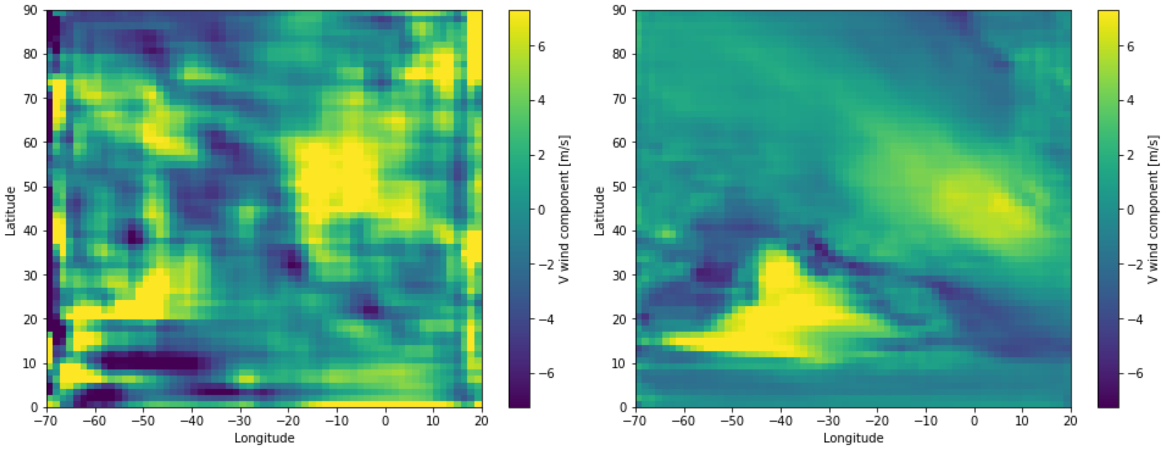
4.3. Latent space interpolation and tracks similarity

From the sample distribution and average fields comparison it has emerged that the best results and performance are obtained with the multi-field Cramèr models. For the investigation of the latent space and for the study of the tracks similarity only the models C-PR and C-PUV will be considered.
The level of similarity between the original track and the synthetic track is evaluated by comparing the SSIM evolution of corresponding snapshots to the evolution of the average SSIM between the original frames and a sample of 1'000 generated images. Because the cyclone have all different life period, instead of considering the evolution in absolute terms, the relative evolution in percentage is taken into account, so that the comparison between cyclones results easier. The synthetic track is generated thanks to the linear interpolation between the vectors of the latent space generating the most similar images to the initial and final snapshots of the storm. The field to field comparison between the interpolated track and the generated track shows how well our models perform compared to a random guess. If the training of the GAN has been successful a meaningful link between the latent space and the images generation is expected. This means that the images



(a) Original

(b) W-PUV



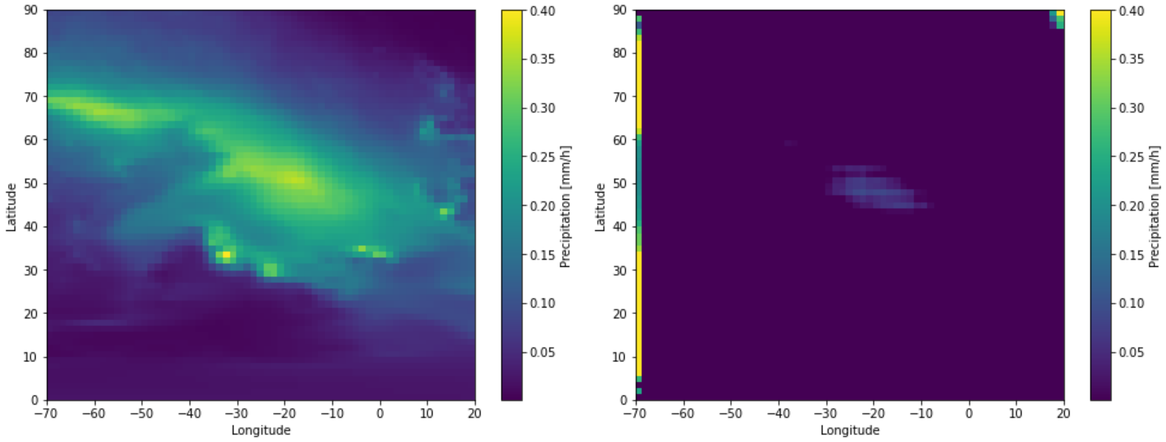
(c) C-UV

(d) C-PUV

Figure 4.11: Average fields of meridional wind component for the original sample and for the generated samples of significant trained models. In this case the average fields of W-V and C-V are not considered because the former is equivalent to W-PUV and the latter to C-PUV. To show the poor representation of combined wind fields only C-UV is considered.

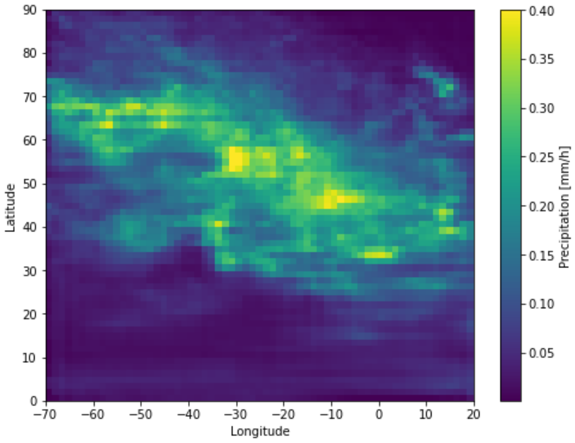
obtained from the interpolated vectors should show similar features to the frames of the original tracks and have higher similarity in respect of the average SSIM.
The analysis of the interpolated track helps to judge the atmospheric structure underlying the generated fields and to investigate the relationship between latent space and images generation. Are small variations in the latent space matched with small variations in the generated images by the network? Do these variations have physical meanings? The comparison between interpolated and original track addresses these questions but it cannot be considered to comment the cyclone evolution in the track. It is important to remember that the network has been trained with cyclone fields, not with cyclone tracks. In fact it is impossible for the network to learn the evolution of a cyclone without having seen one, especially with the mediocre amount of images used for training.

The results of the SSIMs evolution comparison are promising, in fact for both models considered and for almost all cyclones the similarity between original snapshots and interpolated snapshots is always considerably higher than the average with the 1’000 images sample (Fig 4.13 and 4.14), not only for the initial and final snapshots but also for the interpolated images composing the synthetic track. The range of values for the 2 SSIMs is generally higher for model C-PR, that is because the similarity indexes are computed considering rainfall values. In fact the high frequency of nil values in the precipitation distribution causes higher similarity between the original and the interpolated track. There is a common trend in the SSIM evolution to some of the interpolated tracks which first decreases and then grows again drawing a mild convex line (Fig. 4.13a, 4.13d, 4.14a and 4.14d). This behaviour in the similarity evolution was excepted because the ini-



(a) Original

(b) W-PR



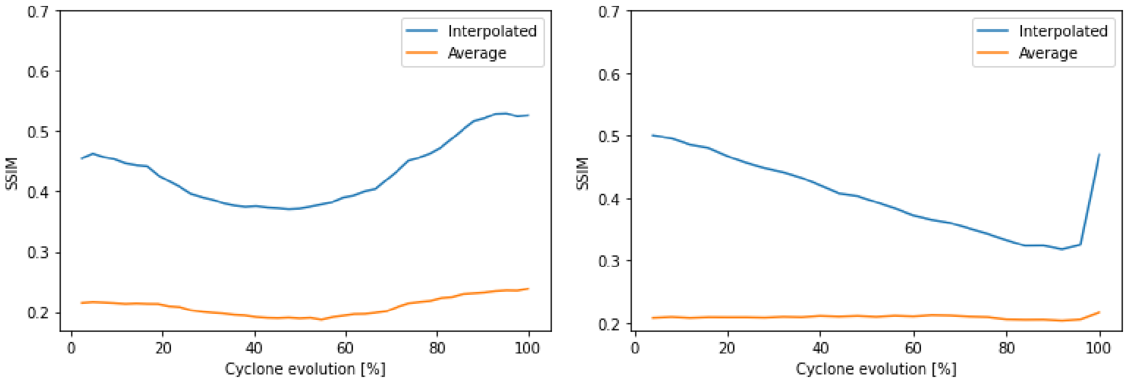
(c) C-PR

Figure 4.12: Average fields of precipitation for the original sample and for the generated samples of significant trained models. In this case the average field of C-R is not considered because is equivalent to C-PR.

tial and final snapshots are specifically chosen considering the most similar images from a generated sample of same dimension to the original. The shortest track, cyclone 441, shows a constant decrease in the SSIM evolution unless increasing rapidly when it arrives at the final snapshot. The decrease is steeper for model C-PR, while in model C-PUV is almost constant in the first and last third of the cyclone evolution (Fig. 4.13b and 4.14b). The SSIM evolution of track 643 decreases irregularly from initial to final frame (Fig. 4.13c and 4.14c). Is interesting to highlight that for almost all the interpolated tracks in both models the SSIM evolution has a maximum decrease within a range that is at least 2 times smaller than the minimum distance from the other line, the only exception is cyclone 441 in model C-PR (Fig. 4.13b). These results is comforting because it is interpretable as a good level of relative similarity between the interpolated snapshots and the original ones.

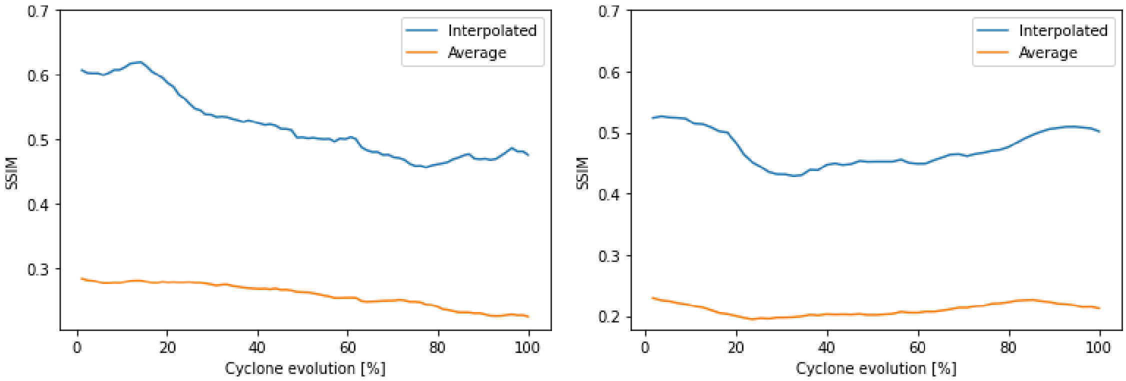
To have a better idea of how well the different interpolated tracks are represented with respect of the original cyclones the visual plot of each field is considered. Because of the long sequences of snapshots for each track GIFs showing their evolution are attached to this document in the repository. The qualitative assessment of the similarity between atmospheric structures in the snapshots of interpolated and original tracks enlightens some interesting properties of the generated fields.

For what concern pressure the low region corresponding to the cyclone centre is generally well represented for all cyclones. Interpolated track 441, 643 and 667 obtained from model C-PR have some differences in extension, size and magnitude of the lows along with the cyclone evolution. In the interpolated tracks obtained from model C-PUV the lows looks more similar but in track 257 and 667 the values in the cyclone center region are about 2 hPa larger on average. The high pressure region characterising the 0°-30°latitude band of the domain is usually represented with bigger values by both models, usually around 2 to 3 hPa, except for



(a) Cyclone 257

(b) Cyclone 441



(c) Cyclone 643

(d) Cyclone 667

Figure 4.13: SSIM evolution between original snapshots of the cyclones tracks and interpolated generated images using Model C-PR. The most similar generated images to the initial and final frame are coming from a dataset that has the same dimension of the original one. The SSIM evolution is compared with the average SSIM evolution between the original snapshots and frames from a sample of 1'000 images.

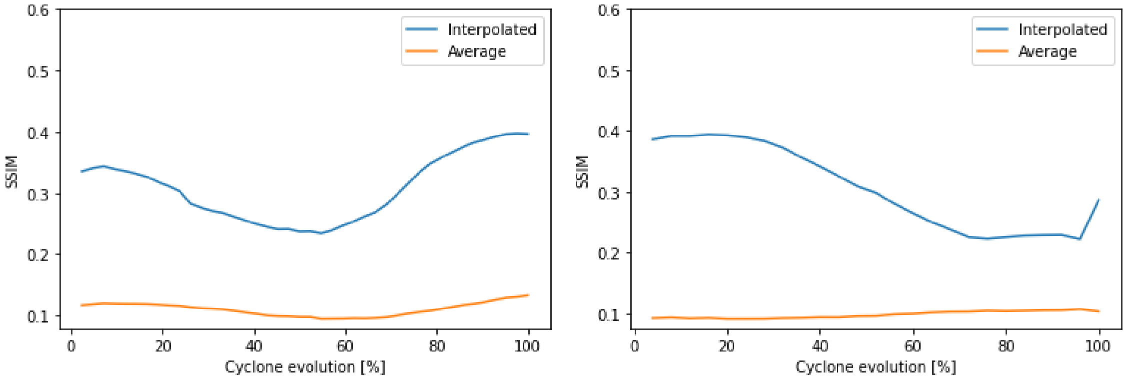
cyclone 667 where the pressure field has values higher than the average across all domain. The high pressure region characterising the 50°N-90°N latitude band of the domain is usually better represented and similar to the original, with the exception of higher values for the interpolated track of cyclone 257 for both models, at certain time-steps the differences are even around 10 hPa. In general the pressure gradient is well represented and the region of transition from low to high pressure have similar values to the original snapshots, differing sometimes for extension.

Wind representation is generally good across all interpolated tracks, both wind speed and direction in the cyclone region look similar to the original tracks. The most problematic part is the southern part of the domain were the wind speed values in the generated track are higher, sometimes even of 10 m/s. Also the direction is poorly represented in this region with differences going up to 100°from N.

For what concerns precipitation, the extension of rainfall systems comprehended in the cyclone is usually similar between the two tracks, especially in the beginning. The interpolated snapshots cannot represent clearly the evolution of shape, location and values of the more intense rainfall system linked to the storms. The evolution of the latter in the generated track does not follow the moving pattern of the cyclones going from west to east across the domain.

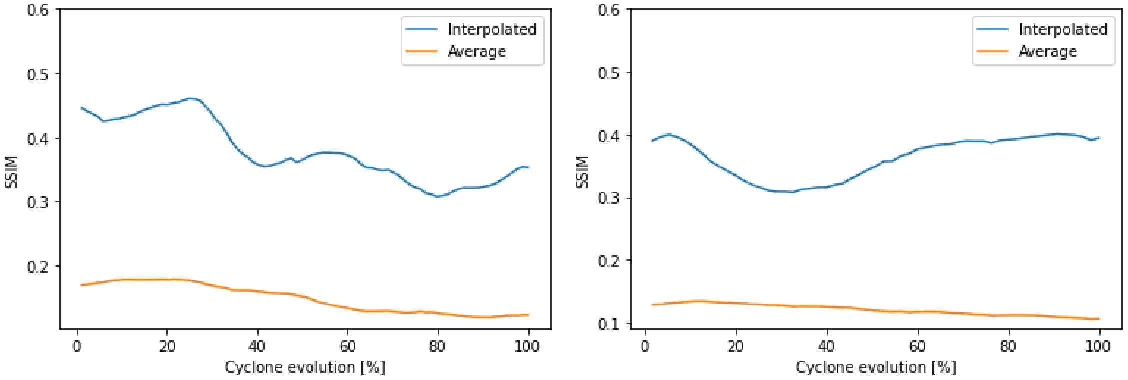
The main discrepancy present across all tracks concerns the fields evolution, and is mainly tangible in pressure and rainfall fields. In the original track the images change one after the other suggest a movement of air masses, especially for the cyclone center where the spinning behaviour of the storm is evident, in the interpolated track this is not felt. However, the lack of these feature in the interpolated tracks is more than comprehensible considering that only the initial and final frame are generated taking into account the similarity with the original track, while the rest of the snapshots are determined with a linear interpolation.

The latent space interpolation between vectors generating the most similar images to the initial and final



(a) Cyclone 257

(b) Cyclone 441



(c) Cyclone 643

(d) Cyclone 667

Figure 4.14: SSIM evolution between original snapshots of the cyclones tracks and interpolated generated images using Model C-PUV. The most similar generated images to the initial and final frame are coming from a dataset that has the same dimension of the original one. The SSIM evolution is compared with the average SSIM evolution between the original snapshots and frames from a sample of 1’000 images.

snapshots of the cyclone is helpful to also draw some conclusion on the latent space interpretation by the network. The network relationship with the latent space and the images generation is investigated thanks to the interpolated tracks. With the help of the similarity comparison it can be safely affirmed that the representation of the network of the ETCs fields has a structure which respects a lot of features of the original data. In fact, the images produced by the interpolation present a high level of relative similarity compared to a random guess, the average similarity across 1’000 random images. Also, the couples of corresponding snapshots compared one by one show that generally the atmospheric structure in the interpolated frames looks like the original with most of discrepancies in the magnitude of the values. So by moving in the latent space from a vector to another vector generating know situations, the images generated by the network have several characteristics in common to the intermediate situations in the original sample. The visual plots of the interpolated tracks show that the system is able to match small variations in the latent space to small variations in the images. These results are promising because they show that the images generated by the network change accordingly to the atmospheric structure of the ETCs fields. Which means that the generation of images is not randomly connected to the latent space. However a more specific and extensive research regarding the latent space is needed to understand better how values in the latent space vectors influence the field generation and if there are any relationships between regions of the latent space and commons situations in the generated images. It would be also interesting to understand better the relationship between values at specific position of the vectors and characteristics of the ETCs fields. To address such problems a technique like the inversion of the Generator must be considered (Creswell and Bharath (2018)).

5

Conclusions

The purpose of the project is to address the possibility for a GAN to learn the distribution of ETCs fields and generate new samples that present similar atmospheric structure to the original one. To deal with the known GANs problems of instability during training and generation of high resolution images a specific architecture has been selected, the Progressive Growing GAN. The method works by progressively growing the resolution of the convolutional layers both in the generator and the discriminator together with the progress of the network learning. This learning framework allows the network to first learn the large scale features in the images while training at lower resolution, afterward the learning focus is shifted on small scale features at higher resolution. The progressive growing of the resolution in the training images guarantees that the network does not have to learn features at different scales simultaneously. The PGGAN framework has been selected also because it easily allows to implement the training also at higher resolution for future developing. In fact, for this project the images have been down-sampled to a 64x64 resolution to guarantee a more stable training.
The PGGAN has been trained on the fields of pressure, wind and precipitation of the storms extracted from 40 years of ERA5 reanalysis data. In particular they are the mean sea level pressure, the zonal wind component at 10 m, the meridional wind component at 10 m and the rainfall rate. To train the model the domain selected is a squared region in the North Atlantic, the squared shape simplifies the filter operation in the convolutional layers and the architecture of the network.
The loss function considered for the network are the Wasserstein distance and the Cramèr distance, the former has been historically adopted in GANs training because it helps resolving the probelm of mode collapse and simplifies the procedure of training simultaneously two networks (Arjovsky et al. (2017), Weng (2019)). The latter has been selected after the evidences of a better performance by the network that are adopting it as loss function for training on images (Bellemare et al. (2017)). These evidences are confirmed by this project, in fact the results show a much better performance in the ETCs' fields representation from the model trained with the Cramèr function, especially for what concern the atmospheric structure underlying the data. WGAN generates more chaotic and blurry fields, usually presenting vertical or horizontal stripes of picture with very different values form the surrounding that do not really look natural. These stripes seems like artifacts created by the network to match the original dataset distribution.

The results obtained with the training of the CGAN are promising. The values distribution of the generated samples matches the distribution of the original sample with very small differences for all atmospheric variables. The histograms of the generated samples have similar range, shape and frequency peak to the original ones and multi-field representation does not significantly improve the features of the generated histograms (Fig. 4.1d, 4.1e, 4.1f, 4.2d, 4.2e, 4.3d, 4.3e, 4.4b and 4.4c). However, in the generated sample distribution of rainfall the network represents the possibility of the variable to assume negative values, which is unrealistic. The negative values are consider by the network probably because the original rainfall distribution is very close to 0 and the peak in frequency is corresponding to nil values. Luckily the magnitude of the negative values is very low.
The average fields comparison confirms the goodness of the images generated by the CGAN, in fact for all the variables the average features of the atmospheric structure are captured, with usually some differences in pixel values. Shape, extension and location of the different regions in the domain are similar between generated average fields and original (Fig. 4.9d, 4.10d, 4.11d and 4.12c). A significant example is the generated

pressure field which represents the areas of low, intermediate and high pressure correctly considering shape, extension, location and shift of values from between regions but the pixel values are higher on average of those in the original field.
The analysis on the interpolated tracks suggest that the network not only has been able to learn the values distribution of the atmospheric variables related to ETCs, but also the atmospheric structure underlying this kind of events. The images generated from the interpolated vectors in the latent space show similar features to those of the original track. The network proves that if the values generated in the vectors are shifted accordingly to known conditions in the atmosphere the correspondent images generated respect the structure of those conditions, meaning that the latent space mapping of the network has a physical structure.
Even if the methods applied to judge the quality of the generated images and the learning of the network show good results overall, it must be considered that they are not strong enough to completely validate the results obtained. The non-existence of general parameters to judge the quality of the images generated by a GAN and the absence of summary indexes which can tell if an image might belong to a cyclone track cause the results comment to be a more subjective analysis relying on a qualitative assessment of the images generated. Even if the methods may not seem strong enough for validation, they are consistent to understand the network ability, eventual improvements needed to the framework and helpful to answer the research questions guiding the thesis. For a future prospective a set of metrics to validate the generated images must be considered, it could consider aspects used in the identification and tracking of cyclone track or more general measures used in GANs validation (Borji (2018)).

Considering the results integrity and the network training behaviour the research question are addressed.
Is it possible to generate meaningful ETCs atmospheric variable fields using a PGGAN?
Yes, it is possible but there are a few things which needs to be improved in the representation. In fact, it is evident from the results that the network is considering possible negative values for precipitation, which is not realistic. Also, the intense rainfall regions evolution in the cyclones is not well represented. On average, across all models for all variables, the generated images present higher values than the original ones. There is some inconsistency in the representation of the southern region of the domain, which might be cause by the different atmospheric regime at tropical latitudes. A possible way to address the problem of negative values in rainfall distribution is applying a different transformation to the training set. In the project it has been normalized considering the maximum an minimum values historically occurred in the North Atlantic. But a different transformation could ensure data quality and accuracy of predictions. Considering the shape of the rainfall distribution the logarithmic transformation might positively affect the performance of the learning algorithm. Log transform is recommended for skewed data because it usually has the effect of spreading out clusters of data and bringing together spread-out data. By applying this transformation at the training set the network might learn a more correct representation of the rainfall phenomenon with considering negative values.
Can the combination of different variable fields improve the PGGAN learning of the atmospheric structure during ETCs?
The combination of different variables to perform a multi-channel training of the network does not necessarily improve the network representation of the atmospheric data. When the network is trained with pressure and wind components together there is a tangible improvement, especially for the winds. The information regarding pressure helps the network to better interpret the wind fields. The improvement from single to multi-field representation is minimal when pressure is combined together with precipitation, rainfall information helps the network to better learn the pressure structure rather then the opposite. Whereas in the case of training using just the two wind components the performance of the network worsen a lot. By having only the wind information the network is not able to reproduce correctly the atmospheric structure underlying the ETCs fields but it generates chaotic and blurry fields. Even though multi-field representation does not considerably improve the result of single-field representation it has a critical advantage. If we use more than one variable for the network training, when we generate a new image the fields composing it are related to each other, so the situation can be analyzed considering more than one atmospheric variable. The combination of all 4 fields together is believed to improve even more the fields representation because of the influence that rainfall has on pressure and that pressure has on the wind components. Because of the lack in computational resources this aspect was not explored.
How to verify consistency in the generated synthetic ETCs fields? Is similarity in statistical distribution a good indicator? Is it possible to evaluate the atmospheric physical structure with quantitative metrics or just by qualitative comments?

To verify the consistency in the generated fields the original sample and the synthetic samples have been compared by considering the histograms and the average fields. 4 cyclones have been taken as case studies to compare the original snapshots to the frames resulting from an interpolation between vectors in the latent space. The interpolation is computed between the vectors originating the most similar ones to the initial and final ones of the original track. Similarity in the statistical distribution is a sufficient indicator to exclude sample produced by models which do not have a statistical resemblance to the original sample. For example model W-UV and W-PR could be easily discarded by looking at the histograms comparison (Fig. 4.2c, 4.3c and 4.1a). While is not helpful to understand the structure and the feature of the generated fields, in fact model like W-PUV could have been considered as good runs while having a scarce representation of the ETCS fields (Fig. 4.1b, 4.2b, 4.3b, 4.9c, 4.10b and 4.11b). The evaluation of the atmospheric physical structure cannot be done using quantitative metrics, mostly because of the difficulties to find summary measures to parameterize ETCs fields. However the possibility to explore an empirical filter which learns pattern of the storms fields by training on the original data could be an option to investigate.

How are latent space values influencing the generation of ETC fields? Is the relationship between latent space vectors and image generation stable and respecting the physical structure of the data?

Understanding the relationship between latent space values and the images generated by the GAN is probably the most difficult analysis regarding this learning framework. It would mean that there is the ability to understand the operations done by the generator in each layer and how they are affected by the initial values in the random vectors. The generator mapping of the latent space is very complicated and difficult to understand, but it can be investigated by adopting different procedures such as looking for semantics encoding or inverting the generator (Creswell and Bharath (2018), Shen et al. (2019)). These procedures are complicated and time consuming for the project. To address the stability and the physical meaning of the relationship between latent space vectors and image generation the linear interpolation in the latent space has been considered. It also gives interesting feedback to understand if the network has learned the atmospheric structure underlying the images. The images generated from the interpolated vectors are very similar, compared to a random guess, to the original snapshots of the cyclone track and present similar atmospheric features. This also means that the latent space is not randomly connected with the generation of images. Instead, the generator mapping of the latent space shows that variations in the latent space vector are coherently matched in the images generation, indicating a stable behaviour of the system.

The project shows promising results for what concerns the generation of ETCs fields in the North Atlantic domain using a GAN, both from a statistical point of view and a physical point of view. In fact, by looking at images generated by the generator it appears that the network has learned most of the features characterizing the atmospheric structure of this events. The results obtained with this analysis encourage to go more deepen in the generation of ETCs data, in particular considering the tracks. The interpolated tracks suggest that the PGGAN is able to learn a good part of the features characterizing the evolution of a cyclone. The implementation of sequential modeling in the network to allow the training on the cyclone tracks looks very promising for generating synthetic tracks presenting similar atmospheric features to the original one. A tool able to generate meaningful cyclone tracks surely is an important resource to evaluate their impact and the related risks. The combination of such a tool with a predictive learning module might be very useful also for cyclone tracks forecast. An ETC track forecast framework can be designed similarly to the one developed in Liu and Lee (2020) for weather forecasting. As commented by the authors the combination of GAN based models and RNN based meteorological predictive models (e.g. ConvLSTM) helps overcome the most relevant problems associated to these methods. In fact predictive models produce blurry predictions while GANs are able to generate realistic looking frames but cannot catch the atmospheric evolution patterns and variations in their forecasts. The combination of both methods helps to better handle the uncertainties in the predicted frames and to generate realistic meteorological forecasts. Unfortunately, it must be considered that for training GANs usually dataset having a big dimension must be taken into account, otherwise is very risk to end up having problem of mode collapse, discriminator overfitting or generation of replicas and noisy outputs. The dataset used for this project contains 12'228 tracks, which is a limited amount for PGGAN training. However, a recent research could help resolve the problem of a small dataset. Karras et al. (2020) proposed a new framework to help GANs training on small dataset, the Adaptive Discriminator Augmentation (ADA). It consists of strategically add training images that have been augmented via cropping, rotating, color filtering or similar operation. It is important in the augmentation to keep the data look realistic.

Bibliography

M. M. Ali, C. M. Kishtawal, and S. Jain. Predicting cyclone tracks in the north indian ocean: An artificial neural network approach. *Geophysical Research Letters*, 34:L04603, 2007.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv e-prints*, art. arXiv:1701.07875, January 2017.

J. Baik and J. Paek. A neural network model for predicting typhoon intensity. *Journal of the Meteorological Society of Japan*, 78:857–869, 2000.

J. Bartholy, R. Pongrácz, and M. Pattantyús-Ábrahám. European cyclone track analysis based on ecmwf era-40 data sets. *International Journal of Climatology*, 26:1517–1527, 2006.

David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *CoRR*, abs/1705.10743, 2017. URL `http://arxiv.org/abs/1705.10743`.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. *Pearson Correlation Coefficient*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-00296-0. doi: 10.1007/978-3-642-00296-0_5. URL `https://doi.org/10.1007/978-3-642-00296-0_5`.

Alexander Bihlo. A generative adversarial network approach to (ensemble) weather prediction, 2020.

J. Bjerknes. On the structure of moving cyclones. *Monthly Weather Review*, 47:95–99, 1919.

J. Bjerknes and H. Solberg. Life cycle of cyclones and the polar front theory of atmospheric circulation. *Geofysiske Publikasjoner*, 3:1–18, 1922.

R. Blender and M. Schubert. Cyclone tracking in different spatial and temporal resolutions. *Monthly Weather Review*, 128:377–384, 2000.

R. Blender, K. Fraedrich, and F. Lunkeit. Identification of cyclone-track regimes in the north atlantic. *Quarterly Journal of the Royal Meteorological Society*, 123:727–741, 1997.

Ali Borji. Pros and cons of GAN evaluation measures. *CoRR*, abs/1802.03446, 2018. URL `http://arxiv.org/abs/1802.03446`.

K. A. Browning and N. M. Roberts. Structure of a frontal cyclone. *Quarterly Journal of the Royal Meteorological Society*, 120:1535–1557, 1990.

Jason Brownlee. How to implement progressive growing gan models in keras. `https://machinelearningmastery.com/how-to-implement-progressive-growing-gan-models-in-keras/`, 8 2019a.

Jason Brownlee. A gentle introduction to the progressive growing gan. `https://machinelearningmastery.com/introduction-to-progressive-growing-generative-adversarial-networks/`, 8 2019b.

Jason Brownlee. Understand the impact of learning rate on neural network performance. `https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning\-neural-networks/`, 1 2019c.

Jason Brownlee. How to train a progressive growing gan in keras for synthesizing faces. `https://machinelearningmastery.com/how-to-train-a-progressive-growing-gan-in-\keras-for-synthesizing-faces/`, 8 2019d.

Jason Brownlee. How to implement wasserstein loss for generative adversarial networks. `https://machinelearningmastery.com/how-to-implement-wasserstein-loss-for-\generative-adversarial-networks/`, 7 2019e.

Vitaly Bushaev. Understanding rmsprop - faster neural network learning. `https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a`, 9 2018.

T. N. Carlson. Airflow through midlatitude cyclones and the comma cloud pattern. *Monthly Weather Review*, 108:1498–1509, 1980.

J. G. Charney. The dynamics of long waves in a baroclinic westerly current. *Journal of Meteorology*, 4:135–163, 1947.

Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Jianxin Cheng, Jin Liu, Zhou Xu, Chenkai Shen, and Qiuming Kuang. Generating high-resolution climate prediction through generative adversarial network. *Procedia Computer Science*, 174:123 – 127, 2020. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2020.06.067. URL `http://www.sciencedirect.com/science/article/pii/S1877050920315817`. 2019 International Conference on Identification, Information and Knowledge in the Internet of Things.

Aidan Clark, Jeff Donahue, and Karen Simonyan. Efficient video generation on complex datasets. *CoRR*, abs/1907.06571, 2019. URL `http://arxiv.org/abs/1907.06571`.

W. R. Cotton and R. A. Anthes. The mesoscale structure of extra-tropical cyclones and middle and high clouds. *Storm and Cloud Dynamics*, 99:527–672, 2011.

Antonia Creswell and Anil A. Bharath. Inverting the generator of A generative adversarial network (II). *CoRR*, abs/1802.05701, 2018. URL `http://arxiv.org/abs/1802.05701`.

H. F. Dacre and S. L. Gray. The spatial distribution and evolution characteristics of north atlantic cyclones. *Monthly Weather Review*, 137:99–115, 2009.

H. F. Dacre, M. K. Hawcroft, M. A. Stringer, and K. I. Hodges. An extratropical cyclone atlas: A tool for illustrating cyclone structure and evolution characteristics. *Bulletin of the American Meteorological Society*, 93: 1497–1502, 2012.

P. De Luca, J. K. Hillier, R. L. Wilby, N. W. Quinn, and S. Harrigan. Extreme multi-basin flooding linked with extra-tropical cyclones. *Environmental Research Letters*, 12:114009, 2017.

Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks, 2015.

S. Eckhardt, A. Stohl, H. Wernli, P. James, C. Forster, and N. Spichtinger. A 15-year climatology of warm conveyor belts. *Journal of Climate*, 17:218–237, 2004.

The editors of Encyclopaedia Britannica. Extratropical cyclone. `https://www.britannica.com/science/extratropical-cyclone`, 1 2020.

T. Eichler and W. Higgins. Climatology and enso-related variability of north american extratropical cyclone activity. *Journal of Climate*, 19:2076–2093, 2006.

K. Fraedrich, R. Bach, and G. Naujokat. Single station climatology of central european fronts: Number, time and precipitation statistics. *Contribution to Atmospheric Physics*, 59:54–65, 1986.

David John Gagne II, Hannah M. Christensen, Aneesh C. Subramanian, and Adam H. Monahan. Machine learning for stochastic parameterization: Generative adversarial networks in the lorenz '96 model. *Journal of Advances in Modeling Earth Systems*, 12(3):e2019MS001896, 2020. doi: 10.1029/2019MS001896. URL `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001896`.

Q. Geng and M. Sugi. Variability of the north atlantic cyclone activity in winter analyzed from ncep-ncar reanalysis data. *Journal of Climate*, 14:3863–3873, 2001.

Carlos Gershenson. Artificial neural networks for beginners. *CoRR*, cs.NE/0308031, 2003. URL `http://arxiv.org/abs/cs/0308031`.

S. Giffard-Roisin, M. Yang, G. Charpiat, B. Kégl, and C. Monteleoni. Fused deep learning for hurricane track forecast from reanalysis data. *Climate Workshop Proceedings*, 2018.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversial nets. *Proceedings of the International Conference on Neural Information Processing Systems*, 114:2672–2680, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017. URL `http://arxiv.org/abs/1701.00160`.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf`.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: 10.1002/qj.3803. URL `https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803`.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

M. Hirsch, A DeGaetano, and S. Colucci. An east coast winter climatology. *Journal of Climate*, 14:882–899, 2001.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

K. I. Hodges. A general method for tracking analysis and its application to meterorological data. *Monthly Weather Review*, 122:2573–2586, 1994.

K. I. Hodges, B. J. Hoskins, J. Boyle, and C. Thorncroft. A comparison of recent reanalysis datasets using objective feature tracking: Storm tracks and tropical easterly waves. *Monthly Weather Review*, 131:2012–2037, 2003.

R. A. Houze Jr. and P. V. Hobbs. Organization and structure of precipitating cloud systems. *Advances in Geophysics*, 24:225–315, 1982.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4), July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073659. URL `https://doi.org/10.1145/3072959.3073659`.

T. Jung, S. K. Gulev, I. Rudeva, and V. Soloviov. Sensitiviy of extratropical cyclone characteristics to horizontal resolution in the ecmwf model. *Quarterly Journal of the Royal Meteorological Society*, 132:1839–1857, 2006.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. URL `http://arxiv.org/abs/1710.10196`.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. URL `http://arxiv.org/abs/1812.04948`.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

Siddharth Krishna Kumar. On weight initialization in deep neural networks. *CoRR*, abs/1704.08863, 2017. URL `http://arxiv.org/abs/1704.08863`.

W. König, R. Sausen, and F. Sielmann. Objective identification of cyclones in gcm simulations. *Journal of Climate*, 6:2217–2231, 1993.

D. A. Lavers, R. P. Allan, E. F. Wood, G. Villarini, D. J. Brayshaw, and A. J. Wade. Winter floods in britain are connected to atmospheric rivers. *Geophysical Research Letters*, 38:1–8, 2011.

S. Lawrence, C. L. Giles, Ah Chung Tsoi, and A. D. Back. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

Y LeCun, Y Bengio, and G Hinton. Deep learning. *Nature*, 521:436–444, 2015. URL `https://doi.org/10.1038/nature14539`.

Yujia Li, Alexander Schwing, Kuan-Chieh Wang, and Richard Zemel. Dualing gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5606–5616. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7144-dualing-gans.pdf`.

Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1498–1507. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/7423-pacgan-the-power-of-two-samples-in-generative-adversarial-networks.pdf`.

H. Liu and I. Lee. Mpl-gan: Toward realistic meteorological predictive learning using conditional gan. *IEEE Access*, 8:93179–93186, 2020.

Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. *CoRR*, abs/1606.07536, 2016. URL `http://arxiv.org/abs/1606.07536`.

T. Loridan and R. P. Crompton. A machine learning approach to modeling tropical cyclone wind field uncertainty. *Monthly Weather Review*, 145:3203–3221, 2017.

Merriam-Webster. Extratropical cyclone. `https://www.merriam-webster.com/dictionary/extratropical%20cyclone`, 3 2020.

U. Neu, M. G. Akperov, Nina Bellenbaum, Rasmus Benestad, Richard Blender, Rodrigo Caballero, Angela Cocozza, Helen F. Dacre, Yang Feng, Klaus Fraedrich, Jens Grieger, Sergey Gulev, John Hanley, Tim Hewson, Masaru Inatsu, Kevin Keay, Sarah F. Kew, Ina Kindem, Gregor C. Leckebush, Margarida L. R. Liberato, Piero Lionello, Igor I. Mokhov, Joaquim G. Pinto, Christoph C. Raible, Marco Reale, Irina Rudeva, Mareike Schuster, Ian Simmonds, Mark Sinclair, Michael Sprenger, Natalia D. Tilinina, Isabel F. Trigo, Sven Ulbrich, Uwe Ulbrich, Xiaolan L. Wang, and Heini Wernli. Imilast: A community effort to intercompare extra-tropical cyclone detection and tracking algorithms. *Bulletin of the American Meteorological Society*, 94:529–547, 2013.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. volume 70 of *Proceedings of Machine Learning Research*, pages 2642–2651, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/odena17a.html`.

J. G. Pinto and P. Ludwig. Extratropical cyclones over the north atlantic and western europe during the last glacial maximum and implications for proxy interpretation. *Climate of the Past*, 16(2):611–626, 2020. doi: 10.5194/cp-16-611-2020. URL `https://cp.copernicus.org/articles/16/611/2020/`.

J.G. Pinto, T. Spangehl, U. Ulbrich, and P. Speth. Sensitivities of a cyclone detection and tracking algorithm: Individual tracks and climatology. *Meteorologische Zeitschrift*, 14:823–838, 2005.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.

J. F. Roberts, L. C. Dawkins, K. I. Hodges, L. C. Shaffrey, D. B. Stephenson, M. A. Stringer, H. E. Thornton, and B. D. Youngman. The xws open access catalogue of extreme european windostrms from 1979 to 2012. *Natural Hazards and Earth System Sciences*, 14:2487–2501, 2014.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016. URL `http://arxiv.org/abs/1609.04747`.

Mario Ruttgers, Sangseung Lee, Soohwan Jeon, and Donghyun You. Prediction of a typhoon track using a generative adversarial network and satellite images. *Scientific Reports*, 9, 2019. URL `https://doi.org/10.1038/s41598-019-42339-y`.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016. URL `http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf`.

Victor Schmidt, Mustafa Alghali, Kris Sankaran, Tianle Yuan, and Yoshua Bengio. Modeling cloud reflectance fields using conditional generative adversarial networks, 2020.

Philip Sedgwick. Spearman's rank correlation coefficient. *BMJ*, 349, 2014. ISSN 0959-8138. doi: 10.1136/bmj.g7327. URL `https://www.bmj.com/content/349/bmj.g7327`.

A. T. Semple. A review and unification of conceptual models of cyclogenesis. *Meteorological Applications*, 10: 39–59, 2003.

National Park Service. Extratropical storms. `https://www.nps.gov/articles/extratropical-storms.htm`, June 2019.

A. Shapiro and D. Keyser. On the structure and dynamics of fronts, jet streams and the tropopause. *Extratropical Cyclones: The Erik Palmén Memorial Volume*, pages 167–191, 1990.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. *CoRR*, abs/1907.10786, 2019. URL `http://arxiv.org/abs/1907.10786`.

Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, Mar 2020. ISSN 0167-2789. doi: 10.1016/j.physd.2019.132306. URL `http://dx.doi.org/10.1016/j.physd.2019.132306`.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015. URL `http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcasting.pdf`.

Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Deep learning for precipitation nowcasting: A benchmark and a new model. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5617–5627. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7145-deep-learning-for-precipitation-nowcasting-a-benchmark-and-a-new-model.pdf`.

Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? *CoRR*, abs/1807.09499, 2018. URL `http://arxiv.org/abs/1807.09499`.

M. Sickmöller, R. Blender, and K. Fraedrich. Observed winter cyclone tracks in the northern hemisphere in re-analysed ecmwf data. *Quarterly Journal of the Royal Meteorological Society*, 126:591–620, 2000.

M. R. Sinclair. An objective cyclone climatology for the southern hemisphere. *Monthly Weather Review*, 122: 2239–2256, 1994.

Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In *ICML*, pages 1017–1024, 2011. URL `https://icml.cc/2011/papers/524_icmlpaper.pdf`.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf`.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

U. Ulbrich, J. G. Pinto, and G. C. Leckebusch. Extra-tropical cyclones in the present and future climate: a review. *Theoretical Applied Climatology*, 96:117–131, 2009.

H. Volkert. Components of the norwegian cyclone model: Observations and theoretical ideas in europe prior to 1920. *The Life Cycles of Extratropical Cyclones*, pages 15–28, 1999.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *CoRR*, abs/1609.02612, 2016. URL `http://arxiv.org/abs/1609.02612`.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

X. L. Wang, V. R. Swail, and F. W. Zwiers. Climatology and changes of extratropical cyclone activity: Comparison of era-40 with ncep-ncar reanalysis for 1958-2001. *Journal of Climate*, 19:3145–3166, 2006.

Lilian Weng. From GAN to WGAN. *CoRR*, abs/1904.08994, 2019. URL `http://arxiv.org/abs/1904.08994`.

Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

6

Appendix A
Discriminator and Generator architecture

	Type of Layer	Output shape	# Parameters
64x64 Resolution Block	Input Layer	(n_samples, 64, 64, n_c)	0
	Conv2D	(n_samples, 64, 64, n_filters)	256
	Leaky ReLU	(n_samples, 64, 64, n_filters)	0
	Conv2D	(n_samples, 64, 64, n_filters)	147’584
	Leaky ReLU	(n_samples, 64, 64, n_filters)	0
	Conv2D	(n_samples, 64, 64, n_filters)	147’584
	Leaky ReLU	(n_samples, 64, 64, n_filters)	0
32x32 Resolution Block	Average Pooling	(n_samples, 32, 32, n_filters)	0
	Conv2D	(n_samples, 32, 32, n_filters)	147’584
	Leaky ReLU	(n_samples, 32, 32, n_filters)	0
	Conv2D	(n_samples, 32, 32, n_filters)	147’584
	Leaky ReLU	(n_samples, 32, 32, n_filters)	0
16x16 Resolution Block	Average Pooling	(n_samples, 16, 16, n_filters)	0
	Conv2D	(n_samples, 16, 16, n_filters)	147’584
	Leaky ReLU	(n_samples, 16, 16, n_filters)	0
	Conv2D	(n_samples, 16, 16, n_filters)	147’584
	Leaky ReLU	(n_samples, 16, 16, n_filters)	0
8x8 Resolution Block	Average Pooling	(n_samples, 8, 8, n_filters)	0
	Conv2D	(n_samples, 8, 8, n_filters)	147’584
	Leaky ReLU	(n_samples, 8, 8, n_filters)	0
	Conv2D	(n_samples, 8, 8, n_filters)	147’584
	Leaky ReLU	(n_samples, 8, 8, n_filters)	0
4x4Resolution Block	Average Polling	(n_samples, 4, 4, n_filters)	0
	MiniBatch Standard Deviation	(n_samples, 4, 4, n_filters)	0
	Conv2D	(n_samples, 4, 4, n_filters)	148’736
	Leaky ReLU	(n_samples, 4, 4, n_filters)	0
	Conv2D	(n_samples, 4, 4, n_filters)	262’272
	Leaky ReLU	(n_samples, 4, 4, n_filters)	0
	Flatten	(n_samples, 2048)	0
	Dense	(n_samples, 1)	2049
Total Number of Parameters			1’593’985

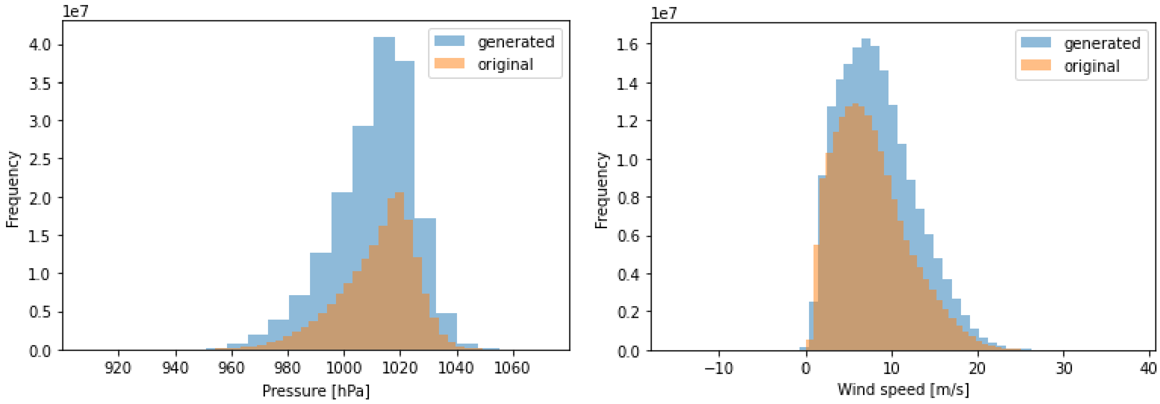
Table 6.1: **Discriminator architecture**: Every resolution block has a similar structure with a layer performing the down-scaling (average pooling), and 2 convolutional layers with Leaky ReLU activation function. The first block has also an input layer for the images and an additional convolutional layer. The last block also has the minibatch standard deviation layer, a flatten layer which converts the image to a vector and a fully connected layer which outputs a scalar. Regarding the output shape of each layer n_samples is the the number of examples passed as input to the discriminator, n_filters is the number of filters in the layer and ls_dim is latent space dimension.

	Type of Layer	Output shape	# Parameters
4x4 Resolution Block	Input Layer	(n_samples, ls_dim)	0
	Dense Layer	(n_samples, 2048)	264’192
	Reshape	(n_samples, 4, 4, n_filters)	0
	Conv2D	(n_samples, 4, 4, n_filters)	147’584
	Pixel Normalization	(n_samples, 4, 4, n_filters)	0
	Leaky ReLU	(n_samples, 4, 4, n_filters)	0
	Conv2D	(n_samples, 4, 4, n_filters)	147’584
	Pixel Normalization	(n_samples, 4, 4, n_filters)	0
	Leaky ReLU	(n_samples, 4, 4, n_filters)	0
8x8 Resolution Block	Up Sampling	(n_samples, 8, 8, n_filters)	0
	Conv2D	(n_samples, 8, 8, n_filters)	147’584
	Pixel Normalization	(n_samples, 8, 8, n_filters)	0
	Leaky ReLU	(n_samples, 8, 8, n_filters)	0
	Conv2D	(n_samples, 8, 8, n_filters)	147’584
	Pixel Normalization	(n_samples, 8, 8, n_filters)	0
	Leaky ReLU	(n_samples, 8, 8, n_filters)	0
16x16 Resolution Block	Up Sampling	(n_samples, 16, 16, n_filters)	0
	Conv2D	(n_samples, 16, 16, n_filters)	147’584
	Pixel Normalization	(n_samples, 16, 16, n_filters)	0
	Leaky ReLU	(n_samples, 16, 16, n_filters)	0
	Conv2D	(n_samples, 16, 16, n_filters)	147’584
	Pixel Normalization	(n_samples, 16, 16, n_filters)	0
	Leaky ReLU	(n_samples, 16, 16, n_filters)	0
32x32 Resolution Block	Up Sampling	(n_samples, 32, 32, n_filters)	0
	Conv2D	(n_samples, 32, 32, n_filters)	147’584
	Pixel Normalization	(n_samples, 32, 32, n_filters)	0
	Leaky ReLU	(n_samples, 32, 32, n_filters)	0
	Conv2D	(n_samples, 32, 32, n_filters)	147’584
	Pixel Normalization	(n_samples, 32, 32, n_filters)	0
	Leaky ReLU	(n_samples, 32, 32, n_filters)	0
64x64 Resolution Block	Up Sampling	(n_samples, 64, 64, n_filters)	0
	Conv2D	(n_samples, 64, 64, n_filters)	147’584
	Pixel Normalization	(n_samples, 64, 64, n_filters)	0
	Leaky ReLU	(n_samples, 64, 64, n_filters)	0
	Conv2D	(n_samples, 64, 64, n_filters)	147’584
	Pixel Normalization	(n_samples, 64, 64, n_filters)	0
	Leaky ReLU	(n_samples, 64, 64, n_filters)	0
	Conv2D	(n_samples, 64, 64, n_c)	129
Total Number of Parameters			1’740’161

Table 6.2: **Generator architecture**: Every resolution block has a similar structure, a layer performing the up-scaling (nearest neighbour), and 2 convolutional layers followed by pixel normalization and Leaky ReLU activation function. The first block has also an input layer for the latent space, a fully connected layer and a layer that reshapes the data to 4x4 images. The last block has an additional convolutional layer which outputs a 64x64 image. Regarding the output shape of each layer n_samples is the number of fake samples that are generated, n_filters is the number of filters for each layer and ls_dim is the dimension of the latent space.

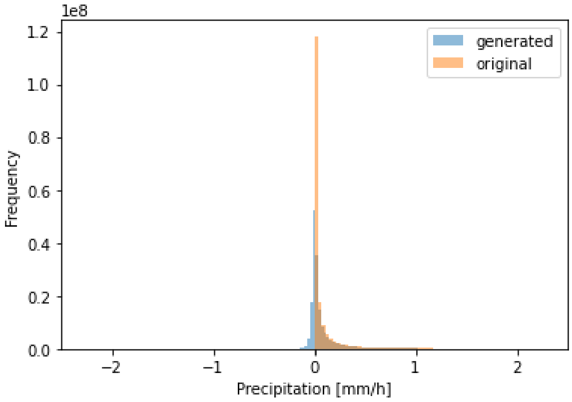
7

Appendix B
Model C-PWSR results



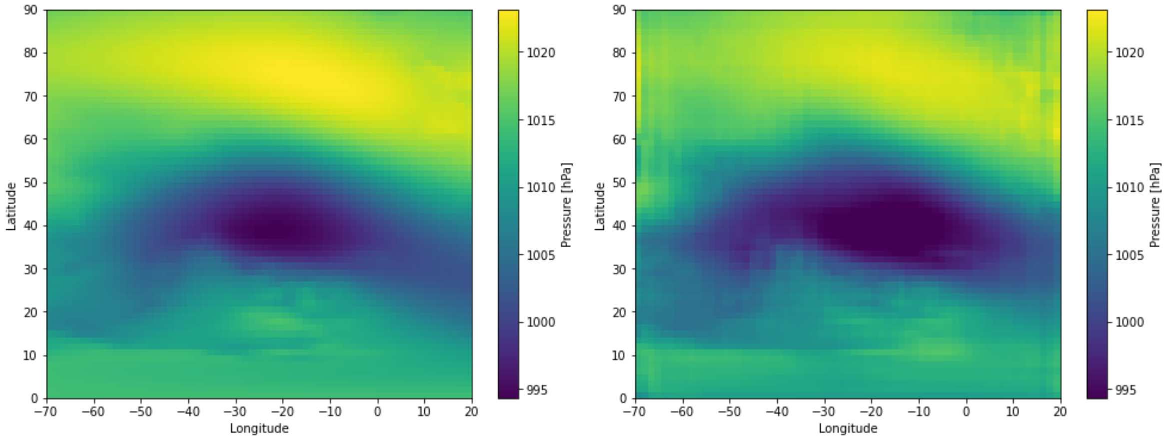
(a) Pressure

(b) Wind Speed



(c) Precipitation

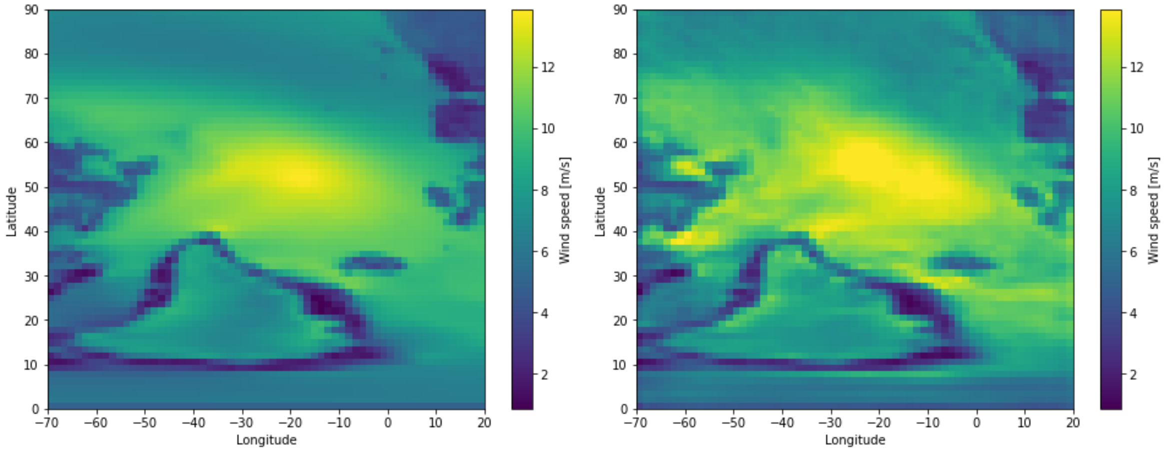
Figure 7.1: Histograms comparison for model C-PWSR.



(a) Original

(b) Generated

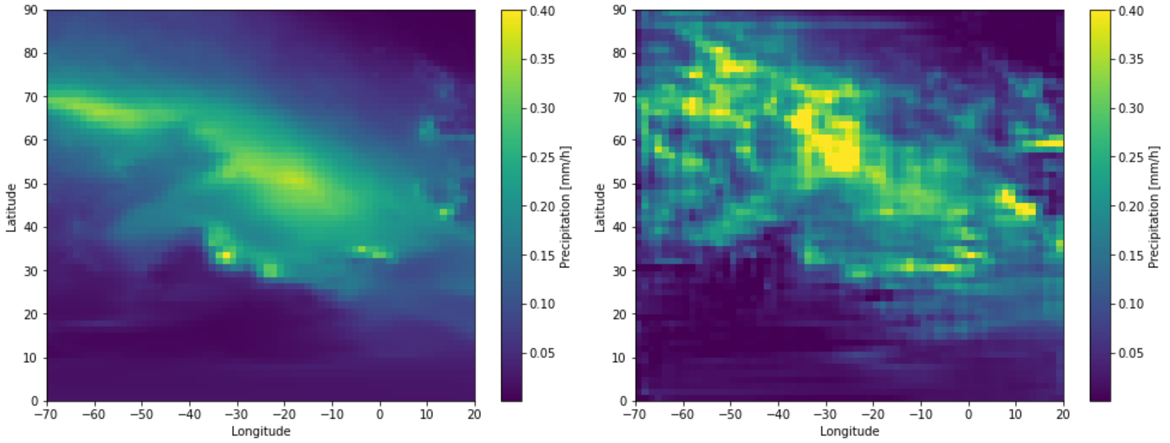
Figure 7.2: Average fields of pressure for the original sample and for the generated sample of model C-PWSR.



(a) Original

(b) Generated

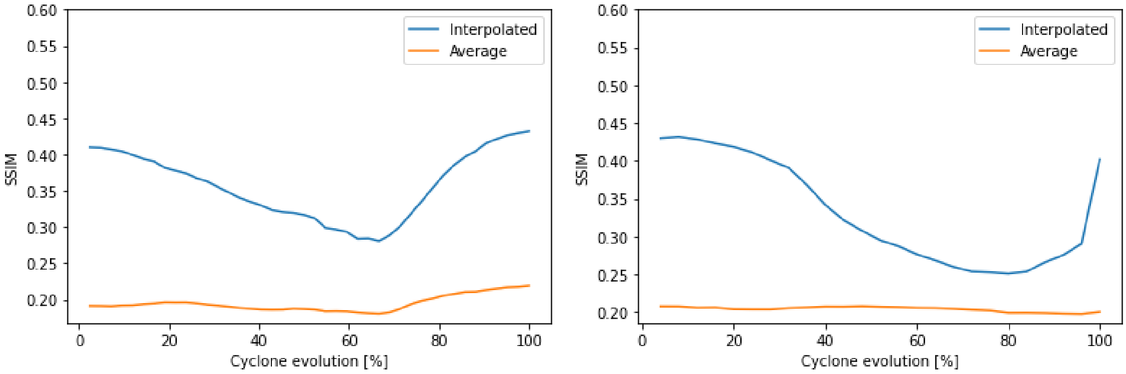
Figure 7.3: Average fields of wind speed for the original sample and for the generated sample of model C-PWSR.



(a) Original

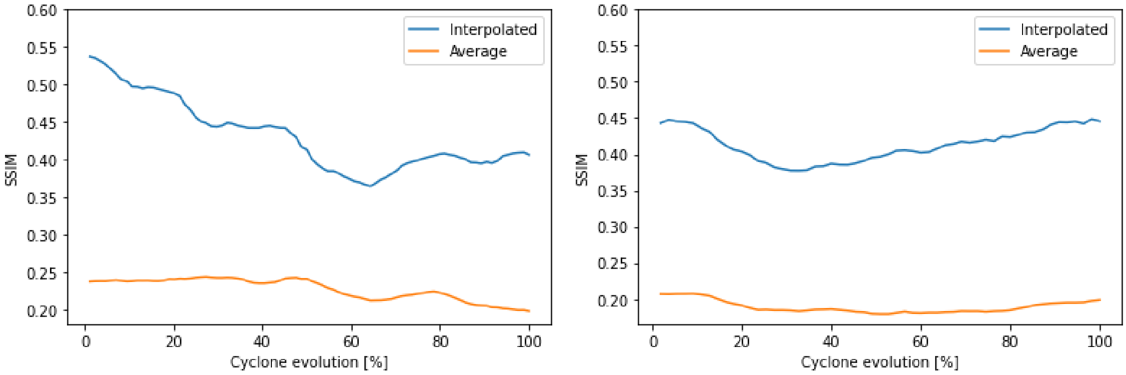
(b) Generated

Figure 7.4: Average fields of precipitation for the original sample and for the generated samples of model C-PWSR.



(a) Cyclone 257

(b) Cyclone 441



(c) Cyclone 643

(d) Cyclone 667

Figure 7.5: SSIM evolution between original snapshots of the cyclones tracks and interpolated generated images using Model C-PWSR. The most similar generated images to the initial and final frame are coming from a sample that has the same dimension of the original one. The SSIM evolution is compared with the average SSIM between the original snapshots and frames from a sample of 1’000 images.