THESIS

DEEP LEARNING FOR RADAR BEAM BLOCKAGE CORRECTION

Submitted by

Songjian Tan

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Fall 2023

Master's Committee:

Advisor: Haonan Chen

V Chandrasekar
Haonan Wang

ABSTRACT

DEEP LEARNING FOR RADAR BEAM BLOCKAGE CORRECTION

This thesis aims to propose a deep learning framework based on generative adversarial networks (GANs) for correcting partial beam blockage regions in polarimetric radar observations. The correction of such data is an essential step in radar data quality control and subsequent quantitative applications, especially in complex terrain environments. The proposed methodology is demonstrated using two S-band operational Weather Surveillance Radar - 1988 Doppler (WSR-88D) located in different regions of the western United States, characterized by different precipitation types. To train the GAN model, observation sectors of both radars are manually cropped to simulate partial beam blockage situations. The effectiveness of the trained models is demonstrated using independent precipitation events in Texas and California, and their generalization capacity is examined by cross-testing the data with different precipitation features. The beam blockage correction performance is compared with a traditional linear interpolation approach, and the results show that the proposed approach significantly improves the continuity of precipitation observations in both domains. While visible discrepancies exist between the models trained based on convective and stratiform precipitation events in Texas and California, respectively, both models outperform the traditional interpolation method. The repaired observations demonstrate great potential for improved quantitative applications, despite the unavailability of ground truth for real blocked radar data.

ACKNOWLEDGEMENTS

I am deeply grateful to all those who have supported me throughout my graduate journey and helped me reach this significant milestone. Firstly, I would like to express my sincere gratitude to my advisor Dr. Haonan Chen for his constant guidance, support, and invaluable feedback throughout my thesis research. His expertise, encouragement, and patience have been invaluable and have shaped this work in many ways. I would also like to thank the members of my thesis committee, Dr. V. Chandrasekar, and Dr. Haonan Wang, for their time, effort, and feedback, which has significantly improved the quality of my thesis. I am thankful to the faculty and staff of the Department of Electrical and Computer Engineering at Colorado State University for providing me with the resources and opportunities to conduct this research. I am also indebted to my family, who have provided me with endless love, support, and encouragement, and without whom this journey would not have been possible. Finally, I would like to thank my friends and colleagues in the Artificial Intelligence and Remote Sensing (AIRS) Laboratory for their valuable insights, stimulating discussions, and unwavering support throughout my graduate studies. Thank you all for being a part of this journey with me.

This research was supported by the U.S. National Science Foundation (NSF) Faculty Early Career Development Program (CAREER).

DEDICATION

I would like to dedicate this thesis to all those who accompanied me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Problem Statement and Motivations

Weather radar is a kind of indispensable tool for monitoring precipitation patterns, storm movements, and extreme weather events such as tornadoes. The information gathered by weather radar is the most intuitive way to describe various weather phenomena. Recent years have also seen advances in weather radar technology, allowing for the collection of more detailed and accurate atmospheric conditions. In practice, however, radar signals are often of low quality or missing due to physical occlusion and/or radio frequency interference. Owing to the distinctive characteristics of radar signals, different wavelengths of which will be absorbed or scattered with different intensities due to various precipitation, temperature, and humidity in the atmosphere, etc. resulting in signal attenuation. To minimize the impact of these problems and other data quality issues, the radar system employs various techniques such as adaptive signal processing, advanced calibration methods, and attenuation correction algorithms [1].

Nevertheless, a number of radar data problems still exist even after applying advanced radar processing techniques. Radar imagery is often susceptible to various forms of interference, including but not limited to noise, object occlusion, and sensor malfunction. These disturbances can result in a reduction in image quality or even partial loss of imagery, ultimately impairing the efficacy of radar image recognition and tracking. The most common of these is beam blockage or partial beam blockage (PBB) [2], which is caused by the presence of physical objects such as tall buildings or mountains within the radar beam, thus creating a wedge-shaped blind spot at the back end of the signal beam which can lead to low-quality application products and inaccurate weather forecasts and warnings [3]. To mitigate radar beam blockage, weather radar systems are often placed at elevated locations, such as on top of hills or high buildings, to reduce the beam blockage effect. In addition, radar networks are often designed to ensure that precipitation is detected

from multiple angles [4] [5] [6], thus improving overlapped coverage areas from multiple radars and reducing the chances of missing major weather echoes. Despite these strategies, radar beam blockage can still occur in some cases, such as when a thunderstorm or tornado approaches a radar station (i.e., not observed by other radars in the network environment). In such cases, meteorologists rely on other sources of weather information, such as ground-based observations, satellite imagery, or traditional radar image restoration methods [7]. This thesis aims to offer a new strategy for beam blockage correction in weather radar data.

On the other hand, image correction has long been a research area in the field of computer vision and image processing. It is a technique used to fill in missing or corrupted parts of an image by synthesizing new pixels based on the surrounding pixels. For radar image restoration, many different approaches have been applied, including traditional interpolation and statistical methods [8, 9]. While interpolation and statistical methods can be effective for image inpainting, they also have some limitations and disadvantages. Since interpolation methods rely on smoothness assumptions and may not be able to capture the texture and structure information in the image, they may produce blurry or unrealistic results, especially when the missing region is large or complex. In addition, interpolation methods may not work well when the missing pixels are located at the edges or corners of the image, as they may introduce artifacts or distortions in the image. Statistical methods may not work well when the missing pixels are part of complex or high-frequency structures, as they may not be able to capture the details and textures of the image. Also, statistical methods may require prior knowledge or assumptions about the image content, which may not always be available or accurate.

In recent years, owing to the latest breakthroughs in deep learning in the computer vision field, there have been different advances in image recognition, image classification, image segmentation, as well as image inpainting [10]. Deep learning-based image inpainting is a rapidly developing field that has shown great promise in restoring damaged or missing image information. Most of the earliest deep learning-based image inpainting methods were based on autoencoders, which could learn the low-dimensional representation of images and then modify the encoded low-dimensional

vectors for image restoration. However, these methods were limited in their ability to capture the global structure and texture information of images, and the results were often unsatisfactory. Elad and Aharon developed a convolutional neural network (CNN) to learn the mapping from the masked image to the restored image for image inpainting [11], which achieved promising results in generating high-quality restored images for simple images, but it was not able to handle complex image structures and texture information. Since the introduction of generative adversarial networks (GANs) [12], GANs are widely used in various applications including image inpainting. Essentially, GANs use two deep neural networks: a generator network and a discriminator network [12]. The generator network could generate the restored image, while the discriminator network would evaluate the quality of the restored image and provide feedback, enabling the generator network to generate a more realistic restored image. However, one of the limitations is that the traditional GANs can be unstable and can potentially produce artifacts in the restored images [13].

This thesis aims to build a conditional GAN (CGAN) framework for weather radar beam blockage correction. This CGAN model can more accurately restore the structure and texture information of images while reducing artifacts and blur in the restored images, thus is more suitable for complex weather radar data processing. A specialized model adapted to the variations of the radar operation environment is designed to address the unique characteristics of radar data. The model is trained and tested using weather data collected from two S-band operational Weather Surveillance Radar – 1988 Doppler (WSR-88D) systems deployed in two different precipitation regimes: KFWS radar in northern Texas which is characterized by convective precipitation, and KDAX radar in northern California, which is typically featured by shallow stratiform precipitation. In addition, this paper performs multifarious blockage simulations to solve beam blockage correction problems with diverse blockage sizes.

1.2 Literature Review

In terms of the field of image inpainting, deep learning based correction systems have demonstrated their strong inpainting performance in numerous articles. A number of deep learning

models with an encoding-decoding structure have been suggested, and the U-Net architecture has received the most attention and achieved the best results. U-Net was originally proposed by Ronneberger et al. [14] for biomedical image segmentation tasks and has since been extended to various other image processing applications, including image inpainting. For the purpose of high-resolution image inpainting, Yu et al. [10] developed a modified U-Net architecture. The suggested network, known as GUNet, produced high-quality inpainted images by combining a local discriminator with a global context encoder. Another U-Net with deep residual architecture for picture inpainting was proposed by Zhang et al. [15], which included the residual block to improve the network's capacity for feature learning. To enhance the U-Net architecture's performance for certain applications, other researchers have investigated a variety of improvements. To increase the accuracy and consistency of the inpainting outcomes, Qu et al. (2020) developed a guided U-Net for medical picture inpainting that included a guidance module.

Overall, U-Net and its variants have demonstrated strong performance in image inpainting tasks. However, the original U-Net has limitations in capturing multi-scale context information, which led to the development of U-Net++ by Zhou et al. in 2018 [16]. U-Net++ introduces dense skip connections to enhance information flow between layers and improve network performance in image restoration tasks. Many studies have shown that U-Net++ performs better than the original U-Net in various image restoration tasks. In a paper by Peng et al [17]., an improved UNet++ architecture is proposed for end-to-end change detection in multi-temporal satellite images. The results on a benchmark dataset demonstrate the effectiveness of the proposed approach in detecting changes with high accuracy. Overall, these studies demonstrate the effectiveness of Unet++ in image inpainting tasks, and suggest that its nested architecture and skip connections can help to capture fine-grained features and pocess large number of complex images. In other word, Unet++ has shown great potential for radar image restoration, and its ability to capture fine-grained details and preserve spatial information make it a promising choice for this task.

GAN-based models, on the other hand, have been shown to generate visually plausible and diverse images by training a generator and a discriminator network to compete against each other.

These models have shown significant improvements in handling complex image structures and preserving the texture and structure of the inpainted regions. For example, D. Pathak et al. proposed a GAN-based approach to inpainting [18], which involves learning to generate missing parts of an image by understanding the surrounding context. They used the GAN framework to generate image patches that are then used to fill in the missing regions. In addition, C. Yang et al. introduced a CGAN-based approach to image inpainting that can generate high-resolution images [19], which utilizes a generator network that is trained to fill in missing regions of an image, and a discriminator network that tries to distinguish between the generated images and the original images.

Overall, the capacity of CGAN to produce high-quality images and handle missing data and noise has demonstrated significant potential for radar image inpainting. The combination of the two techniques can result in more accurate restoration results since CGAN can produce high-quality images and Unet++ can preserve image details and texture information. Furthermore, the network can become more stable and scalable with the help of CGAN and Unet++. Based on the training data, CGAN can automatically adjust to various noise and missing patterns, whereas Unet++ can handle images of various sizes and resolutions.

1.3 Contributions of the Present Work

The main contributions of this work include: (a) manual simulation of weather radar beam blockage situation, (b) Construction, integration, and adjustment of the CGAN and Unet++ networks, (c) training neural networks using radar data from different regions containing different precipitation types, (d) the performance evaluation of the proposed method on each datase. The key contribution of this work is to develop the CGAN and Unet++ based beam blockage correction system for different types of precipitation.

To better describe the beam blockage problem, this thesis will first thoroughly explain how weather radar operates, the reasons for beam blockage in radar data, and how this affects precipitation forecasts to better explain the beam blockage issue. We will examine the climate at the radar location and the origin of convective, and stratiform precipitation in order to introduce both types

of precipitation data. Next, we will explain the general framework of the CGAN and Unet++ based correction system, including the data preprocessing steps and the architecture of the neural networks. We will highlight the training process of the neural networks and the criteria used to evaluate the performance of the system. We will also discuss the specific model used for training and comparison, as well as the experimental setup and any hyperparameters used. On the basis of the correction results, observations and conclusions are made in the end. There will also be a discussion of the system's shortcomings and possible research trajectories.

1.4 Thesis Organization

The structure of the remaining chapters of this thesis is as follows: Chapter 2 presents a comprehensive analysis of the operation of weather radar, typical observations, and missing data for beam blockage. The mechanisms of convective and stratiform precipitation, the associated radars used, and some processing of the radar data are all covered in Chapter 3. In Chapter 4, the network layers of the model, which include Unet++, CGAN's generator, and discriminator are discussed, along with their internal organization and operation. The results obtained are evaluated in Chapter 5, and Chapter 6 draws conclusions regarding the suggested methodology and discusses some potential areas for future research.

Chapter 2

Review of Weather Radar

2.1 Introduction

Radar, also known as radio detecting and ranging, is a technique that uses radio waves to locate and detect objects. It was initially created in the early 20th century as a way to identify the presence of ships and airplanes at a distance. Radar operates by sending out a radio wave signal from a transmitter, which then travels through the air and reflects off of any objects in its path. Subsequent to emission, the reflected signal is apprehended by a receiver that gauges the duration it took for the signal to rebound and computes the distance to the object. By persistently emitting and receiving electromagnetic waves, radar technology can form an intricate illustration of the immediate milieu, encompassing the whereabouts, velocity, and trajectory of mobile objects. The aforementioned data is pivotal in numerous domains, including but not limited to air traffic control, meteorological prognostication, and military maneuvers. Numerous types of radar can be classified using various methods. They can be categorized based on their positioning method, which includes active radar, semi-active radar, and passive radar. They can also be categorized based on their installation location, such as ground-based radar, sea-based radar, aviation radar, and satellite radar. Radar can also be classified based on the type of radiation they emit, such as pulse radar and continuous wave radar. Furthermore, they can be classified based on the wavelength they operate on, such as meter wave radar, decimeter wave radar, centimeter wave radar, and other frequency band radars. Finally, they can also be classified based on their application, which includes target detection radar, reconnaissance radar, weapon control radar, flight support radar, weather radar, navigation radar, and many others.

Therein, weather radar is a type of radar that is specifically designed to detect and measure precipitation in the atmosphere. It uses radio waves to detect and locate the location, intensity, and movement of rain, snow, hail, and other types of precipitation. It operates by emitting short pulses

of radio waves that are transmitted into the atmosphere. These waves then encounter precipitation in the atmosphere, and some of the energy is scattered back to the radar antenna. The radar antenna then receives the scattered energy and uses it to create a picture of the precipitation in the atmosphere [20].

The radar can determine the intensity of the precipitation based on the strength of the scattered energy, and the movement of the precipitation based on the Doppler shift of the scattered energy. By combining this information with data from other sources, such as satellite imagery and surface observations, weather forecasters can create detailed and accurate forecasts of weather conditions. A weather radar is essential for a wide range of applications, from aviation and shipping to agriculture and emergency management. It is also used in research, such as in the study of severe weather events like tornadoes and hurricanes.

Advances in radar technology have led to the development of newer and more sophisticated weather radar systems, such as dual-polarization radar and phased array radar. These systems offer improved accuracy and resolution, making it possible to detect smaller particles in the atmosphere and better predict the path and intensity of storms.

This chapter will comprehensively examine the constituent components of a weather radar system to elucidate its operational mechanism, thereby expounding on the genesis of radar data and observations. Additionally, this chapter will conclude with a detailed analysis of factors that may lead to beam blockage in such systems.

2.2 Weather Radar Working Principle

Weather radar functions by transmitting radio wave pulses that propagate throughout the atmosphere, scattering in all directions when encountering precipitation or other objects. A fraction of the scattered energy returns to the radar antenna and is detected by a receiver, where the time delay between transmitted and received signals, alongside amplitude and phase information, is utilized to compute the distance, direction, and intensity of the precipitation or other targets [21]. The principle of working is as shown in Figure 2.1.

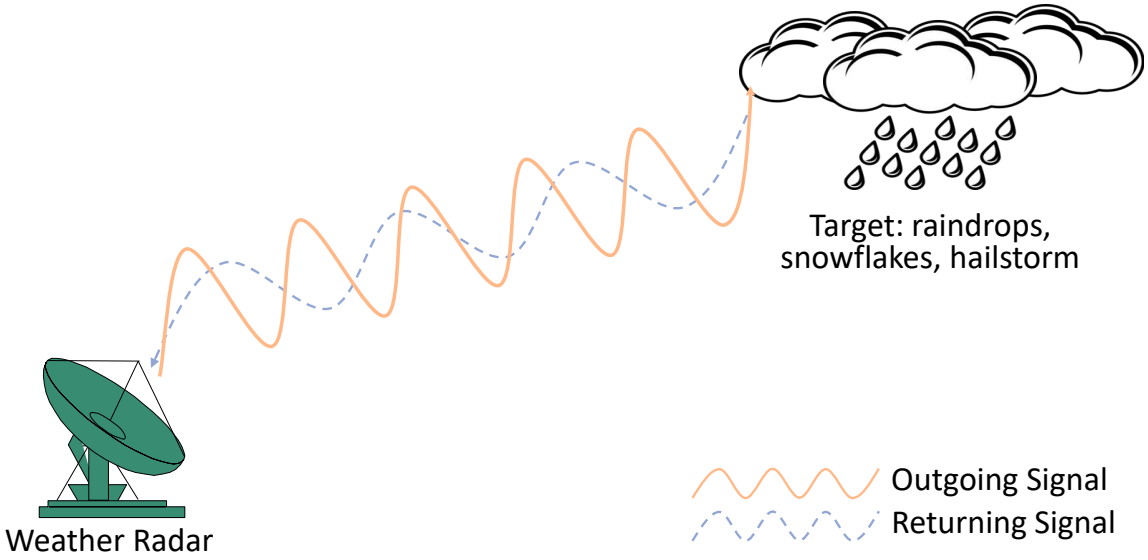

Figure 2.1: Weather radar basic working principle.

A weather radar system constitutes an antenna, a transmitter, a receiver, a signal processor, and a display. The transmitter produces high-frequency electromagnetic energy pulses, typically in the microwave range, directed through a parabolic dish antenna into the atmosphere. When precipitation or other targets are present, the energy scatters in various directions, including back toward the antenna. A second antenna, typically located near the transmitter, receives and amplifies the scattered energy, isolating the desired return signal from ambient noise. The resulting signal is digitized and processed by a computer that applies diverse algorithms to determine the location and intensity of precipitation or other targets. The processed data is then exhibited on a display monitor, providing a visual representation of precipitation or other targets in the atmosphere. Meteorologists use this information to forecast weather patterns and issue warnings for severe weather events such as thunderstorms, tornadoes, and hurricanes.

Figure 2.2 illustrates the interconnections among the diverse constituents of the weather radar, which we will scrutinize individually in the upcoming subsections.

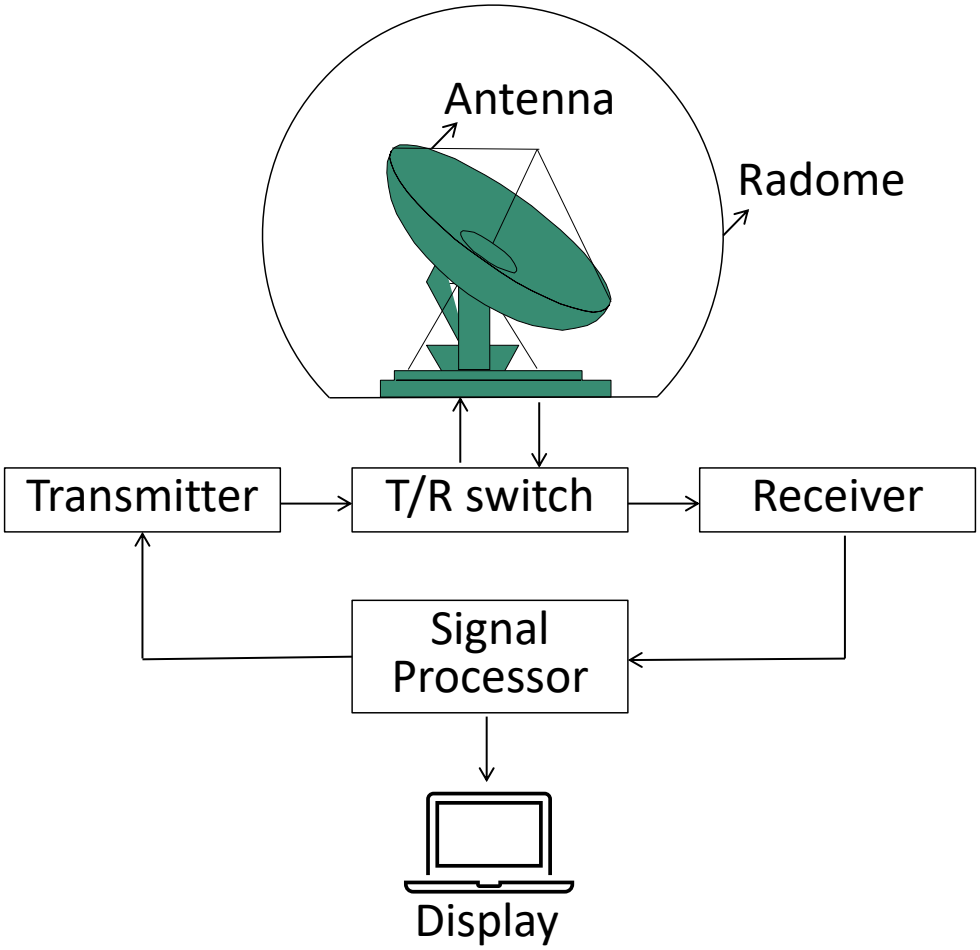

Figure 2.2: Basic Components of a Weather Radar System

2.2.1 Transmitter

A vital component of a weather radar system is the weather radar transmitter, which generates high-frequency electromagnetic pulses that propagate through the atmosphere and reflect back from various particles such as raindrops, snowflakes, and hailstones. The transmitter's primary function is to produce and shape these pulses to meet specific weather detection requirements.

Weather radar transmitters generally consist of a power supply, a modulator, a pulse-forming network (PFN), and a high-power amplifier (HPA). The power supply provides the necessary electrical energy to the transmitter. The modulator shapes the pulses generated by the transmitter by converting a continuous wave (CW) signal into a pulsed signal using an on-off keying (OOK)

technique. The PFN shapes the pulses produced by the modulator into a desired waveform using a series of capacitors and inductors. The HPA amplifies the pulsed signal produced by the PFN to the required power level using a traveling wave tube (TWT) or a solid-state power amplifier (SSPA).

Weather radar transmitters may also include other devices, such as a circulator or a duplexer, which is the T/R switch in Figure 2.2, to separate the transmitted and received signals. The transmitter's performance depends on several factors, such as its peak power output, pulse width, and repetition rate. A high-performance transmitter can produce high-quality radar images with greater accuracy and resolution.

2.2.2 Receiver

A weather radar system necessitates the use of a crucial component known as the weather radar receiver. This element is responsible for detecting and processing signals that are received after they have been transmitted and reflected by atmospheric particles like raindrops, snowflakes, and hailstones. The receiver performs signal extraction and amplification before converting them into digital data that can be further processed and displayed.

The weather radar receiver comprises essential components, such as a low-noise amplifier, a mixer, an intermediate frequency amplifier, a filter, and an analog-to-digital converter (ADC). The low-noise amplifier (LNA) plays a vital role in amplifying weak radar signals received by the antenna while adding minimal noise to the signal. The LNA operates at low temperatures to minimize the impact of thermal noise. The mixer combines the amplified signal with a reference signal from a local oscillator (LO) to produce an intermediate frequency (IF) signal. The IF signal is the difference between the frequency of the received signal and the LO frequency. The IF amplifier then amplifies the IF signal to a level that can be processed by subsequent components in the receiver. The filter eliminates unwanted signals like noise and interference from the IF signal. The filter can be a bandpass filter that only passes the desired radar frequency range, or a low-pass filter that removes high-frequency noise. The analog-to-digital converter (ADC) then converts the filtered analog signal into a digital signal that can be processed by a computer or display device.

Furthermore, weather radar receivers may contain other devices, such as a phase-locked loop (PLL) or a digital signal processor (DSP), which enhance the performance of the receiver. The effectiveness of a weather radar receiver relies on multiple factors, such as its sensitivity, dynamic range, and frequency response. An efficient receiver can precisely detect weak radar signals and accurately measure the intensity and location of precipitation.

2.2.3 Other Components

In addition to the transmitter and receiver, a weather radar system includes several other components that work together to detect and process weather information. These components include the antenna, the signal processor, the display system, and the control system.

The antenna is responsible for transmitting and receiving radar signals. It is typically a parabolic dish or an array of antennas that are designed to focus and direct the radar beam. The size and design of the antenna depend on the frequency and range of the radar system. The signal processor receives the digitized radar signals from the receiver and applies various signal processing techniques to extract information about the weather. These techniques include pulse compression, Doppler filtering, and clutter suppression. The signal processor also calculates the location, intensity, and movement of precipitation.

The display system presents the radar data in a user-friendly format that can be easily interpreted by meteorologists and other users. The display system may include a variety of graphical and numerical displays, such as reflectivity maps, velocity maps, and storm tracks. The control system manages the operation of the weather radar system, including the configuration of the transmitter and receiver, the operation of the signal processor, and the display of the radar data. The control system may also include a user interface that allows meteorologists and other users to interact with the radar system.

The performance of a weather radar system depends on the quality and integration of these components. A high-performance weather radar system can provide accurate and timely informa-

tion about the weather, which is essential for weather forecasting, severe weather warnings, and aviation safety.

2.3 Weather Radar Data

Understanding the radar's scanning pattern is necessary before understanding the data structure of the radar. Weather radar usually uses the PPI (Plan Position Indicator) scanning method.

The PPI scanning principle is based on the rotation of the radar antenna, which sends out a beam of radio waves at a specific frequency and polarization. As the antenna rotates, it scans a complete 360-degree circle around the radar site. The reflected signal from the target is then received by the antenna and processed by the radar receiver. The processing includes amplification, filtering, and demodulation of the signal to extract information about the location, intensity, and velocity of the targets.

The PPI scan mode can be used to provide a two-dimensional display of the radar data. The range of the radar is displayed on the horizontal axis, while the azimuth angle of the radar is displayed on the vertical axis. The radar data is displayed in color, with different colors representing different intensities of the reflected signal.

To create a PPI scan, the radar antenna is tilted at a specific elevation angle, and the antenna rotates at a constant speed around the radar site. The radar signal is transmitted at a fixed pulse repetition frequency (PRF) and pulse width, and the reflected signal is collected by the antenna at a specific time delay after the transmission. The received signal is then processed to determine the location and intensity of the target. Following the execution of a single-layer elevation scan, the weather radar antenna is elevated to the next predetermined elevation angle using the corresponding body sweep mode. This process is iterated until all the prescribed elevation angles have been scanned. The result of a completed scan is a complete set of meteorological data for various elevation angles. In three-dimensional space, this data forms a conical surface with varying inclination angles, centered around the radar station. This conical surface represents the three-dimensional space that is being observed by the radar. This technique is known as volume scanning and is

commonly used in weather radar to provide information about the vertical structure of the atmosphere [22, 23].

Another commonly used scanning method is Volume Coverage Pattern (VCP) scanning. This scanning method allows the radar to scan at different elevation angles and scan rates within a certain altitude range, to obtain higher resolution and more comprehensive weather information.

Weather radar uses various VCPs, including long-range and short-range scanning. Long-range scanning is usually used to detect weather systems such as thunderstorms and hurricanes, while short-range scanning is usually used to detect precipitation and low clouds. In VCP scanning, the radar scans at different height levels and integrates the data into a three-dimensional volume image. These images can help meteorologists better understand the structure and evolution of weather systems and provide more accurate predictions and alerts.

In addition to the conventional and VCP scans, weather radar also supports other scanning modes, such as the tracking mode and the isolated cell scan mode. The tracking mode is used to track a specific weather phenomenon, such as a tornado or a thunderstorm, by focusing the radar on the area where the phenomenon is located. The isolated cell scan mode is used to detect isolated storms that are not part of a larger weather system.

All of these scanning modes and patterns are controlled by the radar's software and can be customized based on the needs of the user. Meteorologists can select the appropriate scanning mode and pattern to obtain the most accurate and useful weather data.

The processed radar signal will contain the following main radar data types.

- Reflectivity data is one of the most fundamental and commonly used data types in weather radar. It refers to the amount of power reflected back to the radar antenna by precipitation or other atmospheric targets. The reflectivity data is obtained by measuring the strength of the returned radar signal, which is proportional to the size and number of the scattering particles in the atmosphere. The units of reflectivity are expressed in decibels of Z (dBZ), which is a logarithmic scale used to represent the intensity of the reflected radar signal. To calculate the reflectivity data, the radar system sends out a pulse of electromagnetic energy, which

travels through the atmosphere until it encounters particles such as raindrops, snowflakes, hailstones, or dust. When the energy meets these particles, it scatters in all directions. Some of the scattered energy returns to the radar antenna, where it is detected and measured. The strength of the returned signal is then converted into a numerical value that represents the reflectivity of the scattering particles. Reflectivity data is useful in determining the intensity and location of precipitation, as well as identifying the type of precipitation, such as rain, snow, or hail. It is also used to detect and track severe weather phenomena, such as thunderstorms, tornadoes, and other hazardous weather conditions.

- Doppler velocity is a type of weather radar data that measures the velocity of precipitation or other targets in the atmosphere relative to the radar itself. It is calculated by measuring the frequency shift of the radar signal due to the Doppler effect. The Doppler effect is the change in frequency of a wave (in this case, a radar signal) as it is reflected off a moving object. If the object is moving toward the radar, the frequency of the reflected signal will be higher than the transmitted signal, and if the object is moving away from the radar, the frequency will be lower. This frequency shift is proportional to the velocity of the object. In weather radar, the Doppler velocity data can be used to detect the motion of precipitation, such as raindrops, hailstones, and snowflakes. By measuring the Doppler velocity of these targets, meteorologists can determine the direction and speed of the precipitation. The calculation of Doppler velocity data involves comparing the frequency of the transmitted radar signal with the frequency of the reflected signal. The difference in frequency, or Doppler shift, is then converted into a velocity value.

- Spectrum width is a weather radar variable that represents the width of the spectrum of the returned signals. It is a measure of the variability in the radial velocities of the scatterers within a resolution volume of the radar. Spectrum width is calculated using the power spectrum of the received signals from each range gate of the radar. The power spectrum is obtained by applying a Fast Fourier Transform (FFT) to the time series data. The resulting power spectrum is a plot of the signal power as a function of frequency. The spectrum

width is then calculated as the second moment of the power spectrum, which represents the spread of the power spectrum about its mean. The second moment is calculated by weighting each frequency by its power and dividing by the total power of the spectrum. The result is a measure of the spread of the frequency spectrum of the radar signal, which is related to the variability in the radial velocities of the scatterers within the resolution volume. Spectrum width is useful for identifying regions of turbulence and microphysical processes in the atmosphere, such as areas of turbulence associated with convective storms and regions of precipitation with large raindrops or ice particles.

- Differential reflectivity (ZDR) is another type of data that can be obtained from weather radar. It is a measure of the difference in reflectivity between horizontally and vertically polarized electromagnetic waves. The calculation of ZDR involves comparing the power received by the radar antenna from the horizontally polarized wave to the power received from the vertically polarized wave. The ratio of these powers is known as the differential reflectivity, which is expressed in decibels. ZDR is particularly useful for distinguishing between different types of precipitation, as different types of particles can cause variations in the polarization of the radar signal. For example, raindrops tend to have a more uniform polarization, while ice particles such as hail or graupel can cause a greater difference in the power received by the horizontally and vertically polarized waves.ZDR data can also be used to estimate the size and shape of the particles in the atmosphere, which can be useful for predicting the type and intensity of the precipitation. For example, a low ZDR value may indicate the presence of large, irregularly shaped particles such as hail, while a high ZDR value may indicate the presence of small, spherical particles such as raindrops.

- Specific differential phase (KDP) is a type of weather radar data that measures the difference in phase shift between two radar pulses with different polarizations as they pass through precipitation. The KDP data is used to calculate the differential phase shift, which can then be used to estimate the specific attenuation of the radar signal due to the precipitation. This information is particularly useful for correcting for the effects of attenuation, which

can distort radar data and make it difficult to accurately measure precipitation rates. KDP is calculated by measuring the difference in phase shift between the horizontally-polarized and vertically-polarized radar signals as they pass through the precipitation. The specific differential phase shift (KDP) is then calculated by dividing the differential phase shift by the distance that the radar signal traveled through the precipitation. This value can then be used to estimate the specific attenuation of the radar signal due to the precipitation. The calculation of KDP involves first measuring the differential phase shift (ϕDP) between the transmitted and received radar signals. This can be affected by factors such as the type and size of the precipitation particles, the orientation of the particles in the radar beam, and the frequency and polarization of the radar signal. Once ϕDP is measured, it is then differentiated with respect to the range to obtain the specific differential phase (KDP). This calculation corrects for the effects of differential attenuation, which can occur when the radar signal passes through regions of varying precipitation intensity. KDP data is particularly useful for correcting radar data for the attenuation caused by heavy precipitation, such as rain, snow, and hail. It can also be used to estimate the size and shape of precipitation particles, as well as to help distinguish between different types of precipitation, such as rain and snow. KDP data is typically displayed in units of degrees per kilometer or degrees per mile and can be visualized as color-coded maps or graphs.

At this juncture, the three most frequently employed data types, namely reflectivity, Doppler velocity, and spectrum width, will be chosen for further experimental data observations in this thesis. These examples are illustrated in Figure 2.3.

2.4 Low-quality Data and Beam Blockage

2.4.1 Low-quality Data

The quality of radar signals can often be adversely affected by physical occlusion or signal interference, which can be caused by various factors such as precipitation, temperature, humidity, and other atmospheric conditions that impact the attenuation of the electromagnetic wave signal.

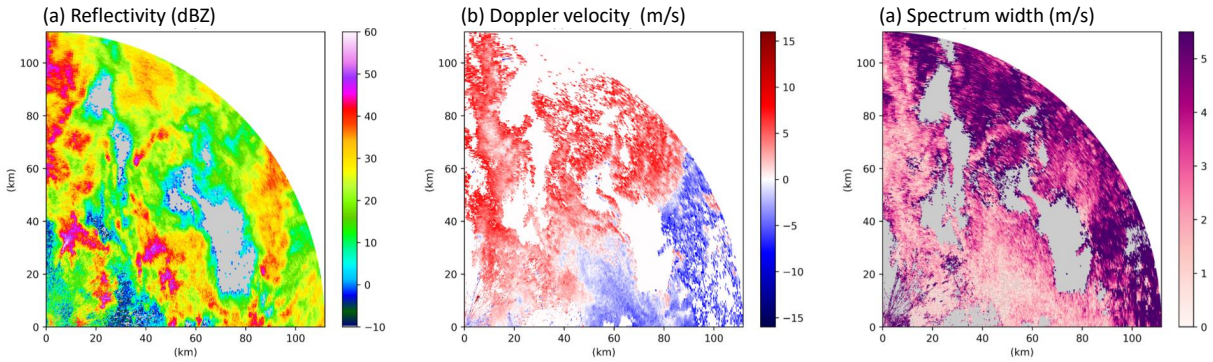

Figure 2.3: Example of the KFWS radar observations at 0357UTC, March 1, 2018: (a) reflectivity measurement; (b) Doppler velocity; (c) spectrum width.

These factors may lead to the scattering or absorption of different wavelengths of the signal, which can result in low-quality or incomplete data.

Considering the radar signal itself, one of the main reasons for the formation of low-quality radar data is signal attenuation caused by the absorption or scattering of radar signals as they travel through the atmosphere. The level of attenuation varies depending on the electromagnetic wave's wavelength and the presence of atmospheric elements like precipitation, temperature, and humidity. As a result, the radar signal's strength can decrease, resulting in missing or low-quality data. This phenomenon is particularly significant for high-frequency radar systems that detect smaller precipitation particles that are more prone to absorption and scattering. Additionally, natural and man-made features such as hills and mountains can also cause radar signal attenuation, leading to low-quality data.

There is also low-quality data that can come from the processing system of the radar. For example, if the radar signal processing algorithms are not well-designed or implemented, they can introduce errors into the radar data. Moreover, the quality of the radar data may be affected by the dynamic range of the receiver and the quantization noise of the analog-to-digital converter. Additionally, the data may be impacted by range aliasing, where targets that are beyond the maximum unambiguous range appear as false targets at shorter ranges, or velocity folding, where Doppler velocities that exceed the maximum unambiguous velocity are wrapped around and appear as false velocities at lower speeds. The presence of calibration errors is another factor contributing to the

formation of low-quality radar data. Calibration is crucial for obtaining accurate and dependable radar data. Nonetheless, calibration errors can emerge due to several factors, such as environmental conditions, malfunctioning equipment, and inappropriate installation. These errors can cause imprecise radar measurements, leading to incorrect interpretations of weather situations, such as misinterpreting reflectivity values. Additionally, the accuracy of other radar parameters, like radial velocity and spectrum width, can also be influenced by calibration errors. Consequently, regular maintenance and calibration checks are crucial to reduce errors and uphold the quality of radar data.

Regarding physical blockage, low-quality radar data can arise from various sources, including ground clutter, which pertains to radar signals reflected from immobile ground-based objects such as mountains, buildings, or other landforms, instead of atmospheric targets like precipitation or clouds. Ground clutter can result in poor-quality radar data that can obscure or distort actual meteorological targets, particularly in regions with intricate terrain or close to large urban areas with many buildings. Furthermore, the radar beam can reflect from ground clutter and return to the radar receiver, leading to false echoes or "ghost" signals. To lessen the impact of ground clutter, advanced filtering algorithms are usually employed by weather radar systems that aim to remove these unwanted signals from the data. Nevertheless, these algorithms are not infallible and can sometimes eliminate real meteorological targets. Low-quality radar data can also be attributed to anomalous propagation and non-meteorological targets. Anomalous propagation occurs when the radar beam refracts in the atmosphere, causing it to deviate from its intended path and follow the curvature of the Earth's surface, thus resulting in stationary false echoes that do not vary with time. Non-meteorological targets, including birds, insects, buildings, and mountains, can similarly cause misleading data by reflecting, refracting, or diffracting the radar signal. Such interference with the radar signal can result in false echoes and compromise the quality of radar data.

Beam blockage is the most prevalent issue that can degrade the quality of radar data, occurring when the radar signal encounters obstructions, such as high-rise buildings, mountains, or trees, before reaching the target. This can cause gaps or missing data in the radar coverage, resulting in

a reduction of signal intensity and distorted beam shapes, leading to errors in estimating reflectivity, velocity, and other parameters, especially in the vicinity of the obstruction. Beam blockage can cause underestimation or omission of precipitation or other weather phenomena, leading to incorrect forecasts and potential hazards. To mitigate the impact of beam blockage on radar data quality, radar installations are generally sited in areas with fewer obstructions, such as hilltops or open fields. Weather radar systems are also equipped with multiple antennas at different elevations to provide more comprehensive coverage and reduce the effects of beam blockage. Moreover, radar operators can adjust the tilt angle of the antenna or use advanced algorithms to estimate missing data.

2.4.2 Beam Blockage Correction

Currently, various methods exist to address weather radar beam blockage, which can be broadly categorized into hardware adjustments, data stage adjustments, and post-processing techniques [24].

One common solution for beam blockage is to install additional radar sites to provide better coverage and redundancy. This approach can help to fill in gaps in coverage caused by natural obstacles or terrain, and can also provide multiple views of a given storm system for improved accuracy and reliability. Additionally, modern radar systems often incorporate advanced technologies like dual-polarization and phased-array antennas, which can help to mitigate the effects of beam blockage and improve data quality overall. Other solutions for beam blockage may include the use of lower frequencies or the deployment of mobile radar systems, which can be moved to different locations as needed to provide better coverage. Employing data assimilation techniques can also be a solution for beam blockage. Data assimilation refers to the process of combining multiple sources of data to generate an optimal estimate of the current state of the atmosphere. This technique can be used to fill in gaps in the radar data caused by beam blockage. By assimilating data from other sources, such as weather models, satellites, and ground-based observations, it is possible to generate a more complete picture of the current weather conditions and make up for

missing radar data. One example of data assimilation techniques is the use of radar data assimilation in numerical weather prediction models. This involves integrating radar data into the model's initial conditions, which can improve the accuracy of short-term forecasts. Another example is the use of machine learning techniques to predict missing radar data based on available data from other sources.

Another solution to beam blockage is to increase the height of the radar antenna. This can help to reduce the impact of obstacles and ensure that the radar beam can reach the desired target. However, increasing the antenna height can also be expensive and may not be possible in some areas due to structural limitations or regulatory restrictions. Radar signals can be affected by different atmospheric conditions, and some frequencies may be more prone to attenuation or blockage than others. Using alternative radar frequencies, such as X-band or Ka-band, can help to overcome the issue of beam blockage [25]. However, this approach can also be expensive and may require additional hardware upgrades.

A common solution at the data level is to apply filtering techniques. Filtering techniques are used to remove or reduce the effects of beam blockage on the data. There are several types of filtering techniques that can be used.

- Spatial Filtering: Spatial filtering is a technique used to remove beam blockage in the data by applying a spatial filter to the data. Spatial filtering involves averaging the data in a local region around the blockage. This helps to reduce the effects of the blockage on the data. However, this technique may lead to loss of resolution in the data [7] [26].

- Temporal Filtering: Temporal filtering is a technique used to remove beam blockage in the data by applying a temporal filter to the data. Temporal filtering involves averaging the data over a period of time. This helps to reduce the effects of the blockage on the data. However, this technique may lead to loss of temporal resolution in the data.

- Adaptive Filtering: Adaptive filtering is a technique used to remove beam blockage in the data by adapting the filter to the data. Adaptive filtering involves estimating the parameters

of the filter based on the characteristics of the data. This helps to reduce the effects of the blockage on the data while preserving the resolution of the data.

- Wavelet Transform Filtering: Wavelet transform filtering is a technique used to remove beam blockage in the data by applying a wavelet transform to the data. Wavelet transform filtering involves decomposing the data into different frequency bands using the wavelet transform. The beam blockage is then removed by applying a filter to the decomposed data.

- Kalman Filtering: Kalman filtering is a technique used to remove beam blockage in the data by modeling the data as a dynamic system. Kalman filtering involves estimating the state of the system based on the data and using this estimate to remove the effects of the blockage on the data.

Algorithm post-processing solutions for beam blockage typically involve using statistical or mathematical techniques to estimate the missing data caused by the blockage. These techniques include interpolation, extrapolation, and data deep learning techniques.

Interpolation involves using known data points to estimate the values of missing points. For example, if a radar beam is blocked at a certain elevation angle, data from neighboring elevation angles can be used to estimate the missing data. Linear interpolation is a common technique, but more advanced methods such as cubic spline interpolation can also be used. Extrapolation involves using known data points to estimate the values of points outside the range of the known data. This technique can be useful in situations where the blockage is partial and data from neighboring areas can be used to estimate the missing data. Deep learning techniques can be used to develop algorithms that can adapt to different weather patterns and overcome the issue of beam blockage. Machine learning algorithms can be trained on large datasets of radar data and can learn to recognize patterns and predict weather patterns accurately.

2.5 Conclusion

This chapter provides an overview of radar technology, covering its fundamental operating principles and various types of weather data that can be obtained using radar, such as reflectivity, Doppler velocity, spectrum width, differential reflectivity, and KDP. The Radar model we used is based on these data types. The chapter also discusses potential sources of error in radar data and factors that can lead to low-quality data, including signal attenuation, ground object echoes, anomalous propagation and non-meteorological targets, signal processing, calibration, and beam blocking. These errors can cause the degradation of radar data and require correction. When repairing radar data, several common methods can be used, such as interpolation using nearby data, correction based on physical models, and data imputation and correction using machine learning algorithms. Moving forward to the following chapter, our analysis will persist, and we will further investigate the data utilized in the model by categorizing it according to the type of precipitation.

Chapter 3

Study Domains and Datasets

3.1 Introduction

Precipitation is a vital aspect of the Earth's water cycle, which involves the continuous movement of water between the atmosphere, land, and oceans. It plays a crucial role in maintaining the balance of water resources on our planet. Precipitation occurs when moisture in the atmosphere condenses into droplets or ice crystals that become heavy enough to fall back to the Earth's surface. This process is driven by a variety of factors, including temperature, humidity, and atmospheric pressure.

Precipitation can take on various forms, depending on the temperature and other atmospheric conditions. Rain is the most common form of precipitation, and it occurs when liquid water droplets fall to the Earth's surface. Snow is another type of precipitation, which happens when atmospheric moisture freezes into ice crystals before falling to the ground. Sleet is a mixture of rain and snow, while hail is precipitation in the form of ice pellets that form when supercooled water droplets freeze onto a nucleus. The characteristics of precipitation can vary widely, including its intensity, duration, and spatial distribution. Some precipitation events are short-lived and relatively light, while others can be intense and long-lasting, leading to flooding and other hazards [27]. The spatial distribution of precipitation can also vary greatly, with some regions receiving much more precipitation than others due to factors such as geography, wind patterns, and climate. Two primary types of precipitation are convective and stratiform precipitation [28]. Convective precipitation occurs when warm, moist air rises rapidly, leading to the formation of towering cumulus clouds that can produce heavy rain, thunderstorms, and other severe weather events. Convective precipitation is typically short-lived but can be intense, and it often occurs in the afternoon and evening in warm and humid regions. Stratiform precipitation, on the other hand, is associated with large-scale weather systems, such as frontal boundaries or low-pressure

systems. Stratiform precipitation is characterized by a more prolonged and uniform distribution of precipitation over a broader area and is more common in regions with a humid continental climate. Understanding precipitation patterns and processes are crucial for many applications, including agriculture, water resource management, weather forecasting, and climate modeling. Accurate precipitation measurements are essential for assessing water availability, predicting crop yields, and managing water resources for human consumption and industrial uses. Precipitation data also plays a vital role in weather forecasting by providing information on the intensity, duration, and spatial distribution of precipitation events, which can help forecasters issue warnings and alerts to the public.

In the upcoming chapter, we will delve deeper into the analysis of radar data used in our model, with a focus on categorizing the data based on different precipitation types. Our analysis will involve examining the radar data obtained from various precipitation areas and identifying patterns and trends specific to each type of precipitation. Once we have gathered and categorized the data, we will proceed to preprocess it, ensuring that it is cleaned and formatted correctly for input into the network layers. This process will allow us to gain a more comprehensive understanding of the underlying structure of the precipitation data and its impact on our model's accuracy and performance.

3.2 Precipitation

3.2.1 Convective Precipitation

Convective precipitation is a type of precipitation that forms due to the rising of moist and unstable air in the atmosphere. This type of precipitation is known for its high intensity and short duration, often causing severe weather conditions such as thunderstorms, lightning, and strong winds. In this essay, we will analyze the formation, state, and impact of convective precipitation from multiple perspectives such as geography, atmosphere, and radar.

From a geographical perspective, convective precipitation is more likely to occur in regions with high temperature and humidity, such as tropical and subtropical areas. These areas are char-

acterized by a high amount of solar radiation, which heats up the surface of the Earth, causing the air to rise and form convective clouds. As these clouds rise, they cool down, and their moisture condenses, leading to the formation of precipitation. In addition, convective precipitation is more likely to occur in areas with complex terrains, such as mountains, where the upward movement of the air is enhanced by the topography. Topography plays a significant role in the formation of convective precipitation, particularly in areas with steep terrain. As moist air is forced to rise over mountains, it cools and condenses, leading to the formation of clouds and subsequent precipitation. This process, known as orographic lifting, can result in heavy rainfall and flash flooding in areas downstream of the mountains. In addition to orographic lifting, convective precipitation can also be influenced by the interaction between land and sea breezes, which can lead to the formation of thunderstorms and other severe weather events along coastlines. Land use can also impact the formation of convective precipitation, particularly in urban areas. Cities with large amounts of impervious surfaces, such as concrete and asphalt, can experience increased runoff and decreased infiltration, leading to more frequent and intense flash flooding during convective precipitation events. Urban areas can also experience what is known as the urban heat island effect, where the concentration of buildings and other infrastructure can result in higher temperatures compared to surrounding rural areas [29]. This can lead to increased instability in the atmosphere, making it more favorable for the formation of convective precipitation. Finally, proximity to large bodies of water can also impact the formation of convective precipitation. Warm, moist air masses that form over the ocean can be transported inland, where they can collide with cooler, drier air masses and create the conditions necessary for the formation of thunderstorms and other severe weather events.

Convective precipitation can be analyzed from an atmospheric perspective to gain a better understanding of its formation and dynamics. Convective precipitation is formed as a result of atmospheric instability caused by the movement of warm, moist air rising into cooler regions of the atmosphere. This instability can be caused by a variety of factors, including solar heating, fronts, and other weather systems. Atmospheric instability results in the formation of clouds, which are

the precursors to convective precipitation. The type of cloud that forms depends on the temperature and moisture content of the air, as well as the height at which the instability occurs. Cumulus clouds are a common precursor to convective precipitation and are formed by warm air rising and cooling, causing moisture to condense and form visible cloud formations. Once a cumulus cloud has formed, it can continue to grow and develop into a convective storm. This process is fueled by the release of latent heat as water vapor condenses into liquid water or ice within the cloud. This release of heat further destabilizes the atmosphere, leading to more rapid ascent of warm, moist air and the formation of even stronger updrafts. The updrafts within convective storms can reach speeds of over 100 miles per hour, carrying moisture and air aloft to heights of several miles. As the moisture rises, it cools and condenses into droplets or ice particles, which grow in size as they collide with other particles in the cloud. Once these particles become too heavy to be supported by the updrafts, they begin to fall to the ground, resulting in convective precipitation. The intensity and duration of convective precipitation can vary widely depending on a variety of atmospheric factors, including the moisture content and temperature of the air, the strength of the updrafts, and the presence of other weather systems. In some cases, convective precipitation can be extremely intense, resulting in flash floods, landslides, and other hazards [30].

Convective precipitation can have significant impacts on both natural and human systems. The intensity, duration, and spatial distribution of convective precipitation can vary greatly, leading to a range of effects on ecosystems, infrastructure, and human health and safety. One of the most immediate impacts of convective precipitation is the potential for flooding. Heavy rainfall from convective storms can quickly overwhelm drainage systems and lead to flash flooding in urban and rural areas. This can cause significant damage to buildings, roads, and other infrastructure, as well as threaten human safety. In addition to flash flooding, convective precipitation can also contribute to river and coastal flooding, leading to longer-term impacts on communities and ecosystems. Convective precipitation can also have significant effects on agriculture and natural ecosystems. While moderate amounts of rainfall can be beneficial for plant growth, excessive or poorly timed rainfall can lead to soil erosion, reduced crop yields, and damage to natural habitats. On the

other hand, prolonged drought conditions caused by the absence of convective precipitation can have similar negative effects on agriculture and ecosystems. In addition to its physical impacts, convective precipitation can also affect human health and safety. Lightning strikes, which often accompany convective storms, can pose a direct threat to individuals outdoors. High winds and hail associated with convective precipitation can also cause damage and pose safety risks. Climate change is expected to impact the frequency and intensity of convective precipitation events, with some studies suggesting an increase in extreme precipitation events. This could have significant implications for the impacts of convective precipitation on natural and human systems, particularly in areas that are already prone to flooding or other weather-related hazards [31].

Therefore, given the significant impacts of convective precipitation on various sectors, including agriculture, water resources, transportation, and public safety, an accurate and timely forecast of convective precipitation is crucial. In this regard, radar technology plays a critical role in providing high-resolution and real-time information on the state and dynamics of convective precipitation. The ability to detect and track the movement of convective precipitation systems is particularly important for predicting the timing, location, and intensity of associated hazards, such as flash floods, landslides, and lightning strikes.

Convective precipitation can be analyzed from a radar perspective, which is crucial in providing accurate forecasts and early warning systems for severe weather events. In this context, radar is an important tool for measuring and tracking convective precipitation.

Convective precipitation displays certain characteristics on radar images that allow meteorologists to identify and track them. One of the most prominent features of convective precipitation on radar images is the presence of reflectivity echoes. These echoes are generated when precipitation particles interact with the radar beam and reflect some of the energy back to the radar receiver. Convective precipitation is characterized by high reflectivity values, indicating a high concentration of precipitation particles within the cloud. Another characteristic of convective precipitation on radar images is the presence of vertically developed clouds. These clouds are often referred to as cumulonimbus clouds and are associated with thunderstorms. On radar images, these clouds

appear as tall, vertical structures with a distinctive anvil-shaped top. Convective cells tend to be more intense, with stronger updrafts and downdrafts, which result in a more pronounced radar signature. The rapid and intense vertical motion of air masses associated with convective precipitation creates a significant amount of turbulence. The vertical development of these clouds is a result of the strong updrafts and downdrafts that occur within convective systems.

The presentation of convective precipitation on radar displays can also be challenging due to the high variability of the precipitation's spatial and temporal distribution. Convective cells tend to be highly localized and often display rapid growth and decay rates, which can make it difficult to track their movement and evolution over time. Moreover, the highly variable nature of convective precipitation often results in the formation of complex structures, such as mesoscale convective systems and supercells, which can present additional challenges for radar data interpretation and analysis.

To accurately detect and forecast convective precipitation events, radar meteorologists use a range of different techniques and algorithms [32]. One widely used approach is to apply a threshold-based approach, which involves setting a specific reflectivity value as a threshold for convective precipitation detection. This approach can be effective in detecting convective cells, but it can also result in false alarms, particularly in regions where there is a significant amount of non-precipitating clutter in the radar signal. Another approach is to use dual-polarization radar, which provides additional information on the shape and size of the precipitation particles. This information can help distinguish between different types of precipitation, such as rain, snow, and hail, and provide more accurate estimates of precipitation intensity and accumulation. Moreover, dual-polarization radar can also help identify regions of high turbulence associated with convective precipitation, which can be useful for aviation safety and other applications. Furthermore, radar meteorologists also use a range of different algorithms and techniques to analyze the movement and evolution of convective cells over time. For example, the velocity of the radar echoes can be used to estimate the speed and direction of the convective cells, which can be used to track their movement and predict their future path. Other techniques, such as the correlation coefficient, can

be used to identify regions of the radar signal that display high variability and are likely to contain regions of turbulence associated with convective precipitation.

3.2.2 Stratiform Precipitation

Stratiform precipitation, also known as non-convective precipitation, is a type of precipitation that is associated with relatively widespread, gentle lifting of air. This type of precipitation is commonly associated with mid-latitude cyclones, frontal systems, and other synoptic-scale weather phenomena.

One of the key factors that influence the formation of stratiform precipitation is the topography of the region. When warm, moist air is forced to rise over mountains or hills, it cools and condenses, forming clouds and ultimately precipitation. This process is known as orographic lifting, and it is a common mechanism for producing stratiform precipitation in mountainous regions. For example, the Pacific Northwest region of the United States is known for its frequent stratiform precipitation events due to the influence of the Cascade Range. In addition to topography, the proximity to bodies of water also plays a role in the formation of stratiform precipitation. Regions near coastlines often experience stratiform precipitation due to the presence of moist ocean air masses. This is particularly true for regions with prevailing westerly winds, such as the West Coast of the United States and Western Europe [33]. Another important geographical factor that can influence the formation of stratiform precipitation is the location of the precipitation zone relative to the equator. In general, areas near the equator receive more convective precipitation, while areas farther away from the equator are more likely to experience stratiform precipitation. This is because the warm, moist air near the equator is more unstable, and is more likely to form thunderstorms and other convective systems. In addition to these factors, the location and characteristics of large-scale weather patterns such as jet streams and high-pressure systems can also influence the formation of stratiform precipitation. Jet streams, for example, can steer moist air masses into regions where they can produce prolonged periods of precipitation, while high-pressure systems can create stable atmospheric conditions that promote the formation of stratiform precipitation.

From an atmospheric perspective, stratiform precipitation is typically associated with large-scale ascent over a broad area, which causes a gradual lifting of the air mass and a long duration of precipitation. The precipitation rate is usually light to moderate, but can persist for several hours or even days. The formation of stratiform precipitation is often associated with the presence of a mid-level cloud deck, such as altocumulus or altostratus clouds, which are characterized by a sheet-like appearance and a relatively uniform thickness. These clouds can develop in regions where warm, moist air is lifted to a level where it becomes saturated and condenses into clouds. As the air continues to rise, the water droplets grow larger and eventually fall to the ground as precipitation.

One of the most significant impacts of stratiform precipitation is its role in replenishing freshwater resources. In areas where rainfall is infrequent or unpredictable, stratiform precipitation can be particularly important in sustaining agriculture and human populations. In addition to its importance for water resources, stratiform precipitation can also have important ecological impacts, particularly in forested regions where precipitation plays a critical role in sustaining plant growth and maintaining biodiversity. However, excessive stratiform precipitation can also lead to flooding and other types of water-related disasters. In areas with inadequate drainage systems, prolonged periods of rainfall can lead to water accumulation and flooding of homes, businesses, and infrastructure. Heavy rainfall can also lead to soil erosion, landslides, and other types of environmental damage.

From the perspective of radar, stratiform precipitation presents unique challenges and opportunities for analysis. One of the key features of stratiform precipitation is its uniformity and gradual intensity changes. This makes it more difficult to distinguish different precipitation rates from the radar reflectivity data, as there are fewer distinct features to identify. However, this uniformity also presents opportunities for more accurate quantitative precipitation estimation (QPE), as the rain rate can be assumed to be relatively constant across the precipitation area [34]. To analyze stratiform precipitation using radar data, a number of techniques have been developed. One common approach is to use dual-polarization radar, which can provide additional information on the

shape and size of the precipitation particles. This can help to distinguish between different types of precipitation, such as rain, snow, and hail, and can also provide more accurate estimates of precipitation rates. Another approach is to use high-resolution radar data to identify and track individual precipitation cells within the stratiform precipitation area.

3.3 Study Domains

By reviewing the two types of typical precipitation, we can find that stratiform and convective precipitation have distinct characteristics. Stratiform precipitation usually forms in widespread regions with light to moderate intensity and tends to last longer. It is associated with large-scale weather systems and is often observed ahead of frontal systems or within areas of low pressure. In contrast, convective precipitation tends to be more localized, intense, and of shorter duration. It often forms in association with thunderstorms or other types of atmospheric instability, and its intensity and duration are highly variable.

When it comes to radar data correction, both types of precipitation need to be considered simultaneously. Stratiform precipitation usually produces less attenuation and is less affected by beam blockage, which can be a problem in mountainous areas. Therefore, stratiform precipitation can be used to calibrate and adjust the radar data in these areas. On the other hand, convective precipitation often produces stronger returns and is more prone to attenuation and beam blockage. Therefore, convective precipitation can be used to verify the calibration and adjustments made using stratiform precipitation.

By using both stratiform and convective precipitation models in cross-training and testing, we can take advantage of the distinct characteristics of each precipitation type and improve the accuracy of radar data restoration. Additionally, this approach can help to address some of the limitations and uncertainties that arise when using only one precipitation type. By incorporating both types of precipitation into the restoration process, we can increase the robustness of the resulting data, which is important for a wide range of applications, including weather forecasting, hydrological modeling, and climate research.

Therefore, study domains characterized by different precipitation features, i.e., convective and stratiform precipitation, will be selected so as to enrich the radar data characteristics and enhance the model inpainting capability. This article selects two distinct regions as study domains. The first of these is Northern California, characterized by a Mediterranean climate featuring damp winters and parched summers. The primary source of precipitation in this region is derived from Pacific Ocean storms, which frequently generate stratiform precipitation that is evenly dispersed across a broad area [35]. In contrast, northern Texas experiences a humid subtropical climate characterized by hot summers and mild winters. This arid climate fosters the development of more powerful, localized storms that generate typical convective precipitation, which may result in thunderstorms and heavy downpours [4] [5]. Two National Weather Service (NWS) Weather Surveillance Radar—1988 Doppler (WSR-88D) radar systems were chosen within the selected study domains, namely, KDAX radar deployed in Davis near Sacramento in Northern California, and KFWS radar deployed in the Dallas-Fort Worth area in northern Texas.

The Weather Surveillance Radar-1988 Doppler (WSR-88D) is a network of radar systems operated by the National Weather Service (NWS) in the United States. It is a highly advanced weather radar system designed for monitoring and forecasting severe weather events. The WSR-88D system is composed of a radar antenna, a transmitter, and a receiver that work together to detect precipitation and atmospheric phenomena, including wind patterns and turbulence. The WSR-88D system is capable of scanning the atmosphere at various elevations, allowing meteorologists to obtain a three-dimensional view of the atmosphere. It operates at S-band frequency, which provides high spatial resolution and sensitivity to small particles such as raindrops and hailstones. The radar data can be processed and analyzed in real time, providing up-to-date weather information for forecasters and emergency management personnel. The primary function of the WSR-88D is to detect and track severe weather events, including thunderstorms, tornadoes, and hurricanes. The Doppler capabilities of the radar system also allow meteorologists to measure the velocity and direction of the wind, providing valuable information about the structure and intensity of the storms. In addition to its severe weather monitoring capabilities, the WSR-88D is also used for aviation

weather surveillance and provides valuable information for air traffic control. The radar system can detect and track aircraft within its range, providing pilots with information about weather conditions in their vicinity. Since its introduction in the 1990s, the WSR-88D has revolutionized weather forecasting and emergency management in the United States. It has provided valuable data for research into severe weather phenomena, helping to improve our understanding of these events and ultimately saving lives through improved warning systems.

Both KDAX and KFWS are Doppler weather radars that can detect the velocity and direction of weather objects, including the location and intensity of the precipitation. Their detection radius spans approximately 230 kilometers, and the range and azimuth angle resolution are 0.25 km and 0.5 degrees, respectively, resulting in a data matrix with a size of 920×720 for a full surveillance scan is shown in Figure 3.1 (a).

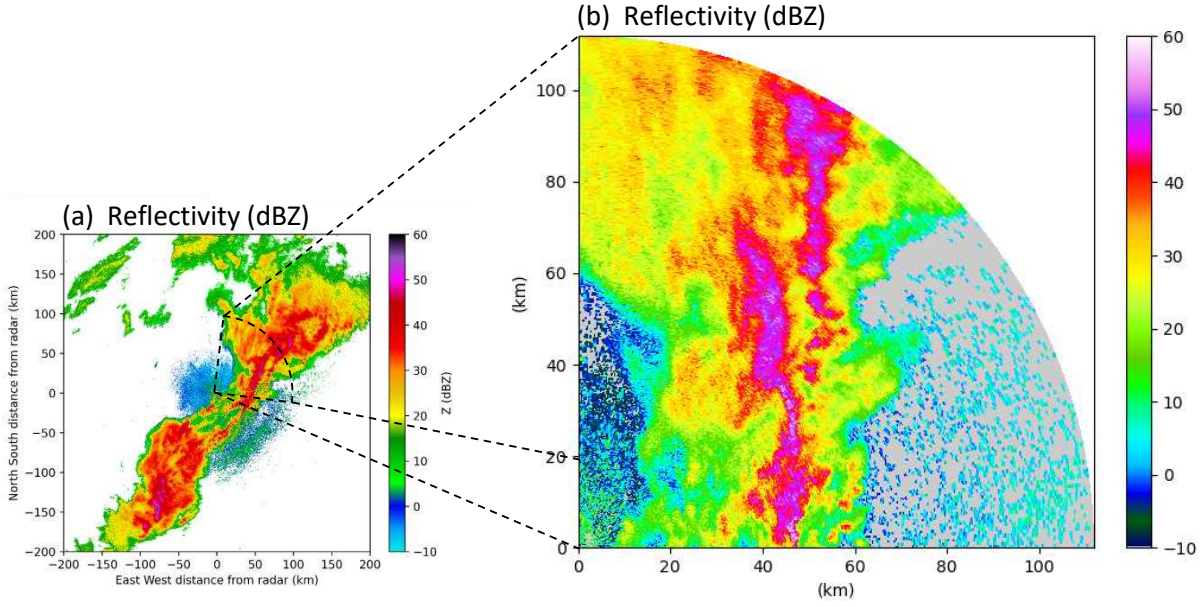

Figure 3.1: Example of the KFWS radar observations at 0555UTC, March 1, 2018: (a) original reflectivity measurement; (b) preprocessed reflectivity for a selected subdomain.

It has been determined that the vast majority of deep learning algorithms demand a substantial quantity of data for practical training, coupled with the high-resolution nature of radar imagery that necessitates significant computational resources and time for complete processing, and the

challenging task of collecting measurement data free of noise and interference, we have opted to preprocess the raw data. Initially, to incorporate more comprehensive precipitation information, we utilized the KFWS and KDAX radar data from May to September during 2019-2021 as the training set [36]. During this stage, high precipitation information exceeding 25 dBZ, with a continuous length of 8 km or more and within a 15-degree range, would be selected for further analysis. Subsequently, to minimize data uncertainty and blank or low-quality data included, we have chosen to crop the original data size to a new subdomain with the size of 448 (range) $\times$ 192 (azimuth), as depicted in Figure 3.1(a), which has been sized for facilitating the convolution operation at a later stage. To enhance the model's robustness, we have rotated and flipped the selected subdomain within the same precipitation event to create pseudo radar observations. Furthermore, in order to focus the algorithm more intently on important data features, we assigned that the probability distribution characterizing the location of the selection center corresponds to the magnitude distribution of the reflectivity data contained within the precipitation information. After preprocessing, the complete dataset will be arranged in descending order according to the cumulative reflectivity information. We extracted the first 10,000 cases of data from each of the two radars from the preprocessed dataset as the final dataset. The first 80% of this data will be allocated for training purposes, and the remaining 20% will be used for validation.

3.4 Normalization

To expedite the convergence of the training data, decrease the possibility of overfitting, and simplify the simulation of PBB masks in the precipitation data, we normalized the preprocessed precipitation data to conform to the target range of -1 to 1 using the following normalization equation [37].

$$x = \frac{1.5 \times (x - \min)}{\max - \min} - 0.5 \tag{3.1a}$$

$$x[x <= -0.5] = -1 \tag{3.1b}$$

$$x[x > 1] = 1 \tag{3.1c}$$

Note that the reflectivity values in the raw precipitation data range between -10 and 60 dBZ. However, we have established the minimum and maximum limits (i.e., min and max) in equation 3.1a as 0 and 60 dBZ, respectively. Consequently, reflectivity data are linearly scaled to fall within the range of [-0.5, 1.0], whereas any reflectivity values below 0 dBZ would be normalized to -1 using equation 3.1b, which is because the convolution operation and edge-filling approach tend to smooth out image edges, including cloud edges in radar data, resulting in irregular artifacts in the output feature maps at the edge locations, which can negatively impact the model's training during CNN training [38]. As a result, all precipitation data that falls below -0.5 was treated as clear day data with a value of -1 in equation 3.1b. This 0.5 difference will emphasize the clouds while enabling the model to obtain more effective gradient information at the edge locations. Finally, equation 3.1c applied a truncated normalization process to incorporate the dataset's rare high-reflectivity data outside the normal range.

3.5 Conclusion

This chapter begins with an analysis of two types of precipitation. In conclusion, the selection of precipitation data is critical for accurate radar analysis and forecast models. It is essential to consider the type and intensity of precipitation and the radar's capabilities to detect it. Both stratiform and convective precipitation types play a crucial role in understanding weather patterns and forecasting precipitation events. Therefore, using data from both precipitation types in beam blockage models is necessary for a more comprehensive analysis and forecast. Therefore, we choose the study domains for our model. The process of selecting and normalizing precipitation data involves careful consideration of various factors such as data quality, availability, and spatial and temporal resolution.

To enhance the robustness of the model, we normalized each data layer. Normalization is crucial for accurate beam blockage correction, which is necessary to obtain reliable and accurate radar rainfall estimates.

In the upcoming chapter, we will conduct manual simulations of beam blockage on the normalized data, and then proceed to train the network at the network layer.

Chapter 4

Methodology

4.1 Introduction

Beam blockage is a well-known problem in radar meteorology, which refers to the obstruction of the radar beam by terrain, buildings, trees, or other objects. This obstruction can lead to the loss of data or the introduction of errors into radar data, which can cause significant problems in weather forecasting and monitoring. To address this issue, researchers have developed various methods to correct beam blockage, the first step of which was a manual simulation of the beam blockage situation.

The manual simulation method for beam blockage correction involves simulating the effect of beam blockage on radar data by manually adjusting the data. This approach requires a good understanding of the terrain and the objects that may obstruct the radar beam, as well as knowledge of the radar system itself. The process involves analyzing the radar data, identifying the areas where the blockage occurs, and adjusting the data to correct for the effects of the blockage. Currently, we have clipped and normalized three radar data layers including reflectivity, Doppler velocity, and spectrum width to ensure that they fit into the beam blockage simulation and into the CGAN network combined with Unet++ as a generator.

The U-Net and U-Net++ models are based on the encoder-decoder architecture, which consists of an encoding path that captures the features of the input data and a decoding path that reconstructs the output data based on the learned features. The U-Net++ model is an extension of the U-Net, which utilizes the nested skip connections to improve the segmentation accuracy. The skip connections enable the model to capture the multi-scale features of the input data and enhance the segmentation performance.

The CGAN model is a variant of the generative adversarial network (GAN), which consists of a generator and a discriminator. The generator generates the output data based on the input data

and a random noise signal, while the discriminator discriminates the generated data from the real data. The generator and discriminator are trained in an adversarial manner, where the generator tries to fool the discriminator by generating realistic data, and the discriminator tries to distinguish the generated data from the real data. The CGAN introduces the conditional information into the GAN model, where the generator and discriminator take the conditional information as input to generate and discriminate the data, respectively.

In this chapter, we will introduce the manual simulation method for beam blockage correction and the neural network approach to train the CGAN and Unet++ based network for automatic correction. To train the U-Net++ CGAN model, a large amount of training data is required. The data used for training the model should be preprocessed to remove the beam blockage and normalize the data, as described in the previous sections. The preprocessed data can then be used to train the U-Net++ CGAN model with an appropriate loss function, such as the binary cross-entropy loss or the mean squared error loss. The trained model can then be used to predict the missing data caused by beam blockage in the test data.

4.2 PBB Simulation

This subsection outlines the manual creation of beam blockage conditions for convective and stratiform precipitation data, wherein specific radial profiles are rendered blank. During the preprocessing stage, the image center locations are determined based on the reflectivity scaling probabilities. To create the PBB mask, a specific region in the center of the image ranging from 4-16 degrees azimuth and starting position in the range 0-248 are designated, which allows for a more accurate representation of the terrain and obstacles in the area. This is especially important in areas with complex terrains, such as mountainous regions or urban areas with tall buildings. By incorporating random size blocking, the simulation can create more realistic blockages that better reflect the actual conditions on the ground. This PBB mask region is assigned a value of 1, while the remainder of the data is set to 0. Following this, the normalized data undergo blockage simulation

based on the prescribed equation as shown in 4.1, allowing the neural network to learn and predict precipitation patterns under these simulated blockage conditions.

$$x_{masked} = x \times (1 - mask) - mask \tag{4.1}$$

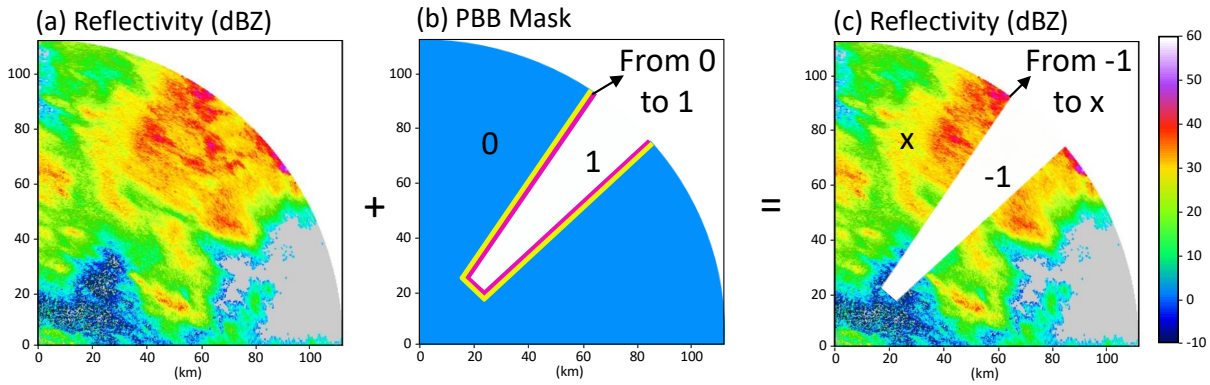

Figure 4.1: Illustration of PBB simulation process: (a) reflectivity data, i.e., x in equation 4.1; (b) PBB mask i.e., $mask$ in equation 4.1; (c) simulated data with beam blockage, i.e., x_{masked} in equation 4.1.

Figure 4.1 illustrates the manual process of simulating beam blockage and the variations in the data boundary conditions. The original data is depicted in Figure 4.1(a), where reflectivity information is used as an example. A randomly sized mask is shown in Figure 4.1(b). As mentioned earlier, the region corrected by the neural network often contains artifacts at the edges. To preserve the details of this portion and prevent unwanted noise, a buffer is added for smoothing with a range value of 1-5 degrees, i.e., the portion between the mask and the blank area in the Figure. By fusing equation 4.1, a new image after manual simulation of beam blockage is obtained, as depicted in Figure 4.1(c). This image contains the original image x, ranging from -1 to 1 at the periphery, and the simulated mask as clear air minimum data.

Moreover, in conjunction with the reflectivity data exemplified earlier, we incorporated two additional radar parameters, namely Doppler velocity and spectrum width, commonly employed in radar applications. These parameters underwent similar preprocessing, normalization, and masking procedures as the reflectivity data, and their details are not enumerated here. Currently, the

model's input layers entail three data dimensions ranging from -1 to 1 and a mask dimension ranging from 0 to 1, collectively constituting four dimensions. The following table provides a detailed description of the data layer input that has been fed into the network layer so far.

Table 4.1: Data layers that input into the network.

	Data Type	Data Size	Data Range
Layer 1	Reflectivity	448×192	[-1, 1]
Layer 2	Doppler Velocity	448×192	[-1, 1]
Layer 3	Spectrum width	448×192	[-1, 1]
Layer 4	PBB mask	448×192	[0, 1]

4.3 CGAN-based Deep Learning Model Architecture

The model employed in this study is an enhanced version of GAN, or Generative Adversarial Networks. Generative Adversarial Networks are a class of generative models that have recently gained immense popularity in the field of deep learning. GANs are unique in that they use two neural networks, a generator and a discriminator, to learn the distribution of a dataset and generate new samples that are similar to the training data. The idea of GANs can be traced back to several predecessors that introduced different concepts and techniques that ultimately led to the development of GANs.

One of the earliest predecessors of GANs was the Boltzmann machine, introduced by John Hopfield in the 1980s, which was a type of recurrent neural network that used a stochastic process to learn the probability distribution of the input data [39]. Specifically, the Boltzmann machine used a form of energy-based modeling, in which each data point was assigned an energy value based on its features. The machine was then trained to minimize the difference between the energy of the training data and the energy of the machine's own output. The Boltzmann machine was a

type of recurrent neural network that used a stochastic process to learn the probability distribution of the input data. However, the Boltzmann machine had several limitations, including difficulty in training on large datasets and scaling to larger models. In the 1990s, Radford Neal introduced the concept of using Markov chain Monte Carlo (MCMC) algorithms to train the Boltzmann machine, which was a technique for sampling from probability distributions [40]. In the context of the Boltzmann machine, the MCMC algorithm was used to sample from the probability distribution of the input data more efficiently. This was done by iteratively updating the state of the machine based on a sequence of random perturbations, so that the machine would eventually converge to the true distribution, but the algorithm was still slow and computationally expensive. To address the limitations of the Boltzmann machine, Geoff Hinton introduced the Restricted Boltzmann Machine (RBM) in 2006 [41]. The RBM was a type of Boltzmann machine that was easier to train and could be used for unsupervised feature learning. The RBM had two layers of neurons, one representing the input data and the other representing a set of latent variables that captured the underlying structure of the data. The machine was trained to maximize the likelihood of the training data, using a contrastive divergence algorithm that iteratively updated the weights of the neurons. Hinton used the RBM to pre-train the layers of a deep neural network, showing that pre-training could improve the performance of the network on a supervised learning task. Another predecessor to GANs was the "adversarial generator-estimator" (AGE) architecture introduced by Jürgen Schmidhuber in the 1990s [42]. The AGE architecture used two neural networks to model the distribution of the input data and the distribution of the generated data, with one network acting as a generator and the other network acting as an estimator. The generator network was trained to generate data that looked similar to the training data, while the estimator network was trained to estimate the probability density function of the input data. The two networks were then used together to generate new data points that had a high probability of being real. However, the AGE architecture was not as successful as GANs and did not receive as much attention. In 2014, Ian Goodfellow and his colleagues introduced GANs [12], which used two neural networks called the generator and the discriminator to train a generative model. The generator network was trained to generate

new data that was similar to the training data, while the discriminator network was trained to distinguish between the generated data and the real data. The two networks were trained together in a minimax game, where the generator tried to fool the discriminator and the discriminator tried to correctly classify the generated data. GANs quickly became popular due to their ability to generate high-quality images, videos, and audio.

The architecture of the generator network is usually designed to be a deconvolutional neural network (also known as a transposed convolutional neural network) that takes a low-dimensional noise vector and outputs a high-dimensional image or sequence of data points, followed by batch normalization and ReLU activation functions. The output layer of the generator network uses an activation function to ensure that the generated data samples are in the range of [-1, 1]. The generator network is trained to produce realistic data samples that can fool the discriminator network. During the training process, the generator network receives feedback from the discriminator network on how to improve the quality of its generated samples. The loss function used to train the generator network is typically the binary cross-entropy loss between the discriminator's predictions on the generated data samples and the ground truth labels. The generator receives feedback from the discriminator, which indicates how well the generator is producing synthetic data that resembles the real data. The generator updates its weights based on the feedback, in order to improve the quality of the synthetic data it produces.

The architecture of the discriminator network is usually designed to be a convolutional neural network that takes an image or sequence of data points as input and outputs a binary classification label (i.e., real or fake), followed by batch normalization and LeakyReLU activation functions. The output layer of the discriminator network uses the sigmoid activation function to ensure that the output is in the range of [0, 1]. The discriminator network is trained to distinguish between real and fake data samples. During the training process, the discriminator network receives feedback from the generator network on how to improve its classification performance. The loss function used to train the discriminator network is typically the binary cross-entropy loss between the discriminator's predictions and the ground truth labels i.e., all ones for real samples and all zeros for

fake samples. The discriminator receives feedback from the generator, which indicates how well the discriminator is able to distinguish between real and synthetic data. The discriminator updates its weights based on the feedback, in order to improve its ability to correctly classify the data.

For our model, GANs have several advantages, such as their ability to generate high-quality, realistic data that can be used for training deep learning models. GANs can also be used for data augmentation, which is a useful technique for increasing the size of a dataset and improving the robustness of machine learning models. However, GANs also have some disadvantages, including their instability during training, the difficulty in tuning their hyperparameters, and the potential for generating biased or unrealistic data.

Conditional GAN (CGAN) is a type of GAN that can be used to generate synthetic data that is conditioned on specific inputs, such as class labels or attribute vectors. CGANs can be used for a variety of tasks, such as image-to-image translation, style transfer, and data correction. One area where CGANs have shown promise is in the correction of radar data, where they can be used to generate synthetic data that is free from noise and other artifacts. There have been several studies that have explored the use of conditional generative adversarial networks (CGAN) for radar data correction, specifically for the task of beam blockage correction. These studies have shown that CGANs are superior to traditional GANs for this task, due to their ability to generate more accurate and realistic data. One of the key advantages of using a CGAN for beam blockage correction is that it allows for the incorporation of additional information, such as weather data, that can be used to improve the accuracy of the correction. This is because the generator network in a CGAN takes both the input radar data and the additional information as input, allowing it to learn a more accurate mapping between the input and output data. Another advantage of CGANs over GANs is that they are better able to handle situations where the training data is imbalanced or incomplete. This is because the conditional nature of the generator network allows it to focus on specific areas of the input data that need to be corrected, rather than trying to generate an entire output image from scratch.

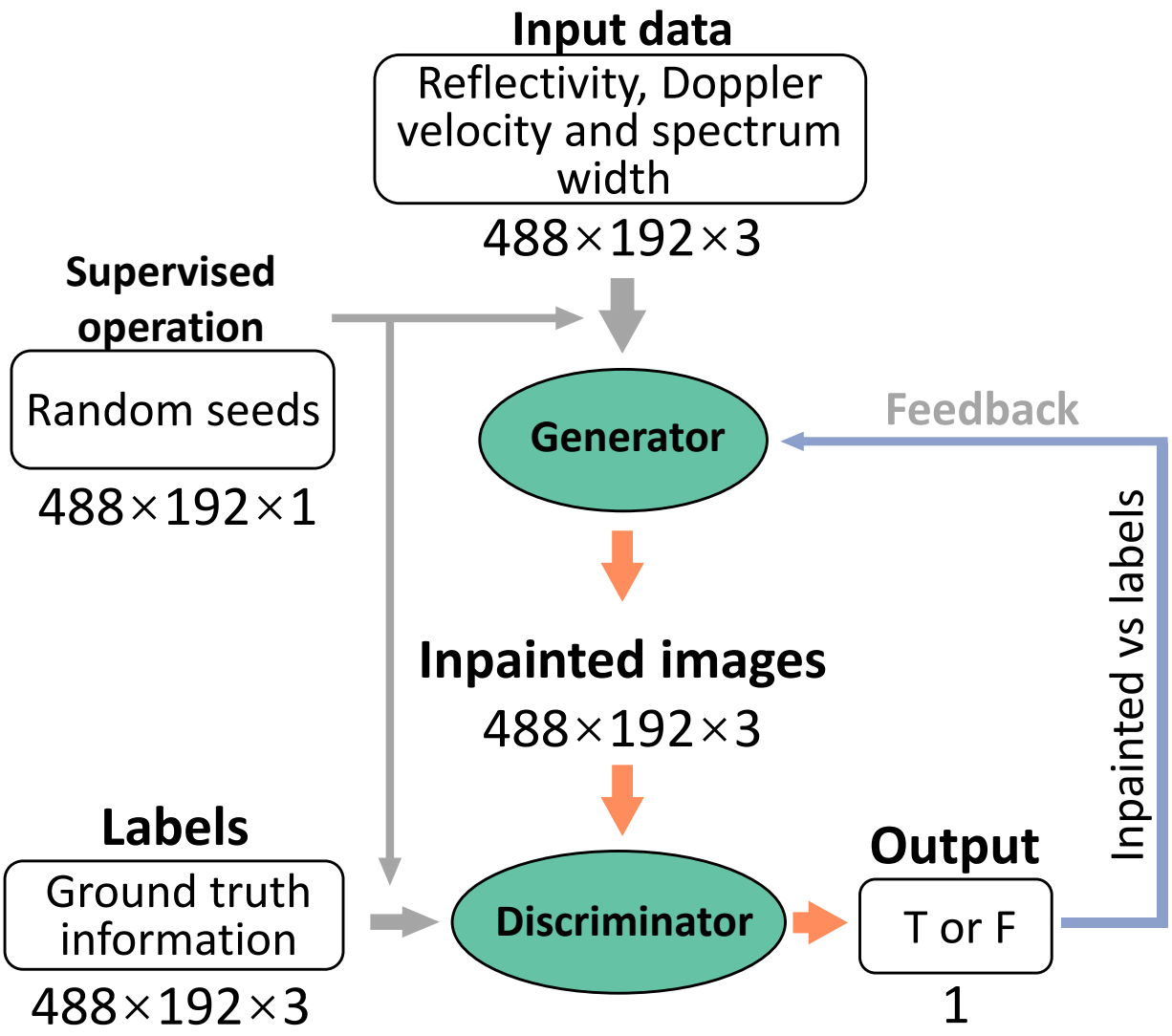

Figure 4.2: CGAN-based deep learning framework for weather radar beam blockage correction.

The workflow of the proposed deep learning model for weather radar beam blockage correction is shown in Figure 4.2. The grey arrows represent the input, the orange arrows represent the output, and the blue arrows represent the feedback layer. As previously stated, the reflectivity, Doppler velocity, and spectrum width data procured from KDAX and KFWS radar, which will be fed into the network as the data layer, undergoes a standardization process, followed by a PBB simulation as shown in Section 3.4 and Section 4.2. To enhance the model's robustness and generalizability by enabling it to learn from diverse noises and anomalies during training,

we introduce a Conditional Generative Adversarial Network (CGAN) as our network architecture [43], necessitating a conditional supervising input layer. In this instance, we utilize Gaussian distributed data with a mean of 0 and a standard deviation of 0.5, which is input into the network at the same size as the radar data, concomitant with the data layer. Subsequently, the simulated input images and random seeds undergo several convolution and deconvolution layers within the generator to extract and generate features from the input. To this end, we utilize the Unet-based network layer as the generator, which exhibits commendable performance in feature extraction and upsampling, which will be detailed in Section III-B1. The images generated by the generator subsequently enter the discriminator along with the ground truth values, which are normalized but not simulated by beam blockage. Within the discriminator, the image and ground truth values undergo a sequence of convolution and pooling operations to extract the final features. These features are then transformed into binary outputs (i.e., true or false) via a final fully connected layer to complete the discrimination process. Notably, the discriminator's classification results are fed back to the generator to optimize the loss function via back-propagation, enabling the generator to generate more realistic images. This adversarial process continuously improves the model's performance, as the generator and discriminator confront each other in this manner. In the following sections, we will examine the generators and discriminators, as well as how to address the integration of the entire correction system.

4.4 Multi-channel Convolution

As previously stated, the four layers of data simulated by the PBB are initially inputted into the generator. To enhance its nonlinearity and interpretability, a multi-channel convolution process is employed on the input data within the generator, as illustrated in Figure 4.3.

Multi-channel convolution, also known as multi-channel feature extraction, is an important step in the preprocessing stage of the GAN training process. It involves the use of a convolutional process to extract features from the input data, which are then used as input for the generator network. The purpose of multi-channel convolution is to capture the complex and high-dimensional

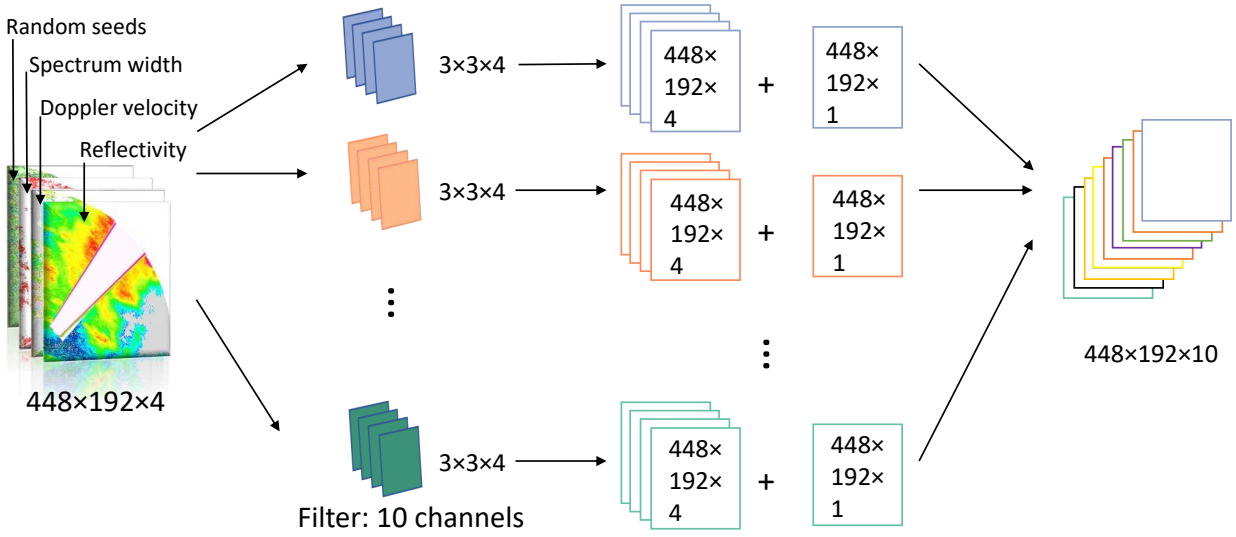

Figure 4.3: Multi-channel convolution of input data layers.

patterns in the input data, such as images or signals. The convolutional layer applies filters, also called kernels, to the input data, and each filter produces a new feature map. The filters are learned through backpropagation during training, and their values determine the output feature maps. By applying multi-channel convolution before entering the generator, we can extract relevant features from the input data that are useful for generating realistic output. For example, in the case of image generation, the convolutional layer can extract features such as edges, textures, and shapes, which are important for creating realistic images. The use of multi-channel convolution also helps to reduce the dimensionality of the input data, which can improve the efficiency of the generator network. By extracting the most important features from the input data, the generator network can focus on generating output that is relevant to the input data, rather than being overwhelmed by the complexity of the input. Furthermore, multi-channel convolution can help to prevent overfitting, which is a common problem in deep learning models. By extracting only the most relevant features from the input data, the generator network is less likely to learn noise in the data which is not useful for generating realistic output.

In this process, the input data is a tensor with a size of $448 \times 192 \times 4$, which includes three layers of data representing reflectivity, Doppler velocity, and spectrum width, along with one layer

of random seed. Initially, a convolution kernel of size $3 \times 3 \times 4$ with a SAME boundary padding and stride of 1 is utilized for convolution in the horizontal direction. The 4-dimensional convolution kernel slides across each dimension of the input image, computes the convolution value for each position and generates a $448 \times 192 \times 4$ feature map. The convolution operation results are summed arithmetically for these four dimensions to obtain a $448 \times 192 \times 1$ feature map. Then, ten different convolution kernels are applied in parallel to extract various features. This process is repeated ten times to produce ten different feature maps. Finally, these ten layers of feature maps are superimposed and input into the Unet-based neural network.

4.5 Generator

The generator is considered the heart of the GAN, as it is responsible for creating new and unique data. The primary goal of the generator is to create examples of a particular class. The specific workflow and training process of the generator is shown in Figure 4.4.

The generator receives input classes X and a noise vector, which consists of random values, and applies some nonlinearities to produce an output image of an example. The input X and noise vector are then passed through a series of nonlinearities, which may include convolutional layers, batch normalization layers, and activation functions, among others. These nonlinearities help to transform the input into a higher-dimensional space where the generator can learn to create more realistic and varied examples. However, the generator can sometimes output the same example every time, which is undesirable. Therefore, to improve its performance over time, we need to input classes X with a random noise vector psi. This process ensures that the generator does not output the same image repeatedly, and the output remains diverse and unique [44].

The output of the generator is an image of an example that belongs to the given class. However, this initial output is not perfect and is likely to contain errors or imperfections. This is where the discriminator comes in. The discriminator's role is to distinguish between the real and generated images. The image produced by the generator is fed to the discriminator, which then outputs a prediction $\hat{Y_d}$ of whether the image is real or fake. Essentially, the generator's goal is to make its

output as realistic as possible, while the discriminator's goal is to distinguish between real and fake images. The cost function is used to measure how far the generator's output is from being real. It is used to update the generator to produce better outputs in the next iteration. The generator tries to minimize this cost function while the discriminator tries to maximize it. This function is used to update the generator and help it produce more realistic images. The generator learns and improves over time based on the feedback from the discriminator [45].

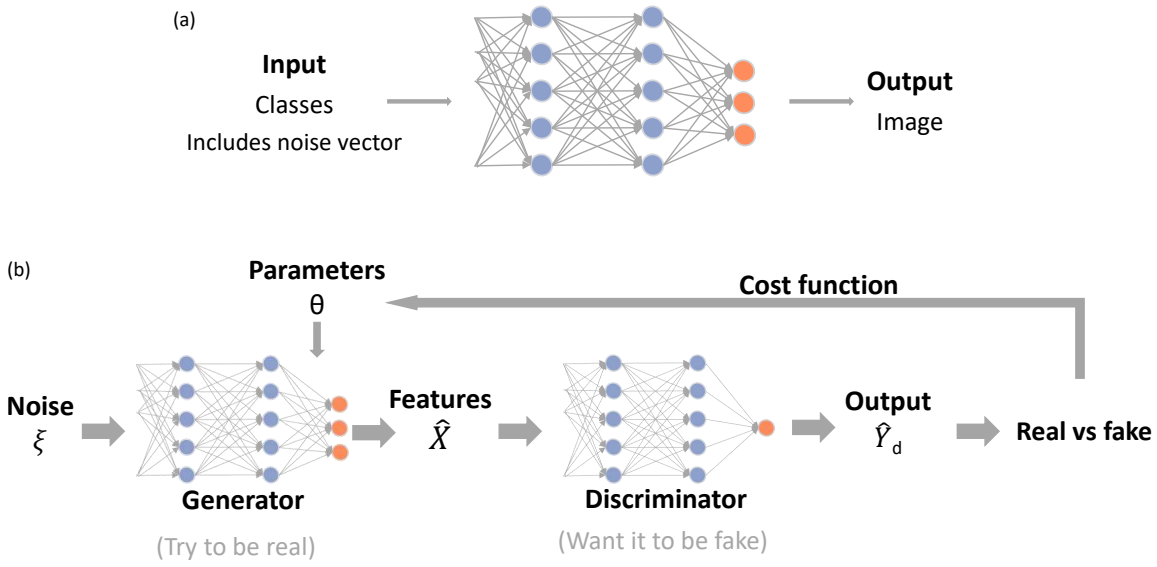

Figure 4.4: The process of generator function in a GAN: (a) generator overview; (b) generator training process.

The data characterized by different feature maps are fed into the Unet-based neural network, where the Unet network [14] initially developed for image segmentation, exhibits a typical encoding-decoding architecture. The encoding-decoding architecture consists of an encoder network that downsamples the input image to extract features and a decoder network that upsamples the features to produce the final output image. However, this architecture suffers from a drawback known as the bottleneck problem, where the information in the input image is lost during the downsampling process and cannot be recovered during the upsampling process. Unet solves the bottleneck problem by introducing skip connections between the encoder and decoder networks. The skip

connections allow the network to preserve the spatial information from the input image by passing it directly to the decoder network. This improves the accuracy of the segmentation results and enables the network to better handle images with complex structures. The architecture is divided into two main parts, an encoder and a decoder, each of which contains several layers. Figure 4.5 shows the initial network structure of Unet with five layers.

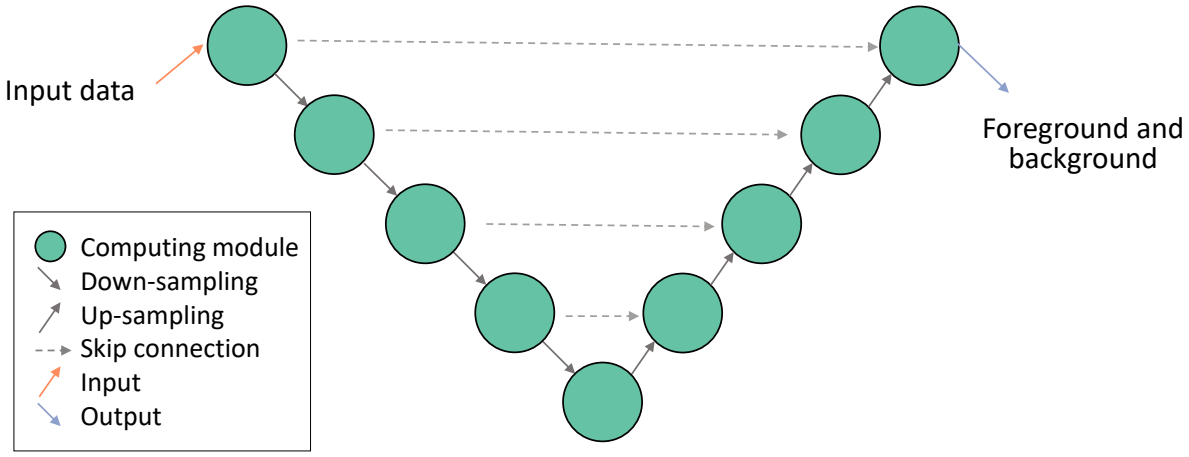

Figure 4.5: Network structure of Unet.

The encoder situated in the first half leverages the pooling layers for gradual down-sampling, augmenting resistance against minor perturbations like image translation and rotation, reducing overfitting risk, minimizing operation numbers, and amplifying the perceptual field's size. It is achieved using max-pooling, which reduces the spatial resolution of the input while increasing the number of channels. This allows for larger receptive fields, which enables the network to capture more context information. In the network structure figure, it is achieved through a series of convolutional layers with a 3x3 filter and a valid mode padding, which is processed in the

computing module in the figure. The filter applied uses valid mode padding that results in a feature map with a smaller size. The down-sampling arrow indicates a 2x2 max-pooling operation which further reduces the size of the image while increasing the number of channels. In the figure, the up-sampling part is used to reconstruct the output image from the features extracted in the down-sampling part. The process of 3x3 convolution reduces the size of the image, so we need to upsample the feature maps to match the size of the input image. The up-sampling is performed using a transposed convolutional layer with a stride of 2. This doubles the size of the input feature map. However, since the output size of the convolutional layer is smaller, we need to crop the feature maps from the down-sampling part to match the size of the up-sampled feature maps.

The other half is the decoder, which performs up-sampling using deconvolution layers, thus restoring the decoded features to their original size and allowing the spatial and edge information of the input image to be gradually regained. Consequently, the low-resolution feature maps get mapped to the pixel-level segmentation result maps. This allows for finer details to be captured in the output, which is important for semantic segmentation tasks. The use of transpose convolutions is also computationally efficient, as it can be performed using matrix multiplication.

The U-Net algorithm further compensates for the lost information during down-sampling at the encoding stage using a skip connection between the encoder and decoder, which combines the feature maps at corresponding positions in both processes, equipping the decoder with additional high-resolution information during up-sampling, resulting in a more precise recovery of detailed information in the original image, thus ameliorating the segmentation accuracy. These connections allow for the decoder to receive information from the encoder at different scales, which enables the network to produce more accurate and detailed segmentations. The skip connections also help to mitigate the vanishing gradient problem, which can occur in deep networks during training. During the training process, the network learns to optimize the weights of the convolutional layers to produce accurate segmentation. The use of skip connections, along with the downsampling and upsampling operations, enables the U-Net to effectively learn the features of the input and produce high-quality segmentations. At the end of the network, we obtain an output image with

two channels. The first channel represents the foreground, while the second channel represents the background. This is achieved through a final convolutional layer with a 1x1 filter.

However, empirical investigations have demonstrated that the appropriate number of layers of the Unet network required for various precipitation models is not consistent [46] [47]. To better capture the global context at different scales for translating unsampled images, we have implemented an enhanced model, Unet++ [16], as a generator. This model retains the benefits of Unet while allowing for the precise integration of features at varying network levels and a significant reduction in the number of parameters.

Unet++ is an extension of the original Unet that introduces a nested and dense skip connection, enabling the model to capture multi-scale contextual information by integrating features from different levels of the network, which improves the model's capacity to generate accurate and detailed output images. Additionally, the dense skip connections in Unet++ augment the information flow between the encoder and decoder, thereby facilitating the recovery of more spatial information in the output image. It also reduces the number of parameters needed for training by reusing the feature maps from different scales and levels, which can reduce overfitting and improve generalization performance. This makes Unet++ more efficient and easier to train than Unet. Unet++ allows for different levels of complexity and scale by adding or removing modules, which can be useful for different applications and datasets. It also provides a more flexible and interpretable architecture for understanding the segmentation process. A hybrid pooling approach is used, which combines max pooling and convolutional pooling, to reduce memory usage during training and inference. This makes Unet++ more memory-efficient and suitable for larger datasets and models. Moreover, Unet++ utilizes fewer parameters and is more efficient than other models. Overall, incorporating Unet++ as the generator for the proposed deep learning model for weather radar beam blockage correction can enhance its performance and robustness, where the operation of the data in the Unet++ based generator network layer is detailed in Figure 4.6.

Among these, after the multi-channel convolution operation of the previous stage, ten feature maps with a size of 448 $\times$ 192 will be input. The data will then undergo down-sampling as

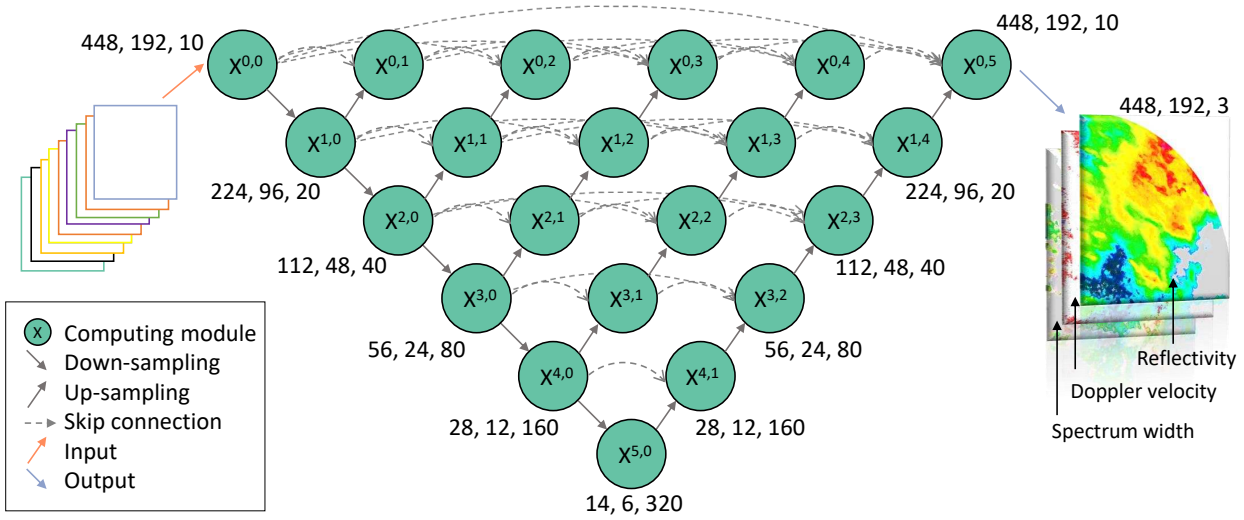

Figure 4.6: Unet++ based generator neural network applied in the weather radar beam blockage correction system.

illustrated in the left half of figure 4.6. Each down-sampling computational unit, i.e., the computing module linked to the down-sampling arrow in the figure, comprises three distinct components. The initial step of the down-sampling calculating unit involves a two-dimensional convolution operation, utilizing a 3×3 convolution kernel, along with SAME boundary fill padding and a stride size of 1. The purpose of This convolution operation aims to train the kernel weights to detect various features across various precipitation events, encompassing cloud edge features for less intense precipitation data like KDAX radar data, and cloudiness texture features for more intense precipitation data such as KFWS radar data [48]. The second component comprises a Leaky Rectified Linear Unit (Leaky ReLU) activation layer featuring a negative partial slope of 0.2 that imparts nonlinearity to the linear weighting while circumventing the issue of vanishing gradients. This results in a superior fit of the precipitation data distribution in more intricate radar data, in line with our model's requirements. The third part comprises a maximum pooling layer with a filter size of 2×2. The layer operates by dividing the input image into several blocks of 2×2 size and selecting the maximum value in each block as the output. Despite the screening and deblending of input data, it still contains a significant amount of noise in the training data. The maximum pooling aids in extracting critical features from the input image, while eliminating

noise and superfluous details, resulting in an enhanced restoration effect. Additionally, the process aids in reducing the spatial dimension of the feature mapping for large volumes of radar data in the model, reducing the number of parameters and computational complexity. The aforementioned process will be repeated five times in a sequence of down-sampling, where the size of the radar data is halved while doubling the dimensionality at each step, which results in a final output of high-dimensional data with the most distinctive features at the end of down-sampling, measuring $14 \times 6 \times 320$.

During the decoding stage of upsampling, a skip connection is established between the down-sampled and up-sampled computing modules. The primary function of this connection is to duplicate the data output generated by the down-sampling pooling layer and combine it with the up-sampling data in the channel dimension through concatenation, which results in twice the number of feature maps in the output. The diminution in feature information during the pooling procedure of down-sampling necessitates the utilization of skip connections to integrate the feature maps before down-sampling with those after up-sampling, thereby preserving the coherence of image features. Moreover, since our correction model utilizes precipitation data from targets with distinct features, feature information at varying scales is required to capture image features more effectively. Unet++ skip connections can proficiently extract multiple-scale feature information by incorporating numerous feature maps and uniting them through skip connections, thus leading to superior image reconstruction outcomes. Multiple skip connections can also diminish the influence of noise in training data and circumvent overfitting predicaments.

The subsequent process is the upsampling stage on the right side of Figure 4.6, where each upsampling computing module comprises four components. The first component is the deconvolution operation with a filter size of 2×2, a stride of 2, and a dilation of 1, which is for increasing the size of the up-sampled input tensor and restoring it to the original input size while retaining the feature information of the original image. In our correction model, as the output of down-sampling is high-dimensional multi-feature data, deconvolution operation can also refine the spatial information of the feature map, thereby enhancing the precision and stability of the model. However,

the deconvolution operation is solely intended to recover the image dimension and some of the features, and therefore is not the complete inverse operation of pooling. As a result, the data copied from the skip connection is necessary to fill the feature gap effectively. Therefore, a merging process is needed for these two components in the second part of the upsampling unit, i.e., the concatenation operation. The merged data contains both up-sampled and down-sampled feature information, which can aid the network in better grasping the contextual and semantic information in the image, consequently enhancing the accuracy and effectiveness of image segmentation or restoration. The third component is a convolutional layer with a filter size of 3×3. As the merged data from the previous stage possesses twice the number of data channels, we require a multi-channel convolutional layer with half the number of channels. Lastly, a similar Leaky ReLU layer is used in down-sampling. This is because the deconvolution operation results in information loss and blurring, necessitating this layer following the deconvolution to increase the nonlinear representation of the network, thereby improving its ability to recover the details and structure of the original image. After five upsampling stages, the data will undergo three-channel convolution with the sigmoid activation function to obtain the corrected data.

During training, the generator's primary task is to produce output images that closely resemble the real images and successfully fool the discriminator. A loss function is employed to train the generator and update its weight parameters through back-propagation. Equation 4.2 specifies the exact function used.

$$\mathcal{L}_{\mathrm{G}} = \lambda E_{x,y,z}[|y - G(x,z)|] - E_{x,z}[\log(D(x, G(x,z)))] \tag{4.2}$$

The generator's loss function is divided into two components. The first component is the $L1$ loss, also known as the mean absolute error (MAE) loss function. In this function, x refers to the input data, which is the radar image generated by PBB simulation, y refers to the corresponding ground truth image, which is the original radar image, z refers to a random noise input, which is the Gaussian distribution random seeds we input, and G is the generator prediction. Thus, this $L1$ loss measures the average absolute difference between the predicted image $G(x, z)$ and the ground truth

image y for all input images x, their corresponding ground truth images y, and condition z. The second component is the adversarial loss, which trains the generator to outsmart the discriminator. Here, D denotes the discriminator output, and the natural logarithm is represented by 'log'. The minus sign is used to convert the maximization problem of the discriminator into a minimization problem for the generator. This component computes the average logarithm of the probability that the generator's output image $G(x, z)$ can deceive the discriminator D, considering all input images x and conditions z. In other words, it measures the discriminator's discriminatory outcome for the real image versus the generated image by the generator. In essence, the objective of the two components of the generator is to impart the ability to create true and real images which can surpass the discriminator, while concurrently minimizing the divergence between the inpainted and ground truth images. To prevent issues such as overfitting, a hyperparameter lambda is incorporated to regulate the weight distribution between the two components.

4.6 Discriminator

The discriminator is another crucial component of the GAN, figure 4.7 provides a detailed illustration of the discriminator's functionality and its training process.

A discriminator is essentially a classifier that takes input features to output the probability of each class as shown in figure 4.7(a). The discriminator takes the features X as input, which is generated by the generator, and computes a series of nonlinearities. It then outputs the probability of each class, which represents the likelihood that the input is real or fake. The discriminator is trained to correctly classify the input data as real or fake.

To train the Discriminator, a set of input features X is given as input along with some parameters theta, which map X to labels Y. The output of the Discriminator will never be exactly the same as Y, so the output is represented as $\hat{Y}$ in figure 4.7(c). The objective of training the discriminator is to minimize the difference between Y and $\hat{Y}$. To achieve this, a cost function is added, which calculates the difference between Y and $\hat{Y}$. The cost function also decides the direction of the parameters to achieve the best result, which is $\hat{Y}$ as close as possible to Y [49].

In the context of GANs, the discriminator's job is slightly different. Instead of classifying inputs into different categories, it tells us how fake the input is as figure 4.7(b). This feedback is used to update the generator, which then tries to generate more realistic data to fool the discriminator, which is why the discriminator plays a critical role in GANs. As the generator generates new examples, the discriminator evaluates them and provides feedback, which is used to adjust the generator's parameters. This process continues iteratively until the generator produces examples that are indistinguishable from real examples.

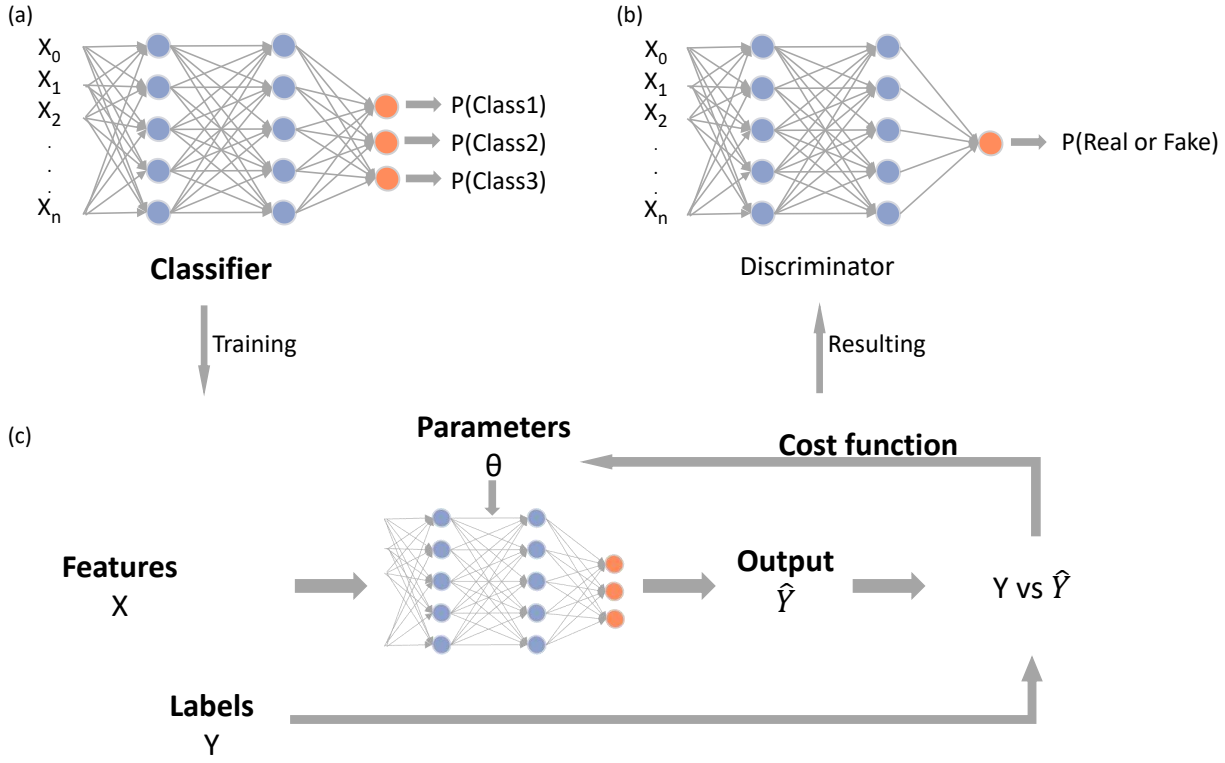

Figure 4.7: The process of discriminator function in a GAN: (a) classifier overview; (b) discriminator overview; (c) discriminator training process.

In our model, the corrected radar data of size $448 \times 192 \times 3$ output from the generator will be fed into the discriminator, along with the ground truth data of the same dimensions. The specific process is shown in Figure 4.8. The primary processing of the discriminator can be broadly categorized into four stages, beginning with the convolutional layer. Similarly to the generator, a filter

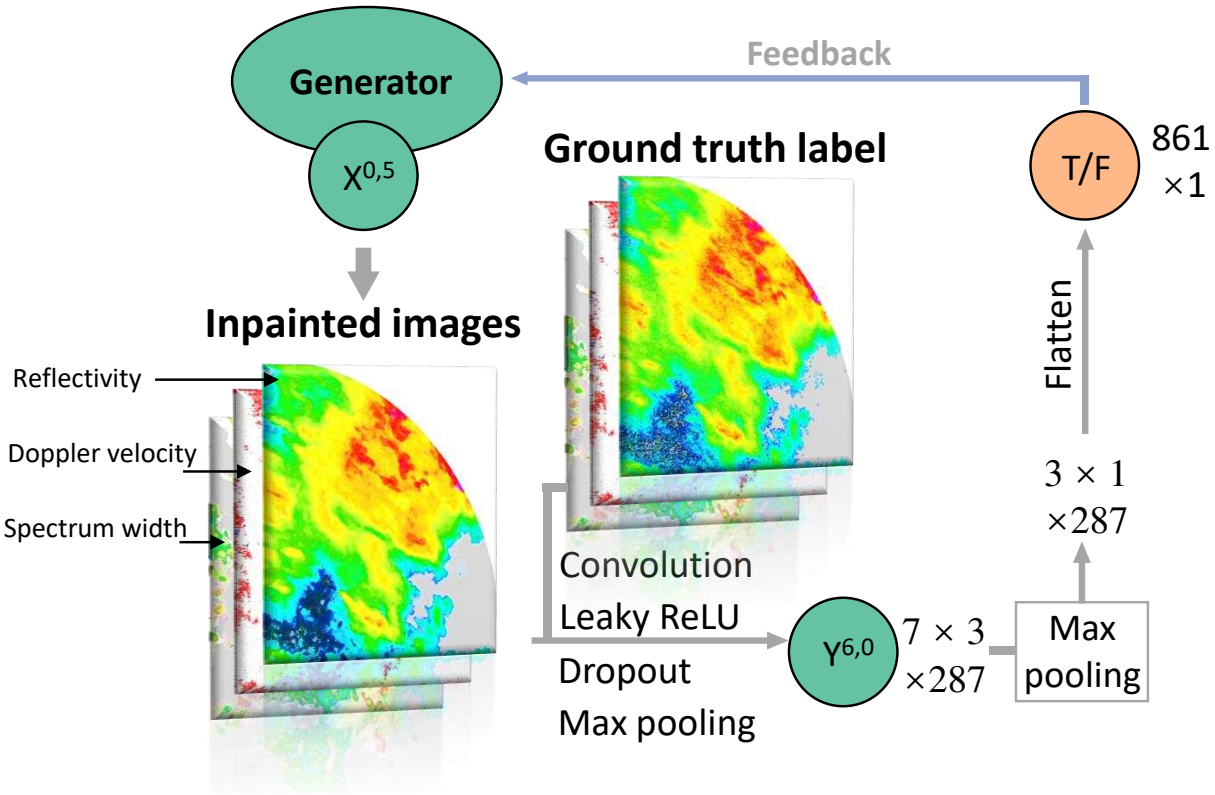

Figure 4.8: Discriminator network applied in the CGAN-based weather radar beam blockage correction system.

of 3×3 is used for convolutional processing. However, unlike the generator's convolutional layer, which aims to capture various features to infer missing pixels, the discriminator's convolutional layer focuses on detecting the same class of features to facilitate comparison with the true values. In the case of radar data, the initial convolutional layers focus on extracting low-level features such as edges and textures, representing cloud patches and noise in the data, which allows the subsequent network to learn higher-level features. The middle layers extract mid-level features such as shape and structure, which aid in separating precipitation information from irrelevant details. The latter convolutional layers extract high-level features that are critical in comparing precipitation size and location in the target region. To further improve the generalization ability of the discriminator and prevent overfitting, the second part of the discriminator also undergoes a Leaky ReLU layer with a negative partial slope of 0.2. Next, a dropout layer is needed to randomly deactivate some neurons to decrease the co-adaptation between neurons in the network, thus enhancing the

model's ability to generalize and prevent overfitting. The dropout layer employed in our correction model is beneficial in reducing the discriminator's reliance on specific local features, which mitigates the risk of the discriminator being excessively sensitive to the repaired image. Consequently, this layer facilitates the discriminator's comprehension and characterization of the entire radar image. The fourth part involves a 2×2 pooling operation that can down-sample the radar image data in the model, thereby extracting the most important features while reducing the spatial size of the data, which helps the discriminator to identify the relevant feature class better. The down-sampling calculating module consisting of the above four parts is integrated within the discriminator, resulting in the processed data size of $7 \times 3 \times 287$ after six layers of this unit. The processed data is then output as $3 \times 1 \times 287$ via the last maximum pooling layer, enabling the transformation of local information in the image into higher-level features for the determination of global features in the final fully connected layer. Subsequently, the feature map is flattened into a one-dimensional vector and mapped to a scalar output value ranging from 0 to 1 via the sigmoid fully connected layer, thus denoting the degree of similarity between the corrected radar image and the ground truth radar image. A value closer to 1 indicates higher similarity, whereas closer to 0 indicates higher dissimilarity. The discriminator generates real or fake feedback representing the difference between the restored radar image and the real radar image, which is then fed to the generator to evaluate the restoration effectiveness.

To provide better guidance for the generator in generating more realistic data, a discriminator loss function is used during training, as expressed in Equation 4.3.

$$\mathcal{L}_{\mathrm{D}} = -E_{x,y}[\log(D(x,y))] - E_{x,\mathrm{z}}[\log(1 - D(x, G(x,z)))] \tag{4.3}$$

The loss function of the discriminator can be bifurcated into two distinct components: the former pertains to the loss incurred on actual data and their corresponding labels, while the latter entails the loss on the data and labels generated by the generator. In the former component, the discriminator assesses the degree of correspondence between the actual data and labels through its probability output, denoted as $D(x, y)$, where x and y correspond to the input data and label,

respectively. As the actual data and labels ought to exhibit high correspondence, a lower loss function value is obtained when the probability of the discriminator output approaches 1, with a minimum value of 0. In the latter component, the loss function evaluates the degree of correspondence between the generated data and labels. Here, x refers to input data, z denotes input random noise, $G(x, z)$ indicates the data and labels generated by the generator, and $D(x, G(x, z))$ represents the probability of the discriminator outputting the generated data and labels. Since the generated data and labels should have low correspondence, a smaller loss function value is obtained when the probability of the discriminator output is closer to 0, with a minimum value of 0. By fusing these two components, the discriminator effectively assesses whether the corrected radar image and ground truth match and outputs the corresponding probability, which is subsequently employed as a component of the generator's loss function to facilitate realistic data and label generation. Notably, our CGAN-based model trains the discriminator and the generator in an alternating manner, where the former is trained first, and its feedback guides the training of the generator in each iteration round [50].

4.7 Post-processing: Merge Output

To better simulate the masked condition in practical scenarios and enable a more thorough evaluation of the PBB component, an overlay merge operation will be executed once the correction data has been generated. This operation, as depicted in Figure 4.9 with the mask Equation 4.4, aims to combine the output of the correction process with the original data to generate a merged image that represents the corrected areas within the context of the original image.

$$x_{merged} = x_{out} \times mask + x \times (1 - mask) \tag{4.4}$$

Figure 4.9 illustrates the merging process of corrected radar data and the variations in the data boundary conditions. The generator produces a corrected radar image as shown in Figure 4.9(a), with reflectivity information used as an example. Figure4.9(b) illustrates a merging mask that is of the same size as the PBB mask layer in the data, as depicted in Figure 2(b). By using the

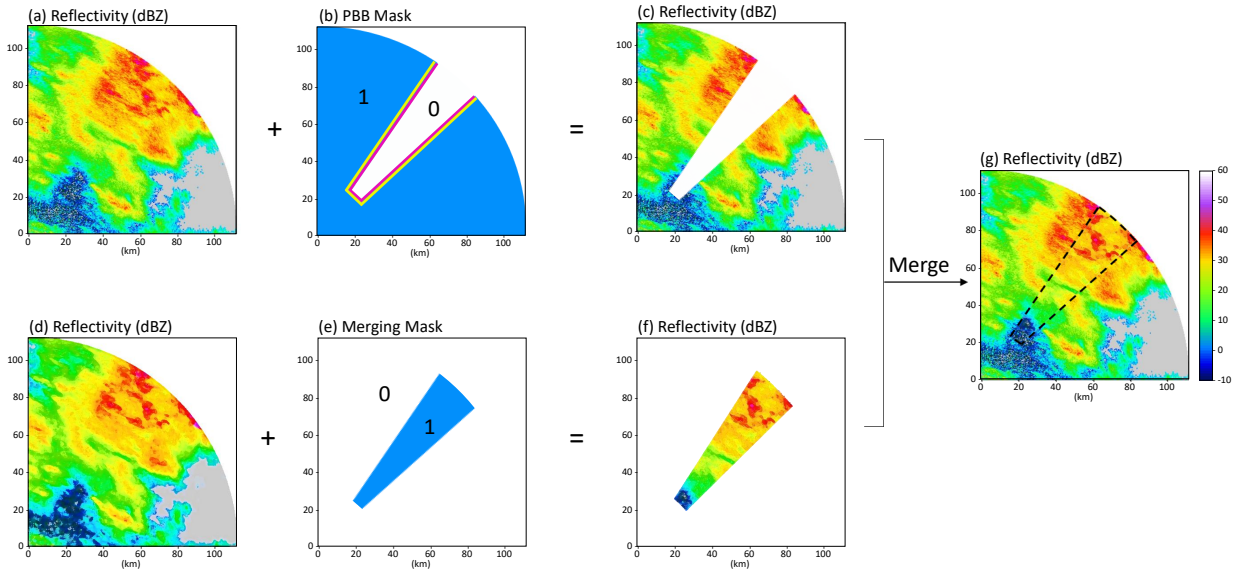

Figure 4.9: Output merging process: (a) raw reflectivity data; (b) PBB mask; (c) simulated data with beam blockage; (d) example of corrected data, i.e., x_{out} in equation 4.4; (e) merging mask, i.e., $mask$ in equation 4.4; (f) inpainted data in PBB region; (g) merged data, i.e., x_{merged} in equation 4.4.

first half of the merging equation, the corrected data that corresponds to the PBB size can be obtained, as shown in Figure 4.9(c). The variable x in the second half of the equation denotes the corresponding ground truth data, which is depicted in Figure 4.1(a). After the merging operation, a corrected image containing PBB position information can be obtained, as shown in Figure 4.9(d).

4.8 Comparison Model

In order to evaluate the effectiveness of the CGAN-based model, we also included a comparison group that uses a one-dimensional linear interpolation method for correction. One-dimensional linear interpolation is a mathematical technique used to estimate values between two known points. It assumes a linear relationship between the two points and uses this relationship to estimate the value at any point between them. In image processing, linear interpolation is used to resize images by adding or removing pixels. When scaling down an image, linear interpolation is used to calculate the values of the new pixels by taking the weighted average of neighboring pixels. This method is often used to fill in missing data points or to smooth out noisy data. In the case of radar

beam blockage, one-dimensional linear interpolation is used to fill in the gaps in the data caused by the beam obstruction.

The process of using one-dimensional linear interpolation to correct radar beam blockage involves several steps. The first step is to identify the location of the beam obstruction. This is done by analyzing the radar data and looking for areas where the signal strength is lower than expected. This region is determined in our model by the boundaries of the mask. Once the location of the beam obstruction has been identified, the next step is to determine the extent of the blockage.

Once the extent of the blockage has been determined, the next step is to use one-dimensional linear interpolation to fill in the missing data points. This is done by estimating the values of the missing data points using linear equations that are based on the values of the data points on either side of the missing points. The result is a reconstructed data set that accurately represents the original data, but with the missing values filled in. This method involves performing linear interpolation from the PBB mask boundary, in dimensions with respect to the azimuth, as specified in equation 4.5.

$$y = y_0 + (y_1 - y_0)\frac{x - x_0}{x_1 - x_0} \tag{4.5}$$

The interpolation process begins by selecting two known points on either side of the blocked point. The values at these two points are used to calculate the slope of the line between them. This slope is then used to estimate the value at the blocked point using a simple equation. In the equation above, x_0 and x_1 represent the known locations of boundaries, y_0 and y_1 represent the precipitation intensity corresponding to those locations, while x and y represent the locations and results of the points to be interpolated, respectively. This formula allows for the estimation of the value of a point between two known points based on their respective values.

4.9 Conclusion

In this chapter, we have discussed the methodology section of the CGAN-based beam blockage correction system, which comprises various essential components that operate in unison to achieve precise and efficient correction outcomes.

The first step involves simulating beam blockages in the radar data to create a training dataset for the CGAN. This step is crucial to ensure that the CGAN can effectively learn how to correct beam blockages in real radar data. Next, multi-channel convolution is used to preprocess the simulated data and extract features from the input images. This step helps to enhance the quality of the input data and improve the overall performance of the system. The UNet++ generator is then used to perform the beam blockage correction. This powerful deep-learning model is designed to generate high-quality output images that accurately represent the corrected radar data. The UNet++ architecture also incorporates skip connections and residual blocks, which help to improve the accuracy and stability of the model. The discriminator plays a crucial role in the CGAN-based beam blockage correction system by providing feedback to the generator. By comparing the generated output images to the real radar data, the discriminator can determine the accuracy of the generator's predictions and provide feedback to help the generator improve over time. Finally, the one-dimensional linear interpolation comparison model is used to evaluate the performance of the CGAN-based beam blockage correction system. This model provides a baseline for comparison and helps to demonstrate the effectiveness of the CGAN-based approach.

In conclusion, the methodology section of the CGAN-based beam blockage correction system involves several crucial steps. Firstly, the beam blockage simulation is conducted to generate realistic beam blockage scenarios. Then, the multi-channel convolution is used to preprocess the simulation data and prepare it for input into the UNet++ generator. The generator is trained to learn the mapping between the input and the desired output, and the discriminator is trained to distinguish between the generated and ground truth. Additionally, a one-dimensional linear interpolation model is used as a comparison group to evaluate the performance of the CGAN model.

Moving on to the next chapter, the focus will be on the training and results of the CGAN-based beam blockage correction system. The performance of the model will be evaluated using various metrics. The results from CGAN models will also be compared with the one-dimensional linear interpolation model to demonstrate the superiority of the CGAN-based approach. Additionally, the chapter will discuss the practical applications of the proposed method and its potential for future improvements.

Chapter 5

Results and Evaluation

5.1 Introduction

In this chapter, we will present the results of training the CGAN-based beam blockage correction system and evaluate its performance. We will start by discussing the dataset used for training and the training process, including the hyperparameters and optimization techniques. Then, we will present the quantitative and qualitative evaluation of the system's performance using various metrics and visualizations. Finally, we will compare our system's performance with the one-dimensional linear interpolation comparison model and discuss the limitations and potential future improvements of our approach. Overall, this chapter will provide a comprehensive analysis of the effectiveness and efficiency of our proposed CGAN-based beam blockage correction system.

5.2 Training and Evaluation Methods

As previously stated, to enhance the adaptability and generalization of the model across various precipitation patterns and thereby augment the predictive accuracy of the model, a total of three models were employed. Specifically, these comprise the convective precipitation CGAN correction model based on KFWS data, the stratiform precipitation CGAN correction model based on KDAX data, and the conventional linear interpolation correction model. In the interest of concision, we designate the machine learning algorithm for KDAX training data as the CA model and for KFWS training data as the TX model. Therein, two CGAN models have been trained using 10,000 precipitation cases each from KDAX radar and KFWS radar, obtained from May to September 2019-2021 in Northern California and Northern Texas, respectively. As a result of the varying precipitation intensities in the data sets, the learning rates of the network layers for the precipitation features are not uniform, which results in differing convergence rates of the two training models. To address

this, the TX model was trained for 20,000 iterations, and the CA model was trained for 30,000 iterations, with learning rate decay applied at 14,000 and 19,000 iterations, respectively.

In order to comprehensively evaluate the performance of the proposed precipitation correction models and enhance the system's ability to handle different types and degrees of blockage can be assessed, we tested a diverse set of precipitation data from May 2017 and May 2021. The test data included random azimuthal PBB masks as well as fixed angle masks of 4, 10, and 16 degrees to enable more rigorous evaluation and comparison of the model's efficacy. Additionally, we performed cross-testing between the two CGAN models to identify the efficacy of correcting various types of precipitation. The same set of tests was also applied to the linear interpolation comparison group to ensure fair comparison and evaluation. The forthcoming tables will display some of the parameters employed in the training process.

Table 5.1: Hyper-parameters and training parameters in CGAN model.

Radar	V	R×A	C_0	g	Depth	Batch Size	Training Batches	LR-Reduction
KFWS	10000	448×192	10	2	6	16	20000	14000
KDAX	10000	448×192	10	2	6	16	30000	19000

Table 5.1 shows some parameters and hyper-parameters when training the model. V is the data volume for precipitation cases. R×A is the range and azimuth size we input. C_0 is the number of channels after multi-channel convolution. g is the multiple after each down-sampling. Depth is the layer numbers of Unet++. Batch Size is the number of samples per batch. Training Batches is the total number of batches we trained. LR-Reduction is the batch number when the learning rate is reduced.

To assess the effectiveness of the proposed beam blockage correction model, this article employed four statistical hypothesis tests to analyze the true positive (TP), false negative (FN), true negative (TN), and false positive (FP) evaluations. Additionally, we applied four commonly used

evaluation metrics, namely probability of detection (POD), false alarm rate (FAR), critical success index (CSI), and Heidke skill score (HSS), to measure the model's performance in detail. The calculation methods for each of these metrics are described below.

$$\mathrm{POD} = \frac{TP}{TP + FN} \tag{5.1}$$

$$\mathrm{FAR} = \frac{FP}{TP + FP} \tag{5.2}$$

$$\mathrm{CSI} = \frac{TP}{TP + FN + FP} \tag{5.3}$$

$$\mathrm{HSS} = \frac{2 \times (TP \times TN - FP \times FN)}{(TP + FN) \times (FN + TN) + (\mathrm{TP} + \mathrm{FP}) \times (FP + TN)} \tag{5.4}$$

In this context, the predicted values are generated by the CGAN correction model or linear interpolation, while the actual values are the ground truth data. To determine true positives (TP), both actual and predicted values must be greater than or equal to a given threshold. False negatives (FN) occur when the actual value is greater than or equal to the threshold, but the predicted value is less than the threshold. False positives (FP) occur when the actual value is less than the threshold, but the predicted value is greater than or equal to the threshold. Finally, true negatives (TN) occur when both the actual and predicted values are less than the threshold. To evaluate the performance of the PBB correction model, this study employs four statistical hypothesis tests, namely TP, FN, TN, and FP, along with four commonly used evaluation metrics:

- The probability of detection (POD), is a measure of how well the correction system detects the actual occurrence of a phenomenon, such as a cloud or precipitation. It is defined as the ratio of true positive detections to the total number of actual occurrences. A high POD value indicates a high detection rate and a low rate of missed detections.

- False alarm rate (FAR), is a measure of how often the correction system falsely detects a phenomenon that is not actually present. It is defined as the ratio of false positive detections to the total number of negative samples. A low FAR value indicates a low rate of false detections, meaning the algorithm is not generating many false alarms.

- Critical success index (CSI), is a measure of how well the correction system both detects the actual occurrences and correctly identifies the absence of the phenomenon. It is defined as the ratio of true positive detections to the sum of true positives, false positives, and false negatives. A high CSI value indicates that the algorithm is correctly detecting the presence or absence of the phenomenon.

- Heidke skill score (HSS), is a measure of how well the correction system performs compared to random chance. It is defined as the difference between the ratio of true positives to the total number of samples and the expected ratio of true positives based on random chance, divided by the difference between the maximum possible ratio of true positives and the expected ratio of true positives based on random chance. A score of 1 indicates perfect performance, while a score of 0 indicates performance no better than random chance.

These metrics provide different perspectives on the performance of a classification model, and can be used together to gain a comprehensive understanding of the model's strengths and weaknesses. For example, POD and FAR can help to identify whether the model is biased towards one type of classification i.e. false positives or false negatives, while CSI and HSS can provide an overall assessment of the model's accuracy, all of which are calculated within the range of 0 to 1.

5.3 Testing Results and Evaluation

Figures 5.1 and 5.2 present the outcomes of the KFWS radar observation with reflectivity data on May 16, 2021, at 20:00 UTC and March 1, 2018, at 7:13 UTC. Overall, the CGAN correction model demonstrates superior accuracy to the linear interpolation approach in both scenarios. Comparing the two CGAN models, it is evident that the TX model provides more realistic and nonlinear

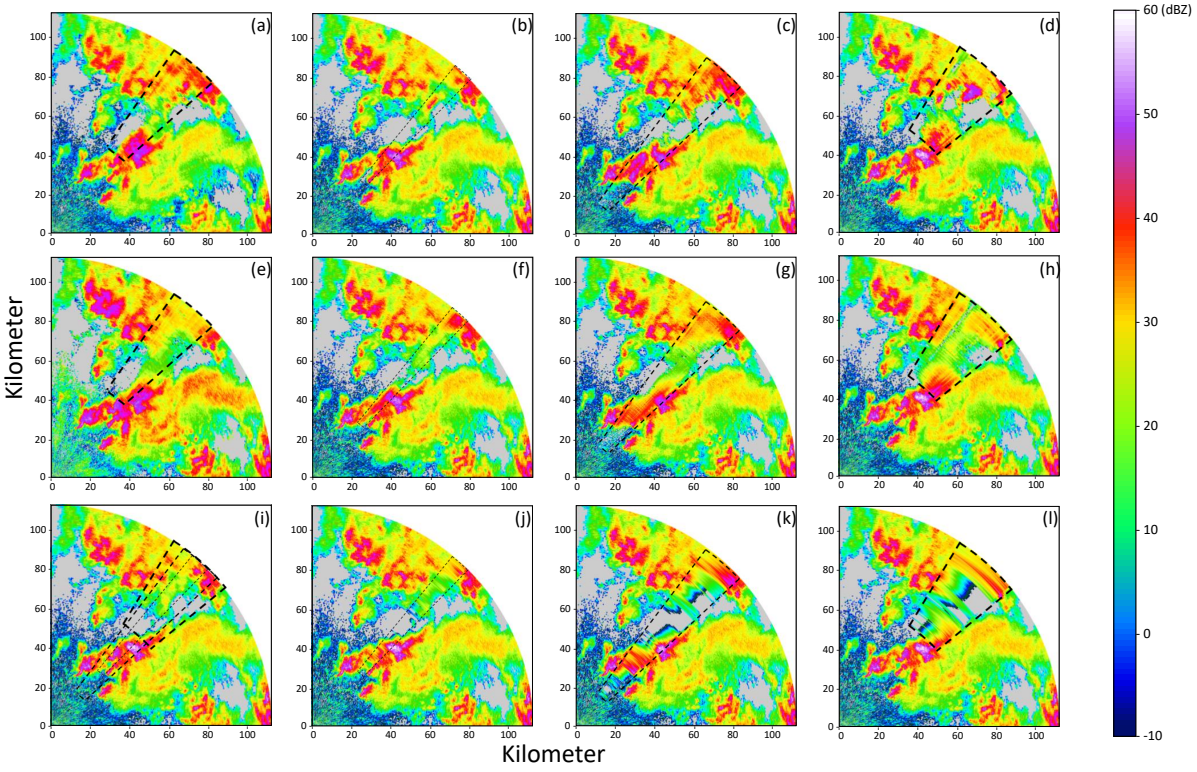

Figure 5.1: KFWS test event at 2000UTC, May 16, 2021: (a)-(d) CGAN output with convective (KFWS) training model of random size, 4°,10° and 16° PBB mask; (e)-(h) CGAN output with stratiform (KDAX) training model of random size, 4°, 10° and 16° PBB mask; (i) ground truth; (j)-(l) linear interpolation output of 4°, 10° and 16° PBB mask.

correction than the CA model, particularly in regions with high precipitation. However, the CA model is more precise in estimating precipitation intensity in low-precipitation areas. Additionally, the TX model exhibits better cloud edge feature recognition and processing compared to the CA model, which fails to remove precipitation and clear air data accurately.

Regarding correcting the precipitation regional center data, the TX model may overfit, which depends on the magnitude of the ground truth center data. As the beam blockage angle increases in this model, some corrected regional center data becomes inaccurate, although this does not affect the accuracy of cloud shape and precipitation characteristics. The correction difference between the radar center point and the far end appears insignificant as the range of beam blockage becomes larger.

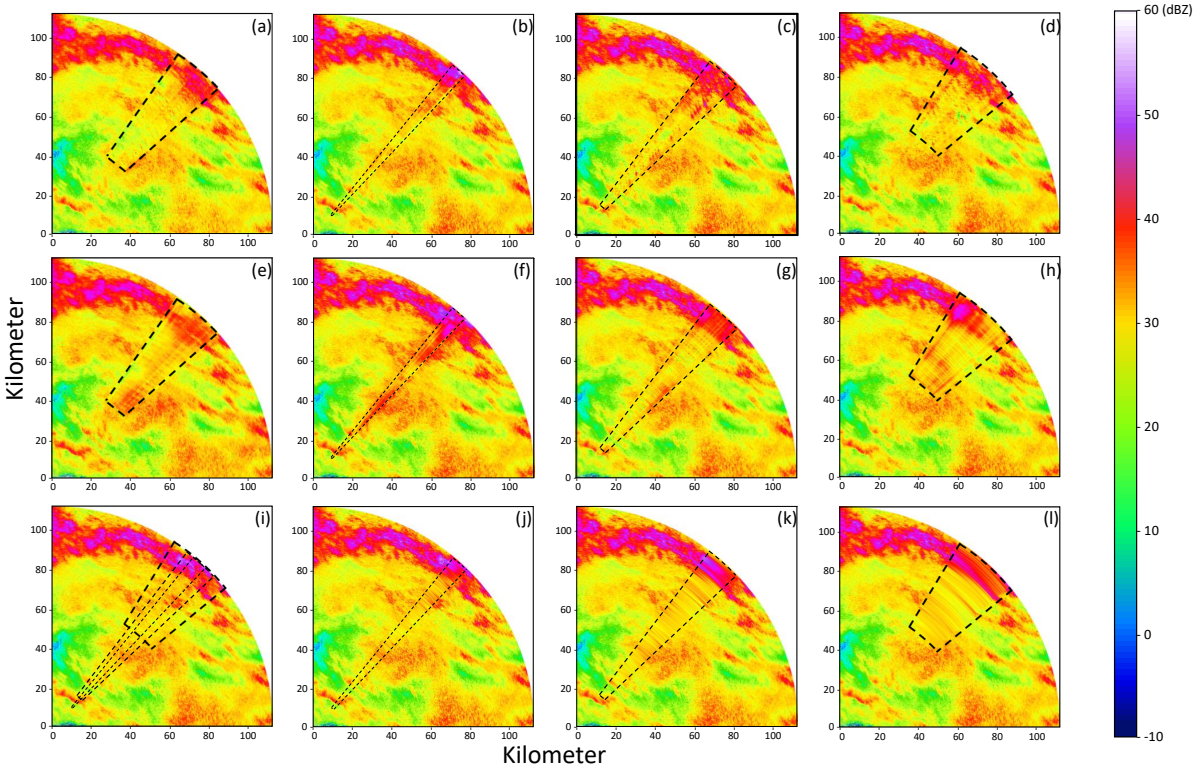

Figure 5.2: KFWS test event at 0713UTC, Mar 1, 2018: (a)-(d) CGAN output with convective (KFWS) training model of random size, 4°,10° and 16° PBB mask; (e)-(h) CGAN output with stratiform (KDAX) training model of random size, 4°, 10° and 16° PBB mask; (i) ground truth; (j)-(l) linear interpolation output of 4°, 10° and 16° PBB mask.

In contrast, as the beam blockage angle increases in the CA model, there is an apparent incoherence of precipitation information in the correction area, with feature loss in the center of the large angle mask and corrected data similar to stratiform precipitation. The correction difference between the center point and the far end of the radar is noticeable, primarily in cloud boundary information. Furthermore, the unfitting of the precipitation model leads to ripple-like data left by precipitation sampling when a large area of beam blockage appears.

The linear interpolation comparison model reveals that as the beam blockage angle increases, the accuracy of precipitation information reduces, and cloud edge information is lost entirely. Note that this model only guarantees the coherence of precipitation information, and the accuracy of data and features is poor.

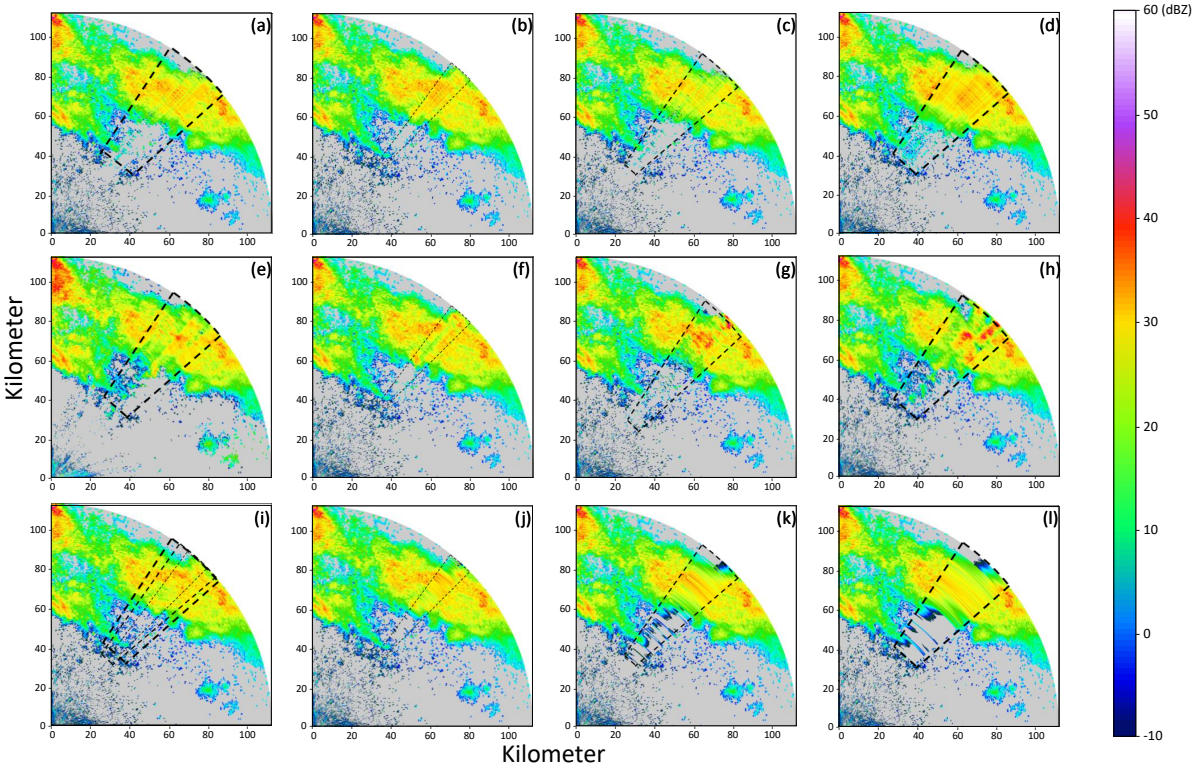

Figure 5.3: KDAX test event at 0310UTC, May 07, 2017: (a)-(d) CGAN output with convective (KDAX) training model of random size, 4°,10° and 16° PBB mask; (e)-(h) CGAN output with stratiform (KDAX) training model of random size, 4°, 10° and 16° PBB mask; (i) ground truth; (j)-(l) linear interpolation output of 4°,10° and 16° PBB mask.

Overall, the CGAN correction model outperforms the interpolation model in terms of precipitation information coherence, accuracy, cloud edge accuracy, and feature integrity. Both TX and CA models have pros and cons in terms of data accuracy, with the TX model performing slightly better in feature reconstruction and cloud edge processing for the convective precipitation cases, but both exhibit higher corrected performance than the interpolation model.

Figures 5.3 and 5.4 depict the outcomes obtained on May 7, 2017, at 3:10 UTC and May 31, 2017, at 13:55 UTC, based on the KDAX radar observation with reflectivity data. Upon comparing the two CGAN models, it is evident that the CA model can effectively reconstruct the features of stratiform precipitation and reinstate interior and edge information. In contrast, the TX model exhibits slight overfitting in the center, leading to the correction of extraneous detail that is not present in the edge and clear air regions.

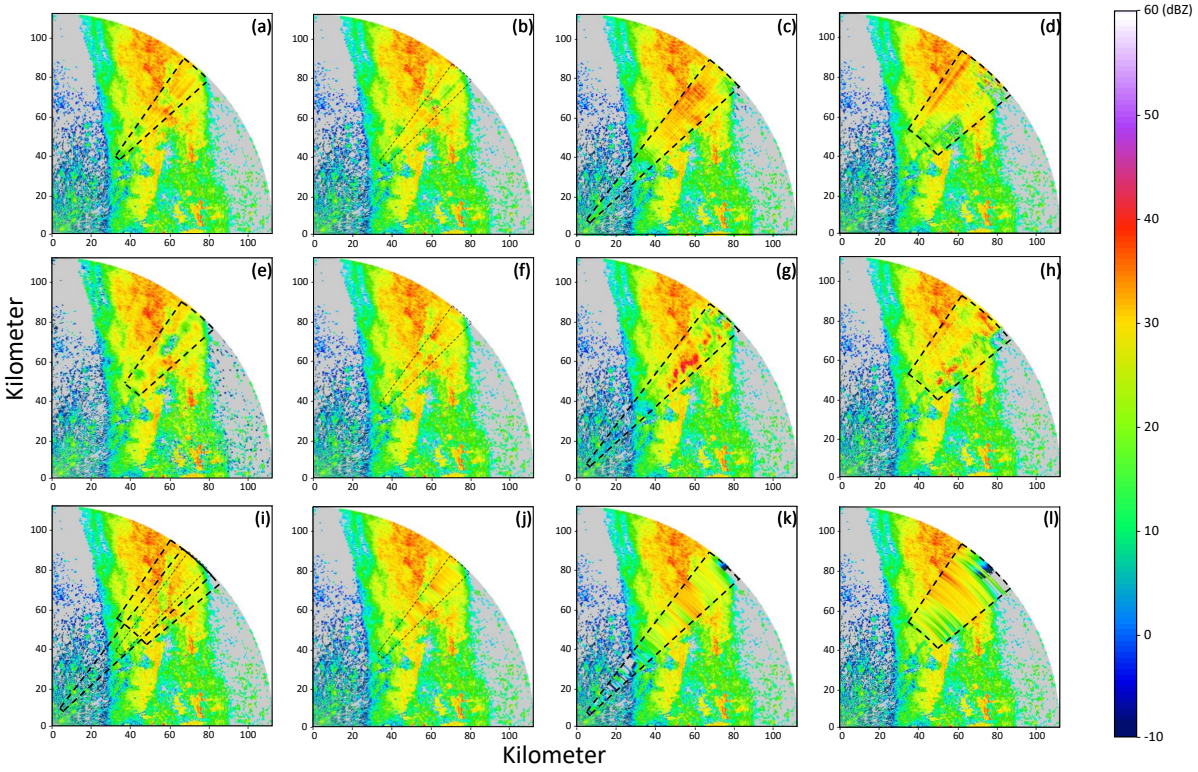

Figure 5.4: KDAX test event at 1355UTC, May 31, 2017: (a)-(d) CGAN output with convective (KDAX) training model of random size, 4°,10° and 16° PBB mask; (e)-(h) CGAN output with stratiform (KDAX) training model of random size, 4°, 10° and 16° PBB mask; (i) ground truth; (j)-(l) linear interpolation output of 4°,10° and 16° PBB mask.

In the case of the CA model, simulated precipitation features become more pronounced as the beam blockage angle increases, albeit not entirely restored. However, the corrected data becomes smoother as the range of beam blockage grows, with no substantial corrected differences between the distance to the radar center point and the far end.

For the TX model, an increase in beam blockage azimuthal angle results in the appearance of convective precipitation features in small magnitudes, which leads to a slight fraction of precipitation above the actual value. Furthermore, the model tends to simulate complex boundary cases at the cloud edge that differ from the ground truth values when there is a large blockage angle.

In contrast to the interpolation model, the CGAN correction model exhibits significantly higher data and cloud edge accuracy when dealing with large-scale stratiform precipitation blockage. While the CA model marginally has better performance in terms of feature completeness than the

TX model, both models perform well overall in correction and are superior to the interpolation model.

Upon comparing the outcomes of the two experiments, it is evident that the CGAN model demonstrates superior corrective capability over the interpolation model, except for restoring stratiform precipitation with low variation values and small blockage angles. In addition, the TX model exhibits higher adaptability and can significantly enhance the accuracy and comprehensiveness of both its own and CA model corrections. However, the CA model is solely effective for rectifying stratiform precipitation.

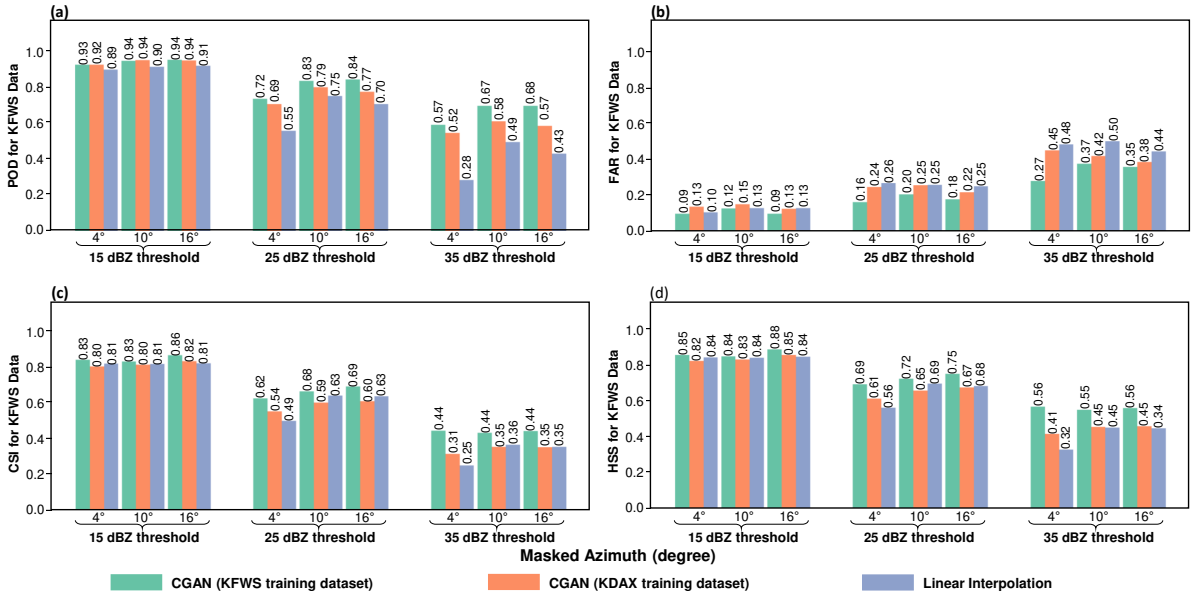

Figure 5.5: Evaluation results of the beam blockage correction performance for KFWS test data with different inpainting methods, blockage sizes, and reflectivity thresholds: (a) POD; (b) FAR; (c) CSI; (d) HSS.

Figures 5.5 and 5.6 summarise the quantitative assessment results for different beam blockage angles under various correction models. Ten test results were randomly selected from data outside the training set spanning from 2017 to 2021. These results were evaluated and weighted ties were applied to obtain global results. To emphasize differences in correction performance at different rainfall intensities, evaluation scores were computed using three thresholds (15, 25, and

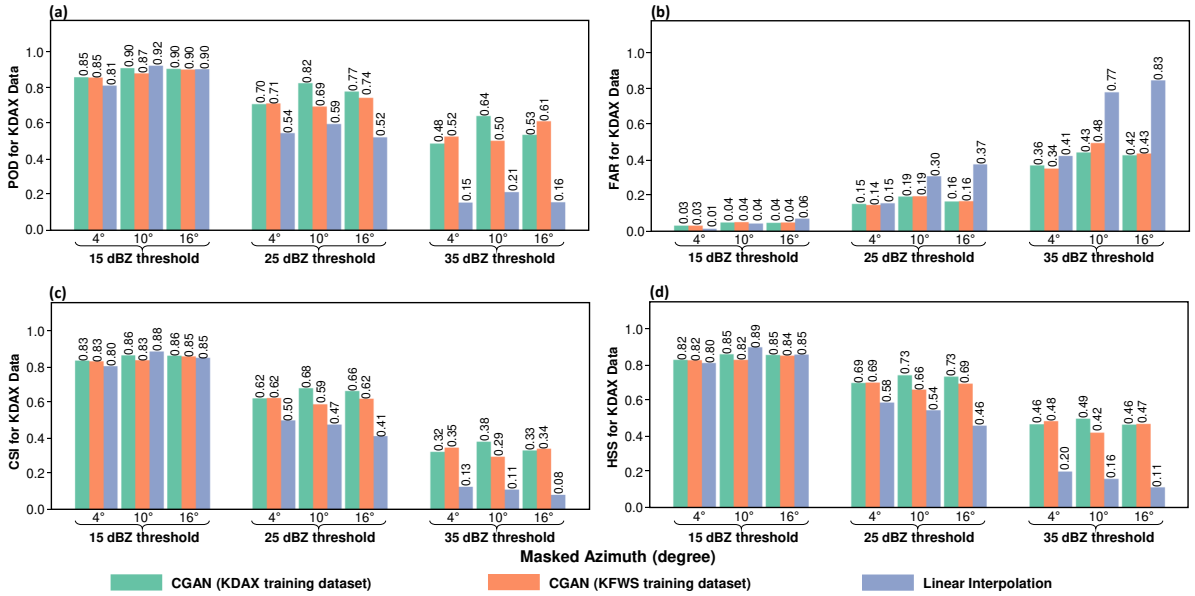

Figure 5.6: Evaluation results of the beam blockage correction performance for KDAX test data with different inpainting methods, blockage sizes, and reflectivity thresholds: (a) POD; (b) FAR; (c) CSI; (d) HSS.

35 dBZ). Figure 5.5 reveals that the correction performance decreases as the blockage angle and threshold increase. The performance is observed to decrease progressively in accordance with the same-model test, cross-model test, and interpolation-model gradient. The CA model sometimes exhibits less effectiveness than the interpolation model, which aligns with the conclusion reached in Figures 5.1 and 5.2. Furthermore, the correction performance tends to decrease more steeply as the blockage angle increases; i.e., the difference in correction performance between 16-degree and 10-degree angles is much more significant than between 10-degree and 4-degree angles for the same model, which indicates that correction difficulty and performance are not linearly related to blockage size.

Figure 5.6 exhibits a similar performance gradient as presented in Figure 5.5. Notably, for certain small and medium angle corrections, the TX model's performance approaches or surpasses that of the CA model, which aligns with the precipitation intensity inclusiveness highlighted earlier. Additionally, linear interpolation for stratiform precipitation correction leads to entirely erroneous data when evaluated at high thresholds, which explains the gap in correction outcomes between the

CGAN model and the interpolation model is more evident in the TX model. It was also found that some of the 25 dBZ threshold performances appeared better than the 15 and 35 dBZ thresholds. This is because the 25 dBZ threshold is an intermediate value between the two extremes of 15 and 35 dBZ. The 15dBZ threshold represents a lower value of reflectivity, and the 35 dBZ threshold represents a higher value of reflectivity. When using a 15 dBZ threshold, there is a higher likelihood of overestimating the beam blockage and correcting non-blocked regions, leading to false positives. On the other hand, when using a 35dbz threshold, there is a higher likelihood of underestimating the beam blockage and leaving blocked regions uncorrected, leading to false negatives. The 25 dBZ threshold strikes a balance between these two extremes and allows for more accurate correction of beam blockages.

5.4 Conclusion

In this chapter, two models were trained to compare the performance of a conditional generative adversarial network (CGAN) correction model with that of a linear interpolation approach for correcting precipitation data in reflectivity observations.

The TX model provides better nonlinear correction, particularly in regions with high precipitation, and better cloud edge feature recognition and processing compared to the CA model, which struggles to remove precipitation and clear air data accurately. However, the TX model may overfit, leading to inaccuracies in correcting the precipitation regional center data when the beam blockage angle increases. In contrast, the CA model is more precise in estimating precipitation intensity in low-precipitation areas and can effectively reconstruct features of stratiform precipitation.

Quantitative assessment results showed that the correction performance decreases as the blockage angle and threshold increase, and the performance decreases progressively in the same-model test, cross-model test, and interpolation-model gradient. Additionally, the correction performance tends to decrease more steeply as the blockage angle increases, indicating that correction difficulty and performance are not linearly related to blockage size.

In conclusion, the CGAN model demonstrates superior corrective capability over the interpolation model, except for restoring stratiform precipitation with low variation values and small blockage angles. The TX model exhibits higher adaptability and can significantly enhance the accuracy and comprehensiveness of both its own and CA model corrections. However, the CA model is solely effective for rectifying stratiform precipitation. The evaluation results provide useful insights into the performance of the CGAN models in correcting beam blockage effects in radar precipitation data.

In the forthcoming chapter, we shall deliberate upon the concerns highlighted in the preceding section, culminating in an exhaustive summary and an appraisal of the prospects of the future's practicability.

Chapter 6

Discussion and Conclusion

6.1 Discussion

6.1.1 Impact of Pseudo-Radar Observations on the Beam Blockage Correction Performance

In Chapter 3, it was mentioned that the dataset would incorporate pseudo-radar observations after undergoing cut and filter processes, which can effectively eliminate low-quality observations and decrease the noise and outliers in the training data, thus enhancing the model's robustness and accuracy. However, it is necessary to analyze the differences between the pseudo-radar observations and the original radar data to determine the impact on the model. In convective precipitation, cloud data typically exhibit a significant reflectivity core with irregular echo interior partitions [28]. The screening of pseudo-radar observations primarily focuses on areas with elevated reflectivity, while features of the echo interior partition, which are considerably lower in magnitude than the core peak, are barely selected. This can result in a small range of unsmoothed data in the corrected data. Although the impact can be negligible on the KFWS test data, it renders the KDAX test data less plausible, particularly when significant angle correction is necessary. In the case of the CA model, the precipitation in the original data is in the form of low-intensity and large areas. Clipping in the image's center can cause the training set to lose some features of the cloud boundaries, leading to a tendency to round off the boundary when handling convective precipitation with complex boundary information, which can limit the model's generalization to new and unseen storm events thus reducing the practical utility. To address this issue, the optimal solution is to increase the amount of training data and add small precipitation events to the training set.

Furthermore, the original plan position indicator (PPI) radar images used in the model are two-dimensional images with centroids. Rotational filtering of the pseudo-radar observations leads to the loss of centroid orientation information. In other words, the distal data of the pseudo-radar

observations may point closer to the radar, which can confuse the precipitation characteristics at the far with near radar points during training, making it impossible to accurately evaluate the difference in performance between the two ends of the data in the range direction or continue the experiment to analyze the effect of beam blockage shape on the corrected data. To address this issue, pseudo-radar observations can be modified to employ sectors within concentric circles, ensuring that the range direction always points towards the radar. Alternatively, models with high rotational robustness can be utilized to ensure that the network captures the appropriate features.

6.1.2 Potential of Dual Model Systems

In the testing phase, a cross-model testing approach was employed, wherein the convective precipitation model and the stratiform precipitation model were tested for their feasibility in using the correction of the other model. As inferred in 4, the convective precipitation training model can accurately simulate beam blockage in stratiform precipitation, encompassing precipitation intensity and cloud edge characteristics, while the stratiform precipitation training model is better equipped for corrections in a limited range of beam blockage. Therefore, the possibility of fusing the two models can be examined. For instance, in northern Texas, convective precipitation is typical in summer, particularly during late afternoons and evenings, while stratiform precipitation is expected in spring and winter. However, simultaneous occurrences of both precipitation types may make using a single model impractical [51]. Therefore, we can merge the dual CGAN models and use the surrounding unblocked data to determine the appropriate precipitation model for the corrected area. In northern California, we can use a dual-model system that combines traditional interpolation methods with CGAN models. Since the edge features of stratiform precipitation clouds are simple, the interpolation method first provides a baseline correction to the radar image, while the CGAN model refines the correction and handles more intricate precipitation patterns.

However, the dual-model system has some potential drawbacks, including increased computational cost and complexity due to the need to train and use multiple models for calibration. Additionally, inconsistent calibration results may arise when using different models, making it dif-

ficult to compare and evaluate system performance, thereby necessitating the development of new evaluation methods. Despite these concerns, the benefits of a dual-model system outweigh the drawbacks and are essential in the practical application of the model.

6.1.3 Impact of Beam Blockage Size on the Correction Model Performance

In Chapter 5, we discussed a scenario where the linear interpolation model achieved a similar correction performance to the deep learning model in a small-angle blockage test of stratiform precipitation. Since linear interpolation is a straightforward method that estimates missing data points based on the surrounding data, however, this method assumes a linear variation of data between the available data points, which may result in inaccurate estimation when the amount of missing data is large or when the data varies non-linearly. Small areas of stratiform precipitation where the precipitation core is not apparent lead to a linear increase in reflectivity from the cloud edge to the center, corresponding to the linear interpolation assumption. When the radar beam blockage size is significant, the increase in non-linear data results in linear interpolation correction that cannot accurately estimate the missing data. In convective precipitation correction, convective clouds usually comprise several precipitation cores characterized by different peaks, making it challenging to have a uniform linear distribution. Consequently, the interpolation and deep learning test results in the TX model vary considerably.

On the other hand, CGAN can learn to generate realistic data points that match the available data. Nevertheless, the performance of the CGAN model heavily depends on the quantity and quality of the training data. The size and information blocked can impact the quality of the training data. Small radar beam blockages may not affect the quality of the training data, while large radar beam blockages can reduce the quality of the training data, leading to less accurate calibration results. Therefore, the extent and severity of beam blockage and the precipitation distribution within the radar data determine the correction performance. It is crucial to consider these factors when choosing the appropriate correction method for a particular application. The fusion model discussed in the previous subsection can also be beneficial in this regard.

6.2 Conclusion

The seamless continuation of weather radar imagery is a crucial aspect of data processing and quantitative applications. However, the correction of beam blockages that occur frequently and can result in missing or low-quality data has not been thoroughly investigated. In this study, we have developed and executed a beam blockage correction system based on conditional adversarial neural networks. We trained our dual model using stratiform precipitation data from Northern California (collected by KDAX radar) and convective precipitation observations over the Dallas-Fort Worth area in northern Texas (collected by KFWS radar), and conducted cross-model testing. A conventional linear interpolation approach is also employed during the testing phase for comparison purposes. The major findings of this study are summarized as follows.

1) From a visual standpoint, the CGAN-based model for correcting radar beam blockage surpasses conventional interpolation models in terms of its correction efficacy. The correction performance generally declines in the following order: CGAN models trained using similar precipitation events, crossover models trained using distinct precipitation events, and linear interpolation models. In contrast, the TX model is more comprehensive, enabling improved correction of precipitation features and cloud edge prediction in diverse test sets not only in TX but also in CA. Meanwhile, the CA model exhibits suboptimal performance in correcting large-angle beam blockage with notable precipitation differences.

2) From the evaluation results, it is concluded that the correction efficacy has a non-linear correlation with the beam blockage size, and the superiority of the CGAN model becomes increasingly prominent as the threshold on precipitation intensity (in terms of radar reflectivity) and blockage size increase.

3) The disparity between the performance of the CGAN model and that of the interpolation model is considerably greater during convective precipitation cases than stratiform cases.

6.3 Future Work

Despite the favorable outcomes attained by the PBB correction technique, there are still several possibilities for extending and enhancing its performance.

Incorporating Precipitation Type Analysis and Fusing the CGAN Model: In order to address the challenge cases where convective and stratiform precipitation coexist, we plan to incorporate precipitation type analysis and fuse the CGAN model. This will involve analyzing the type of precipitation and using this information to guide the correction process. By understanding the differences between convective and stratiform precipitation, we can tailor the correction method to better account for the beam blockage effects in each case. Furthermore, we plan to fuse the CGAN model with other models to improve its overall performance. This will involve integrating the CGAN model with other models that can provide additional information about precipitation, such as temperature, humidity, and wind speed. By combining different models, we can improve the accuracy of the correction process and provide more reliable estimates of precipitation.

Adopting Novel Pseudo-Radar Observation Filters and Measuring Metrics: To refine the correction efficacy, we plan to adopt novel pseudo-radar observation filters and measuring metrics. Pseudo-radar observation filters are designed to simulate radar observations based on other types of data, such as satellite imagery or numerical weather prediction models. By using these filters, we can improve the quality of the data used for the correction process and reduce the impact of beam blockage effects. Measuring metrics, on the other hand, are used to evaluate the performance of the correction method. By developing new and more sophisticated measuring metrics, we can better understand the strengths and weaknesses of the correction process and identify areas for improvement.

Deploying the Beam Blockage Correction Model to Real-World Occluded Areas: Perhaps the most important step in improving the beam blockage correction method is to deploy it to real-world occluded areas. In particular, we plan to deploy the model to mountain regions west of the KDAX radar, where beam blockage effects are particularly severe. By comparing the corrected radar data to ground truth measurements, we can quantify the improvement in precipitation esti-

mation product based on the corrected radar data. This will allow us to evaluate the effectiveness of the correction method and identify areas for further improvement. Additionally, by deploying the model to real-world occluded areas, we can gain a better understanding of the practical challenges associated with the correction process and develop more robust and reliable methods for addressing these challenges.

Bibliography

[1] Jeffrey Snyder, Howard Bluestein, Guifu Zhang, and Stephen J. Frasier. Attenuation correction and hydrometeor classification of high-resolution, x-band, dual-polarized mobile radar measurements in severe convective storms. *Journal of Atmospheric and Oceanic Technology*, 27:1979–2001, 2009.

[2] Yabin Gou and Haonan Chen. Combining radar attenuation and partial beam blockage corrections for improved quantitative application. *Journal of Hydrometeorology*, 22(1):139–153, 2021.

[3] J Vivekanandan, David N Yates, and Edward A Brandes. The influence of terrain on rainfall estimates from radar reflectivity and specific propagation phase observations. *Journal of Atmospheric and Oceanic Technology*, 16(7):837–845, 1999.

[4] Haonan Chen and Venkatachalam Chandrasekar. The quantitative precipitation estimation system for dallas–fort worth (dfw) urban remote sensing network. *Journal of Hydrology*, 531:259–271, 2015.

[5] V. Chandrasekar, Haonan Chen, and Brenda Philips. Principles of high-resolution radar network for hazard mitigation and disaster management in an urban environment. *Journal of the Meteorological Society of Japan. Ser. II*, 96A:119–139, 2018.

[6] Rob Cifelli, V. Chandrasekar, L. Herdman, D. D. Turner, A. B. White, T. I. Alcott, M. Anderson, P. Barnard, S. K. Biswas, M. Boucher, J. Bytheway, H. Chen, H. Cutler, J. M. English, L. Erikson, F. Junyent, D. J. Gottas, J. Jasperse, L. E. Johnson, J. Krebs, J. van de Lindt, J. Kim, M. Leon, Y. Ma, M. Marquis, W. Moninger, G. Pratt, C. Radhakrishnan, M. Shields, J. Spaulding, B. Tehranirad, and R. Webb. Advanced quantitative precipitation information: Improving monitoring and forecasts of precipitation, streamflow, and coastal flooding in the san francisco bay area. *Bulletin of the American Meteorological Society*, 2022.

[7] Irene Crisologo, Robert A Warren, Kai Mühlbauer, and Maik Heistermann. Enhancing the consistency of spaceborne and ground-based radar comparisons by using beam blockage fraction as a quality filter. *Atmospheric Measurement Techniques*, 11(9):5223–5236, 2018.

[8] Pascal Getreuer. Linear methods for image interpolation. *Image Processing On Line*, 1:238–259, 2011.

[9] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999.

[10] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[11] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[13] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. *Advances in neural information processing systems*, 28, 2015.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[15] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.

[16] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018.

[17] Daifeng Peng, Yongjun Zhang, and Haiyan Guan. End-to-end change detection for high resolution satellite images using improved unet++. *Remote Sensing*, 11(11):1382, 2019.

[18] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[19] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6721–6729, 2017.

[20] Mark E Weber. Advances in operational weather radar technology. *Lincoln Laboratory Journal*, 16(1):9, 2006.

[21] Guifu Zhang. *Weather radar polarimetry*. Crc Press, 2016.

[22] V N Bringi and V Chandrasekar. *Polarimetric Doppler weather radar: principles and applications*. Cambridge university press, 2001.

[23] Peter Meischner. *Weather radar: principles and advanced applications*. Springer Science & Business Media, 2005.

[24] Alexander Ryzhkov, Malte Diederich, Pengfei Zhang, and Clemens Simmer. Potential utilization of specific attenuation for rainfall estimation, mitigation of partial beam blockage, and radar networking. *Journal of Atmospheric and Oceanic Technology*, 31(3):599–619, 2014.

[25] Jian Zhang, Kenneth Howard, Carrie Langston, Brian Kaney, Youcun Qi, Lin Tang, Heather Grams, Yadong Wang, Stephen Cocks, Steven Martinaitis, et al. Multi-radar multi-sensor (mrms) quantitative precipitation estimation: Initial operating capabilities. *Bulletin of the American Meteorological Society*, 97(4):621–638, 2016.

[26] Bushra N Alsunbuli, Widad Ismail, and Nor M Mahyuddin. Convolutional neural network and kalman filter-based accurate csi prediction for hybrid beamforming under a minimized blockage effect in millimeter-wave 5g network. *Applied Nanoscience*, pages 1–22, 2021.

[27] Arthur Y Hou, Ramesh K Kakar, Steven Neeck, Ardeshir A Azarbarzin, Christian D Kummerow, Masahiro Kojima, Riko Oki, Kenji Nakamura, and Toshio Iguchi. The global precipitation measurement mission. *Bulletin of the American meteorological Society*, 95(5):701–722, 2014.

[28] Zuzana Rulfová and Jan Kyselỳ. Disaggregating convective and stratiform precipitation from station weather data. *Atmospheric research*, 134:100–115, 2013.

[29] L Rivas Soriano, JM Sánchez Llorente, A González Zamora, and F de Pablo Dávila. Influence of land cover on lightning and convective precipitation over the european continent. *Progress in Physical Geography: Earth and Environment*, 43(3):352–364, 2019.

[30] Peter Berg, Christopher Moseley, and Jan O Haerter. Strong increase in convective precipitation in response to higher temperatures. *Nature Geoscience*, 6(3):181–185, 2013.

[31] Ernesto Jáuregui and Ernesto Romales. Urban effects on convective precipitation in mexico city. *Atmospheric Environment*, 30(20):3383–3389, 1996.

[32] Daniela Rezacova, Zbynek Sokol, and Petr Pesice. A radar-based verification of precipitation forecast for local convective storms. *Atmospheric Research*, 83(2-4):211–224, 2007.

[33] Ana M Ruiz-Leo, E Hernández, Sara Queralt, and G Maqueda. Convective and stratiform precipitation trends in the spanish mediterranean coast. *Atmospheric Research*, 119:46–55, 2013.

[34] Guillaume Penide, Alain Protat, Vickal V Kumar, and Peter T May. Comparison of two convective/stratiform precipitation classification techniques: Radar reflectivity texture versus drop size distribution–based approach. *Journal of Atmospheric and Oceanic Technology*, 30(12):2788–2797, 2013.

[35] Haonan Chen, Robert Cifelli, and Allen White. Improving operational radar rainfall estimates using profiler observations over complex terrain in northern california. *IEEE Transactions on Geoscience and Remote Sensing*, 58(3):1821–1832, 2019.

[36] Lei Han, Jianchang Zhang, Haonan Chen, Wei Zhang, and Shun Yao. Toward the predictability of a radar-based nowcasting system for different precipitation systems. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[37] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002.

[38] C-C Jay Kuo. Understanding convolutional neural networks with a mathematical model. *Journal of Visual Communication and Image Representation*, 41:406–413, 2016.

[39] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[40] Radford M Neal. *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, ON, Canada, 1993.

[41] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[42] Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural computation*, 4(6):863–879, 1992.

[43] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[44] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[45] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

[46] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[47] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[48] Shun Yao, Haonan Chen, Elizabeth J Thompson, and Robert Cifelli. An improved deep learning model for high-impact weather nowcasting. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:7400–7413, 2022.

[49] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

[50] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[51] Thorwald HM Stein, Chris E Holloway, Isabelle Tobin, and Sandrine Bony. Observed relationships between cloud vertical structure and convective aggregation over tropical ocean. *Journal of Climate*, 30(6):2187–2207, 2017.