ThOR: Thunderstorm Observations and Radar Nowcasting with Generative Deep Learning, a case study in Ho Chi Minh City

Ta Gia Khang*
Falculty of Computer Science and Engineering
Ho Chi Minh city University of Technology
IVS Joint Stock Company
Ho Chi Minh City
tagiakhang19@gmail.com

Nguyen Van Gia Thinh
Institute of Mathematical and Computational Sciences
Ho Chi Minh city University of Technology
Ho Chi Minh City
nguyenvangiathinh@hcmut.edu.vn

Tran Van Hoai*
Faculty of Computer Science and Engineering
Ho Chi Minh city University of Technology
Ho Chi Minh City
hoai@hcmut.edu.vn

Phan Thanh An
Institute of Mathematical and Computational Sciences
Ho Chi Minh city University of Technology
Ho Chi Minh City
thanhan@hcmut.edu.vn

Pham Thanh Huu
Falculty of Applied Science
Ho Chi Minh city University of Technology
IVS Joint Stock Company
Ho Chi Minh City
ADAI LAB, Nha Trang City, Vietnam
pthuu@indivisys.jp

Nguyen Minh Bao An
Falculty of Computer Science and Engineering
Ho Chi Minh city University of Technology
IVS Joint Stock Company
Ho Chi Minh City
an.nguyenmichael@hcmut.edu.vn

***Abstract*—Quantitative precipitation nowcasting using radar reflectivity images is a challenging and crucial task for accurate weather forecasting. While state-of-the-art deep learning models excel in short-term, high-resolution predictions, their operational usability is often limited by a lack of interpretability and the loss of high-frequency features due to mean error optimization. This paper explores the enhancement of Convolutional Neural Networks (CNNs) combined with Gating Recurrent Networks (LSTM) and a visual Attention techniques for quantitative precipitation nowcasting (QPN) using radar reflectivity images. By employing a Convolution-LSTM with Axial Attention as the generator, we propose a GAN-based model incorporating soft physical constraints to develop more resilient and adaptive forecasting systems. Our findings reveal the potential of advanced neural network architectures to significantly improve short-term weather predictions. The models are trained and validated using real-world data from the Doppler radar in Nha Be, Ho Chi Minh City, ensuring practical relevance and applicability. Results show that combining physical prior knowledge with deep learning techniques yields high metric scores, underscoring the effectiveness of this approach.**

***Index Terms*—Thunderstorm nowcasting, Doppler radar, CAPPI representation, Convolutional-LSTM, Axial Attention, Generative Adversarial Networks, Physics-informed Neural Network.**

Significant Statement

Thunderstorms pose significant risks to urban areas, affecting safety, infrastructure, and daily life. Effective nowcasting can mitigate these risks by providing timely warnings when high radar signal predictions occur. Our study leverages generative adversarial deep learning models to enhance thunderstorm nowcasting accuracy for a specific region in Ho Chi Minh City. We propose a many-to-one method using a generator with backbone from ConvLSTM and Axial Attention to forecast thunderstorm radar reflectivity images at ten-minute intervals. Our findings show that the network can accurately predict the motion, growth, and decay of thunderstorms. When properly tuned with sufficient data, it effectively determines the probability of thunderstorm movement. This capability supports more informed decision-making regarding short-term thunderstorm risks, particularly in regions prone to extreme weather, such as Southern Vietnam.

I. Introduction

Climate change has intensified extreme weather phenomena, making thunderstorms increasingly dangerous and unpredictable. In Vietnam, natural disasters in 2022 resulted in 175 fatalities and economic losses of about 19,452 billion VND. [1] Weather nowcasting—short-term predictions of atmospheric conditions—plays a crucial role in decision-making for agriculture, disaster management, and infrastructure planning. Quantitative precipitation nowcasting (QPN) is considered to be one of the most challenging aspects of this field. Traditional forecasting methods often struggle with complex weather patterns, especially in regions with diverse topography [2] [3] [4].

Machine learning (ML) approaches to nowcasting can be categorized as *object-based* and *grid-based*. Object-based nowcasting uses geometric constraints to detect storm objects and predict their development, while grid-based nowcasting operates directly on gridded data, avoiding complexities of storm detection but requiring advanced ML methods for predicting thunderstorm cell motion. Recent studies have employed techniques such as Convolutional Neural Networks (CNNs) [5] [6] [7], Decision Trees [8] [2] [9], and other ML methods for lightning nowcasting.

Recent advances in computer vision (CV) have significantly improved the application of deep-learning (DL) techniques to quantitative precipitation nowcasting (QPN). For instance, Klocek et al. [10] introduced a convolutional LSTM network with an encoder-forecaster structure, which outperformed

*Corresponding Author.

baseline models by 20-25 percent on various metrics. They used weather radar data to forecast radar reflectivity over the next 6 hours. Ravuri et al. [11], in collaboration with the British National Weather Service, employed a deep generative model for accurate precipitation nowcasting up to 90 minutes. Ning et al. [12] applied a multi-input, multi-output approach, which showed superior performance over recurrent strategies in radar echo predictions. Veillette et al. [13] utilized Unet to reconstruct and extend vertical liquid water content data beyond weather radar coverage. Despite these advances, the lack of interpretability in DL models reduces their trustworthiness among meteorologists. Additionally, the use of mean error loss functions in regression formulations often smooths out important high-frequency features, which are crucial during storm events, and performance tends to decline with longer forecast times.

A key limitation of these deep learning models is their lack of interpretability, which often leads to predictions that may not align with fundamental physical laws, especially in scenarios not seen during training. Physics-informed machine learning (PIML) [14] has emerged as a promising approach to address this issue by enhancing physical consistency in weather and climate forecasting [15]. In this paper, we propose a soft-PIML model that integrates deep generative models with physical priors for precipitation, aiming to produce forecasts that are both accurate and physically consistent. In summary, our main contribution in this paper includes:

- We visualized raw radar signals into image representations as CAPPI radar products for prediction purposes.
- We implemented a lightweight GAN-based neural network called ***ThOR***, using a combination of ConvLSTM with Axial Attention for the generator.
- We modified the GAN's initial loss function by adding physical constraints and a special technique called *intensity classification loss*, then trained and fine-tuned the model using real data from Ho Chi Minh City.

This paper is organized as follows: A brief overview of related research on radar reflectivity image forecasting using traditional and machine learning approach and some physics-informed method is presented in Section 2. In Section 3, we introduce the data source and describe the method of converting raw data signal into image representation. ThOR's main network architecture and the technique of loss function modification for precipitation nowcasting is described in Section 4. Section 5 describes the conducted experiments and the obtained results. Lastly, this paper end with some conclusive remarks in Section 6.

II. Related Works

A. Radar Meteorological Data

1) Meteorological data

In weather monitoring, Data Doppler is utilized to measure the velocity of precipitation particles, such as raindrops or snowflakes, allowing meteorologists to track storm movement and intensity with high precision. The radar will rotate 360° and continuously switch between transmitting and receiving states. After the radar completes one full rotation, it tilts up to the next elevation angle and continues to scan 360° relative to the ground to detect objects at different heights. These angles are called elevation angles. Typically, the radar will have a fixed set of elevation angles [7].

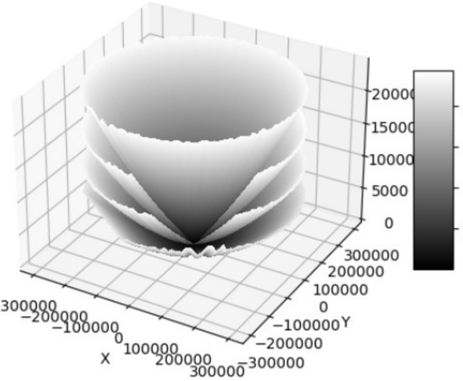

Fig. 1. Represents four-time scan with frequency value of Nha Be's Weather Radar data [7]

Doppler radar differs from conventional radar in that the Doppler effect not only gives the distance between the target and the observer, but also measures the speed of the target. It is possible to detect the change in signal length between pulses, measuring the difference in electromagnetic wave frequency due to the movement of the target. This information allows the relative speed of the target to be estimated. For many different purposes, knowing the relative speed of the observed target compared to a fixed point on the ground is very useful.

In summary, to get data within a fixed volume, the radar will sweep across multiple elevation angles. For each elevation angle, the radar will rotate 360 degrees and perform the transmit and receive processes at a fixed elevation angle relative to the ground to capture the following data fields:

- Reflectivity (dBZ).
- Velocity of objects.

By using those values from radar, a thunderstorm cell is defined as a continuous region of reflectivity values in three-dimensional space that exceed a threshold T_z and must ”reasonably” match from one scan to the next. Depending on the value of T_z, different types of storms can be identified. To distinguish various types of storms, Tz typically has the following values:

- $T_z = 40 - 50$ dBZ: Individual convective cells.
- $T_z = 30 - 40$ dBZ: Convective storms.
- $T_z = 25 - 35$ dBZ: Mesoscale convective com-plexes.
- $T_z = 15 - 25$ dBZ: Snow bands.

In this study, we will only use the reflectivity numerical values from raw data to generate 2D reflectivity images for prediction purpose later.

2) CAPPI representation

The Constant Altitude Plan Position Indicator (CAPPI) provides a two-dimensional representation of radar data by extracting a horizontal slice from three-dimensional volume scans at a constant altitude. Due to the curvature of the Earth and the radar's minimum elevation angle of approximately 0.5°, radar data beyond a certain range (around 100 km) will be positioned above the selected low CAPPI altitudes, such as 1 to 3 km. For low altitudes, as the range increases, the CAPPI data is derived from the lowest elevation angles, resulting in data from higher altitudes. Conversely, for ranges close to the radar (less than about 40 km), most of the data is from below the selected height, with data being collected from the highest elevation angles. Thus, at these shorter ranges, the data height increases with range.

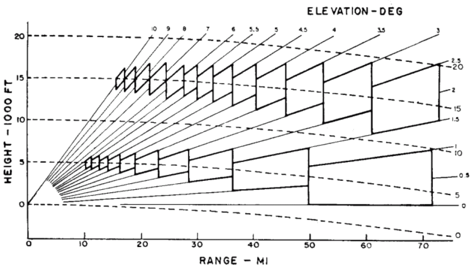

Fig. 2. Lines of constant altitude (dashed) adjusted for the Earth's curvature and normal refraction, plotted against range. This diagram illustrates how a CAPPI is derived [16].

B. Radar Precipitation Nowcasting Techniques

1) Tradditional Radar Nowcasting

Traditional weather forecasting methods have long relied on numerical weather prediction (NWP) models, which solve partial differential equations (PDEs) representing atmospheric dynamics. These models provide forecasts based on the physical laws governing the atmosphere, but their accuracy is often constrained by computational limitations and the need for high-resolution data.

The TITAN (Thunderstorm Identification, Tracking, Analysis, and Nowcasting) system represents a significant advancement in meteorology, especially in thunderstorm prediction and analysis. By leveraging sophisticated radar data, TITAN offers detailed identification and tracking of storm cells. It utilizes reflectivity and velocity data to analyze storm structures, monitor their movement, and forecast their future behavior. TITAN excels at classifying hydrometeors and detecting severe weather phenomena like tornadoes and hail, which enhances the accuracy of weather forecasts and severe weather alerts. Its real-time analysis and nowcasting capabilities allow meteorologists to issue timely warnings, improving public safety and preparedness during severe weather events. However, TITAN's reliance on radar data means that its effectiveness can be limited in areas with sparse radar coverage or in cases of extreme weather where radar data may be less reliable. [2], [3]

The pySTEP (Python Storm Tracking and Evaluation Package) [17] complements traditional radar nowcasting systems by offering advanced tools for storm tracking and evaluation. This system enhances the capabilities of radar-based models by providing more refined tracking algorithms and evaluation metrics. pySTEP integrates seamlessly with radar data, allowing for detailed analysis of storm progression and behavior. Its features include sophisticated tracking algorithms that improve the accuracy of storm forecasts and better evaluation of storm characteristics. By incorporating pySTEP into meteorological workflows, researchers and meteorologists can achieve more precise tracking and forecasting, further advancing our ability to predict and respond to severe weather events. However, the pySTEP's performance can be affected by the quality and resolution of input data, and it may require significant computational resources for processing and analysis.

2) Deep Learning for Reflectivity Radar Image Prediction

Recent advancements in artificial intelligence have significantly advanced data-driven extrapolation in quantitative precipitation nowcasting (QPN) and radar nowcasting, surpassing traditional methods. The surge in computational resources, including both hardware (e.g., graphics processing units - GPUs) and software (e.g., open-source deep learning libraries), has driven the development of deep learning (DL)-based spatio-temporal prediction models. Notable progress has been achieved in grid-based machine learning (ML) techniques for thunderstorm forecasting. For example, Lin et al. (2019), Zhou et al. (2020), and Geng et al. (2021) have utilized Convolutional Neural Networks (CNNs) to enhance lightning prediction accuracy by leveraging CNNs' robust feature extraction capabilities to analyze complex weather patterns [5] [6] [7].

The Convolutional-LSTM approach for radar image nowcasting was introduced by S. Klocek et al. [10]. To predict future frames, it is essential to capture both the spatial structure within frames and its temporal evolution. Convolutions in RNN-based models are particularly effective for extracting spatial correlations, motivating the use of Convolutional LSTM (ConvLSTM) units. Shi et al. [4] proposed incorporating convolutions into LSTM networks, leading to the following ConvLSTM formulation:

$$
\begin{aligned}
f^{(t)} &= \sigma\left(\mathbf{U}_f * X^{(t)} + \mathbf{W}_f * H^{(t-1)} + \mathbf{P}_f \odot C^{(t-1)} + \mathbf{b}_f\right) \\
i^{(t)} &= \sigma\left(\mathbf{U}_i * X^{(t)} + \mathbf{W}_i * H^{(t-1)} + \mathbf{P}_i \odot C^{(t-1)} + \mathbf{b}_i\right) \\
G^{(t)} &= \tanh\left(\mathbf{U}_g * X^{(t)} + \mathbf{W}_g * H^{(t-1)} + \mathbf{b}_g\right) \\
o^{(t)} &= \sigma\left(\mathbf{U}_o * X^{(t)} + \mathbf{W}_o * H^{(t-1)} + \mathbf{P}_o \odot C^{(t-1)} + \mathbf{b}_o\right) \\
C^{(t)} &= f^{(t)} \odot C^{(t-1)} + i^{(t)} \odot G^{(t)} \\
H^{(t)} &= o^{(t)} \odot \tanh\left(C^{(t)}\right)
\end{aligned}
$$

where $*$ denotes convolution operation, and $\odot$ denotes element-wise multiplication. Our approach use ConvLSTM blocks to extract temporal features.

Nowcasting involves predicting pixel values for the immedi-

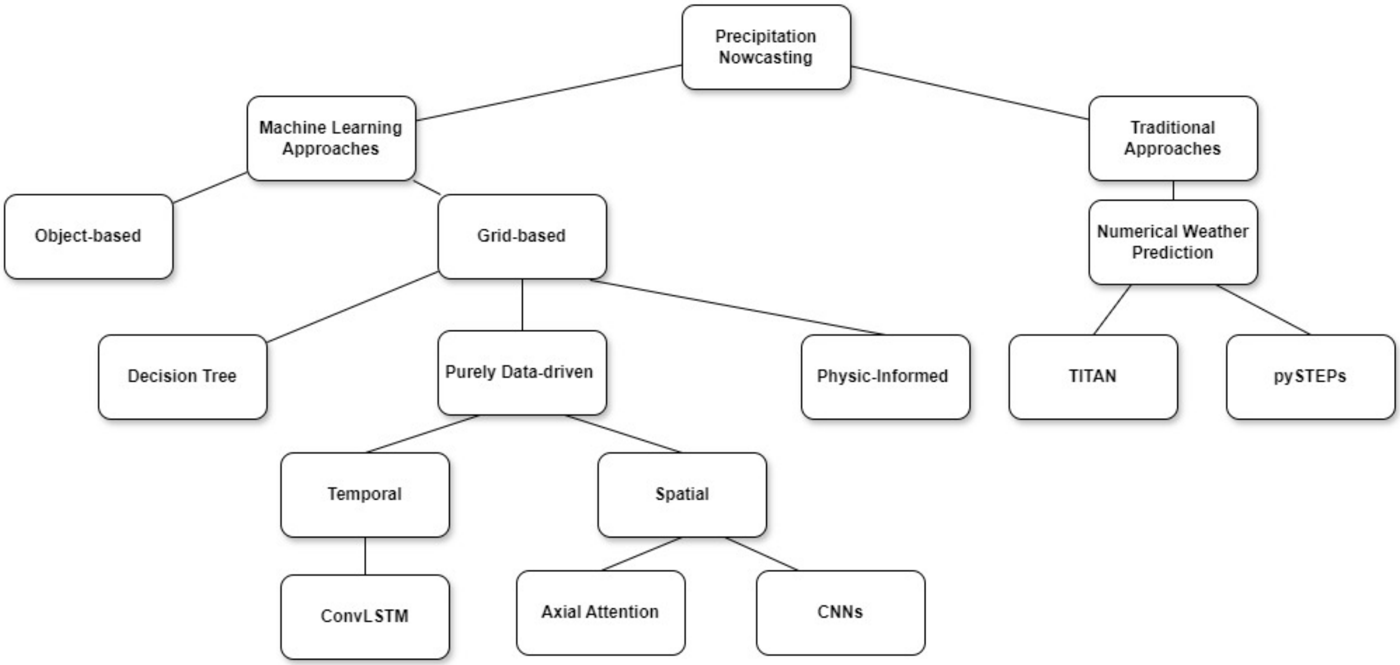

Fig. 3. An overview of some approaches for precipitation nowcasting.

ate future, such as the next 10 minutes, using past precipitation radar echo images to forecast future radar frames. Our model, an encoder-decoder architecture utilizing ConvLSTM as the first core computational unit, leverages its recurrent nature to capture temporal information in radar echoes [5] [4].

Xie et al. [18] introduced Axial Attention to address the quadratic complexity of traditional self-attention mechanisms when applied to high-dimensional data like radar images. By decomposing the attention operation along individual dimensions as each row or column, Axial Attention reduces complexity to linear per dimension. This factorization enhances the model's ability to capture long-range dependencies in radar image sequences, improving prediction accuracy while maintaining computational efficiency. Our experiments demonstrate that integrating Axial Attention significantly enhances the model's ability to predict the motion, growth, and decay of thunderstorms, making it a valuable tool for nowcasting in regions prone to extreme weather [19].

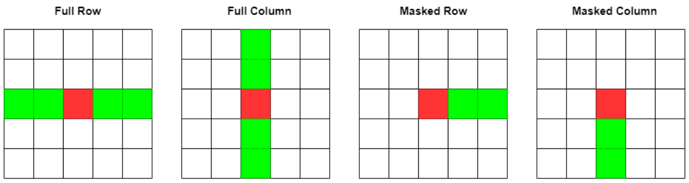

Fig. 4. Types of axial attention layers, the building blocks of the Axial Transformer. The blue locations represent the receptive field of the output red location [19].

Generative Adversarial Networks (GANs), recognized for their state-of-the-art performance in various fields, have demonstrated significant adaptability in recent years, proving their value in precipitation nowcasting tasks. For example, the Adversarial Extrapolation Neural Net (AENN) [20], a GAN variant with two discriminators, has shown superior performance in weather radar echo extrapolation compared to numerical weather prediction approaches. Another study [21] proposed a GAN utilizing both temporal and spatial discriminators, highlighting GANs' potential to enhance precipitation nowcasting accuracy through data generation techniques.

The GAN-based models consist of two networks: generator G and a discriminator D, which are trained through adversarial processes. The generator G creates data samples resembling a given distribution, while the discriminator D distinguishes between real samples from the true distribution and fake samples generated by G. The training process is framed as a minimax game, where the generator minimizes the following objective function:

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

where $p_{\text{data}}(x)$ denotes the real data distribution, $p_z(z)$ is the prior distribution over the latent space, $G(z)$ is the generated sample, and $D(x)$ is the probability that x is a real sample. This adversarial loss encourages the generator to produce increasingly realistic samples while improving the discriminator's ability to differentiate between real and fake samples, enhancing generative performance over time. We modeled our architecture with the shape of traditional GAN network, with a special Generator and one Discriminator.

Despite advancements, GANs face challenges such as delivering consistently clear and accurate forecasts and generalizing across diverse weather conditions, especially those not well-represented in training datasets. Additionally, the interpretability of sophisticated neural networks like AENN remains a significant challenge. This lack of *interpretability* is the motivation for us to make some modification by adding some physics contraints.

Physical constraints in neural networks have been a long-standing challenge. Two types of constraints are recognized: soft and hard constraints. Soft constraints are incorporated into the loss function of Physics-Informed Neural Networks (PINNs) [22] by embedding partial differential equations (PDEs) and their initial and boundary conditions. In contrast, hard constraints are enforced by designing custom neural networks that inherently satisfy these conditions while embedding the PDEs in the loss function. This study adopts the former approach, adjusting it for atmospheric applications. PINNs, as universal approximators, can solve various types of equations, including fractional equations, integral-differential equations, and stochastic PDEs. By encoding the governing physical equations and augmenting the loss function with a residual term derived from the governing equation, PINNs offer a way to integrate process-based numerical weather prediction with deep learning models. Geiss and Hardin [23] demonstrated the integration of known physics and physical constraints in neural networks for super-resolution tasks. Furthermore, hybrid models, which are more interpretable, hold promise for addressing current atmospheric and climate issues [24]. In our study, we modified the model's loss function to incorporate soft physical constraints relevant to initial conditions.

III. Data analysis

A. Study area

The data used for this study is provided by the Nha Be Weather Radar Station of Ho Chi Minh City. This historical data consist of 10980 sequences spans from January 1, 2023, to March 31, 2023. For different angles of elevation, the radars perform a sweep of revolution, each of which lasts for 10 minutes [7].

B. Data preprocessing

1) Generating Image Data from Raw Data

The collected data is stored in the SIGMET (Significant Meteorological Information) file format, which provides the reflectivity, velocity, and spectrum width at various elevation angles, is first preprocessed to filter out noise and correct for any inconsistencies. Next, the data is interpolated onto a three-dimensional Cartesian grid using a suitable interpolation method, such as inverse distance weighting or kriging, to ensure spatial consistency. From this gridded data, a horizontal slice at a constant altitude is extracted to create the CAPPI. This slice provides a two-dimensional view of the radar data at a specific altitude, allowing for detailed analysis of atmospheric phenomena such as precipitation patterns, storm structures, and wind fields. The CAPPI image is crucial for meteorologists and researchers as it offers a clear and comprehensive visualization of weather events, aiding in better understanding and forecasting of meteorological conditions. The explanation of the methods mentioned in the paper by Dixon, M [2].

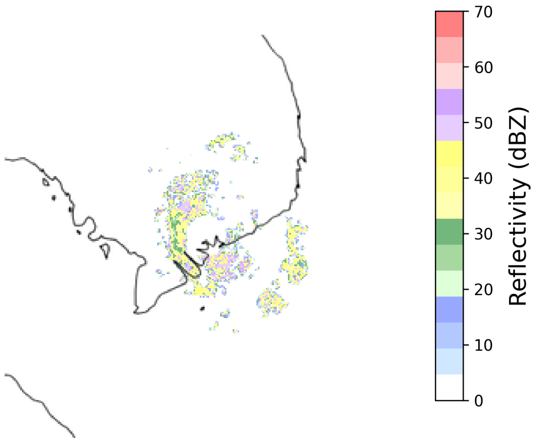

Fig. 5. A visual example of CAPPI representation from RAW radar data after a color mapping step

The raw data is subsequently processed using the PyART [25] software. This preprocessing facilitates the extraction of essential information, which is then transformed into grayscale images for use as inputs to our models. The dataset contains 10980 radar reflectivity maps of 20 elevations, equally spaced between 0.5km and 20km, covering a forecast area of 200km × 200km. For our study, we only chose the data of the elevation 2km to 3km because it contains most Radar radio response in the study time period. Each sequence in the dataset has 144 time steps recorded in 24 hours with an interval of 10 minutes. Each image is of size 200 × 200 and is resized to 128 x 128 for cheaper computation purpose and the reflectivity values are converted into gray-scale (between 0 and 255).

2) Enhance data quality for prediction model

Initially, our image data in black-white exhibits a characteristic where regions with lower pixel intensities (darker shades) correspond to higher dBZ values, leading to predominantly white areas in the original imagery. Recognizing that such a distribution could potentially compromise the clarity and fidelity of predictive outcomes, we conducted several experiments. Through these trials, we observed that inverting the images—altering bright areas to dark and vice versa—proved effective in enhancing the quality of the predicted images.

This approach addresses the initial imbalance and ensures that the predictive models can capture and interpret visual features more effectively, ultimately improving the robustness and accuracy of our findings. Finally, to be used by our model, the data will be stack together, create a sequence of images. Our proposed model will take a sequence of 5 images which take in 50-minute and predict the next 10-minute image of radar signal.

TABLE I
DETAILED INFORMATION OF NHA BE DATA

Name	The physical variable described by the data	Spatial resolution	Image Size	Temporal resolution
Nha Be Data	Radar Echo	1km	128x128	10 minutes

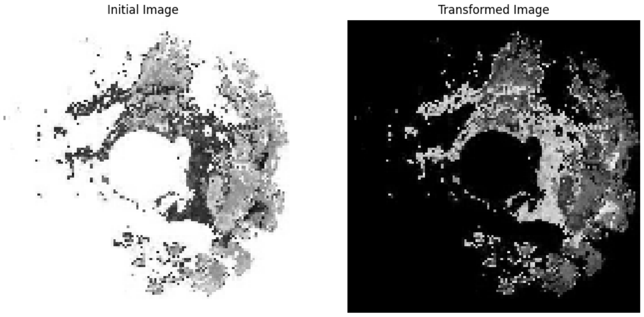

Fig. 6. The initial image (left) and the image after preprocessing (right).

IV. PROPOSED METHODS

A. Problem Formula

Given an input spatio-temporal sequence of N frames $\boldsymbol{x}_{\text{in}} \in \mathbb{R}^{N\times H\times W\times C}$, where H and W denote the spatial resolution, and C represents the image channels or different types of measurements (e.g., radar maps, humidity maps, etc.), the goal is to predict the next M precipitation maps $\boldsymbol{x}_{\text{out}} \in \mathbb{R}^{M\times H\times W\times 1}$ and classify extreme events based on a defined threshold. Throughout this case study, we use $C = 1$, $N = 5$, and $M = 1$.

The raw data from the Nha Be Meteorological Station is processed into image-like sequences for our model. These sequences consist of 5 images captured at 10-minute intervals over a 50-minute period. Our model is designed to take this sequence and predict the radar reflectivity image for the next 10-minute interval.

B. The ThOR pipeline

1) Overall Architecture

The ThOR pipeline is based on a GAN architecture, consisting of two main components: the generator and the discriminator. The overall structure of the architecture is illustrated in Fig. 7.

In the generation phase, a batch of input precipitation images, $\mathbf{X}_{in} \in \mathbb{R}^{B\times 5\times 128\times 128}$ (where B represents the batch size), is passed through the generator, which extracts latent features to predict the next time step. The specific details of the generator will be discussed in the following section. The generated frame is then processed by the discriminator, which consists of three layers of 4x4 convolutional filters and utilizes the LeakyReLU activation function with Dropout-rate of 0.2. By training the generator and discriminator together, the model produces realistic radar CAPPI products for the next time step.

2) Generator

Our methodology is founded on the principle that relying solely on either *temporal* or *spatial* features is insufficient for building a robust model. We aim to leverage both types of latent features, integrating them to generate the final output.

At the first core of our architecture is the ConvLSTM, inspired by the encoder-decoder design in Shi et al. [4]. However, our approach diverges by employing a stack of three ConvLSTM blocks followed by three 3D Convolution layers to extract temporal features. This modification represents a unique contribution of our model. To preserve the input frame's resolution throughout the network, we apply padding at every convolution stage. The ConvLSTM layers use kernel sizes of 3 × 3, and the three 3D convolutional layers map these feature maps to predict the next image, shifted by one time step. ReLU activation is applied between layers to introduce non-linearity, assisting in the recognition of complex patterns.

The incorporation of an Axial Attention block allows our model to better capture long-range dependencies within radar image sequences, enhancing the prediction accuracy of subsequent frames while maintaining computational efficiency. This is especially beneficial in our context, where handling large-scale radar data is crucial. Axial Attention improves the model's ability to extract spatial features without sacrificing performance. By factorizing this attention mechanism across different dimensions, the model maintains high accuracy in predictions while operating more efficiently.

The output of the ConvLSTM blocks is passed into the Axial Attention block, and is concatenated with the output from this Attention block during the decoding step. An upsampling block, composed of two Conv3D layers with ReLU activation and Dropout in between, generates the final output by combining both types of latent features. A Sigmoid activation function is applied at the end of the generator to constrain the pixel output values within the range [0,1]. The full architecture of our Generator is shown in Fig. 8 while the upsampling block is shown in Fig.9 .

C. Loss Adjustment

1) Initial GAN Loss Function

As part of our overall GAN architecture, the initial loss function plays a critical role in training both the generator and discriminator. In this study, the losses for the generator and discriminator are designed to optimize their respective objectives. The discriminator's loss, $\mathcal{L}_D$, is based on Binary

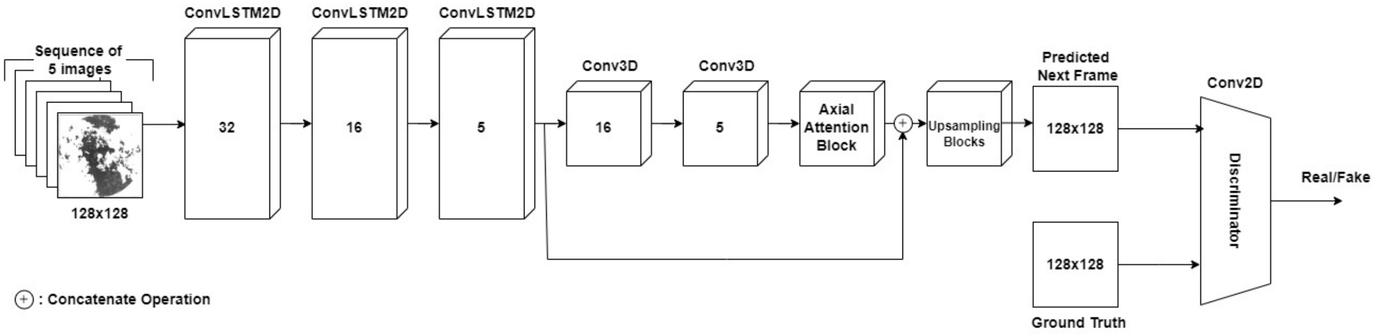

Fig. 7. Overview of the ThOR architecture as a GAN-based model

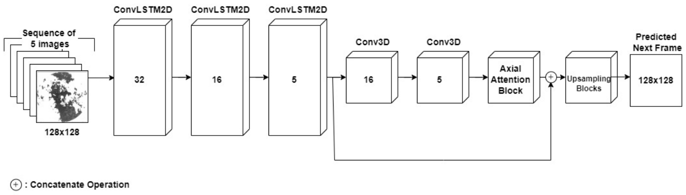

Fig. 8. Our Generator architecture

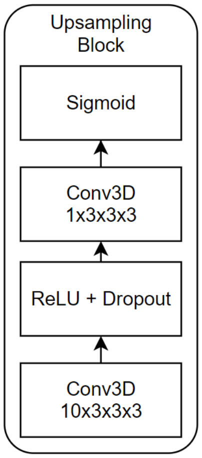

Fig. 9. The Upsampling block

Cross-Entropy (BCE), where the discriminator aims to correctly classify generated data as fake:

$$\mathcal{L}_D = \text{BCE}(\hat{y}, 0)$$

For the generator, the loss $\mathcal{L}_G$ is a combination of two components: the Mean Absolute Error (MAE), which encourages the generator to produce outputs close to the real distribution, and the BCE loss, which drives the generator to fool the discriminator by making the generated outputs be classified as real:

$$\mathcal{L}_G = \text{MAE}(G(\mathbf{x_{in}}), \mathbf{x_{out}}) + \text{BCE}(\hat{y}, 1)$$

Here, $G(\mathbf{x_{in}})$ represents the generated output, $\hat{y}$ is the discriminator's prediction, where it assigns 0 to fake data and 1 to real data.

2) Intensity Classification Loss

A key motivation for precipitation nowcasting systems is the ability to deliver high-resolution forecasts during storm events. Severe weather risks like hail, flash floods, strong winds, and lightning are tied to convective systems and intense precipitation. Accurate nowcasting, even for short durations, benefits both the public and meteorologists. Automated nowcasting offers precise forecasts and enhances severe weather alert systems. However, framing neural networks training as a regression problem often leads to models smoothing out high-frequency features and high-intensity predictions. Standard loss functions like MAE or MSE penalize incorrect predictions twice—once for misplacing storms and again for missing the correct location, leading models to prefer uncertain, smoothed predictions.

While these smoothed outputs minimize prediction errors, they lack vital details for storm events. To focus on high-intensity prediction, we propose adding a new output predicting the ”probability” of severe rainfall exceeding a range. The output $G(\mathbf{x_{in}})$ is computed using a generator where $\mathbf{x_{out}}$ is the model's original output. The total loss for a training the generator is updated as:

$$\mathcal{L}_G = \text{MAE}(G(\mathbf{x_{in}}), \mathbf{x_{out}}) + \text{BCE}(\hat{y}, 1) + \text{CE}(\text{M}(G(\mathbf{x_{in}}), \text{M}(\mathbf{x_{out}}))$$

Here, $\text{M}(\mathbf{x})$ is a trinary image obtained by a piecewise *Masked* function $\text{M}(.)$, which map from each pixel's reflectivity into label as:

$$\text{M}(\mathbf{x}) = \begin{cases} 0 \text{ if } x \in (0; 10) \\ 1 \text{ if } x \in [11; 40) \\ 2 \text{ if } x \in [40; \infty) \end{cases}$$

This $\mathcal{L}_{\text{icl}}$ (ICLoss) is calculated via cross-entropy.

3) Physical Constraint for Loss Function and Final Loss Function

The global atmosphere is governed by several fundamental conservation laws, such as mass, momentum, angular momentum, energy, and potential vorticity. However, smaller weather systems like precipitation events do not always adhere to these large-scale conservation laws. Additionally, our dataset lacks crucial variables, such as wind speed, heat, and pressure, needed to apply these conservation laws. These variables cannot be obtained from reanalysis data, as reanalysis data fails to match the 1-km resolution of radar echo data and cannot be retrieved in real-time during precipitation events. This makes direct application of conservation laws to QPN tasks challenging.

To address this, we propose applying the convection-diffusion equation from fluid mechanics as a physical constraint. The convection-diffusion equation, describing fluid flow and material transport, is given by:

$$\frac{\partial Q}{\partial t} + \nabla \cdot (QV) + D\nabla^2 Q = 0 \tag{2}$$

where Q represents the transported quantity, V is the convection velocity, and D is the diffusion coefficient. In this context, Q is replaced by reflectivity data from radar echoes. The first term describes the change of the quantity Q over time, the second term represents convection (bulk transport by velocity), and the third term represents diffusion.

Given the data limitations, we assume that the precipitation system moves at a fixed velocity within a 60-minute period. Thus, V is considered constant. The equation is discretized using a numerical method, resulting in the following expression:

$$\begin{aligned}&\frac{Q_{i,j}^{n+1} - Q_{i,j}^{n}}{\Delta t} + u\frac{Q_{i,j}^{n} - Q_{i-1,j}^{n}}{\Delta x} + v\frac{Q_{i,j}^{n} - Q_{i,j-1}^{n}}{\Delta y} \\ &+ D\frac{\left(Q_{i+1,j}^{n} - Q_{i,j}^{n}\right) - \left(Q_{i,j}^{n} - Q_{i-1,j}^{n}\right)}{\Delta x \cdot \Delta x} \\ &+ D\frac{\left(Q_{i,j+1}^{n} - Q_{i,j}^{n}\right) - \left(Q_{i,j}^{n} - Q_{i,j-1}^{n}\right)}{\Delta y \cdot \Delta y} = 0\end{aligned}$$

Here, u and v are the velocities in the x and y directions, assumed constant, while Δt, Δx, and Δy represent the time and space steps. This equation is finally discritized to:

$$\begin{aligned}\alpha\left(Q_{i,j}^{n+1} - Q_{i,j}^{n}\right) + \beta_1\left(Q_{i,j}^{n} - Q_{i-1,j}^{n}\right) + \beta_2\left(Q_{i,j}^{n} - Q_{i,j-1}^{n}\right) \\ +\gamma\left(Q_{i+1,j}^{n} + Q_{i-1,j}^{n} + Q_{i,j+1}^{n} + Q_{i,j-1}^{n} - 4Q_{i,j}^{n}\right) = 0\end{aligned}$$

where α, β_1, β_2, and γ are constants derived from the time step, spatial steps, velocity components, and diffusion coefficient. These constants are computed through a convolution layer in the neural network, similar to approaches used in approximating unknown partial differential equations (PDEs).

The final loss function is divided into four parts:

$$\begin{aligned}\mathcal{L}_G = &\, a_1 \cdot \text{MAE}(G(\mathbf{x}_{\text{in}}), \mathbf{x}_{\text{out}}) \\ &+ a_2 \cdot \text{BCE}(\hat{y}, 1) \\ &+ a_3 \cdot \text{CE}(\text{M}(G(\mathbf{x}_{\text{in}})), \text{M}(\mathbf{x}_{\text{out}})) \\ &+ a_4 \cdot \text{CD}(G(\mathbf{x}_{\text{in}}))\end{aligned}$$

where the coefficients a_1, a_2, a_3, and a_4 are determined based on the relative importance of each metric for the training set. These coefficients are chosen to balance the different terms in the loss function, ensuring that no single metric dominates the training process.

V. Experiment Results

A. Implementation Detailed

The ThOR network was implemented in Python using PyTorch [26] , NumPy (Harris et al., 2020) [3], and Matplotlib (Hunter, 2007) [27], with OpenCV for visualization.

To address overfitting in our neural network model, we applied batch normalization and dropout with a 0.2 rate between layers. Dropout enhances generalization by randomly deactivating units and their connections during training, reducing reliance on specific pathways. This strategy aims to improve the model's robustness and performance across various tasks and datasets, ensuring better generalization to unseen data.

We trained the model to predict the next time step using observational data from five past time steps, corresponding to a 50-minute past window and a 10-minute future prediction. Training on an Nvidia P100 GPU cluster node. The best model weights were saved based on validation SSIM. The validation and testing sets were shuffled, and no random data augmentation was applied. Training typically concluded after 90-100 epochs. In contrast to the training duration, model evaluation is rapid. We can generate a single sample and produce a prediction in 0.73 seconds on a modern computer with 16 CPU cores. Thus, operational use of the model does not require GPU hardware, and any delays in producing warnings are likely due to data acquisition rather than computation.

B. Numerical Metrics

This article uses four metrics to evaluate the model's performance on the test set. The *Root Mean Square Error (RMSE)* assesses the average magnitude of prediction errors, quantifying the difference between predicted and observed radar images. The *Structural Similarity Index (SSIM)* measures the perceptual similarity between images, focusing on structural aspects such as luminance, contrast, and texture. The *Mean Absolute Error (MAE)* provides a measure of prediction accuracy by calculating the average absolute errors between predicted and actual values. Additionally, the *Critical Success Index (CSI)* evaluates the model's ability to correctly identify thunderstorm events by considering the ratio of hits to the sum of hits, misses, and false alarms, offering insight into the model's operational effectiveness in forecasting thunderstorm occurrences. The formula of each metric is given below:

Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Range: $[0, \infty)$; *Optimum:* 0
RMSE measures the average magnitude of the prediction error. A lower value indicates a better fit of the model to the observed data.

Structural Similarity Index (SSIM):

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Range: $[-1, 1]$; *Optimum:* 1
SSIM evaluates the perceived quality of an image by comparing luminance, contrast, and structure between the predicted and original images. Higher SSIM means the predicted image is more structurally similar to the reference image.

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Range: $[0, \infty)$; *Optimum:* 0
MAE measures the average magnitude of the errors in a set of predictions. It calculates the average of the absolute differences between predicted and actual values.

Critical Success Index (CSI):

$$\text{CSI} = \frac{\text{hits}}{\text{hits} + \text{misses} + \text{false alarms}}$$

Range: $[0, 1]$; *Optimum:* 1
CSI measures the proportion of correct predictions (hits) out of the total number of events (hits + misses + false alarms). A higher CSI reflects better model performance in detecting events.

C. Evaluation

By dividing Nha Be Station radar data into training set, valid set and test set, we will evaluate our model using test data to assess its generalization capabilities and performance. This evaluation will involve testing the model on a dataset that it has not encountered during the training phase. By doing so, we aim to obtain a realistic measure of how well our model performs in real-world scenarios.

We will demonstrate both qualitative and quantitative results to provide a comprehensive assessment. The qualitative evaluation will involve visually inspecting the outputs to ensure they meet the desired standards and are perceptually relevant. For the quantitative evaluation, we will use the above key metrics for this evaluation, which are SSIM, RMSE, MAE, and CSI.

1) Qualitative result

Figure 10 illustrates the output of our model applied to previously unseen data. Although the model was trained on black-and-white images, we used Matplotlib to apply color mapping, enhancing the visual interpretation of the results. These figures highlight the effectiveness of our approach, showing that fine-tuning on a specific pixel-value dataset produces accurate and insightful predictions.

The results indicate that our model not only accurately predicts regions of interest but also preserves small-scale features, providing detailed reflectivity information without excessive smoothing. Additionally, our experimental setup is highly adaptable, capable of meeting a variety of requirements, further emphasizing the robustness and versatility of our methodology.

2) Quantitative result

Our results demonstrate that the ***ThOR*** network achieved an *RMSE* of 7.3654, an *MAE* of 0.752, an *SSIM* score of 0.951, and a *CSI* of 0.21. These results indicate that incorporating the Axial Attention mechanism with ConvLSTM is effective for this specific task, as it enhances the model's ability to distinguish between the presence and absence of thunderstorms in radar images. The high *SSIM* score confirms that our model not only predicts the movement and evolution of thunderstorms with high precision but also maintains the visual quality of the generated radar images, ensuring their practical applicability in real-world nowcasting scenarios.

TABLE II
METRICS SCORE FOR TEST SET

Proposed Model	**Metric**			
	RMSE	***MAE***	***SSIM***	***CSI***
ThOR Network	7.3654	0.252	0.951	0.21

VI. Conclusion

The network's capability to predict the movement and evolution of thunderstorms through convolutional recurrent layers with attention mechanisms underscores its effectiveness in nowcasting radar images. Using a local radar dataset from Ho Chi Minh City, the study demonstrates the model's ability to accurately predict radar images at 10-minute intervals based on the preceding 50 minutes of data. This allows end users to leverage probability outputs for issuing timely warnings tailored to specific thresholds, enhancing decision-making while reducing the risk of false alarms.

Additionally, the findings emphasize the model's proficiency in identifying the developmental stages of convection, including the initiation, growth, and dissipation of thunderstorms. By incorporating a physics-based constraint into the initial loss function with intensity classification, the model achieves more accurate forecasts while deepening the understanding of thunderstorm dynamics.

In conclusion, the GAN-based neural network architecture presented in this study marks a significant advancement in forecasting the spatial-temporal evolution of thunderstorm radar images. Its integration of convolutional and recurrent layers with axial attention proves critical in capturing complex weather patterns, offering valuable insights and operational benefits for meteorological applications in urban settings like Ho Chi Minh City.

Acknowledgment

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number NCM2021-20-02

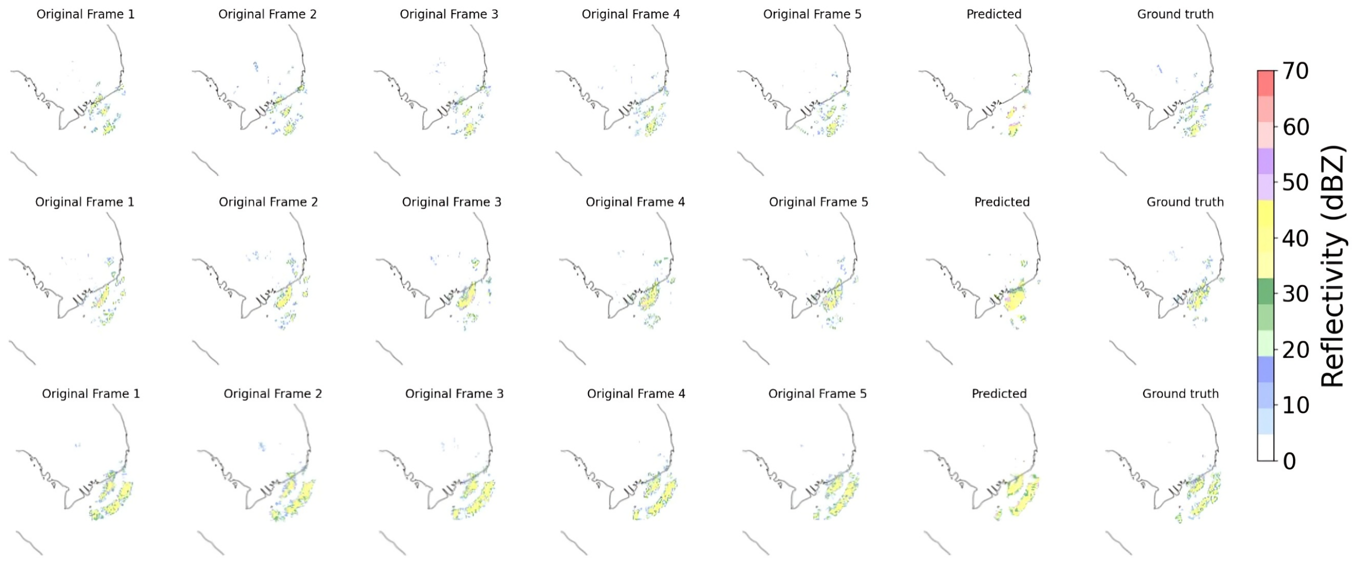

Fig. 10. The visual result of ThOR model when test with unseen data after a color mapping step

Conflicts of interest

All authors declare that they have no conflicts of interest.

References

[1] VietnamNews, “Natural disasters leave 33 dead and missing in first 6 months,” June 22nd 2023.

[2] M. Dixon, “Thunderstorm cell tracking using radar data,” *Weather Forecasting Journal*, vol. 31, no. 3, pp. 512–525, 2016.

[3] C. R. e. a. Harris, “Array programming with numpy,” *Nature*, vol. 585, pp. 357–362, 2020.

[4] X. e. a. Shi, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” *Advances in Neural Information Processing Systems*, vol. 28, pp. 802–810, 2015.

[5] W. Lin, P. Zhou, and X. Geng, “Convolutional neural networks for grid-based lightning prediction,” *Atmospheric Research*, vol. 220, pp. 141–152, 2019.

[6] P. Zhou, W. Lin, and X. Geng, “Enhancing lightning prediction using deep convolutional networks,” *Journal of Atmospheric and Oceanic Technology*, vol. 37, no. 11, pp. 2161–2173, 2020.

[7] X. Geng, P. Zhou, and W. Lin, “Deep learning for lightning nowcasting: Convolutional neural networks for high-resolution weather prediction,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 3452–3463, 2021.

[8] J. Blouin and C. La Fata, “Decision tree methods for lightning prediction in meteorological data,” *Journal of Meteorological Sciences*, vol. 23, no. 4, pp. 678–690, 2016.

[9] A. Mostajabi, J. Blouin, and C. La Fata, “Lightning nowcasting at weather stations using machine learning,” *Journal of Atmospheric and Oceanic Technology*, vol. 36, no. 9, pp. 1793–1805, 2019.

[10] S. e. a. Klocek, “Ms-nowcasting: Operational precipitation nowcasting with convolutional lstms at microsoft weather,” *arXiv:2111.09954*, 2021.

[11] S. e. a. Ravuri, “Skilful precipitation nowcasting using deep generative models of radar,” *Nature*, vol. 597, no. 7878, pp. 672–677, 2021.

[12] S. e. a. Ning, “Mimo is all you need : A strong multi-in-multi-out baseline for video prediction,” *arXiv:2212.04655*, 2022.

[13] M. Veillette, S. Samsi, and C. Mattioli, “Sevir: A storm event imagery dataset for deep learning applications in radar and satellite meteorology,” in *Proc. Adv. Neural Inf. Process. Syst*, vol. 33, 2020, pp. 22 009–22 019.

[14] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, “Integrating physics-based modeling with machine learning: A survey,” *arXiv preprint arXiv:2003.04919*, vol. 1, no. 1, pp. 1–34, 2020.

[15] K. Kashinath, M. Mustafa, A. Albert, J. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, and A. e. a. Singh, “Physics-informed machine learning: case studies for weather and climate modelling,” *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200093, 2021.

[16] C. U. C. for Atmospheric Research (UCAR), “Constant altitude plan position indicator (cappi).” [Online]. Available: https://courses.comet.ucar.edu/pluginfile.php/3786/mod_imscp/content/1/cappi.html

[17] S. Pulkkinen, D. Nerini, A. A. Pérez Hortal, C. Velasco-Forero, A. Seed, U. Germann, and L. Foresti, “Pysteps: an open-source python library for probabilistic precipitation nowcasting,” *Geoscientific Model Development*, 2019.

[18] Y.-M. Xie, Y.-L. Zhao, and S.-Y. Huang, “A deep learning model with axial attention for radar echo extrapolation,” *Applied Artificial Intelligence International Journal*, 2024.

[19] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, “Axial attention in multidimensional transformers,” *arXiv preprint arXiv:1912.12180*, 2019.

[20] J. Jing, Q. Li, X. Ding, N. Sun, R. Tang, and Y. Cai, “Aenn: A generative adversarial neural network for weather radar echo extrapolation,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 89–94, 2019.

[21] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, and S. e. a. Madge, “Skilful precipitation nowcasting using deep generative models of radar,” *Nature*, vol. 597, no. 7878, pp. 672–677, 2021.

[22] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[23] A. Geiss and J. C. Hardin, “Strictly enforcing invertibility and conservation in cnn-based super resolution for scientific datasets,” *Artif. Intell. Earth Syst.*, 2023.

[24] C. e. a. Irrgang, “Towards neural earth system modelling by integrating artificial intelligence in earth system science,” *Nature Mach. Intell.*, vol. 3, no. 8, pp. 667–674, 2021.

[25] J. J. Helmus and S. M. Collis, “pyart: A python-based radar toolkit,” *Journal of Open Source Software*, vol. 3, no. 31, p. 862, 2018.

[26] A. e. a. Paszke, “Pytorch: An imperative style, high-performance deep learning library,” in *NeurIPS*, 2019, pp. 8024–8035.

[27] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science Engineering*, vol. 9, no. 3, pp. 90–95, 2007.