DEEP LEARNING-BASED HYBRID SHORT-TERM SOLAR FORECAST USING SKY IMAGES AND METEOROLOGICAL DATA

Thesis submitted to the University of Nottingham for the degree of
Doctor of Philosophy, March 2023.

LIWENBO ZHANG

4342668

Supervised by

Wu, Prof Yupeng
Wilson, Dr Robin
Sumner, Prof Mark

Signature: ____________________

Date: March/30/2023

Abstract

The global growth of solar power generation is rapid, yet the complex nature of cloud movement introduces significant uncertainty to short-term solar irradiance, posing challenges for intelligent power systems. Accurate short-term solar irradiance and photovoltaic power generation predictions under cloudy skies are critical for sub-hourly electricity markets. Ground-based image (GSI) analysis using convolutional neural network (CNN) algorithms has emerged as a promising method due to advancements in machine vision models based on deep learning networks.

In this work, a novel deep network, "ViT-E," based on an attention mechanism Transformer architecture for short-term solar irradiance forecasting has been proposed. This innovative model enables cross-modality data parsing by establishing mapping relationships within GSI and between GSI, meteorological data, historical irradiation, clear sky irradiation, and solar angles. The feasibility of the ViT-E network was assessed the Folsom dataset from California, USA .

Quantitative analysis showed that the ViT-E network achieved RMSE values of 81.45 W/m^2, 98.68 W/m^2, and 104.91 W/m^2 for 2, 6, and 10-minute forecasts, respectively, outperforming the persistence model by 4.87%, 16.06%, and 19.09% and displaying performance comparable to CNN-based models. Qualitative analysis revealed that the ViT-E network successfully predicted

20.21%, 33.26%, and 36.87% of solar slope events at 2, 6, and 10 minutes in advance, respectively, significantly surpassing the persistence model and currently prevalent CNN-based model by 9.43%, 3.91%, and -0.55% for 2, 6, and 10-minute forecasts, respectively.

Transfer learning experiments were conducted to test the ViT-E model's generalisation under different climatic conditions and its performance on smaller datasets. We discovered that the weights learned from the three-year Folsom dataset in the United States could be transferred to a half-year local dataset in Nottingham, UK. Training with a dataset one-fifth the size of the original dataset achieved baseline accuracy standards and reduced training time by 80.2%. Additionally, using a dataset equivalent to only 4.5% of the original size yielded a model with less than 2% accuracy below the baseline. These findings validated the generalisation and robustness of the model's trained weights.

Finally, the ViT-E model architecture and hyperparameters were optimised and searched. Our investigation revealed that directly applying migrated deep vision models leads to redundancy in solar forecasting. We identified the best hyperparameters for ViT-E through manual hyperparameter space exploration. As a result, the model's computational efficiency improved by 60%, and prediction performance increased by 2.7%.

Acknowledgements

The past four and a half years have been a challenging yet rewarding journey. First and foremost, I would like to express my heartfelt gratitude to my supervisor, Professor Yupeng Wu, for his foresight, insight, guidance, and understanding. His wisdom and vision have continually guided me, both in research and in life, molding me into a more motivated and thoughtful researcher. His dedication to his work serves as the perfect example to follow, and I will be forever grateful to him.

In addition, I would like to sincerely thank my supervisor, Dr. Robin Wilson, for his patient guidance, understanding, and encouragement. His gentle yet insightful guidance has inspired me to polish my work to perfection. I would also like to extend my heartfelt appreciation to my supervisor, Professor Mark Sumner, for his profound thoughts and insights that have helped elevate my work to new heights.

I am grateful to the staff at the University of Nottingham for their support of both me and my project. I would like to give special thanks to Mr. Karl Booker and Mr. Adrian Quinn for their assistance in setting up and managing the laboratory, approving, and facilitating my intricate experimental work.

Moreover, I would like to express my gratitude to my colleagues in Professor Yupeng Wu's research group for their help and support in both my academic and personal life. I am thankful for the guidance of Dr. Yanyi Sun, Dr. Hao Gao, Dr. Xiao Liu, and Mr. Kun Du in my research. I appreciate the assistance and guidance of Dr. Milad Dakka, Dr. Jan-Frederik Flor, and Dr. Fedaa Abd-AlHamid in my experiments. I am grateful for the discussions and companionship of Dr. Dingming Liu, Yuexing Yang ,Yang Ming, Haoming Wang, and Kaiqi Yan in my work and life.

I would like to extend special thanks to Dr. Mu Li and Dr. Yi Zhu. Although we have never met in person, your selfless sharing of knowledge, perspectives, and insights on deep learning has been invaluable to my growth and development.

Finally, I would like to extend my deepest gratitude to my family. I am thankful for the love, companionship, and understanding of my fiancée, Ms. Chen Ziqi, over the past eleven years. Meeting you has been the greatest fortune of my life, and I look forward to growing old together. I would like to give special thanks to my parents, Professor Hongshan Zhang and Professor Min Li, for their selfless nurturing, understanding, and love. Thank you for creating a boundless sky for me to explore. Their love has been the greatest source of strength for me during my time studying abroad.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Solar photovoltaic (PV) power generation has experienced rapid growth in recent years, playing an increasingly important role in the global transition to renewable energy sources. According to data from the International Energy Agency (IEA), solar PV power generation increased significantly from 679 terawatt-hours in 2019 to 823.8 terawatt-hours in 2020 globally. This upward trend continued as solar PV generation reached 1002.9 terawatt-hours in 2021, demonstrating the substantial progress in adopting and deploying solar technologies worldwide [1].

In parallel with the growth in power generation, solar PV installed capacity has also seen remarkable expansion. In 2022, installed capacity increased by an impressive 188.6 GW, with projections suggesting that it will reach a record high of nearly 200 GW in by the end of 2023. This rapid growth can also be observed in the UK, where the total installed capacity of solar power generation reached 14.4 GW by the end of 2022. Despite challenges posed by the Covid-19 pandemic, such as reduced efficiency and project delays, solar PV installations in the UK increased by 87% year-on-year in 2022, amounting to 0.61 GW [2].

To achieve the ambitious targets outlined in the Net Zero Scenario, the world's total solar PV installed capacity must maintain a growth rate of 25%. By maintaining this pace, it is possible to complete the mission of significantly increasing the share of renewable energy in the global energy mix by 2030 [3]. This rapid expansion of solar PV capacity highlights the importance of addressing the technical and operational challenges associated with integrating this intermittent power source into the electricity grid. As solar PV play a significant role in energy systems worldwide, innovative forecasting methods and grid management strategies will become increasingly crucial to ensuring grid stability and maximising the benefits of this clean, renewable energy source [4], this is because solar energy is intermittent and greatly affected by factors such as clouds, humidity, and other environmental conditions.

Despite its rapid development, the intermittent nature of photovoltaic power generation due to the sudden change of the environment poses a challenge to grid stability. Ground-based solar irradiance is highly variable and uncertain due to complex interactions between radiation and atmospheric components such as water vapour, aerosols and clouds [5]. In addition, the high temporal and spatial variability in the presence and concentration of atmospheric constituents further contributes to the variability of terrestrial solar irradiance [6]. The rapid alternation between clouds and sunlight can lead to voltage fluctuations and imbalances that introduce flicker into the network during grid connection, with significant adverse effects on distribution systems sensitive to short-term fluctuations [7]. Accurate prediction of rapid solar transients within one hour, also known as solar Ramp Event

(RE), is, therefore, essential for Inter-Hour Solar Forecasting (IHSF), also known as very short-term solar forecasting or nowcasting.

To address this challenge, using fisheye cameras to capture Ground-based Sky Image (GSI) for IHSF has become increasingly popular [8]. Sky images contain high-resolution local spatial information on cloud cover. Continuous sky image collection can obtain high-resolution temporal information at the minute or even sub-minute level, helping to predict rapid changes between clouds and sunlight [9]. The initial sky image model was based on image analysis methods, predicting future cloud positions by analysing cloud positions in consecutive frames [10]. In recent years, with the rapid rise of Deep Learning (DL) method, specifically deep computer vision networks, their excellent performance and fast computational efficiency have significantly improved the performance benchmark of image models, attracting widespread attention from researchers. However, using DL methods for IHSF based on GSI is still an emerging research area. Although the forecasting efficiency and performance of these models far surpass other GSI-IHSF methods [11], many challenges still need to be addressed, such as interactivity of sky images and observations, interpretability of black box models, etc. In this paper, we discuss the main challenges and shortcomings of the current multimodal forecasting models and improve the existing mainstream DL-GSI-IHSF models through modifications and optimisations at the base model algorithm and model structure levels.

1.2 Introduction

1.2.1 Solar Forecasting

Solar energy forecasting has been identified as a typical research area in energy meteorology [4]. Firstly, as a meteorological parameter, solar irradiance has typical meteorological characteristics, such as the bi-seasonal feature, that is, the solar energy time series has a typical annual cycle and diurnal cycle; the spatiotemporal feature, that is, spatiotemporal properties influence the solar irradiance; and the probabilistic feature, that is, the inherently probabilistic nature of meteorological forecasting itself. At the same time, as a forecast serving the energy and electrical engineering field, its forecasting content and form are subject to various requirements and regulations for power system control and operation. For example, the day-ahead forecast requires a net load forecast for the next operating day in order to enable power operators to make day-ahead commitments for thermal power units, optimise the dispatch of generators to balance the demand of the second day at the lowest cost, and meet the use of electricity [12]; the intra-day forecast requires a forecast of hours to minutes ahead, to further coordinate the scheduling schedule between traditional power generation and renewable energy [13]; the intra-hour forecast, through the highest resolution of up to sub-minute level forecasting, assists automatic power generation control to pre-schedule thermal power generators to provide a gap in solar power generation caused by frequent, rapid and steep slope Ramp Event (RE) [14]. Therefore, an excellent solar energy forecasting model should not only make full use of the prior knowledge of atmospheric physics to model solar irradiance but also consider the expression form of the forecast based on the constraints and requirements of electrical engineering [4]. Table 1.1 shows different types of solar energy forecasting based on the forecast horizon [4].

Table 1.1: Classification of solar forecast by forecast horizon.

Forecast Classification	Forecast Horizon	Method
Long Term Forecast	1 Week Above	Statistical models based on historical data
Day-ahead Forecast	1 Day to 1 Week	Numerical Weather Prediction (NWP)
Intra-day Forecast	1 Hour to 1 Day	Satellite data, NWP
Inter-hour Forecast	15 Sec to 1 Hour	Microscale sensor, NWP, Sky image, Satellite data

The spatiotemporal characteristics of solar irradiance have made its prediction particularly challenging. Unlike other atmospheric parameters, such as temperature and humidity, which only change continuously in time, solar irradiance is also influenced by spatial information. Specifically, incoming clouds can significantly impact solar irradiance, and a fast-approaching cloud can cause the solar irradiance to drop by more than half instantly. Therefore, solar energy forecasting techniques need to model the temporal information of solar irradiance and resolve the spatial information. Four main methods have been developed to capture the spatial information of incoming clouds: ground-based all-sky imaging, satellite remote sensing, numerical weather prediction, and microscale sensors. Table 1.2 presents the adequate time and space coverage of each method [4].

Table 1.2: Effective time and space coverage of spatial input in solar forecasting.

Spatial information data source	Effective time	Space coverage
Microscale sensors	10 sec to 2 min	1 m to 1 km
Ground-based all-sky imaging	30 sec to 15 min	1 m to 2 km
Satellite remote sensing	5 min to 6 hour	1 km to 10 km
Numerical weather prediction	2 min to 35 hour	200 m to 20 km

1.2.2 Ground-based All-sky Image in Solar Forecasting

Ground-based all-sky images captured by fish-eye cameras can provide much higher temporal and spatial resolutions than satellite data. Regarding temporal resolution, image data can capture slope events of less than one minute. Theoretically, this temporal resolution is only limited by cloud speed, image resolution, and image acquisition frequency. Meanwhile, in terms of spatial resolution, all-sky images can predict ground ranges from 1 meter to 1 kilometre from the camera [15]. Compared to other high economic cost observation instruments, relatively inexpensive monitoring cameras can be directly used as high-precision sky image acquisition devices. Therefore, in tasks with high temporal and spatial resolutions, sky images have been favoured by researchers.

Two main methods have been used to utilise sky images as external data to assist solar irradiance prediction. The first method is based on classical image analysis, which applies specific algorithms to analyse each sky image at the pixel level, extract spatiotemporal features, and perform prediction. To determine spatial features, methods such as red-blue ratio (RBR) have been used or red-blue difference (RBD) [16, 17, 18], 3D cross correlation [19], or image feature correlation [20] are used to identify cloud pixels in the sky image. To determine temporal features, the most common approach is to use the cross-correlation method [17], which calculates the cloud motion vector by comparing two consecutive cloud maps. In addition to cross-correlation, other methods include optical flow [21, 22] and ray tracing [23]. The optical flow method has been used to determine the velocity of feature pixels based on the intensity of two consecutive images. It uses this to calculate the position of the cloud in relation to the ground projection of the cloud at the approaching time point. The ray-tracing approach uses multiple images of the sky taken simultaneously from different positions, combined with ground shadow maps, to model clouds in 3D. The advantage of this approach is that the 3D model solves the problem of individual site images not being able to determine the height of the cloud base [19]. At the same time, both the cross-correlation and optical flow methods require additional instrumentation to measure the height of the cloud base to determine the correct ground projection of the cloud [24]. Image-based forecasts determine the impact on solar irradiance estimates by combining the estimates of cloud position with estimates of cloud transmittance, and general methods used to determine the latter include fixed transmittance [21, 17], cloud density-based transmittance [25, 26] and cloud height-based transmittance approaches [27]. However, these modelling approaches to image analysis are still limited by the complex physical properties of clouds. For example, cloud motion is assumed to involve shifting only and does not account for cloud generation and dissipation. Additionally, cloud transmittance depends on the transparency of the cloud, but it is not currently feasible to measure the transmittance of all cloud types directly. Therefore, this approach remains of limited use in improving the accuracy of future irradiance forecasts [28].

Another approach is to use the deep computer vision algorithms. Through training deep models on sky image datasets to extract the relationship between image features and future irradiance, this approach utilizes the learned relationship to make predictions on new data. Theoretically, this method, based on feature learning from big data, has no inherent physical assumptions and relies solely on the relationship between features in the dataset for prediction, thus being considered to have the potential to capture the underlying physical characteristics of cloud layers. In this chapter, we provide a detailed review of the DL-GSI-IHSF model. We break down the development methodology of the DL-GSI-IHSF model into three stages: data, model, and analysis, and summarise the current state and existing problems of the model discovered at different stages.

1.2.3 Deep Learning

Another essential background knowledge used in this paper is deep learning algorithms. The inspiration for deep learning algorithms comes from biological neural networks. By simulating the information processing and distributed nodes in biological systems, artificial neural networks can achieve representation learning for targets. The basic architecture of an artificial neural network is the neuron, which is a linear regression processing with an additional activation function. Taking the Rectified Linear Unit (ReLU) activation function as an example, an artificial neuron can be represented as:

$$y = \mathrm{ReLU}(\mathrm{wx} + \mathrm{b}) \tag{1.1}$$

$$where\ \mathrm{ReLU}(\mathrm{x}) = \max(\mathrm{x}, 0) \tag{1.2}$$

The basic artificial neural network is the multi-layer neural element structure formed by the horizontal and vertical accumulation of neurons. In this paper, to avoid misunderstanding, the term ”Multilayer perceptron (MLP)” has been used to describe artificial neural networks with fully connected structures. In some work, this architecture may be directly referred to as an artificial neural network. The network architecture of an MLP is that all neuron nodes in the previous layer are connected to the neurons in the next layer. A layer in this structure can be represented as:

$$\mathbf{Y} = \mathbf{X}\mathbf{W} + \mathbf{b} \tag{1.3}$$

term "Multilayer perceptron (MLP)" is used to describe artificial neural networks with fully connected structures. In some work, this architecture may be directly referred to as an artificial neural network. The network architecture of an MLP is that all neuron nodes in the previous layer are connected to the neurons in the next layer. A layer in this structure can be represented as:

where the input matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has n x-samples, each with d features. The weights $\mathbf{W} \in \mathbb{R}^{d \times h}$ and bias $\mathbf{b} \in \mathbb{R}^{1 \times h}$ are computed to form a full connection from the elements in each input $\mathbf{X}$ to the output $\mathbf{Y} \in \mathbb{R}^{n \times h}$. A fully connected MLP is formed by accumulating multiple layers, where the layers in between, except for the final output layer, are called hidden layers.

According to the learning paradigm of deep networks, training a network can be divided into supervised, unsupervised, and reinforcement learning. Supervised learning refers to training a model to explicitly pair inputs and outputs (or labels), aiming to establish a mapping relationship between the input and output through the deep model. To the best of our knowledge, almost all works implementing the DL-GSI-IHSF model and similar methods have employed supervised learning to train the model, which is achieved by directly or indirectly setting the model's prediction target as the goal of solar forecasting. In this paper, this relationship is represented by the following equation:

$$\mathbf{y}_{t+\Delta t} = F(\mathbf{X}, \mathbf{W}) \tag{1.4}$$

where $y_{t+\Delta t}$ and $\mathbf{y}_{t+\Delta t}$ is the predicted value and representation feature vector after Δt time, respectively, F is the model calculation function, $\mathbf{X}$ is the model input, $\mathbf{W}$ is the trained model weight. Please note that here, $\mathbf{W}$ refers to the generalised parameters, including weights and biases, which can be adjusted through model training, and is different from the $\mathbf{W}$ in Equation 1.3.

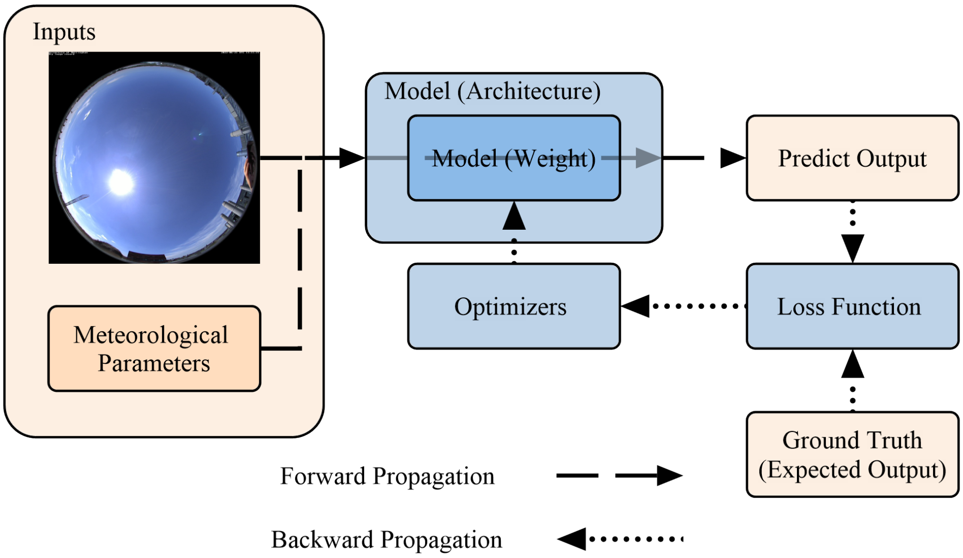


Figure 1.1: Forward propagation (prediction) and backward propagation (gradient descent) in deep learning.

The training process of a neural network, i.e., the model fitting process, is shown in Figure 1.1. The model first generates a $\hat{y}$ through forward propagation and then quantifies the difference between the predicted output, $\hat{y}$, and the expected output, y, through a loss function. The most common loss function for regression problems is the Mean Square Error

(MSE), while for classification problems, the most common loss function is the cross-entropy function. The training process of a supervised deep neural network is essentially the process of iteratively searching for the model parameters $\mathbf{W}$ that minimise the model loss function. This process can be represented as follows:

$$L(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} l_i(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{1.5}$$

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} L(\mathbf{W}) \tag{1.6}$$

where L is the set of loss functions for the i-th sample loss function l, $\mathbf{W}^*$ refers to the final parameters in the trained model. During the model iteration process, the model loss is continuously adjusted by updating the model weights to search for the minimum weight values, as shown in Figure 1.1. This process is called gradient descent because the gradient of the loss calculates it concerning the model's weights. The algorithm used to calculate the update of the model weights in the gradient descent process is called the optimiser of the model. For example, the most common optimiser is Stochastic Gradient Descent (SGD), which can be represented as follows:

$$\mathbf{W} \leftarrow \mathbf{W} - \eta\partial_{\mathbf{W}} L(\mathbf{W}) \tag{1.7}$$

where η is the learning rate of the model, which describes the ratio of gradient descent. The computational process will be very inefficient if the entire dataset is traversed each time the model weights are updated. Therefore, in practical training, a smaller sample set $\mathbf{B}$ is usually extracted each time the update needs to be calculated and updated using minibatch stochastic gradient descent. This process can be represented as follows:

$$\mathbf{W} \leftarrow \mathbf{W} - \frac{\eta}{|\mathbf{B}|} \sum_{i \in \mathbf{B}} \partial_{\mathbf{W}} l_i(\mathbf{W}) \tag{1.8}$$

where $|\mathbf{B}|$ represents the number of samples in $\mathbf{B}$, also called batch size. In the derivation process above, batch size $|\mathbf{B}|$ and learning rate *eta* are set directly by the researcher during model training rather than optimised. These parameters are called hyperparameters, and they significantly affect model training results. They are generally tuned on a validation set that does not participate in the iteration of model weights.

1.3 Aims and Objectives

The primary aim of this project is the enhancement of the predictive performance, interpretability, generalizability, and computational efficiency of the multimodal DL-GSI-IHSF model through a series of experiments and investigations. To achieve this aim, the following objectives have been set:

1. A comprehensive review of the existing model was undertaken, with a step-by-step analysis of the current work's strengths and weaknesses by categorising the existing work's structure (Chapter 2).

2. Specify the methodology used in the three sub-categories, including enhancing modality fusion, generalizability, and calculation accuracy and efficiency. Collect and develop data sets with specific collect data methods (Chapter 3).

3. Enhance modality fusion mechanism and model interpretability by

incorporating attention and gating mechanisms in deep models to reinforce inter-modality interactions (Chapter 4).

4. Improve the model's generalisability to different climatic conditions using inductive transfer learning methods for model training with limited datasets under various climates (Chapter 5).

5. Optimise and simplify the model architecture and boost computational efficiency through model optimisation. Increase efficiency and prevent architectural redundancy (Chapter 6).

Chapter 2

Literature Review: A Review of Deep Learning-based Inter-Hour Solar Energy Forecasting using Ground-based Sky Images

Chapter Abstract

A comprehensive overview of the model construction process was provided in this chapter for deep learning-based solar energy prediction using ground-based sky images, including data pre-processing, model architecture, and evaluation methods. The current state-of-the-art techniques were summarised and discuss their advantages and limitations. The challenges and future research directions were also highlighted in this field, such as data heterogeneity, model generalisation, and interpretability.

Contents

2.1 General Approach of DL-GSI-IHSF

Although there have been many reviews on GSI-IHSF in previous work [29, 30, 8, 31], these articles mainly introduce a subclass of solar forecasting that uses Machine Learning (ML) combined with GSI for prediction. To the best of our knowledge, no comprehensive review summarizes these contents from the perspective of DL, for example, the specification of data sets, DL frameworks, and training strategies were used. This paper reviews articles in the past five years that used DL and ground-based sky images for solar forecasting. We also compare these articles with other methods for solar forecasting using ground-based sky images.

The earliest work that used DL and computer vision for GSI-IHSF can be traced back to the work of Sun et al. [32], who used Convolutional Neural Network (CNN) to establish the mapping relationship between current sky images and PV outputs. Since then, many works have improved existing DL architectures or added new modules to existing ones [33]. After analysing all published literature, we classify the solar forecasting framework that combines DL and ground-based sky images into three main stages, as shown in Figure 2.1:

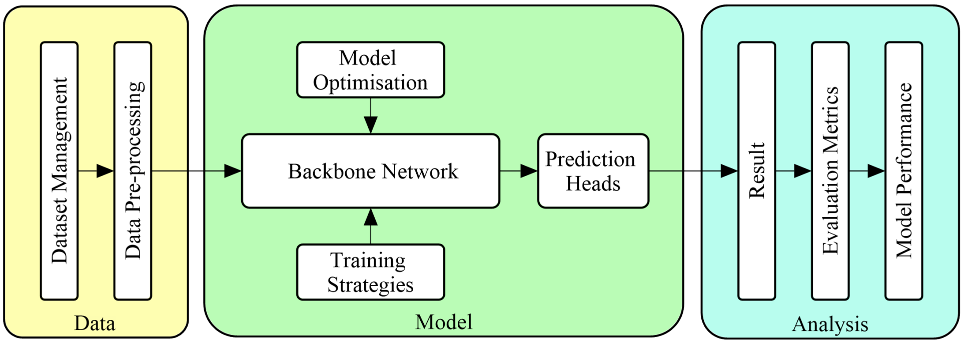


Figure 2.1: DL-GSI-IHSF framework.

Data Phase This phase includes the entire process data processing, from obtaining the raw data to starting the model training. Specifically, it includes acquiring the raw data, pre-processing the data and matching the forecast target. Moreover, to improve the efficiency or accuracy of DL model in data analysis, methods are often used to change the state of the original data set, such as re-sampling [34], image augmentation [34], or image distortion [35, 11].

Model Phase This phase aims to train one or more DL models to establish the mapping relationship from input to output. In this phase, the first step is to determine the backbone network, and its purpose is to extract features from the data for forecast. The backbone network is often a validated DL model in Computer Vision (CV) field, such as VGGNet, ResNet, 3D-CNN, and ConvLSTM. Since the input type of GSI-IHSF can be more than one modality [33], i.e., images and other numerical information are input to the model simultaneously, the backbone network may contain more than one backbone model to parsing different modality data. In general, DL models that use other modalities besides GSI as input can be defined as "Hybrid Models," even if they are under the same DL model framework. The trained backbone Network can victories the input data and provide it to the prediction head. The prediction head refers to the part that maps the data features extracted by the backbone network to the predicted results. At the same time, in the training process, most works often use some model optimisation methods and training strategies to improve the model's stability, efficiency, and accuracy.

Analysis Phase This phase analyses the performance based on the result generated from the operation of the trained DL model on a test data set.

The test set is a separate data set divided before training and contains data that never appeared during the model's training. This step aims to validate the model's performance under realistic working conditions. A series of metrics are defined in this phase to describe the model's performance and are compared with other benchmark models.

The structure of this paper has been organised as follows. Sections 2.2, 2.3, and 2.4 review the three main phases of the DL-GSI-IHSF framework, including the data phase, model phase, and analysis phase, respectively. Section 2.5 provides the current research prospects of GSI-IHSF and summarises this paper.

2.2 Data Phase

The process of acquiring GSI for IHSF using DL methods is similar to that of non-DL GSI works, which involves regularly capturing sky images with ground-based sky imaging systems. However, compared to traditional image analysis techniques or machine learning-based methods, DL methods often require enormous data and datasets. As a result, data acquisition and dataset management pose challenges in developing GSI-IHSF models using DL techniques.

The performance and development of DL models depend highly on data quantity, quality, and diversity. In supervised learning, the training set size must meet minimum requirements to estimate the input-output mapping and prevent overfitting or underfitting adequately. At least a year of data is necessary for solar energy prediction models to account for time periodicity [36]. Data quality control for solar energy monitoring platforms demands costly equipment, trained personnel, and maintenance. Data diversity is crucial in generalising DL models and preventing overfitting while improving their robustness. Creating a well-formulated DL-GSI-IHSF dataset often demands considerable staffing, time, and finances.

Data augmentation is a widely used technique in deep learning that involves generating synthetic data by applying transformations to the existing dataset. It is an effective method for increasing the amount and diversity of training data, thus improving the model's generalizability and reducing overfitting. For example, in DL-GSI-IHSF, Paletta et al. [11] demonstrated that compelling image pre-processing techniques could significantly improve the model's solar irradiance prediction performance.

This section describes the methods for using datasets in existing DL-GSI-IHSF work, including dataset-related items, dataset pre-processing methods and dataset enhancement methods. Table 2.1 details all the data relevant information we collected on DL-GSI-IHSF work. Specifically, Section 2.2.1 introduces existing datasets and data collection instruments, while Section 2.2.2 describes the data pre-processing process, including dataset standardisation and data enhancement. Table 2.1 below includes detailed information about the data used in the DL-GSI-IHSF work, including model inputs, dataset size and partitioning.

2.2.1 Dataset

The GSI dataset is generally obtained by continuously capturing fish-eye lens-based sky images through a sky image acquisition system. In studies involving DL models, the sky image acquisition systems used include both image databases generated from existing sky image acquisition systems in meteorological stations, such as SRRL [37], SIRTA [38], UCSD-Folsom [39], as well as using ready-made cameras for capturing sky images directly to generate DL datasets, such as SKIPP'D [40]. Maturely designed sky image acquisition systems are often additionally designed for solar monitoring, such as using High Dynamic Range (HDR) technology to reduce overexposed areas in the Sun's halo [18, 41], using neutral density filters to reduce incident radiation [41], and capturing the Sun and cloud information separately through multiple exposures [42]. As a result, it is believed that such systems can provide better image quality than ready-made cameras. However, to our knowledge, no research has been quantitatively investigating the direct and practical impact of different image qualities on model quality. Since this article does not focus on the data collection method, readers interested in the sky image acquisition system may refer to [8].

Inputs Due to the modular nature of DL, models can freely combine architectures or share parameters. Therefore, the input to the model in the dataset can be not only the sky image itself. These models with additional inputs are called hybrid or fused models. This work can be traced back to Sun et al. [33], where PV Log was incorporated into the prediction model to train SUNSET for forecasting. The study found that the fused model combining the current sky image with measured values performed better than independent models using only sky images or measured values and thus became popular in subsequent work. Inspired by this work, using atmospheric observations such as temperature, humidity [43, 44, 45], or solar-related parameters such as solar altitude, solar azimuth [46], clear-sky global horizontal irradiance [47], or NWP-related parameters such as 500AOD [48] as fusion inputs have become a new research hot-spot. The multi-input model will be discussed in Section 2.3.1.

Outputs Currently, all the GSI-IHSF work based on DL adopts the supervised learning mode, which requires a well-defined expected output for model training. Thus, the sky image dataset should have a deterministic output to fit the mapping relationship. It can be the current or future prediction target, such as GHI or PV output, or indirect parameters of the prediction target, such as the Clear Sky Index (CSI) representing the relative irradiance to clear-sky irradiance. In other solar forecasting methods, using relative coefficients, such as CSI, as the prediction target for Global Horizontal Irradiance (GHI) has been widely validated to be superior to directly predicting GHI [49]. However, in DL, whether using CSI as the prediction target for PV output has better performance [33] is still controversial. Furthermore, since the DL network training based on the loss function allows multiple possible targets, such as using a series of loss functions, including relative change rate, Zhang et al. [50] achieved excellent performance in one-minute ahead prediction. The details of this direction will be discussed in Section 2.3.4.2.

Dataset size The size of the dataset is also an essential factor affecting the model's performance. For example, A study showed that using 70K samples from 2 years can achieve a 10% improvement compared to using 35K samples from 1 year [42]. Additionally, Feng et al. [36] pointed out that collecting data for a complete solar cycle, i.e., one year of data, is necessary to ensure temporal diversity in the dataset. They argued that using datasets collected for less than a year may have limited validity regarding model performance. Such datasets may only be convincing in exploring model feasibility. In addition to expanding data diversity in the temporal dimension, data diversity can be expanded in the spatial dimension using multiple sites. Existing knowledge in the field of CV suggests that it is practical to train models by migrating pre-training weights using large-scale pre-training methods. This approach can reduce the data and time required to train the model through a priory pre-training knowledge [51, 52]. In the work of DL-GSI-IHSF, Nie et al. [53] found that based on the transfer learning mechanism, it is possible to train with just 20% of a 10-month dataset to get performance beyond that of training 100% of the dataset from scratch.

2.2.2 Data Pre-processing

Data pre-processing is converting raw data into input data for the model. Typically, the data pre-processing process involves two steps: the first step is data standardisation, including downsampling, segmentation and normalisation. The primary purpose of this step is to reduce the size of the dataset as much as possible without losing the data characteristics to reduce the computational cost of the model fitting. The second step is data augmentation, including dataset resampling, image distortion correction, and image data enhancement. This step alters the original features of the data and aims to improve the model's ability to extract data features through data editing to obtain better performance.

2.2.2.1 Data Standardisation

The data standardisation process mainly includes downsampling and normalisation, whereas downsampling includes both dataset and image downsampling.

Data Downsampling In an ideal situation, the more considerable the amount of data input to a DL model, the better the model's generalisation and the less likely it is to overfit. However, training a model on a massive dataset requires significant time and cost. Therefore, some researchers choose to extract a subset of the data from a large dataset through downsampling for model training and validation [33, 46, 35, 34]. As mentioned, when downsampling, it is crucial to consider the issue of sample balance, i.e., the samples should cover all possible data distributions as much as possible.

Dataset Segmentation Splitting the dataset divides the dataset used for DL into training, validation, and testing sets. The training and validation sets are used to fit the model, and the testing set is used to evaluate the model's performance in real-world scenarios. Based on our review, there are three main methods for dataset splitting: (1) random splitting, which involves randomly sampling the total sample to obtain the training, validation, and testing sets in specified proportions [54, 43, 46, 55]; (2) continuous splitting, which involves dividing continuous data into three sets based on timestamps [47, 56, 57]; and (3) intra-day continuous splitting, which involves selecting typical days manually as continuous testing/validation sets [33, 58, 59]. It should be noted that some studies only use the training and validation sets for model training and directly use the validation set results as the standard for model evaluation. We believe this approach lacks rigour [60], and cannot verify whether the improvement in model performance is due to overfitting. In addition, each subset should ensure complete data diversity. Specifically, using a validation or dataset spanning three months or half a year may also lead to erroneous estimates of model performance. In their work, Paletta et al. [35] used odd and even days of the year to avoid this situation when the dataset size was limited.

Data Normalisation The purpose of general data normalisation is to scale the input data to the same range in order to eliminate the influence of the input data scale on the weight distribution in the model. There are two standard normalisation methods in DL. The first method is to scale by proportion, where each pixel in each channel (red, green, blue or gray-scale value) of the image data or each data in a numerical dataset is scaled by its relative intensity value, which can be represented as equation (2.1).

$$x_i^{norm} = \frac{x_i - min(\mathbf{X})}{max(\mathbf{X}) - min(\mathbf{X})} \tag{2.1}$$

Where $\mathbf{X}$ represents the image channel or the numerical dataset and x_i represents the pixel in the image channel or the data point in the numerical dataset, respectively. It is worth noting that, since the pixel values range from 0 to 255, a simplified way to apply this method in images is to use the extremal values of the pixel range as the extremal values of the image, which simplifies the formula to equation 2.2

$$x_i^{norm} = \frac{x_i}{255} \tag{2.2}$$

Another common approach is standardising the image information to follow a normal distribution with mean 0 and variance 1. The specific method can be represented as equation (2.3)

$$x_i^{stan} = \frac{x_i - mean(\mathbf{X})}{\sigma_{adj}} \tag{2.3}$$

. Where σ_{adj} represents the adjusted standard deviation, which is calculated as

$$\sigma_{adj} = max(\sigma, 1/\sqrt{N}) \tag{2.4}$$

Where N is total element in $\mathbf{X}$ set.

In addition to standard DL methods, there are other proprietary methods from solar forecasting for data standardisation. For image data, unlike standard CV datasets, sky image datasets often contain only limited colours, specifically the blue of the sky and the white of clouds. Therefore, using the relationship between the red channel to blue channel [61, 21, 62] as a parameter and using a fixed or adaptive threshold to distinguish between sky and clouds can also be used as an image normalisation method. In [35], the authors used this method to classify pixels in sky images and used a matrix representing the pixel classification instead of the original sky images for prediction. For numerical data, some meteorological applications use angle information, such as solar or wind direction angles [56], which can be normalised to 0 to 1 using trigonometric transformations.

2.2.2.2 Data Enhancement

Data augmentation refers to a set of data pre-processing techniques in CV that enhance the size and quality of a dataset, such as data resampling, image augmentation, and image pre-classification [63]. It is mainly applied in CV datasets with limited data to improve the model's performance and generalisability by increasing the data diversity. Meanwhile, image pre-processing methods from the solar energy prediction field, such as fish-eye image correction, are also applicable in DL-GSI-IHSF.

Enhancement for Imbalanced Dataset Data imbalance in DL datasets refers to a situation where the number of samples in different classes varies significantly. The imbalanced dataset can cause the model to predict more towards the classes with larger sample sizes and exhibit bias towards the classes with smaller sample sizes [64]. This imbalance is particularly evident in short-term solar energy forecasting, where solar energy data collection stations are often located in sunny areas, resulting in a much larger number of sunny samples than cloudy samples, which are the minority samples of interest in short-term solar energy forecasting using ground-based sky images. For example, in the SKIPP'D [40] dataset, the sample imbalance ratio used for prediction is as high as 7.82 [34], which means that the relevant samples only account for one-eighth of all the samples. In machine learning, there are three types of methods for dealing with such imbalanced datasets: data-level techniques that involve resampling the dataset, algorithm-level techniques that involve redefining the loss function, and hybrid-level techniques that use both methods [65].

In [34], three resampling methods were tested to address imbalanced datasets. Approach 1 oversamples the positive and undersamples the antagonistic classes while maintaining the original dataset size. Approach 2 only oversamples the positive class, resulting in an expanded dataset. Approach 3 only undersamples the negative class, resulting in a reduced dataset. Regarding oversampling, replicate, Gaussian noise, colour casting [66], and synthetic minority over-sampling technique (SMOTE) [67] were tested to augment samples. It was found that the first two methods significantly improved the performance of the nowcasting method. In contrast, the third method was effective only at specific undersampling rates—however, none of the three methods significantly improved model performance in the prediction task.

In addition to using sampling methods to balance the dataset, another approach is to directly classify the dataset into subsets using validated statistical metrics or physical methods and then train the model on the classified subsets. In the work of Nie, et al. [59], the dataset was pre-segmented into the clear sky, cloudy, and overcast subsets using the Clear Sky Index (CSI) and a physical method based on solar regions, and DL models were trained on each subset. This approach directly avoids dataset imbalances under different weather conditions, resulting in an 8% improvement in model performance compared to training on the original dataset.

Enhancement for Images Image augmentation refers to transforming and perturbing images in various ways to generate new images for training and optimising DL models. Research on image augmentation for classical DL methods applied in the solar forecasting area is minimal. For image transformations, techniques such as flipping, rotation, cropping, and scaling are unsuitable for all-sky images captured with fish-eye lenses. [34] has used techniques such as adding Gaussian noise, colour transformation, brightness adjustment, SMOTE, and image blending for image augmentation. However, the results show that such methods do not significantly improve the forecast model's performance.

In addition to the general image augmentation methods in DL, some methods based on ground-based sky images have also attracted attention in recent years. For example, high dynamic range (HDR) technology uses multiple fast consecutive images with different exposure times to blend into one image, to address the over-exposure or under-exposure caused by the high brightness contrast between the sun region and cloud layer in sky images [68]. The multiple images this technology generates can be concatenated into one sample and fed into a CNN model [50, 42]. The work of Zhang et al. [50] found that using multiple exposures of high dynamic range to generate composite images improves the model's performance by about

10% compared to using a single image. However, this technology needs to be implemented during the image collection phase. Another commonly used sky image pre-processing method is distortion correction. Fish-eye lenses cause significant distortion in the samples, resulting in non-uniform spatial geometric relationships within the image. Therefore, distortion correction is often used in short-term non-DL-based ground-based sky image forecasting. Previous work systematically tested the application of distortion correction in DL methods and found that it did not improve the performance of the deep model [35]. However, another pre-processing transformation method, SPIN, proposed by Paletta et al. [11], uses a polar-coordinate transformation to convert fish-eye images to Cartesian coordinates. This method converts the relative position relationship between the sun and cloud layers in the image into a structure that is more conducive to CNN learning and has achieved significant results. In 2-minute forecasting, this method significantly increased the model prediction score from 8.4% to 23.1%.

Table 2.1: Datasets used in the DL-GSI-IHSF work.

Ref	Input (excluding GSI)	Collection time	Total Samples	Test/Val. set ratio	Val/Test set splitting method
[32]	GHI, PV Output	7 M	36804	17.8%/10.8%	Intra-day consecutive
[50]	PV Log	1.5 Y	N/A	20%	Random
[69]	PV Log	1 Y	76908 @Baseline, 830069 @High Freq.	10.30%	Intra-day consecutive
[70]	N/A	11 M	N/A	45%/14%	Intra-day consecutive
[33]	PV Log	1 Y	76908 @Baseline, 830069 @High Freq.	10%	Intra-day consecutive
[71]	PV Log	1 Y	76908	10%	Intra-day consecutive
[54]	GHI	11 H	1580	20%	Random
[58]	GHI	16 D	N/A	56.00%	Intra-day consecutive
[47]	DNI, Clear DNI	2 Y	N/A	5%/45%	Consecutive

continued in next page...

Table 2.1 Datasets used in the DL-GSI-IHSF work (Continued).

Ref.	Input (excluding GSI)	Collection time	Total Samples	Val/Test set ratio	Val/Test set splitting method
[43]	N/A	1 Y	25000	20%	Random
[56]	Clear GHI, RH, Wind Speed Solar angle,Temperature, Surface pressure	12 Y @Golden, 7 M @Tuscan	1297410 @Nowcast, 31005 @Forecast	47.2% @Nowcast, Golden 17.3% @Nowcast, Tuscan	Consecutive
[57]	N/A	6 Y	155644	16%/33%	Consecutive
[59]	N/A	1 Y	102885	9%/9.6%	Intra-day consecutive
[72]	N/A	10 Y	259949	30%/20%	Consecutive
[73]	N/A	1 Y	25000	10%/10%	Random
[74]	N/A	N/A	6000000	N/A	-
[75]	N/A	20 D	N/A	25%	Intra-day consecutive
[76]	N/A	5 D	N/A	70 sample	Intra-day consecutive

continued in next page...

Table 2.1 Datasets used in the DL-GSI-IHSF work (Continued).

Ref.	Input (excluding GSI)	Collection time	Total Samples	Val/Test set ratio	Val/Test set splitting method
[46]	GHI, Solar angle	7 M	20000	20%	Random
[77]	N/A	16 M @Golden, 16 M @Folsom	35552 @Golden, 341572 @Folsom	10.8%/9.5% @Golden, 15.8%/6.8% @Folsom	Consecutive, Random
[55]	PV Log	3 M	31273	25.5%/14.8%	Random
[42]	GHI	3 Y	55000	18.18%/18.18%	Random
[34]	PV Output	2.5 Y	135527	10%/4% @Nowcast, 9%/7% @Forecast	Intra-day consecutive
[44]	GHI, RH, Wind Speed, Temperature	4.5 M	N/A	20%+3 typical days	Intra-day consecutive
[78]	DNI, RH, Air mass, Solar zenith angle	2 Y	N/A	18%/50%	Consecutive

continued in next page. . .

Table 2.1 Datasets used in the DL-GSI-IHSF work (Continued).

Ref.	Input (excluding GSI)	Collection time	Total Samples	Val/Test set ratio	Val/Test set splitting method
[79]	GHI	2 Y	52429	25%/25%	Consecutive
[45]	PV Log, Rainfall, Wind Speed, Rainfall intensity, Temperature, Wind Direction, Mean sea level pressure,	6 M	N/A	20%	Random
[80]	N/A	3 Y	56640	17.8%/20.3%	Consecutive
[81]	GHI	3.75 Y	N/A	26.7%/26.7%	Consecutive
[82]	Clear Sky Index	3 Y	141805	34.10%	-
[83]	N/A	N/A	24000	5%/5%	Random
[36]	N/A	6 Y	N/A	16%/33%	Consecutive

continued in next page...

Table 2.1 Datasets used in the DL-GSI-IHSF work (Continued).

Ref.	Input (excluding GSI)	Collection time	Total Samples	Val/Test set ratio	Val/Test set splitting method
[53]	GHI,PV Output	32 M @Stanford, 3 Y @SIRTA, 1 Y @DEWA	135527 @Stanford, 448268 @SIRTA, 91979 @DEWA	9%/7% @Stanford, 10%/2% @SIRTA, 9%/7% @DEWA	Consecutive, Intra-day consecutive
[84]	GHI	4 M	1186	15%/15%	Random
[40]	PV Log	3 Y	N/A	N/A	Intra-day consecutive
[35]	N/A	3 Y	N/A	12.5%/12.5%	Intra-day consecutive
[11]	N/A	3 Y	N/A	9.1%/9.1%	Intra-day consecutive
[48]	500AOD, GHI, Clear GHI, Surface pressure, Temperature, Wind Speed, Total precipitation	3 M	5110	15%	Random

continued in next page...

Table 2.1 Datasets used in the DL-GSI-IHSF work (Continued).

Ref.	Input (excluding GSI)	Collection time	Total Samples	Val/Test set ratio	Val/Test set splitting method
[85]	GHI	1 Y	25000	20%	Random
[86]	GHI, Surface pressure, Temperature, Wind Speed	6 Y	N/A	16.7%/16.7%	Consecutive
[87]	N/A	1 Y	N/A	N/A	-

2.3 Model Phase

Model development, debugging, iteration and optimisation are the core of DL-GSI-IHSF. The dominant approach to building a DL-GSI-IHSF model is to migrate proven frameworks from the CV domain. Researchers need first replicate the proven backbone model architecture of the CV field. Secondly, for the multi-source, high-dimensional, and complex spatiotemporal relationships, researchers need to redesign the model based on previous solar energy prediction experience, including network architecture, model fusion algorithms, loss functions, prediction heads, and other aspects, in order to optimise the model performance and to enhance the generalisability of the model.

This section introduced the relevant content of the DL-GSI-IHSF model. Specifically, section 2.3.1 introduced the model prediction mechanism, section 2.3.2 introduced the main backbone models currently used, section 2.3.3 adjustable variables in the model framework, including validation of validity, optimisation of hyperparameters, reduction of randomness and other details, and section 2.3.4 introduced the prediction head for generating model predictions.

Table 2.2 below includes detailed information about the model used in the DL-GSI-IHSF work. The table includes the DL-GSI-IHSF backbone model, target attributes, forecast time relevant items, and optimisation algorithms methods.

2.3.1 Forecasting Mechanisms

The overall prediction mechanism of the DL-based solar forecasting method using sky images can be summarised as follows: a representative future-state feature vector is extracted from the input at the current time using a backbone network, which is then converted to a prediction output using a prediction head. The following equation can represent this mechanism:

$$\mathbf{y}_{t+\Delta t} = F_{backbone}(\mathbf{X}, \mathbf{W}_{backbone}) \tag{2.5}$$

$$y_{t+\Delta t} = F_{head}(\mathbf{y}_{t+\Delta t}, \mathbf{W}_{head}) \tag{2.6}$$

where $y_{t+\Delta t}$ and $\mathbf{y}_{t+\Delta t}$ is the predicted value and feature vector after Δt time, respectively, F is the model calculation function, $\mathbf{X}$ is the model input, $\mathbf{W}$ is the trained model weight.

According to our investigation, based on the model's inference logic framework, the models can be classified into two different categories based on the following criteria: input content and backbone model architecture. The former determines the input-to-output mapping logic of the model's extrapolation, while the latter determines the calculation logic within the model.

Classification by Inputs Content Categorised by input content, deep solar energy prediction models aim to clarify the mapping relationship between the input $\mathbf{X}$ to the feature vector $\mathbf{y}$ and the predicted output y when $\mathbf{X}$ is added or modified. As mentioned above, the model's input can include only one modality, such as an image or an image sequence, or multiple modalities, such as an image and numerical data. When the input modality of the model only includes sky images, we classify the model as a single modality model. The prediction mechanism of this type of model is the same as the overall prediction mechanism, directly establishing the mapping relationship between the image and the prediction target through the backbone model, as shown in equation (2.7).

$$\mathbf{y}_{t+\Delta t} = F_{backbone}(\mathbf{X}_{img}, \mathbf{W}_{backbone}) \tag{2.7}$$

In earlier work, Sun et al. [33] found that adding PV log as numerical input can effectively improve the model's forecast performance, making its Forecast Skill (FS) higher than that of models with only image or numerical inputs. Since then, much work has used such multimodal input models. The design of Sun et al. was followed in some work to fuse numerical measurements of the current moment into the network architecture for improved prediction of future values [58, 55, 42, 81, 85]. In addition. Work has been done to further extend the model architecture based on this concept, using clear sky irradiance [47], solar angles [46], meteorological data [56, 45] including temperature, humidity, pressure, wind conditions, and Numerical Weather Prediction (NWP) parameters [48] such as AOD500 added to the model to aid prediction. The comparison experiment by Kong et al. [55] found that incorporating numerical inputs as mixed inputs with image networks resulted in significantly better quantitative prediction performance than all models based solely on image inputs. In their work, this hybrid input method made it the only model to outperform the baseline model in quantitative prediction. It is worth noting that the correlation of additional parameters directly determines the effectiveness of the aid to the model prediction. Therefore, using cross-validation or ablation experiments to verify the correlation of input parameters with the model prediction target is necessary. For example, Zuo et al. [48] applied a linear correlation analysis approach to screen a range of NWP parameters and excluded multiple mean wind speed and station pressure parameters with low correlation to the GHI.

When the model has more than one modality input, according to the multimodal learning classification method, the model can be divided into data-level fusion, feature-level fusion, and decision-level fusion according to the interaction and modality resolution order of different modalities. Taking image data and numerical data as an example, data-level fusion means extracting and fusing data between different modalities and inputting them into the model as a unified input. This can be expressed as decomposing Equation (2.7) as follows:

$$\mathbf{X}_{biomodal} = f(\mathbf{X}_{img}, \mathbf{X}_{num}) \tag{2.8}$$

$$\mathbf{y}_{t+\Delta t} = F_{backbone}(\mathbf{X}_{biomodal}, \mathbf{W}_{backbone}) \tag{2.9}$$

Where f represents the algorithm for data fusion, which can be matrix concatenation, matrix addition, or other algorithms, in the early exploration of data fusion in [71], feasible data-level fusion was achieved by upsampling the numerical input to the same size as the image matrix, and then with different algorithms, such as adding, multiplying, and concatenating to obtain the fused data. However, the result shows that such methods did not surpass the feature- and decision-level fusion methods. Recently, in a study by Paletta et al. [35], it was suggested that adding the numerical data by upsampling and concatenating it into the image as the fourth channel (making RGB become RGBI, where I stands for irradiance) can improve the model's performance, especially when using the ConvLSTM model as the backbone architecture. Feature-level fusion, which fuses the features obtained by encoding the data through a particular encoder, is currently the most popular architecture for deep solar energy prediction using multiple modalities. Feature-level fusion fuses the features obtained by encoding the data, which reduces the total computational cost compared to data-level fusion. Feature-level fusion can be further divided into early fusion and late fusion. The critical difference is whether the main module responsible for model inference is before or after feature fusion. The following equation shows the detailed process of equation 2.5 in the feature-level fusion.

$$\mathbf{y}_{img_{t+\Delta t}} = F_{img}(\mathbf{X}_{img}, \mathbf{W}_{img}) \tag{2.10}$$

$$\mathbf{y}_{num t+\Delta t} = F_{num}(\mathbf{X}_{num}, \mathbf{W}_{num}) \tag{2.11}$$

$$\mathbf{y}_{biomodal t+\Delta t} = f(\mathbf{y}_{img_{t+\Delta t}}, \mathbf{y}_{num t+\Delta t}) \tag{2.12}$$

$$\mathbf{y}_{t+\Delta t} = F_{fusion}(\mathbf{y}_{biomodal t+\Delta t}, \mathbf{W}_{fusion}) \tag{2.13}$$

Currently, almost all hybrid DL models for multimodal solar energy forecasting, which belong to feature-level fusion, use the method of concatenating feature vectors for modality fusion [33, 58, 42, 44, 78, 45, 82, 48]. The concatenated longer vector of image and numerical feature vectors is used as the model's fused feature vector. This method of directly connecting two eigenvectors allows for straightforward modal fusion. However, based on experience in multimodal learning, it is shown that this method has some potential drawbacks [88]. First, it oversimplifies the potential semantic space representation between the two classes of feature vectors to level the process of projecting each of them to a shared semantic space. Since different backbone networks usually extract feature vectors, they often have different semantic space representations. Directly concatenating vectors ignore the existence of such potential representations. Secondly, there is no information exchange during the vector concatenation process, and the information interaction process is realised by subsequent cross-modal network extraction. In other words, the direct concatenating method has a lower utilisation rate of information. Although the above two views have been repeatedly verified in text-image multi-modal [89], they are still not widespread in deep multimodal networks for solar energy prediction, and the vector concatenation method is still the most popular method at present. Other viable frameworks based on multimodal learning, such as probabilistic graphical models [90], deep canonical correlation analysis [91], generative adversarial network [92] and attention mechanism [93], have not yet been practised in the DL-GSI-IHSF.

Decision-level fusion refers to assigning the deep learning task to two or more different sub-networks for prediction and then fusing the results based on the outputs from all models. It can be represented as:

$$\mathbf{y}_{numt+\Delta t} = F_{num}(\mathbf{X}_{num}, \mathbf{W}_{num}) \tag{2.14}$$

$$y_{numt+\Delta t} = F_{numhead}(\mathbf{y}_{numt+\Delta t}, \mathbf{W}_{numhead}) \tag{2.15}$$

$$\mathbf{y}_{img_{t+\Delta t}} = F_{img}(\mathbf{X}_{img}, \mathbf{W}_{img}) \tag{2.16}$$

$$y_{img_{t+\Delta t}} = F_{imghead}(\mathbf{y}_{img_{t+\Delta t}}, \mathbf{W}_{imghead}) \tag{2.17}$$

$$y_{t+\Delta t} = f(y_{numt+\Delta t}, y_{img_{t+\Delta t}}) \tag{2.18}$$

Decision-level fusion can be more flexible since the sub-models used for fusion do not necessarily need to be deep-learning models. For instance, classic non-deep learning methods for solar forecasting based on image analysis can be considered a generalised form of decision-level fusion. Specifically, these methods use a high-accuracy clear sky model as the first part of the decision and a sky image-based solar irradiance attenuation rate prediction as the second part. Finally, the two parts of the decision are multiplied to obtain the final result. Using DL-based decision-level fusion is a hybrid model that combines one of the parts with deep learning techniques. Venugopal et al.'s [71] work first explored this strategy in a deep learning model. In their two-step model, they use PV log as input to predict the PV output using four different prediction models: a persistence model, an intelligent persistence model, an auto-regression model and anMultilayer perceptron (MLP) model, using only numerical inputs to forecast in the first step. Then, in the second step, they used a CNN network with only image inputs to predict the error value of the numerical network prediction results to correct the prediction. This method achieved the best results in hundreds of experiments with various fusion methods. However, the authors believe that since the two sub-models cannot share information, the information cross-utilisation rate of this method is almost non-existent, and its potential for improvement is relatively small compared to other fusion methods. Another feasible decision-level fusion method is the weight allocation method. In [73] work, the authors used three different models to generate prediction results and ultimately used an adaptive weight allocation system consisting of MLP to allocate weights respectively. The three models jointly obtain the final decision output. As expressed by the authors, this method enables the model's superiority under different weather conditions to be fully reflected with a reasonable allocation of weights.

Classification by Forecast Mechanisms In any form of the predictive model, the model needs to infer future information based on the information collected in the present. In the DL-GSI-IHSF work, the model's inference about the future can be implicitly included in the image feature extraction process, or the inference about sequence features can be implemented using a specialised Recurrent Neural Network (RNN) model. There are currently three mainstream architectures for implementing predictive mechanisms. The first method is the implicit method, which assumes that the spatial feature encoder implicitly includes the time feature, and the spatial features establish the mapping relationship between the current input $\mathbf{X}$ and the future feature vector $\mathbf{y}_{t+\Delta t}$. For instance, in CNNs, Equation 2.5 can be rewritten as follows.

$$\mathbf{y}_{t+\Delta t} = F_{CNN}(\mathbf{X}_t, \mathbf{W}_{CNN}) \tag{2.19}$$

Due to its nature, this method has strong usability and is used as a benchmark model in many machine vision prediction tasks. Moreover, since the patterns are non-recursive and the convolutional computation supports massive parallelism, making it is less computationally expensive than other methods. However, we believe this direct inference method lacks internal reasoning for time features within the model, thus resulting in limited interpretability. The second method uses a recurrent neural network to extract spatially encoded image features, such as LSTM. This approach usually involves simultaneously establishing a mapping relationship between images and feature vectors through a spatial feature encoder. Then, a time sequence of spatial feature vectors is constructed by parallelising multiple feature vectors. The next moment's spatial feature vector is recursively obtained by searching for patterns in this sequence. The following formula can represent this process:

$$\mathbf{y}_{[t,t-\Delta t,t-2\Delta t,\ldots]} = F_{CNN}([\mathbf{X}_t, \mathbf{X}_{t-\Delta t}, \mathbf{X}_{t-2\Delta t}, \ldots], \mathbf{W}_{CNN}) \tag{2.20}$$

$$\mathbf{y}_{t+\Delta t} = F_{RNN}(\mathbf{y}_{[t,t-\Delta t,t-2\Delta t,\ldots]}, \mathbf{W}_{RNN}) \tag{2.21}$$

The recursive nature of RNNs allows for continuous prediction within the same model, referred to as multi-step forecasting. It should be noted that models performing extrapolation use the output feature vector from the previous step as the input, causing error accumulation and performance degradation in long sequence prediction. In addition, the nonparallelisability of recursive computations results in lower computational efficiency for models with similar complexity than the first method.

The third method involves hybrid network architectures, such as 3D-CNN [94] and ConvLSTM [95], which simultaneously encode spatial and temporal information. This approach typically takes a sequence of images as input and extracts both spatial and temporal features. 3D-CNN is a typical spatiotemporal encoding network architecture, which requires stacking the images in the channel dimension to form a time series and sliding the convolutional kernels into three dimensions to capture spatiotemporal correlations. Compared to standard CNNs, this method can better capture temporal information. Another typical spatiotemporal network architecture is ConvLSTM, which evolved from the standard LSTM network. Unlike LSTM, ConvLSTM takes a four-dimensional sequence of images rather than a sequence of feature vectors as input. Replacing matrix multiplication with convolutional operations within the LSTM architecture extracts features directly from continuous three-dimensional matrices and performs recurrent calculations. The generalisation process for this type of approach can be expressed as follows:

$$\mathbf{y}_{t+\Delta t} = F_{STN}([\mathbf{X}_t, \mathbf{X}_{t-\Delta t}, \mathbf{X}_{t-2\Delta t}, \ldots], \mathbf{W}_{STN}) \tag{2.22}$$

The STN abbreviation stands for the spatiotemporal network.

Current research has shown that ConvLSTM is one of the state-of-the-art models among multimodal models. Its 10-minute FS reaches 20.4%, which means 20.4% reduces its Root Mean Square Error (RMSE) loss compared to the baseline (intelligent persistence) model. However, the model still has limitations, such as lower sensitivity to sudden solar slope events and higher training costs than LSTM. Having reviewed all the available articles [50, 55, 42], we believe that the one conclusion that can be drawn is that methods using temporal inference modules outperform implicit methods in terms of both qualitative and quantitative predictors.

2.3.2 Backbone Network

In this section, we focus on the backbone models used in the models mentioned in the review, namely the feature encoders. Depending on the different roles of the encoders, we classify them into spatial, temporal, spatiotemporal hybrid and multimodal fusion encoders. In addition, we list several reference-worthy cutting-edge deep learning models that have not yet been applied in solar forecasting.

2.3.2.1 Spatial Feature Encoders

The image feature encoder is a crucial component of computer vision-based deep networks. Two mainstream image feature encoders in deep learning are Convolutional Neural Network (CNN) [96, 97, 98, 99] and Vision Transformer (ViT) network [100]. Due to the long development history and more extensive related research, the CNN network has been more thoroughly studied in the field of image analysis. Thus there is a richer body of work related to solar energy forecasting. In contrast, the ViT network was first proposed in 2020 and has been widely acclaimed for its outstanding performance in deep learning. However, due to its more complex model structure, shorter research time, and higher training costs, it has not yet received widespread attention in solar energy forecasting.

Convolutional Neuron Networks Convolutional Neural Network (CNN) are a type of deep neural network used for image recognition and processing, inspired by the workings of neurons in the visual cortex. At the heart of a CNN is the convolutional layer, which uses convolution kernels to extract local features from an image. Each convolutional layer contains multiple convolutional kernels that simultaneously convolve the model. The computation process can be represented as:

$$[\mathcal{H}]_{i,j,n} = \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} [\mathcal{V}]_{a,b,c,n} [\mathcal{X}]_{i+a,j+b,c} \tag{2.23}$$

Where $\mathcal{X}$ is the input tensor (image) with length $i+a$ and width $j+b$ and channel c, $\mathcal{V}$ is the number of n convolutional kernels with length a and width b and channel c, $\mathcal{H}$ is the feature tensor of input image under convolutional kernels, Δ is convolution kernel sliding boundaries. By stacking multiple layers of convolutional and pooling operations, a CNN can learn hierarchical and abstract representations of an image, which can then be mapped to corresponding labels or categories. As shown in Figure 2.2

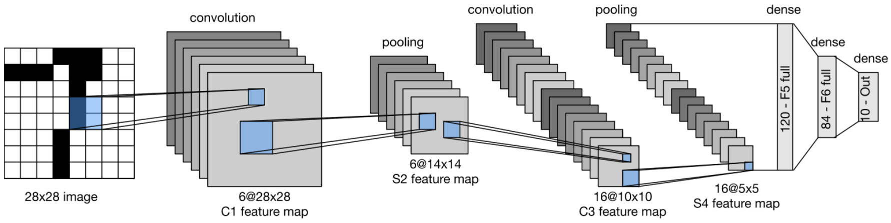


Figure 2.2: Data flow in LeNet [96], Figure token form [101]

Compared to traditional fully connected neural networks, CNNs have the properties of weight sharing and local connectivity, significantly reducing network parameters and computation costs and enhancing training efficiency and generalisation ability. In a CNN, each convolution kernel only connects with a small local area of an image. These convolution kernels share the same set of parameters, enabling them to extract and represent features of an image in a parameter-efficient manner. CNNs include pooling layers, which downsample feature maps, reducing their size while retaining essential features.

In recent years, CNNs have achieved significant progress in areas such as image classification, object detection, and semantic segmentation, making them a key research focus in deep learning. For instance, AlexNet [97], VGGNet [98], ResNet [99] and DenseNet [102] are CNN models that have achieved good results in image classification competitions. It is, therefore, widely used in image coding work for solar irradiation forecasts.

Vision Transformer The practical application of Vision Transformer (ViT) has yielded an impressive repertoire of state-of-the-art performances in several computer vision tasks [100]. Specifically, ViT has surpassed CNN models in the highly esteemed ImageNet image classification competition [103]. Furthermore, ViT has demonstrated highly competitive performances in other computer vision tasks, such as object detection and semantic segmentation. These remarkable achievements have opened up a new vista for ViT, positioning it as a highly viable alternative to conventional CNN models.

Aside from its outstanding performance, ViT possesses several salient characteristics that have engendered its widespread application in computer vision. One such attribute is its strong interpretability, making it an attractive option for tasks that demand interpretability, such as medical diagnosis [104]. This remarkable feature of ViT is based on the fact that it leverages self-attention mechanisms to encode an input image into a sequence of tokens, facilitating straightforward interpretation, as shown in Figure 2.3 below. In contrast to CNNs, the logical inference of ViT relies on the correlation between image patches defined by the positional embedding. The model is sensitive to absolute and relative spatial relationships between image patches. Therefore, we consider ViT a promising model architecture for solar forecasting work. However, despite its outstanding

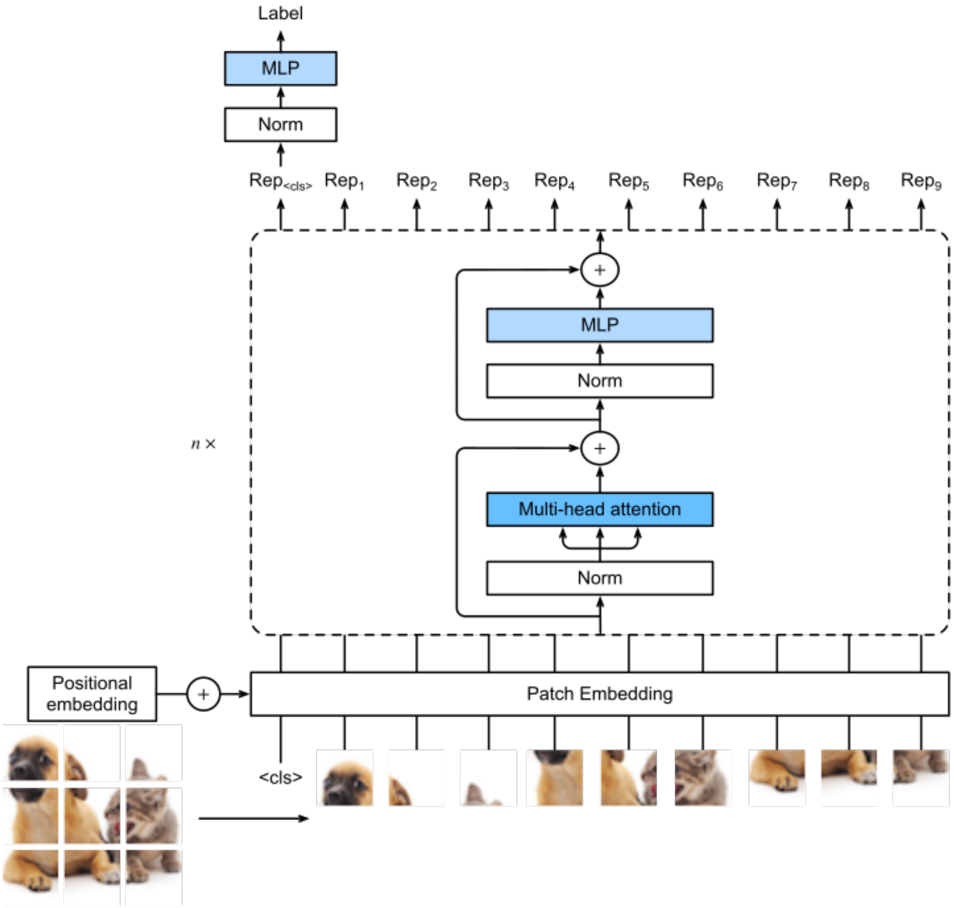


Figure 2.3: Architectural of ViT [100], Figure token form [101].

performance and exceptional interpretability, ViT still presents some challenges that must be addressed. These challenges include significant computational and memory consumption, prolonged training and inference time, and the need for further research to enhance its computational efficiency and performance. Addressing these challenges is critical to ensuring that ViT remains a widely applicable technology in computer vision [105].

Furthermore, in the field of solar energy prediction, the utilisation of ViT is limited and presents several challenges. To our knowledge, only a meagre number of works have exploited ViT in the context of solar irradiance prediction [106]. Regrettably, these works exhibit considerable variations in their model data sets and evaluation metrics compared to conventional deep learning frameworks based on CNNs. Therefore, the overall effectiveness of ViT in solar energy prediction is difficult to ascertain and remains uncertain.

Nevertheless, there is a growing need for better and more accurate prediction models in solar energy. The utilisation of ViT in this domain presents an opportunity to overcome some of the limitations of conventional deep learning frameworks. However, addressing the challenges above and limitations is necessary to leverage the full potential of ViT in solar energy prediction.

2.3.2.2 Temporal Feature Encoder

In the deep learning-based GSI-IHSF framework, inter-temporal feature extraction and future feature prediction are promising directions. As mentioned earlier, the extraction of serialised information relies on Recurrent Neural Network (RNN), which requires serialised input items. Among them, Long Short-Term Memory (LSTM) is the most popular serialised feature extraction model. It can effectively capture continuous sequential patterns and long-term data dependencies and has received the most extensive attention and research in solar energy prediction. Meanwhile, due to the rapid development of deep learning, some emerging models, such as convolutional gated recurrent networks, have also started to attract the attention of researchers.

Long and short-term memory networks Long Short-Term Memory (LSTM) is a highly specialised RNN type widely applied in several fields, including time series prediction and natural language processing. The salient characteristic of LSTM is its unique loop unit structure, composed of input, forget, and output gates, as shown in the equation below. These gates are highly effective in mitigating the issues of gradient vanishing and exploding that are often encountered in traditional RNN models when dealing with long sequences.

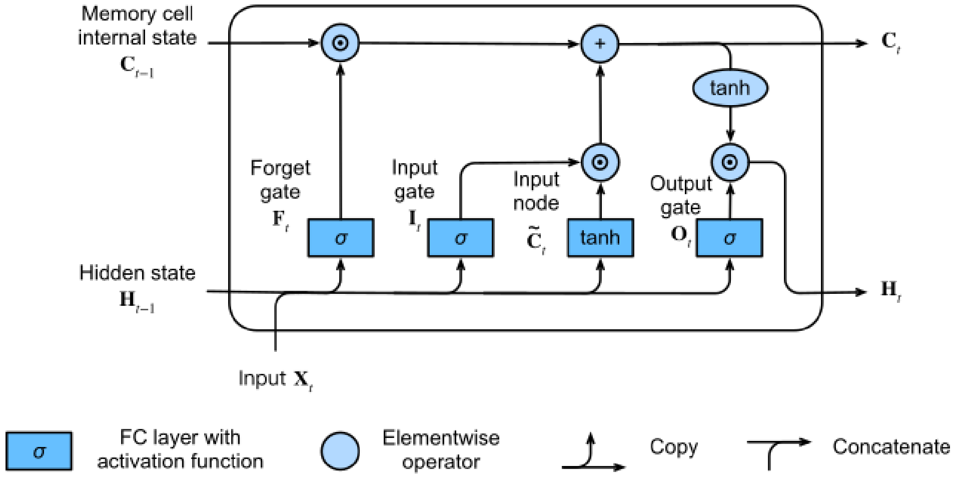

Memory cell internal state $\mathbf{C}_{t-1}$
$\mathbf{C}_t$
tanh
Forget gate $\mathbf{F}_t$
Input gate $\mathbf{I}_t$
Input node $\tilde{\mathbf{C}}_t$
Output gate $\mathbf{O}_t$
σ
σ
tanh
σ
Hidden state $\mathbf{H}_{t-1}$
$\mathbf{H}_t$
Input $\mathbf{X}_t$
σ
FC layer with activation function
Elementwise operator
Copy
Concatenate

Figure 2.4: Recurrent Unit of LSTM [107], Figure token form [101].

$$\mathbf{I}_t = \sigma(\mathbf{X}_\mathrm{t}\mathbf{W}_\mathrm{xi} + \mathbf{H}_{\mathrm{t}-1}\mathbf{W}_\mathrm{hi} + \mathbf{b}_\mathrm{i}), \tag{2.24}$$

$$\mathbf{F}_t = \sigma(\mathbf{X}_\mathrm{t}\mathbf{W}_\mathrm{xf} + \mathbf{H}_{\mathrm{t}-1}\mathbf{W}_\mathrm{hf} + \mathbf{b}_\mathrm{f}), \tag{2.25}$$

$$\mathbf{O}_t = \sigma(\mathbf{X}_\mathrm{t}\mathbf{W}_\mathrm{xo} + \mathbf{H}_{\mathrm{t}-1}\mathbf{W}_\mathrm{ho} + \mathbf{b}_\mathrm{o}), \tag{2.26}$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{X}_\mathrm{t}\mathbf{W}_\mathrm{xc} + \mathbf{H}_{\mathrm{t}-1}\mathbf{W}_\mathrm{hc} + \mathbf{b}_\mathrm{c}), \tag{2.27}$$

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t, \tag{2.28}$$

$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh(\mathbf{C}_\mathrm{t}) \tag{2.29}$$

where $\mathbf{I}$, $\mathbf{F}$, $\mathbf{O}$ are the input, forgetting and output gates, respectively, which are determined by the hidden state $\mathbf{H}_{t-1}$, i.e. short-term memory from last moment, with input $\mathbf{X}$, and bias term $\mathbf{b}$ through the Sigmoid activation function. Thereafter, $\mathbf{I}$, $\mathbf{F}$, $\mathbf{O}$ gating for the input node $\tilde{\mathbf{C}}$, the previous moment's long term memory $\mathbf{C}_{t-1}$ and the current long term memory $\mathbf{C}$, respectively, to determine the direction of data flow.

LSTM is famous for forecasting future sequence values in time series prediction. The fundamental idea behind LSTM is to predict future state values by incorporating historical and current input information. Compared to traditional RNN models, LSTM is better suited for handling long sequences and long-term dependencies and exhibits enhanced generalisation and robustness.

Despite its many advantages, LSTM still has certain limitations. One of the common problems with RNN networks is the long training time, as recurrent networks require recursive computation and, therefore, cannot be massively parallelised, resulting in slower training relative to parallelisable networks. In addition, when dealing with highly long sequences, the LSTM still suffers from gradient disappearance, failing the model prediction function.

In solar energy prediction, LSTM is commonly used to predict future sequence values in serialised image feature vectors to achieve future state prediction. In the early work of Zhang, LSTM based on image feature sequences achieved the best performance in quantitative prediction, outperforming CNN works without a time encoder under all weather conditions. Recent work in DL-GSI-IHSF, LSTM models have been widely used in some state-of-the-art composite model architectures [48, 86, 78, 79, 56, 43, 45, 53].

Convolutional Gated Recurrent Network Convolutional Gated Recurrent Network (CGRN) [108] is an advanced deep learning model that combines the strengths of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), specifically designed for processing time series data. The unique architecture of CGRN consists of a complex combination of convolutional layers, recurrent layers, and gated mechanisms. The convolutional layer is responsible for extracting local features from the input data, while the recurrent layer simulates time dependencies in the data. The gated mechanism, inspired by the use of gates in Long Short-Term Memory (LSTM), controls the flow of information between different layers and helps to avoid the problem of vanishing gradients.

Recent studies [35] have applied this framework to predict the future states in solar energy prediction, achieving significant progress. Results show that as a single-modal model with an only image input, its prediction score surpassed ConvLSTM [42], based on the two modalities of sky images and meteorological data. The framework and its subsequent study [11] have achieved state-of-the-art performance in deep learning.

Compared to solar energy prediction models based on LSTM, CGRN has several advantages. Firstly, it can capture local and global features of time series data by utilising 3D convolutional layers. Secondly, it can simulate sequence relationships in the data by adopting recurrent layers and gated mechanisms. Thirdly, because the convolutional layer dramatically reduces the required recurrent connections, it is more computationally efficient than traditional RNNs.

The application of CGRN in solar energy prediction has achieved significant achievements, providing unprecedented accuracy and stability in predicting future states. The emergence of CGRN represents a significant breakthrough in deep learning, bringing enormous potential for future research and practical applications in different fields.

2.3.2.3 Spatio-temporal Feature Hybrid Encoder

A spatiotemporal feature fusion encoder is an architecture that integrates spatial and temporal feature search mechanisms within a single encoder module. This encoder type achieves joint feature extraction by complex integration, making it difficult to analyse its interpretability independently. Currently, validated spatiotemporal feature encoders that can be used for solar energy prediction include 3D-CNN [94], ConvLSTM [95], PRED-Net [109] and PhyDNet [110].

3D-CNN Three-dimensional convolutional neural networks (3D-CNNs) are an iterative version of the CNN network. In the original work, the authors aimed to improve the model's ability to analyse multi-layer brain MRI images by adding a dimension to the 2D convolution kernel to enable 3D convolution [94]. This method is widely used as a spatiotemporal joint feature encoder for image stack sequences, as it can extract cross-image features while extracting 2D image information.

The structure of 3D-CNN is quite similar to that of 2D-CNN, but with the addition of a time dimension. The network takes in a sequence of three-dimensional volumes, each corresponding to a frame in an image sequence. The network comprises convolutional layers, pooling layers, and fully connected layers, with the primary objective of learning spatial and temporal features from the input data. The formula of 3D convolutional can be described as:

$$[\mathcal{H}]_{i,j,k,n} = \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} \sum_{c=-\Delta}^{\Delta} [\mathcal{V}]_{a,b,c,d,n} \, [\mathcal{X}]_{i+a,j+b,c,k+d} \tag{2.30}$$

In solar energy prediction, 3D-CNNs have been applied in multiple works and have been proven effective in capturing spatial and temporal patterns, thus improving prediction accuracy [47, 82, 35]. However, recent works [42] have shown that 3D-CNNs still lag behind state-of-the-art models in solar energy prediction. Therefore, further research is needed to address this limitation and improve the performance of 3D-CNNs in solar energy prediction. In addition, due to the increased computational complexity of the encoder, 3D-CNNs also have a significantly higher computational cost.

ConvLSTM In solar forecasting, ConvLSTM (Convolutional Long Short-Term Memory) is a cutting-edge deep learning model that combines the strengths of both convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. ConvLSTM is designed to handle complex spatiotemporal data, such as sky and satellite images, by embedding spatial feature extraction in a time sequence model.

Like LSTM, ConvLSTM includes a memory cell that retains information over time and three gates that control the flow of information in and out of the cell. However, in ConvLSTM, the matrix multiplication operation in LSTM is replaced with convolutional operations, enabling the model to capture spatial features. In ConvLSTM, the input is a 4D tensor representing a sequence of 3D matrices. The network processes data in space and time, enabling it to model spatial and temporal dependencies in the data. The formula can express the process

Compared to the concatenation of CNN+LSTM structure, ConvLSTM offers several advantages. First, it can jointly model spatial and temporal features in the data, allowing it to deal with more complex spatiotemporal patterns. Second, using embedded convolutional calculations, ConvLSTM is more computationally efficient than CNN+LSTM networks.

Kong et al. [55] first introduced the ConvLSTM model in DL-GSI-IHSF and found that it could capture the most Ramp Events (RE) among all the models, achieving a balanced accuracy of 74.75%. However, the authors also discovered that, despite capturing the most RE, the ConvLSTM model still lagged behind the LSTM model in quantitative RE measurements. In Paletta et al. [42] benchmark study, the authors systematically compared four state-of-the-art models, including CNN, LSTM, 3D-CNN, and ConvLSTM, and found that the ConvLSTM model still exhibited superior performance among the models. The authors attributed this to its good spatiotemporal feature extraction relationship. However, the authors also found that in two ultra-short-term prediction scales, 2-minute and 6-minute lead prediction, the performance of the ConvLSTM model was lower than that of the LSTM and 3D-CNN models. At the same time, a common problem with all fusion models is that they still behave like a very smart SPM, lacking pre-judgment of RE.

However, ConvLSTM also has some limitations. As a highly complex model, interpreting it remains a significant challenge. Furthermore, as an LSTM model based on RNNs, linear training processes require significant computational resources, rendering training cost, debugging, and deployment as limiting factors.

Deep Predictive Coding Networks Deep Predictive Coding Network (PredNet) is a self-supervised deep learning architecture for predicting future frames in videos [109]. Inspired by the workings of the human brain, which is more interested in unexpected elements than expected ones, the network simulates a neural system consisting of four modules: input, representation, prediction, and error. At each time step, the model uses a representation module composed of ConvLSTM and a prediction module to jointly make image predictions and compare them with the input brought by the input module. The error module then calculates the "unexpected" error portion. Finally, the error is fed back to the representation module for training and as input for predicting the next time step. At the same time, the representation module of the next time step returns data to guide the current representation module.

Unlike all other methods mentioned, PredNet is a self-supervised or unsupervised model. The model's predictive behaviour is recursive, with the input of the next frame being the current frame's expected output and the current frame's input being the expected output of the previous frame. This unsupervised approach is a double-edged sword, as the data set can be trained without a clear goal, but at the same time, the model cannot be constrained by specifying a goal. In addition, unsupervised models require higher-quality data sets than supervised learning.

A feasibility test was performed on solar energy direction prediction using PredNet [55]. The authors extracted the feature vectors from the PredNet representation module at each time step and used a fully connected layer to generate predictions. The results show that PredNet has competitive quantitative and qualitative predictive performance. However, the authors also found that during periods of stable irradiance, the prediction error of PredNet was still more considerable than that of other models. In addition, due to the cumulative effect of error in recursive models, the prediction for cloud boundaries in the image is unclear.

Although this method has not achieved optimal performance, self-supervised learning, as a more objective approach, still has great potential in solar energy prediction.

Physical Dynamics Network Physical Dynamics Network (PhyDNet) [110] is another popular video frame prediction network, which is also trained through self-supervision by recursive behaviour, i.e., each input frame is also the expected output of the previous frame. The highlight of this model is that it has a potential physical constraint on the predicted content. In the prediction module of the model, there are two branches: the first one is the recursive model prediction through ConvLSTM, and the second one is the explicit physical constraint, i.e., partial differential equations dynamics, for prediction. Both are combined by addition and complement each other.

In the experimental case of solar forecast [74], the authors found that physical constraints only require added few parameters and can outperform the single-mode ConvLSTM that only inputs images in quantitative performance. We believe that the importance of PhyDNet lies in its exploration of feasible methods for integrating physical constraints into deep learning models, which is crucial for the interpretability and robustness of the model.

2.3.2.4 Feature Fusion Encoder

The term "feature fusion encoder" refers to the part of the fusion process that encodes the fused information to enable effective interaction between different modalities. While many studies have explored fusing two or more modalities for solar energy prediction, relatively little attention has been paid to the effectiveness and efficiency of feature-level fusion. This section introduces a commonly used fusion architecture for solar energy forecasting and two potential fusion mechanisms applied in other fields.

Multi-layer Perceptron Multi-layer perceptron (MLP) is a basic, fully connected neural network model with activation functions. It is the most common network architecture to process fused features in multimodal solar energy prediction networks. This approach makes no assumptions about the semantic space heterogeneity of different modalities and adapts the process of modality interaction through weight allocation between neurons. During modality interaction, feature vectors from different modalities are fused additively or multiplicatively and then input to the MLP for the following calculation. The most commonly used fusion method is concatenation, which can be considered a particular case of additive calculation without data interaction or exchange after the additive operation.

In the work of [71], different concatenation methods for modality fusion were explored. They found that designing an enhanced modality interaction by separately sending PV log and image data to their own fully connected layers and simultaneously sending the joint data to a shared fully connected layer could significantly improve the model's prediction ability. However, this method has not been used in subsequent research.

Attention Mechanisms In deep learning, attention can be viewed as a differentiable dictionary retrieval process [111]. The principle of the attention mechanism is to use a regularised dictionary that stores many specific key-value pairs (k, v). When a specific query q is given, the dictionary searches for a matching k and retrieves the corresponding v as the model's return. This type of machine is widely used in natural language processing to retrieve the relative relationships between word elements within a sentence, or in visual image tasks, as in the previously introduced ViT. The advent of self-attentive mechanism networks has revolutionised the realm of multimodal learning by enabling the identification of correlations between features via vector embeddings. This unique approach facilitates integrating data from diverse modalities within the attention module. Unlike the commonly used tandem or fixed-weight fusion techniques, the attention-based methodology can dynamically balance the contributions of distinct modalities, thereby enabling optimal utilisation of the available data sources [89].

Groundbreaking research by Long et al. [112] has demonstrated the remarkable potential of attention-based mechanisms for video classification. This technique has exhibited remarkable robustness across multiple data sets in all cases of data-level fusion, feature-level fusion, or decision-level fusion. These remarkable findings have far-reaching implications and underscore the importance of adopting cutting-edge techniques for optimal data analysis and processing. In the current DL-GSI-IHSF literature, the use of attention mechanisms as input encoders is minimal. In Zhen et al. [79] recent work, attention mechanisms were embedded in an LSTM structure to assist in exploring the effective sequence length for future predictions. The attention mechanism confirmed the general prior knowledge that time points further away from the present have a minor potential impact on future predictions. The authors used this method to quantify the attention distribution and obtained a reasonable value for the length of the input LSTM network time step.

Gating Mechanisms The gate mechanism is another structure in deep learning networks crucial for regulating the computation, and it is a mechanism used in LSTM and recurrent gated unit (GRU) networks. The gate mechanism controls the information flow by multiplying a neuron's output vector with the weight coefficients element by element. Unlike attention mechanisms, which normalise attention weights, gate mechanisms do not distort the specifics of the information flow [111]. For instance, the input, forget, and output gates in the LSTM network are structures. The gates are opened or closed through the Sigmoid function, limiting the output to 0 and 1.

Applying gating mechanisms in solar forecasting has been limited to LSTM working in spatial coding. In the process of modal fusion, to our knowledge, this approach has not been used.

2.3.3 Model Adjustment

As mentioned, in DL-GSI-IHSF applications, the mainstream approach is to transfer mature architectures from computer vision networks and train them. However, replicating a complete deep-learning training process is extremely difficult. First, the hardware and software platforms the model runs usually have different models and version differences. Therefore, it is difficult to ensure that the differences will not affect the model calculation in learning based on different software and hardware versions. Secondly, hyperparameters have a significant impact on model performance. Therefore, much work is needed to optimise hyperparameters, and the settings of hyperparameters need to be meticulously recorded and logged during the tuning process. Finally, even in the same training process, there are inevitably random factors, such as the order of image data in the input batch or the stochasticity of the Stochastic Gradient Descent (SGD) optimiser itself.

Therefore, model debugging is a complex, time-consuming, and essential part of deep learning architecture. This section introduces model debugging from four aspects: model architecture adjustment and verification, hyperparameter tuning, reducing randomness, and transfer learning.

2.3.3.1 Model Architecture Adjustment and Validation

Optimising deep learning network architectures is an essential component in building effective models. When replicating model architectures for the GSI-IHSF task, optimising the models can ensure their efficacy and improve their performance and efficiency.

Reducing Model Complexity In deep learning-based DL-GSI-IHSF tasks, comprehensive computer vision networks are often designed to tackle more complex and diverse tasks. However, DL-GSI-IHSF tasks themselves are relatively less complex. For example, in the work of Wen et al. [77], it was found that using deeper networks, such as ResNet-34 or ResNet-50, or more complex models, such as DenseNet, did not significantly improve the performance of a CNN model used for processing sky images, but required high additional computational cost. Similarly, in the work of [80], it was also pointed out that using deeper networks, such as ResNet-152, only improves the model's performance by less than 1% than ResNet-34. Therefore, pruning the model by removing unnecessary depth can significantly reduce the model size and computation time and save computational resources.

Ablation Experiments Rigorous experiments that remove specific model components and evaluate the resulting performance or feature representation are called ablation experiments. Ablation experiments are a crucial step in assessing the effectiveness and necessity of model design choices. The specific method of removing model components can involve directly removing modules or rendering them ineffective through meaningless noise or zero inputs. As mentioned, many DL-GSI-IHSF models are adapted from mainstream computer vision models. As such, many studies overlook the fundamental need for ablation experiments in validating model design choices. Thus, we believe it is necessary to perform ablation studies when adding new modules to the model.

2.3.3.2 Hyperparameter Tuning

Hyperparameters refer to model parameters that cannot be iteratively updated during training, such as learning rate, batch size, optimiser parameters, number of neurons, and network layers. Grid and random search are two mainstream methods for hyperparameter tuning. These methods exhaustively enumerate all possible combinations of hyperparameters to obtain the optimal solution. However, the disadvantage of this approach is that the computational complexity grows exponentially with the number of hyperparameters, making it extremely expensive to tune a model's hyperparameters ideally [113]. Additionally, not all hyperparameters affect the model's accuracy equally, and the model is often sensitive to some hyperparameters but not others. Therefore, it is necessary to restrict the hyperparameter tuning matrix. Another feasible way is to use machine learning, such as Bayesian optimisation, which treats hyperparameter tuning as a regression problem and gradually improves the model's performance in optimisation.

Not all GSI-IHSF works based on deep learning will elaborate on the hyperparameters used. Some argue that the impact of hyperparameter tuning on model performance is far lower than that of model architecture, and hyperparameter tuning requires a significant amount of resources and computational cost. Therefore, in some works, the original hyperparameters of the referenced model were directly used without further optimisation. However, we believe that demonstrating the hyperparameter adjustment process or publicising the hyperparameters is an essential part of the replication in the DL-GSI-IHSF work. Some important hyperparameters include learning rate, optimiser type, and batch size. It can have a direct and significant impact on model performance or model calculation efficiency, so it is necessary to publicise the underlying hyperparameters.

2.3.3.3 Reduced Randomness

The randomness of the model during training manifests itself in various aspects. For example, Python libraries such as numpy and pandas used in model training have randomness in computation, and different order of batch sampling directly affects the model fitting process. The SGD optimiser itself is based on stochastic sampling for gradient computation. Many initialisers for network weight initialisation are based on random generation, which directly impacts the model's performance. Fixing the seed of the random generator is a standard method to avoid some random factors causing random fluctuations in model performance. For some randomness that cannot be eliminated, k-fold cross-validation is also an effective method [114]. Many work [47, 33, 34] applied this method by dividing the data into ten parts, each time taking one part as the training set and one part as the validation set without repetition, and the average of the ten results is taken as the model accuracy. In addition, some studies use specific methods to combat random numbers.

2.3.3.4 Transfer Learning

Transfer learning refers to storing the knowledge obtained during a model's training and applying it to another, different but related problem. Training weights from scratch in a large or complex model is called pre-training in deep learning, and loading pre-trained weights into a downstream task as initial weights and further training is called fine-tuning. Fine-tuning with pre-trained weights in CNNs has proven effective in knowledge transfer, even in tasks with significant differences, such as natural image classification to medical grayscale images. Research has shown that fine-tuned pre-trained CNNs can perform as well as CNNs trained from scratch, even in the worst-case scenarios of such significant transfers [115]. Fine-tuning can reduce training time and still work on small-scale data sets. This method can also be applied in the DL-GSI-IHSF domain. Recent studies have found that pre-training on a solar irradiance model trained on two large data sets can be successfully transferred to new sub-tasks in different climates [31]. This transfer can reduce training time by four-fifths and slightly improve model accuracy. The authors speculate that this may be due to the inadequate scale of global training. We believe that classical solar statistical prediction models exhibit significantly different performances in the same model in different climates. In other words, the weights of the local model implicitly contain features of the local climate conditions. Global learning may not effectively capture these features without additional annotations during training, thereby failing to establish a practical model. Therefore, the ideal use of transfer learning may require constraints on climate models for different locations to improve model performance. Additionally, this research suggests the feasibility of establishing a universal model framework for prediction under different climates, which can promote the exchange and development of prediction models in different climates.

2.3.4 Prediction Head

The prediction head refers to the part of the model that maps features extracted by the backbone to specific output results. Implementing transfer learning often involves adapting the prediction head after transferring the pre-trained model locally. In this section, we will explain the model's prediction head from two aspects: output format and prediction target, and summarise some experiences from related works.

2.3.4.1 Deterministic Prediction and Probabilistic Prediction

Deterministic and probabilistic predictions refer to two different output forms of the prediction head. They differ in reliability, accuracy, and usage scenarios. As the name suggests, deterministic prediction outputs a single numerical value, such as $1000W/m^2$. This output is the unique and definite result directly from the neural network, without including errors or uncertainties. Although this output form is currently the mainstream output mode in DL-GSI-IHSF, its limitations are evident. IHSF applications often require high reliability for short-term grid control. In this context, the practicality of deterministic prediction is limited.

Probabilistic prediction incorporates model uncertainty into the design of the prediction head. For example, for classification prediction, such as sky condition or slope event prediction, the probability value of each predicted result can be output as the prediction results through the Sigmoid or Softmax activation function. This prediction head based on probabilistic prediction can be further designed with a judgement threshold through statistical methods to improve the reliability and robustness of prediction. For numerical prediction, a specific loss function or model architecture can be designed to give the confidence interval of the model prediction to cope with different random times. There are two specific implementation methods: the first is to use the quantile loss function, which assigns different weights to the model's overestimation and underestimation to achieve fixed confidence interval prediction [43]. The second method is to change the regression problem into a classification problem, that is, to transform the output from a numerical value into a numerical interval [35]. The model's confidence interval is determined by the probability distribution of the model's prediction in the numerical interval. The second method may lack mathematical rigour, even if it produces seemingly correct results. Criteria for analysis and evaluation explicitly based on model performance are discussed in the next section.

2.3.4.2 Targets

According to the different subsequent demands of solar forecasting, implementing prediction targets can be diverse. For example, for irradiance prediction, it is feasible to directly use irradiance as the prediction target, the irradiance change rate as the prediction target, or the CSI as an indirect prediction target. However, overall, there are two prediction targets for solar energy forecasting. The first is to quantitatively predict future solar energy output, such as irradiance prediction (indirect PV output prediction) or direct PV output prediction, to estimate future PV production. The second is to predict sudden solar energy events caused by cloud cover, also known as ramp events, to reduce the negative impact of sudden events on PV systems and power generation quality.

Multi-Task Learning In deep learning, multi-task learning uses the same model feature vector to predict multiple targets through multiple prediction heads. Specifically, multi-task learning can be achieved through two different modes. The first is hard parameter sharing, which maps the same feature vector to multiple targets. The second is soft parameter sharing, where their models still implement multiple tasks. However, an additional regulariser is set to encourage the similarity of parameters between the models. Ideally, when a deep model learns multiple tasks simultaneously, the noise topology of each task can be ignored to obtain a more general representation [116]. In DL-GSI-IHSF, some work has already used multi-task learning, such as Zhang et al. [50], who used two prediction heads to strengthen the constraints on different temporal or spatial prediction sub-tasks, achieving good results. Through multi-task learning, their model obtained a 20.8% improvement in RMSE prediction performance compared to the persistence model in one-minute prediction. Zhang et al. found that their model obtained better-balanced performance under different weather conditions, explicitly improving performance on cloudy and sunny days while slightly decreasing performance on partly cloudy days. Paletta et al. [35, 11] also used multi-task learning, finding that segmentation tasks as prediction sub-tasks with a cross-entropy loss could comprehensively improve the model's F1 score at all scales, achieving nearly 4% improvement at the 2-minute scale and nearly 2% improvement at the 10-minute scale.

Table 2.2: Summary of model architectures, prediction times, prediction head design and optimisation methods used in current work in the DL-GSI-IHSF work.

Ref.	Year	Backbone Model				Forecast Time[1]			Target attributes[2]			Optimisation algorithms[3]
		Spatial Encoders	Temporal Encoders	Numerical Encoder	Fusion Encoder	FH	FR	FLT	Types	TS	Target	
[32]	2018	CNN	-	-	-	0^*	-	-	Det.	O	PV Output	AO
[50]	2018	CNN	LSTM	MLP	MLP	10	1	1	Det.	O	PV Output	BC, HA
[69]	2018	CNN	-	MLP	MLP	15	15	15	Det.	O	PV Output	BC, AS
[33]	2019	CNN	-	MLP	MLP	15	15	15	Det.	O	PV Output, CSI	AO, AS, HA
[71]	2019	CNN		MLP	MLP	15	15	15	Det.	O	PV Output	AF, HA
[54]	2019	MLP	-	-	-	5	1	1	Det.	O	GHI	AO
[58]	2019	CNN	-	MLP	MLP	20	5	5	Det.	O	GHI	BC
[47]	2019	3D-CNN	3D-CNN	MLP	MLP	30	10	10	Det.	O	DNI	HA

continued in next page...

Table 2.2 Summary of model architectures, prediction times, prediction head design and optimisation methods used in current work in the DL-GSI-IHSF work (Continued).

Ref.	Year	Backbone Model				Forecast Time			Target attributes			Optimisation algorithms
		Spatial Encoders	Temporal Encoders	Numerical Encoder	Fusion Encoder	FH	FR	FLT	Types	TS	Target	
[43]	2019	CNN	LSTM	-	-	10	10	10	Det., Prob.	O	GHI	AO, BC, HA
[56]	2019	CNN	LSTM	LSTM	LSTM	240	60	60	Det.	M	GHI	BC, HA
[57]	2020	CNN(VGGNet)	-	-	-	60	10	10	Det.	O	GHI	HA
[59]	2020	CNN	-	-	-	0	-	-	Det.	O	PV Output	HA, AO
[72]	2020	CNN(VGGNet)	-	-	-	10	10	10	Det.	O	GHI	HA
[73]	2020	CNN, MLP	LSTM	MLP	MLP	15	15	15	Det.	O	GHI	AO, AF, BC

continued in next page. . .

Table 2.2 Summary of model architectures, prediction times, prediction head design and optimisation methods used in current work in the DL-GSI-IHSF work (Continued).

Ref.	Year	Backbone Model				Forecast Time			Target attributes			Optimisation algorithms
		Spatial Encoders	Temporal Encoders	Numerical Encoder	Fusion Encoder	FH	FR	FLT	Types	TS	Target	
[74]	2020	ConvLSTM, PredRNN, PhyDNet	ConvLSTM, PredRNN, PhyDNet	-	-	5	1	1	Det.	M	GHI	BC
[75]	2020	MLP	-	-	-	10	1	1	Det.	M	GHI	-
[76]	2020	CNN	-	-	-	0	-	-	Det.	O	GHI	-
[46]	2020	CNN	-	MLP	MLP	20	2	2	Det.	O	GHI	-

continued in next page. . .

Table 2.2 Summary of model architectures, prediction times, prediction head design and optimisation methods used in current work in the DL-GSI-IHSF work (Continued).

Ref.	Year	Backbone Model				Forecast Time			Target attributes			Optimisation algorithms
		Spatial Encoders	Temporal Encoders	Numerical Encoder	Fusion Encoder	FH	FR	FLT	Types	TS	Target	
[77]	2020	CNN (VGGNet, ResNet, DenseNet)	-	-	-	10	1	5	Det.	O	GHI	-
[55]	2020	CNN, ConvLSTM, PredNet	LSTM, ConvLSTM, PredNet	MLP, LSTM	MLP	20	4	4	Det.	M	PV Output	AF, BC

continued in next page...

Table 2.2 Summary of model architectures, prediction times, prediction head design and optimisation methods used in current work in the DL-GSI-IHSF work (Continued).

Ref.	Year	Backbone Model				Forecast Time			Target attributes			Optimisation algorithms
		Spatial Encoders	Temporal Encoders	Numerical Encoder	Fusion Encoder	FH	FR	FLT	Types	TS	Target	
[42]	2021	CNN, 3D-CNN, ConvLSTM	LSTM, ConvLSTM	MLP	MLP	30	2	2	Det.	O	GHI	BC, HA
[34]	2021	CNN	-	-	-	0	-	-	Det.	O	PV Output	DE, HA
[34]	2021	CNN	-	MLP	MLP	15	15	15	Det.	O	PV Output	DE ,HA
[44]	2021	CNN(VGGNet)	-	MLP	MLP	15	15	15	Det.	O	GHI	-
[78]	2021	CNN(AlexNet)	LSTM	MLP	MLP	10	10	10	Det.	O	DNI	AS,BC

continued in next page. . .

Table 2.2 Summary of model architectures, prediction times, prediction head design and optimisation methods used in current work in the DL-GSI-IHSF work (Continued).

Ref.	Year	Backbone Model				Forecast Time			Target attributes			Optimisation algorithms
		Spatial Encoders	Temporal Encoders	Numerical Encoder	Fusion Encoder	FH	FR	FLT	Types	TS	Target	
[79]	2021	CNN	LSTM	LSTM, MLP, Attention	MLP	60	10	10	Det.	O	GHI	BC, AF, AO
[45]	2021	CNN	LSTM	CNN	MLP	15	5	5	Det.	O	PV Output	HA
[80]	2021	CNN(ResNet)	-	-	-	0	-	-	Det.	O	GHI	AO, HA
[81]	2021	CNN(ResNet)	LSTM	MLP	MLP	10	10	10	Det., . Prob.	O	GHI	-
[82]	2021	3D-CNN	3D-CNN	MLP	Attention	30	5	5	Det.	O	CSI	TL, AS, HA
[83]	2021	CNN(ResNet)	-	-	-	0	-	-	Det.	O	GHI	AO, HA

continued in next page. . .

Table 2.2 Summary of model architectures, prediction times, prediction head design and optimisation methods used in current work in the DL-GSI-IHSF work (Continued).

Ref.	Year	Backbone Model				Forecast Time			Target attributes			Optimisation algorithms
		Spatial Encoders	Temporal Encoders	Numerical Encoder	Fusion Encoder	FH	FR	FLT	Types	TS	Target	
[36]	2022	CNN(VGGNet), 3D-CNN	3D-CNN	-	-	60	10	10	Det.	O	GHI	DE
[53]	2022	CNN	-	MLP	MLP	15	15	15	Det.	O	GHI, PV Output	TL, HA, BC
[53]	2022	ConvLSTM	ConvLSTM	LSTM	MLP	15	15	15	Det.	O	GHI, PV Output	TL, HA, BC
[84]	2022	CNN	-	-	-	0	-	-	Det.	O	GHI	AF
[35]	2022	CNN	3D-CNN, CGRN	-	-	10	2	2	Det., Prob.	M	GHI	DE, AS, BC, AF, AH

continued in next page...

Table 2.2 Summary of model architectures, prediction times, prediction head design and optimisation methods used in current work in the DL-GSI-IHSF work (Continued).

Ref.	Year	Backbone Model				Forecast Time			Target attributes			Optimisation algorithms
		Spatial Encoders	Temporal Encoders	Numerical Encoder	Fusion Encoder	FH	FR	FLT	Types	TS	Target	
[11]	2022	CNN	3D-CNN, CGRN	-	-	10	2	2	Det., Prob.	M	GHI	DE, AS, HA
[48]	2022	Manual	LSTM	LSTM	MLP	10	10	10	Det.	M	CSI	HA, AS
[85]	2022	CNN(DenseNet)	-	MLP	MLP	15	15	15	Det., Prob.	M	GHI	-
[86]	2022	CNN	-	LSTM	MLP	60	60	60	Det.	O	GHI	AO, HA
[87]	2023	CNN(AlexNet)	-	-	-	15	1	1	Det.	O	GHI	TL, HA

continued in next page...

Table 2.2 Summary of model architectures, prediction times, prediction head design and optimisation methods used in current work in the DL-GSI-IHSF work (Continued).

Ref.	Year	Backbone Model				Forecast Time			Target attributes			Optimisation algorithms
		Spatial Encoders	Temporal Encoders	Numerical Encoder	Fusion Encoder	FH	FR	FLT	Types	TS	Target	

[1] FH: Forecast horizon, forecast total length in minutes; FR:Forecast resolution, minimum time interval between two forecasts; FLT:Forecast lead time, time from the start of the forecast to the first forecast generation

[2] Det.: Deterministic; Prob.: Probabilistic; TS: Forecast Time Step; O: One-step forecast, or End2End forecast, each model forecast a result; M: Multi-step forecast, or recursive prediction, a model can forecast a sequence of result.

[3] HA: Hyperparameters Announcement; BC: Backbone model Comparison; AF: model Architectural Fusion comparison; AS: Ablation Study; AO: model Architectural Optimisation; DE: Data Enhancement; TL: Transfer Learning.

[*] 0-minute forecast horizon means nowcasting, i.e. extrapolating a forecast target from a picture of the sky at the same moment.

2.4 Analysis Phase

In the analysis phase, researchers need to analyse and compare the results generated by the model after the training process. First, due to the black-box nature of deep learning models, it is necessary to conduct rationality and interpretability analysis on both the model output results and the training weights. Second, to compare the superiority of models in terms of performance, researchers also need to set some metrics to compare the models thoroughly.

2.4.1 Baseline Model

The baseline model is a simple model used to evaluate and compare the performance of other complex models. Typically, the baseline model has some predicted performance but with fewer parameters and a more straightforward structure. Its easy-to-implement nature sets the benchmark accuracy standard for more complex models. In solar energy forecasting, since data sets used in different works are usually collected under different climatic conditions and equipment accuracy, directly comparing models using statistical metrics is not advisable. Therefore, the Persistence Model (PM) or Smart Persistence Model (SPM) is often used as the baseline model for solar energy forecasting.

$$\hat{y_{t+\Delta t}}_{PM} = y_t \tag{2.31}$$

$$\hat{y_{t+\Delta t}}_{SPM} = y_{t+\Delta t_{clr}} \times \frac{y_t}{y_{t_{clr}}} \tag{2.32}$$

y_{clr} represents clear-sky irradiance, which can be obtained from a clear-sky model [49]. The persistence model assumes that irradiance remains constant at the forecast horizon. The intelligent persistence model assumes that the ratio of irradiance to clear-sky irradiance, i.e., the clear-sky index (CSI), remains constant at the forecast horizon.

In DL-GSI-IHSF, researchers often add other models as baseline models in addition to persistence models, such as MLP and LSTM models used as baseline models in previous works [55]. Using additional general deep learning frameworks as baseline learning is essential for improving model interpretability. This method not only provides more references for the model but also helps to understand the strengths and weaknesses of the model in different algorithms through multiple baseline models. At the same time, comparing the prediction details with the baseline models can improve the model's robustness and generalisation.

2.4.2 Evaluation Metrics

Evaluation metrics intuitively represent the gap in model comparison and judgement. In IHSF work, a universal standard is that the model's forecast results should be as close as possible to future measurement results, that is, to obtain a minor quantitative error. Meanwhile, GSI improves the model's response to rapidly changing solar radiation by adding spatial features to the prediction end. This prediction should first be a qualitative behaviour, whether the model can capture the Ramp Event (RE). Secondly, the quantitative method can be used to determine the model's response to the captured RE, that is, the magnitude of the change in RE.

2.4.2.1 Standard Metrics

Statistical methods are the most common way to quantify the difference between models. By measuring the statistical errors, such as Root Mean

Square Error (RMSE), Mean Absolute Error (MAE), Mean Bias Error (MBE), between the model predictions $\hat{y}_{t+\Delta t}$ and the ground truth $y_{t+\Delta t}$, the gap between the model predictions and actual values can be quantified, which can be expressed as equations below:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{2.33}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{2.34}$$

$$MBE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) \tag{2.35}$$

The performance of a model is often homogeneous across different statistical indicators, meaning that a model that performs well in one statistical indicator often performs well in other indicators. Moreover, assessing a model's performance using statistical indicators heavily depends on the sampling frequency, meteorological features, and model assumptions. Therefore, commonly used statistical indicators are unsuitable for measuring model quality. Chu et al. [117] recommended the use of Forecast Skill (FS) [26] as a quantitative indicator for evaluating and comparing the performance of different forecasting models in a review article. FS is defined as:

$$FS_{model} = 1 - \frac{RMSE_{model}}{RMSE_{baseline}} \times 100\% \tag{2.36}$$

Using the same parameters as complex models, baseline models can obtain quantified scores independent of the data set, sampling frequency, and model assumptions. It is worth noting that FS still strongly correlates with weather conditions [117]. In completely sunny or cloudy weather conditions, FS based on SPM is still a difficult-to-exceed quantitative indicator [18]. However, FS has certain limitations. As there is no RE impact in its indicator design, there is a limitation in evaluating the performance of models caused by REs due to cloud cover. Vallance al.'s work [118] showed that a mean prediction model without RE prediction capability could have better FS than a complex model with RE prediction capability.

2.4.2.2 Qualitative Metrics for Ramp Event

Qualitative comparison aims to evaluate the ability of a model to capture rapid changes in irradiance under cloud influence. For qualitative assessment of REs, there are currently two mainstream methods. The first method is the swing-door algorithm [119], which identifies slopes by defining a swing-door threshold ε as a tolerance value, as shown in the figure below: The value of ε determines the width of the swing door. Therefore, when ε is small, the model is sensitive to noise or small fluctuations, while when ε is large, the model skips small slopes. To cope with variations in solar irradiance over the seasons, Vallance et al. [118] proposed the sensitivity, τ, as an auxiliary parameter to help determine the value of e based on quantitative analysis, and the sensitivity τ is defined as the ratio of

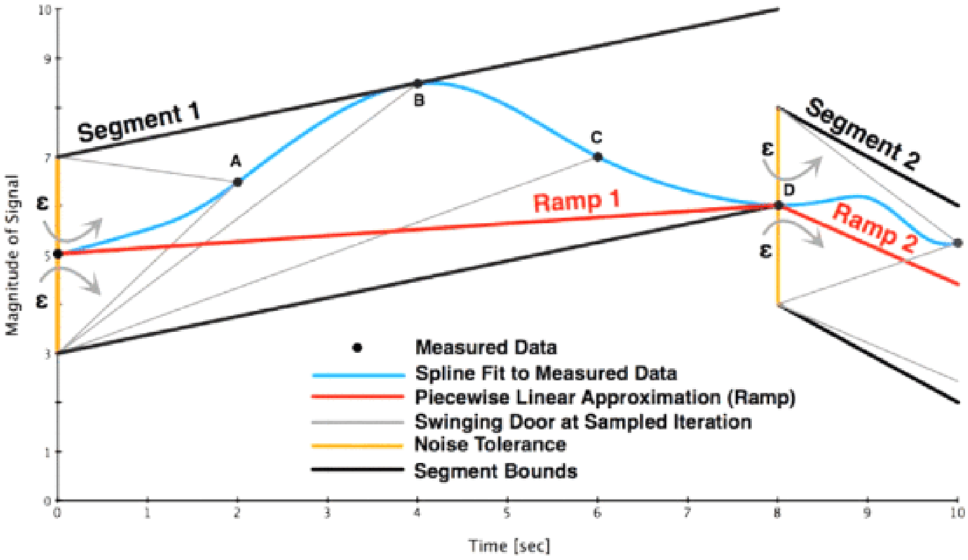


Figure 2.5: Demonstration of swinging door algorithm, Figure token form [119]

threshold ε under the maximum value of the clear-sky irradiance on the day, i.e.,

$$\varepsilon(d) = \tau \max_{day\ d} I_{clr} \tag{2.37}$$

By defining t, the width of the swing door is not affected by seasonal changes in irradiance. The authors recommend τ=18% as a choice that fits the measurement behaviour, but they also suggest that the choice of t should be defined according to the model design requirements. For example, in the DL-GSI-IHSF work, the authors used a more sensitive value of 5% as the choice of τ.

In Chu et al.'s work [10], another method for identifying RE was defined. RE was defined as solar irradiance changes with slopes exceeding a threshold ε within a specific time. If the model's predicted slope for a RE exceeded the threshold ε and was in the same direction as the RE, it was considered that the model successfully captured the RE. This definition method can use general anomaly detection as an evaluation criterion. Specifically, based on the actual positive and negative results and whether the model prediction is correct for a binary data set, the prediction results can be divided into one of the four categories in the confusion matrix. Specifically, true positive (TP) is the number of samples correctly predicted as positive, true negative (TN) is the number of samples correctly predicted as negative, false positive(FP) is the number of samples incorrectly predicted as positive, and false negative (FN) is the number of samples incorrectly predicted as negative. Based on the components of the confusion matrix, the evaluation metrics of the model can be defined: precision is the ratio of true positive results to actual positive results, and recall is the ratio of true positive results to all actual positive results. The F1 score can be calculated using the precision and recall harmonic mean. As the equation below shows:

$$Precision = \frac{TP}{TP + FP} \tag{2.38}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.39}$$

$$F_1\ score = 2 \times \frac{Precision \times Recall}{Precision + Racall} \tag{2.40}$$

In solar energy forecasting, due to the uncertainty of RE direction, the actual prediction classification can be divided into three categories, namely upward RE, nearly constant, and downward RE. Therefore, in this multi-classification task, the average value of accuracy for each class can be used to represent the overall accuracy of the model prediction, i.e.:

$$Balanced\ Precision = \frac{1}{N} \sum_{Class\ i}^{N} Precision_i \tag{2.41}$$

$$Balanced\ Recall = \frac{1}{N} \sum_{Class\ i}^{N} Recall_i \tag{2.42}$$

$$Balanced\ F_1\ score = \frac{1}{N} \sum_{Class\ i}^{N} F_1\ score_i \tag{2.43}$$

2.4.2.3 Quantitative Metrics for Ramp Event

Accurate RE prediction requires qualitative predictions of ramp events and equally accurate quantitative predictions. A straightforward way is to directly compare the statistical errors of RE value predictions, such as RMSE. In the work of [55], the authors found that the model that captured the most REs did not perform the best in quantitative analysis by comparing the qualitative and quantitative indicators of RE prediction among different model architectures. Therefore, there is still room for improvement in directly using general indicators for quantitative analysis of REs.

Another feasible and straightforward approach is to sort all predicted errors by the magnitude and report a higher percentile, such as the 95th percentile of absolute errors in [42]. A smaller number means a better match between the model and extreme error times. As RE must imply significant errors in solar prediction, this approach is representative of the ability to estimate the RE prediction of the model.

Therefore, [118] pointed out that using RMSE or other metrics to evaluate RE prediction performance has a potential drawback. These metrics do not incorporate the time component into the error measurement. For example, a model makes a correct prediction for a RE, but it is ahead or lagging in prediction time. In the RMSE metric, this would result in two large error values and lead to a poor model evaluation. However, this time-distorted prediction behaviour may not be considered negative. To overcome this problem, [reference] proposed two new metrics to define the model's prediction behaviour. The RE metric is based on the equivalence between the predicted RE and the actual RE's amplitude difference within a period. The ramp score can be expressed as:

$$ramp\ score = \frac{1}{t_{max} - tmin} \int_{t_min}^{t_{max}} \left| SD(T(t) - SD(R(t)) \right| dt \tag{2.44}$$

Ramp Score describes the integral based on the cumulative error in the ramp of the swing door between the tested and real values over a given time.

The Table **??** summarises the performance results reported by the models available for statistical purposes in this article. For comparison purposes, the 10-minute forecast performance with higher usage was chosen as the baseline forecast horizon in preference to models with different forecast horizons in the same job by default.

Table 2.3: Result reported in the DL-GSI-IHSF work.

Ref.	Statistical Error[1]	FS[1]	Qualitative RE score	Quantitative RE score	Weather[2]	Multi Site [2]
[32]	28% rRMSE @0min	-	-	-	✓	✓
[50]	140.5 W @1min	20.8% @1min	-	-	✓	-
[69]	4.51 kW @15min	26.22% @Sunny 16.11% @ Cloudy	-	-	✓	-
[33]	2.51 kW @15min	15.70% @15min	-	-	✓	-
[71]	2.47kW @15min	17.11% @15min	-	-	✓	-
[54]	˜$150W/m^2$	-	-	-	-	-
[58]	$150W/m^2$ @Sunny $103W/m^2$ @Cloudy $71W/m^2$ @Overcast	-	-	$19W/m^2$ @Sunny $55W/m^2$ @Cloudy $58W/m^2$ @Overcast	✓	-
[47]	40.15% nRMSE	28.89%	-	-	-	-
[43]	88.35 W/m^2	-	-	-	-	-

continued in next page...

Table 2.3 Result reported in the DL-GSI-IHSF work (Continued).

Ref.	Statistical Error	FS	Qualitative RE score	Quantitative RE score	Weather	Multi Site
[56]	31.9% nMAP @1Hour	-	-	-	-	✓
[57]	8.85% nRMSE	25.14%	-	-	✓	-
[59]	2.20 kW @0min	-	-	-	✓	-
[72]	80.14W/m^2	11.88%	-	-	✓	-
[73]	80.47 W/m^2 @15min	-	-	-	✓	-
[74]	23.5% nRMSE @5min	-	-	-	-	-
[75]	117 W/m^2	-	-	-	✓	-
[76]	8.7% nRMSE	-	-	-	-	-
[46]	-	20% (MSE)	-	-	-	-
[77]	22.5% nRMSE	17.70%	98.9% Capture	11.3% energy curtailment	✓	-
[55]	28.6w	26%	74.75% BP@ ConvLSTM-H	60.99 W	-	-

continued in next page...

Table 2.3 Result reported in the DL-GSI-IHSF work (Continued).

Ref.	Statistical Error	FS	Qualitative RE score	Quantitative RE score	Weather	Multi Site
[42]	-	20.40%	-	19.6 W/m^2/min RS, 0.34 TDM	-	-
[34]	˜32% rRMSE @0min	-	-	-	✓	-
[34]	-	˜18%	-	-	✓	-
[78]	23.47% nRMSE	22.56%	-	-	✓	-
[79]	131.85 W/m^2	-	-	-	-	-
[45]	˜12% nRMSE	˜19%	-	-	✓	-
[80]	63.98 W/m^2 @0min	-	-	-	✓	-
[82]	62.6 W/m^2	14.2	-	-	-	-
[83]	41.74 W/m^2	-	-	-	✓	-
[36]	71.3 W/m^2	21.45-	-	-	-	-
[53]	Multiple	Multiple	-	-	✓	✓

continued in next page. . .

Table 2.3 Result reported in the DL-GSI-IHSF work (Continued).

Ref.	Statistical Error	FS	Qualitative RE score	Quantitative RE score	Weather	Multi Site
[84]	18.27% rRMSE	-	-	-	✓	✓
[35]	109.1 W/m^2	24.00%	-	11.9 TDI	✓	-
[11]	101.1W/m^2	30.90%	-	-	✓	-
[48]	15.25% nRMSE	-	-	-	✓	-
[85]	44.292 W/m^2 MAE	-	-	-	-	-
[86]	80.02 W/m^2	-	-	-	-	-
[87]	75.18 W/m^2	-	-	-	-	-

[1] For comparison purposes, the prediction result for unstated results are all 10 minutes. Please see the original article for specific result.

[2] Model performance is calculated separately for weather conditions.

[3] Model performance differences are compared across multiple sites.

2.5 Current Achievements in DL-GSI-IHSF

The DL-GSI-IHSF field is a nascent and rapidly developing area. On the one hand, its predictive performance is continuously improving due to the fast proliferation of computer vision fieldwork. On the other hand, compared to rigorous traditional solar forecasting methods based on statistical, physical, or image models, deep learning algorithms lack clear interpretability and rigorous logical chains. Therefore, in the solar energy forecasting field, where robustness is required, the reliability of using deep learning methods in the experimental deployment stage is still controversial. As Reichstein et al. [120] suggested in applying deep learning in Earth science forecasting, deep learning models have great potential in data-driven Earth system science fields. When applying deep learning methods, traditional physical models should not be abandoned but should strongly complement existing physical knowledge. At the same time, in the field, since researchers are still exploring the transfer of suitable deep learning models to the solar energy forecasting model stage, some plug-and-play methods lack research on the validity, practicality, and generalisation of the transferred models. Therefore, in studying DL-GSI-IHSF models, their forecasting performance should be considered, and interpretability and rationality should also be given importance in the research process. Nevertheless, the work in the DL-GSI-IHSF field has achieved remarkable achievements.

In Sun et al.'s early work [32], the DL-GSI-IHSF deep model was first developed and validated. Their SUNSET (Stanford University Neural network for Solar Electricity Trend) network, based on a standard convolutional neural network, achieved 28% rRMSE performance in early nowcasting work, which predicts PV output using simultaneous sky images. They also explored the impact of different network depths and architectures on prediction performance and identified the optimal prediction architecture. In later work [69], they added PV output history to the model, enabling future 15-minute PV output forecasts with a forecast skill of 15.7%. Additionally, they proposed a model with architecture that uses the complementary nature of image and numerical inputs, which could be leveraged by concatenating the feature representations, leading to improved model performance [33]. In addition, they also investigated various fusion methods [71], including data-level, feature-level, and decision-level fusion. They found that early data-level fusion did not effectively extract features from multimodal data to improve performance. However, feature-level solid fusion or direct use of decision-level distribution prediction was beneficial for the joint expression of the existing models and achieved performance optimisation. Among them, the two-step model with late decision-level fusion achieved the best performance, with a forecast skill of 17.11%. Regarding data set-specific research, the team attempted to balance the data set by using pre-classified data sets [59] and data resampling methods [34] to address the issue of uneven distribution of different sky conditions in the model data set. The results showed that the nowcast model achieved about 6% performance improvement using the pre-classification method, while the resampling method improved performance by about 1.74%. However, the resampling method did not significantly improve the forecast method.

In the early work of Zhang et al. [50], the LSTM architecture was explicitly used as a framework for extracting time-related features from the spatial representation vectors of sequential images rather than implicitly embedding the prediction information in the spatial features. Their work found that using the LSTM model achieved a 20.8% 1-minute forecast skill, outperforming the model architectures that used CNN (12%) and MLP (7%). In addition, this work also practised the method of fusing ground observation values with image analysis results and proposed a hybrid loss weight method based on multitask learning. As mentioned above, subsequent work has validated that mixed inputs can form complementary information on target representation by their joint representation. Interestingly, the LSTM-Full model developed by Zhang et al. has an inverted relationship between prediction horizon and prediction performance. Generally speaking, the longer the prediction horizon, the lower the prediction performance of the baseline model (PM or SPM), making it easier to achieve higher prediction scores. However, in their model, the model achieved the best prediction score in the one-minute prediction, and the prediction performance decreased with the extension of the prediction horizon.

Zhao et al. [47] were the first to apply 3D-CNN architecture to the DL-GSI-IHSF field and achieved prediction of DNI 10 to 30 minutes in advance. Their results showed that in the 10-minute forecast, the model using MLP as the prediction head achieved an overall forecast skill of 17.06%. In cloudy weather, where the baseline model did not perform well, it achieved a forecast skill of 28.89%. In addition, their work also adopted a module that embeds ground observation information to assist in prediction. It used a pre-classification result based on cloud types as an additional learning objective for multitask learning. Unfortunately, the work did not compare the 3D-CNN with the general CNN architecture but compared it with manually classified predictions. Therefore, although the model has spatially extracted convolutional kernels and a theoretically superior framework, it has not been directly proven in work.

Guen and Thome's [74] and Kong et al. [55] in the same year incorporated ConvLSTM into the research. In Kong's work, the author comprehensively compared LSTM, CNN, CNN-LSTM, ConvLSTM, and PredNet architectures. The results showed that using a more complex spatiotemporal architecture did not beat the simple architecture of using numerical input concatenation to extract image feature representations with CNN. The study also found that using numerical input concatenation as an additional input was a generally effective method for improving quantification results for all models. However, the authors used BP in the qualitative analysis to compare the specific RE that the model did not achieve. The results showed that using a more reasonable ConvLSTM model could capture more Ramp Events, but ConvLSTM still lagged behind CNN architecture in estimating Ramp amplitude. In addition, the study also innovatively used the self-supervised learning PredNet architecture as the prediction end to predict irradiance by extracting potential feature representations of the sky images in the model's logical loop. Unfortunately, this method did not achieve better results in either qualitative or quantitative analysis.

In Guen and Thome's work [74], the author pointed out the unreasonable assumptions of previous spatiotemporal models in the computer vision field. He pointed out that previous spatiotemporal models assumed that the model had complete prior knowledge of physics, which was unreasonable [121] In his work, he used a simple physical constraint, namely a partial differential equation, as a physical module to improve the model's performance by imposing additional physical constraints on the ConvLSTM model in parallel. The results showed that the developed PhyDNet-Dual achieved 23.5% nRMSE under the multitask learning method. In addition, his work also found that multi-step prediction of RNN models could improve the model's performance.

The work of Paletta et al. [42] extensively compared the performance of previously used CNN, CNN-LSTM, 3D-CNN, and ConvLSTM architectures. They also innovatively used ramp score and TDI as RE and time distortion measures, respectively. The results showed that the ConvLSTM architecture remained the most promising model, with a prediction performance of up to 20.4% in a 10-minute forecast. However, the authors also pointed out that although all models used images as spatial representation inputs, they all behaved like a more intelligent SPM model, where the temporal component of the model prediction did not advance with the injection of spatial information. In other words, images did not fulfil their intended function of enabling the model to anticipate the arrival of Ramp Events through the dynamic display of spatial features.

In their subsequent work, Paletta et al. [35] reconstructed the deep inference framework and developed the ECLIPSE model, which encoded the image using a CNN architecture and extracted temporal features from the continuous image feature representation using a 3D-CNN architecture. Finally, the model predicted the following spatiotemporal representation by recursively combining the current state's spatiotemporal joint representation through the CGRN architecture. This innovative approach fundamentally changes the framework for spatiotemporal continuity prediction in solar energy forecasting. The model has three separate encoding modules for inferring different feature vector representations. Ultimately, the model achieved 10.2%, 23.6%, and 24.0% prediction performance in predicting 2-, 6-, and 10-minute lead times, respectively. In addition, the work demonstrated the roles and necessity of each module in the prediction through detailed ablation experiments and feature extraction. It is worth noting that the authors found that predicting relative irradiance changes improved the prediction performance compared to absolute irradiance changes. Additionally, similar to using observational data as an additional input group in previous studies, the authors found that adding irradiance as an extra channel to the image data to achieve data-level fusion significantly improved the model's prediction performance, increasing the final prediction performance to 18.5%, 26.1%, and 26.3%, respectively. In the subsequent expansion [11], the authors used a method to reconstruct the image by converting the polar coordinates of the sun and cloud layers in fisheye sky images to Cartesian coordinates. This approach fundamentally reconstructed the data and achieved unprecedented performance gains. Finally, the model achieved excellent forecast skills of 23.1%, 32.8%, and 30.9% in predicting 2-, 6-, and 10-minute forecast horizons, respectively.

2.6 Scientific Challenges based on Literature Review

In this section, based on the knowledge gap summarised above, the main scientific questions addressed in this thesis are presented.

2.6.1 Fusion of Visual Modalities with Other Modalities in DL-GSI-IHSF

In the field of DL-GSI-IHSF, the black-box nature and high complexity of deep learning models make them challenging to interpret, limiting their development. Early work discovered that integrating image data with historical solar energy records, as a multimodal fusion method, could effectively improve model performance. This approach is similar to solar energy forecasting based on image analysis models, which involves using extracted features from images to adjust statistical models [33]. However, the recent work of Paletta et al. [42] shows that mainstream convolutional and recurrent deep models still perform like intelligent persistence models in terms of prediction, assuming that atmospheric conditions remain constant. In other words, while reducing the model prediction error, the complex spatial information did not affect the model's perception and prediction of spatial information to make it respond to RE. The current mainstream Late feature fusion-based Convolutional Neural Network model does not exhibit sensitivity to different image inputs. This phenomenon in solar energy forecasting models highlights existing issues in integrating spatial information with other modalities, namely:

- In multimodal fusion models for intra-hour solar energy forecasting, how can we quantify and determine each modality's role in the final forecast?
- How can we strengthen the modality fusion aspect of the current architecture to truly utilise the spatial information in sky images for solar energy forecasting?

2.6.2 Transfer Learning for DL-GSI-IHSF Models under Different Climates

The geographical location of observation sites and climatic conditions strongly influence solar irradiance, as indicated by historical statistics [122]. Therefore, solar irradiance prediction models usually rely on local datasets. However, this poses a challenge for DL-GSI-IHSF models since their performance is limited by the observatory's spatial location and the local dataset's unique characteristics. Local datasets can directly impede the generalizability of DL-GSI-IHSF models with distinctive features. Moreover, acquiring new datasets can be prohibitively expensive in terms of time, money, and personnel resources, further hindering the development and deployment of these models for site-specific testing.

In deep learning, one potential solution for datasets with limited data is to transfer the knowledge, i.e., the weights within the model architecture, from a model trained on a larger dataset to a model with limited data, thereby improving the latter's generalizability performance. This migration method can reduce the data collection and training time when deploying new models, effectively improving model training efficiency. However, similar work has not yet been done in the DL-GSI-IHSF domain. Given this background, we pose the following questions:

- Can the transfer learning approach based on weight transfer enable the model to transfer the IHSF prediction features learned under one set of climate and geographical conditions to a completely different set?
- If transfer learning can be used, how can it impact model training?

2.6.3 Improving the Model Calculating Efficiency by Simplify Model Architecture

The development of the DL-GSI-IHSF model primarily involves the transfer of prior models from the computer vision domain. However, the development and validation of computer vision models are often carried out for highly complex tasks. For instance, ImageNet [123], the most well-known dataset in computer vision, categorises 1.2 million images into 1000 classes based on content. In computer vision work, models are often developed in various depths to cope with tasks of different complexity levels. Deeper models imply greater computational demands and higher performance within a reasonable range. However, in the DL-GSI-IHSF work, the phenomenon is quite the opposite. It has been found that using deeper or better-performing models in the computer vision domain to analyse sky images does not effectively improve the model's predictive performance; instead, deeper models may lead to a decline in performance [77]. This phenomenon is similar to the model overfitting issue in computer vision when using deep networks [99]. In the ResNet study [99], the authors discovered that the 1202-layer model performed slightly worse on the CIFAR-10 dataset than the 32-layer model, and the cause of this was speculated to be overfitting. However, in the context of DL-GSI-IHSF, only the 50-layer model displayed evidence of overfitting [77], which is significantly less than in deep vision studies. It is postulated that this discrepancy may be attributed to the complexity of the prediction task, for which the dataset of DL-GSI-IHSF is likely to be less intricate than generic depth models. Therefore, we raise the following questions:

- Is even the simplest model in the computer vision domain still too complex for the DL-GSI-IHSF work? In other words, does the DL-GSI-IHSF require a complex and deep visual model to analyse images?
- Is there still room for simplification in the current DL-GSI-IHSF model framework?
- How can we ensure that the computational requirements are met without overfitting?

Chapter 3

General Methodology and Dataset

Chapter Abstract

The methodology and datasets utilised throughout the subsequent three chapters were discussed in this chapter of this thesis. The methodology section presents the DL-GSI-IHSF model framework grounded on the data-model-analysis development process. The dataset section offered the online public dataset retrieved from the Folsom and Nottingham datasets collected from the local observatory archives. Both are applied in the subsequent chapters.

Contents

3.1 Research Framework

This section primarily introduced the overall research framework used in this thesis, as shown in Figure 3.1. Each column in the figure represents the three main research chapters and one experimental data collection part of this paper. Each row corresponds to the primary steps of developing deep learning models, followed by each chapter, namely data preparation, model training, and model evaluation. Firstly, the data preparation part mainly involves preparing the dataset required for model training, including data collection, multi-source data alignment, data quality control, data downsampling, data normalisation, and dataset partitioning. The parts using publicly available datasets or data accessible through public calculation methods are indicated by their data sources. The sections using data collected from local testbeds will be introduced later in this chapter. Secondly, the model training mainly involves fitting the developed deep learning model to the designed dataset. The overall framework, algorithm, and other essential information of the model will be introduced in detail in each research chapter. The specific framework of the model, including the number of layers, details, resolution of each layer, and the number of channels, will be disclosed in Appendix B. Thirdly, the model evaluation part primarily focuses on the prediction results of the trained model on an independent test set and comprehensively assesses the model's predictive performance according to the evaluation criteria in each chapter.

This thesis starts by reviewing the existing DL-GSI-IHSF work and revealing the model architecture issues concerning multi-modal vector representation fusion that existing models have overlooked. In the first research chapter, a model was developed and debugged based on a relatively complete online public dataset - the Folsom dataset from California, USA [39].

This work mainly focuses on improving multi-modal representations' fusion and interaction processes. Next, in the second research chapter, we address the poor generalisation of the prediction model in different environments by attempting to transplant the prior weights obtained from training the model on the Folsom dataset in the first part to the local dataset collected in Nottingham, UK. Simultaneously, we verify the role of transfer learning in tackling the problem of insufficient data during model generalisation. Finally, in the third research chapter, we develop a model using local data from Nottingham. We attempt to streamline the model architecture and optimise the algorithm to achieve higher computational efficiency while improving model performance.

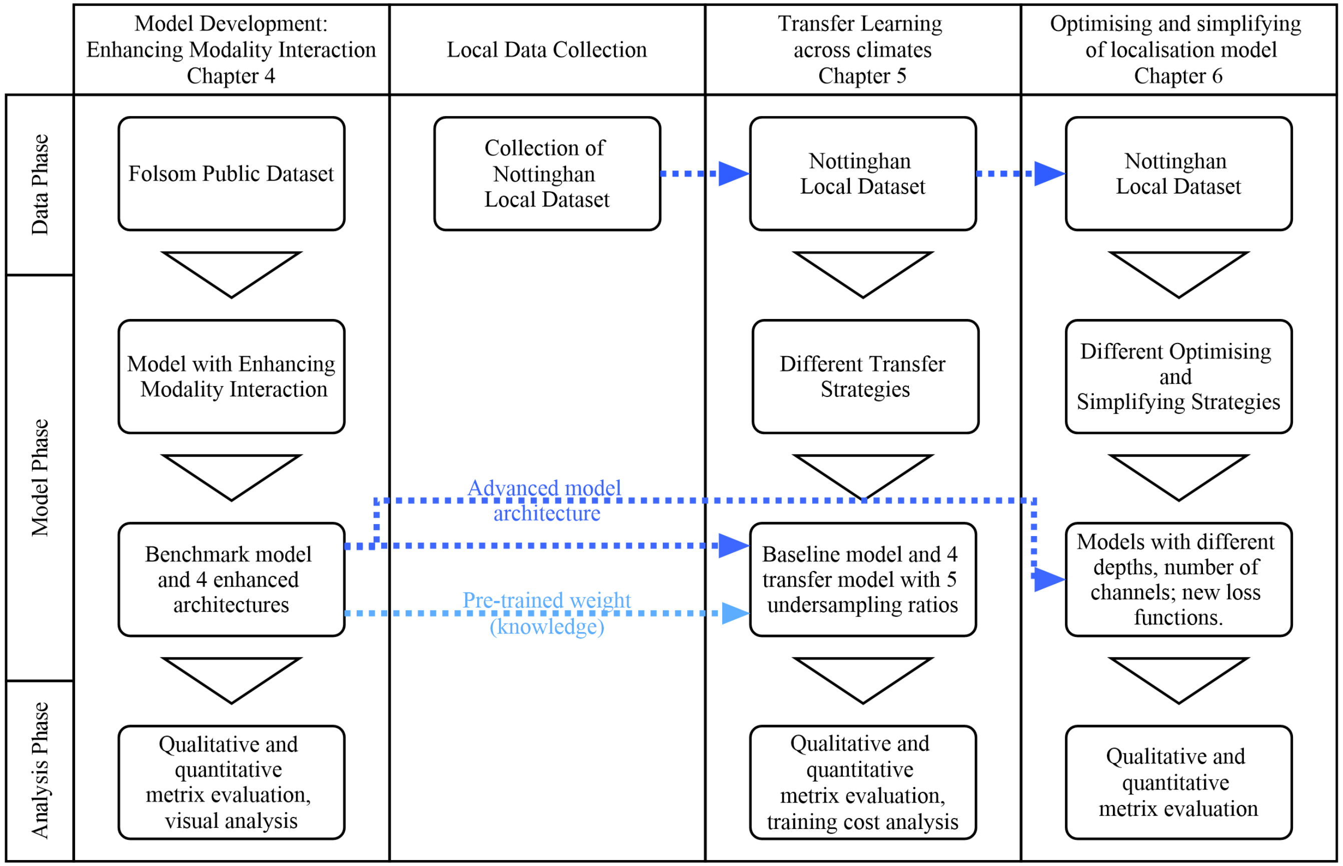

Model Development:
Enhancing Modality Interaction
Chapter 4
Local Data Collection
Transfer Learning
across climates
Chapter 5
Optimising and simplifying
of localisation model
Chapter 6
Data Phase
Model Phase
Analysis Phase
Folsom Public Dataset
Collection of Nottinghan Local Dataset
Nottinghan Local Dataset
Nottinghan Local Dataset
Model with Enhancing Modality Interaction
Different Transfer Strategies
Different Optimising and Simplifying Strategies
Advanced model architecture
Benchmark model and 4 enhanced architectures
Baseline model and 4 transfer model with 5 undersampling ratios
Models with different depths, number of channels; new loss functions.
Pre-trained weight (knowledge)
Qualitative and quantitative metrix evaluation, visual analysis
Qualitative and quantitative metrix evaluation, training cost analysis
Qualitative and quantitative metrix evaluation

Figure 3.1: The research framework of this thesis, with each column representing a research topic and each row representing three different phases of the topic.

3.1.1 Development of DL-GSI-IHSF with Enhancing Modality Interaction using Attention Mechanism and Gate Mechanism

In the first research part of this paper, we reveal a potential flaw in the current DL-GSI-IHSF model. Specifically, as reviewed in Sections 1 and 2, the current DL-GSI-IHSF models, although continuously incorporating more advanced and complex spatiotemporal reasoning-based computer vision models, do not show significant influence from temporal image information in the final predictions and still perform like a sophisticated SPM model. It is believed that the widely-used method of concatenating multimodal feature representations is overly simplistic, lacking specific feature representation alignment and feature interaction enhancement modules. The model has not gained the ability to perform cross-modal reasoning. Therefore, inspired by other multi-modal domains, the proven attention and gating mechanisms were used to enhance feature representation and interaction in multi-modal joint features. Based on these two patterns, we develop four different architectures of interaction-enhanced models and compare their performance with the mainstream CNN architecture based on modal vector concatenation. In the model trained using the Folsom dataset in the United States, it was found that the ViT-E model, which employs early feature-level fusion, can search for features across modalities through the global attention mechanism, achieving the most balanced performance.

3.1.2 Transfer Learning across Climates: Multi-climate Training based on Model Transfer

The second part of this study identified a dilemma in promoting and validating the DL-GSI-IHSF model. Regions with different climates require data collection to train new models, which is costly and challenging. Additionally, the model's application is limited by its dependence on climate and geography, necessitating additional training to predict outcomes in new regions. To address this issue, transfer learning was employed as a compelling solution. Initially, observational sites are established in the Nottingham region of the United Kingdom, and six months of data are collected to serve as the transfer target. The ViT-E model trained on the Folsom dataset is used as a pre-trained platform, and pre-trained knowledge is extracted. The Folsom dataset is particularly suitable for short-term solar energy forecasting among arid and sunny California regions and undergoes rigorous data quality control. During the transfer process, two different methods are extant: one directly deploying the Nottingham dataset on the already trained pre-trained model for further training, the other using some or all of the pre-trained weight without additional training fine-tuning the model's prediction head. Additionally, we modelled another scenario in which deep models are trained and deployed in datasets with scarce data utilising source domain knowledge migration to overcome the training bottlenecks caused by limited data. Lastly, we comprehensively compare the performance and training costs of transfer learning with the brand-new models trained from scratch to evaluate the application of transfer learning in DL-GSI-IHSF models.

3.1.3 Optimising and Simplifying the Localisation Model

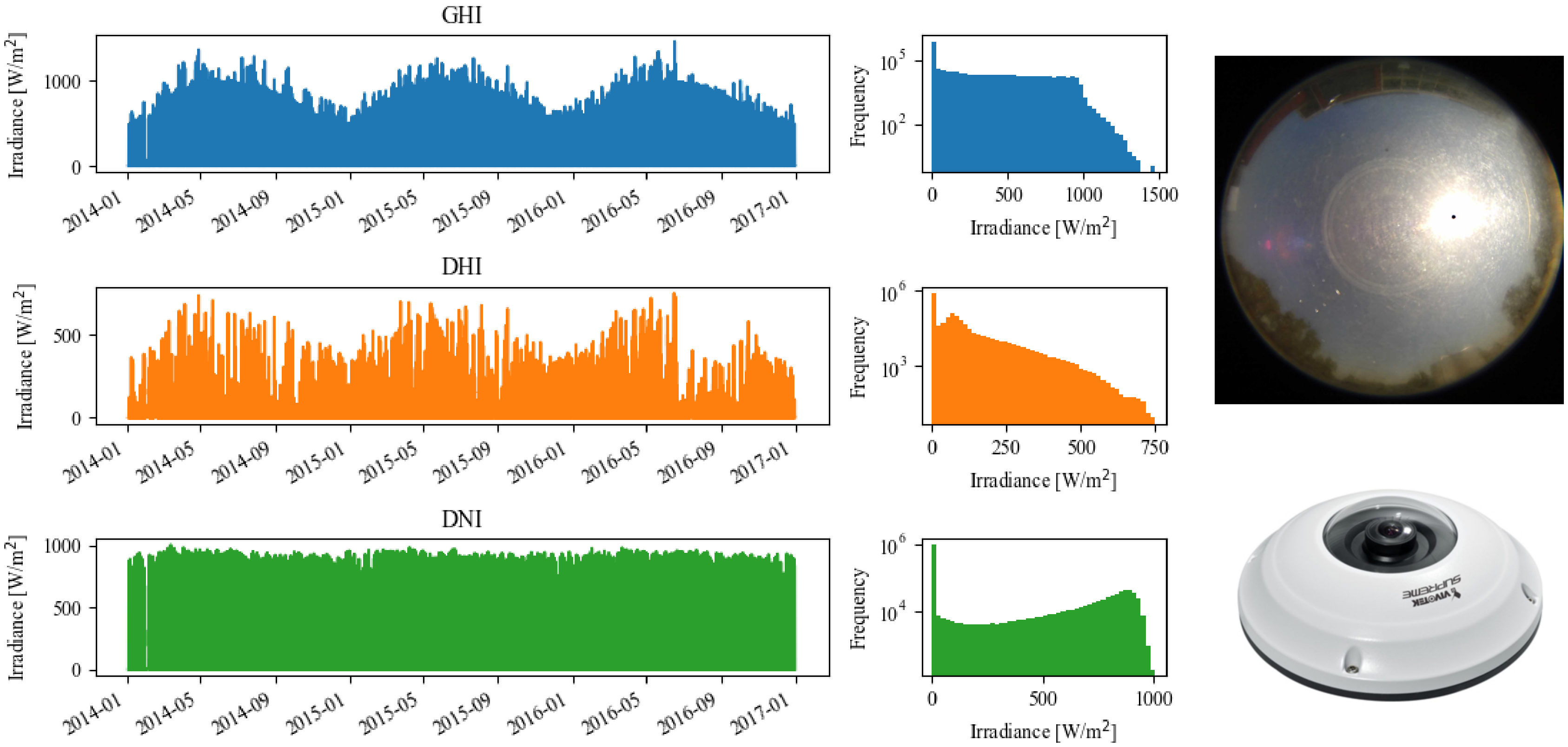
In the third research part of this paper, another phenomenon was revealed when transferring computer vision networks to the solar energy forecasting domain. The network architecture's performance is inconsistent with the original computer vision and solar energy forecasting tasks. Specifically, when transferring networks with significant advantages in computer vision datasets to the solar energy forecasting domain, their architecture does not demonstrate a significant performance gap compared to essential external networks. Simultaneously, in solar energy forecasting tasks, using more complex visual networks does not exhibit higher spatial and temporal resolution capabilities in model performance. Therefore, it was speculated that capturing spatial features in sky images does not require highly complex deep networks. Based on this background, we attempt to simplify the ViT network architecture transferred from the computer vision domain and optimise it through hyperparameters to improve the model's prediction performance while streamlining the model and enhancing computational efficiency. Finally, using a unified comparative modelling system, we systematically compare differences in model performance and similarities in computational efficiency, and explore the best local model structure.

3.2 Data Collection Platform

As introduced in the first chapter, for IHSF, sky images provide valuable exogenous input with high spatiotemporal resolution. In our work, we utilised two datasets. One is a dataset collected explicitly for solar forecasting, released by the University of California, San Diego (UCSD) [39] This dataset covers data from 2014 to 2016 with a 1-minute interval, including sky images, solar irradiance data, and meteorological data. The other dataset is a local Nottingham dataset, collected based on the solar observation equipment at the University of Nottingham. The data collection started in November 2021 and includes sky images, solar irradiance data, meteorological data, solar spectral data, and PV output data.

3.2.1 Folsom Dataset

The Folsom dataset is released by UCSD, specifically designed and collected to accelerate the development of solar forecasting [39]. The dataset comprises four parts: 1. Multi-year quality-controlled solar irradiance information and meteorological data; 2. High-resolution sky images were taken at the exact location and time; 3. Satellite images of the exact location and time; 4. NWP data of the exact location and time. The observation station is located at 38.642N, 121.148W, Folsom, CA, which belongs to the Csa climate in the Koppen Climate classification (C = Warm temperate, s f = dry summer, a = hot summer). In this paper, we used irradiance information, weather data, and high-resolution sky images from the dataset as the raw data for model development. According to the earliest traceable work [9], sky images were taken by a Vivotek FE8171V fisheye network camera with a 3.1-megapixel complementary metal oxide semiconductor (CMOS) sensor. The time resolution is 1 minute, the spatial resolution is 1536x1536, and the camera faces 15 degrees west of north. The dataset includes three years of data (from January 2014 to December 2016). Figure 3.2 (b) shows the sky camera and the images generated. The RSR-2 device made solar irradiance observations from Augustyn, Inc. The device has two Licor-200SZ pyranometers for measuring GHI and DHI, and DNI is calculated based on the results and the solar angle simultaneously. It is worth noting that the typical error of this instrument is about 5% lower than that of the highest precision class Precision Spectral Pyranometer and slightly lower than the accuracy range of class 2 Pyranometer in the IS0 standard. Meteorological measurements, such as temperature and wind speed, were collected by the Vaisala WXT520 Micro Weather Station. Meteorological and irradiance data were eventually recorded and synchronised to the online database by the Campbell Scientific CR1000 data logger. All instruments are connected to a local Network Time Protocol (NTP) server to ensure time synchronisation. Figure3.2 (a) shows the distribution and density relationship of solar irradiance data from 2014 to 2016.



(a) Solar irradiance distribution in the Folsom dataset.

(b) A sample of clear sky images from the Folsom dataset and the sky camera Vivotek FE8171V.

Figure 3.2: Data distribution and sample image presentation in the Folsom dataset.

3.2.2 Nottingham Dataset

The Nottingham dataset includes solar irradiance and meteorological data recorded by the data observation station on the roof of The Energy Technologies Building at the University of Nottingham Jubilee campus. Since March 2019, solar irradiance and meteorological data have been collected at an observation frequency of 1-minute average. Since November 2021, sky images have been provided at an observation frequency of 15s/sample. Since April 2022, full sky horizontal, tilted, and direct spectral information has been collected at a resolution of 15s/sample. Moreover, since August 2022, the specific PV panel output data have been collected at a time resolution of 1 minute/PV curve. The dataset is obtained from the observation point of 52.952N, 1.184W, which belongs to the Cfb climate in the Koppen Climate classification (C = Warm temperate, f = Fully humid, b = Warm summer). The layout of the observation point is illustrated in Figure 3.3. Sky images are acquired by the Mobitix Q26 fisheye network camera, which is equipped with a 6.0-megapixel 1/1.8" CMOS sensor, as indicated in Figure 3.4 (b). In order to minimise the impact of camera factors on image quality, the automatic exposure adjustment, noise reduction, and contrast adjustment functions of the camera are deactivated, and the camera faces west, reducing the resolution to 1028x1028 images. Solar irradiance is monitored by the Razon+ Sun Tracker, which has a built-in PR1 Pyranometer for measuring DHI and a PH1 Pyrheliometer for measuring DNI, and GHI values are calculated based on the results. The instrument records one minute of irradiance data and provides the average value. PR1 and PH1 belong to the second class Pyranometer in the ISO accuracy standard, and the verified error and spectral instrument measurement difference is less than 0.2%. The Maplins N23DQ Weather Station collects meteorological data, including temperature, humidity, air pressure, wind speed, and wind direction. All data are transmitted to a unified router and then sent to the data server for archiving. Furthermore, a Raspberry Pi with GPS signal timing is added to the router as an NTP server because the error of the internet-based NTP server may exceed 500 milliseconds, and a Pulse-per-Second chip is used as an auxiliary calibration signal. The local NTP server can reduce the error to less than 1 microsecond and broadcast to all instruments every 6 hours. The irradiance data for a whole year, from October 2021 to September 2022, is depicted in Figure 3.4 (a).

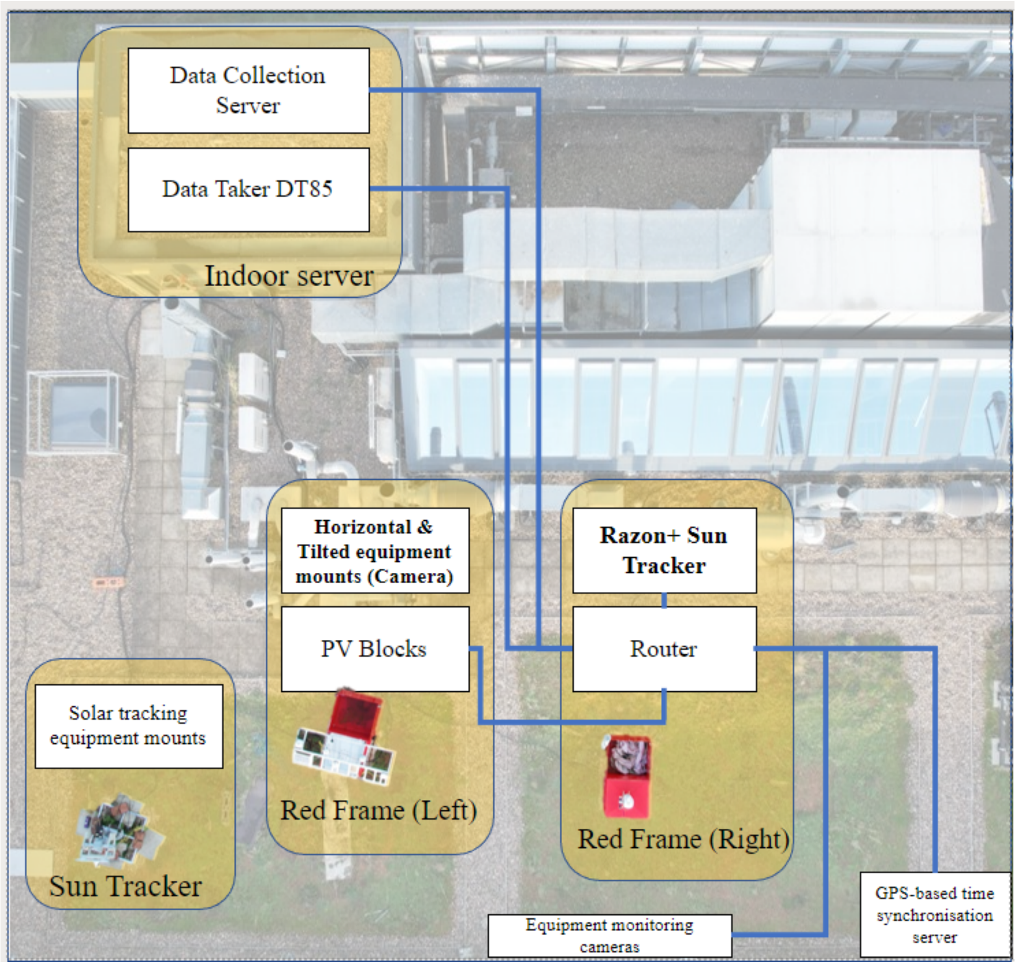
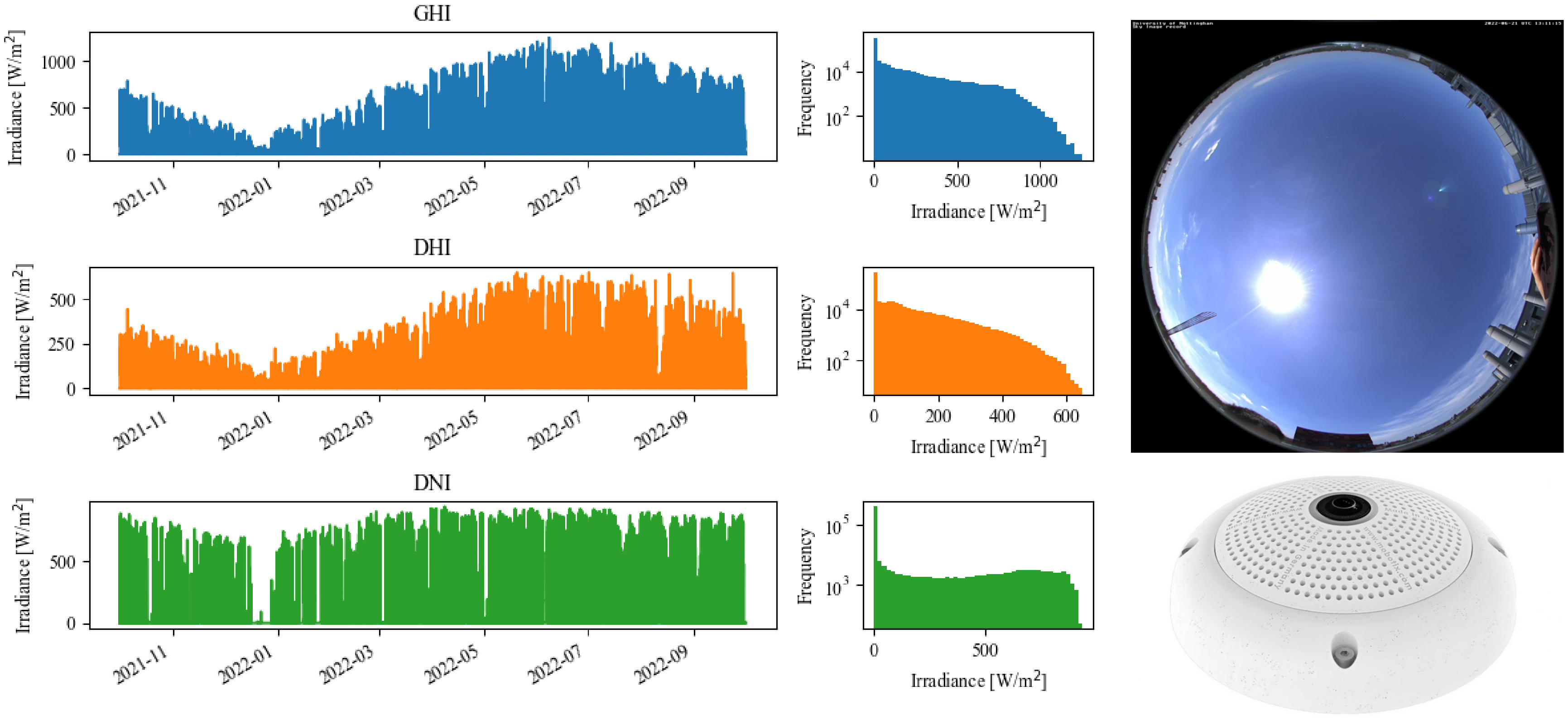

Data Collection Server
Data Taker DT85
Indoor server
Horizontal & Tilted equipment mounts (Camera)
PV Blocks
Red Frame (Left)
Razon+ Sun Tracker
Router
Red Frame (Right)
Solar tracking equipment mounts
Sun Tracker
Equipment monitoring cameras
GPS-based time synchronisation server

Figure 3.3: Distribution Map of Overhead Instruments at Nottingham Site.



(a) Solar irradiance distribution in the Nottingham dataset

(b) A sample of clear sky images from the Nottingham dataset and the sky camera Mobotix Q25.

Figure 3.4: Data distribution and sample image presentation in the Nottingham dataset.

3.3 Outline of Following Chapters

This paper focused on expanding and improving existing DL-GSI-IHSF models by examining their forecasting performance, evaluation metrics, interpretability, scalability, practicality, and computational efficiency and making necessary modifications.

In Chapter 4, an issue with the existing framework was identified: the lack of emphasis on multimodal fusion. This issue was addressed by deploying two specific mechanisms to optimise this module on the Folsom dataset and compare the effects of five different network architectures on model performance and sensitivity to image information. Based on the solar energy forecasting performance, the best-performing ViT-E model was ultimately chosen for further optimisation and investigation.

In Chapter 5, transfer learning was used to adapt the knowledge acquired from the Folsom dataset to the Nottingham dataset. First, using the same approach, we train a new baseline model on six months of Nottingham data. Then, the parameters learned in the Folsom model were gradually transferred to the Nottingham dataset, adapting it to the local climate through training or fine-tuning. The models obtained through transfer learning and training from scratch were compared regarding model performance, training cost, and data acquisition cost. Finally, a simulation experiment was conducted, training a model on two weeks of real-world continuous raw data to compare the practical effects of transfer learning.

In Chapter 6, the ViT-E model was further streamlined and optimisedl through model architecture search and hyperparameter search techniques. The model was refined and modified by adjusting the model architecture, loss function, and optimiser. Through model architecture search, we determine the most computationally efficient model architecture that ensures model accuracy. By adjusting the loss function and optimiser, we further reveal the effective parameter structure used in training Transformer-based models. Ultimately, the findings were validated by comparing the performance and computational efficiency of the optimised and unoptimised models.

Chapter 4

Enhancing Modality Interaction with Attention and Gate Mechanism

Chapter Abstract

The limitations of directly concatenating two modality features were emphasised in this chapter in the current state-of-the-art models. As an alternative, we suggest enhancing the process of modality interaction. We introduce gate and attention mechanisms to strengthen the correlation between the two modality vectors in the interaction stage. We use Forecast Skill and Balance Precision to qualitatively and quantitatively compare the performance of the models, respectively, to provide a unified perspective. It was found that the ViT-E model using a global attention mechanism gives a balanced approach to obtaining relatively optimal quantitative and qualitative results.

Contents

4.1 Introduction

As solar power generation grows, its inherent variability presents the grid with issues related to reserve costs, dispatchability and ancillary generation, and grid reliability in general [124]. Accurate solar irradiance forecasting at different time scales is a prerequisite for effective solar energy utilisation and a critical step in the grid integration and management of solar farms [125, 126]. Reliable solar forecasting tools improve the economics of PV power generation and reduce the negative impact of PV uncertainty on grid stability [127].

Changes in cloud cover are the leading cause of rapid changes in solar irradiance. Since the prediction models based on statistical numerical regression used in very short-term forecast models do not include information on fast-moving clouds, alternative or additional data inputs that account for these rapidly changing meteorological phenomena are required if accuracy at this time scale is to be improved.

Ground-based sky imagery represents one such exogenous data source. It plays a crucial role in solar energy forecasting due to its ability to provide information on cloud distribution and motion. Solar irradiation models informed by cloud motion data offer the potential to deliver accurate forecasts of very short-term solar irradiation and thus provide valuable supporting information for grid management and informing the market around power supply and demand [128].

Currently, sky images taken by fish-eye cameras contain rich spatiotemporal features. Thus, the academic community widely accepts them as exogenous data for intra-hourly level sky modelling [21, 26, 129]. The main methods for predicting solar irradiance based on sky images can be divided into two categories. The first is a sky modelling approach based on classical image analysis. Various methods are used to identify cloud pixels in sky images and determine their spatial and temporal features, including the red-blue ratio or difference, 3D cross-correlation, image feature correlation, optical flow, and ray tracing. These methods are used to forecast the impact of clouds on solar irradiance by combining cloud position estimates with estimates of cloud transmittance, which can be determined through fixed, density-based, or height-based approaches. However, these modelling approaches have limitations due to the complex physical properties of clouds, such as their motion and transparency, which cannot be accurately accounted for using current methods. Therefore, the accuracy of future irradiance forecasts using this approach remains limited. This approach is based on decision-level fusion, i.e. solar irradiation forecasts and ramp forecasts are made independently of each other and only influence each other when combined in the final stage, as shown in Figure 4.1 (a).

The second approach uses deep learning methods [42, 77, 50, 47, 54, 130, 131, 132, 133, 134]. This usually employs a combination of convolutional neuron networks (CNN) [96] and recurrent neural networks [107] (RNN) based methods to predict solar irradiance information for future periods. The widely used CNN-based computer vision models, such as ResNet [99] and VGGNet [98], can extract feature information from a dataset containing many sky images using deep convolutional neuron networks to obtain spatial dimensional perception capability. After extracting the spatial information of the images, various methods can be used to obtain time-series-based information. These include pre-processing by stacking a time series of images [77], convolution processes using 3D-CNN with an extra-temporal dimension [47], convolution-based long and short-term memory (LSTM) network [42], convolution followed by feature-based LSTM networks [50, 133], directly using regression algorithms for continuous results [77, 47], or combine feature engineering techniques with LSTM techniques [131]. Combining the architecture of two networks and fitting them using a large amount of data can obtain a network model with both spatial and temporal feature perception. This stitching model can map the relationship between specific features in continuous input image data and forecast targets. This model type has been applied to short-term forecast intervals for different resolutions. In contrast to models based on image analysis, current deep learning models can be mainly categorised as late feature fusion models, where the image and numerical values respectively abstract features as a high-dimensional vector in their respective models and concatenate the two vectors at the end of their respective operations, as shown in Figure 4.1(b). The tandem high-dimensional vector can be considered a common feature extract based on the two modalities. The final prediction is based on extracting available information from that vector.

While deep learning networks have been shown to deliver predictions with greater accuracy than those based on feature engineering in ground-based sky picture solar prediction, researchers cannot assess the relationships between variables that affect performance due to its black box nature. For example, using sky images as exogenous data to aid solar prediction has improved model performance at time scales ranging from 2 minutes ahead [35] to 1-hour ahead [57]. It was evident that the images play a different role at these two different time scales, but the features it identifies are not understood.

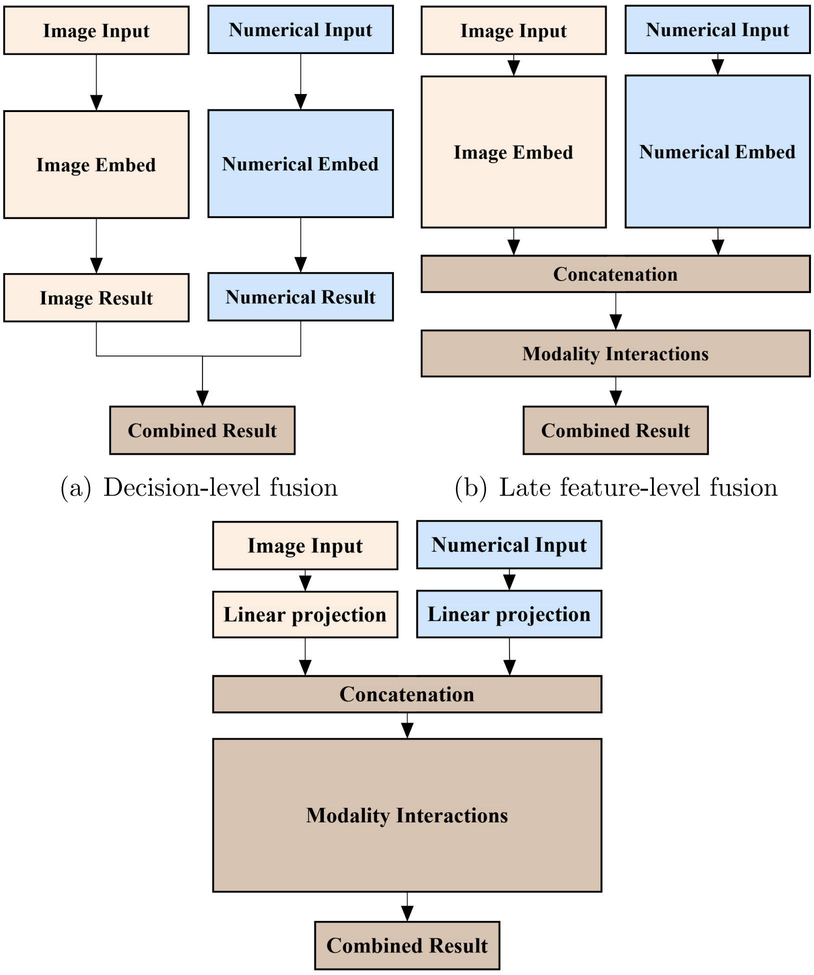
This paper argues that solar irradiance forecasting using ground-based images from which numerical features are extracted that describe the solar field can be categorised as a general multimodal learning domain rather than a pure computer vision domain. Multimodal learning is the combination of different data types in deep learning to predict a target variable through classification or regression tasks. For example, a medical diagnosis can be made by combining CT images and clinical data (classification). A speaker's emotion can be predicted by analysing video images and audio data (regression). As discussed in Chapter 2, the original motivation for using sky images to assist solar energy prediction was to fill the gap of short-term spatial information missing in statistical algorithms by utilising the spatial information contained in the images.

As shown in Figure 4.1, for the broad field of image-informed multimodal learning, besides the two architectures mentioned above, i.e. decision-level and late feature-level fusion of image information, the fusion methods also include data-level fusion (not shown in Figure) and early feature-level fusion. Early feature-level fusion and late feature-level fusion extract feature fusion within the model, with early fusion focusing on modal interactions and late fusion focusing on feature extraction [135]. In deep learning models used for solar forecasting, two architectures are currently applied: late feature-level fusion [42, 33, 50, 73] and decision-level fusion [71, 77]. In the work of Paletta et al. [42], the use of numerical data as additional inputs fused with a computer vision model improved the 2-minute forecast skill (FS), which rose from -3.4% to 12.9% and the 10-minute FS, which rose from 18.8% to 23.9%.

However, the literature suggests that the interest of researchers is currently focused on the image feature side to improve overall forecasting power through a more robust image network. This approach neglects both the numerical component's role in the model and whether it interacts effectively with the image component. For example, the numerical regression-based fully connected Multi-Layer neural network module (MLP) has been added to forecasting models by default due to the use of PV logarithms as an additional numerical input in the work of Sun et. al. [33] and significantly improved the performance of the model.

Another potential area of research responds to the fact that the image-numerical bimodal model currently in use is not modal interaction friendly. The prevailing image feature framework is the convolutional neuron network (CNN), where specific features of an image are extracted by sliding convolutional modules through the image and gradually constructing a high-dimensional vector representation of the image by multi-layer superposition. This architecture means it is impossible to extract features present in the 3D image and use these directly with complementary data held in a 1D array. Therefore, if data features of different dimensions are extracted simultaneously by convolutional computation, i.e. early feature-level fusion, this must be done by projecting the 1D data to a higher dimension and concatenating it with another, a process that may lead to distortion of the low-dimensional data. Venugopal et al. [71] compared CNN networks against PV output-based regression predictions with different fusion methods. Their results showed that late feature- and decision-level fusion achieved better prediction performance. However, data- and early feature-level fusion failed to effectively interact with information across modalities to achieve results beyond the baseline.

Multimodal learning adopts a unique feature extraction approach. Its transformer architecture enables data from different modalities to be fed into the encoder in parallel to achieve early feature-level fusion, as shown in Figure 4.1(c). It can effectively address the challenges of inherent data misalignment arising from the variable sampling rate and establish cross-modal element correlations of each modality's sequence [135]. Thus, the transformer-based model is widely used in the multimodal learning fields of image-language interpretation [136], image-sentiment recognition [137],



(a) Decision-level fusion

(b) Late feature-level fusion

(c) Early feature-level fusion

Figure 4.1: Schematic diagram of the model architecture for the different fusion stages. The higher box represents the main inference module.

the joint expression of video-audio-text [138, 139], etc. These applications share commonality with the mixed-mode data feeds available for irradiation forecasting. The original contributions of this chapter are:

1. To present two new approaches for picture-numerical bimodal model interaction. Namely, an improvement of the late feature-level fusion method using a gate architecture and a new early feature-level fusion method based on the Transformer architecture.

2. To assess the performance of the model 2-, 6-, and 10-minute forecasting horizons by scoring its quantitative statistical performance using the Smart Persistence Model (SPM)-based FS metric and the qualitative performance of the model using the Ramp Events (RE)-based Balanced Precision (BP) metric.

3. To show contradictions in the quantitative and qualitative performance of late feature-level fusion models regarding single image and numerical fusion. In particular, the widely used CNN model based on late feature-level fusion obtained higher FS, resulting in lower BP. We speculate on and attempt to demonstrate a link between this and the poor sensitivity of its architecture to images.

4. To demonstrate that for the end-to-end single picture-numerical bimodal model, the central variability of the model, both architecturally and algorithmically, was most pronounced for the 2 minutes ahead forecast. This variability fades with longer forecasting horizons. At 10 minutes ahead of forecast, the validity of the image information is extremely low, and all models have degenerated into a mean reversion model that relies primarily on irradiance and clear sky irradiance.

The remainder of the paper is structured as follows: Section 4.2 presents the overall experimental approach, including Data pre-processing, model architecture, and evaluation methods; Section 4.3 presents results that show quantitative and qualitative evaluation results for all models and discusses the results; Section 4.4 presents the results-based comparison of model performance with a critical discussion of existing architectures; and Section 4.3 presents our conclusions and recommendations for future work.

4.2 Methodology

Figure 4.2 illustrates the methodology adopted in this study—the approach to building a deep learning solar forecasting model based on image-numerical fusion comprised of three stages. The first was a pre-processing data stage, which aligned, filtered, sampled, and grouped the raw data into a format suitable for training a deep learning model. The second was a training stage, where the training dataset was fed into the model, and the weights within the model were fixed by backpropagation. Following this, the model was evaluated on a validation set to assess the performance trained in the training dataset. Continuous iteration saves the model that achieves the optimal result on the validation set, i.e. the model with the most negligible loss, from ending the training process. The final stage involved using a test dataset to obtain a forecast for comparison with ground truth data to quantify the final performance of the different models studied in this paper.

Clear sky index (CSI), i.e. the solar irradiance as a percentage of the clear sky irradiance, was chosen as the target for forecasts rather than the GHI, reflecting consensus within the solar forecasting community around its ability to improve the accuracy of solar irradiance forecasts made using numerical regression algorithms [49], including those that involve image-numerical multimodality approaches. Additionally, using CSI as a forecast target has a beneficial inductive bias compared to the direct irradiance forecast, i.e., the model assumes a priori knowledge of the clear sky background. Forecasts generate an atmospheric transmission rate (or attenuation rate) based on the transparent sky background, which is also consistent with traditional image analysis methods when harnessed for irradiance forecasting.

The reach of the forecast target was informed by the approach of Kong et al. [55]. A forecast resolution of 4 minutes and a forecast span of 10 minutes were selected. The input data set was used in three models to generate independent solar irradiance forecasts, each over 2-, 6-, and 10-minute time horizons. Results were compared to quantify the relative forecasting performance of the models under three different forecast horizons.

As shown in Figure 4.2, Section 2.1, the data pre-processing, explains the process of going from raw to trainable data. Section 2.2 describes the process of this paper's five main supervised image-numerical multimodality models, along with other standard model architectures. Section 2.3 evaluation matrix introduces the two main criteria for model prediction performance evaluation.

4.2.1 Data Pre-processing

Data for the experiments were obtained from the Folsom, California [39] public database, supplemented by clear sky irradiance values from the McClear [140] clear sky irradiance model. Output from the latter was generated using the timestamps of corresponding Folsom data points.

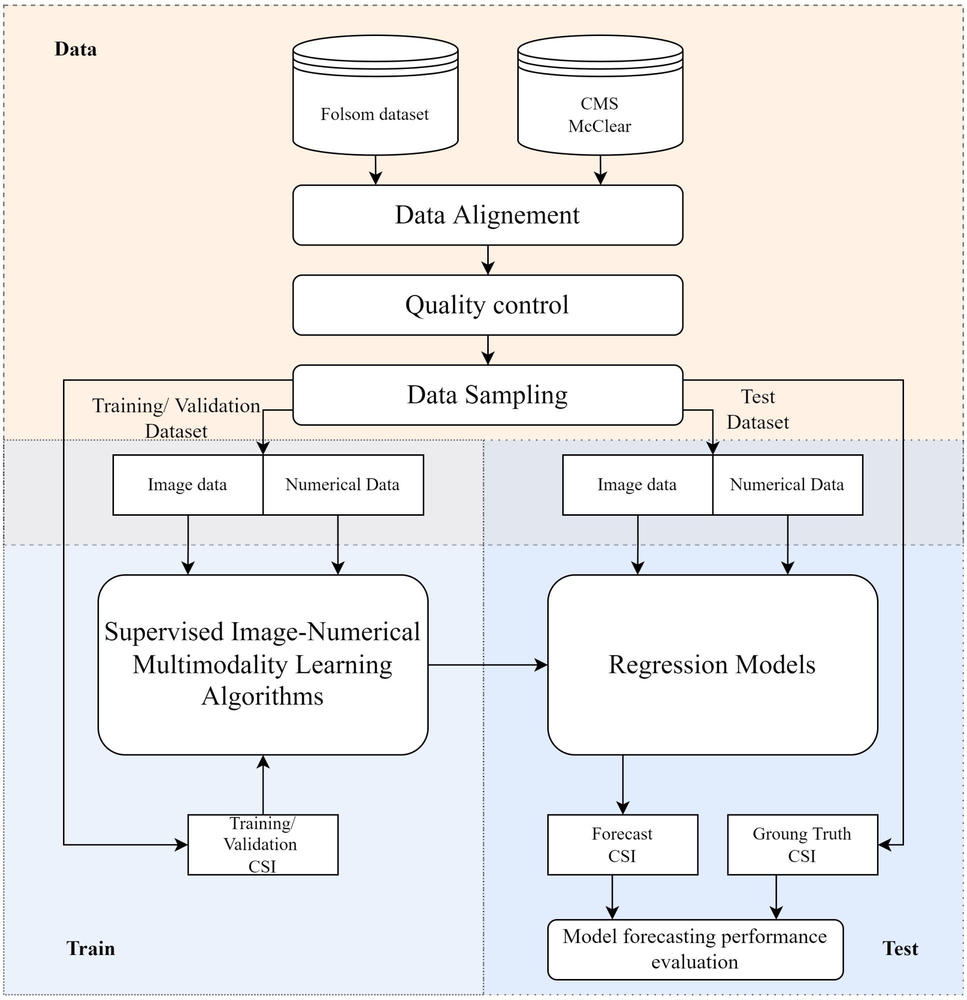


Figure 4.2: Overview of the solar forecasting framework.

Inputs to each of the models comprised a set of time-synchronised data that included clear sky irradiance (GHI, DNI, and DHI), measured irradiance (GHI, DNI, DHI), weather data (dry bulb air temperature, humidity, relative air pressure, wind speed, and wind direction) measured at ground base stations, and solar geometry (solar zenith and solar azimuth angles).

Data Alignment and Quality Control The initial pre-processing stage involves image compression, image alignment to numerical data, quality control, and data normalisation. The Folsom dataset provides raw image data (1536 pixels × 1536 pixels), solar irradiance data, and weather data. These data first went through a process of temporal alignment using timestamps, and the corresponding clear sky irradiance was then sourced from the McClear clear sky model. Following this, quality control filters were applied to screen each piece of data.

For numerical data, a quality control strategy following Yang's [141] work was used to reject data outliers, with decisions being made based on identifying extremely-rare limits [142], a diffuse ratio test [142], and other filters [128].

Images were down-sampled to 128 pixels × 128 pixels, considered the smallest resolution that can be maintained for sky information, using the bilinear method to match the input format of the ANN. In addition, the image dataset showed occasional time shifts possibility due to cumulative errors resulting from the continuous shooting. Data points that showed significant offsets (more than 15 seconds from the timestamps) were removed. Finally, to balance the weights of all inputs, all RGB channels and numerical data of the images were normalised to the interval $[0, 1]$, except for the solar altitude angle, which was normalised to $[-1, 1]$ after a trigonometric transformation.

Segmentation and Resampling of Dataset The Folsom dataset provides numerical and image data for three years from 2014-2016. In this study, the 2014 data was used as the training set, the 2015 data as the validation set, and the 2016 data as the test set. Following the data alignment and quality control stage, these contained 195k, 233k, and 228k data points, respectively. Within these datasets, the sample size for sunny periods was much larger than that for non-sunny days, the former accounting for approximately 60% of the entire dataset. As may be inferred from the cumulative distribution of CSI on the left side of Figure 4.3, the dataset is unbalanced, with clustering of CSI values between approximately 0.9 and 1.05. Recent research [34] suggests that unbalanced datasets can generate models biased towards non-critical conditions – in the case of the Folsom dataset, the sunny periods. To guard against potential bias, a simple algorithm was used to filter out consecutive data points within the sunny period. Expressly, a data point was excluded if the preceding five and following 10 points were' sunny' as defined by the limits of the data clustering, i.e., a CSI greater than 0.9 and less than 1.05. The right side of Figure 4.3 shows the data distribution after resampling, suggesting it is better balanced. The remaining datasets contain 86K, 100K and 94K data points, respectively.

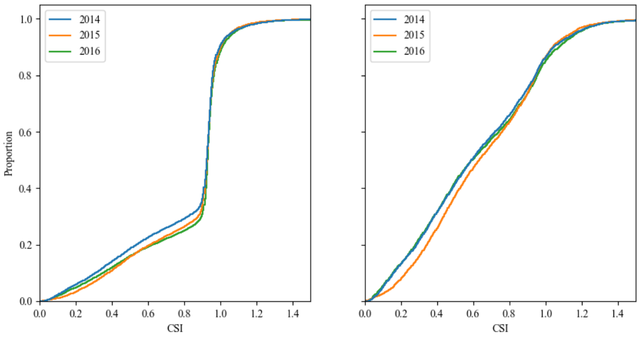


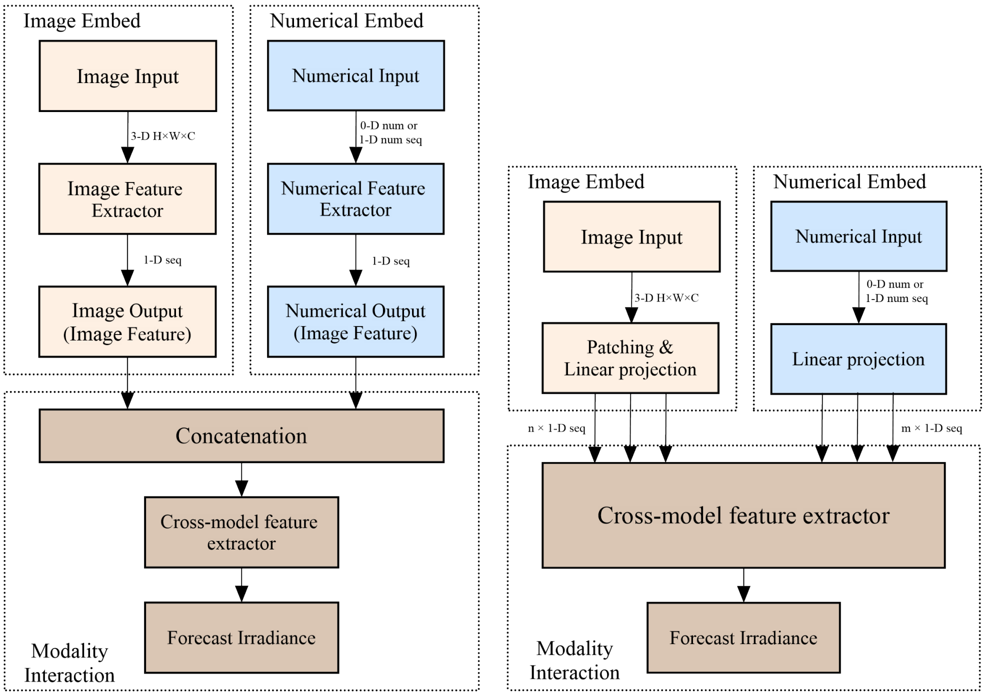
Figure 4.3: Data before (left) and after (right) resampling CSI distribution

Due to computer memory and training time constraints, it was verified that a quarter of the data was randomly sampled(in Appendix A, Figure A.1). The final training, validation and test datasets used for analysis contained approximately 21k, 25k and 23k data points, respectively. The detailed monthly distribution of the final data is shown in Appendix A, Figure A.2

Due to the computer memory and training time constraints, only a quarter of the training data were used, randomly sampled from the training dataset. The final training, validation, and test datasets used in the analysis contained approximately 21k, 25k, and 23k data points, respectively.

4.2.2 Development of Deep-learning Based Irradiance Forecast Model

This section describes the model architectures and modules used in this study. It presents the current dominant architecture for image-numerical bimodal prediction models, i.e., late feature-level fusion architecture, before presenting the new model proposed in this paper, which is based on Transformer encoder architecture and implements early feature-level fusion. The two reference models against which this is benchmarked are then described before comparing the performance of the three approaches. In reviewing the findings, it is essential to bear in mind that modules with a temporal dimension, such as the LSTM module or other Recurrent Neural Networks (RNN), do not form part of the discussion as the focus of this paper is on the exploration of the process of model fusion.



(a) Late Feature-level Fusion [33]

(b) Early Feature-level fusion

Figure 4.4: Schematic diagram of the numerical-image bimodality model

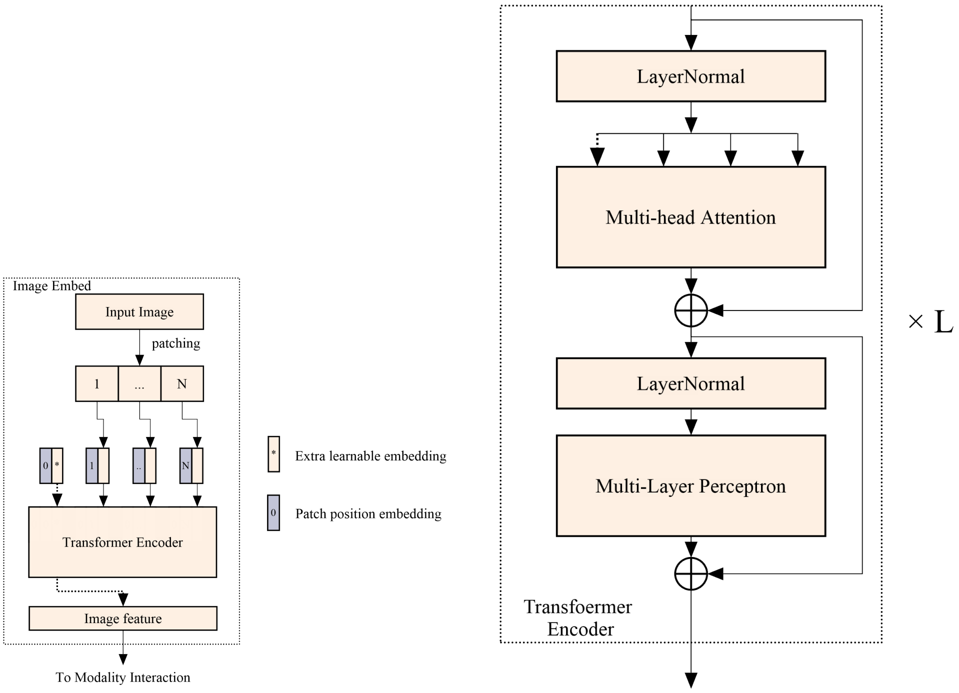
4.2.2.1 Bimodal Model based on Late Feature-level Fusion

Currently, mainstream deep learning-based image-numerical bimodal models are based on late-stage feature-level fusion architectures [47, 50, 42, 33, 55], as illustrated in Figure 4.4(a). The architecture consists of three main components: an image embedding process that extracts the input image features as high-dimensional vectors; a numerical embedding process that extracts the numerical input features as high-dimensional vectors; and a modal interaction module that extracts the joint features from the two vectors after a process of concatenation, which ultimately derives the forecasting results.

CNN - Current Image Embedding Among the sky image-based PV forecast models, CNN and other variants based on convolutional computation are currently the dominant image feature extractors due to their excellent image resolution performance [55, 47, 42]. These extract features from images in a continuous convolutional scan, building a weighting system from detailed to macroscopic images by sequentially expanding the receptive field size of the model through a multilayer repetitive architecture. This study used the most widely accepted ResNet-18 model [99] as a baseline model for CNN image extractors.

ViT - Proposed Image Embedding As mentioned above, methods based on Transformer encoder architecture are emerging as a widely used backbone network for various tasks. The Vision Transformer (ViT) has been developed for image feature extraction [100]. Unlike the convolution-based scanning adopted by CNN models, ViT-based vision models build a weighted system by extracting interconnections between image patches.

As a result, such models can establish relationships between pixels at different areas within the image. This paper postulates that since the main feature of the sky image in short-term solar forecasts is primarily the relative relationship between regions occupied by clouds, clear sky and the sun, the relative importance of fine-grain texture/detail in the image is lower. Based on multiple self-attention, Vit models can extract the more critical larger-scale features in sky images more efficiently.



(a) ViT architecture

(b) Transformer Encoder

Figure 4.5: Schematic diagram of Vision Transformer (ViT) image embedding.

For a module that acts only as an image feature extractor, the computational process can be expressed as

$$\mathbf{z}_{i0} = \left[\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}\right] + \mathbf{E}_{\text{pos}} \quad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C)\times D}, \mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1)\times D} \tag{4.1}$$

$$\mathbf{z}'_{il} = \text{MSA}\left(\text{LN}\left(\mathbf{z}_{il-1}\right)\right) + \mathbf{z}_{il-1}, \qquad l = 1 \ldots L \tag{4.2}$$

$$\mathbf{z}_{il} = \text{MLP}\left(\text{LN}\left(\mathbf{z}'_{il}\right)\right) + \mathbf{z}'_{il}, \qquad l = 1 \ldots L \tag{4.3}$$

$$\hat{\mathbf{z}}_i = \text{LN}\left(\mathbf{z}^0_{i\,L}\right) \tag{4.4}$$

As shown in Figure 4.5(a), the image input $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$ is divided into N patches of side length P and stitched into a 2D sequence $\mathbf{x}_p \in \mathbb{R}^{N\times(P^2\cdot C)}$. Following this, the pixels of each patch are projected linearly onto D dimensions via transformer embedding, a learnable latent vector $\mathbf{E} \in \mathbb{R}^{(P^2\cdot C)\times D}$. Following the process described by Devlin et al. [143], the input after reshaping is stitched with an additional learnable class token, $\mathbf{x}_{\text{class}}$, and embedded with a learnable position component $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1)\times D}$, which describes the spatial relationships between patches. Eventually, the image part of the input is represented as $\mathbf{z}_{i0} \in \mathbb{R}^{(N+1)\times D}$. This input is added to a standard Transformer encoder module, shown in Figure 4.5(b), i.e., a module based on a Multihead Self-Attention (MSA) process [144] and a Multilayer Perceptron (MLP) process, iterated L times. Ultimately, the learnable class token, $\mathbf{x}_{\text{class}}$, is extracted, and after Layer Normalisation (LN), is output as a high-dimensional vector $\hat{\mathbf{z}}_i$, representing the image feature. Where the MSA module is calculated as:

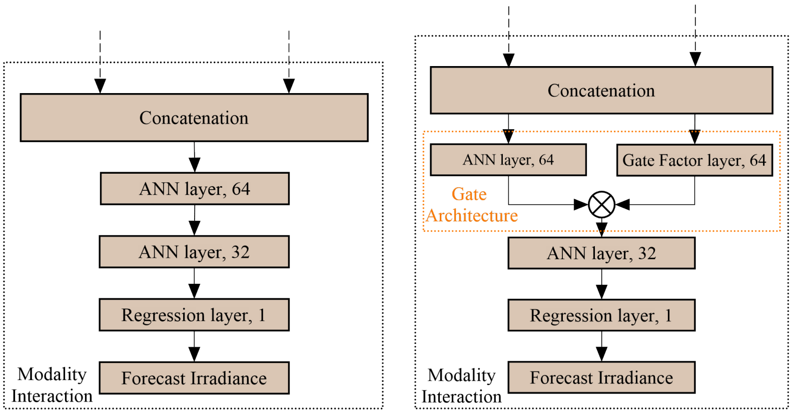
$$\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{\mathrm{d_k}}})\mathbf{V} \tag{4.5}$$

$$\mathrm{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{Concat}(\mathrm{head_1},\ldots,\mathrm{head_h})\mathbf{W}^{\mathrm{O}} \tag{4.6}$$

$$\mathrm{where\ head_i} = \mathrm{Attention}(\mathbf{Q}\mathbf{W}_{\mathrm{i}}^{\mathrm{Q}},\mathbf{K}\mathbf{W}_{\mathrm{i}}^{\mathrm{K}},\mathbf{V}\mathbf{W}_{\mathrm{i}}^{\mathrm{V}}) \tag{4.7}$$

Where, Equation 4.5 refers to the Scaled Dot-Product Self-Attention algorithm, which searches attention by calculating the dot product of self-attention points of the quarry, **Q**, and Key, **K**, and scales the attention based on the dimension d_k of **Q** and **K**, and then calculates the softmax activation to obtain the relative attention weight. Finally, it performs the **V** value. Equation 4.6 and 4.7 refer to the Multi-Head Attention algorithm. First, **Q**, **K**, and **V** are projected into different representation subspaces, i.e., multiple heads, by trainable matrices $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$, respectively, and self-attention is calculated independently using different representation for **Q**, **K** and **V**. Then, the attention of multiple heads is concatenated and scaled back to the input size through the trainable matrix $\mathbf{W}^O$.

MLP - Current Modality Interaction Embedding Currently, Multilayer perceptron (MLP), also known as multilayer feedforward Artificial Neural Networks (ANN), are widely used as one-dimensional vector feature extractors in models with numerical inputs [145]. MLPs are also used widely in the modal fusion phase of image-numerical bi-modal solar forecasting models [33, 47, 50, 42]. As mentioned above, when MLPs are used as a cross-modal feature extractor, as shown in Figure 4.6(a), the direct concatenation that takes place before feature extraction fails to make effective connections between the input parameters and the interaction of the inter-model outputs is entirely dependent on the subsequent adaptive of the network architecture to such outputs. Also, due to the heterogeneity of the different data, models based on MLPs face multiple challenges when performing mapping (converting image information into irradiance data) and fusion forecasting (combining information from two modalities to predict ramp events). These challenges include instances where information from different modalities has different predictive power and noise topology, or instances where models cannot capture features from one of the modalities.



(a) MLP feature extractor (b) Gated-MLP feature extractor

Figure 4.6: Schematic diagram of modality interaction in late feature-level fusion models.

MLP with Gate Architecture - Proposed Modality Interaction Embedding In order to improve the attention given to target features in the modality processed by the MLP and suppress feature activation in irrelevant regions, this paper proposes adding a layer based on attention gate architecture, as shown in Figure 4.6(b). This is like the input gate architecture in LSTM [107]. The gate architecture generates a gating coefficient for each node in MLP with the same dimensionality as the input feature. Then it converts this into an attention weight map multiplied by the original feature. The attention gate focuses the model's attention on essential regions of the input data and neglects irrelevant regions. The simplicity of this approach makes it possible to improve feature extraction without a significant increase in computing cost.

4.2.2.2 Transformer-based Early Feature-level Fusion

As mentioned above, the MSA-based ViT model finds applications beyond image processing. Because the MSA module inputs are a series of 1D multi-dimensional vectors or tensors, it is possible to input images and numerical data in parallel. As an alternative to CNNs, such backbone networks have been shown to offer outstanding capabilities in several fields dealing with multimodality tasks, such as image and text [146], video and text [147], etc. However, there is, as yet, no such work applied to the field of solar energy forecasting. Therefore, inspired by Kim et al. [148], this paper speculates that multimodality input short-term irradiance forecast models that combine sky images and measurement logs can also be constructed using the Transformer Encoder module as the backbone network to replace both the CNN visual layer and the MLP numerical regression computational layer to construct input data with early feature-level fusion.

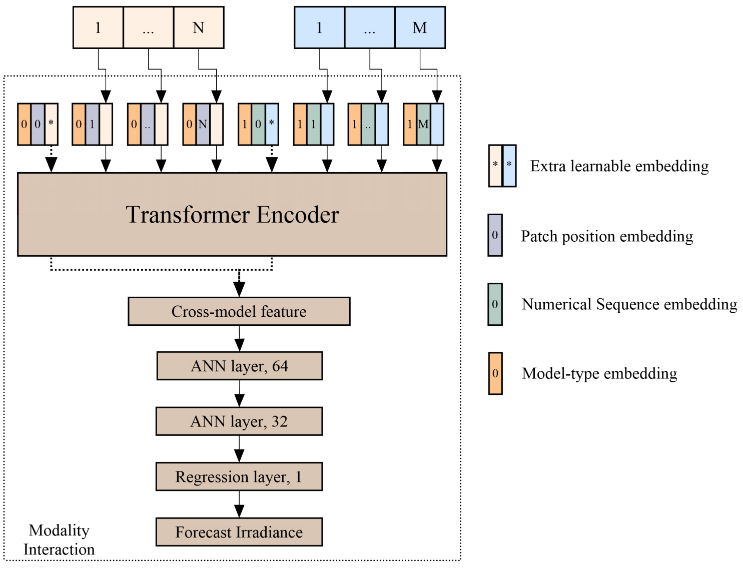


Figure 4.7: Schematic diagram of image/text bimodal transformer architecture.

The proposed early feature and fusion model is based on the Transformer encoder architecture shown in Figure 4.7. The main inputs to the model comprise image data and numerical data. For the image data, input follows the patching process illustrated in Fig 4.5(a). For the numerical data, a standard unbiased MLP for numeric features is used to up dimension the numeric information to D, $\mathrm{MLP}(\mathbf{y}) \in \mathbb{R}^{1\times D}$, and provide a learnable class token. The numerical data are divided into five groups based on type: solar irradiance, clear sky solar irradiance, sun angle, ground wind conditions, and weather parameters (dry bulb air temperature, humidity and relative air pressure). As with image processing similar to the ViT process, the image part of the inputs is represented as $\mathbf{z}_{i0}$. Meanwhile, the learnable class token for numerical data, $\mathbf{y}_{\text{class}}$, combined with learnable position embedding $\mathbf{E}_{\text{seq}} \in \mathbb{R}^{(M+1)\times D}$ is used to describe the position relationships within the data sequence. The numerical part of the input is represented as $\mathbf{z}_{n0} \in \mathbb{R}^{(M+1)\times D}$. Finally, $\mathbf{z}_{i0}$ and $\mathbf{z}_{n0}$ are embedded separately in the model type embedding process as $\mathbf{z}_i^{\text{type}}$ and $\mathbf{z}_n^{\text{type}}$, before the process of concatenation to generate $\mathbf{z}_0 \in \mathbb{R}^{(M+N+2)\times D}$. The vector $\mathbf{z}_0$ is iteratively updated through L-depth transformer layers up until the final sequence $\mathbf{z}_l$. The final $\hat{\mathbf{z}}$ representing the forecast vector is generated by a linear projection of the two learnable vectors $\mathbf{z}_{i\,L}^{0}$ and $\mathbf{z}_{nL}^{0}$ in series with hyperbolic tangent activation.

The overall data processing can be described as

$$\mathbf{z}_{i0} = \left[\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}\right] + \mathbf{E}_{\text{pos}} \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C)\times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D} \tag{4.8}$$

$$\mathbf{z}_{n0} = \left[\mathbf{y}_{\text{class}}; \text{MLP}(\mathbf{y}^1); \cdots; \text{MLP}(\mathbf{y}^M)\right] + \mathbf{E}_{\text{seq}} \qquad \mathbf{E}_{\text{seq}} \in \mathbb{R}^{(M+1)\times D} \tag{4.9}$$

$$\mathbf{z}_0 = \left[\mathbf{z}_{i0} + \mathbf{z}_i^{\text{type}}; \mathbf{z}_{n0} + \mathbf{z}_n^{\text{type}}\right] \tag{4.10}$$

$$\mathbf{z}'_l = \text{MSA}\left(\text{LN}\left(\mathbf{z}_{l-1}\right)\right) + \mathbf{z}_{l-1}, \qquad l = 1 \ldots L \tag{4.11}$$

$$\mathbf{z}_l = \text{MLP}\left(\text{LN}\left(\mathbf{z}'_l\right)\right) + \mathbf{z}'_l, \qquad l = 1 \ldots L \tag{4.12}$$

$$\hat{\mathbf{z}} = \text{LN}\left(\left[\mathbf{z}^0_{i\,L}; \mathbf{z}^0_{nL}\right]\right) \tag{4.13}$$

For all experiments presented in this paper, hidden size D of 192, later depth L of 12, patch size P of 8, MLP size of 192, and the number of attention heads of 12 are used.

4.2.2.3 Smart Persistent Model

This paper uses the Smart Persistent Model (SPM) as the benchmark for evaluating the performance of alternative modelling approaches. In contrast to the Persistent Model (PM), which assumes that solar irradiance remains constant throughout the forecast interval, the SPM assumes that the clear sky index remains constant. This offers the advantage that potential seasonal and temporal factors are added to the model as default preconditions and can be expressed as follows:

$$\hat{\mathbf{z}}_{\text{SPM}}(T + \Delta T) = \frac{\mathbf{z}(T)}{\mathbf{z}_{\text{clear}}(T)} \cdot \mathbf{z}_{\text{clear}}(T + \Delta T)$$

Implicit in using an SPM is the requirement for a clear sky model as a reference for clear sky irradiance. This paper uses the McClear model [140] for clear sky irradiance generation.

4.2.2.4 AutoML - Additional Machine Learning Benchmarks

As part of evaluating the performance of image-numerical multi-modal learning an additional predictive regression model based on only the numerical input data was created to serve as an additional benchmark. This used the AutoGluon [149] tool, which was used to train a forecast model and is based on automated machine learning (AutoML). AutoGluon can automate the model selection, hyper-parameter tuning and model integration. The final model was generated by integrating one or more of neural networks: LightGBM boosting trees [150], CatBoost boosting trees [151], random forests, extreme randomisation trees, and K-nearest Neighbours, and based on multilayer stack resembling and repeated k-fold bagging strategy to increase the final accuracy [149]. In the presentation and discussion of the results, this model is referred to using the abbreviation NUM.

4.2.2.5 Summary of Models and Criteria for Evaluating Performance

A summary of the models used in this paper is provided in Table 4.1. The SPM, NUM, and CNN-L models represent benchmarks for persistence, numerical-based machine learning, and combined image-numerical-based deep approaches, respectively. ViT represents the image backbone network based on the Transformer encoder architecture proposed here as the alternative to using a CNN. The terms appended to CNN and ViT define the approach taken to fusion where -L represents late feature-level fusion architecture, -LG represents different gate architecture, and -E represents feature-level fusion architecture. More detailed model architecture is presented in Appendix B.

Table 4.1: Irradiance forecasting models explored through this paper.

Models	Inputs		Encoder architecture		Fusion	Reference
	Numerical	Images	Numerical	Images		
SPM	✓		Persistence	/	/	
NUM	✓		AutoGluon	/	/	[149]
CNN-L	✓	✓	MLP	Res-18	Late	[33, 42, 55]
CNN-LG	✓	✓	MLP	Res-18	Late, Gated	[107]
ViT-L	✓	✓	MLP	ViT-Base-patch8-128	Late	[100]
ViT-LG	✓	✓	MLP	ViT-Base-patch8-128	Late, Gated	[100, 107]
ViT-E	✓	✓	Transformer	ViT-Base-patch8-128	Early	

4.2.3 Evaluation Matrix

Two evaluation criteria were used to evaluate the performance of these models. The first involved quantifying the error between the predicted irradiance $\hat{\mathbf{z}}$ and the ground truth data $\mathbf{z}^*$. Standard metrics widely used by the solar forecasting community, and adopted in this paper, include FS based on metrics such as RMSE, MAE or MSE to measure the running accuracy of the forecast. The second criterion was based on BP, which quantifies forecasting ability in the presence of a Ramp Event, i.e., a sudden rise or fall in irradiance due to sudden changes in cloud cover.

Forecast Skill Statistical indicators such as RMSE, MAE or MSE tend to behave in a homo-trending manner in solar forecasting. The Forecast Skill (FS), adopted in this paper, used the Smart Persistent Model (SPM) clear-sky model to represent the baseline performance and only RMSE to quantify error, as follows:

$$\text{Forecast Skill} = (1 - \frac{RMSE_{\text{Model}}}{RMSE_{\text{Baseline}}}) \times 100\%$$

Balanced Precision Although FS can quantify the general error between model forecasts and ground truth, it does not demonstrate the ability of models to forecast ramp events. These qualitative behaviours are critical in PV generation as the rapid power fluctuations increase the system frequency stabilisation cost. Balanced precision (BP) is a metric developed for ramp events [152], which defines a ramp as a rapid solar irradiance event with a rate of change exceeding 10% of the maximum installed capacity. This paper uses a modified version of the metric where periods exhibiting a rate of change in GHI exceeding 100 $W/m^2/min$ are defined as ramp events

– this is to reflect the fact that for the database used, there is not a grid as a reference., Following the suggestions of Kong et al. [55], this paper also defines the ramp direction. For each forecast, data can be classified into three categories based on the magnitude and direction of change in solar irradiance, i.e., positive ramp events where cloud cover diminishes, adverse ramp events where cloud cover grows, and periods of relatively consistent irradiation, implying an absence of ramp events. After categorising the forecast data to identify ramp events, BP may be defined as:

$$\text{Balanced Precision} = \frac{1}{2}\sum_{c\in C}\frac{\mathscr{T}_c}{\mathscr{N}_c}$$

Where $\mathscr{T}_c$ represents successfully forecast events in the positive or negative ramp category and $\mathscr{N}_c$ represents the total sample in the positive or negative ramp category.

4.3 Results

Modelling was undertaken using a PC with a 3.8 GHz AMD Ryzen 9 3900X CPU and a GeForce RTX 2080 SUPER GPU on the Tensorflow 2.5 [153] platform with Keras [154] built in. Five replicate trials were carried out for each image model to reduce errors introduced by the random nature of observation order and the randomness in random number generator in training.

4.3.1 Quantitative Solar Irradiance Forecasting

Results for the criteria used to evaluate the quantitative capabilities of the five image-numerical models (CNN-L, CNN-LG, ViT-L, ViT-LG, ViT-E) and two numerical models (SPM and NUM) are summarised in Table 4.2.

It may be seen that all models outperformed the SPM model, which was used as the FS baseline predictive power. The AutoML-based NUM model achieved the best forecast results at the 2-minute horizon; the CNN model with a gate architecture achieved the best results for the 6-minute and 10-minute forecasts. Overall, there was a significant difference in model FS levels at the 2-minute horizon, and this difference diminished as the forecast horizon was extended. In particular, the models based on ViT as the graphical feature extractor were all inferior to the CNN-based models in FS.

It is worth noting that for the late feature level fusion models, the effect of gate architecture is not significant, with the difference in FS being less than 1% across all models, except the ViT-LG model, which delivers significantly lower FS at the 2-minute time horizon. The ViT-E model, where the numerical and image inputs share a single encoder, outperforms both the ViT-L and ViT-LG models, where features are extracted separately and then fused at all forecast time horizons. As shown by the linear regression curves in Figure 4.8, the errors in all models manifest as an overestimation of irradiance at lower irradiance and an underestimation at higher irradiance.

Table 4.2: GHI forecast results. The errors are expressed as mean ± standard deviation. Predicted skill was calculated relative to the SPM model.

Models	2 min		6 min		10 min	
	RMSE (W/m^2) ↓	FS (%) ↑	RMSE (W/m^2) ↓	FS (%) ↑	RMSE (W/m^2) ↓	FS (%) ↑
SPM	85.62	N/A	117.57	N/A	129.67	N/A
NUM	77.31	**9.70**	98.69	16.06	113.14	12.75
CNN-L	79.37±0.55	7.29±0.64	98.68±0.45	16.07±0.38	105.15±0.49	18.9±0.37
CNN-LG	79.89±0.66	6.68±0.76	98.54±0.64	**16.18±0.54**	104.15±0.37	**19.68±0.29**
ViT-L	82.77±0.82	3.32±0.96	99.97±0.65	14.97±0.55	105.28±1.27	18.81±0.98
ViT-LG	85.16±1.34	0.53±1.56	101.29±0.8	13.84±0.67	105.26±0.45	18.82±0.34
ViT-E	81.45±0.68	4.87±0.79	98.68±0.72	16.06±0.61	104.91±0.7	19.09±0.53

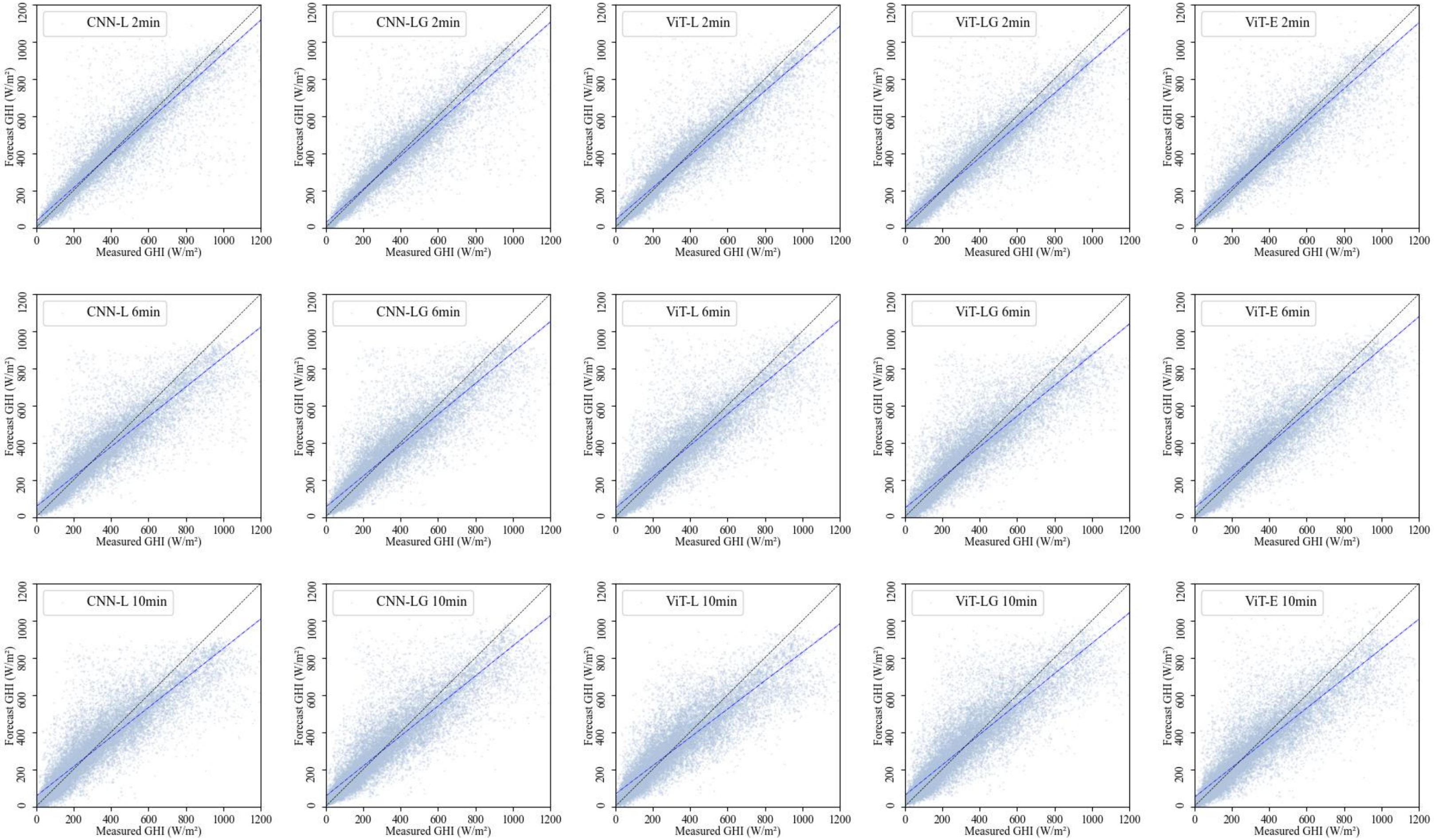


Figure 4.8: Forecasts using the image-numerical bimodal models over three time horizons. The blue dashed line is the predicted linear regression and the black dashed line is the expected regression (predicted value = actual value).

4.3.2 Qualitative Solar Irradiation (Ramp Event) Forecasting

Table 4.3 presents the qualitative results for all models regarding how often Ramp Events were accurately predicted, and Figure 4.9 illustrates performance as a confusion matrix. It may be seen that models based on the ViT framework achieve the best performance across all time horizons. It may also be seen that the qualitative results exhibit a similar trend to the quantitative results, i.e., the variability between models decreases as the forecast time horizon increases. However, the variability is more pronounced in the case of qualitative results. At all horizons, the BP of the ViT-based models was more significant than or equal to that of the CNN-based models. Additionally, the performance of the models with gate architectures exceeded or equalled that of the non-gated models. Interestingly, the BP of the widely used CNN-L fusion framework was even lower than that of the purely numerical forecast-based model NUM for the 2-minute forecast. Even after the addition of the gate architecture enhanced the model's BP ability, its performance was still lower than that of NUM. Finally, it may be seen that models successfully captured falling RE more frequently than rising RE, the exception being the ViT frame model over the 2-minute horizon.

Table 4.3: Ramp Event forecasting results. For image-numerical models, results are expressed as the mean ± standard deviation of the results of five replicate trials.

Horizon	Models	Increase RE ↑	Decrease RE ↑	BP (%) ↑
2 min	SPM	0/1131	4/1071	0.19
	NUM	135/1131	214/1071	15.96
	CNN-L	62.6±62/1131	171.8±34.9/1071	10.78±3.41
	CNN-LG	96.2±58.2/1131	188.6±29.7/1071	13.05±1.94
	ViT-L	226.8±52.5/1131	180.8±55/1071	18.46±1.02
	ViT-LG	241±29.6/1131	185.4±34.9/1071	19.31±1.1
	ViT-E	239.4±18.8/1131	206.2±28.6/1071	**20.21±2.01**
6 min	SPM	0/1979	23/2028	0.57
	NUM	421/1979	697/2028	27.82
	CNN-L	518±84.7/1979	659.8±95.3/2028	29.35±2.26
	CNN-LG	537.4±91.5/1979	759.4±59.7/2028	32.3±1.03
	ViT-L	548.8±63.3/1979	752.6±33.2/2028	32.42±1.35
	ViT-LG	609.2±25.8/1979	752.2±55.6/2028	**33.93±1.78**
	ViT-E	671.8±28.7/1979	660.6±27.8/2028	33.26±0.9
10 min	SPM	0/2483	42/2603	0.81
	NUM	212/2483	426/2603	12.45
	CNN-L	808±61.7/2483	1101±74.9/2603	37.42±1.52
	CNN-LG	819.8±33.5/2483	1072.8±85.6/2603	37.11±1.52
	ViT-L	788±76.4/2483	1133.8±123.1/2603	**37.64±1.58**
	ViT-LG	852.4±93.5/2483	1050±93.2/2603	37.33±2.55
	ViT-E	819.6±140.4/2483	1060.6±148.6/2603	36.87±2.55

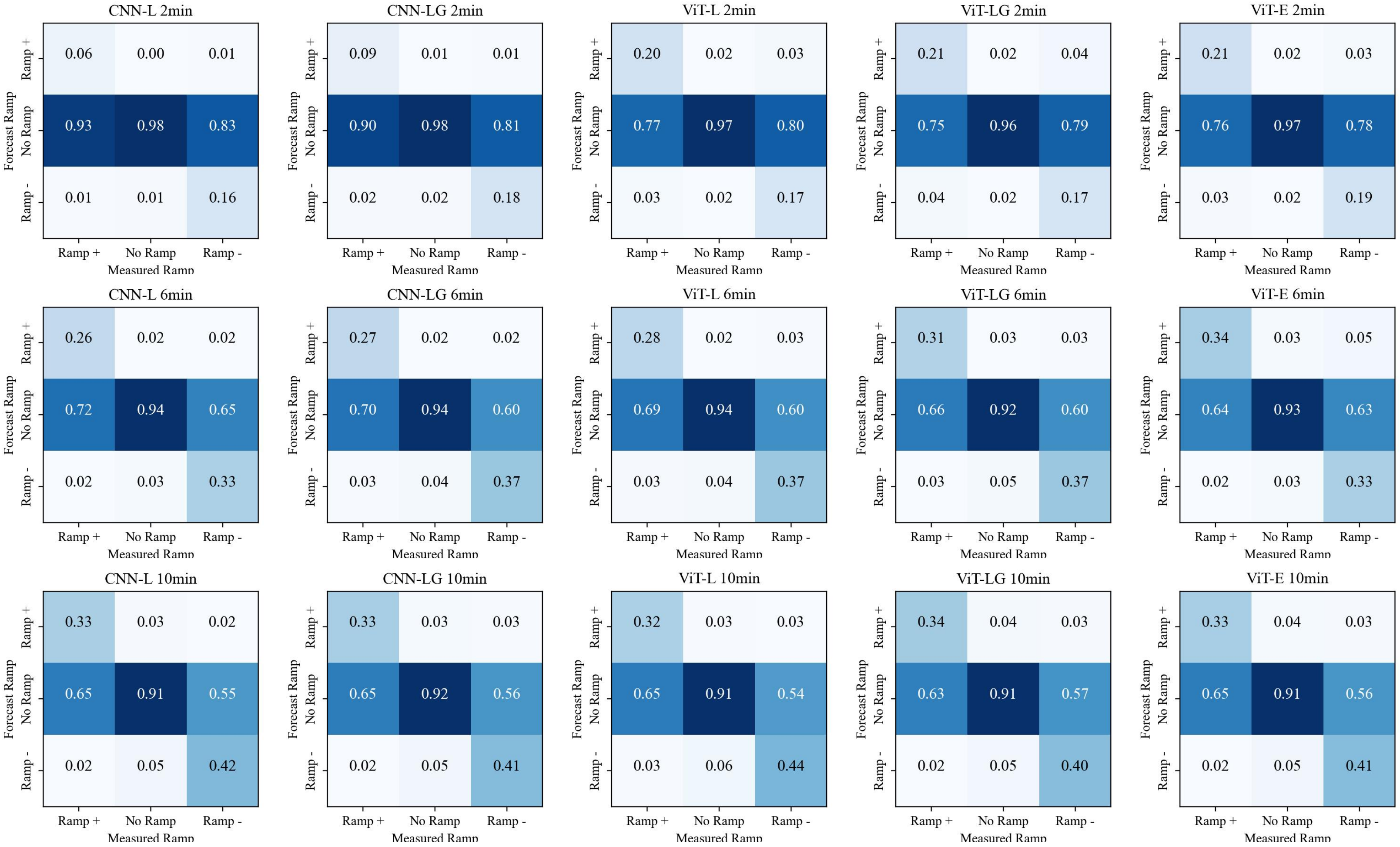


Figure 4.9: Confusion matrix of Ramp predictive power for 5 different image-numerical models on 3 time horizon.

4.4 Discussion

4.4.1 Comparison of Model Variability

Figure 4.10 shows all models' combined FS and BP performance. As the SPM model has little RE predictive power, it can be approximated as being at the coordinate system's origin and not plotted in the figure. As observed in the work of Paletta et al., [35], the effect of architecture used in different models fed by the same inputs gradually decreases as the size of the forecast horizon grows. For the bimodal frameworks studied here, it is difficult to identify any significant variability in the models at the 10-minute time horizon.

In reflecting upon performance, distinguishing between the relative importance of quantitative versus qualitative measures is worth distinguishing. In the field of solar forecasting, the merit of a model is usually determined using quantitative error, i.e., FS. The optimal strategy for such models fitted by statistical errors for rapidly changing cloudy weather is often based on mean reversion. However, capturing Ramp events is more critical for very short-term solar forecasting (10 minutes or less) as the information may be used to inform grid operability.

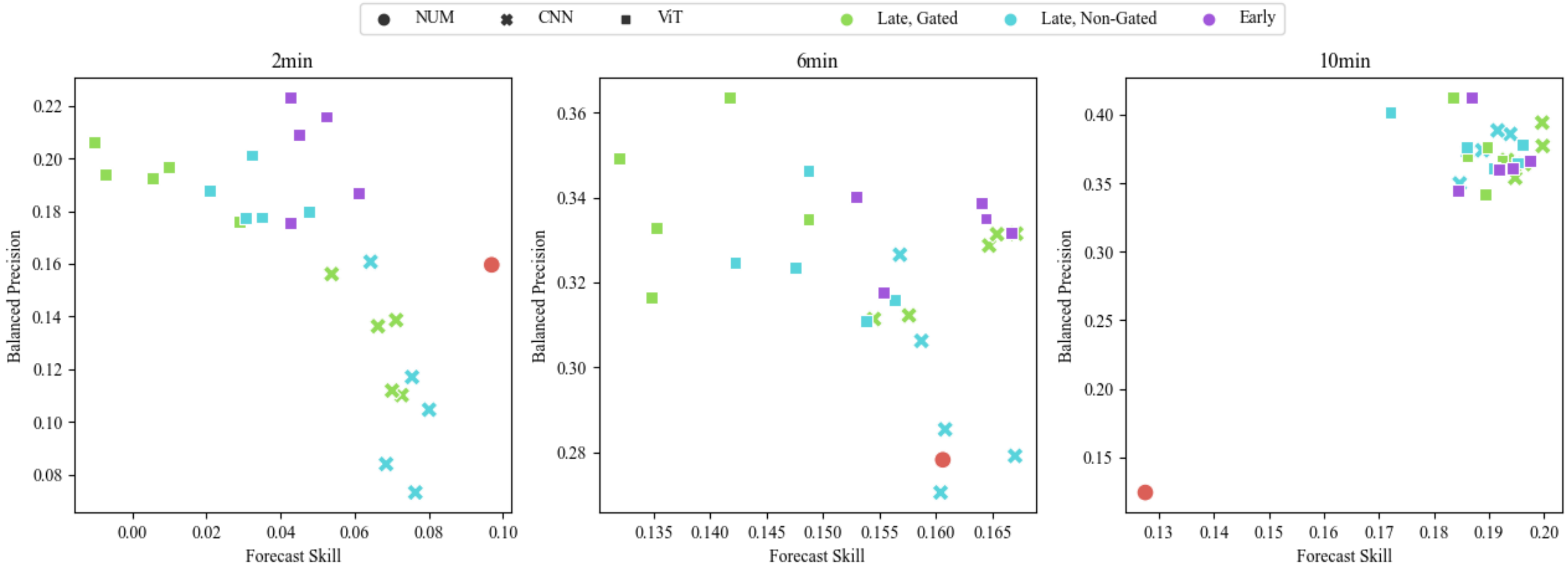


Figure 4.10: FS and BP results for all models over different time horizons.

Such ramp forecasts require the model to predict the occurrence of sudden and significant changes in irradiance, as opposed to consistent predictions of absolute irradiance, and metrics that quantify performance in terms of statistical error, e.g., RMSE, tend to penalise the former qualities. The 2- and 6-minute results from Figure 10 show that the models with high BP performance, i.e., ViT-L and ViT-LG, perform poorly when performance is expressed as FS, while the opposite is true for CNN models. The early feature-level fusion model, ViT-E, maintained relatively strong BP performance in the 2- and 6-minute predictions compared to the late model, and both delivered the best FS. It is posited here that there are two main reasons for this: the model's ability to abstract image features and the dual-modality strategy the model adopts to accommodate the visual and numerical inputs.

4.4.2 Impact of Images in Bimodal Models

To explore the sensitivity of different models to the image input, randomly selected images were used as inputs to the models on 17 June at 18:35 while keeping the numerical input unchanged. The condition of the sky at this time is shown in Image 1 of Figure 4.11, as are the replacement images used in the analysis - Images 2 to 5, are taken from the same day but with different sky conditions and Image 6, which is fabricated and comprises only black pixels. The output from this analysis is plotted in Figure 4.11 and shows that models based on ViT as an image feature extractor are more significantly affected by the image input than those based on CNN under complex sky conditions. In addition, most models with gate architecture (light blue in the figure) are more sensitive to images than those based on late fusion (light brown in the figure). Furthermore, the ViT-E model

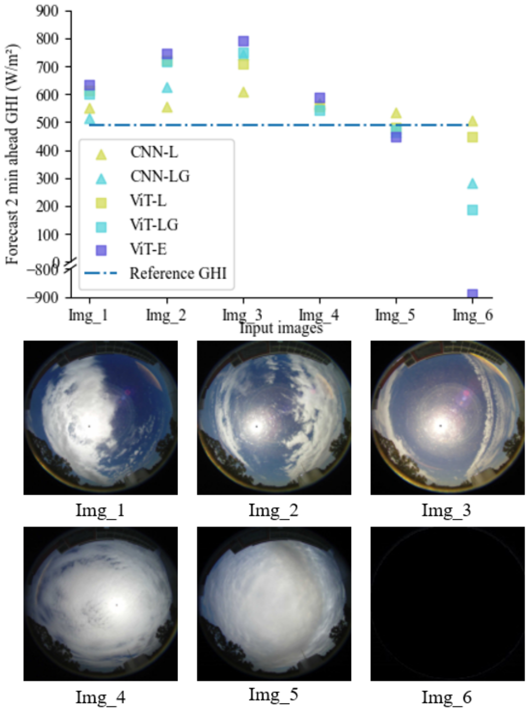


Figure 4.11: Image sensitivity testing for a 2-minute time horizon. Image 1 is the original image input, and Images 2 to Image 6 are replacement inputs. The upper panel shows the 2-minute ahead prediction from the five image-numerical bimodal models. The blue dashed line represents the output from the SPM model.

is always the most sensitive to images. Interestingly, when fed a picture without any information, the output of CNN-L is almost unaffected, while ViT-E deviates significantly from the reference GHI value. These results suggest that the widely used CNN-L architecture is relatively insensitive to image inputs. In particular, the model is highly insensitive to incorrect input. The findings of Paletta et al. may explain this, [42] who suggest, after evaluating multiple graphical models, that fusion models always behave like a smarter SPM. i.e., the model lacks interaction between image and numerical inputs, including alignment, translation, and co-representation. This makes the model dependent on the numerical inputs and relatively insensitive to the image-based output. To address this shortcoming, methods that use an image feature extractor that is more effective at parsing images, such as ViT, or enhancing the interaction between image and numerical data, such as a gate architecture, can be considered as more effective approaches.

4.4.3 Interaction of Image and Numerical Data in ViT-E

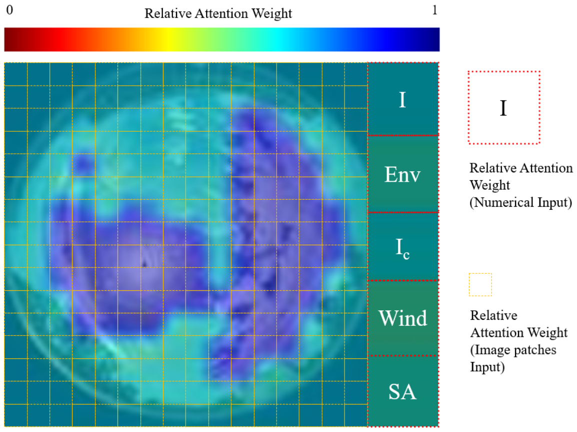
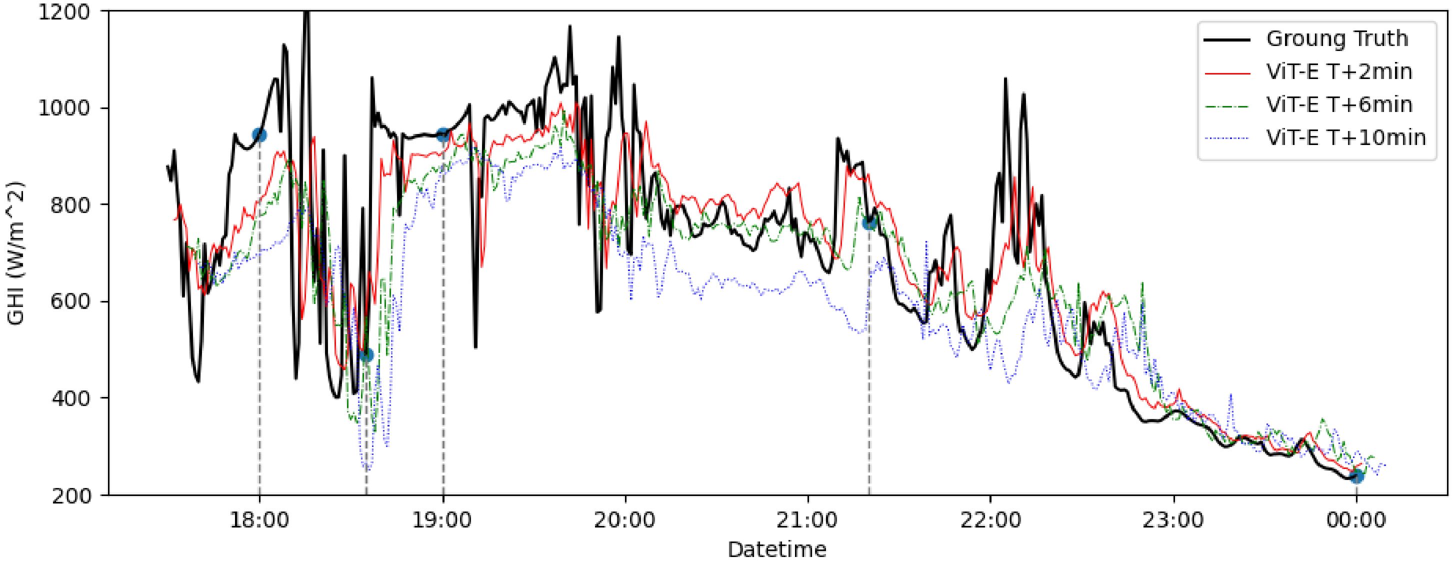


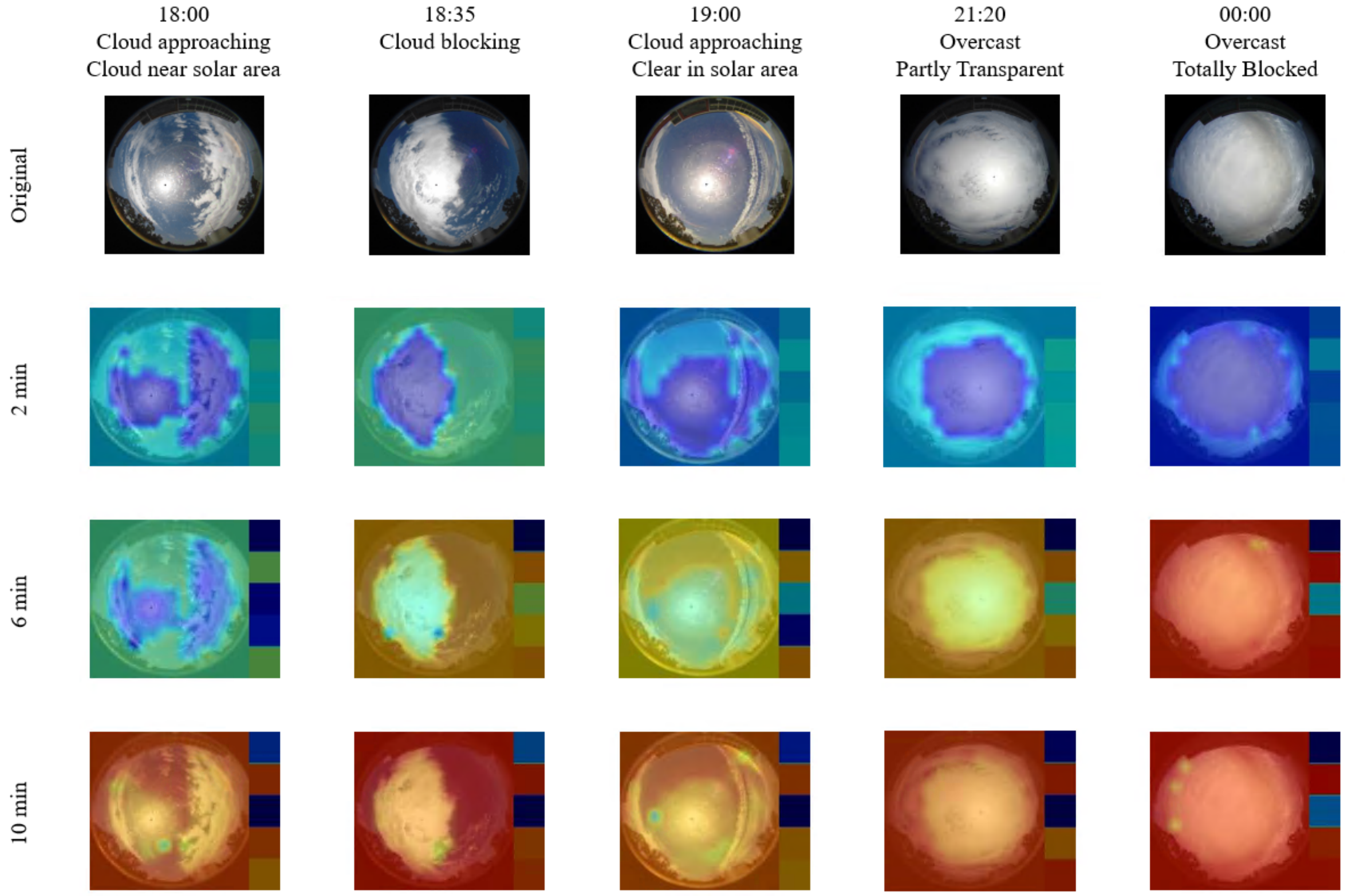
Figure 4.12: ViT-E model visualisation indicating relative attention weights. The colour of the heat map within each patch reveals its relative value in terms of average attention across all heads.

To understand how the Self-Attention mechanism processes image-numerical information across modalities, the attention layer of the ViT-E model was abstracted and overlaid with the input for visualisation, as shown in Figure 4.12. The visualised heat map consists of two main parts: on the left side are the relative attention weights corresponding to the 256 patches in the image input, and on the right side are the relative attention weights corresponding to five sets of numerical inputs, in order from top to bottom: irradiance, ambient environment, clear sky irradiance, wind condition, and solar angle. Figure 4.13 shows the GHI prediction from the ViT-E model for three different forecast horizons for 17 June. A sample of five images, including those used in Figure 4.12, representing a range of sky conditions were extracted and processed to visualise the model attention weights described above, shown in Figure 4.14.



(a)

Figure 4.13: GHI predictions from 17 June, based on ViT-E 2-, 6-, and 10-minute forecasts.



(a)

Figure 4.14: Attention map of the ViT-E model based on five representative GHI conditions from Figure 4.13.

It may be seen from Figure 4.14, that the longer the forecasting horizon, the lower the attention weight of the model to the image-side input and the higher the attention weight to the numerical input. In the 2-minute ahead prediction, different levels of cloud cover and sun position significantly affect the model's attention. For scenarios with low cloud-sun correlation, such as those with significant areas of clear sky in the region around the sun, or those where the sun is obscured by cloud, the model assigns weights to numerical and image models in a balanced manner. The model assigns more attention to the images for scenarios with high cloud-sun correlation, such as cloud approaching or cloud blocking part of the sun. In the 6-minute ahead model, although the distribution of attention weights for the images reflects that of the 2-minute ahead model, the weighting of the numerical data is the most critical part of the model. This trend of assigning a gradually decreasing image weighting continues in the 10-minute ahead model. The model primarily depends on irradiance and clear sky irradiance numerical inputs rather than the images.

This pattern of behaviour explains the variability in model performance observed in Figure 4.10 10, where the forecast accuracy declines as the prediction window is lengthened. That is, the impact of the details in the pictures on the prediction decreases as the prediction scale is lengthened. Although other potentially valuable information visible in the images (e.g., air mass) might still benefit the model's predictive capabilities and thus outperform models without an image input, enhancing the feature extraction capability for the images for these more extended time horizon forecasts is unlikely to deliver better model performance. This observation matches that made concerning models based on the classical image analysis method for forecasting GHI [26], i.e., the gain offered by including image data in predictions is more pronounced for time horizons below five minutes and gradually decreases for those beyond five minutes.

We believe the trend is a good explanation for model performance variability in Figure 4.10 declines as the prediction window is lengthened. That is, the impact of the details in the pictures on the prediction is gradually decreasing as the prediction scale is lengthened. Although other potentially visible information in the images (e.g., air mass) can still enable the model to benefit in prediction and thus outperform the model without image input, enhancing the model's feature extraction capability for the images at this point no longer leads to better model performance. This is similar to the model based on the classical image analysis method for forecasting GHI [26], i.e., the gain of image data on prediction is more pronounced within five minutes, while it starts to decrease gradually after five minutes.

The results from this study suggest that there are advantages to using the transformer framework for combined image-numerical ultra-short-term solar forecasting. Specifically, the model extracts features based on the association between each input element, i.e., image patches and numerical features and dynamically assigns the impact of each element on the final prediction based on these features. ANN-based architectures do not confer this functional advantage as model fusion feature extractors.

In addition, as shown in Figure 4.14, the 10-minute forecast irradiance has a similar weighting to the clear irradiance. In other words, clear sky irradiance is of equal importance to prevailing irradiance for solar irradiance prediction. The advantages of using CSI, i.e. the ratio of GHI to clear GHI, rather than using GHI directly as a prediction target [49], are intuitively demonstrated.

4.5 Summary

Accurate short-term forecasting is essential for predicting solar power output and, thus, for effective grid management. This study found that the modal interaction component has been under-appreciated in previous studies of deep learning models for solar forecasting that combine images with numerical inputs. Also, there is ambivalence between the quantitative and qualitative performance of late feature-level fusion models for single images and numerical fusion in such models. Therefore, this project proposed the ViT-E model as complementarity in quantitative and qualitative forecast performance by varying the modal interactions to achieve relatively superior performance. In addition, the study explored the weighting of image inputs in this model class. The results show that the longer the forecast duration in a single image forecast, the less importance the image accounts for. At forecasts of up to 10-minute horizons, the features that can be extracted from the image input by current vision models are minimal. As mentioned in [120], the model's accuracy is as important as its interpretability in advancing its understanding and development. This study reveals a potential shortcoming in current multimodal solar prediction: model validation relies only on performance improvements for the results, and there is a lack of interaction studies between the actual performance of the different modes of the model, such as ablation experiments. Transformer-like models have full potential in hybrid modelling for solar energy prediction due to the intuitive interpretability of their framework. Furthermore, in future work, we propose to use the RNN framework in combination with the Transformer framework for Seq2sqe models with dynamic picture data streams as a framework to drive the current prediction framework.

Chapter 5

Rapid Deployment of Pre-trained Models for Climate across Domains using Transfer Learning

Chapter Abstract

A solution to the difficulties of forecasting solar energy was presented in this chapter using deep learning models by emphasising the importance of data diversity and the expenses involved in collecting data. Transfer learning for the DL-GSI-IHSF domain was proposed to adapt models to different climates and situations where training data is limited. We introduce two methods of adaptation - feature space adaptation and label adaptation - and confirm their effectiveness by calculating cosine similarity between latent representation vectors of multimodal features. The quantitative analysis based on Smart Persistence Model and F_1 score demon-

strated the feasibility of transfer learning in solar energy forecasting with stable model performance using just 4.5% of the dataset size and a 90% reduction in training time.

Contents

5.1 Introduction

There are still many difficulties and challenges in applying deep learning models in solar energy forecasting. Firstly, since solar irradiance information is highly correlated with climate, weather, and geographical factors, verifying or deploying models under different climate or geographical conditions requires recollecting data and retraining [4] The current DL-GSI-IHSF model is trained with the assumption of forecasting under the same climate and geographical conditions, so it cannot directly use well-trained models for forecasting in different locations [35, 55, 77, 50] Studies have shown that running a pre-trained model on a dataset of the exact specifications at another location results in significant errors. Furthermore, collecting new datasets is extremely costly [53]. Compared to traditional solar energy forecasting methods, deep learning-based approaches demand higher quality and larger quantities of datasets [31]. The generalisation performance of a model dramatically depends on the diversity of the dataset [101]. It is generally accepted that at least one year of data is required to ensure data diversity for training models in DL-GSI-IHSF. Thus, a considerable amount of time is needed to collect data when assessing the potential performance of a model locally. The diversity of datasets is crucial to the generalisation performance of a model, as it ensures that the model can handle various scenarios encountered in real-world applications. Training a model with a dataset lacking diversity can lead to sample bias, affecting the model's accuracy. Taking Nottingham as an example, clear-sky solar irradiance can reach up to $1200W/m^2$ in the summer and only $300W/m^2$ in the winter. Training a model with only a portion of incomplete data, which does not cover the full range of solar irradiance values, will likely result in overestimating or underestimating the final prediction. This highlights the importance of comprehensive data collection for solar energy forecasting models. Thus, costly data collection is an unavoidable process for developing new solar prediction models or deploying already proven solar prediction models in new regions. In summary, although deep learning models have potential in solar energy forecasting, practical applications have many difficulties and challenges, such as data diversity and the costs associated with data collection. Addressing these issues requires rethinking data collection methods and model training to improve models' generalisation performance and accuracy, ensuring that they can be effectively applied across various geographical locations and climates.

For incomplete or small datasets, transfer learning is a popular method in deep learning. Transfer learning improves the performance of a model in related sub-tasks by transferring the "knowledge" learned from a large dataset. For example, in the work of Zeiler et al.[155], the authors achieved impressive results by transferring the weights pre-trained on the ImageNet dataset to another natural image dataset with a small number of images. Compared to the retrained model, the pre-trained model with transfer weights increased the accuracy on the Caltech-256 dataset from 46.5% to 86.5%. Moreover, subsequent research found that the knowledge that a model can transfer is not limited to similar tasks. For instance, Shin et al.[115] discovered that transferring pre-trained weights from ImageNet to medical images for segmentation also improves model performance and reduces training time, even though medical images in the new task have completely different properties from natural images in ImageNet.

In this chapter, we proposed applying two different transfer learning methods, aiming to transfer the knowledge of a well-trained model on the LA Folsom dataset in the United States to the Nottingham dataset in the UK. The two datasets were collected at geographically different locations with different climates and slightly different equipment. The original contributions of this chapter are as follows:

1. We propose feature space adaptation and label adaptation methods in the DL-GSI-IHSF domain for training models using transfer learning between different climates. Moreover, we limit the original dataset based on various undersampling rates to simulate the transfer learning of models when the training dataset is insufficient.

2. By calculating the cosine similarity between the latent representation vectors of multimodal features, we directly verify the effectiveness of model transfer.

3. At a 2-minute-ahead prediction scale, we perform quantitative analysis on the model using the FS metric based on the Smart Persistence Model (SPM), as well as a quantitative comparison of the model's Ramp Event (RE) detection rate using the F_1 score based on RE detection.

4. Through cross-comparison, we confirm the feasibility of applying transfer learning in DL-GSI-IHSF. By using transfer learning methods, we can obtain stable model performance with only 5% of the dataset size and save 90% of the training time.

The remainder of the paper is structured as follows: Section 5.2 systematically compares the source dataset for transfer learning, the Folsom dataset, with the target dataset, the Nottingham dataset; Section 5.3 verifies the feasibility of transfer learning by compare the proposed model in Chapter 4 on the Nottingham dataset; section 5.4 presents the methodology of transfer learning, with the experimental setup transfer learning; Section 5.5 present the results of all experiment, Section 5.6 presents a discussion of the experimental results, and Section 5.7 concludes the whole Chapter.

5.2 Survey of Datasets

This section mainly compares the differences between the Folsom dataset and the Nottingham dataset. Although both datasets contain the same types of data, there are significant differences in the details due to different observation equipment, installation standards, and geographical locations. The comparison focuses on differences in meteorological features due to geographical location, similarities and differences in data observation instruments, and a comparison of image data features with the accuracy and distribution of meteorological data.

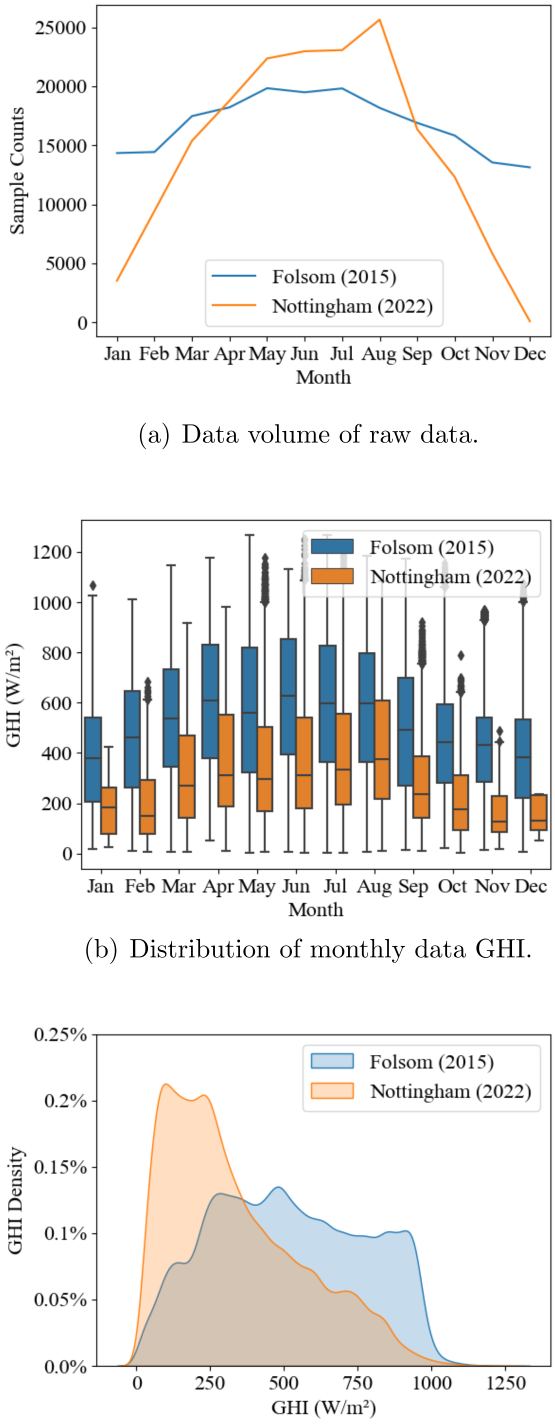
5.2.1 Meteorological Data

Table 5.1: Differences in meteorological data information between the data sets.

	Folsom	Nottingham
Longitude and latitude	38.642° N 121.148° W	52.952° N 1.184° W
Köppen climate classification	Csa	Cfb
GHI Measuring Instruments	LI-200SZ Pyranometers	Calculated
DNI Measuring Instruments	Calculated	RaZON+ PH1 Pyrheliometer
DHI Measuring Instruments	LI-200SZ Pyranometers	RaZON+ PR1 Pyranometer
Classification to ISO 9060:1990	˜±5% Typical error compare to First Class	Second Class
Data set size	656k	96k
Duration of data set collection	3 Years	6 Months
Train/val/test set size	21K/25K/23K	58K/19K/19K

Differences in geographical locations of the observation stations directly lead to differences in the sample distribution of the observed datasets. Table 5.1 shows the geographical environmental conditions of the two dataset collection sites. Taking the Folsom dataset from 2015 and the Nottingham dataset from 2022 as examples, as shown in Figure 5.1 (a), the amount of data collected per month differs significantly due to the impact of latitude differences on daylight duration. The UK dataset has longer summers and shorter winters. Additionally, the limitation of the fisheye lens's effective viewing angle (SZA less than or equal to 75 degrees) exacerbates this quantity difference. For example, in Nottingham, the solar zenith angle rarely exceeds 75 degrees throughout December. Therefore, after quality control screening, only 87 samples remain for the entire month. Furthermore, the Folsom data station belongs to the Csa type (C = temperate climate, s = dry summer, a = hot summer) in the Köppen climate classification, while the Nottingham data station belongs to the Cfb type (C = temperate climate, f = no dry season, b = warm summer). These climatic differences are reflected in the datasets as shown in Figures 5.1 (b) and (c). In Figure 5.1 (b), Folsom's irradiance distribution is generally higher than Nottingham's on a monthly basis. Even during the summer months when daylight duration and solar angles in Folsom are slightly lower than those in Nottingham, the dry and hot climate conditions with fewer clouds result in higher average values despite lower extremes. Looking at the whole year, as shown in Figure 5.1 (c), GHI density distribution in Folsom is relatively even, while Nottingham has a significantly higher density of low irradiance values due to the wet and cloudy climate. The two-dimensional visual heatmaps also shows the same trend. In Figure 5.2, the data for Nottingham lacks continuity throughout the year, especially for the DNI data. This means that direct sunlight is being disturbed by cloud cover at frequent intervals. In contrast, Folsom's dry summer idiosyncrasies allow for sufficient and continuous sunlight in the summer. For example, there was little blockage throughout the summer of 2016 in Folsom dataset.

In addition to data distribution, the equipment used to collect solar irradiance and meteorological data at the two sites is also different. According to [39], the Folsom site uses two LI-200SZ Pyranometers [156] to collect GHI and DHI data, and then calculates the difference and solar angle to derive DNI. In contrast, at the Nottingham site, as mentioned earlier, the Razon+ automatic solar tracker [157] collects DHI and GHI data, and GHI values are calculated based on the solar angle sum. It is worth noting that the pyranometers used at the two observation sites have significant differences in accuracy. As reported in [158], the LI-200SZ, as a photodiode pyranometer, has a GHI measurement error of 4.4% under clear sky conditions, which is consistent with the "typical error of about 5% according to ISO 9060:2018 First Class" stated in [39]. On the other hand, PR1 and PH1, as thermopile pyranometers and pyrheliometers, have clear sky GHI errors of 0.3% and 0.03%, respectively [157]. According to ISO 9060:2018 [159], they can be classified as Second Class, with typical errors less than 1% different from First Class. Moreover, the authors point out that although the accuracy of the LI-200SZ can be improved through calibration, its error remains an order of magnitude higher than that of thermopile pyranometers due to its limited accuracy outside the low-error field of view (60 degrees).

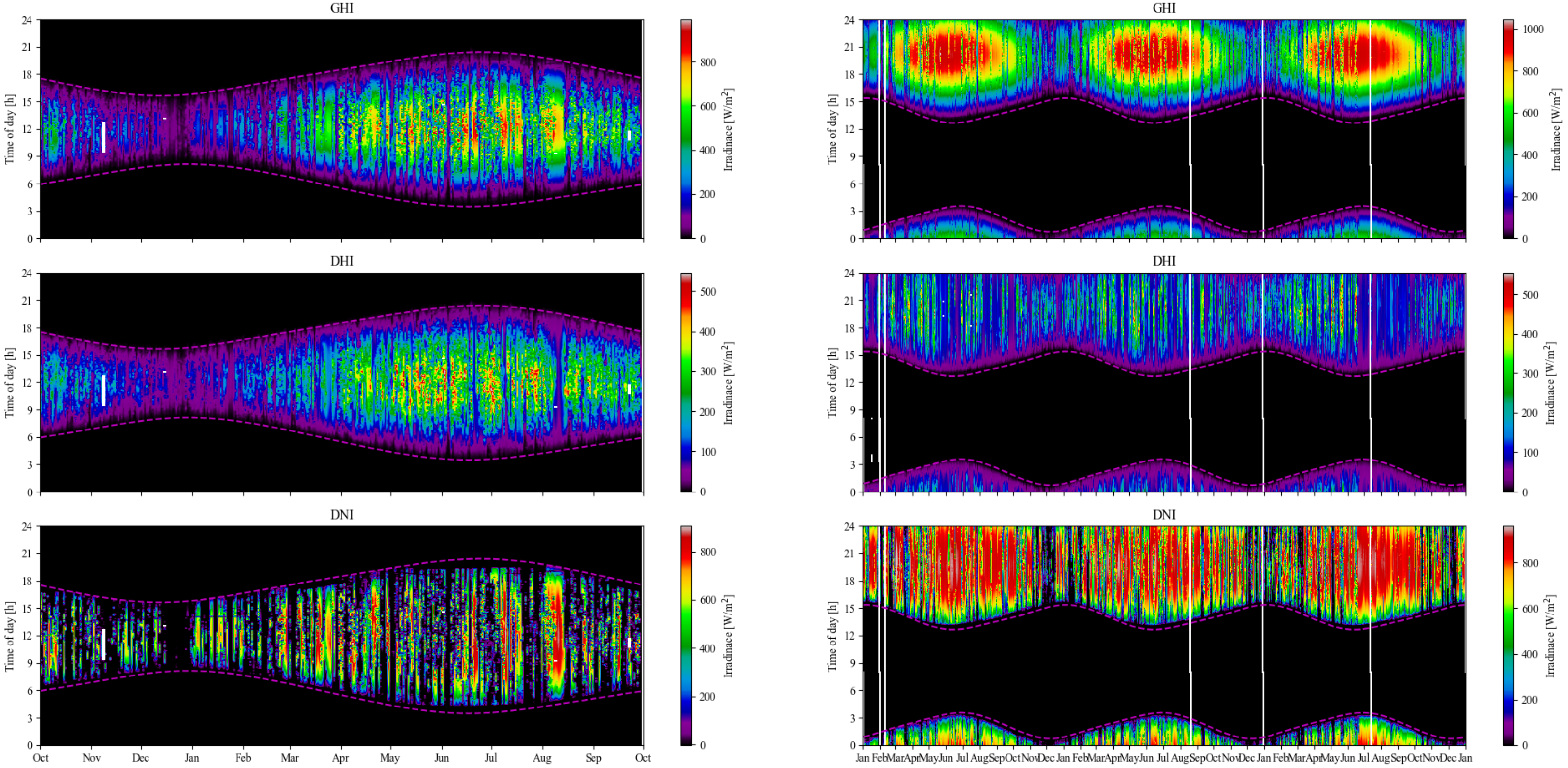


(a) Data volume of raw data.

(b) Distribution of monthly data GHI.

(c) Density distribution of annual GHI data

Figure 5.1: Comparison of sample size and distribution between the Folsom dataset and the Nottingham dataset, for 2015 and 2022, respectively.



(a) Visual heatmaps of Nottingham dataset, from 2021 Oct. to 2022 Sep.

(b) Visual heatmaps of Folsom dataset, from 2014 to 2016.

Figure 5.2: Two-dimensional visual heatmaps of Folsom and Nottingham dataset.

5.2.2 Image Data

In this section, we compare the Folsom and Nottingham datasets, particularly focusing on the differences and similarities in the image samples. As shown in Table 5.2, the cameras used to record images in the two datasets are not identical. The Nottingham dataset camera has superior performance, with higher original pixel resolution, a larger aperture, and better light sensitivity. It is worth discussing the noticeable differences in the actual image output quality, as illustrated in Figure 5.3.Firstly, for the Nottingham dataset, the intense sunlight leads to overexposure in the solar region, resulting in a lack of information around this area. In contrast, the Folsom dataset has the fully overexposed (RGB values all at 255) solar region blackened. As a result, the information on whether the sun is directly visible can be discerned in the Folsom dataset. For instance, as shown in the Cloudy Sky image on the Figure 5.3 right, the Folsom dataset preserves the position and visibility of the sun shining through thin clouds, while this information is entirely unknown in the Nottingham dataset. We once attempted to determine the solar azimuth in the Nottingham dataset using the same method, but due to camera quality limitations and an extensive overexposure area, it proved to be impractical. Additionally, another noteworthy point is the prominent image noise in the Folsom dataset, as

Table 5.2: Differences in image data between the data sets.

	Folsom	Nottingham
Camera model	Vivotek FE8171V	MOBOTIX Q25
Sensor size	1/2” CMOS	1/1.8” CMOS
Original Resolution	2480 × 1536	3072 × 2048
Aperture size	f/2.8	f/2.0
Light sensitivity	1.17 Lux	0.1 Lux
Output Resolution	1536 × 1536	1028 × 1028
Orientation	15 degrees west of north	Due West
Sun Marker	Yes	No

demonstrated in the bottom-left image. We speculate that this might be due to the wear and tear of the transparent protective shell. The image clearly shows noise created by the refraction of sunlight on the protective shell, as well as lens flare reflections formed in the centre of the image due to the shell's imperfect transparency. This phenomenon is particularly noticeable on sunny days.

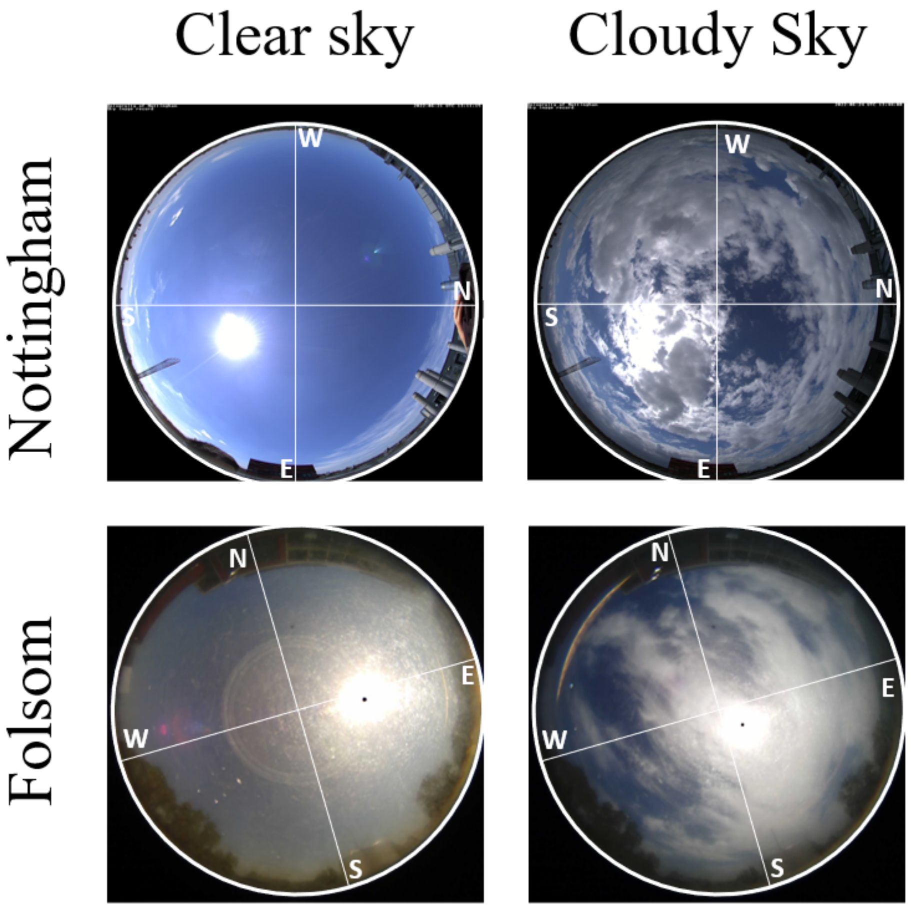


Figure 5.3: Schematic of the sky images for the two data sets under different weather conditions.

In summary, the differences between the two datasets are mainly attributed to the geographical location and the observation equipment used. The geographical location leads to differences in meteorological features, while the observation equipment contributes to the variations in data accuracy and distribution. Understanding these differences is essential when applying transfer learning or comparing the performance of models across different datasets. As such, it is crucial to consider the impact of these factors when developing and deploying solar prediction models in various regions or under different conditions.

5.3 Feasibility Study

This section shows the feasibility experiments conducted before the transfer experiments. We believe that a prerequisite for the transfer experiment is that the ViT-E model maintains the architectural superiority described in Chapter 5 when applied to the Nottingham dataset. Therefore, we repeated the experiments from Chapter 5 using the Nottingham dataset without changing the training mode, optimisation strategy, loss function, or any other details. The only change was to emphasise the model degradation trend observed in Chapter 5 by reducing the forecast resolution from 4 to 2 minutes. The advance forecast was changed from 2-, 6-, and 10-minute forecasts to 2-, 4-, 6-, 8-, and 10-minute forecasts. It is worth noting that since the loss function of the training model is the mean square error, the model's response to RE is influenced by the dataset itself and is not constrained by the loss function. Therefore, the model's BP performance, as exhibited by dataset characteristics, was not demonstrated in the feasibility study. Figure 5.4 shows the experimental results of the predictions. As shown in the figure, the overall trend of the model's performance on the Nottingham dataset regarding FS on the test set is similar to that on the Folsom dataset. Specifically, model differences diminish as the prediction time increases. However, unlike the Folsom dataset, the CNN-LG architecture on the Nottingham dataset did not exhibit a comparable

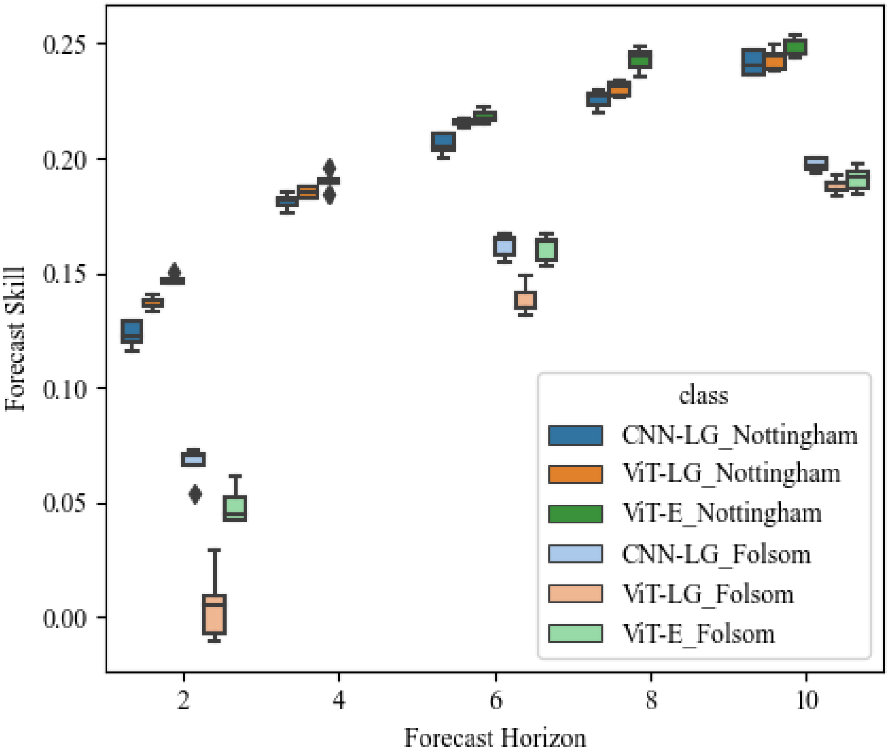


Figure 5.4: FS results for feasibility studies on both dataset.

advantage in FS; rather, it performed worse than the ViT-LG and ViT-E models. We speculate this may be due to the dataset differences caused by climatic factors. The warmer, humid climate leads to a much higher frequency of cloudy weather in Nottingham than in Folsom. Although, based on the method mentioned in Section 4.2, continuous clear-sky samples were removed from both datasets to balance the number of samples, the Nottingham dataset still contains many more RE samples than the Folsom dataset. In the Nottingham dataset validation set, RE samples account for 16.69% of the total samples (3210 out of 19237), while in the Folsom dataset, RE samples account for only 9.3% (2197 out of 23583). We believe that such climatic differences make it difficult for the CNN-LG framework, which relies on numerical inputs, to maintain an advantage in FS through numerical inference. Meanwhile, the climatic differences also contribute to the variation in FS performance between the two datasets. The SPM model performs worse in cloudy and rapidly changing weather, leading to a higher FS, calculated based on the SPM model. For example, the 2-minute RMSE error of the SPM model in the Folsom dataset is $85.62W/m^2$, while it is $117.89W/m^2$ in the Nottingham dataset.

In summary, we believe that the ViT-E model, based on early modal interaction, demonstrates similar advantages in the feasibility study as in Chapter 5, as it can effectively balance image and numerical inputs. Moreover, the Nottingham dataset exhibits more serious prediction difficulty and spatial complexity than the Folsom dataset. The ViT-E model demonstrates superior quantitative performance from the outset when processing more complex datasets. Furthermore, the trend of diminishing model differences with increasing prediction time is again confirmed.

5.4 Methodology

In this section, we mainly introduce the transfer learning methods used in the following sections, the experimental deployment, and the evaluation metrics.

5.4.1 Definition

Domain A domain $\mathcal{D}$ consists of two parts: the feature space $\mathcal{X}$ and a marginal distribution $\mathbf{X}$, where $\mathbf{X}$ represents a set of instances, such as the input $\mathbf{X}$ in Equation 2.13. It can be expressed as:

$$\mathcal{D} = \{\mathcal{X}, P(\mathbf{X})\} \tag{5.1}$$

$$\text{where } \mathbf{X} = \{\mathbf{x} | \mathbf{x}_i \in \mathcal{X}, i = 1, \ldots, n\} \tag{5.2}$$

Task A task, $\mathcal{T}$, is composed of a label space $\mathcal{Y}$ and a decision function F. In this paper, the labels y in $\mathcal{Y}$ are the results of the elements $\mathbf{x}$ in $\mathcal{X}$ passed through the decision function F, which is a specific deep model regression, i.e.

$$\mathcal{T} = \{\mathcal{Y}, F\} \tag{5.3}$$

$$\text{where } F(\mathbf{x}_{\text{j}}) = \{E(y_k | \mathbf{x}_j) | \mathbf{y}_k \in \mathcal{Y}, k = 1, \ldots, |\mathcal{Y}|\} \tag{5.4}$$

In this chapter, in deep model based on a task for regression, $y_{t+\Delta t}$ is the only element in the target space $\mathcal{Y}$ and the only expectation when the input $\mathbf{x}_j$ is given. In the multitask learning mentioned in Chapter 2, the target space can contain multiple different targets $\mathcal{Y}$. Also, if the prediction is based on probability prediction rather than regression, the mapping relationship of F should be $P(y_k | \mathbf{x}_j)$.

Transfer Learning Transfer learning refers to the process of improving the performance of the decision function, i.e., the deep model, F_T, in the target domain by leveraging the implicit knowledge in the source domain for a given specific source domain and source task $(\mathcal{D}_S, \mathcal{T}_S)$, and target domain and target task $(\mathcal{D}_T, \mathcal{T}_T)$.

5.4.2 Method of Transfer Learning in DL-GSI-IHSF

In this chapter, we employ the Inductive Transfer Learning method [160], which specifically means that the label in source and target domains are available. Meanwhile, in DL-GSI-IHSF, based on the categorisation of the consistency between the source and the target feature spaces and label spaces. The transfer learning method can be considered as homogeneous transfer learning, which represent consistency in source and target domains, i.e., $\mathcal{X}^S = \mathcal{X}^T$ and $\mathcal{Y}^S = \mathcal{Y}^T$. Please note that the prerequisite here is for the prediction target to be CSI, and it is assumed that the Clear Sky Index calculation can completely eliminate the effects of irradiance cycles and geographical factors on solar irradiance through normalisation. If the prediction target is Irradiance itself, due to meteorological cycles and geographical influences, even with identical input conditions, the prediction outputs would be different, i.e., $\mathcal{Y}^S \neq \mathcal{Y}^S$. Specifically, as shown in Figure 5.11,

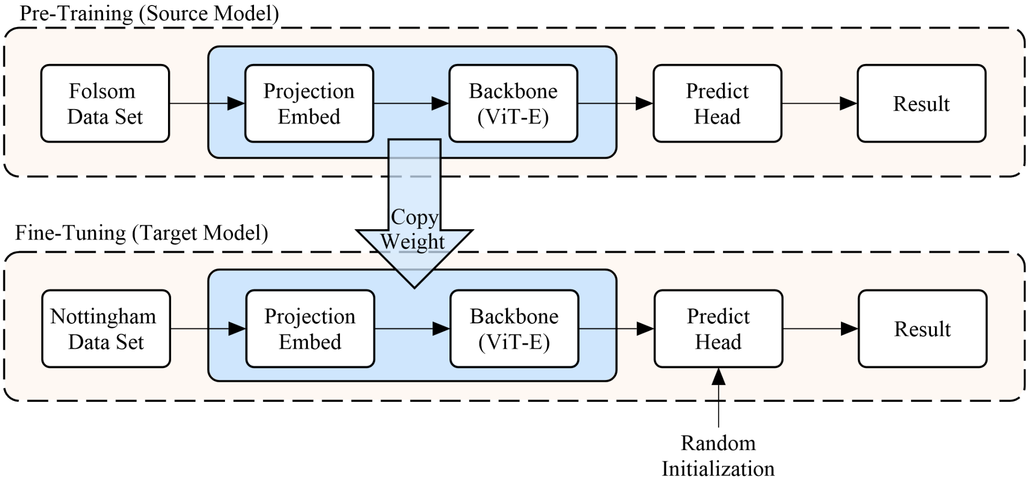


Figure 5.5: Schematic diagram with transfer learning process.

transfer learning is achieved through weight transfer and fine-tuning. The model first undergoes sufficient pre-training on the source domain, i.e., the Folsom dataset, using the ViT-E architecture. After completing the pre-training process, the weights obtained from the model training are saved.

Next, the model weights are loaded into the model before training on the target domain, i.e., the Nottingham dataset, to achieve knowledge transfer. It is worth noting that the transferred weights only include the projection embedding and backbone layer, while the predict head for each training is randomly initialised. Finally, the model completes the entire training process on the Nottingham dataset with the pre-trained weights as the starting point. This retraining process on the new model is also called fine-tuning.

5.4.3 Experiment Setup

Experiment 1: Exploring the Effectiveness of Transfer Learning First, we validated the effectiveness of transfer learning. Here, effectiveness is defined as the migrated weights not being fully iterated away during the model weight search. Effectiveness can be obtained by cosine similarity. Cosine similarity is a method for measuring the difference between two vector individuals in the feature space; if the directions of the two vectors are consistent, i.e., the angle between them is close to zero, the two vectors are more similar. If transfer learning is effective, the model weights fine-tuned based on one pre-training weight should have the highest similarity to the pre-training weights themselves. The formula of Cosine similarity expressed as

$$\cos\Theta = \frac{\mathbf{a} * \mathbf{b}}{|\mathbf{a}| * |\mathbf{b}|} \tag{5.5}$$

We use the output vector of the backbone network, which represents the multimodal semantic representation vector in the model, i.e., $\mathbf{y}$ in Equation 2.5, as the standard vector. Through repeated experiments, we obtained five sets of pre-trained weights. These weights are identical in the training process, with only differences caused by randomness. First, we compared the similarity of $\mathbf{y}$ in the five pre-trained models and the similarity of **y** in the target models trained using the weights of the five different pre-trained models with the source model **y**. Finally, the effectiveness of model transfer learning is validated through cross-comparison of model similarities.

Experiment 2: Effect of Pre-training Randomness on Transfer Learning. As discussed in Chapter 2, due to the inherent randomness in deep learning, the pre-trained weights used for transfer learning themselves contain a certain degree of uncertainty and variability. To verify the impact of randomness-induced differences on transfer learning in Experiment 1, we collected the time required to fit models based on five pre-trained weights and the performance of the models for comparison.

Experiment 3: Applying Different Target Domain Adaptation Strategies During the fine-tuning process of the model in the target domain, there are various methods to choose from [161]. In this chapter, we conducted experiments using two basic methods. The first method is feature space adaptation. Specifically, based on the label pairing in the target task (i.e., $(\mathbf{x}_T, y_T)$), the transferred weights are further trained to adapt the inference part of the model to the new task $\mathcal{T}_T$. Meanwhile, to prevent the model from completely changing the pre-trained weights during fine-tuning, the learning rate is set to one-tenth of the original value. The second method is label space adaptation. This method directly uses the pre-trained weights without any modification. By freezing the inference layer during the training process, the feature extraction and inference methods from the source domain are used to obtain the representation vector. During the training process, only the inference head, which is responsible for mapping the representation vector to the prediction result, is trained. Moreover, since there are two parts to the inference module for the ViT-E model, the projection embedding and backbone interface layer (i.e., Transformer Encoder), we tested freezing different parts separately for a detailed comparison.

Experiment 4: Training Models using a Limited Target Domain Dataset As mentioned earlier, one potential advantage of transfer learning is that when the size of the source domain dataset is larger than that of the target domain dataset, the transferred knowledge can still achieve better generalisation in the target domain with a limited dataset. To verify this, we further reduced the size of the dataset. Specifically, based on the dataset sensitivity validation in Chapter 3 and report in work of Paletta et al. [42], we consider that in a complete one-year dataset, the trend of improving the model by enriching the dataset size begins to slow down starting from a dataset size of 25K samples. In other words, the model's generalisation reaches a bottleneck at 25K samples, making it improve continues more difficult. However, due to limitations in data collection and data quality, the Nottingham dataset actually only contains six months of data, so we still used the entire dataset for fitting experiments. Overall, we started with the entire Nottingham dataset (a total of 55K samples), randomly downsampled the dataset size to 25K, and further randomly downsampled to 12.5K, 7.5K, 5K, and 2.5K. We compared transfer learning with learning from scratch, testing the fitting performance of transfer learning in datasets with insufficient sample sizes.

Experiment 5: Modelling Finite Datasets in the Real World In the setting of Experiment 4, random downsampling was used to obtain a limited dataset. This approach has a limitation in that random sampling preserves the diversity of the dataset to the maximum extent, thereby enabling the model to achieve better performance. Therefore, we used another downsampling method, namely, random continuous sampling. We randomly extracted three consecutive 14-day subsets from the original dataset as the dataset, simulating whether the model can achieve training with extremely limited dataset sizes through transfer learning in real situations. Specifically, we used data segments starting from March 4, June 7, and September 9, 2022, as training sets, without applying clear sky filters to the data. Ultimately, the three datasets contained 6.4K, 9.6K, and 6.8K data points, respectively, as shown in Figure 5.6.

5.4.4 Summary

This section mainly introduces the methodology of the experiments in this chapter. First, we defined the specific concept of transfer learning and provided a formula explanation. Then, we explained the implementation of the transfer learning method in this paper. Finally, we designed step-by-step experiments for the parts we were interested in. The specific experimental design is shown in Table 1. In Experiments 1 and 2, we compared the effectiveness and results of using different random weights in transfer learning and used the best-performing model in Experiments 1 and 2, i.e., Model #3, as the pre-trained weights for all remaining models. In Experiments 3 and 4, we cross-compared the effects of freezing different modules at all downsampling ratios. In Experiment 5, we further integrated the two best-performing patterns from Experiments 3 and 4 for live simulation experiments. It is worth noting that the transfer experiments in this chapter were conducted on 2-minute ahead forecasts only. Here we use a conclusion from Chapter 4 that the superior performance of the model is consistent across forecasting scales and that 2-minute ahead forecast under 2-minute forecast resolution is the forecasting horizon with the greatest model variability. In addition, the hyperparameters used in this chapter are fully inherited from those shown in Chapter 4.

	Source Model	Target Dataset Size	Frozen Layers
Exp. 1 & 2	5 Trained Models	55k	Unfrozen layer
Exp. 3 & 4	#3 Model in Exp. 1	55k to 2.5k	Unfrozen layer Freeze projection layer Freeze inference layer Freeze all layer
Exp. 5	#3 Model in Exp. 1	6.4k (Mar 4 to 18)	Unfrozen layer Freeze all layer
	#3 Model in Exp. 1	9.6k (Jun 7 to 21)	Unfrozen layer Freeze all layer
	#3 Model in Exp. 1	6.8k (Sep 9 to 23)	Unfrozen layer Freeze all layer

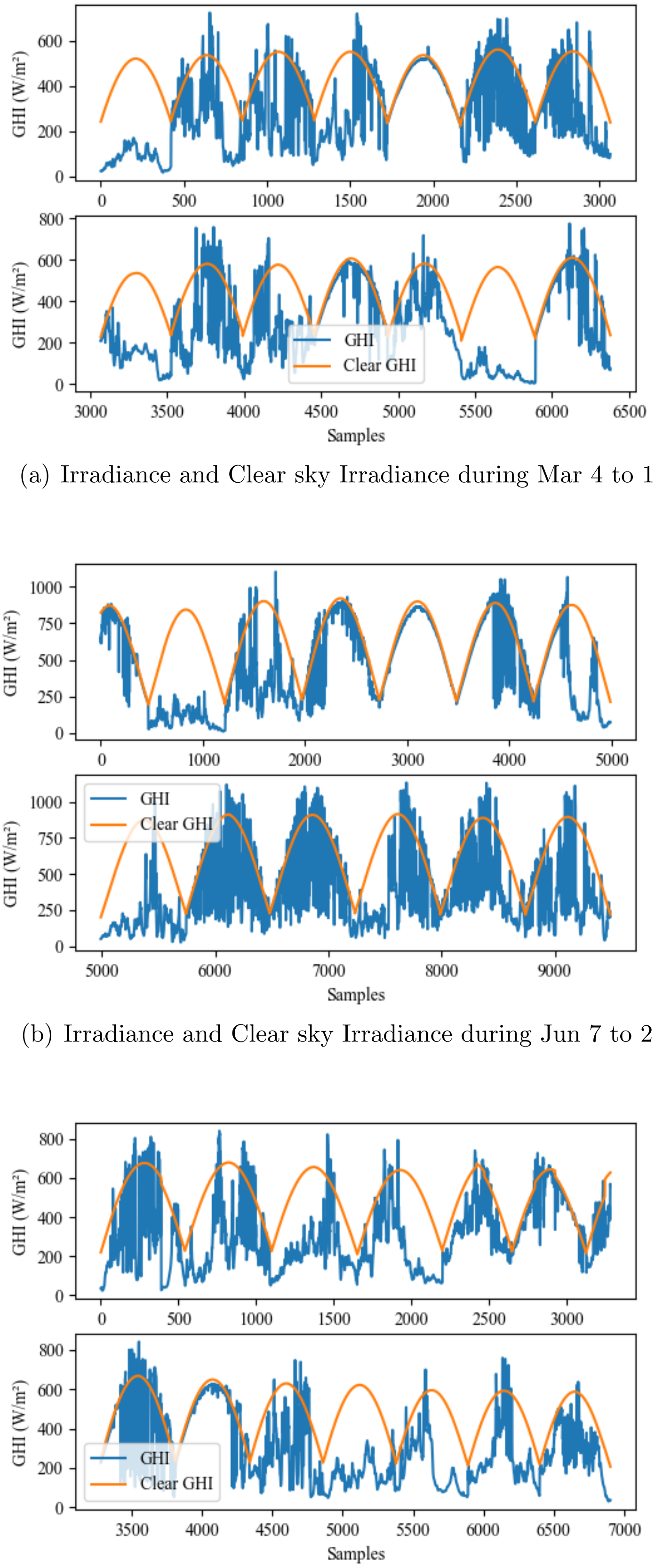
5.5 Result

Modelling was undertaken using a PC with a 3.8 GHz AMD Ryzen 9 3900X CPU and a GeForce RTX 2080 SUPER GPU on the Tensorflow 2.10 [153] platform with Keras [154] built in. To reduce errors introduced by random nature in modelling, including the randomness in observation order and the randomness in random number generator in training, five replicate trials were carried out for each image model.

5.5.1 Experiment 1: Effectiveness of Transfer Learning

The results of the model effectiveness verification are shown in Figure 5.7. The cosine similarity is calculated based on the angle between feature vectors, so the closer the result is to 1, the higher the similarity. Figure 5.7

(a) shows the cosine similarity between 5 source domain models, #1 to #5. It can be observed that despite the inevitable randomness in the training process, the impact of the model's randomness on the final trained model is minimal. All models exhibit consistency in the latent semantic space. Figure 5.7 (b) shows the similarity between the target domain models and the source domain models after further training on the target domain dataset using the five source domain models. As can be seen from the figure, on the one hand, the target domain models after transfer learning are mostly more similar to their corresponding source domain models. On the other hand, the diversity trend between source domain models is preserved after transfer learning. For example, the high similarity between $\mathbf{y}^S_{\#2}$ and $\mathbf{y}^S_{\#4}$ is consistent with the similarity between the post-transfer learning $\mathbf{y}^T_{\#2}$ and $\mathbf{y}^T_{\#4}$.

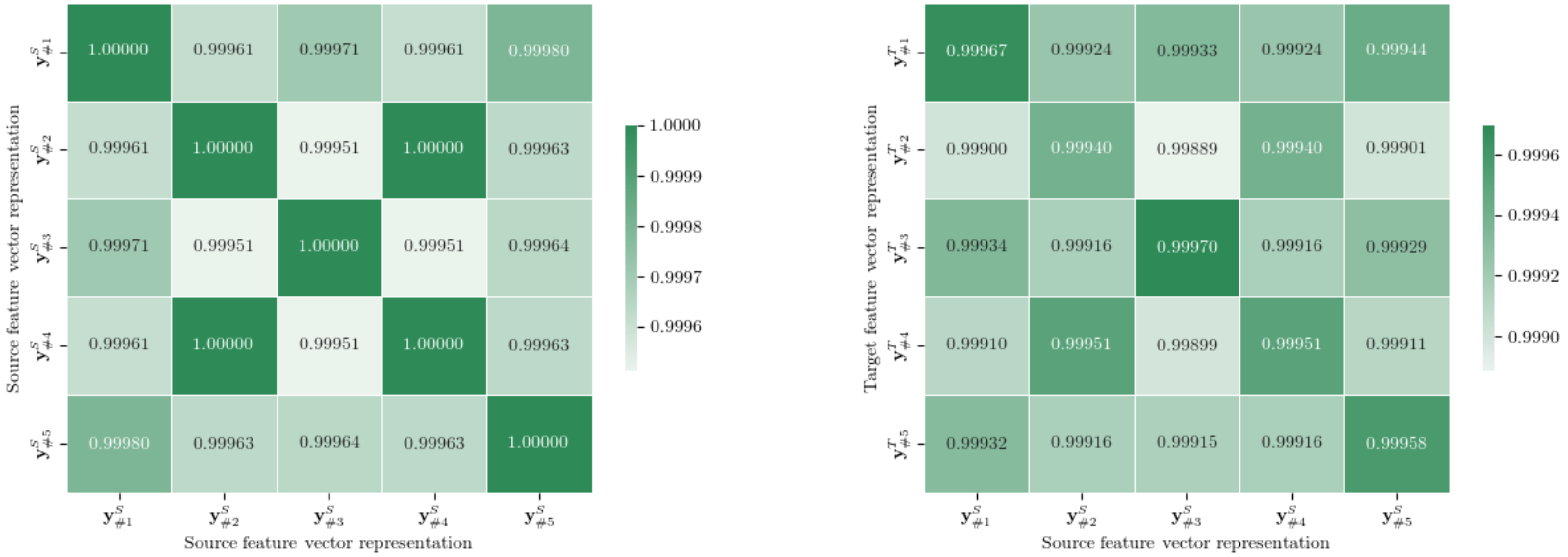


(a) Irradiance and Clear sky Irradiance during Mar 4 to 18.

(b) Irradiance and Clear sky Irradiance during Jun 7 to 21.

(c) Irradiance and Clear sky Irradiance during Sep 9 to 23.

Figure 5.6: Three consecutive two-week datasets of the downsampled dataset in the mock-up experiment.



(a) Cosine similarity between five source domain models.

(b) Cosine similarity between source domain model and target domain model.

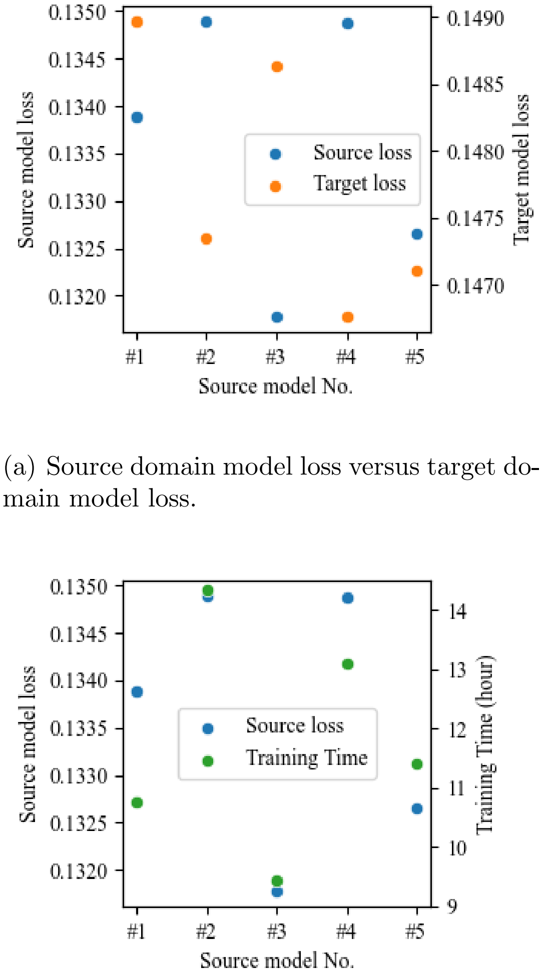
Figure 5.7: Results on the validity of transfer learning based on cosine similarity. Note that the actual value of $\mathbf{y}^S_{\#2}$ and $\mathbf{y}^S_{\#4}$ similarity in figure (a) is 0.9999998987, limited by the progress display, which shows 1.00000.

5.5.2 Experiment 2: Effect of Pre-training Randomness on Transfer Learning

Figure 5.8 illustrates the influence of performance gaps, resulting from the randomness of 5 distinct source domain models, on the target domain models during transfer learning. Figure 5.8 (a) presents a comparison between the performance of source and target domain models. As evident in the figure, the impact of randomness on model performance is comparable across different domains, with an approximate error of 2%. Furthermore, the source and target domain models exhibit no consistency in performance. For instance, the top-performing model #3 in the source domain lags after transfer learning, whereas the underperforming model #4 in the source domain excels in the target domain. Figure 5.8 (b) highlights the influence of source domain model performance on transfer learning duration. Notably, compared to target domain model performance, the transfer training duration exhibits a stronger correlation with source domain model performance. The top and bottom-performing models #3 and #2 in the source domain demonstrate a significant difference in retraining time. Despite model #3 in the source domain having a performance gap of 2.3% compared to model #2, its transfer learning time is reduced by one-third.

5.5.3 Experiment 3 & 4: Influence of Freezing Modules and Limited Datasets in Transfer Learning

Table 5.3 and Figure 5.9 show the performance of models in terms of FS under various dataset sizes and transfer learning strategies. As the figures indicate, on one hand, from the perspective of dataset dimensions, the FS of the models decreases as the number of samples in the dataset diminishes.



(a) Source domain model loss versus target domain model loss.

(b) Source domain model loss versus transfer training time.

Figure 5.8: comparison of source domain model performance in terms of target domain model performance and training efficiency.

The most prominent characteristic is that the approach of training new models from scratch without utilising transfer learning methods becomes highly unstable and struggles to maintain model accuracy when data is limited. When the sample size drops to 7.5k, models trained from scratch exhibit a significantly lower FS compared to transfer learning, with some instances displaying a substantial decline in FS. As the sample size further decreases to 5k, models trained from scratch start to experience a noticeable decline in FS.

On the other hand, even when the sample size is reduced to 2.5k, the four transfer learning methods still manage to maintain relatively high performance levels despite the overall decrease in model performance. Notably, the method that freezes all layers during training on 2.5k samples even surpasses the performance of CNN-LG trained on the entire dataset. Comparing different transfer learning methods, employing Feature Space Adaptation approaches, such as not freezing any layers or only freezing the projection layer while leaving the core inference layer unfrozen, can achieve performance levels similar to training from scratch when sufficient data is available. In contrast, methods that lock the core inference layer exhibit relatively poorer performance when data is abundant. However, an exception occurs with the 2.5k dataset, in which the two methods that fully inherit the source domain model's core inference layer achieve the best performance.

Table 5.3: Forecast skill for different transfer learning strategies and dataset sizes.

Dataset Size	Forecast Skill (%)				
	New Training	Unfrozen layer	Freeze inference layer	Freeze projection layer	Freeze all layer
2.5k	10.24±2.44	12.85±0.33	13.12±0.52	12.96±0.25	13.26±0.34
5k	11.64±2.34	13.47±0.25	12.97±0.52	13.62±0.23	12.94±0.49
7.5k	12.69±1.09	13.54±0.24	13.38±0.30	13.53±0.42	13.22±0.37
12.5k	14.07±0.45	13.48±0.61	13.87±0.13	13.77±0.65	13.88±0.19
25k	14.42±0.31	14.41±0.10	14.16±0.60	14.38±0.33	14.21±0.46
55k	16.15±0.50	16.20±0.52	15.66±0.35	16.04±0.52	14.51±0.40

Table 5.4 and Figure 5.10 display the F_1 Score[1] of the models in forecasting RE. As the figures demonstrate, the F_1 Score, which is derived entirely from the regression forecast values without explicit loss function

[1] In practice, we found that the recall rate of the models also exhibited significant fluctuations on the Nottingham dataset, characterised by frequent abrupt changes in sky conditions. Consequently, we employed the balanced F_1 Score (Eq. 2.43) as a comprehensive measure, replacing the balanced precision (Eq. 2.42) used in Chapter 4.

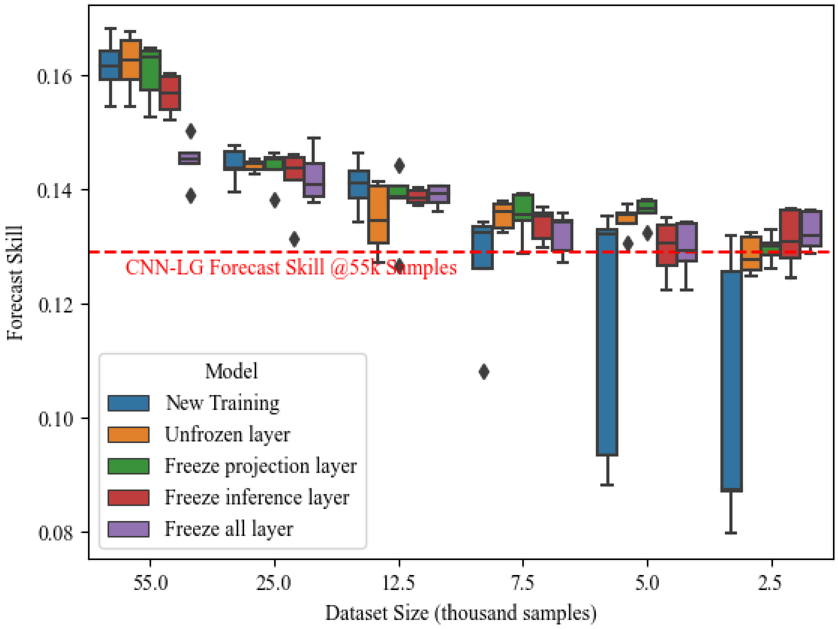


Figure 5.9: Forecast skill for different dataset sizes and transfer learning approaches.

constraints, is not noticeably affected by the dataset size. Apart from achieving better performance on the complete dataset, the models exhibit comparable results across all downsampled datasets. Interestingly, if all layers are completely frozen and only the prediction head is adapted, the F1 Score performance on the 55k and 2.5k datasets is virtually identical.

Overall, the two models trained entirely on independent datasets, i.e., those trained exclusively on the new dataset and those trained on the source dataset with all layers locked on the new dataset, exhibit relatively superior performance. In contrast, models that combine both datasets for feature adaptation perform less optimally.

Figure 5.11 illustrates the time consumed in training the models. Overall, training models with less data can reduce the training time; however, this also implies a loss in model performance. On the other hand, transfer learning methods that freeze the inference core can significantly save train-

Table 5.4: F_1 score for different transfer learning strategies and dataset sizes.

Dataset Size	F_1 Score (%)				
	New Training	Unfrozen layer	Freeze inference layer	Freeze projection layer	Freeze all layer
2.5k	35.85±0.84	35.14±1.01	33.60±0.28	34.11±1.90	36.36±0.45
5k	35.05±0.80	34.58±1.57	34.65±1.21	34.78±0.91	35.22±1.39
7.5k	35.94±2.00	34.94±1.17	34.45±1.12	33.81±2.16	35.02±0.73
12.5k	36.18±2.04	35.21±0.94	34.22±2.15	34.78±1.52	35.75±1.35
25k	35.04±1.30	35.59±2.00	34.78±0.47	35.94±1.61	35.82±0.97
55k	38.87±1.43	39.10±2.00	38.57±1.55	37.24±2.19	36.54±0.95

ing time when a larger dataset is. With 55k samples, freezing the inference layer can reduce training time by approximately 10%, while freezing both the inference and projection layers can save around 40% of the training time. When the dataset is smaller than 12.5k, all transfer learning methods shorten the training time. At 2.5k samples, utilising transfer learning methods takes less than one hour, compared to the three hours required for training from scratch, reducing the training time cost by over 60%.

Another intriguing observation is that continuing to train the inference module with feature adaptation methods demands more training time on the original dataset compared to training from scratch.

5.5.4 Experiment 5: Modelling Finite Datasets in the Real World

Figure 5.12 shows the training results under the simulated conditions. We also used the 7.5K-sized dataset from Experiment 4 to facilitate the comparison of model performance. The figure shows that the models trained from scratch using the three simulated datasets did not achieve stable results. Among them, the performance of the model trained with the two-

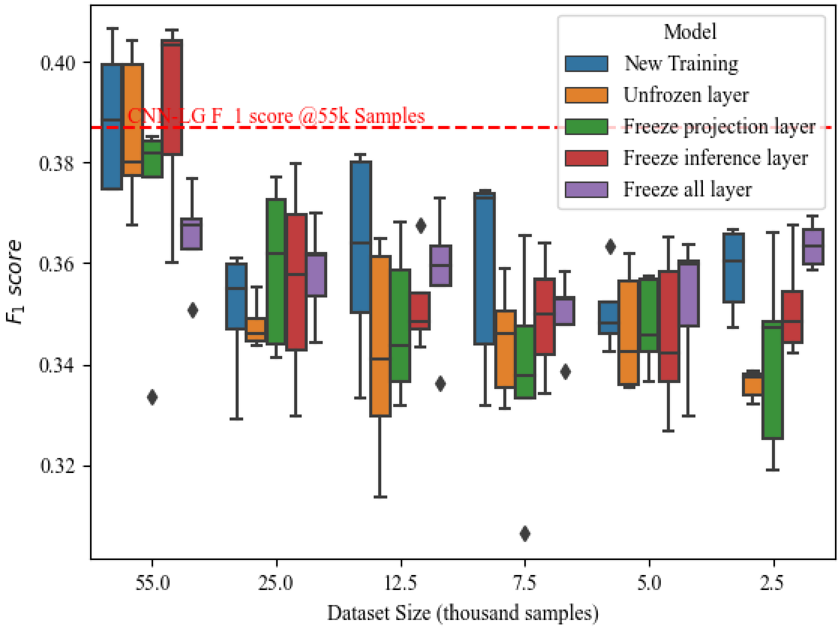


Figure 5.10: F_1 Score for different dataset sizes and transfer learning approaches.

week dataset from March is only comparable to that of the SPM model, about 4% FS. In the models obtained through transfer learning, the two-week dataset from June achieves results close to those obtained with the full-year downsampled data. It is worth noting that due to the duration of sunlight, the June data contains about 30% more data points than the March and September data, reaching 9.6K. In addition, the solar path also influences the diversity of the images. The March data's minimum solar zenith angle (SZA) is as high as 54 degrees, while the minimum SZA for June is 29 degrees and 47.7 degrees for September. Smaller SZA values mean that the sun is closer to the centre of the image, which implies that the diversity of sun positions in the sky image data from March is limited.

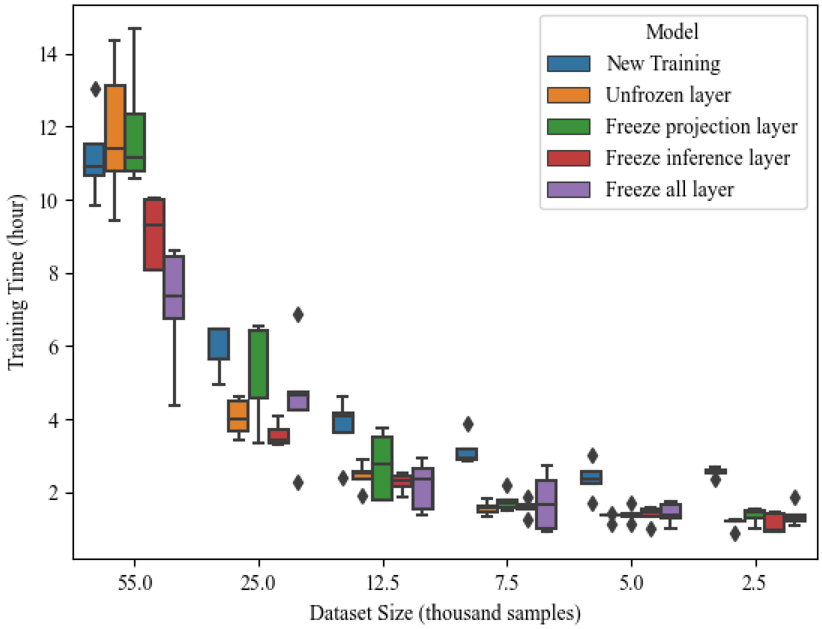


Figure 5.11: Training Time for different dataset sizes and transfer learning approaches

5.6 Discussion

In this discussion, we have integrated the findings from three sections to provide a comprehensive analysis of transfer learning in the context of DL-GSI-IHSF. Our study offers valuable insights into the impact of transfer learning on model performance, training time, and data requirements and highlights potential avenues for future research.

Our results show that transfer learning can effectively improve the performance of the ViT-E model in the DL-GSI-IHSF domain, mainly when the available training data is limited. Figures 5.7 and 5.8 demonstrate that some features of the source domain model are preserved during the transfer learning process, while Figure 5.9 and 5.10 emphasise the superiority of transfer learning when working with limited datasets.

As shown in Figure 5.13 with the horizontal axis representing training time

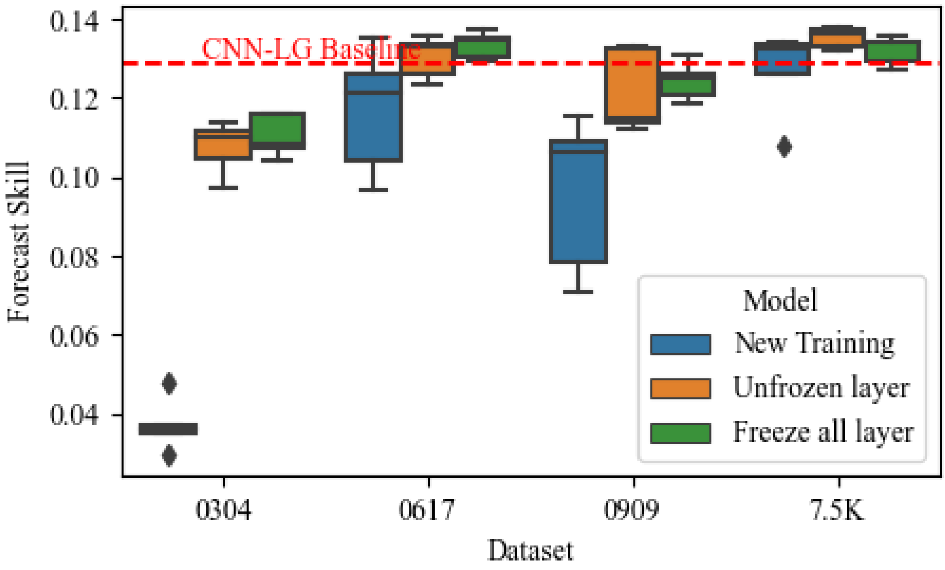


Figure 5.12: The model's results trained by collecting data for two consecutive weeks are compared with a full year of data downsampled to 7.5K data.

and the vertical axis representing model accuracy, further highlights the time efficiency of transfer learning. It demonstrates that transfer learning can achieve stable and relatively superior performance with only 2.5k samples (4.5% of the total dataset) and 10% of the training time compared to traditional learning methods.

We can optimise model performance and training time by providing the target domain model with a better starting point and utilising different transfer learning strategies. However, the performance consistency between source and target domain models remains an open question that warrants further investigation, as illustrated in Figure 5.8.

Future research should explore more complex transfer learning mechanisms, such as shared representations or alignment methods, which could constrain the target domain model during training and improve the consistency between the source and target domain models in terms of performance. More-

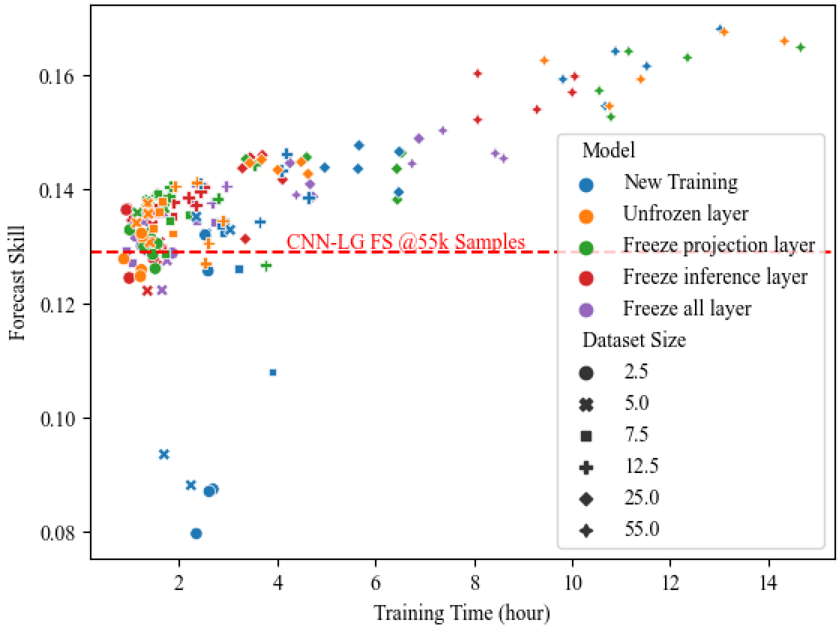


Figure 5.13: Training time versus forecast skill for all models and all datasets.

over, investigating the potential for 0-shot or few-shot learning [162] in the DL-GSI-IHSF domain could provide new ideas and methods for widespread deployment and use.

Transfer learning offers two crucial benefits regarding time and cost: reduced training time and dataset collection time. With transfer learning, the model's training time reduces significantly without compromising the model's accuracy level; this is enabled by giving the model a better starting point and freezing a fraction of the model's weights. Furthermore, transfer learning facilitates quicker model performance evaluation at the start of data collection without waiting for a complete data collection cycle.

Nevertheless, our study has identified certain limitations of current transfer learning methods, particularly related to generalisation when fine-tuning actual collected data. Our findings suggest that the model's excellent performance on the June data and average performance in the other two datasets were most likely due to a lack of picture diversity. This issue might be because transfer learning, as applied in this chapter, only offers a better initial model value and lacks guidance on the target domain task during training aids, such as common feature representation. Future research directions could focus on enhancing the adaptation of feature space during transfer learning, for example, by using prior knowledge constraints on the target domain. Such an approach would address the issue of limited generalizability in fine-tuned data and ultimately improve the model's overall performance post-transfer learning.

In conclusion, our study demonstrates the potential benefits of employing transfer learning in the DL-GSI-IHSF domain, offering valuable insights and paving the way for future research. By refining our understanding of transfer learning strategies, exploring more sophisticated mechanisms, and addressing current limitations, we can continue to advance the field and develop more effective solutions for real-world applications.

5.7 Summary

Extracting and generalising the empirical knowledge of solar energy forecasting from existing datasets is crucial for advancing DL-GSI-IHSF research. Focusing on and developing the transferability of models can not only positively impact model research and communication but also save time and cost in practical deployment. In this chapter, we first compared the limited Nottingham dataset with the Folsom dataset used in Chapter 4, clarifying the differences between the datasets before and after transfer. We found that solar irradiance variations are more complex in the colder and wetter Nottingham area compared to the Folsom dataset. Frequent

RE occurrences can be observed throughout the year. Next, we compared the performance of the three models proposed in Chapter 3 based on the Nottingham dataset, verifying the superiority of the ViT-E model architecture. The results show that in the more complex Nottingham region, the ViT-E model, which employs attention mechanisms for early modality interaction, achieves better FS and meets the prerequisites for transfer learning.

In the transfer experiments of this paper, we used a relatively basic weight transfer method. Through gradual exploration, we found that the weight transfer-based method can effectively transfer the prior knowledge of pretrained models under different climatic conditions, enabling them to play a role in DL-GSI-IHSF downstream tasks in different climates. Moreover, the model can achieve effective training with very few datasets by combining different transfer strategies. At the same time, transfer learning can save the time and cost of model training and data collection. In this project, the prediction model can perform the CNN architecture on a complete dataset using only one-tenth of the training time and one-twentieth of the dataset size compared to training from scratch.

Furthermore, this chapter combined transfer learning methods and phenomena from other research fields to conduct a detailed analysis of transfer learning in the DL-GSI-IHSF domain of this work. In the simulation experiment, although the method of using continuous two-week data for transfer learning also gained a specific performance improvement from the source dataset, its performance was not as good as the potential demonstrated by the dataset with excellent generalizability. Therefore, in future work, we recommend adopting more advanced transfer learning strategies to achieve more efficient transfers in datasets with limited temporal diversity.

Chapter 6

Simplification and Optimisation of ViT-E

Chapter Abstract

The focus of this chapter was on enhancing the performance of the ViT-E model by implementing architectural refinements and conducting hyperparameter tuning. As the model architecture was investigated, it became evident that the DL-GSI-IHSF model possesses potential for further simplification. By employing merely 25% of the depth of the source visual model, we succeed in preserving the model's computational efficiency and overall performance. Moreover, we utilise hyperparameter search techniques to meticulously fine-tune the model's primary hyperparameters. The ultimate results demonstrated that the optimised model can significantly reduce training time by nearly 60% while simultaneously boosting forecast skill by 2.7%.

Contents

6.1 Motivation

In various deep learning applications, model structure, hyperparameters tuning and optimisation play a crucial role in model performance. Proper model structure enhances the efficiency and accuracy of the prediction model. Furthermore, fine-tuning hyperparameters can optimise the training process, rendering it more efficient and stable, increasing the odds of the model reaching its theoretical optimal performance.

However, the current DL-GSI-IHSF article does not sufficiently address model architecture. Optimising and debugging model architectures is a subjective task that even experienced researchers sometimes find challenging to explain. Additionally, model debugging does not have a significant degree of regularity, requiring a combination of guesswork and iterative experimentation [113]. Debugging often entails a high computational cost, requiring hundreds or even thousands of iterations, in contrast to model training. Furthermore, local computational units can significantly influence hyperparameter tuning, leading to non-generalisable results. As a consequence, the model debugging and tuning efforts are considerably hampered.

This section ensures that the ViT-E model in the previous section complies with basic model architecture simplification and hyperparameter tuning. This work also aims to test several deep model hyperparameter optimisation strategies designed primarily for the DL-GSI-IHSF sector to verify their efficiency. This testing is done to optimise model performance.

Unless otherwise stated, the experiments described in this chapter were carried out in the same experimental environment as in Chapter 5.5.

6.2 Model Architecture Simplification

As introduced in Chapter 2, in the work of DL-GSI-IHSF, the current mainstream model development method is to transplant validated computer vision models. By extracting image features from all-sky images through spatiotemporal analysis, single-modality solar forecasting can be performed directly, or multi-modality forecasting can be conducted with historical numerical input. In this process, the architecture of the computer vision model is often directly used as it was during the initial model development, without further editing or validation. On the one hand, the direct use of general-purpose machine vision models has been extensively validated in various fields and has higher credibility. On the other hand, general-purpose machine vision works are usually built into deep learning frameworks, such as TensorFlow, which can be quickly and easily called directly without the need to build the model framework.

However, this simple, quick, and validated approach in other fields is not rigorous when applied to solar forecasting. Specifically, the transplanted models have not been thoroughly investigated and researched for their architectural effectiveness, and their design goals and concepts do not fully align with the needs of solar forecasting. In terms of model architecture, Chapter 4 examines the modal validity of the current model and potential flaws in fusion. In addition to the model architecture, solar forecasting differs from computer vision tasks regarding prediction objectives. In particular, the solar forecasting task is not a pure regression task. As previously introduced, solar forecasting is a numerical prediction task with regression properties on the one hand, and its values have a dual-periodicity characteristic in the ideal clear-sky state, i.e., the daily-cycle variation of irradiance and the annual-cycle variation of solar angle. In addition to the regression task, the periodic numerical prediction will be interrupted by ramp events caused by cloud movement, resulting in instantaneous changes of up to several tens of percent [4]. In other words, the solar forecasting task is a dual-periodicity numerical regression task and an anomaly detection task. The model needs to predict upcoming anomalies and estimate the magnitude of the anomalies. Therefore, directly transplanting computer vision models aimed at image classification without testing in model construction lacks rigour.

At the same time, the complexity of image feature extraction requirements differs between solar forecasting and traditional computer vision classification tasks. Taking ImageNet [123], the most widely used dataset in general CV model design, as an example, its main task is to classify the main content of images into 1,000 predefined categories. In contrast, the specific task of sky images is to identify and distinguish cloud pixels, sky pixels, and sun pixels. Some models also include the derivation of potential ramp events through pixel flow. Regarding image data complexity, ImageNet images are much more complex than sky images. ImageNet images can contain diverse information, such as objects commonly found in nature, textures, structures, and colours. In contrast, sky images only contain the sky background, clouds, and the sun, with colours limited to blue and white. Therefore, it is unreasonable to apply models in tasks of different complexity directly. Similar phenomena have been observed in previous DL-GSI-IHSF work, as shown in Figure 6.1. In the work by Wen et al. [77], they compared three advanced models at the time, VGGNet [98], ResNet [99], and DenseNet [102], when selecting backbone models. The authors used the shallowest versions of VGGNet and DenseNet present in their original work, VGG-11, and Dense-121 (where 11 and 121 represent the number of model layers), while they used three different depth models for ResNet:

Res-18, Res-34, and Res-50. The authors found that, on the one hand, the improvement brought to the solar forecasting domain by increasing model complexity was more significant at lower complexity levels, such as from VGG-11 to Res-18. However, as the model's complexity continued to deepen, the improvement began to slow down. For example, Res-34 only improved solar forecasting by less than 0.5% compared to Res-18, while it improved ImageNet performance by nearly 3%. On the other hand, the authors found that overfitting began to occur when the model depth was further increased to Res-50 in solar forecasting. In the performance of the original work of ResNet on ImageNet, according to the original authors, performance improvement could still be observed with a 152-layer model. Overfitting was reported when the model was deepened to 1,202 layers [99]. In our experiments, we also observed similar phenomena. As shown in the figure, according to the results in Table 4.2 of Chapter 4, CNN-L (based on Res-18 architecture) and ViT-L (based on ViT-B/16 architecture [100], but using different image patch size) also showed similar characteristics. On the one hand, as analysed in Chapter 4, this is due to the influence of unbalanced modalities; on the other hand, the results also indicate that models that perform better on ImageNet may not necessarily be suitable for solar forecasting.

Therefore, we believe that, whether from direct observation or objective experiments, both reveal the same problem: the complexity of the image feature extraction task in DL-GSI-IHSF work is far lower than that of computer vision-based image classification tasks. As a result, we hypothesise that although the ViT-E model based on the ViT architecture has been designed using the simplest version of the ViT-B//16 model, it may still be further simplified in terms of model architecture without affecting performance. In this section, we test and simplify two dimensions of the ViT-E

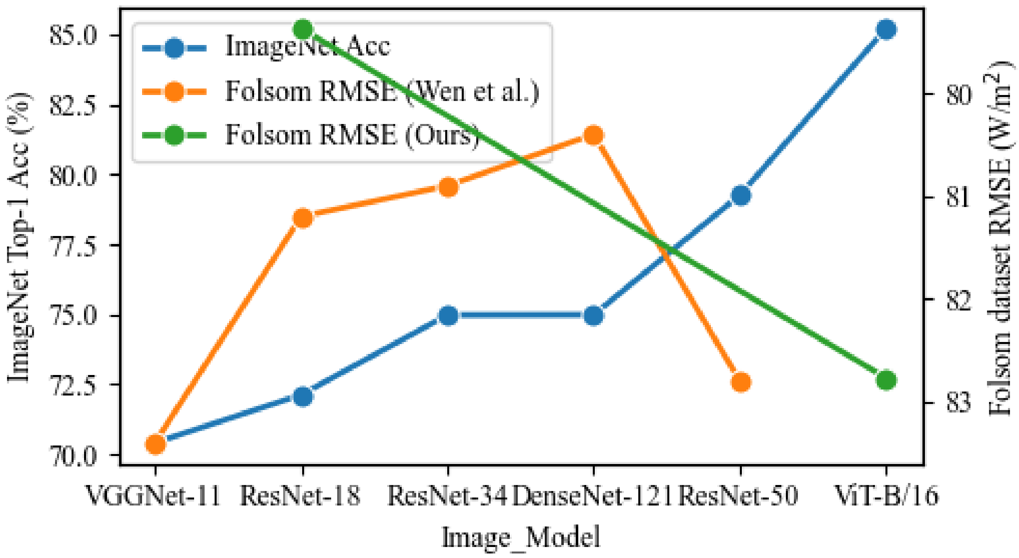


Figure 6.1: Performance of the same model in ImageNet dataset versus Folsom Dataset, evaluated in ImageNet as Top-1 Accuracy of Image Classification [98, 99, 102, 100], and in Folsom dataset as a simultaneous prediction of RMSE [77]. Note that the negative axis is inverted as smaller RMSE values represent better model performance.

Table 6.1: Search space for model architecture.

Search items	Default	Search space
Depth	12	1,2,3,6,12,16,24
Number of Head	12	1,2,3,6,12,18,24

model: the depth of the Transformer Encoder Layer, the main inference layer of the model, and the number of heads in the multi-headed self-attention mechanism. We used a manual grid search method to compare models by traversing the parameters in the search space. Table 6.1 shows the search space for the different search items. All models were repeated five times to minimise the effect of randomness on model performance. It is worth noting that as the model architecture search requires a large number of iterative trials, the smallest dataset that can train a stable model from Figure 5.9 in Chapter 5, i.e. 12.5K dataset, is used in this section for validation, while training the model only at the 2 minutes ahead forecast where the model differences are most pronounced.

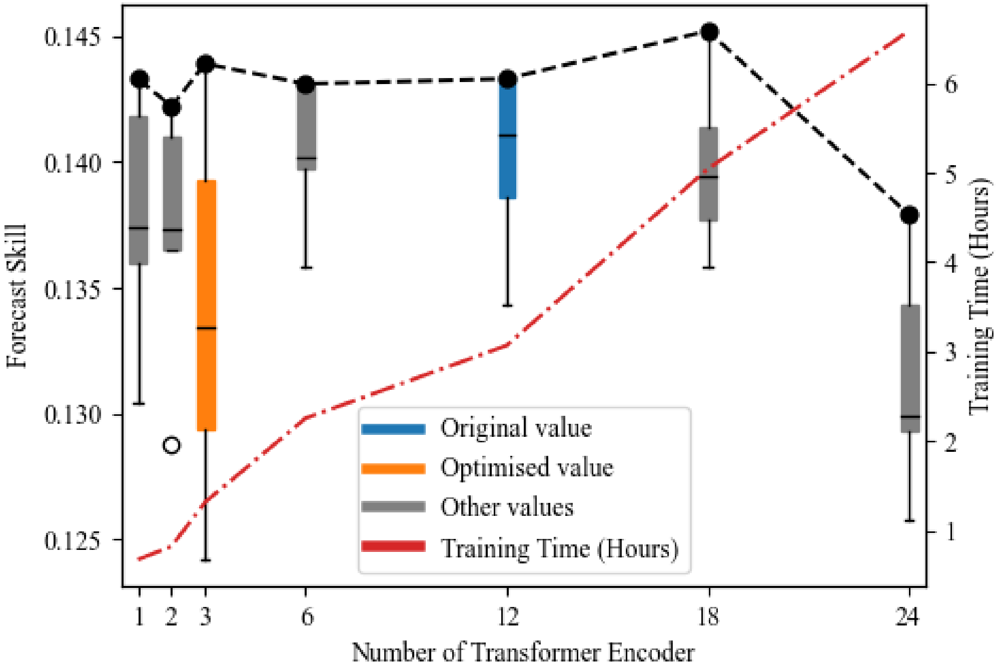


Figure 6.2: Performance of the model in 2-minute forecasts at different depths. The folded line shows the trend of the optimal performance in the repeat models.

6.2.1 Model Simplification on Depth of Transformer Encoder

The depth of a model refers to the number of layers used for computation. Theoretically, deeper models should have a better generalisation and representational capabilities [163]. However, in practical applications, it is not always the case that deeper models yield better performance [99]. Overly deep models can cause overfitting, reducing the model's generalisation ability [101]. As mentioned earlier, we believe the complexity of parsing graphical information in solar forecasting is far lower than that in general computer vision tasks. Therefore, the main goal of this section is to investigate whether there is room to reduce the model depth in the ViT-E module. We train models with different depths while maintaining the number of heads at 12. The results are shown in Figure 6.2.

As shown in the figure, the model maintains nearly the same optimal performance within the search space from depth 3 to depth 18 and experiences a significant drop when reaching depth 24. The optimal FS performance is achieved at 18 with 14.52%, followed by 14.39% at a depth of 3. At the same time, regarding training efficiency, the model depth directly determines the number of trainable parameters, which influence the training efficiency of the model. In ViT-E, each Transformer Encoder layer has 0.44 million parameters, accounting for more than half of the total 0.85 million parameters in the other two modules (projection layer and prediction head). The figure shows that the model computation time increases linearly with depth. When the depth reaches 24, the average training duration is 6.6 hours. Therefore, we choose to compress the model depth to 3 layers to balance performance and computation time. It is worth noting that the 3-layer model has a significant drawback. Although the optimal performance of a single model is comparable to that of other optional depths, the training process is not stable enough. Compared to the deeper 6 to 18 layers, the 3-layer model has the most considerable performance variation in repeated experiments among all models, indicating that it is most susceptible to randomness. Therefore, we made a series of adjustments to this depth model in the following experiments, aiming to improve the stability of the model training process.

6.2.2 Model Simplification on Head in MSA

The multi-head self-attention mechanism is the core reasoning mechanism of the Transformer architecture. The term "multi-head" refers to the MSA block mapping the input matrix to different attention heads through the trainable matrix $\mathbf{W}^O$, as shown in Equation 4.6. The multi-head mecha-

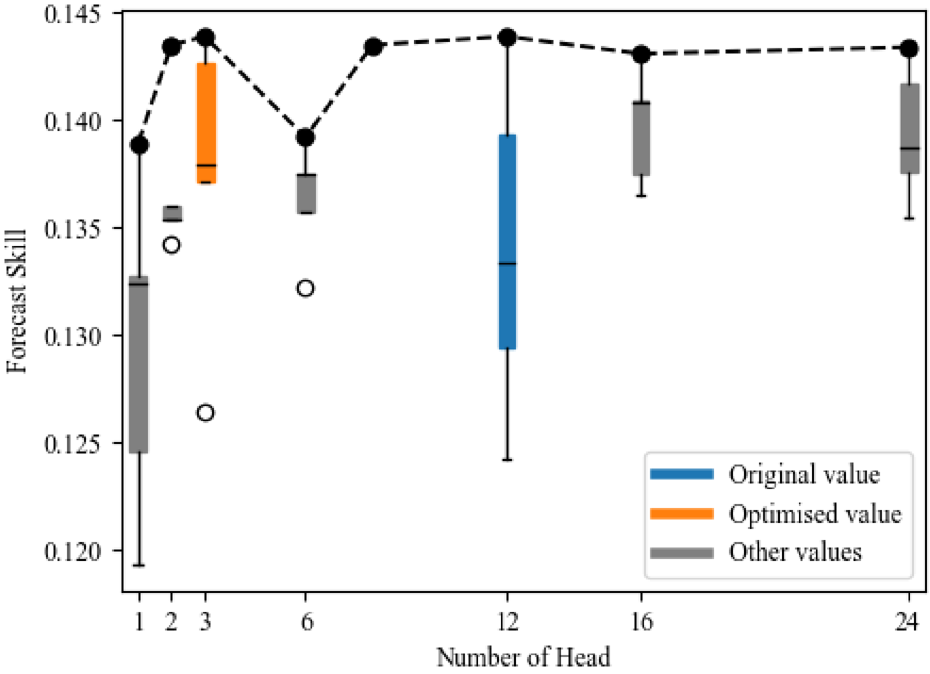


Figure 6.3: Performance of the model in 2-minute forecasts at different head in MSA. The folded line shows the trend of the optimal performance in the repeat models.

nism allows a single set of queries, keys, and values to perform different query behaviours (such as different distances and dimensions) in different representation subspaces [164]. Therefore, more heads are considered to capture richer information, improving the model's expressive power and accuracy. In the original design of ViT-E, we directly followed the original design of ViT-B/16, i.e., 12 heads. In this section, we search for the number of heads based on the three-layer model depth in Figure 6.2, and the results are shown in Figure 6.3.

As shown in Figure 6.3, there is no clear pattern in the number of heads in MSA for model prediction. Except for the significant decrease in optimal performance when the number of heads is 1 and 6, the optimal performance of the model under other head numbers does not differ much. In the model architecture design, the ViT-E model refers to the native ViT work. The authors evenly split the weights parameters into each representation subspace when mapping a set of queries, keys, and values to different subspaces. Therefore, the total parameter amount of the queries, keys, and values do not change during the calculation, so using a different number of prediction heads does not significantly affect the computation time of the model under the premise of unchanged model depth. Regarding training process stability, the optimised result in Figure 6.2 based on 12 prediction heads shows the worst stability. Finally, we decided to adjust the number of prediction heads to 3. On the one hand, three prediction heads achieve the best performance in the weight search architecture. On the other hand, three heads can keep the matrix dimensions in each head consistent with the original model (ViT-B/16 has a matrix dimension of $\mathbb{R}^{64\times(N+1)}$ in each head).

6.3 Hyperparameter Tuning

6.3.1 Introduction

The performance of deep learning models mainly manifests in the model's prediction performance, such as accuracy and generalisation, and the model's computational efficiency. The primary purpose of hyperparameter tuning is to maximise the performance of deep learning models. However, the tuning process often faces significant challenges. On the one hand, different architectures, algorithms, and optimises have unique optimal hyperparameter combinations. When adjusting the model's architecture or using different optimises, the best hyperparameter combination will change accordingly [165]. Additionally, as previously mentioned, the training of deep learning models inevitably faces the influence of randomness. Consequently, the hyperparameter adjustment process in deep model design often requires extensive time for repeated testing [166].

On the other hand, due to the black-box nature of deep learning, the influence of hyperparameters in the model is often based on the model's results and researchers' subjective judgement. Simultaneously, the process lacks deductive reasoning [167]. Therefore, the model's hyperparameter adjustment process is not strictly a rigorous scientific exploration but a process driven by experience, which may include a large number of subjective attempts and even unfounded guesses by researchers.

In this section, we made a series of adjustments to the ViT-E model's hyperparameters. Our goal is slightly different from the original intention of hyperparameter tuning. In addition to searching the hyperparameter space to determine the best combination, the ViT-E project employs some "default" deep learning optimisation techniques, such as weight decay [168], learning rate decay [169, 170], and others. These methods generally improve the model's computational performance during training. This section tested these techniques to determine their roles in ViT-E. It is worth noting that the model's hyperparameters are also influenced by the data pipeline, which is responsible for transferring data from the hard disk to the network [171]. The training platform's hardware determines the data transmission rate in the data pipeline.

Therefore, this section's specific values of hyperparameter adjustments have absolute numerical reference significance only on the local machine or when using completely identical machines [172]. To provide a more generalised understanding, we recommend that researchers consider the hardware constraints, data pipeline efficiency, and the specific requirements of their tasks when adjusting hyperparameters. Additionally, employing automated hyperparameter tuning techniques, such as Bayesian optimisation [173], random search [165], and Hyperband [166], can help to alleviate the challenges faced during the hyperparameter optimisation process, improving both prediction performance and computational efficiency.

6.3.2 Methodology

This study employed the ViT-E model as a basis for hyperparameter tuning. We first fixed the model architecture, utilising a three-layer model with three MSA heads, as validated in the preceding section. Although the Adam optimiser was initially considered, it was abandoned due to its poor performance in preliminary experiments. Consequently, we continue to opt for the SGD optimiser with a momentum of 0.9. In our weight search, we refrained from utilising automated search methods such as random or Bayesian optimisation. Instead, we employed a more straightforward manual approach to gain insights into the model's performance during the convergence process.

Batch Size Batch size denotes the number of samples input into the model for gradient computation during the deep model parallel process, as illustrated in Equation 1.8. It signifies the number of samples the model can process in parallel simultaneously. Theoretically, with hyperparameter optimisation, batch size does not impact the model's computation accuracy. Regardless of batch size, the model can achieve its maximum potential performance [174]. However, other batch-sensitive hyperparameters, such as weight decay and learning rates, may influence the model's final performance [113]. Furthermore, batch size significantly affects the model's computational efficiency, as it determines the training time and consumption of computational resources. Ideally, doubling the batch size would double the model's parallelism, increasing training throughput. In practice, however, factors such as disk read and write speeds or CPU processing speeds become bottlenecks, ultimately affecting the model's computational efficiency. Additionally, a larger batch size can reduce sample variance in each batch, accelerating the training process [174]. However, reduced sample variance translates to less noise, which can render the model more susceptible to overfitting.

Learning Rate The learning rate dictates the proportion of the gradient to the weight during the model fitting process, significantly impacting the model's computational performance and efficiency. As the model minimises loss, the learning rate governs its convergence efficiency. A more significant learning rate enables faster parameter updates, thereby hastening model convergence. Conversely, a lower learning rate facilitates fine-tuning the minimum value of the loss as the model approaches the optimal solution. Moreover, since the loss may have local optima, a larger learning rate can help the model escape local optima in search of the global optimum [170]. Therefore, determining an appropriate learning rate curve is essential during model training. In this section, we searched for the optimal learning rate size and experimented with custom learning rate curves, including cosine decay and cosine annealing strategies [169].

Weight Decay Weight decay, or L2 regularisation, is a strategy to prevent model overfitting. It restricts the growth of model weight values by introducing a weight penalty term to the loss function. Limiting the model's weight values reduces overfitting risks during training and simplifies the model to some extent, as smaller weight values are closer to the vector

Table 6.2: Search space for hyperparameters in ViT-E.

Parameters	Default	Turning Range	Description
Batch Size	64	[4, 8, 16, 32, 64, 128, 256]	The number of training samples per iteration.
Learn Rate	8e-4	[5e-3, 1e-3, 8e-4, 5e-4, 1e-4, 8e-5, 5e-5]	The step size in updating model weights.
Learn Rate Curve	CA-R	[Constant, CA-NR, CA-R]	Adjusting the learning rate over training iterations.
Weight Decay	1e-4	[1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 0]	The strength of the weight decay regularisation.

origin. This simplification decreases the model's complexity, enhancing its generalisation capabilities [168]. Consequently, weight decay effectively balances the trade-off between model complexity and generalisation performance.

In conclusion, this study focused on refining the ViT-E model by carefully tuning hyperparameters, including batch size, learning rate, and weight decay. We emphasised the importance of selecting an appropriate model architecture, optimiser, and weight search method to optimise performance during the convergence process. Our investigation into the relationship between batch size and computational efficiency, learning rate and convergence efficiency, and weight decay and overfitting prevention provided valuable insights for improving the model's overall performance. Furthermore, exploring custom learning rate curves, such as cosine decay and cosine annealing strategies, contributed to a more nuanced understanding of learning rate allocation during training. Table 6.2 shows the hyperparameter search space. These findings underscore the critical role that hyperparameter tuning plays in maximising the potential of deep learning models and achieving optimal results in various applications.

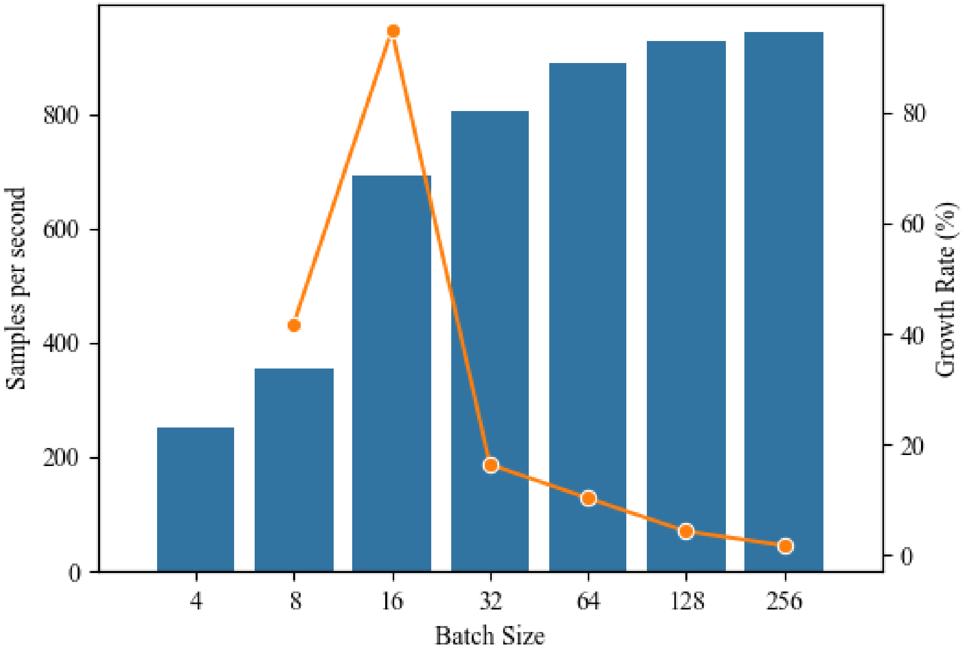


Figure 6.4: Comparison of model throughput in different batch sizes.

6.3.3 Result

Batch size Batch size is a hyperparameter limited by hardware and data pipeline I/O (Input and Output) considerations. As previously mentioned, theoretically, batch size does not affect model accuracy. Generally, to maximise computational efficiency by arranging data neatly in memory, the batch size is set to a power of 2. In our computing environment, the maximum batch size the device's memory can support is 2^8. In this section, we tested the computational efficiency of batch sizes ranging from 2^2 to 2^8 on our local device. The results are shown in Figure 6.4. When hardware and data pipeline constraints are absent, doubling the batch size can double the model's throughput range in an ideal state. However, as shown in the figure, only increasing the batch size from 8 to 16 doubled the model's computational efficiency. At other batch sizes, while increasing batch size can improve throughput, it does not result in throughput growth close to doubling. Our investigation found that this phenomenon is due to different parts of the data pipeline limiting data processing efficiency. When the batch size is small, the total batch count is large because the total batch count is inversely proportional to the batch size. The data pipeline needs to repeat the read-map-pack preparation process for each batch, decreasing batch processing speed due to the impact of CPU computational efficiency. When the batch size is too large, many samples must be read and written to the disk simultaneously, which becomes the bottleneck of training efficiency. Additionally, increasing the batch size reduces the number of iterations per epoch, resulting in slower model fitting speeds. Based on our findings, we ultimately conducted a hyperparameter search with a batch size of 64 samples per batch, as it had an average training speed 50% faster than a batch size of 128.

Learning Rate The ratio of the gradient to the weight during the model's gradient descent process. A larger learning rate is beneficial for quick model fitting, while a lower learning rate is beneficial for fine-tuning the weight parameters. In order to adjust the learning rate at different stages of training, deep models are typically trained using a learning rate curve that varies with the training step, as shown in Figure 1. In previous training, we defaulted to using a cosine decreasing strategy to adjust the learning rate, which gradually decreases as the model iterates, while using a larger initial learning rate. We first searched for the effect of the initial learning rate on the model performance. In this work, the initial learning rate, i.e. the learning rate of the model at the first epoch, is also the maximum learning rate during the whole training process. Figure 6.5 shows the effect of the learning rate on the prediction performance of the model: We conducted repeated experiments with seven different learning rates to test the results.

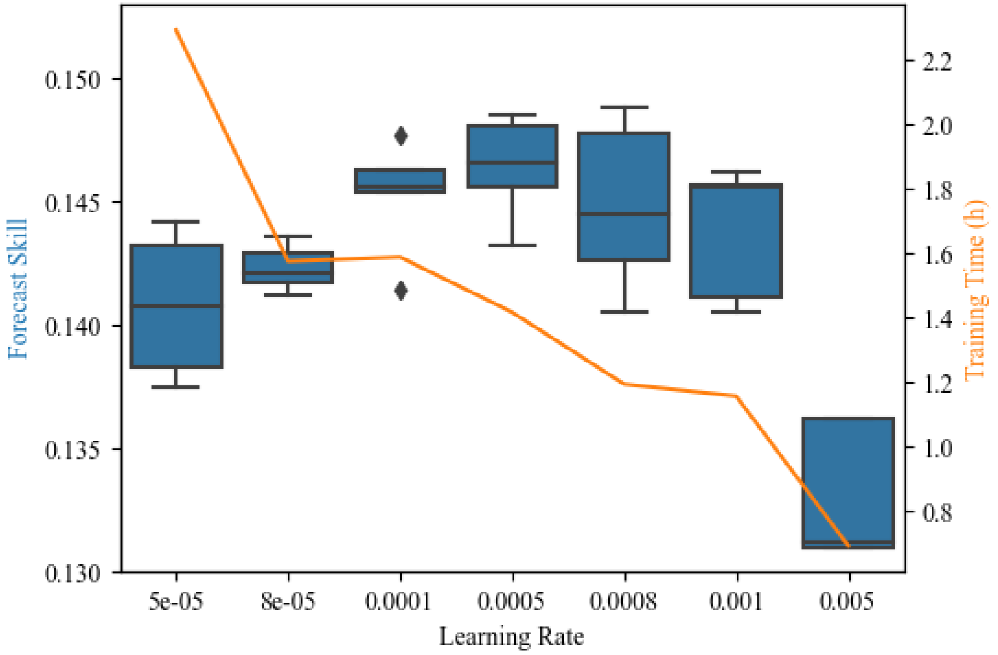


Figure 6.5: Comparison of model forecast skill in different learning rates with batch size 64.

The default learning rate used was 8e-4, also the recommended learning rate for the ViT model [100]. The figure shows that the recommendation learning rate shows optimal performance across the search space. The model obtains the maximum performance improvement with an increased learning rate before 8e-4 and decreases after exceeding it. Additionally, the model's training time gradually decreases as the model's initial learning rate increases. Considering training time and accuracy, we maintained a learning rate of 8e-4 for further exploration.

It is worth noting that the learning rate set in the above figure is the initial learning rate. As mentioned earlier, we hope the learning rate can be relatively large in the initial stage of training to improve the model's rough fitting efficiency. When the model approaches the optimal point, we hope the learning rate can gradually decrease to achieve fine-tuning. In this study, we used the standard Cosine annealing [169] strategy by gradually

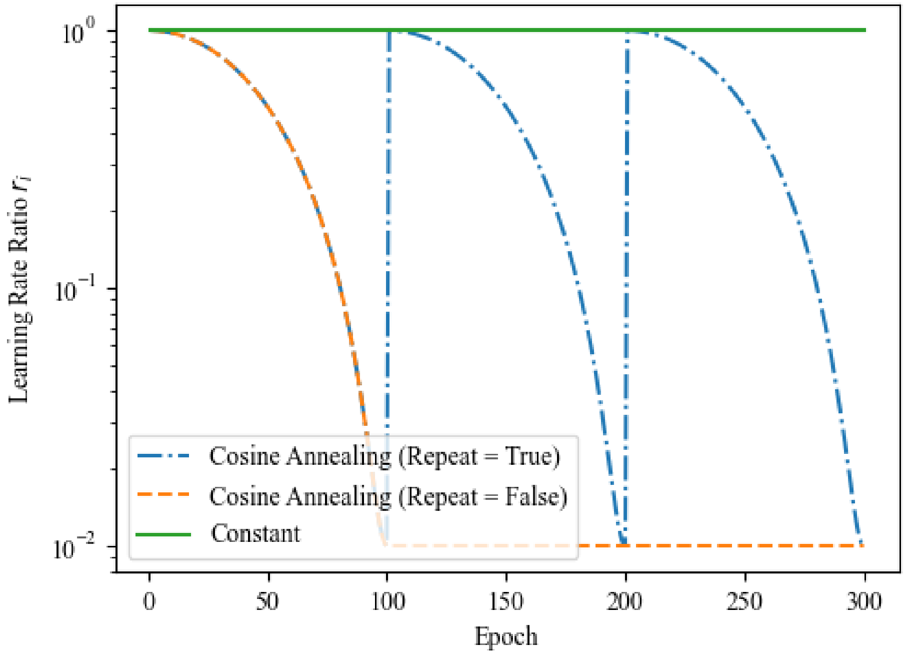


Figure 6.6: Three different learning rate control strategies.

reducing the learning rate through a cosine function during the training process, as shown in Figure 6.6. It can be represented by the equation below:

$$\eta_i = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + \cos(\frac{i}{I}\pi)) \tag{6.1}$$

$$= \eta_{max}(r + \frac{1}{2}(1 - r)(1 + \cos(\frac{i}{I}\pi)) \tag{6.2}$$

$$\text{where } r = \frac{\eta_{min}}{\eta_{max}} \tag{6.3}$$

where, η represents the learning rate, i and I represent the current epoch and the total number of epochs in the decay process, respectively, and r represents the total decay rate. In this study, η_{max} is the initial learning rate, 8e-4, I is set to 100, and r is set to 0.01. The learning rate decay process through the Cosine annealing strategy is shown in Figure 6.6. In the standard Cosine annealing strategy, the decay process is repeated several times (i.e., Repeat=True in the figure). There is a variant of this strategy

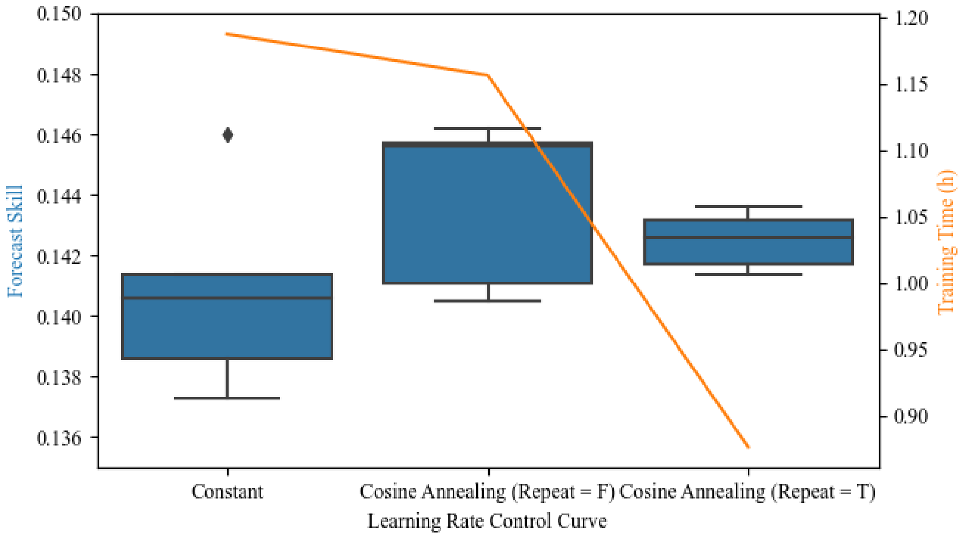


Figure 6.7: Comparison of model forecast skill in different learning rate control strategies with start learning rate 8e-4.

where no repetition occurs. After reaching 100 epochs, the model's learning rate maintains 0.01 times the initial learning rate and does not change. The impact of different strategies on model weight is shown in Figure 6.7.

Weight Decay Weight decay is a regularisation technique employed to mitigate overfitting during the network training process. Specifically, weight decay restricts the complexity of model weights by incorporating the squared sum of all weights in the network into the model's loss function, following a certain proportion. In this section, the hyperparameter under investigation is the intensity of weight decay, i.e., the multiplication coefficient associated with the squared sum of weights. The results are illustrated in Figure 6.8.

It should be noted that, due to the inability of some models to fit properly when the weight decay is excessively large, a direct comparison using validation set results was conducted, rather than evaluating the models on the test set after training. Consequently, the vertical axis in the figure

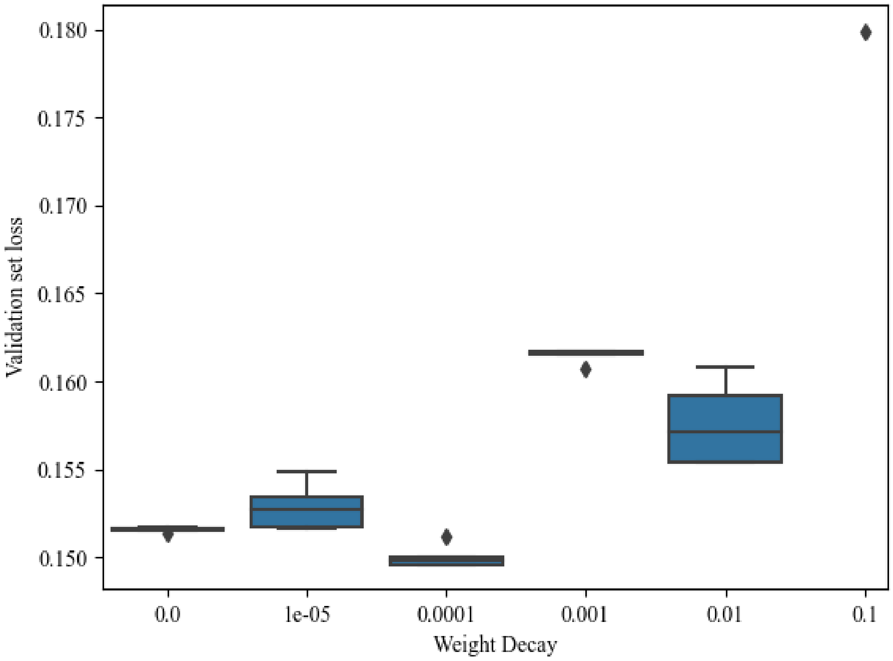


Figure 6.8: Comparison of model forecast skill in different weight decay rate.

represents the validation set loss, not the test set FS mentioned previously. In other words, a smaller loss implies superior performance, which is contrary to the previous interpretation. As demonstrated in the figure, the model's performance initially improves and then deteriorates as the weight decay coefficient gradually increases. The optimal performance is achieved with a weight decay ratio of 1E-4. To further analyse the impact of weight decay on model training, Figure 6.9 displays the training process of the best-performing models in six repeated experiments with different weight decay rates.

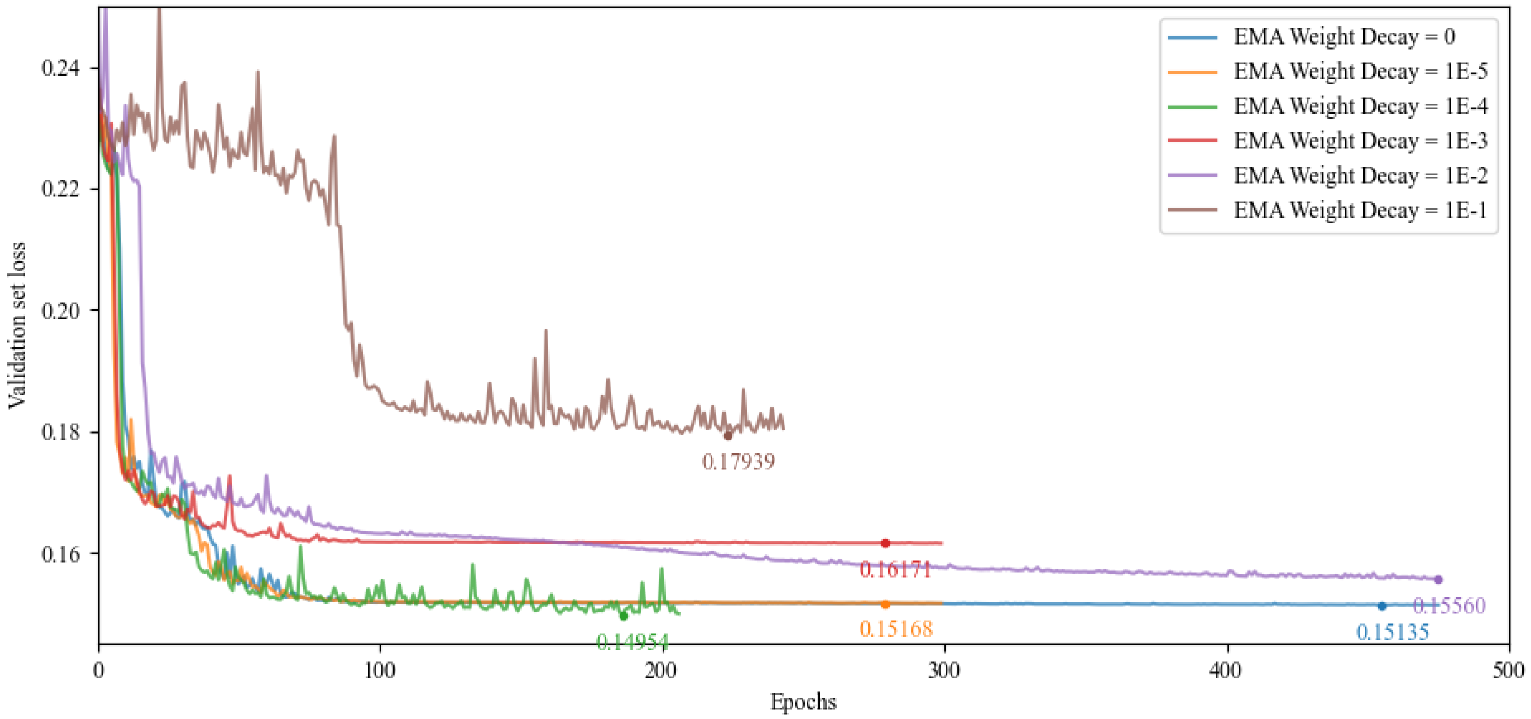


Figure 6.9: Figure: Comparison of model training performance with different weight decay rates, where the training curve is smoothed using an EMA with a smoothing factor of 0.2. The labeled points in the figure represent the actual minimum loss values saved during the training process.

As shown in the figure, the model is almost unable to complete training when the weight decay value is 0.1. In the experiments, the model's weight fitting was effective only once among the five repeated trials, as shown in the figure, and was unsuccessful in the other four instances. When the weight decay is set to 0.01, the training efficiency of the model significantly declines. The section shown in the figure is not fully displayed due to space constraints. Limited by the maximum training epochs of 500, the model could not complete the entire training process. In an unrestricted-duration test, training stopped at 1.2K epochs, and the best loss achieved was 0.1552. In conclusion, the default weight decay of 1E-4 continues to be the optimal choice for the model.

6.4 Deploy optimisation strategies on full datasets

In this section, we synthesise the search results from the previous two sections and conduct performance optimisation tests on the complete dataset using the ViT-E model. Due to the optimisation process, in order to save training time costs, the actual adjustments were performed on a smaller dataset (12.5K samples), so the limited dataset size still affects the model's performance. To test the model's achievable optimal performance, after the optimisation process, the results of the optimisation method on the full Nottingham dataset are shown in the following Table:

Table 6.3: Performance comparison of optimisation strategies on two different datasets

Dataset		Model Architecture			Hyperparameter			Result (Average)		
		Model Depth	Number of Head	Batch Size	Initial Learning Rate	Learning Rate Decay	Weight Decay	Forecast Skill	F_1 Score	Training Time (h)
12.5K	Original	12	12	8	8.00E-03	CA-R	1.00E-04	14.07%	35.67%	4.13
	Optimised	3	3	64	8.00E-03	CA-NR	1.00E-04	14.48%	36.19%	1.18
55K	Original	12	12	8	8.00E-03	CA-R	1.00E-04	16.1%	38.87%	6.47
	Optimised	3	3	64	8.00E-03	CA-NR	1.00E-04	16.55%	39.38%	2.62

As shown in the table, after optimisation, the model's prediction score increased by 2.9%, while the F1 score decreased by -1.4% in the smaller debugging dataset. Additionally, the model's computational efficiency improved by 71.4%. After deploying the optimised model on the complete Nottingham dataset, both the prediction scores and F_1 scores increased by 2.7% and 1.3%, respectively, and computational efficiency improved by 59.5

6.5 Summary

The model's hyperparameters have a significant impact on its performance. In this chapter, we conducted a careful search for the model's architecture and hyperparameters. Ultimately, on the complete dataset, the entire optimisation strategy improved the model's forecast skill by 2.7% and the F1 metric score by 1.3%. Furthermore, it saved nearly 60% of the training time. During the model tuning process, we confirmed the speculation proposed earlier in the text. Specifically, we found that the parsing complexity of sky images is far lower than the standard image classification tasks in computer vision, so the demand for the model's parsing ability is lower and does not require a deep model architecture. In addition, in this chapter, we also tested some mainstream model optimisation strategies, such as cosine annealing strategy and weight decay strategy. The results showed that the performance of the model in the DL-GSI-IHSF work can be optimised based on general deep model optimisation strategies. It is worth noting that this paper only searched for some major hyperparameters, and there are still other potentially effective hyperparameters not within the search scope. Moreover, the model's hyperparameters are directly related to the model's architecture and data pipeline design, so the conclusions of this paper have certain limitations, particularly in the specific numerical results of hyperparameter searches.

Chapter 7

Conclusion

Contents

7.1 Summary of Contributions

In general, in this paper we explore the current state of DL-GSI-IHSF and iterate and optimise existing methods from different perspectives. Specifically, we optimise the existing DL-GSI-IHSF work in three ways.

7.1.1 Fusion Interaction of Sky Images and Measurement Data

Drawing from experience, a description of objects based on multiple modalities and dimensions is invariably more comprehensive than one based on a single modality. The heterogeneous noise topologies across different modalities can be filtered through cross-validation, leading to a more accurate description of the subject. This is also true for the DL-GSI-IHSF framework, where the developed models achieve superior results on specific metrics by cross-processing image information and measurement data compared to single modality approaches. However, in Chapter 4, we pointed out that the deep learning methods guided by minimising loss function inherently exhibit certain biases. Loss functions designed based on numerical regression direct the model training toward numerical regression, resulting in a lack of sensitivity to anomalous predictions. Consequently, most models in existing work demonstrate a deficiency in their ability to anticipate Rapid Events (RE).

In Chapter 4, we enhanced the modality interaction of existing models at the architectural level. By incorporating advanced structures controlling information interaction in two types of deep models, gate structures and attention mechanisms, we constructed a more comprehensive model interaction mechanism. This method effectively improved the models' anomaly detection capabilities. In particular, the ViT-E model, based on early modality interaction, increased the RE capture rate by 10% in the solar energy prediction task two minutes ahead without sacrificing regression performance.

Through the visualisation analysis of model results, we found that researchers overestimated the effective prediction range of sky images. The prevailing consensus suggests that sky images can effectively forecast solar irradiance and slope events within 15 minutes. However, in our visualisation analysis, the effectiveness of numerical inputs surpassed images at the 6-minute mark, and images became almost insignificant for predictions 10 minutes ahead. Additionally, we highlighted the inherent logical contradiction between regression prediction and anomaly prediction when using a single generic regression function as the model's loss function. Specifically, regression loss functions constrain model weights to converge toward mean regression, which inadvertently leads to the failure of anomaly detection.

7.1.2 Transfer Learning Based on Different Climate Conditions

In practice, the deployment and validation process of DL-GSI-IHSF is lengthy. Even after device deployment, the data collection platform still requires a continuous year of localised information for complete deep model training. Such a time cost is excessive and impractical for model validation. Therefore, in Chapter 5, we propose using transfer learning methods to propagate knowledge within the models. By inheriting experience from rich datasets, the time cost of data collection and debugging is reduced when deploying the model at new locations. Moreover, we selected two datasets with significant climate differences to verify whether the transferred knowledge can successfully migrate under complex climate changes.

In Chapter 5, we accomplished model knowledge propagation under different climates through weight transfer methods. We first compared the similarities and differences between the two datasets in detail before the actual transfer and conducted feasibility pre-experiments to verify the advanced model architecture. In the formal experiment, we confirmed the effectiveness of model transfer learning by comparing the cosine similarity before and after the transfer. Next, we compared the transfer performance of weight space adaptation and label space adaptation. By downsampling and continuously sampling the original dataset, we compared the specific performance of transfer learning in situations with insufficient data.

The transfer learning results confirmed the feasibility of implementing transfer learning in the DL-GSI-IHSF domain. Furthermore, the best-performing experiment demonstrated that provided data diversity is ensured, transfer learning can achieve a relatively stable model using only 4.5% of the original dataset and 10% of the time. We employed transfer learning to conduct transfer experiments using a 2-week dataset in the continuous sampling method representing pseudo-real experiments. The results indicate that the generalisation of the dataset is crucial in transfer learning. In some cases where diversity is insufficient, the model may fail to complete transfer training.

These findings emphasise the importance and effectiveness of transfer learning in the DL-GSI-IHSF domain, highlighting its potential to reduce data collection and training time while maintaining model performance. Transfer learning's success, however, is contingent upon ensuring adequate data diversity and generalisation in the datasets used for training.

7.1.3 Local-based Model Simplification and Optimisation

In our previous work, we found that leading models in the deep vision domain did not exhibit a significant performance advantage in DL-GSI-IHSF. We speculate that the difference in performance superiority may be due to the varying task complexity, that is, the complexity of sky images is much lower than that of images in general image classification tasks. Moreover, we noticed a lack of effective validation for plug-and-play optimisation modules in the deep learning domain in previous work. Therefore, in Chapter 6, we first carried out simplification tests on the model architecture to compare whether reducing the model depth would affect its performance. In addition, we further adjusted the hyperparameters of the simplified model and tested the effectiveness of some popular deep model optimisation strategies in DL-GSI-IHSF.

During the model depth simplification process, we found that when the ViT-E model was simplified to a quarter of the original model's inference module depth, the stability of model training was reduced. However, the achievable optimal performance of the model was not affected at this point. As the model depth was further reduced, the model's optimal performance began to decline. In the process of simplifying the model's parsing head, we further reduced the number of parsing heads to three. In the hyperparameter adjustment, we made further adjustments to several important hyperparameters to optimise the model's performance. Ultimately, through model architecture optimisation and hyperparameter tuning, the model achieved a 2.7% performance improvement on the complete dataset and reduced training time by 60%.

During the model optimisation process, the speculation that graphic parsing complexity in DL-GSI-IHSF work is lower than general image classification tasks was confirmed. The effectiveness of several basic model tuning strategies was also examined. This has guiding significance for the future design of DL-GSI-IHSF model.

7.2 Future Work

In this thesis, we provide a systematic review of DL-GSI-IHSF. As a class of methods within the GSI-IHSF framework, Deep Learning (DL) methods are currently the most advanced prediction methods in the evaluation system. By learning cloud movement patterns in large datasets, deep models can effectively capture spatiotemporal information in sky images, predict upcoming RE, and assist in real-time power system dispatching. Deep learning requires high-quality data and computational resources during model development compared to traditional image analysis-based modelling methods. However, its calculation time during actual prediction is almost negligible compared to traditional methods. Chapter 2 summarises the general development framework of DL-GSI-IHSF, data collection and preprocessing, model architecture design and optimisation, and model evaluation from different perspectives. On the other hand, although Chapters 4, 5 and 6 of this paper explore and extend the DL-GSI-IHSF work to some extent, this field is far from perfect. It requires more time to address the remaining issues and emerging challenges. In the end, we distil several important issues and potential research directions in DL-GSI-IHSF that are worth further exploration in the future:

1. Dataset influence on the model: As the main factor affecting computer vision deep learning models, the quantity and quality of image

data have not yet received attention in DL-GSI-IHSF work. Most current studies are based on modelling a single local or publicly available online dataset rather than multiple datasets simultaneously. At the same time, strict comparative experiments are required for image or dataset model validation. For example, climate and geographic factors at different dataset collection sites will significantly affect the performance of solar energy forecasting models. It is unrealistic to compare datasets across climates. Therefore, a potential research direction is the comparative study of dataset collection equipment, such as sky cameras and pyranometers. On the one hand, comparing datasets helps improve model performance validation and determine the source of performance differences. On the other hand, comparative research on data collection equipment can provide recommendations for equipment selection and installation during the actual deployment of DL-GSI-IHSF.

2. Data preprocessing based on solar energy forecasting background: Existing work has demonstrated the potential of solar energy forecasting prior knowledge in data preprocessing for deep models. Therefore, preprocessing methods combining physics-based solar energy forecasting background knowledge with deep models are promising, for example, fisheye distortion correction, Clear Sky Library method, optical flow, and cloud dynamics. Transforming the data into an expression easier for deep models to recognise through prior knowledge or incorporating it directly as part of deep modelling is a promising research direction.

3. Multi-task learning and loss functions for ramp event-oriented tasks: DL-GSI-IHSF models mostly use MAE or MSE loss functions, which are general-purpose loss functions for regression tasks. However,

short-term changes in solar irradiance as a forecasting target are not purely regression-based predictions but a combination of regression-based predictions and anomaly detection. Anomalous signals, i.e., ramp events, interrupt the model's continuously differentiable regression distribution. To our knowledge, no models have been explicitly designed for anomaly detection in ramp event prediction. Therefore, applying multi-task learning methods and designing specific model architectures and loss functions for ramp events may be a potential research direction.

4. Probabilistic forecasting: Most current deep learning networks provide deterministic predictions, which are numerical values lacking any probabilistic or interval-based information. However, this approach presupposes an unwarranted precision that may not align with meteorological forecasting's inherently probabilistic nature. Therefore, presenting forecasts with a confidence level is crucial for meteorological applications. Recent studies have proposed proprietary model architectures to generate probabilistic ranges, providing additional information on the uncertainty of the forecast. Incorporating probabilistic predictions in GSI-IHSF can improve solar energy forecasting and enable the real-time design of downstream power system modules during operation.

5. Standardised evaluation metrics: As mentioned earlier, current model evaluation metrics, especially for ramp event evaluation, are diverse and inconsistent. This diversity hinders the comparison between models and the evaluation of algorithms. Therefore, future research should focus on developing standardised evaluation metrics for DL-GSI-IHSF models, particularly for assessing ramp events. Establishing a unified set of evaluation metrics would facilitate more accurate

comparisons among different models and help researchers identify the most effective approaches in solar energy forecasting.

Bibliography

[1] IEA. World energy outlook 2022, 2022.

[2] Energy & Industrial Strategy Department for Business. Solar photovoltaics deployment, 2023.

[3] IEA. Solar pv power generation in the net zero scenario, 2010-2030 – charts – data & statistics.

[4] Dazhi Yang, Wenting Wang, Christian A Gueymard, Tao Hong, Jan Kleissl, Jing Huang, Marc J Perez, Richard Perez, Jamie M Bright, and Xiang'ao Xia. A review of solar forecasting, its dependence on atmospheric sciences and implications for grid integration: Towards carbon neutrality. *Renewable and Sustainable Energy Reviews*, 161:112348, 2022.

[5] Matthew Lave and Jan Kleissl. Solar variability of four sites across the state of colorado. *Renewable Energy*, 35(12):2867–2873, 2010.

[6] Rich H Inman, Hugo TC Pedro, and Carlos FM Coimbra. Solar forecasting methods for renewable energy integration. *Progress in energy and combustion science*, 39(6):535–576, 2013.

[7] Mazaher Karimi, H Mokhlis, Kanedra Naidu, Sohel Uddin, and AH Abu Bakar. Photovoltaic penetration issues and impacts in distri-

bution network–a review. *Renewable and Sustainable Energy Reviews*, 53:594–605, 2016.

[8] Fan Lin, Yao Zhang, and Jianxue Wang. Recent advances in intra-hour solar forecasting: A review of ground-based sky image methods. *International Journal of Forecasting*, 2022.

[9] Yinghao Chu, Hugo TC Pedro, Lukas Nonnenmacher, Rich H Inman, Zhouyi Liao, and Carlos FM Coimbra. A smart image-based cloud detection system for intrahour solar irradiance forecasts. *Journal of Atmospheric and Oceanic Technology*, 31(9):1995–2007, 2014.

[10] Yinghao Chu, Hugo TC Pedro, Mengying Li, and Carlos FM Coimbra. Real-time forecasting of solar irradiance ramps with smart image processing. *Solar Energy*, 114:91–104, 2015.

[11] Quentin Paletta, Anthony Hu, Guillaume Arbod, Philippe Blanc, and Joan Lasenby. Spin: Simplifying polar invariance for neural networks application to vision-based irradiance forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5182–5191, 2022.

[12] Sadra Babaei, Chaoyue Zhao, and Lei Fan. A data-driven model of virtual power plants in day-ahead unit commitment. *IEEE Transactions on Power Systems*, 34(6):5125–5135, 2019.

[13] Yuri V Makarov, Clyde Loutan, Jian Ma, and Phillip De Mello. Operational impacts of wind generation on california power systems. *IEEE transactions on power systems*, 24(2):1039–1050, 2009.

[14] David Ganger, Junshan Zhang, and Vijay Vittal. Forecast-based anticipatory frequency control in power systems. *IEEE Transactions on Power Systems*, 33(1):1004–1012, 2017.

[15] Samuel R West, Daniel Rowe, Saad Sayeef, and Adam Berry. Short-term irradiance forecasting using skycams: Motivation and development. *Solar Energy*, 110:188–207, 2014.

[16] Qingyong Li, Weitao Lu, Jun Yang, and James Z Wang. Thin cloud detection of all-sky images using markov random fields. *IEEE Geoscience and remote sensing letters*, 9(3):417–421, 2011.

[17] K Stefferud, J Kleissl, and J Schoene. Solar forecasting and variability analyses using sky camera cloud detection[online] motion vectors. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–6. IEEE, 2012.

[18] Handa Yang, Ben Kurtz, Dung Nguyen, Bryan Urquhart, Chi Wai Chow, Mohamed Ghonima, and Jan Kleissl. Solar irradiance forecasting using a ground-based sky imager developed at uc san diego. *Solar Energy*, 103:502–524, 2014.

[19] Zhenzhou Peng, Dantong Yu, Dong Huang, John Heiser, Shinjae Yoo, and Paul Kalb. 3d cloud detection and tracking system for solar forecast using multiple sky imagers. *Solar Energy*, 118:496–519, 2015.

[20] Yinghao Chu, Mengying Li, Hugo TC Pedro, and Carlos FM Coimbra. A network of sky imagers for spatial solar irradiance assessment. *Renewable Energy*, 187:1009–1019, 2022.

[21] Chi Wai Chow, Bryan Urquhart, Matthew Lave, Anthony Dominguez, Jan Kleissl, Janet Shields, and Byron Washom. Intra-hour forecasting with a total sky imager at the uc san diego solar energy testbed. *Solar Energy*, 85(11):2881–2893, 2011.

[22] Samuel R West, Daniel Rowe, Saad Sayeef, and Adam Berry. Short-

term irradiance forecasting using skycams: Motivation and development. *Solar Energy*, 110:188–207, 2014.

[23] Bijan Nouri, Pascal Kuhn, Stefan Wilbert, Christoph Prahl, Robert Pitz-Paal, Philippe Blanc, Thomas Schmidt, Zeyad Yasser, Lourdes Ramirez Santigosa, and Detlev Heineman. Nowcasting of dni maps for the solar field based on voxel carving and individual 3d cloud objects from all sky images. In *AIP Conference Proceedings*, volume 2033, page 190011. AIP Publishing LLC, 2018.

[24] Guang Wang, Ben Kurtz, and Jan Kleissl. Cloud base height from sky imager and cloud speed sensor. *Solar Energy*, 131:208–221, 2016.

[25] Lydie Magnone, Fabrizio Sossan, Enrica Scolari, and Mario Paolone. Cloud motion identification algorithms based on all-sky images to support solar irradiance forecast. In *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*, pages 1415–1420. IEEE, 2017.

[26] Ricardo Marquez and Carlos FM Coimbra. Intra-hour dni forecasting based on cloud tracking image analysis. *Solar Energy*, 91:327–336, 2013.

[27] Bijan Nouri, Stefan Wilbert, Luis Segura, P Kuhn, Natalie Hanrieder, A Kazantzidis, Thomas Schmidt, L Zarzalejo, Philipp Blanc, and Robert Pitz-Paal. Determination of cloud transmittance for all sky imager based solar nowcasting. *Solar Energy*, 181:251–263, 2019.

[28] Julien Nou, Rémi Chauvin, Julien Eynard, Stéphane Thil, and Stéphane Grieu. Towards the intrahour forecasting of direct normal irradiance using sky-imaging data. *Heliyon*, 4(4):e00598, 2018.

[29] Can Wan, Jian Zhao, Yonghua Song, Zhao Xu, Jin Lin, and Zechun Hu. Photovoltaic and solar power forecasting for smart grid energy

management. *CSEE Journal of Power and Energy Systems*, 1(4):38–46, 2015.

[30] Dhivya Sampath Kumar, Gokhan Mert Yagli, Monika Kashyap, and Dipti Srinivasan. Solar irradiance resource and forecasting: a comprehensive review. *IET Renewable Power Generation*, 14(10):1641–1656, 2020.

[31] Yuhao Nie, Xiatong Li, Quentin Paletta, Max Aragon, Andea Scott, and Adam Brandt. Open-source ground-based sky image datasets for very short-term solar forecasting, cloud analysis and modeling: A comprehensive survey. *arXiv preprint arXiv:2211.14709*, 2022.

[32] Yuchi Sun, Gergely Szűcs, and Adam R Brandt. Solar pv output prediction from video streams using convolutional neural networks. *Energy & Environmental Science*, 11(7):1811–1818, 2018.

[33] Yuchi Sun, Vignesh Venugopal, and Adam R Brandt. Short-term solar power forecast with deep learning: Exploring optimal input and output configuration. *Solar Energy*, 188:730–741, 2019.

[34] Yuhao Nie, Ahmed S Zamzam, and Adam Brandt. Resampling and data augmentation for short-term pv output prediction based on an imbalanced sky images dataset using convolutional neural networks. *Solar Energy*, 224:341–354, 2021.

[35] Quentin Paletta, Anthony Hu, Guillaume Arbod, and Joan Lasenby. Eclipse: Envisioning cloud induced perturbations in solar energy. *arXiv preprint arXiv:2104.12419*, 326:119924, 2021.

[36] Cong Feng, Jie Zhang, Wenqi Zhang, and Bri-Mathias Hodge. Convolutional neural networks for intra-hour solar forecasting based on sky image sequences. *Applied Energy*, 310:118438, 2022.

[37] T Stoffel and A Andreas. Nrel solar radiation research laboratory (srrl): Baseline measurement system (bms); golden, colorado (data). Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 1981.

[38] M Haeffelin, Laurent Barthès, Olivier Bock, C Boitel, S Bony, Dominique Bouniol, H Chepfer, Marjolaine Chiriaco, J Cuesta, Julien Delanoë, et al. Sirta, a ground-based atmospheric observatory for cloud and aerosol research. In *Annales Geophysicae*, volume 23, pages 253–275. Copernicus GmbH, 2005.

[39] Hugo TC Pedro, David P Larson, and Carlos FM Coimbra. A comprehensive dataset for the accelerated development and benchmarking of solar forecasting methods. *Journal of Renewable and Sustainable Energy*, 11(3):036102, 2019.

[40] Yuhao Nie, Xiatong Li, Andea Scott, Yuchi Sun, Vignesh Venugopal, and Adam Brandt. Skipp'd: a sky images and photovoltaic power generation dataset for short-term solar forecasting. *arXiv preprint arXiv:2207.00913*, 2022.

[41] Rémi Chauvin, Julien Nou, Stéphane Thil, and Stéphane Grieu. Generating high dynamic range images using a sky imager. *IFAC-PapersOnLine*, 50(1):219–224, 2017.

[42] Quentin Paletta, Guillaume Arbod, and Joan Lasenby. Benchmarking of deep learning irradiance forecasting models from sky images–an in-depth analysis. *Solar Energy*, 224:855–867, 2021.

[43] Fei Wang, Zhanyao Zhang, Hua Chai, Yili Yu, Xiaoxing Lu, Tieqiang Wang, and Yuzhang Lin. Deep learning based irradiance mapping model for solar pv power forecasting using sky image. In *2019 IEEE industry applications society annual meeting*, pages 1–9. IEEE, 2019.

[44] Omaima El Alani, Mounir Abraim, Hicham Ghennioui, Abdellatif Ghennioui, Ilyass Ikenbi, and Fatima-Ezzahra Dahr. Short term solar irradiance forecasting using sky images based on a hybrid cnn–mlp model. *Energy Reports*, 7:888–900, 2021.

[45] Ruiyuan Zhang, Hui Ma, Tapan Kumar Saha, and Xiaofang Zhou. Photovoltaic nowcasting with bi-level spatio-temporal analysis incorporating sky images. *IEEE Transactions on Sustainable Energy*, 12(3):1766–1776, 2021.

[46] Quentin Paletta and Joan Lasenby. Convolutional neural networks applied to sky images for short-term solar irradiance forecasting. *arXiv preprint arXiv:2005.11246*, 2020.

[47] Xin Zhao, Haikun Wei, Hai Wang, Tingting Zhu, and Kanjian Zhang. 3d-cnn-based feature extraction of ground-based cloud images for direct normal irradiance prediction. *Solar Energy*, 181:510–518, 2019.

[48] Hui-Min Zuo, Jun Qiu, Ying-Hui Jia, Qi Wang, and Fang-Fang Li. Ten-minute prediction of solar irradiance based on cloud detection and a long short-term memory (lstm) model. *Energy Reports*, 8:5146–5157, 2022.

[49] Dazhi Yang. Choice of clear-sky model in solar forecasting. *Journal of Renewable and Sustainable Energy*, 12(2):026101, 2020.

[50] Jinsong Zhang, Rodrigo Verschae, Shohei Nobuhara, and Jean-François Lalonde. Deep photovoltaic nowcasting. *Solar Energy*, 176:267–276, 2018.

[51] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for

recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[52] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[53] Yuhao Nie, Quentin Paletta, Andea Scotta, Luis Martin Pomares, Guillaume Arbod, Sgouris Sgouridis, Joan Lasenby, and Adam Brandt. Sky-image-based solar forecasting using deep learning with multi-location data: training models locally, globally or via transfer learning? *arXiv preprint arXiv:2211.02108*, 2022.

[54] Jane Oktavia Kamadinata, Tan Lit Ken, and Tohru Suwa. Sky image-based solar irradiance prediction methodologies using artificial neural networks. *Renewable Energy*, 134:837–845, 2019.

[55] Weicong Kong, Youwei Jia, Zhao Yang Dong, Ke Meng, and Songjian Chai. Hybrid approaches based on deep whole-sky-image leaing to photovoltaic generation forecasting. *Applied Energy*, 280:115875, 2020.

[56] Talha Ahmad Siddiqui, Samarth Bharadwaj, and Shivkumar Kalyanaraman. A deep learning approach to solar-irradiance forecasting in sky-videos. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2166–2174. IEEE, 2019.

[57] Cong Feng and Jie Zhang. Solarnet: A sky image-based deep convolutional neural network for intra-hour solar forecasting. *Solar Energy*, 204:71–78, 2020.

[58] Anto Ryu, Masakazu Ito, Hideo Ishii, and Yasuhiro Hayashi. Preliminary analysis of short-term solar irradiance forecasting by using total-sky imager and convolutional neural network. In *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*, pages 627–631. IEEE, 2019.

[59] Yuhao Nie, Yuchi Sun, Yuanlei Chen, Rachel Orsini, and Adam Brandt. Pv power output prediction from sky images using convolutional neural network: The comparison of sky-condition-specific sub-models and an end-to-end model. *Journal of Renewable and Sustainable Energy*, 12(4):046101, 2020.

[60] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[61] Qingyong Li, Weitao Lu, and Jun Yang. A hybrid thresholding algorithm for cloud detection on ground-based color images. *Journal of atmospheric and oceanic technology*, 28(10):1286–1296, 2011.

[62] Anna Heinle, Andreas Macke, and Anand Srivastav. Automatic cloud classification of whole sky images. *Atmospheric Measurement Techniques*, 3(3):557–567, 2010.

[63] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[64] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[65] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[66] Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, and Gang Sun. Deep image: Scaling up image recognition. *arXiv preprint arXiv:1501.02876*, 7(8):4, 2015.

[67] Luís Torgo, Paula Branco, Rita P Ribeiro, and Bernhard Pfahringer. Resampling strategies for regression. *Expert Systems*, 32(3):465–476, 2015.

[68] Bryan Urquhart, Mohamed Ghonima, Dung Nguyen, Ben Kurtz, Chi Wai Chow, and Jan Kleissl. Sky-imaging systems for short-term forecasting. *Solar energy forecasting and resource assessment*, pages 195–232, 2013.

[69] Yuchi Sun, Vignesh Venugopal, and Adam R Brandt. Convolutional neural network for short-term solar panel output prediction. In *2018 IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC)(A Joint Conference of 45th IEEE PVSC, 28th PVSEC & 34th EU PVSEC)*, pages 2357–2361. IEEE, 2018.

[70] Dinesh Pothineni, Martin R Oswald, Jan Poland, and Marc Pollefeys. Kloudnet: Deep learning for sky image analysis and irradiance forecasting. In *Pattern Recognition: 40th German Conference, GCPR 2018, Stuttgart, Germany, October 9-12, 2018, Proceedings 40*, pages 535–551. Springer, 2019.

[71] Vignesh Venugopal, Yuchi Sun, and Adam R Brandt. Short-term solar pv forecasting using computer vision: The search for optimal cnn architectures for incorporating sky images and pv generation history. *Journal of Renewable and Sustainable Energy*, 11(6):066102, 2019.

[72] Cong Feng and Jie Zhang. Solarnet: A deep convolutional neural network for solar forecasting via sky images. In *2020 IEEE Power*

& Energy Society Innovative Smart Grid Technologies Conference (ISGT), pages 1–5. IEEE, 2020.

[73] Zhao Zhen, Jiaming Liu, Zhanyao Zhang, Fei Wang, Hua Chai, Yili Yu, Xiaoxing Lu, Tieqiang Wang, and Yuzhang Lin. Deep learning based surface irradiance mapping model for solar pv power forecasting using sky image. *IEEE Transactions on Industry Applications*, 56(4):3385–3396, 2020.

[74] Vincent Le Guen and Nicolas Thome. A deep physical model for solar irradiance forecasting with fisheye images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 630–631, 2020.

[75] Fei Wang, Zhiming Xuan, Zhao Zhen, Yu Li, Kangping Li, Liqiang Zhao, Miadreza Shafie-khah, and João PS Catalão. A minutely solar irradiance forecasting method based on real-time sky image-irradiance mapping model. *Energy Conversion and Management*, 220:113075, 2020.

[76] Huaiguang Jiang, Yi Gu, Yu Xie, Rui Yang, and Yingchen Zhang. Solar irradiance capturing in cloudy sky days–a convolutional neural network based image regression approach. *IEEE Access*, 8:22235–22248, 2020.

[77] Haoran Wen, Yang Du, Xiaoyang Chen, Enggee Lim, Huiqing Wen, Lin Jiang, and Wei Xiang. Deep learning based multistep solar forecasting for pv ramp-rate control using sky images. *IEEE Transactions on Industrial Informatics*, 17(2):1397–1406, 2020.

[78] Tingting Zhu, Yiren Guo, Zhenye Li, and Cong Wang. Solar radiation prediction based on convolution neural network and long short-term memory. *Energies*, 14(24):8498, 2021.

[79] Zhao Zhen, Xuemin Zhang, Shengwei Mei, Xiqiang Chang, Hua Chai, Rui Yin, and Fei Wang. Ultra-short-term irradiance forecasting model based on ground-based cloud image and deep learning algorithm. *IET Renewable Power Generation*, 16(12):2604–2616, 2022.

[80] Lei Chen, Yangluxi Li, Hu Du, and Yukun Lai. Solar radiation nowcasting through advanced cnn model integrated with resnet structure. 2021.

[81] Mingjun Xiang, Wenkang Cui, Can Wan, and Changfei Zhao. A sky image-based hybrid deep learning model for nonparametric probabilistic forecasting of solar irradiance. In *2021 International Conference on Power System Technology (POWERCON)*, pages 946–952. IEEE, 2021.

[82] Hao Yang, Long Wang, Chao Huang, and Xiong Luo. 3d-cnn-based sky image feature extraction for short-term global horizontal irradiance forecasting. *Water*, 13(13):1773, 2021.

[83] IM Insaf, HMKD Wickramathilaka, MAN Upendra, GMRI Godaliyadda, MPB Ekanayake, HMVR Herath, DMLH Dissawa, and JB Ekanayake. Global horizontal irradiance modeling from sky images using resnet architectures. In *2021 IEEE 16th International Conference on Industrial and Information Systems (ICIIS)*, pages 239–244. IEEE, 2021.

[84] Javier Huertas-Tato, Inés M Galván, Ricardo Aler, Francisco Javier Rodríguez-Benítez, and David Pozo-Vázquez. Using a multi-view convolutional neural network to monitor solar irradiance. *Neural Computing and Applications*, pages 1–13, 2021.

[85] Xinyang Zhang, Zhao Zhen, Yiqian Sun, Yagang Zhang, Hui Ren, Hui Ma, Jian Yang, and Fei Wang. Solar irradiance prediction interval

estimation and deterministic forecasting model using ground-based sky image. In *2022 IEEE/IAS 58th Industrial and Commercial Power Systems Technical Conference (I&CPS)*, pages 1–8. IEEE, 2022.

[86] Amirhossein Dolatabadi, Hussein Hassan Abdeltawab, and Yasser Abdel-Rady I Mohamed. Deep reinforcement learning-based self-scheduling strategy for a caes-pv system using accurate sky images-based forecasting. *IEEE Transactions on Power Systems*, 2022.

[87] Prajowal Manandhar, Marouane Temimi, and Zeyar Aung. Short-term solar radiation forecast using total sky imager via transfer learning. *Energy Reports*, 9:819–828, 2023.

[88] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[89] Wenzhong Guo, Jianwen Wang, and Shiping Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019.

[90] Nitish Srivastava and Ruslan Salakhutdinov. Learning representations for multimodal data with deep belief nets. In *International conference on machine learning workshop*, volume 79, pages 978–1, 2012.

[91] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 251–260, 2010.

[92] 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.

[93] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 299–307, 2017.

[94] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.

[95] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[96] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[97] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[98] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[99] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep

residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2014.

[100] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[101] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

[102] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[103] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.

[104] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 574–584, 2022.

[105] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

[106] Huiyu Gao and Miaomiao Liu. Short-term solar irradiance prediction from sky images with a clear sky model. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2475–2483, 2022.

[107] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[108] Mennatullah Siam, Sepehr Valipour, Martin Jagersand, and Nilanjan Ray. Convolutional gated recurrent networks for video segmentation. In *2017 IEEE international conference on image processing (ICIP)*, pages 3090–3094. IEEE, 2017.

[109] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

[110] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020.

[111] Guangyu Robert Yang and Xiao-Jing Wang. Artificial neural networks for neuroscientists: a primer. *Neuron*, 107(6):1048–1070, 2020.

[112] Xiang Long, Chuang Gan, Gerard Melo, Xiao Liu, Yandong Li, Fu Li, and Shilei Wen. Multimodal keyless attention fusion for video classification. In *Proceedings of the aaai conference on artificial intelligence*, volume 32, 2018.

[113] Varun Godbole, George E. Dahl, Justin Gilmer, Christopher J. Shallue, and Zachary Nado. Deep learning tuning playbook, 2023. Version 1.0.

[114] Tadayoshi Fushiki. Estimation of prediction error by using k-fold cross-validation. *Statistics and Computing*, 21:137–146, 2011.

[115] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[116] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[117] Yinghao Chu, Mengying Li, Carlos FM Coimbra, Daquan Feng, and Huaizhi Wang. Intra-hour irradiance forecasting techniques for solar power integration: A review. *Iscience*, 24(10):103136, 2021.

[118] Loïc Vallance, Bruno Charbonnier, Nicolas Paul, Stéphanie Dubost, and Philippe Blanc. Towards a standardized procedure to assess solar forecast accuracy: A new ramp and time alignment metric. *Solar Energy*, 150:408–422, 2017.

[119] Anthony Florita, Bri-Mathias Hodge, and Kirsten Orwig. Identifying wind and solar ramping events. In *2013 IEEE Green Technologies Conference (GreenTech)*, pages 147–152. IEEE, 2013.

[120] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, et al. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

[121] Vincent Le Guen. Deep learning for spatio-temporal forecasting–application to solar energy. *arXiv e-prints*, pages arXiv–2205, 2022.

[122] Dazhi Yang. Estimating 1-min beam and diffuse irradiance from the global irradiance: A review and an extensive worldwide comparison of latest separation models at 126 stations. *Renewable and Sustainable Energy Reviews*, 159:112195, 2022.

[123] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[124] Rich H Inman, Hugo TC Pedro, and Carlos FM Coimbra. Solar forecasting methods for renewable energy integration. *Progress in energy and combustion science*, 39(6):535–576, 2013.

[125] Anna-Lena Klingler and Lukas Teichtmann. Impacts of a forecast-based operation strategy for grid-connected pv storage systems on profitability and the energy system. *Solar Energy*, 158:861–868, 2017.

[126] Dazhi Yang, Stefano Alessandrini, Javier Antonanzas, Fernando Antonanzas-Torres, Viorel Badescu, Hans Georg Beyer, Robert Blaga, John Boland, Jamie M Bright, and Carlos FM Coimbra. Verification of deterministic solar forecasts. *Solar Energy*, 210:20–37, 2020.

[127] Utpal Kumar Das, Kok Soon Tey, Mehdi Seyedmahmoudian, Saad Mekhilef, Moh Yamani Idna Idris, Willem Van Deventer, Bend Horan, and Alex Stojcevski. Forecasting of photovoltaic power generation and model optimization: A review. *Renewable and Sustainable Energy Reviews*, 81:912–928, 2018.

[128] Christian A Gueymard and Jose A Ruiz-Arias. Extensive worldwide validation and climate sensitivity analysis of direct irradiance predictions from 1-min global irradiance. *Solar Energy*, 128:1–30, 2016.

[129] Dazhi Yang, Weixing Li, Gokhan Mert Yagli, and Dipti Srinivasan. Operational solar forecasting for grid integration: Standards, challenges, and outlook. *Solar Energy*, 224:930–937, 2021.

[130] Hugo TC Pedro, Carlos FM Coimbra, Mathieu David, and Philippe Lauret. Assessment of machine learning techniques for deterministic and probabilistic intra-hour solar forecasts. *Renewable Energy*, 123:191–203, 2018.

[131] Ardan Hüseyin Eşlik, Emre Akarslan, and Fatih Onur Hocaoğlu. Short-term solar radiation forecasting with a novel image processing-based deep learning approach. *Renewable Energy*, 200:1490–1505, 2022.

[132] M Caldas and R Alonso-Suárez. Very short-term solar irradiance forecast using all-sky imaging and real-time irradiance measurements. *Renewable energy*, 143:1643–1658, 2019.

[133] Stavros-Andreas Logothetis, Vasileios Salamalikis, Stefan Wilbert, Jan Remund, Luis F Zarzalejo, Yu Xie, Bijan Nouri, Evangelos Ntavelis, Julien Nou, Niels Hendrikx, et al. Benchmarking of solar irradiance nowcast performance derived from all-sky imagers. *Renewable Energy*, 199:246–261, 2022.

[134] D Anagnostos, T Schmidt, S Cavadias, D Soudris, J Poortmans, and F Catthoor. A method for detailed, short-term energy yield forecasting of photovoltaic installations. *Renewable Energy*, 130:122–129, 2019.

[135] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

[136] Jun Yu, Jing Li, Zhou Yu, and Qingming Huang. Multimodal transformer with multi-view visual representation for image captioning. *IEEE transactions on circuits and systems for video technology*, 30(12):4467–4480, 2019.

[137] Jian Huang, Jianhua Tao, Bin Liu, Zheng Lian, and Mingyue Niu. Multimodal transformer fusion for continuous emotion recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3507–3511. IEEE, 2020.

[138] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34:24206–24221, 2021.

[139] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2019, page 6558. NIH Public Access, 2019.

[140] Mireille Lefevre, Armel Oumbe, Philippe Blanc, Bella Espinar, Benoît Gschwind, Zhipeng Qu, Lucien Wald, Marion Schroedter-Homscheidt, Carsten Hoyer-Klick, and Antti Arola. Mcclear: a new model estimating downwelling solar radiation at ground level in clear-sky conditions. *Atmospheric Measurement Techniques*, 6(9):2403–2418, 2013.

[141] Dazhi Yang and Christian A Gueymard. Ensemble model output

statistics for the separation of direct and diffuse components from 1-min global irradiance. *Solar Energy*, 208:591–603, 2020.

[142] Chuck N Long and Yan Shi. An automated quality assessment and control algorithm for surface radiation measurements. *The Open Atmospheric Science Journal*, 2(1), 2008.

[143] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[144] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[145] Abdul Rahim Pazikadin, Damhuji Rifai, Kharudin Ali, Muhammad Zeesan Malik, Ahmed N Abdalla, and Moneer A Faraj. Solar irradiance measurement instrumentation and power solar generation forecasting based on artificial neural networks (ann): A review of five years research trend. *Science of The Total Environment*, 715:136848, 2020.

[146] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[147] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*, pages 214–229. Springer, 2020.

[148] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, PMLR, 2021.

[149] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

[150] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[151] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*, 2018.

[152] Mohamed Abuella and Badrul Chowdhury. Forecasting of solar power ramp events: A post-processing approach. *Renewable Energy*, 133:1380–1392, 2019.

[153] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[154] Francois Chollet et al. Keras, 2015.

[155] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th Eu-*

ropean Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13, pages 818–833. Springer, 2014.

[156] LI-COR Biosciences UK Ltd. *Light Sensor Brochure 15539, LI-200R*, 2023.

[157] Kipp & Zonen B.V. *Instruction Manual, RaZON+ Solar Monitoring System*, 2023.

[158] Stefan Wilbert, Wilko Jessen, Anne Forstinger, Anton Driesse, Aron M Habte, Manajit Sengupta, Aitor Marzo, Frank Vignola, and Luis Zarzlejo. Application of the clear-sky spectral error for radiometer classification in iso 9060. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2019.

[159] International Organization for Standardization. Solar energy — Specification and classification of instruments for measuring hemispherical solar and direct solar radiation. Standard, International Organization for Standardization, Geneva, CH, March 2018.

[160] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*, pages 270–279. Springer, 2018.

[161] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[162] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell,

Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[163] Matus Telgarsky. Benefits of depth in neural networks. In *Conference on learning theory*, pages 1517–1539. PMLR, 2016.

[164] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[165] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[166] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

[167] Holger H Hoos. Automated algorithm configuration and parameter tuning. *Autonomous search*, pages 37–71, 2012.

[168] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[169] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[170] Leslie N Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[171] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[172] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade: Second Edition*, pages 437–478, 2012.

[173] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[174] Christopher J Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E Dahl. Measuring the effects of data parallelism on neural network training. *arXiv preprint arXiv:1811.03600*, 2018.

Appendices

Appendix A

Examining the Impact of Dataset on Model Performance

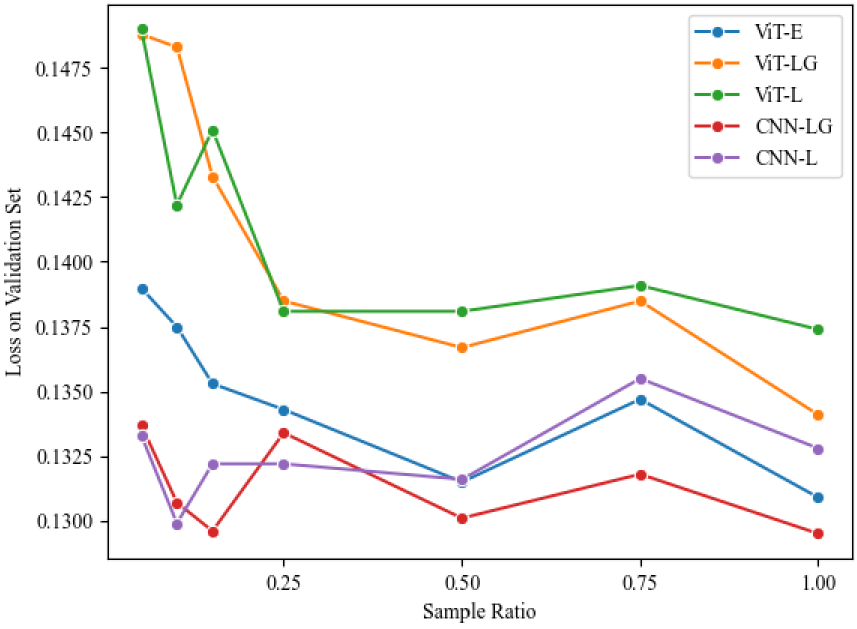


Figure A.1: Sampling rate validation experiments on Folsom dataset with ViT-E model. The training set was used to train five different models with sampling rates of 0.05, 0.1, 0.15, 0.25, 0.5, 0.75 and 1.0. The models were then validated under the same validation set. The model loss tends to flatten out above 0.25 sample ratio.

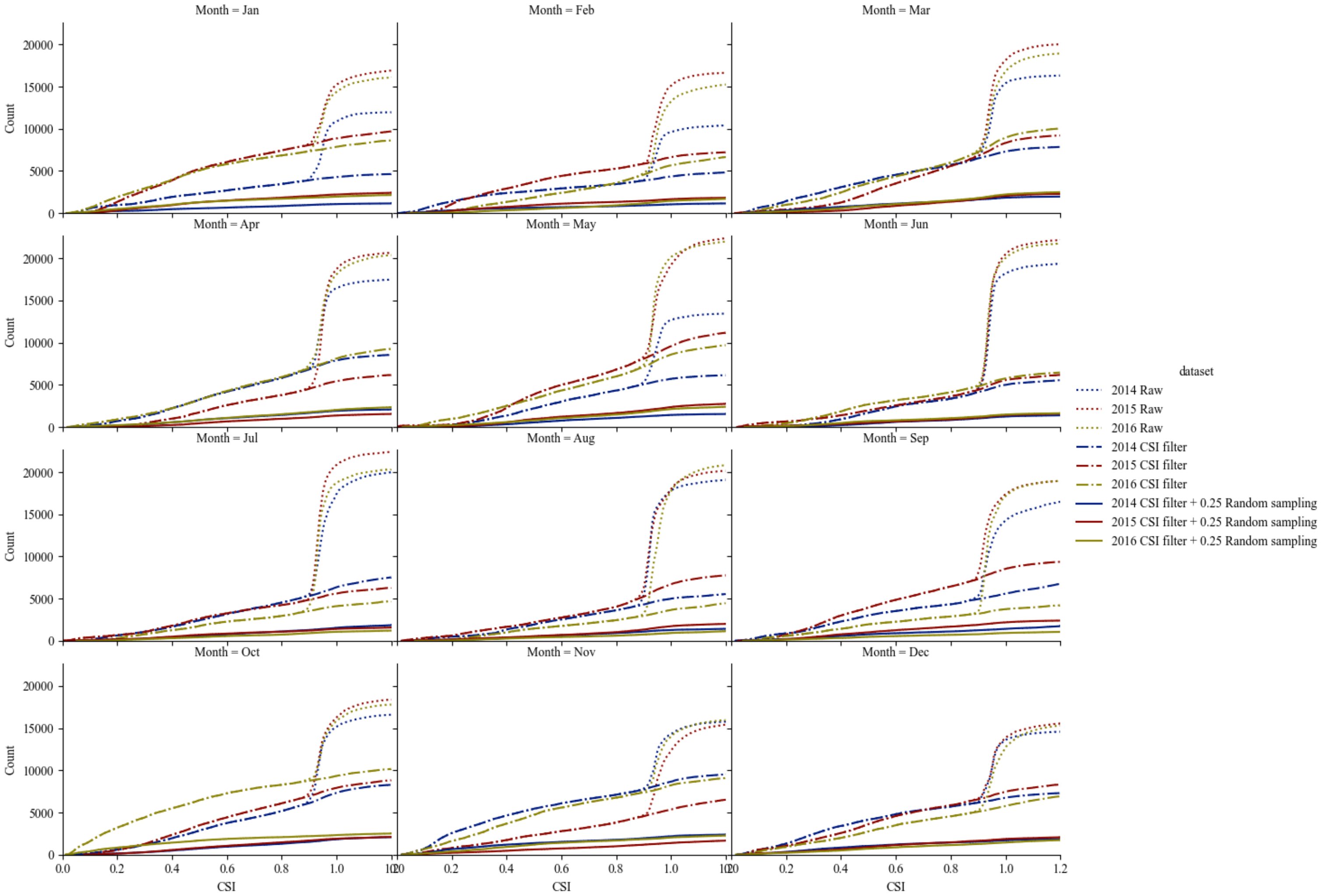


Figure A.2: Monthly CSI distribution of raw data on Folsom dataset, compared to Clear sky filtered data and 25% randomly sampled filtered data.

Appendix B

Model Algorithm and Architecture

Algorithm 1 Late Feature-level fusion

```
for input $[\mathbf{x}_{\text{numerical}}, \mathbf{x}_{\text{image}}]$ $in$ $(\mathbf{X}_{\text{numerical}}, \mathbf{X}_{\text{image}})$ do
    $\mathbf{x}_{\text{numerical feature}} = \text{MLP}(\mathbf{x}_{\text{numerical}})$
    if Image feature extractor is CNN (model CNN-L(G)) then
        $\mathbf{x}_{\text{image feature}} = \text{MLP}(\mathbf{x}_{\text{image}})$
    else if Image feature extractor is ViT (model ViT-L(G)) then
        $\mathbf{x}_{\text{image feature}} = \text{ViT}(\mathbf{x}_{\text{image}})$
    end if
    $\mathbf{x}_{\text{fusion feature}} = concat(\mathbf{x}_{\text{image feature}}, \mathbf{x}_{\text{numerical feature}})$
    if Gated model (CNN-LG or ViT-LG) then
        $\mathbf{f}_{\text{gated factor}} = \text{MLP}(\mathbf{x}_{\text{fusion feature}},\ \text{activation function} = \tanh)$
        $\mathbf{x}_{\text{gated fusion feature}} = \mathbf{x}_{\text{fusion feature}} \times \mathbf{f}_{\text{gated factor}}$
        $x = \text{MLP}(\mathbf{x}_{\text{gated fusion feature}})$
    else if non-Gated model (CNN-L or ViT-L) then
        $x = \text{MLP}(\mathbf{x}_{\text{image feature}})$
    end if
    return Output $x$
end for
```

Algorithm 2 Early Feature-level fusion

```
for input $[\mathbf{x}_{\text{numerical}}, \mathbf{x}_{\text{image}}]$ $in$ $(\mathbf{X}_{\text{numerical}}, \mathbf{X}_{\text{image}})$ do

    Patching images and Linear projection from $\mathbf{x}_{\text{image}}$ to $\mathbf{x}_p$ (Eq 4.8)
    Concatenating $\mathbf{x}_p$ with an additional learnable token $\mathbf{x}_{\text{class}}$ (Eq 4.8)
    Superimposing $\mathbf{x}_p$ with a learnable position matrix $\mathbf{E}_{\text{pos}}$ become $\mathbf{z}_{i0}$(Eq 4.8)

    Linear projection numerical input from $\mathbf{x}_{\text{numerical}}$ to MLP($\mathbf{y}^{\text{M}}$) (Eq 4.9)
    Concatenating MLP($\mathbf{y}^{\text{M}}$) with an additional learnable token $\mathbf{y}_{\text{class}}$ (Eq 4.9)
    Superimposing MLP($\mathbf{y}^{\text{M}}$) with a learnable sequence matrix $\mathbf{E}_{\text{seq}}$ become $\mathbf{z}_{n0}$(Eq 4.9)

    Superimposing $\mathbf{z}_{i0}$, $\mathbf{z}_{n0}$ with learnable modality type matrix $\mathbf{z}_i^{\text{type}}$, $\mathbf{z}_n^{\text{type}}$, respectively (Eq 4.10)

    $\mathbf{x}_{\text{fusion feature}} = \text{ViT}([\mathbf{z}_{\text{i}0},\ \mathbf{z}_{\text{n}0}])$ (Eq 4.11 4.12 4.13)
    $x = \text{MLP}([\mathbf{x}_{\text{class}},\ \mathbf{y}_{\text{class}}])$
    return Output $x$
end for
```

Table B.1: Hyperparameters of the SGD optimizer for training models

Hyperparameters	**CNN-L**	**CNN-LG**	**ViT-L**	**ViT-LG**	**ViT-E**
Learning rate	0.01	0.01	0.0008	0.0008	0.0008
Optimiser	SGD	SGD	SGD	SGD	SGD
Optimiser momentum	0.9	0.9	0.9	0.9	0.9
Loss	MSE	MSE	MSE	MSE	MSE
Weight decay	0.0001	0.0001	0.0001	0.0001	0.0001
Batch size	64	64	8	8	8
Training epochs	80	80	80	80	80
Warm up percentage	25%	25%	0	0	0
Learning rate decay	Cosine	Cosine	Cosine	Cosine	Cosine
Early stop	True	True	True	True	True
Early stop tolerance	20	20	20	20	20

Table B.2: The details of ViT-E model

Block	Layer	Resolution	Channels
Image Inputs	-	128 × 128 × 3	1
Image Patch Embedding	Conv 8 × 8	128 × 128 × 3 → 8 × 8 × 3	1 → 256
Image Class Token	Transfer Embedding Projection	8 × 8 × 3 → 192	256 → 256
	Class Token Concat	192	256 → 257
Position Embedding	Position Embedding	192	257
Numerical Inputs	-	14 (3 + 3 + 3 + 2 + 3)	1
Numerical Class Token	Numerical Projection (MLP)	14 → 192	5
	Class Token Concat	192	5 → 6
Sequence Embedding	Sequence Embedding	192	6
Concatenation	Concat	192	263 (257 + 6)
Attention Block × 12	LayerNorm	192	263
	Multi-Head Attention × 12	192	263
	Add (residual connection)	192	263
	LayerNorm	192	263
	Multi-Head Attention × 12	192	263
	Add (residual connection)	192	263
Layer Normalization	LayerNorm	192	263
Regression Head	Extract Class Token	384	1
	MLP	768	1
	MLP	512	1
	MLP	64	1
	MLP	1	1

Table B.3: The details of ViT-LG model.

Block	Layer	Resolution	Channels
Image Inputs	-	128 × 128 × 3	1
Image Patch Embedding	Conv8 × 8	128 × 128 × 3 → 8 × 8 × 3	1 → 256
Image Class Token	Transfer Embedding Projection	8 × 8 × 3	256 → 256
	Class Token Concat	8 × 8 × 3	256 → 257
Position Embedding	Position Embedding	8 × 8 × 3	257
Image Attention Block × 12	LayerNorm	192	257
	Multi-Head Attention × 12	192	257
	Add (residual connection)	192	257
	LayerNorm	192	257
	Multi-Head Attention × 12	192	257
	Add (residual connection)	192	257
Image Feature Vectorisation	Extract Class Token	192	1
	MLP	768	1
	MLP	64	1
Numerical Inputs	-	14 (3 + 3 + 3 + 2 + 3)	1
Numerical Feature Vectorisation	MLP	14 → 16	1
	MLP	16	1
Concatenation	Concat	80 (64 + 16)	1
Regression Head	MLP	80	1
	Gate MLP	80	1
	Gate Multiply	80	1
	MLP	64	1
	MLP	16	1
	MLP	1	1

Table B.4: The details of ViT-L model.

Block	**Layer**	**Resolution**	**Channels**
Image Inputs	-	128 × 128 × 3	1
Image Patch Embedding	Conv 8 × 8	128 × 128 × 3 → 8 × 8 × 3	1 → 256
Image Class Token	Transfer Embedding Projection	8 × 8 × 3	256 → 256
	Class Token Concat	8 × 8 × 3	256 → 257
Position Embedding	Position Embedding	8 × 8 × 3	257
Image Attention Block × 12	LayerNorm	192	257
	Multi-Head Attention × 12	192	257
	Add (residual connection)	192	257
	LayerNorm	192	257
	Multi-Head Attention × 12	192	257
	Add(residual connection)	192	257
Image Feature Vectorization	Extract Class Token	192	1
	MLP	768	1
	MLP	64	1
Numerical Inputs	-	14 (3 + 3 + 3 + 2 + 3)	1
Numerical Feature Vectorization	MLP	14 → 16	1
	MLP	16	1
Concatenation	Concat	80 (64 + 16)	1
Regression Head	MLP	80	1
	MLP	64	1
	MLP	16	1
	MLP	1	1

Table B.5: The details of CNN-LG model.

Block	**Layer**	**Resolution**	**Channels**
Image Inputs	-	128 × 128 × 3	1
ResNet Block Conv 1	Conv 7 × 7	128 × 128 × 3 → 64 × 64 × 3	1 → 64
	Max Pooling 3 × 3	64 × 64 × 3 → 32 × 32 × 3	64
ResNet Block Conv 2 × 2	Conv 3 × 3	32 × 32 × 3	64
	BatchNormal	32 × 32 × 3	64
	Conv 3 × 3	32 × 32 × 3	64
	BatchNormal	32 × 32 × 3	64
	Add (residual connection)	32 × 32 × 3	64
ResNet Block Conv 3 × 2	Conv 3 × 3	32 × 32 × 3 → 16 × 16 × 3	64 → 128
	BatchNormal	16 × 16 × 3	128
	Conv 3 × 3	16 × 16 × 3	128
	BatchNormal	16 × 16 × 3	128
	Add(residual connection)	16 × 16 × 3	128
ResNet Block Conv 4 × 2	Conv 3 × 3	16 × 16 × 3 → 8 × 8 × 3	128 → 256
	BatchNormal	8 × 8 × 3	256
	Conv 3 × 3	8 × 8 × 3	256
	BatchNormal	8 × 8 × 3	256
	Add (residual connection)	8 × 8 × 3	256
ResNet Block Conv 5 × 2	Conv 3 × 3	8 × 8 × 3 → 4 × 4 × 3	256 → 512
	BatchNormal	4 × 4 × 3	512
	Conv 3 × 3	4 × 4 × 3	512
	BatchNormal	4 × 4 × 3	512
	Add(residual connection)	4 × 4 × 3	512
Image Feature Transformation	Global Average Pooling	512	1
	MLP	64	1
Numerical Inputs	-	14 (3 + 3 + 3 + 2 + 3)	1
Numerical Feature Transformation	MLP	14 → 16	1
	MLP	16	1
Concatenation	Concat	80 (64 + 16)	1
Regression Head	MLP	80	1
	Gate MLP	80	1
	Gate Multiply	80	1
	MLP	64	1
	MLP	16	1
	MLP	1	1

Table B.6: The details of CNN-L model.

Block	**Layer**	**Resolution**	**Channels**
Image Inputs	-	128 × 128 × 3	1
ResNet Block Conv 1	Conv 7 × 7	128 × 128 × 3 → 64 × 64 × 3	1 → 64
	Max Pooling 3 × 3	64 × 64 × 3 → 32 × 32 × 3	64
ResNet Block Conv 2 × 2	Conv 3 × 3	32 × 32 × 3	64
	BatchNormal	32 × 32 × 3	64
	Conv 3 × 3	32 × 32 × 3	64
	BatchNormal	32 × 32 × 3	64
	Add (residual connection)	32 × 32 × 3	64
ResNet Block Conv 3 × 2	Conv 3 × 3	32 × 32 × 3 → 16 × 16 × 3	64 → 128
	BatchNormal	16 × 16 × 3	128
	Conv 3 × 3	16 × 16 × 3	128
	BatchNormal	16 × 16 × 3	128
	Add (residual connection)	16 × 16 × 3	128
ResNet Block Conv 4 × 2	Conv 3 × 3	16 × 16 × 3 → 8 × 8 × 3	128 → 256
	BatchNormal	8 × 8 × 3	256
	Conv 3 × 3	8 × 8 × 3	256
	BatchNormal	8 × 8 × 3	256
	Add (residual connection)	8 × 8 × 3	256
ResNet Block Conv 5 × 2	Conv 3 × 3	8 × 8 × 3 → 4 × 4 × 3	256 → 512
	BatchNormal	4 × 4 × 3	512
	Conv 3 × 3	4 × 4 × 3	512
	BatchNormal	4 × 4 × 3	512
	Add (residual connection)	4 × 4 × 3	512
Image Feature Transformation	Global Average Pooling	512	1
	MLP	64	1
Numerical Inputs	-	14 (3 + 3 + 3 + 2 + 3)	1
Numerical Feature Transformation	MLP	14 → 16	1
	MLP	16	1
Concatenation	Concat	80 (64 + 16)	1
Regression Head	MLP	80	1
	MLP	64	1
	MLP	16	1
	MLP	1	1