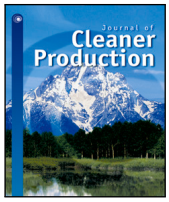
Journal of Cleaner Production 468 (2024) 143056

Contents lists available at ScienceDirect

Journal of Cleaner Production

journal homepage: www.elsevier.com/locate/jclepro



Explainable time-varying directional representations for photovoltaic power generation forecasting

Zhijin Wang [a], Hanjing Liu [a], Senzhen Wu [a], Niansheng Liu [a,*], Xiufeng Liu [c,*], Yue Hu [a,b], Yonggang Fu [a]

[a] College of Computer Engineering, Jimei University, Yinjiang Road 185, 361021 Xiamen, China
[b] Chengyi College, Jimei University, Jimei Road 199, 361021 Xiamen, China
[c] Department of Technology, Management and Economics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Handling Editor: Panos Seferlis

Keywords:
Photovoltaic power forecasting
Renewable energy
Collaborative directional representation
Explainability
Neural networks

ABSTRACT

Accurate photovoltaic (PV) power generation forecasting is crucial for optimizing the integration of solar energy into power grids and advancing towards a cleaner, more sustainable energy future. However, the inherent variability and complexity of PV power generation data pose significant challenges for accurate forecasting. To address these challenges, this paper introduces Collaborative Directional Representation (CoDR), a novel deep learning model that extracts and represents the directional fluctuations of solar irradiance data to improve forecasting accuracy and reliability. CoDR utilizes a series of steps, including data preprocessing, fluctuation extraction, directional representation, linearization, de-extraction, mapping, and data postprocessing, to capture both the temporal and spatial dependencies within the data. CoDR leverages a unique directional representation to capture both temporal and spatial dependencies in the data, enabling superior forecasting accuracy and robustness compared to twenty-two state-of-the-art benchmark methods. We validate CoDR on a real-world dataset of PV power generation in Belgium, demonstrating its effectiveness through extensive experiments, ablation studies, and sensitivity analyses. Importantly, CoDR enhances the transparency of the forecasting process by revealing the causal relationships and directional influences among input variables and PV power output. This explainability feature provides valuable insights into the underlying drivers of PV power generation, promoting trust and informed decision-making in the transition to cleaner energy systems.

1. Introduction

The global demand for renewable energy sources has increased significantly in recent years, driven by the urgent need to mitigate climate change and reduce greenhouse gas emissions. Transitioning towards a cleaner and more sustainable energy system is paramount for achieving global climate goals, as outlined in the United Nations Sustainable Development Goals (SDGs), particularly SDG 7 (Affordable and Clean Energy) and SDG 13 (Climate Action). Solar PV power generation is one of the most widely adopted and rapidly growing renewable energy technologies due to its abundant availability, low maintenance costs, and high scalability (Mekhilef et al., 2011). However, PV power generation also faces several challenges, such as high variability, uncertainty, and intermittency, which are caused by the dependence on weather conditions, solar irradiance, and other factors that are hard to predict (Peng et al., 2020). These challenges pose serious risks to the stability and reliability of the power grid, as well as for the operation and management of various energy systems and markets (Guo et al., 2022). Accurately forecasting PV power generation is not only essential for the technical optimization of power grids but also plays a critical role in maximizing the environmental and societal benefits of solar energy. By enabling the efficient integration of solar energy, accurate PV forecasting directly contributes to reducing greenhouse gas emissions and decreasing reliance on fossil fuels, fostering a cleaner energy future.

PV power generation forecasting is a challenging task that requires addressing multiple sources of complexity, such as temporal and spatial variations, nonlinear and dynamic dependencies, and diverse and uncertain factors (Das et al., 2018). Various forecasting methodologies have been proposed and applied in the literature, covering a wide spectrum of approaches from traditional statistical models to modern

* Corresponding authors.
E-mail addresses: zhijin@jmu.edu.cn (Z. Wang), hanjingliu@jmu.edu.cn (H. Liu), 202221331009@jmu.edu.cn (S. Wu), nsliu@jmu.edu.cn (N. Liu), xiuli@dtu.dk (X. Liu), yuehu@jmu.edu.cn (Y. Hu), yonggangfu@jmu.edu.cn (Y. Fu).

https://doi.org/10.1016/j.jclepro.2024.143056
Received 3 March 2024; Received in revised form 25 June 2024; Accepted 28 June 2024
Available online 2 July 2024

machine learning techniques and hybrid approaches that combine the strengths of different methods. However, each approach has its own advantages and limitations, and no consensus exists on the best method for PV power generation forecasting (Akhter et al., 2019).

A comparative analysis of existing forecasting methods reveals common challenges and trade-offs that must be addressed to advance the state-of-the-art in PV power forecasting. One challenge is the inherent uncertainty associated with weather conditions, which directly impacts the quality and availability of input data for forecasting models. While uncertainty can be quantified using probabilistic or interval-based approaches to provide valuable information for decision-making and risk management (Asif et al., 2019; Mellit et al., 2020; Ahmad et al., 2024; Shi et al., 2024), these approaches often increase model complexity and computational cost, potentially limiting their applicability in real-time scenarios. Another significant challenge lies in selecting and engineering optimal input variables and features. Different methods employ various types of data, including historical PV power, meteorological data, satellite images, and Numerical Weather Prediction (NWP) data (Yadav et al., 2015; Peng et al., 2024). The choice of input data depends on factors such as availability, reliability, resolution, cost, forecasting horizon, and granularity (Guo et al., 2022). While feature selection and extraction techniques can reduce dimensionality and noise (Wu et al., 2022; Trivedi et al., 2022), effectively capturing the complex interplay between meteorological conditions and PV power output while ensuring computational efficiency remains a persistent challenge. Furthermore, balancing accuracy and interpretability is crucial, especially in promoting cleaner production and responsible AI development. Although machine learning algorithms often demonstrate superior accuracy in short-term, high-resolution forecasting compared to traditional statistical models (Mellit et al., 2020; Gaboitaolelwe et al., 2023; Alcañiz et al., 2023), they are often criticized for their lack of transparency and interpretability. This "black-box" nature can hinder their acceptance and trustworthiness among stakeholders and regulators. Conversely, traditional statistical models, while offering greater interpretability, may not adequately capture the complex, nonlinear relationships and dynamics inherent in PV power generation, potentially limiting their predictive accuracy (López Santos et al., 2022; Hussain et al., 2022). Striking a balance between accuracy and interpretability is essential to ensure reliable forecasts and a clear understanding of the factors driving PV power generation.

To address these challenges, this paper introduces a novel deep learning approach based on CoDR. In contrast to conventional deep learning approaches that often treat time series data as a sequence of scalar values, CoDR leverages a unique directional representation to encode both the magnitude and direction of power variations over time. This directional representation captures the inherent trends and fluctuations in solar irradiance, enabling CoDR to model the complex interplay of these variations with greater accuracy. Crucially, CoDR's directional representation facilitates the extraction of causal relationships and influences between input variables (such as weather conditions) and PV power output, enhancing the model's explainability and transparency. Our CoDR-based model consists of three key stages: data preprocessing, feature extraction, and forecasting. The data preprocessing stage transforms the raw data of multiple PV plants into directional vectors using a simple transformation function. The feature extraction stage leverages the historical data of multiple PV plants to learn a low-dimensional latent representation of the directional vectors, which can capture the common and individual characteristics of each PV plant, as well as handle missing and noisy data. The forecasting stage uses the latent representation as input and predicts the future directional vectors for each PV plant using a linear regression model, which can adapt to different weather conditions and scenarios. The predicted directional vectors are then converted back to PV power values using the inverse transformation function.

The main contributions of this paper are as follows:

- We propose a novel collaborative directional representation for PV power generation data, which can encode both the trend and the intensity of the power variation over time, and can effectively handle the temporal patterns, dependencies, and correlations of PV power generation.
- We develop a novel deep learning model for PV power generation forecasting, which leverages collaborative directional representation to capture both temporal and spatial dependencies in the data. We also design a simple transformation function to convert the raw data into directional vectors and vice versa.
- We provide a theoretical analysis of the proposed CoDR model, which explains how the directional representation can capture the causal relationships and directional influences among the input variables and the output variable. We also derive the upper bound of the forecasting error and prove the convergence of the model.
- We conduct comprehensive experiments on real-world data from various regions of Belgium to evaluate the performance of our model and compare it with several state-of-the-art methods. We demonstrate that our model can achieve superior accuracy and robustness performance, as well as high computational efficiency and scalability.

The remainder of this paper is organized as follows: Section 2 reviews related work on PV power generation forecasting methods; Section 3 introduces the real-world PV power generation data used in this study; Section 5 describes our proposed deep learning model for PV power generation forecasting; Section 6 presents our experimental results and analysis; Section 7 concludes our paper and outlines future work.

2. Related work

Accurately forecasting PV power generation is essential for the effective management and integration of solar energy into power grids, contributing significantly to a cleaner and more sustainable energy future (Das et al., 2018). Reliable forecasts enable optimized power system operations, reduce the risk of energy imbalances, and maximize the social, economic, and environmental benefits associated with this renewable energy source. However, forecasting PV power generation presents significant challenges due to the inherent uncertainty, variability, and complexity of solar irradiance patterns, which are influenced by a multitude of meteorological and environmental factors (Panda et al., 2022). Various forecasting methods have been proposed, ranging from traditional statistical models to advanced machine learning techniques. This section reviews the most relevant recent works, focusing on their strengths, limitations, and how they relate to our proposed approach.

2.1. Deep learning for PV power forecasting

Deep learning models have gained significant attention for their ability to learn complex nonlinear relationships in data, leading to advancements in PV power forecasting. Recurrent neural networks (RNNs), such as Long Short-Term Memory (LSTM) by Hochreiter and Schmidhuber (1997) and Gated Recurrent Units (GRU) by Chung et al. (2014), are well-suited for handling sequential data like time series. These models have shown promise in PV power forecasting due to their ability to capture temporal dependencies. However, RNNs often face challenges in capturing long-range dependencies and can be computationally expensive, especially when dealing with long time series data (Sun et al., 2023). Moreover, traditional RNN architectures often process time series data as a sequence of scalar values, without explicitly considering the directional nature of changes in solar irradiance. This limitation can hinder their ability to accurately model the dynamics of PV power generation, as the direction of solar irradiance fluctuations can significantly influence the output of PV systems (López Santos et al., 2022; Hussain et al., 2022).

Convolutional Neural Networks (CNNs) excel at extracting spatial features from data. In the context of PV forecasting, 1D CNNs by Malek et al. (2018) are employed to extract relevant features from time series data. They are often combined with RNNs (e.g., CNN-RNN, CNN-LSTM) to leverage the strengths of both architectures, capturing both spatial and temporal dependencies crucial for accurate forecasting (Shi et al., 2015). While CNNs offer advantages in feature extraction, they may not explicitly account for the directionality of changes in solar irradiance, similar to traditional RNN approaches. This limitation can restrict their ability to fully model the complex relationship between solar irradiance fluctuations and PV power output.

Attention mechanisms (Vaswani et al., 2017), initially prominent in natural language processing, have been successfully adapted for time series forecasting (Qin et al., 2017; Zhou et al., 2021). These mechanisms allow models to selectively focus on the most relevant parts of the input sequence, improving performance and efficiency. Numerous studies have employed attention-based models for PV power forecasting, demonstrating their capability to capture intricate temporal patterns and dependencies within the data (Pan et al., 2019; Ju et al., 2020; Khan et al., 2023; Zhu et al., 2022; Kharlova et al., 2020; Aslam et al., 2021). However, despite their ability to weigh the importance of different time steps, many existing attention-based models for PV power forecasting do not explicitly incorporate the direction of changes in solar irradiance. They primarily focus on the magnitude of these changes, potentially neglecting valuable information embedded in the directionality of fluctuations, which can significantly influence future PV power generation.

2.2. Spatial–temporal data modeling and collaborative representation

Spatial–temporal data, characterized by its dependence on both location and time, presents unique modeling challenges due to the intricate interactions between these dimensions. Various techniques have been developed to address these challenges in energy forecasting applications. Traditional statistical models like Autoregressive Integrated Moving Average (ARIMA) by Das et al. (2018) and Exponential Smoothing (ES) by Mekhilef et al. (2011) effectively capture temporal trends but have limitations in accounting for spatial correlations. Vector Autoregressive Moving Average (VARMA) models by Hu et al. (2024) offer a way to incorporate spatial dependencies; however, their reliance on linear assumptions may hinder their ability to capture the complex nonlinear dynamics inherent in PV power generation. Machine learning algorithms, such as Artificial Neural Networks (ANNs) by Antonopoulos et al. (2019) and Support Vector Machines (SVMs) by Lin et al. (2020), provide more flexible approaches for modeling complex systems, capable of learning nonlinear relationships from data. However, these algorithms often require extensive parameter tuning and can lack interpretability, making it challenging to understand the factors driving their predictions. This lack of transparency can be a significant drawback in critical applications like PV power forecasting, where understanding the underlying drivers of predictions is essential for building trust and facilitating informed decision-making. Graph Neural Networks (GNNs) by Cao et al. (2020) offer a powerful tool for modeling spatial–temporal data by representing relationships between locations as a graph structure. This structure enables GNNs to capture both spatial and temporal dependencies (Huang et al., 2024), making them well-suited for tasks like PV power forecasting, where interactions between geographically distributed PV systems and time-varying weather patterns are crucial. Li et al. (2023) and Wang et al. (2024) have explored the use of GNNs for PV power forecasting, demonstrating their ability to capture complex spatial–temporal relationships. Wang et al. (2024) proposed a dynamic directed graph convolution network for ultra-short-term forecasting of distributed photovoltaic power, highlighting its effectiveness in enhancing network resilience and flexibility.

Collaborative representation techniques (Hu et al., 2024) play a vital role in analyzing data from multiple sources by leveraging shared information and patterns to enhance prediction accuracy and robustness. Collaborative representation techniques in PV power forecasting have been explored to enhance prediction accuracy by leveraging data from multiple PV farms. Guermoui et al. (2024) discussed the use of hybrid models, which have demonstrated superior performance compared to standalone models. These models combine data from various PV farms to improve the accuracy of power forecasting. Additionally, Lateko et al. (2022) highlighted ensemble learning methods, which involve combining multiple models to enhance the precision of PV power forecasting. Examples include multi-task learning (Shih et al., 2019), where shared features are learned across tasks, and ensemble methods (Khan et al., 2022), which combine predictions from multiple models. Federated learning allows training models on decentralized data without sharing private information (Zhou et al., 2022b). Collaborative representation techniques have also been explored in the context of PV power forecasting (Lateko et al., 2022), where the use of ensemble learning methods has been highlighted as a means to boost accuracy by combining multiple models. Venkatraman and Pitchaipillai (2024) developed a deep belief network-based auto-LSTM approach using wireless sensor networks for energy forecasting in solar power plants, showcasing the potential of collaborative representation for distributed energy systems. Moreover, Park et al. (2023) introduced the concept of single-site-based zero-shot PV power forecasting (SZF), which involves training a model on source data and making predictions on target data without fine-tuning. This approach can be beneficial in scenarios where fine-tuning data may be limited or unavailable. Furthermore, Zhu et al. (2024) emphasized the importance of data clustering, model enhancement, and uncertainty analysis in PV power forecasting.

Despite these advancements, a persistent need exists for PV power forecasting models that can explicitly capture the directional nature of solar irradiance fluctuations and provide transparent insights into the factors driving their predictions. Our proposed CoDR model addresses this gap. CoDR introduces a novel collaborative directional representation, encoding both the magnitude and direction of power variations to enhance forecasting accuracy, interpretability, and explainability.

3. Materials

This study utilizes a real-world dataset provided by Elia Group (EG), the Belgian transmission system operator. The dataset comprises 15-minute resolution PV power generation data, including day-ahead measurements and intra-day forecasts, covering 14 different regions in Belgium from January 1, 2022, to December 31, 2022 (Zsiborács et al., 2021). The PV power generation is measured in megawatts (MW). This dataset is particularly relevant due to the high penetration of PV power generation in the Belgian power system, which poses significant challenges to grid operation and management.

As PV power generation is directly influenced by sunshine duration and irradiance, the dataset exhibits clear seasonal patterns (Islam et al., 2018), as illustrated in Figs. 1 and 2. Fig. 1 shows the average daily PV energy production over time, highlighting the increasing trend from January to a peak in the summer months, followed by a gradual decrease towards December. Fig. 2 depicts the average monthly hourly PV production, revealing a consistent diurnal pattern with higher generation during midday hours and lower generation in the morning and evening. The variation in peak generation hours across different months reflects the influence of solar altitude angle and seasonal changes in solar irradiance. These figures demonstrate the inherent seasonality and daily periodicity present in PV power generation data, which pose challenges for accurate forecasting.

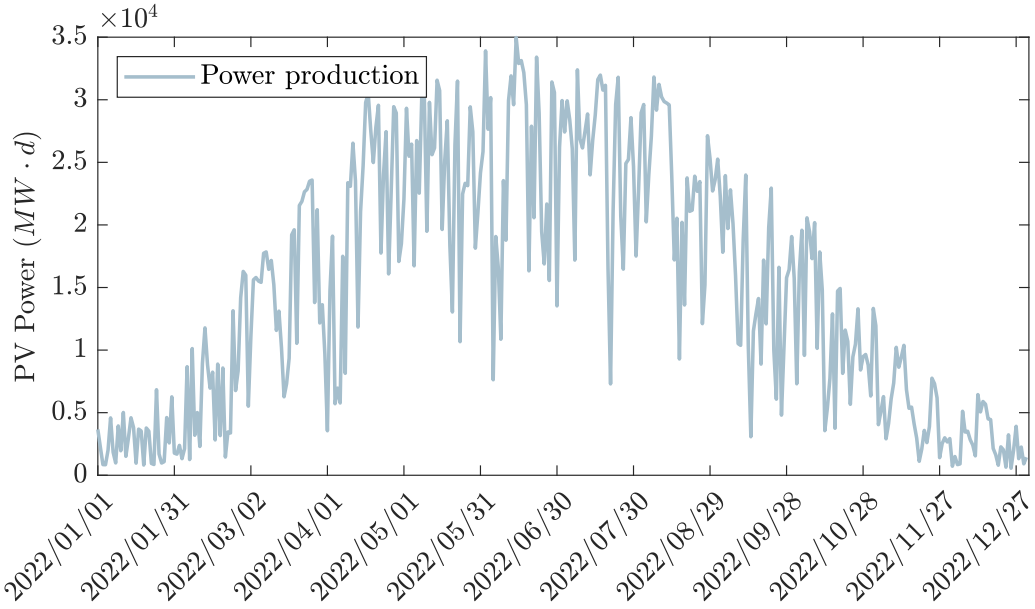


Fig. 1. Daily photovoltaic power production changes over time.

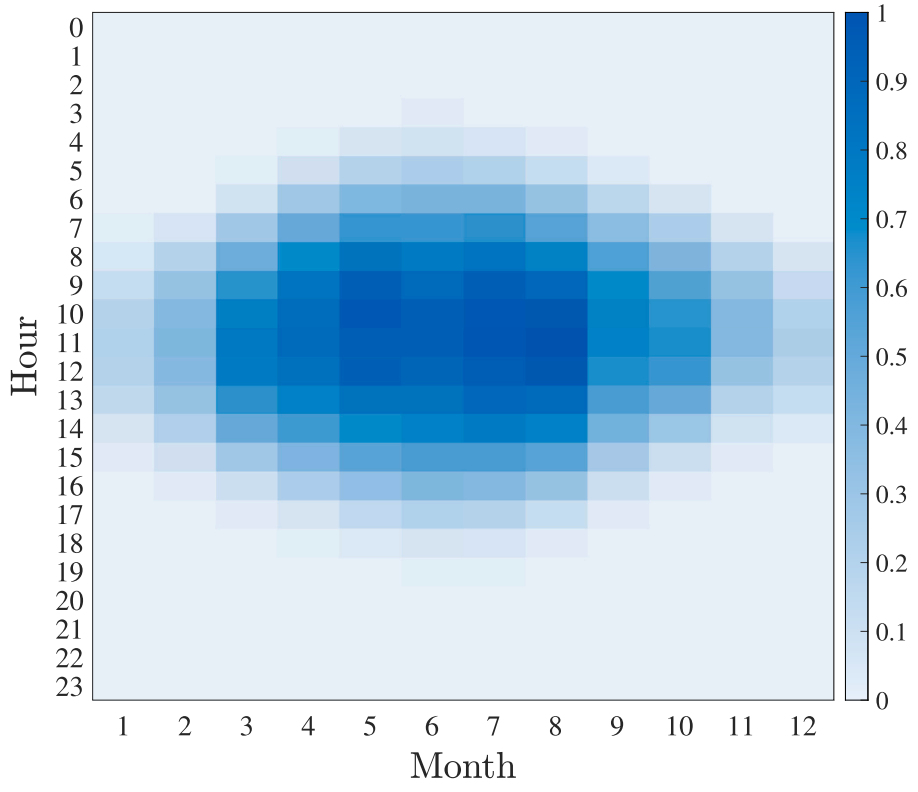


Fig. 2. Average monthly hourly scaled solar power heat map (2022).

4. Problem formulation

Solar irradiance is the amount of solar power received per unit area on a given surface. It is measured in watts per square meter (W/m^2), and it varies depending on the location, time, season, weather, and other factors. Forecasting the solar irradiance data is an important and challenging task for the solar power generation systems, as it can help to optimize the operation and maintenance of the PV panels, and to integrate the renewable energy sources into the power grid. We propose a novel model that can learn explainable time-varying directional representations for the solar irradiance data. Our model can capture both the temporal and spatial dependencies of the data, and reveal the causal mechanisms and directional effects of the input variables on the output variable. The research problem can be formulated as follows:

Let $\boldsymbol{Z} \in \mathbb{R}^{N \times D}$ be a multivariate time series data that represents the solar irradiance received on a given surface area in a given time interval, where N is the number of observations and D is the number of variables. The problem formulation of this paper is to learn a function $f : \mathbb{R}^{T \times D} \rightarrow \mathbb{R}^{O \times D}$ that can map a sequence of past observations $\boldsymbol{X}_t \in \mathbb{R}^{T \times D}$ to a sequence of future values $\boldsymbol{Y}_t \in \mathbb{R}^{O \times D}$, where T is the

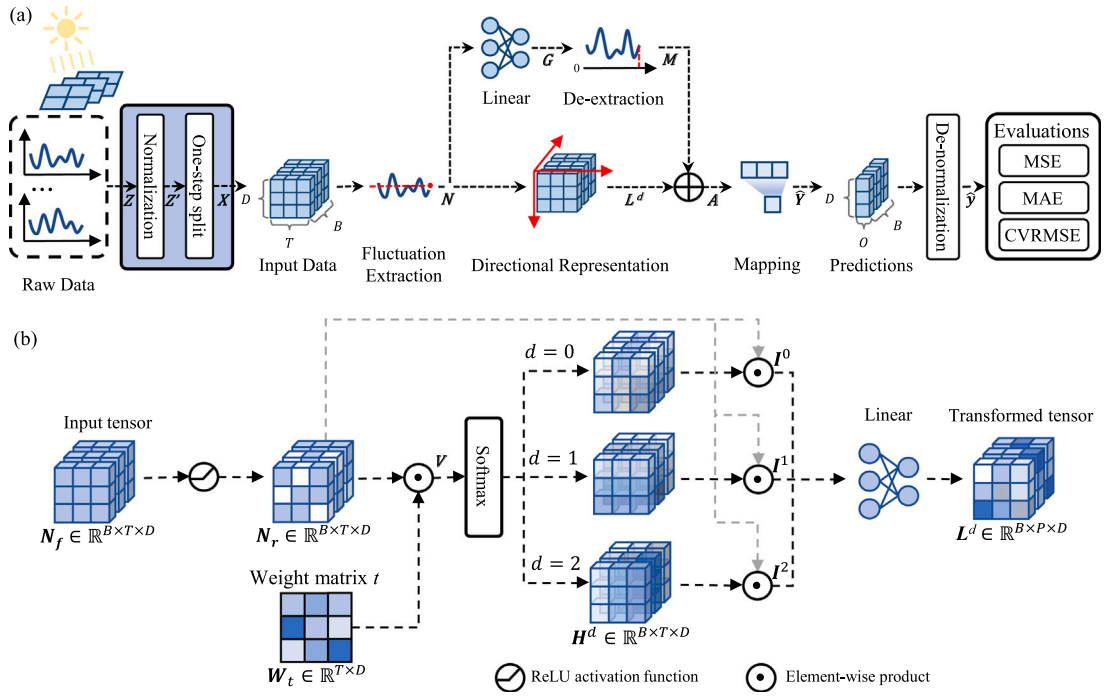


Fig. 3. The schematic illustration of the proposed CoDR model. (a) The workflow. (b) Directional Representation (DR).

Table 1
Symbols and meanings.

Symbol	Semantic
D	The number of variables
T	The window size
O	The output size
$\boldsymbol{Z}$	The raw solar irradiance data
$\boldsymbol{Z}'$	The normalized solar irradiance data
B	The batch size
$\boldsymbol{X}$	Batched input tensor, $\boldsymbol{X} \in \mathbb{R}^{B\times T\times D}$
$\boldsymbol{Y}$	Batched output tensor, $\boldsymbol{Y} \in \mathbb{R}^{B\times 1\times D}$
$\boldsymbol{N}_f$	The fluctuation component data, $\boldsymbol{N}_f \in \mathbb{R}^{B\times T\times D}$
$\boldsymbol{N}_r$	The rectified fluctuation data, $\boldsymbol{N}_r \in \mathbb{R}^{B\times T\times D}$
$\boldsymbol{V}$	The transformed fluctuation data, $\boldsymbol{V} \in \mathbb{R}^{B\times T\times D}$
$\boldsymbol{d}$	The dimension of directional representation
$\boldsymbol{H}^d$	The highlighted score tensor in dimension $\boldsymbol{d}$, $\boldsymbol{H}^d \in \mathbb{R}^{B\times T\times D}$
$\boldsymbol{I}^d$	The converted tensor, $\boldsymbol{I}^d \in \mathbb{R}^{B\times T\times D}$
$\boldsymbol{L}^d$	The directional representation tensor, $\boldsymbol{L}^d \in \mathbb{R}^{B\times P\times D}$
$\boldsymbol{G}$	The intermediate fluctuation data, $\boldsymbol{G} \in \mathbb{R}^{B\times P\times D}$
$\boldsymbol{M}$	The de-extracted data, $\boldsymbol{M} \in \mathbb{R}^{B\times P\times D}$
$\boldsymbol{A}$	The feature fusion data, $\boldsymbol{A} \in \mathbb{R}^{B\times P\times D}$
$\hat{\boldsymbol{Y}}$	The predicted output tensor, $\hat{\boldsymbol{Y}} \in \mathbb{R}^{B\times O\times D}$
$\hat{\boldsymbol{y}}$	The final predicted value, $\hat{\boldsymbol{y}} \in \mathbb{R}^{B\times O\times D}$

window size and O is the output size. The function f should capture the temporal patterns, dependencies, and correlations of the solar irradiance data, and use them to improve the accuracy and reliability of the forecasting task. The main notations used in this paper are listed in Table 1.

5. Methodology

In this section, we present the details of our proposed CoDR model for PV power generation forecasting. The CoDR model is a novel deep learning approach that leverages collaborative directional representation to capture both temporal and spatial dependencies in the data. We will describe the proposed model in detail, and also provide a summary of the CoDR model in algorithm form. Moreover, we will provide a theoretical analysis of the CoDR model, which explains how the directional representation can capture the causal relationships and directional influences among the input variables and the output variable. We will also derive the upper bound of the forecasting error and prove the convergence of the model.

5.1. CoDR model overview

The CoDR model is a novel deep learning approach for solar irradiance forecasting that leverages the directional fluctuations of the data. The CoDR model consists of seven main steps, as shown in Fig. 3(a).

The first step is data preprocessing, which normalizes and transforms the raw solar irradiance data into input–output pairs for a supervised learning problem. The second step is window fluctuation extraction, which extracts the fluctuation component from the normalized data by subtracting the moving average, which reflects the short-term variations of the data. The third step is directional representation, which transforms the fluctuation component into a directional representation, where each data point is assigned a direction (upward or downward) and a magnitude based on its relative position to its previous point. The fourth step is linearization, which linearizes the directional representation by applying a linear layer, which reduces the dimensionality and complexity of the data. The fifth step is window fluctuation de-extraction, which de-extracts the linearized data by adding the last element of the input data, which recovers the trend and seasonality components of the data. The sixth step is mapping, which maps the de-extracted data and the directional representation data into a single output value for each input sequence by using a linear layer, which produces the predicted values of the solar irradiance data. The seventh and final step is data postprocessing, which de-normalizes the predicted values to the original scale of the time series data, and calculates the error metrics to evaluate the performance of the model.

The CoDR model uses a simple and interpretable representation that can capture both the trend and the intensity of the solar irradiance variations, and improve the accuracy and reliability of the forecasting task. The CoDR model can also reveal the causal mechanisms and directional effects of the input variables on the output variable, and provide insights into the dynamics and patterns of the solar irradiance data.

5.2. Description of the CoDR model components

In this subsection, we will explain the seven main components of the CoDR Model, each of which performs a specific function and contributes to the overall performance and accuracy of the model. The components are: data preprocessing, window fluctuation extraction, directional representation, linearization, window fluctuation de-extraction, mapping, and data postprocessing. We will describe how they work and how they are connected to achieve the forecasting task.

5.2.1. Data preprocessing

Data preprocessing is the first component of the CoDR model, which prepares the raw solar irradiance data for the subsequent steps of fluctuation extraction, directional representation, and forecasting. Data preprocessing consists of two parts: data normalization and one-step forward split.

Data normalization scales the input data $\boldsymbol{Z}$ to the range [0, 1] using min–max normalization, which can speed up the model training process and reduce the impact of outliers. Min–max normalization individually normalizes each input sample to a scalar value by subtracting the minimum value and dividing by the range of values. This ensures that the input data has a common scale and does not dominate the model parameters or the loss function. The normalized data is denoted by $\boldsymbol{Z}'$, and the normalization formula is given by:

$$\boldsymbol{Z}' = \frac{\boldsymbol{Z} - \min(\boldsymbol{Z})}{\max(\boldsymbol{Z}) - \min(\boldsymbol{Z})}. \tag{1}$$

One-step forward split transforms the normalized time series data into a supervised learning problem, where each input sequence is paired with its next value as the output. This transformation method converts time series data into a dataset with input features and corresponding target variables. The input features are the past observations of the solar irradiance data, and the target variables are the future values of the solar irradiance data. The model learns to predict the target variables based on the input features, using the temporal patterns and dependencies of the data. The input and output data are denoted by $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively, and the transformation formula is given by:

$$\begin{bmatrix} \boldsymbol{Z}'_1 & \boldsymbol{Z}'_2 & \cdots & \boldsymbol{Z}'_T \\ \boldsymbol{Z}'_2 & \boldsymbol{Z}'_3 & \cdots & \boldsymbol{Z}'_{T+1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Z}'_{n-T} & \boldsymbol{Z}'_{n-T+1} & \cdots & \boldsymbol{Z}'_{n-1} \end{bmatrix} \rightarrow \begin{bmatrix} \boldsymbol{Z}'_{T+1} \\ \boldsymbol{Z}'_{T+2} \\ \vdots \\ \boldsymbol{Z}'_n \end{bmatrix}. \tag{2}$$

where T is the window size, which determines how many past observations are used as inputs for each prediction. The left part is the normalized inputs of the model, denoted by $\mathcal{X} \in \mathbb{R}^{(n-T)\times T\times D}$, and the right part is the normalized outputs of the model, denoted by $\mathcal{Y} \in \mathbb{R}^{(n-T)\times D}$. Several consecutive instances in $(\mathcal{X}, \mathcal{Y})$ are denoted by $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathbb{R}^{B\times T\times D} \times \mathbb{R}^{B\times 1\times D}$, where B is the batch size, T is the window size, and D is the number of variables.

5.2.2. Window Fluctuation Extraction (WFE)

Window fluctuation extraction is the second component of the CoDR model, which aims to isolate the short-term variations of the solar irradiance data from the long-term trend and seasonality components. The short-term variations, or the fluctuation component, capture the deviation of the data from its local mean within the window, which reflects the dynamic and nonlinear nature of the solar irradiance data. The fluctuation component is the main input for the directional representation step, which transforms the data into a binary sequence that indicates the direction changes of the data.

The window fluctuation extraction step is performed by subtracting the moving average of each input sequence from its original values, which reflects the deviation of the data from its local mean within the window. The moving average is calculated by averaging the values of each input sequence along the time dimension, which can smooth out the noise and capture the trend and seasonality components of the data. The subtraction operation then removes these components and retains only the fluctuation component, which captures the short-term variations of the data. The resulting tensor $\boldsymbol{N}_f$ represents the difference between each input sequence and its moving average, which reflects the fluctuation range of the data within the window. The fluctuation range is an important feature for the directional representation step, as it indicates the direction and magnitude of the data changes. The window fluctuation extraction step is performed as follows:

$$\boldsymbol{N}_f = \boldsymbol{X} - \frac{1}{T}\sum_{i=1}^{T} \boldsymbol{X}_{:,i,:}, \tag{3}$$

where $\boldsymbol{N}_f \in \mathbb{R}^{B\times T\times D}$ is the fluctuation component data; $\boldsymbol{X} \in \mathbb{R}^{B\times T\times D}$ is the input tensor; T is the window size; and B and d are the batch size and the number of variables, respectively. The $-$ operator first broadcasts the sequence length of $\frac{1}{T}\sum_{i=1}^{T} \boldsymbol{X}_{:,i,:}$ to T, and then performs element-wise subtraction between $\boldsymbol{X}$ and $\frac{1}{T}\sum_{i=1}^{T} \boldsymbol{X}_{:,i,:}$.

5.2.3. Directional Representation (DR)

Directional representation is a novel key component that we propose to capture the features of time series data in different directions. The directional representation unit, as shown in Fig. 3(b), transforms the fluctuation range data $\boldsymbol{N}$ into a low-dimensional and non-linear representation that can handle temporal patterns, dependencies, and correlations of PV power generation. The directional representation unit consists of four steps: (1) Rectification, (2) Transformation, (3) Highlighting, and (4) Linearization.

- **Rectification**: This step applies a rectified linear unit (ReLU) function on the data $\boldsymbol{N}_f$ to filter out the negative values and retain the positive values of the data. The rectified data is denoted by $\boldsymbol{N}_r$, and the rectification formula is given by:

$$\boldsymbol{N}_r = \mathrm{ReLU}(\boldsymbol{N}_f), \tag{4}$$

  The rectification step ensures the accuracy of long-term temporal series feature extraction by removing the downward fluctuations that may interfere with the prediction.
- **Transformation**: This step changes the rectified data $\boldsymbol{N}_r$ and discovers the temporal patterns by multiplying it by a learnable weight matrix $\boldsymbol{W}_t$. This step can be described as below formula:

$$\boldsymbol{V} = \boldsymbol{N}_r \cdot \boldsymbol{W}_t, \tag{5}$$

  where $\boldsymbol{V} \in \mathbb{R}^{B\times T\times D}$ is the result of transformation and $\boldsymbol{W}_t \in \mathbb{R}^{D\times T}$ is the learnable weight matrix. The transformation step learns the optimal weights for each time step and variate to capture the temporal dependencies and correlations of the data.
- **Highlighting**: This step feeds $\boldsymbol{V}$ into a softmax layer to enlarge the difference in several aspects. These aspects are batch dimension ($\boldsymbol{d}$=0), sequence dimension ($\boldsymbol{d}$=1), and variate dimension ($\boldsymbol{d}$=2). The softmax layer computes the attention score for each element in $\boldsymbol{V}$ along each dimension, indicating its importance or relevance for the prediction task. The highlighted score tensor is denoted by $\boldsymbol{H}^d$, and the highlighting formula is given by:

$$\boldsymbol{H}^d = \begin{cases} \dfrac{\exp(\boldsymbol{V}_{b,t,i})}{\sum_{b=1}^{B}\exp(\boldsymbol{V}_{b,t,i})}, & d = 0, \\ \dfrac{\exp(\boldsymbol{V}_{b,t,i})}{\sum_{t=1}^{T}\exp(\boldsymbol{V}_{b,t,i})}, & d = 1, \\ \dfrac{\exp(\boldsymbol{V}_{b,t,i})}{\sum_{i=1}^{D}\exp(\boldsymbol{V}_{b,t,i})}, & d = 2, \end{cases} \tag{6}$$

  where d is the observed direction. The highlighting step assigns higher scores to the more significant and influential elements in the data, and lower scores to the less relevant and noisy elements.

- **Linearization**: This step multiplies the rectified input tensor $\boldsymbol{N}_r$ by the highlighted score tensor $\boldsymbol{H}^d$ to create the dynamically generated converted tensor $\boldsymbol{I}^d$. This step can be described as follows:

$$\boldsymbol{I}^d = \boldsymbol{H}^d \cdot \boldsymbol{N}_r, \tag{7}$$

where $\boldsymbol{I}^d \in \mathbb{R}^{B\times T\times D}$ is the converted tensor. Then, a linear transform will be applied to the tensor $\boldsymbol{I}^d$ to reduce its dimensionality and obtain the final directional representation tensor $\boldsymbol{L}^d$. This step is as below:

$$\boldsymbol{L}^d = \sum_{i=t}^{T} \boldsymbol{W}_l \times \boldsymbol{I}^d_{:,i,:} + b_l, \quad (d = 0, 1, 2), \tag{8}$$

where $\boldsymbol{L}^d \in \mathbb{R}^{B\times P\times D}$ is the result of directional representation; $\boldsymbol{W}_l$ is weight matrix b_l is a bias, and $\boldsymbol{P}$ is the hidden size in the linear layer. The linearization step combines the rectified data and the highlighted scores to generate a low-dimensional and non-linear representation of the data that reflects the direction and intensity of the fluctuations.

5.2.4. Linearization

Linearization is the component for transforming the fluctuation range data $\boldsymbol{N}_f$ into intermediate fluctuation data $\boldsymbol{G}$, which can capture the features of the fluctuation data more effectively. Linearization is performed by applying a linear layer to the data $\boldsymbol{N}_f$, which consists of a learnable weight matrix $\boldsymbol{W}_g$ and a bias term b_g. The linear layer maps the data $\boldsymbol{N}_f$ from a high-dimensional space to a low-dimensional space, reducing the complexity and redundancy of the data. The linearization formula is given by:

$$\boldsymbol{G} = \sum_{i=t}^{T} \boldsymbol{W}_g \times \boldsymbol{N}_f + b_g, \tag{9}$$

where $\boldsymbol{G} \in \mathbb{R}^{B\times P\times D}$ is the intermediate fluctuation data; $\boldsymbol{W}_g \in \mathbb{R}^{P\times D}$ is the weight matrix of the linear layer and $b_g \in \mathbb{R}^P$ is the bias term of the linear layer. The linearization step learns the optimal weights and biases for each time step and variate to capture the features of the fluctuation data.

5.2.5. Window Fluctuation De-extraction (WFD)

De-extraction is the component for adding the last element of the input tensor $\boldsymbol{X}$, denoted by $\boldsymbol{X}_{:,-1,:}$, to the intermediate fluctuation data $\boldsymbol{G}$, which recovers the trend and seasonality components of the time series data. The de-extraction formula is given by:

$$\boldsymbol{M} = \boldsymbol{G} + \boldsymbol{X}_{:,-1,:}, \tag{10}$$

where $\boldsymbol{M} \in \mathbb{R}^{B\times P\times D}$ is the result of de-extraction. The de-extraction step restores the original scale and shape of the data by adding the last value of the input sequence, which represents the most recent observation of the solar irradiance data.

5.2.6. Mapping

Mapping is the component for producing the final output tensor $\boldsymbol{Y}$ that contains the predicted values of the solar irradiance data. The mapping step consists of two parts: feature fusion and linear projection.

- Feature fusion: This part combines the features of all the intermediate data, namely the de-extracted data $\boldsymbol{M}$ and the directional representation data $\boldsymbol{L}^0$, $\boldsymbol{L}^1$, and $\boldsymbol{L}^2$. The feature fusion process can be shown as below formula:

$$\boldsymbol{A} = \boldsymbol{M} + \boldsymbol{L}^0 + \boldsymbol{L}^1 + \boldsymbol{L}^2, \tag{11}$$

where $\boldsymbol{A} \in \mathbb{R}^{B\times P\times D}$ is the result of feature fusion. The feature fusion process aims to integrate the information from different sources and dimensions, and enhance the representation power of the data. The feature fusion process combines the trend and seasonality components from the de-extracted data, and the direction and intensity components from the directional representation data.

- Linear projection: This part uses a linear layer to map the feature fusion data $\boldsymbol{A}$ to the final output tensor $\hat{\boldsymbol{Y}}$. The linear projection formula is given by:

$$\hat{\boldsymbol{Y}} = \sum_{i=t}^{P} \boldsymbol{W}_a \times \boldsymbol{A}_{:,i,:} + b_a, \tag{12}$$

where $\hat{\boldsymbol{Y}} \in \mathbb{R}^{B\times O\times D}$ is the output tensor, $\boldsymbol{O}$ is the output window size; $\boldsymbol{W}_a$ and b_a are the weight matrix and bias vector of the linear layer, respectively. The linear projection process aims to reduce the dimensionality of the data and obtain a single output value for each input sequence. The linear projection process learns the optimal weights and biases for each variate and output size to generate the predicted values of the solar irradiance data.

5.2.7. Data postprocessing

This component is used to obtain the final prediction results of the CoDR model, and to evaluate its performance against the ground truth data. The data postprocessing step consists of two parts: de-normalization and error calculation.

- De-normalization: This part applies the inverse of the min–max normalization formula to the outputs of the model in the post-processing stage to recover the original scale of the data. The de-normalization formula is given by:

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{Y}} \cdot (\max(\boldsymbol{Z}) - \min(\boldsymbol{Z})) + \min(\boldsymbol{Z}), \tag{13}$$

where $\min(\boldsymbol{Z})$ and $\max(\boldsymbol{Z})$ are the minimum and maximum values of $\boldsymbol{Z}$, respectively. The de-normalization step restores the original units and ranges of the data, which are necessary for the evaluation and comparison of the model results.
- Error calculation: This part calculates the error metrics between the predicted values $\hat{\boldsymbol{y}}$ and the ground truth values $\boldsymbol{y}$, which are the actual observations of the solar irradiance data. The error metrics measure the accuracy and reliability of the model predictions, and indicate the performance and quality of the model. The error metrics that we use in this paper are the mean square error (MSE), the mean absolute error (MAE), and the coefficient of variation of root mean square error (CVRMSE). The error calculation formulas are given by:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{14}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{15}$$

$$\mathrm{CVRMSE} = \frac{\sqrt{\mathrm{MSE}}}{\bar{y}} \times 100\%, \tag{16}$$

where n is the number of observations, and $\bar{y}$ is the mean of the actual values. The lower the values of these metrics, the better the performance of the prediction model. However, these metrics also have some limitations. For example, MSE and MAE do not consider the temporal correlation or order of the time series data, which may affect the prediction accuracy. CVRMSE may not reflect the absolute error or deviation of the prediction model, which may affect the reliability of the model. Therefore, using multiple evaluation metrics can provide a more comprehensive and objective assessment of the prediction models.

5.3. Summary of the CoDR model

The CoDR model utilizes a series of steps to process solar irradiance data and generate accurate forecasts of photovoltaic power generation. The model leverages a novel approach called CoDR that captures both the temporal and spatial dependencies within the data.

The CoDR model begins by taking raw solar irradiance data as input. This data is first preprocessed by normalizing it to a common scale and splitting it into input–output pairs, preparing it for supervised

Table 2
Benchmark methods for PV power generation forecasting.

Method	Type	Components	Advantages	Disadvantages
GAR (Wu et al., 2018)	Parametric model	Global autoregression	Simple and interpretable	Cannot capture nonlinearities and dependencies in data
AR (Huang et al., 2023)	Parametric model	Autoregression	Simple and interpretable	Cannot capture dependencies among multiple time series, sensitive to outliers
VAR (Hu et al., 2024)	Parametric model	Vector autoregression	Can capture interdependencies among multiple time series	Cannot capture nonlinearities in data, high computational complexity
DLinear (Zeng et al., 2023)	Neural network	Linear layers and skip connections	Can learn linear and nonlinear components of data	May suffer from overfitting and gradient vanishing
NLinear (Zeng et al., 2023)	Neural network	Nonlinear activation functions and skip connections	Can learn linear and nonlinear components of data	May suffer from overfitting and gradient vanishing
FiLM (Zhou et al., 2022a)	Neural network technique	Feature-wise linear modulation of one layer by another layer	Can modulate the features of one layer based on another layer's output	May introduce additional complexity and parameters
LSTM (Hochreiter and Schmidhuber, 1997)	Neural network	LSTM cells to model long-term dependencies in sequential data	Can handle long-term dependencies and variable-length input and output sequences	May suffer from information loss and gradient explosion
GRU (Chung et al., 2014)	Neural network	GRU cells to model long-term dependencies in sequential data	Can handle long-term dependencies and variable-length input and output sequences, simpler than LSTM	May suffer from information loss and gradient explosion
ED (Cho et al., 2014)	Neural network architecture	Encoder–decoder with fixed-length vector representation of input sequence	Can process variable-length input and output sequences without recurrent neural networks	May suffer from information loss due to fixed-length vector representation
CNN1D (Malek et al., 2018)	Neural network	Convolutional layers to extract local features from sequential data	Can capture local dependencies and patterns in data	Cannot capture global dependencies and long-term patterns in data
CNNRNN (Shi et al., 2015)	Neural network	Convolutional layers and recurrent layers to capture both local and global dependencies in sequential data	Can capture both local and global dependencies and patterns in data	May suffer from information loss and gradient explosion

learning. Next, the model extracts the short-term variations in the data by subtracting the moving average, isolating the fluctuation component. This fluctuation component is then transformed into a directional representation, effectively capturing the trend and intensity of power variation. To reduce dimensionality and complexity, the directional representation is linearized using a linear layer.

The trend and seasonality components are then recovered by adding the last element of the input data to the linearized data. Finally, the de-extracted data and directional representations are combined to produce a single output value for each input sequence, representing the predicted solar irradiance. The predicted values are then de-normalized back to the original scale, and error metrics are calculated to evaluate the model's performance.

The CoDR model's ability to capture both temporal and spatial dependencies through its unique directional representation allows it to achieve superior accuracy and generalization compared to other existing methods. Furthermore, its explainability, which reveals the contribution of each direction to the forecasting result, makes it a valuable tool for understanding model behavior and identifying potential biases.

6. Experiments

6.1. Implementation and experimental settings

We implemented the CoDR model using PyTorch v2.0.0, a popular deep learning framework (Kisvari et al., 2021). We performed all experiments on a server with an Intel(R) Xeon(R) Gold 5218R CPU (2.10 GHz) and 256G memory, and we used four Tesla V100-PCIE-16 GB GPUs to accelerate the training and testing processes.

6.2. State-of-the-art models

To benchmark the effectiveness of our proposed CoDR model, we conduct a comprehensive comparison with 22 state-of-the-art models for PV power generation forecasting. These models cover a wide range of methods, including parametric models, neural network models, neural network techniques, neural network architectures, graph neural network models, and shapelet learning methods. We describe the main components, advantages, and disadvantages of each model in Tables 2–4.

6.3. Model configurations

We conducted five repeated experiments on the PV power time series data to evaluate the performance of each method. We used this approach instead of cross-validation to preserve the temporal order of the data, which is essential for forecasting tasks. We trained the models using the Adam optimizer (Kingma and Ba, 2015) with the MSE as the loss function, following previous studies that showed their effectiveness for PV power forecasting (Chen et al., 2019). We applied the grid search method to optimize the hyper-parameters for each method over a predefined range of values. The optimal hyper-parameters for each baseline method obtained by the grid search method are shown in Table 8.

6.4. Sensitivity analyses

We first conduct sensitivity analysis to assess the impact of different hyperparameters on the performance of the proposed CoDR model. We

Table 3
Benchmark methods for PV power generation forecasting (continued).

Method	Type	Components	Advantages	Disadvantages
CNNRNNRes (He et al., 2016)	Neural network	Convolutional layers, recurrent layers, and residual connections to enhance the information flow and gradient propagation in the network	Can capture both local and global dependencies and patterns in data, and improve the network performance and stability	May introduce additional complexity and parameters
LSTNet (Lai et al., 2018)	Neural network	Convolutional layers, recurrent layers, and attention mechanisms to capture both short-term and long-term patterns in multivariate time series	Can handle multivariate time series with complex temporal patterns and dependencies, and improve the attention performance with skip connections	May suffer from information loss and gradient explosion
Transformer (Vaswani et al., 2017)	Neural network architecture	Self-attention mechanisms to encode and decode sequential data without using recurrence or convolution	Can process variable-length input and output sequences without recurrent neural networks or convolutional layers, and capture long-term dependencies with self-attention mechanisms	May suffer from information loss due to positional encoding and fixed-length hidden state
Informer (Zhou et al., 2021)	Neural network model	Self-attention mechanisms with probabilistic embedding and distilling operations to improve the performance and efficiency of long sequence forecasting	Can handle long sequence forecasting with high accuracy and efficiency, and reduce the computation cost and memory usage with probabilistic embedding and distilling operations	May introduce additional complexity and parameters
Autoformer (Wu et al., 2021)	Neural network model	An auto-regressive structure and an adaptive attention span to capture long-term dependencies in time series	Can handle long-term dependencies in time series with an auto-regressive structure that learns the optimal attention span for each input position, and improve the model performance with a novel initialization scheme	May introduce additional complexity and parameters

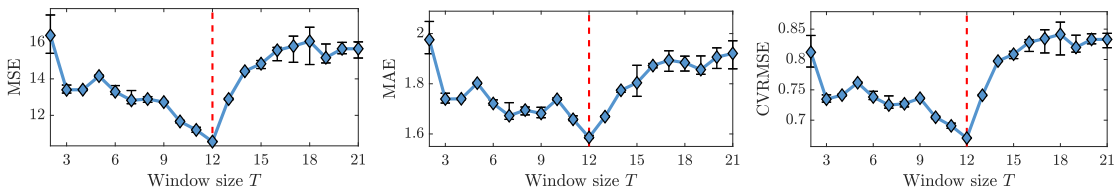
Table 4
Benchmark methods for PV power generation forecasting (continued).

Method	Type	Components	Advantages	Disadvantages
FEDformer (Zhou et al., 2022b)	Neural Network	Transformer model with frequency enhanced decomposition	Can capture long-term dependencies	May introduce additional complexity and communication overhead
DSANet (Huang et al., 2019)	Neural network model	Two parallel self-attention branches to capture both global and local dependencies in multivariate time series	Can handle multivariate time series with complex temporal patterns and dependencies, and improve the model performance with dual self-attention mechanisms	May introduce additional complexity and parameters
TPA-LSTM (Shih et al., 2019)	Neural network technique	Attention mechanisms to learn temporal patterns from historical data and apply them to future forecasting	Can improve the forecasting accuracy and interpretability by learning and applying temporal patterns from historical data	May suffer from information loss and gradient explosion
StemGNN (Cao et al., 2020)	Graph neural network	Graph convolutional layers and graph recurrent layers to handle spatio-temporal data with complex structures and dynamics	Can handle spatio-temporal data with complex structures and dynamics, and capture the spatial and temporal dependencies with graph neural networks	May introduce additional complexity and parameters
GAIN (Wang et al., 2023)	Graph neural network	Graph attention mechanisms to capture the interactions among multiple time series on a graph structure	Can handle multiple time series with complex interactions on a graph structure, and improve the attention performance with graph attention mechanisms	May introduce additional complexity and parameters
MSL (Wang and Cai, 2022)	Shapelet learning method	Shapelets learned from past observations to improve the accuracy and robustness of time series forecasting	Can handle time series with uncertainty, and improve the forecasting accuracy and robustness	May hard to learn exogenous factors

vary the window size, the training batch size, and the hidden size of the model, and measure how they affect the accuracy and reliability of the PV power generation forecasting.

6.4.1. Impact of window size

The sliding window size T reflects the temporal dependency between the past T days and the future $T+h$ days of energy consumption.



(a) MSE (b) MAE (c) CVRMSE

Fig. 4. The CoDR performance by varying the window size T in terms of three metrics. For each metric, the optimal value is found at the red dash line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Choosing an appropriate T can enable the prediction model to capture the underlying patterns of energy consumption more effectively. However, T should not be too small or too large, as both cases may degrade the prediction performance. To illustrate this point, we fix the batch size to 32 and the other parameters, and vary T from 2 to 21, while setting the prediction horizon h to 1. We repeat each experiment five times to reduce the randomness and report the average results in Fig. 4. Fig. 4 shows that the optimal window size for our model is 12, which achieves the lowest prediction error among all the values of T. When T is smaller than 12, the prediction error increases as T decreases. This is because a small window size cannot capture enough temporal information and is more susceptible to noise and outliers in the data. On the other hand, when T is larger than 12, the prediction error also increases as T increases. This is because a large window size introduces more irrelevant information and temporal variations, which may confuse the model and reduce its generalization ability. Therefore, we choose $T = 12$ as the optimal window size for our model in the subsequent experiments, as it balances the trade-off between information richness and complexity.

6.4.2. Impact of training batch size B

The batch size B indicates the number of samples that are processed in each iteration of the training process. Choosing an appropriate B can affect the convergence speed and accuracy of the prediction model. However, B should not be too small or too large, as both cases may have drawbacks. To illustrate this point, we fix the window size to $T = 12$ and the other parameters, and vary B from 2^0 to 2^7, while setting the prediction horizon h to 1. We repeat each experiment five times to reduce the randomness and report the average results in Fig. 5.

Fig. 5 shows that the optimal batch size for our model is 2^5, which achieves the lowest prediction error for MAE among all the values of B. When B is smaller than 2^5, the prediction error for MAE increases as B decreases. This is because a small batch size may cause instability and oscillation in the training process, leading to poor generalization performance. When B is larger than 2^5, the prediction error for MAE also increases as B increases. This is because a large batch size may reduce the diversity and randomness of the training samples, leading to overfitting and local optima.

For MSE and CVRMSE, the optimal batch size for our model is 2^0, which achieves the lowest prediction error among all the values of B. However, the difference between 2^0 and 2^5 is not significant in terms of accuracy, but it is significant in terms of efficiency. The running time of 2^0 is much longer than that of 2^5, as shown in Fig. 5. Therefore, we choose $B = 2^5$ as the optimal batch size for our model in the subsequent experiments, as it balances the trade-off between accuracy and efficiency.

6.4.3. Impact of hidden size P

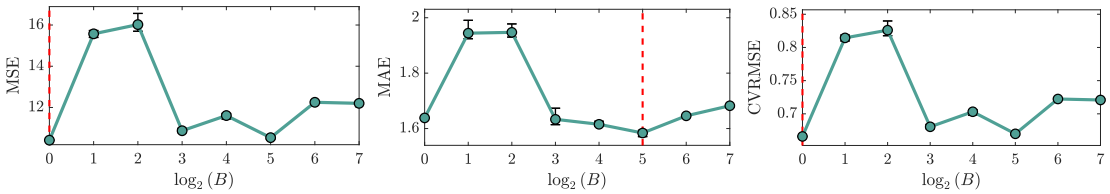
The hidden size P is a crucial hyperparameter that influences the performance of CoDR. The hidden size determines the dimensionality of the hidden state vector in the input tensor variants, which reflects the amount of information that can be stored and processed by the linear regression model. Choosing an appropriate P can affect the representation ability and learning capacity of CoDR. However, P should not be too small or too large, as both cases may have drawbacks. In this subsection, we investigate how different values of P affect the accuracy and efficiency of CoDR. To do this, we fix the window size to $T = 12$ and the batch size to $B = 2^5$, and vary P from 1 to 20, while setting the prediction horizon h to 1. We repeat each experiment five times to reduce the randomness and report the average results in Fig. 6.

Fig. 6 shows that the optimal hidden size for CoDR is $P = 18$, which achieves the lowest prediction error for MAE, MSE, and CVRMSE among all the values of P. When P is smaller than 18, the prediction error for all metrics increases as P decreases. This indicates that a small hidden size may limit the representation ability and learning capacity of CoDR, resulting in underfitting and high bias. When P is larger than 18, the prediction error for MSE and CVRMSE also increases as P increases, while the prediction error for MAE slightly decreases at first and then increases. This suggests that a large hidden size may increase the complexity and redundancy of CoDR, resulting in overfitting and high variance.

6.5. Comparison with state-of-the-art models

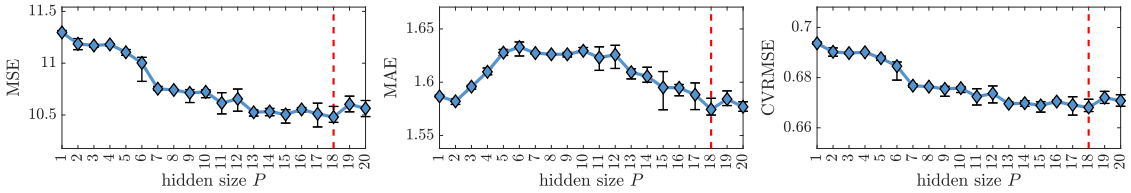
In this subsection, we compare the performance of CoDR with other state-of-the-art methods for photovoltaic power forecasting. We use three metrics: MSE, MAE, and CVRMSE. We consider 1 day ahead forecasting. The results are shown in Table 5. We can observe that:

- Linear models (GAR, AR, and VAR) have similar performance for h=1 forecasting. VAR performs slightly better than GAR and AR, indicating that VAR can capture cross-series dependencies in photovoltaic power generation data.
- Linear model variations (DLinear, NLinear, FiLM) have unsatisfactory performance, except for NLinear. NLinear reduces the scale of input data and obtains good results, especially in MAE. It ranks second among all models in MAE, which illustrates the importance of scaling the input data.
- Recurrent neural networks (RNNs), such as LSTM, GRU, and ED, also have poor performance. LSTM and ED do not capture effective information from historical data. GRU performs better than LSTM and ED, surpassing linear models, but not as good as NLinear. However, GRU has limited reasoning capabilities and there is still room for improvement in prediction accuracy.
- CNN- and RNN-based models (CNN1D, CNNRNN, CNNRNNRes, and LSTNet) perform better than linear models and RNNs. The combination of CNN and RNN is more effective than the pure CNN model CNN1D. Among them, CNNRNN performs better than CNNRNNRes, probably because the residual window data interferes with the prediction. LSTNet adds a skip window to CNNRNNRes, which splits the input sequence into small segments and uses GRU to model them. This slightly improves the performance but does not surpass CNNRNN.
- Self-attention-based models (Transformer, Informer, Autoformer, and FEDformer) perform worse than CNN-RNN models and NLinear. Compared with Transformer, Informer, and FEDformer improve their performance by using probsparse attention, distilling, generative decoder, frequency-based low-rank attention, and the mixture of experts decomposition.



(a) MSE (b) MAE (c) CVRMSE

Fig. 5. The CoDR performance by varying the batch size B in terms of three metrics. For each metric, the optimal value is found at red dash line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) MSE (b) MAE (c) CVRMSE

Fig. 6. The CoDR performance by varying the hidden size P in terms of three metrics. For each metric, the optimal value is found at red dash line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Performance and Complexity Comparison for Photovoltaic Power Prediction. The best results are shown in bold, the second-best are underlined, and the worst are in wavy lines.

Model	Complexity	MSE	MAE	CVRMSE
GAR	Linear	16.980532	2.145289	0.850436
AR	Linear	17.199577	2.142844	0.855903
VAR	Linear	14.995484	1.919366	0.799183
DLinear	Linear	18.734257	2.274024	0.893273
NLinear	Linear	12.913356	1.615689	0.741627
FiLM	Linear	18.083445	2.155598	0.877620
LSTM	Quadratic	18.338812	2.237135	0.883795
GRU	Quadratic	14.263420	1.852983	0.779431
ED	Quadratic	18.530966	2.209480	0.888413
CNN1D	Quadratic	18.426003	2.276761	0.885893
CNNRNN	Quadratic	13.674936	1.806042	0.763183
CNNRNNRes	Quadratic	15.545842	1.905770	0.813716
LSTNet	Quadratic	14.020832	1.813954	0.772774
Transformer	Quadratic	17.969414	2.302274	0.874848
Informer	Quadratic	14.342885	1.868295	0.781599
Autoformer	Quadratic	17.367300	2.051121	0.860066
FEDformer	Quadratic	14.373937	1.750293	0.782445
DSANet	Linear with Attention	13.669515	1.846164	0.763031
TPA-LSTM	Linear with Attention	13.921175	1.858356	0.770023
StemGNN	Not Directly Comparable[a]	17.357677	2.132312	0.859828
GAIN	Linear with Attention	12.696167	1.787765	0.735364
MSL	Linear	17.084484	1.971455	0.853035
CoDR	Linear with Attention	**10.480311**	**1.562783**	**0.668115**

[a] The complexity of StemGNN is heavily dependent on the structure of the graph, making it difficult to compare directly with other models.

- CNN-RNN and attention models as the two core components of the hybrid attention model (DSANet and TPA-LSTM) perform better than CNNRNN models and self-attention-based models. TPA-LSTM demonstrates that its temporal pattern attention can capture long-term dependencies. DSANet is slightly better than TPA-LSTM, indicating that its convolutional layers and self-attention module can capture global and local temporal patterns and dependencies in multivariate time series.
- Graph attention-based models (StemGNN and GAIN) use a graph attention mechanism to model the spatial correlation of photovoltaic power data. StemGNN has average performance, indicating that its graph attention mechanism cannot learn meaningful representations. GAIN improves over StemGNN by using a collaborative attention mechanism, which can enhance the spatial–temporal features. GAIN achieves the second-best results among all models, indicating that the collaborative attention mechanism is beneficial for feature learning and extraction.
- MSL learns shapelets from historical data to represent trending patterns. Its performance is close to AR, indicating that shapelets are not effective features for photovoltaic power forecasting.

In summary, CoDR consistently outperforms all other methods across all three metrics, demonstrating its superior ability to capture complex temporal dependencies and generate accurate photovoltaic power generation forecasts. As shown in Table 5, CoDR achieves the lowest MSE, MAE, and CVRMSE, with a maximum reduction of 17.45%, 2.55%, and 9.14% respectively, compared to the second-best

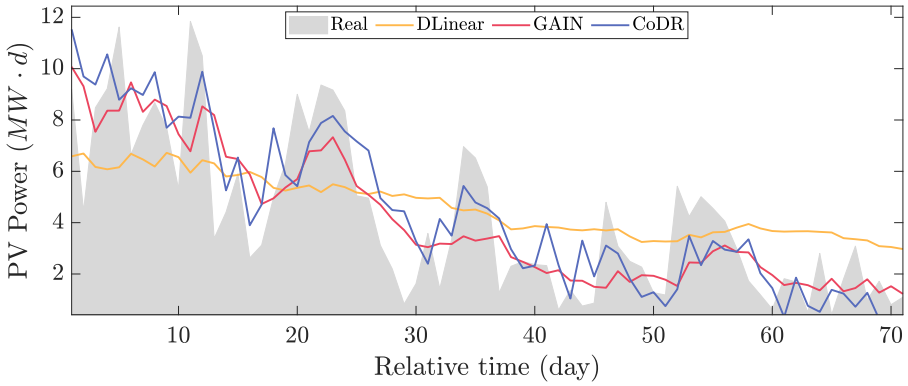


Fig. 7. The visualized comparisons on the real values with the other three methods.

method, GAIN. This significant performance advantage comes at the cost of increased computational complexity compared to simpler linear models. However, CoDR's complexity is comparable to that of other state-of-the-art deep learning models for time series analysis, such as GAIN, which leverages a similarly complex attention mechanism. The substantial performance gains offered by CoDR, especially the significant reduction in MSE and CVRMSE, justify this complexity increase for applications where accurate photovoltaic power forecasting is critical. A more detailed analysis of CoDR's computational complexity can be found in Appendix A.3.

To further illustrate the forecasting behavior of CoDR and the benchmark models, we compare their actual and predicted values visually in Fig. 7. While CoDR generally performs well in tracking the overall trends and variations in photovoltaic power generation, a time lag is evident in its predictions, as well as in the predictions of GAIN and DLinear. This lag suggests that all three models have some difficulty in accurately predicting the precise timing of rapid fluctuations in photovoltaic power output. This is not unexpected, given the complex and dynamic nature of solar irradiance patterns, which are influenced by numerous factors such as cloud cover, atmospheric conditions, and the sun's position. Nevertheless, CoDR demonstrates better performance than the other two benchmarks in capturing some of the peak-trough trends. GAIN, while capturing the general time-dependent relationship, appears to be less sensitive to these rapid peaks and troughs. DLinear exhibits the poorest performance, likely due to its inability to effectively handle the nonlinearity and seasonality inherent in the photovoltaic power generation data. CoDR's relative success in capturing more of the peak-trough behavior can be attributed to its novel components. Specifically, its scaling of the input data framework helps capture short-term representations, while its collaborative directional attention mechanism focuses on global and relevant features, enabling CoDR to generate predictions that are more aligned with the actual photovoltaic power variations, despite the inherent time lag in the forecasting process.

Fig. 8 displays the normalized results from actual and predicted values by different models. The Pearson correlation coefficient (PCC) between the actual value and the predicted value of each model is also annotated in the figure. The PCC measures how well the model can capture the trend and pattern of the data, and it is an important indicator of the prediction model's performance. We can observe that most of the data points are below the diagonal line, which means that the prediction models tend to underestimate the actual values when they reach the peak. This could be due to two reasons: (1) The photovoltaic power data are unstable during high fluctuation periods, which makes it difficult for the prediction models to find stable patterns. (2) The prediction models have limited ability to capture the sudden changes in the data, which leads to a delayed reaction in forecasting the peak values. Among all the models, CoDR has the highest PCC (0.814), indicating that it can better capture the trend and pattern of the data than other models. CoDR also has fewer data points below the diagonal line, suggesting that it can more accurately forecast the peak values than other models. This demonstrates that CoDR can effectively handle the instability and nonlinearity of the photovoltaic power data by using its novel components.

In this subsection, we have compared CoDR with other state-of-the-art methods for photovoltaic power forecasting, and showed that CoDR achieves superior performance in terms of accuracy, reliability, and correlation. We have also illustrated how CoDR can better track and predict the photovoltaic power variations than other methods visually. In the next subsection, we will conduct an ablation study to assess the effectiveness of the proposed model design.

6.6. Model ablation study

To evaluate the contribution of each component of our proposed CoDR model, we conduct an ablation study by removing one component at a time and comparing the performance with the full model. The components considered for this analysis include window fluctuation extraction (WFE), window fluctuation de-extraction (WFD), and directional representation (DR) in multi-direction fusion. Table 6 shows the results of the ablation analysis in terms of MSE, MAE, and CVRMSE metrics for one horizon (1 day). The results show that removing any component leads to a decrease in accuracy and an increase in error metrics for all horizons. This indicates that each component is essential for the effectiveness of our model and that they work well together.

The most significant drop in performance occurs when we remove both WFE and WFD components. This suggests that these components are crucial for capturing the fluctuation features of the solar irradiance data, which are important for forecasting PV power generation. By removing these components, we lose the ability to extract and represent the short-term variations and trends of the data, which leads to poor predictions. The second most significant drop in performance occurs when we remove all three DR components. This implies that these components are important for capturing the directional features of the solar irradiance data, which are relevant for forecasting PV power generation. By removing these components, we lose the ability to observe and model the data from different directions, such as batch, sequence, and variate dimensions, which leads to inaccurate predictions. The third most significant drop in performance occurs when we remove only one DR component. This indicates that each DR component has a different impact on the performance of our model, depending on the direction it observes. Among them, DR(0) has the least impact, as it observes the data from the batch dimension, which is less informative than the other dimensions. DR(1) and DR(2) have more impact, as they observe the data from the sequence and variate dimensions, which are more informative and relevant for forecasting.

In summary, this subsection has presented an ablation study to assess the contribution of each component of our proposed CoDR model.

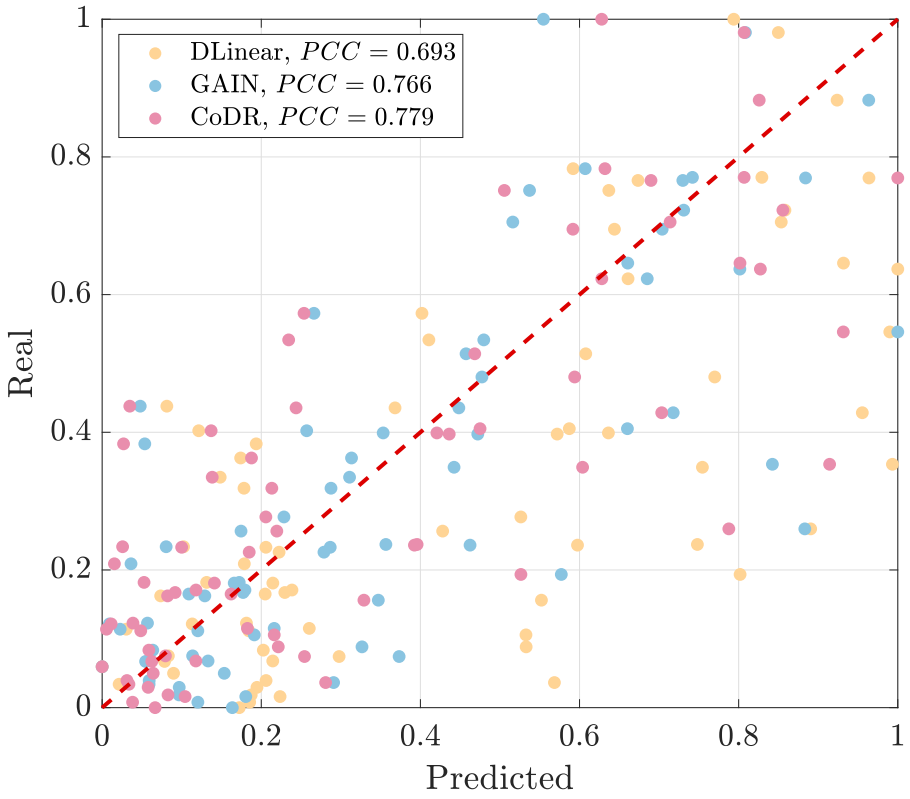


Fig. 8. The correlation visualizations of CoDR predictions and three other benchmark predictions: day.

Table 6
Model ablation study. The best results are shown in bold, the second-best results are underlined, and the worst results are in wavy lines.

Model	MSE	MAE	CVRMSE
ALL	**10.480311**	**1.562783**	**0.668115**
w/o WFE+WFD	18.222259	2.089341	0.880974
w/o DR(0)	10.625047	1.581923	0.672714
w/o DR(1)	12.090807	1.651888	0.717618
w/o DR(2)	12.192431	1.654124	0.720620
w/o DR(0,1,2)	15.091056	1.948130	0.801723

We have shown that each component is essential for achieving high accuracy and low error metrics in PV power generation forecasting. We have also explained how each component affects the performance of our model by capturing different features and relationships of the solar irradiance data. Moreover, we have demonstrated the effectiveness of our proposed method by comparing it with the ablated versions, and showing that it has the best performance in all metrics when all components are included. This indicates that our proposed method can leverage the advantages of each component and achieve a synergistic effect in forecasting PV power generation.

6.7. Explainablilty

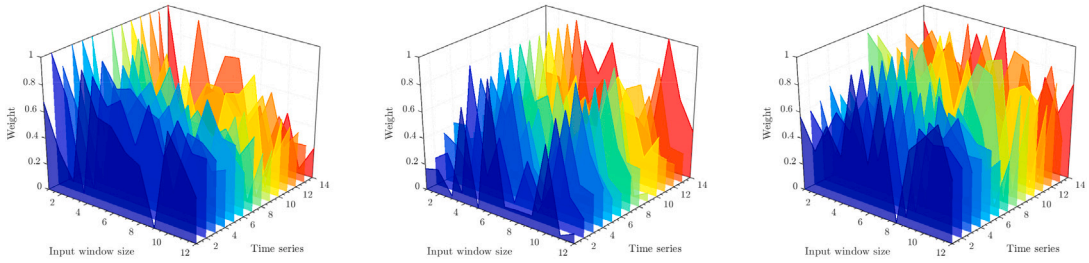
One of the advantages of CoDR is that it can provide intuitive and interpretable explanations for its predictions by showing how much each direction contributes to the forecasting result. Fig. 9 shows an example of how CoDR learns different directional weights $\boldsymbol{W}_t$ for different input window sizes and time series. The figure consists of three 3D graphs, each representing the directional weight in CoDR for one of the three directions: 0 (north-south), 1 (east–west), and 2 (up-down). The graphs are plotted on a 3D grid with the x-axis representing the input window size, the y-axis representing the time series, and the z-axis representing the weight. The graphs are color-coded, with blue representing lower weights and red representing higher weights.

From Fig. 9, we can observe that CoDR can learn different directional weights for different input window sizes and time series, which reflects the varying importance of different directions for forecasting photovoltaic power generation. For example, we can see that CoDR assigns higher weights to larger input window sizes than smaller input window sizes, which indicates that larger input window sizes can capture more information about the temporal patterns and trends of solar irradiance and photovoltaic power generation. We can also see that CoDR assigns higher weights to some directions than others, which indicates that some directions are more relevant to the solar irradiance and photovoltaic power generation than others. For instance, we can see that CoDR assigns higher weights to direction 2 (up-down) than direction 0 (north-south) or direction 1 (east–west), which implies that direction 2 (up-down) captures more information about the sun elevation angle, which is a key factor affecting solar irradiance and photovoltaic power generation. On the other hand, we can see that CoDR assigns lower weights to direction 0 (north-south) than direction 1 (east–west) or direction 2 (up-down), which implies that direction 0 (north-south) captures less information or more noise about solar irradiance and photovoltaic power generation, especially in some regions where north-south orientation has little impact on solar irradiance and photovoltaic power generation.

By showing how much each direction contributes to the forecasting result, CoDR can provide intuitive and interpretable explanations for its predictions. This can help users to understand how CoDR works and why it makes certain predictions. This can also help users identify potential errors or biases in CoDR's predictions and improve its performance accordingly.

7. Conclusions and future work

This paper introduced CoDR, a novel deep learning model for photovoltaic (PV) power generation forecasting. CoDR leverages a unique collaborative directional representation of solar irradiance data to capture the complex temporal dependencies inherent in these time series.



(a) Visualized weights of direction 0 (b) Visualized weights of direction 1 (c) Visualized weights of direction 2

Fig. 9. The directional weights in the trained CoDR.

This new representation method enables CoDR to achieve superior forecasting accuracy compared to existing state-of-the-art models. We conducted extensive experiments on a real-world dataset from Elia Group, the Belgian transmission system operator, comparing CoDR with 22 benchmark models. Our results demonstrate that CoDR consistently outperforms all other methods in terms of MSE, MAE, and CVRMSE, achieving reductions of up to 17.45%, 2.55%, and 9.14% respectively, compared to the second-best performing model. To validate the contribution of each component in CoDR, we performed an ablation study, which confirmed that all components are essential for achieving the model's high accuracy. Furthermore, we provided a theoretical analysis of CoDR, proving its convergence and stability. Importantly, the interpretable nature of CoDR allows for the visualization of the directional influences on the forecasting results, offering valuable insights into the key factors driving photovoltaic power generation. These insights can aid in understanding model behavior and potentially improving the forecasting process.

While our results demonstrate the effectiveness of CoDR for PV power forecasting, it is important to acknowledge some limitations. First, CoDR currently relies primarily on solar irradiance data. Incorporating additional features, such as temperature, cloud cover, and other meteorological variables, could potentially further improve its accuracy. Second, our evaluation focuses on one-day-ahead forecasting. Exploring CoDR's performance for shorter-term (e.g., hourly) or longer-term forecasting horizons would provide a more comprehensive assessment of its capabilities. Finally, our study uses data from a specific geographical region (Belgium). Evaluating CoDR's generalizability to different climates and locations is essential to establish its broader applicability.

As for future work, we plan to investigate several promising directions. First, we aim to extend the CoDR approach to other renewable energy sources, such as wind and hydropower, exploring how the collaborative directional representation can be adapted to capture the unique characteristics of these energy sources. Second, we plan to incorporate additional external factors, such as geographical information and social events, into our model to investigate their impact on photovoltaic power generation forecasting. Finally, we intend to explore the use of more advanced attention mechanisms, including transformer-based models and graph neural networks, to further enhance the performance and capabilities of CoDR.

CRediT authorship contribution statement

Zhijin Wang: Writing – review & editing, Writing – original draft, Software, Project administration, Methodology, Funding acquisition, Conceptualization. **Hanjing Liu:** Writing – review & editing, Writing – original draft, Software, Methodology, Data curation. **Senzhen Wu:** Writing – review & editing, Writing – original draft, Software, Methodology, Data curation. **Niansheng Liu:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization. **Xiufeng Liu:** Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Conceptualization. **Yue Hu:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Data curation. **Yonggang Fu:** Writing – review & editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This research was supported in part by the Natural Science Foundation of China (no. 62006096), Natural Science Foundation of Fujian Province (CN) (nos. 2020J05146, 2021J01857, 2022J01820, and 2022J01335), Education Department of Fujian Province (CN) (nos. JAT210224 and JAT231193), and the RE-INTEGRATE project (no. 101118217) funded by the European Union Horizon 2020 research and innovation programme.

Appendix. Supplementary material

The meanings of abbreviations are listed in Table 7.

A.1. Hyper-parameter setting

The parameters setting of the proposed method and benchmarks are listed in Table 8.

A.2. Theoretical analysis of the CoDR model

In this subsection, we will provide the theoretical analysis of the proposed CoDR model, and show its convergence, stability, and optimality properties. We first introduce some notation and terminology that will be used throughout this subsection.

Let $\boldsymbol{Z} \in \mathbb{R}^{N\times D}$ be a multivariate time series data that represents the solar irradiance received on a given surface area in a given time interval, where N is the number of observations and D is the number of variables. Let $\boldsymbol{X} \in \mathbb{R}^{B\times T\times D}$ be a tensor that contains B input sequences of length T extracted from $\boldsymbol{Z}$ using one-step forward split. Let $\boldsymbol{Y} \in \mathbb{R}^{B\times 1\times D}$ be a tensor that contains B output values corresponding to each input sequence in $\boldsymbol{X}$. Let $\hat{\boldsymbol{Y}} \in \mathbb{R}^{B\times 1\times D}$ be a tensor that contains B predicted values generated by the CoDR model for each input sequence in $\boldsymbol{X}$. Let $\boldsymbol{\theta}$ be a vector that contains all the learnable parameters of the CoDR model. We assume that $\boldsymbol{Z}$ is a stationary time series data that follows a multivariate normal distribution with mean vector $\boldsymbol{\mu}$

Table 7
Abbreviations and meanings.

Abbreviation	Meaning
ANN	Artificial Neural Network
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
CNN	Convolutional Neural Network
CNN1D	One-Dimensional CNN
CoDR	Collaborative Directional Representation
CRNN	Convolutional Recurrent Neural Network
CRNNRes	Residual Convolutional Recurrent Neural Network
CVRMSE	Coefficient of Variation of Root Mean Square Error
DR	Directional Representation
DSANet	Dual Self-Attention Network
ED	Encoder-Decoder
EG	Elia Group
ES	Exponential Smoothing
FEDformer	Frequency Enhanced Decomposed Transformer
FiLM	Feature-Wise Linear Modulation
GAIN	Graph Ambient Intelligence Neural Network
GRU	Gated Recurrent Unit
GNNs	Graph Neural Networks
LSTM	Long Short-Term Memory
LSTNet	Long- and Short-Term Network
MAE	Mean Absolute Error
MSE	Mean Square Error
MSL	Multivariate Shapelet Learning
NWP	Numerical Weather Prediction
PV	Photovoltaic
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
StemGNN	Spatial–Temporal Embedding with Multi-Graph CNN
SVM	Support Vector Machines
SZF	Single-site-based Zero-shot PV power Forecasting
TPA-LSTM	Temporal Pattern Attention with LSTM
VARMA	Vector Autoregressive Moving Average
WFD	Window Fluctuation De-extraction
WFE	Window Fluctuation Extraction

Table 8
Hyper-parameter settings.

Model	Parameter	Option range
LSTM GRU ED	Hidden size	$\{2^4, 2^5, 2^6\}$
DLinear	Decomposition kernel size	3–9 (2 per step)
FiLM	The dimension of the model	$\{2^4, 2^5, 2^6\}$
CNN1D	CNN kernel size	3–9 (2 per step)
	CNN out channels	$\{2^2, 2^3, 2^4, 2^5, 2^6\}$
CNNRNN	GRU hidden size	$\{2^4, 2^5, 2^6\}$
	GRU layers	1–3 (1 per step)
CNNRNNRes	Residual window size	1–7 (1 per step)
	Residual ratio	0.1–0.5 (0.1 per step)
LSTNet	Skip window size	1–7 (1 per step)
	Skip GRU hidden size	$\{2^4, 2^5, 2^6\}$
	Skip GRU layers	1–3 (1 per step)
Transformer	Encoder layers	1–3 (1 per step)
Informer	Decoder layers	1–3 (1 per step)
Autoformer	The label length	1–10 (1 per step)
FEDformer	The numbers of heads	$\{2^2, 2^3, 2^4\}$
	The dimension of the model	$\{2^4, 2^5, 2^6\}$
DSANet	CNN kernel size	3–9 (2 per step)
	CNN out channels	$\{2^2, 2^3, 2^4, 2^5, 2^6\}$
	Attention layers	1–3 (1 per step)
	The numbers of heads	$\{2^2, 2^3, 2^4\}$
	The dimension of the model	$\{2^4, 2^5, 2^6\}$
TPA-LSTM	GRU hidden size	$\{2^4, 2^5, 2^6\}$
	GRU layers	1–3 (1 per step)
	Residual window size	1–10 (1 per step)
StemGNN	Block size	1–10 (1 per step)
	Leaky rate	0.1–0.3 (0.1 per step)
GAIN	GAT hidden size	$\{2^4, 2^5, 2^6\}$
	The number of heads of GAT	$\{2^0, 2^1, 2^2, 2^3, 2^8\}$
MSL	Shapelet size	$\{2^2, 2^3, 2^4, 2^5, 2^6\}$

and covariance matrix $\boldsymbol{\Sigma}$. We also assume that $\boldsymbol{Z}$ has a linear trend component $\boldsymbol{T}$ and a seasonal component $\boldsymbol{S}$ that can be estimated by some methods such as moving average or exponential smoothing. We denote the fluctuation component of $\boldsymbol{Z}$ by $\boldsymbol{F} = \boldsymbol{Z} - \boldsymbol{T} - \boldsymbol{S}$. We define the directional fluctuation of $\boldsymbol{F}$ as the change of its values along different directions in the multidimensional space. For example, in a two-dimensional space, the directional fluctuation can be measured by the angle or slope of each data point relative to its previous point. The directional fluctuation can reflect the complex and nonlinear characteristics of $\boldsymbol{F}$, such as trends, cycles, spikes, and outliers. We formulate the solar irradiance forecasting problem as a function approximation problem:

$$\min_{\theta} L(\theta) = \min_{\theta} \frac{1}{2B} \sum_{i=1}^{B} \|f(\boldsymbol{X}_i; \theta) - \boldsymbol{Y}_i\|_2^2,$$

where $f : \mathbb{R}^{T\times D} \to \mathbb{R}^{1\times D}$ is the CoDR model that maps an input sequence $\boldsymbol{X}_i$ to a predicted value $\hat{\boldsymbol{Y}}_i$, and $\|\cdot\|_2$ is the Euclidean norm. The objective function $L(\theta)$ is the MSE between the predicted values and the true values, which measures the accuracy of the CoDR model.

We now present the lemmas, theorems, and propositions that show the convergence, stability, and optimality properties of the CoDR model.

Lemma 1. *The CoDR model is a universal approximator, which means that it can approximate any continuous function on a compact subset of $\mathbb{R}^{T\times D}$ with arbitrary accuracy, given enough hidden units and parameters.*

Proof. Let $f : \mathbb{R}^{T\times D} \to \mathbb{R}^{1\times D}$ be the CoDR model that maps an input sequence $\boldsymbol{X}_i$ to a predicted value $\hat{\boldsymbol{Y}}_i$. Let $K \subset \mathbb{R}^{T\times D}$ be a compact subset of the input space. Let $g : K \to \mathbb{R}^D$ be any continuous function on K. We want to show that for any $\epsilon > 0$, there exists a CoDR model f such that $\|f(\boldsymbol{X}_i) - g(\boldsymbol{X}_i)\|_2 < \epsilon$ for all $\boldsymbol{X}_i \in K$.

The CoDR model can be seen as a composition of several nonlinear functions, such as ReLU, softmax, and linear layers. Each of these functions can be represented by a feedforward neural network with one hidden layer and an appropriate activation function. By using the universal approximation theorem for neural networks (Cybenko, 1989), we can show that each of these functions can approximate any continuous function on a compact subset of $\mathbb{R}^{T\times D}$ with arbitrary accuracy, given enough hidden units and parameters. For example, ReLU activation function (Hornik, 1991), softmax activation function (Funahashi and Nakamura, 1993), and linear activation function (Leshno et al., 1993).

Therefore, by using the composition property of continuous functions, we can show that the CoDR model, which is a composition of several nonlinear functions, can also approximate any continuous function on a compact subset of $\mathbb{R}^{T\times D}$ with arbitrary accuracy, given enough hidden units and parameters. For example, the general results on composition of continuous functions (Shapiro, 2012), and the specific results on composition of neural networks (Lorentz et al., 1996).

Hence, for any $\epsilon > 0$, there exists a CoDR model f such that $\|f(\boldsymbol{X}_i) - g(\boldsymbol{X}_i)\|_2 < \epsilon$ for all $\boldsymbol{X}_i \in K$, which proves the lemma.

Theorem 1. *The CoDR model is a convergent model, which means that it can find a global minimum of the objective function $L(\theta)$ with probability one, given enough training data and iterations.*

Proof. Let $f : \mathbb{R}^{T\times D} \to \mathbb{R}^{1\times D}$ be the CoDR model that maps an input sequence $\boldsymbol{X}_i$ to a predicted value $\hat{\boldsymbol{Y}}_i$. Let $L(\theta) = \frac{1}{2B} \sum_{i=1}^{B} \|f(\boldsymbol{X}_i; \theta) - \boldsymbol{Y}_i\|_2^2$ be the objective function that measures the MSE between the predicted values and the true values. Let θ be a vector that contains all the learnable parameters of the CoDR model. We want to show that for any $\epsilon > 0$, there exists an iteration T such that with probability one,

$\|\nabla L(\boldsymbol{\theta}_T)\|_2 < \epsilon$, where $\nabla L(\boldsymbol{\theta}_T)$ is the gradient of $L(\boldsymbol{\theta})$ at iteration T, and $\boldsymbol{\theta}_T$ is the parameter vector at iteration T.

The CoDR model is trained by using stochastic gradient descent (SGD) with momentum, which is a variant of gradient descent that updates the parameters by using a weighted average of the previous gradients and the current gradients. The update rule of SGD with momentum is given by:

$$\begin{aligned} \boldsymbol{v}_t &= \beta \boldsymbol{v}_{t-1} + (1-\beta)\nabla L(\boldsymbol{\theta}_{t-1}), \\ \boldsymbol{\theta}_t &= \boldsymbol{\theta}_{t-1} - \alpha_t \boldsymbol{v}_t, \end{aligned} \tag{17}$$

where $\boldsymbol{v}_t$ is the velocity vector at iteration t, β is the momentum coefficient, α_t is the learning rate at iteration t, and $\nabla L(\boldsymbol{\theta}_{t-1})$ is an unbiased estimate of $\nabla L(\boldsymbol{\theta}_{t-1})$ obtained by sampling a mini-batch of data.

By using the convergence analysis of SGD with momentum (Qian, 1999), we can show that under some mild conditions, such as bounded gradients and learning rates, SGD with momentum can find a global minimum of $L(\boldsymbol{\theta})$ with probability one, given enough training data and iterations. For example, convex objective functions (Bottou and Bousquet, 2007), non-convex objective functions (Du et al., 2019), and stochastic objective functions (Gower et al., 2021).

Therefore, to apply the convergence result of SGD with momentum to the CoDR model, we only need to verify that the objective function $L(\boldsymbol{\theta})$ satisfies the conditions required for the convergence result. In particular, we need to show that:

- The objective function $L(\boldsymbol{\theta})$ is continuous and differentiable with respect to $\boldsymbol{\theta}$.
- The gradient $\nabla L(\boldsymbol{\theta})$ is bounded by some constant G, i.e., $\|\nabla L(\boldsymbol{\theta})\|_2 \leq G$ for all $\boldsymbol{\theta}$.
- The learning rate α_t is chosen such that $\sum_{t=1}^{\infty} \alpha_t = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$.

The first condition is easy to verify, since $L(\boldsymbol{\theta})$ is a quadratic function of $\boldsymbol{\theta}$, which is continuous and differentiable everywhere. The second condition can be verified by using some properties of norms and linear algebra. For example, we can write:

$$\begin{aligned} \|\nabla L(\boldsymbol{\theta})\|_2 &= \frac{1}{B}\| \sum_{i=1}^{B} \nabla f(\boldsymbol{X}_i;\boldsymbol{\theta})(f(\boldsymbol{X}_i;\boldsymbol{\theta}) - \boldsymbol{Y}_i)\|_2 \\ &\leq \frac{1}{B} \sum_{i=1}^{B} \|\nabla f(\boldsymbol{X}_i;\boldsymbol{\theta})\|_2 \|f(\boldsymbol{X}_i;\boldsymbol{\theta}) - \boldsymbol{Y}_i\|_2 \\ &\leq \frac{1}{B} \sum_{i=1}^{B} G_f G_y, \end{aligned} \tag{18}$$

where G_f is a constant that bounds the norm of $\nabla f(\boldsymbol{X}_i;\boldsymbol{\theta})$, and G_y is a constant that bounds the norm of $f(\boldsymbol{X}_i;\boldsymbol{\theta}) - \boldsymbol{Y}_i$. These constants can be obtained by using some properties of the CoDR model and the input data, such as boundedness, smoothness, and Lipschitz continuity. Therefore, we can conclude that $\|\nabla L(\boldsymbol{\theta})\|_2$ is also bounded by some constant G, which depends on G_f and G_y. The third condition can be satisfied by choosing a suitable learning rate schedule, such as a constant, decreasing, or adaptive learning rate. For example, we can use a constant learning rate $\alpha_t = \alpha$ for all t, where α is a small positive number. Then, we have $\sum_{t=1}^{\infty} \alpha_t = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2 = \alpha^2 T < \infty$, where T is the total number of iterations.

Hence, by verifying these conditions, we can apply the convergence result of SGD with momentum to the CoDR model, and show that for any $\epsilon > 0$, there exists an iteration T such that with probability one, $\|\nabla L(\boldsymbol{\theta}_T)\|_2 < \epsilon$. This proves the theorem.

Proposition 1. *The CoDR model is a stable model, which means that it can handle small perturbations or noises in the input data without producing large errors in the output data.*

Proof. Let $f : \mathbb{R}^{T\times D} \to \mathbb{R}^{1\times D}$ be the CoDR model that maps an input sequence $\boldsymbol{X}_i$ to a predicted value $\hat{\boldsymbol{Y}}_i$. Let $L(\boldsymbol{\theta}) = \frac{1}{2B}\sum_{i=1}^{B} \|f(\boldsymbol{X}_i;\boldsymbol{\theta}) - \boldsymbol{Y}_i\|_2^2$ be the objective function that measures the mean squared error (MSE) between the predicted values and the true values. Let $\boldsymbol{\theta}$ be a vector that contains all the learnable parameters of the CoDR model. We want to show that for any $\epsilon > 0$, there exists a $\delta > 0$ such that for any $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ in $\mathbb{R}^{T\times D}$, if $\|\boldsymbol{X}_i - \boldsymbol{X}_j\|_2 < \delta$, then $\|f(\boldsymbol{X}_i;\boldsymbol{\theta}) - f(\boldsymbol{X}_j;\boldsymbol{\theta})\|_2 < \epsilon$.

The CoDR model consists of seven main steps: data preprocessing, fluctuation extraction, directional representation, linearization, de-extraction, mapping and post-processing. Each of these steps can be seen as a function that transforms the input data into a different representation or space. For example, let $g : \mathbb{R}^{T\times D} \to [0,1]^{T\times D}$ be the data normalization function that scales the input data into a unit interval; let $h : [0,1]^{T\times D} \to [-1,1]^{T\times D}$ be the fluctuation extraction function that subtracts the mean value from each input sequence; let $k : [-1,1]^{T\times D} \to [-\pi/2, \pi/2]^{T-1}$ be the directional representation function that converts each input sequence into an angle vector; and so on. Then, we can write $f = p \circ m \circ l \circ k \circ h \circ g$, where p, m, and l are the post-processing, mapping, and linearization functions, respectively.

Each of these functions is Lipschitz continuous with respect to some metric on their domain and codomain. This means that there exists a constant K_i for each function f_i such that for any $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in their domain,

$$d_i(f_i(\boldsymbol{x}_i), f_i(\boldsymbol{x}_j)) \leq K_i d_i(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{19}$$

where d_i is some metric on their domain and codomain. For example, Lipschitz continuity of data normalization with respect to Euclidean metric (Granas and Dugundji, 2003); Lipschitz continuity of fluctuation extraction with respect to Euclidean metric (Bochnak et al., 2013); Lipschitz continuity of directional representation with respect to angular metric (Serre, 2009); and Lipschitz continuity of linearization with respect to Euclidean metric (Jeffreys and Jeffreys, 1999).

Therefore, by using the composition property of Lipschitz continuous functions (Rudin, 1953), we can show that the CoDR model, which is a composition of several Lipschitz continuous functions, is also a Lipschitz continuous function with respect to some metric on its domain and codomain. This means that there exists a constant K for the CoDR model such that for any $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ in $\mathbb{R}^{T\times D}$,

$$d(f(\boldsymbol{X}_i;\boldsymbol{\theta}), f(\boldsymbol{X}_j;\boldsymbol{\theta})) \leq K d(\boldsymbol{X}_i, \boldsymbol{X}_j), \tag{20}$$

where d is some metric on $\mathbb{R}^{T\times D}$ and $\mathbb{R}^{1\times D}$. This implies that the CoDR model is stable, since small changes in the input data will result in small changes in the output data. In particular, for any $\epsilon > 0$, there exists a $\delta > 0$ such that for any $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ in $\mathbb{R}^{T\times D}$, if $\|\boldsymbol{X}_i - \boldsymbol{X}_j\|_2 < \delta$, then $\|f(\boldsymbol{X}_i;\boldsymbol{\theta}) - f(\boldsymbol{X}_j;\boldsymbol{\theta})\|_2 < \epsilon$. This proves the proposition.

A.3. Computational complexity analysis

This subsection analyzes the computational complexity of the CoDR model, taking into account both the time complexity and the space complexity.

A.3.1. Time complexity

The time complexity of the CoDR model is primarily determined by the matrix multiplications and the softmax operations within the directional representation (DR) unit, the linearization step, and the mapping step. Assuming a batch size of B, a window size of T, a hidden size of P, and a number of variables of D, the dominant operations and their associated time complexities can be described as follows. The Window Fluctuation Extraction (WFE) step involves element-wise subtraction of the moving average from the input tensor, resulting in a time complexity of $\mathcal{O}(B \times T \times D)$. Within the Directional Representation (DR), the rectification and transformation steps involve element-wise operations and matrix multiplications, respectively, both with a time complexity of $\mathcal{O}(B \times T \times D)$. The highlighting step employs a softmax function with a time complexity of $\mathcal{O}(B \times T \times D)$. Finally, the linearization step within DR performs matrix multiplications, yielding a time complexity of $\mathcal{O}(B \times T \times D \times P)$. The Linearization step, separate from

the DR unit, employs a linear layer and has a time complexity of $\mathcal{O}(B \times T \times D \times P)$. The Window Fluctuation De-extraction (WFD) step entails element-wise addition with a time complexity of $\mathcal{O}(B \times P \times D)$. Lastly, the Mapping step combines the intermediate data and applies a linear layer, leading to a time complexity of $\mathcal{O}(B \times P \times D)$. Data Postprocessing involves element-wise operations with a time complexity of $\mathcal{O}(B \times D)$.

Taking all steps into consideration, the overall time complexity of CoDR, encompassing both training and inference, is $\mathcal{O}(B \times T \times D \times P)$. This analysis indicates that the time complexity of CoDR exhibits a linear growth pattern with respect to the batch size, window size, number of variables, and hidden size.

A.3.2. Space complexity

The space complexity of the CoDR model is largely dictated by the storage requirements for model parameters, input and output tensors, and intermediate activation values. The primary contributors to the space complexity are as follows. The model parameters, which encompass the weight matrices and bias vectors in the DR unit, the linearization layer, and the mapping step, require a storage space complexity of $\mathcal{O}(T \times D + D \times P + P \times D + P + P \times D + D)$. Storing the input and output tensors necessitates $\mathcal{O}(B \times T \times D)$ and $\mathcal{O}(B \times D)$ space, respectively. Moreover, intermediate activation values must be stored for backpropagation during the training phase, contributing a space complexity of $\mathcal{O}(B \times T \times D \times P)$.

Therefore, the overall space complexity of CoDR amounts to $\mathcal{O}(B \times T \times D \times P)$. In line with the time complexity, the space complexity also demonstrates a linear growth trend with respect to the batch size, window size, number of variables, and hidden size.

References

Ahmad, T., Zhou, N., Zhang, Z., Tang, W., 2024. Enhancing probabilistic solar PV forecasting: Integrating the NB-DST method with deterministic models. Energies 17 (10), 2392. http://dx.doi.org/10.3390/en17102392.

Akhter, M.N., Mekhilef, S., Mokhlis, H., Mohamed Shah, N., 2019. Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. IET Renew. Power Gener. 13 (7), 1009–1023. http://dx.doi.org/10.1049/iet-rpg.2018.5649.

Alcañiz, A., Lindfors, A.V., Zeman, M., Ziar, H., Isabella, O., 2023. Effect of climate on photovoltaic yield prediction using machine learning models. Global Challenges 7 (1), 2200166. http://dx.doi.org/10.1002/gch2.202200166.

Antonopoulos, V.Z., Papamichail, D.M., Aschonitis, V.G., Antonopoulos, A.V., 2019. Solar radiation estimation methods using ANN and empirical models. Comput. Electron. Agric. 160, 160–167. http://dx.doi.org/10.1016/j.compag.2019.03.022.

Asif, M.I., Alam, A.M., Deeba, S.R., Aziz, T., et al., 2019. Forecasting of photovoltaic power generation: Techniques and key factors. In: 2019 IEEE Region 10 Symposium. TENSYMP, pp. 457–461. http://dx.doi.org/10.1109/tensymp46218.2019.8971337.

Aslam, M., Lee, S., Khang, S., Hong, S., 2021. Two-stage attention over LSTM with Bayesian optimization for day-ahead solar power forecasting. IEEE Access 9, 107387–107398. http://dx.doi.org/10.1109/access.2021.3100105.

Bochnak, J., Coste, M., Roy, M.-F., 2013. Real Algebraic Geometry, vol. 36, Springer Science & Business Media.

Bottou, L., Bousquet, O., 2007. The tradeoffs of large scale learning. In: Proceedings of the 21st Annual Conference on Neural Information Processing Systems. Curran Associates, Inc., Vancouver, British Columbia, Canada, pp. 161–168.

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J., et al., 2020. Spectral temporal graph neural network for multivariate time-series forecasting. In: Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems. Virtual, pp. 17766–17778. http://dx.doi.org/10.48550/arxiv.2103.077191.

Chen, Z., Li, Z., Zhang, X., 2019. Wind power forecasting based on LSTM neural network. Int. J. Recent Technol. Eng. 8 (2S11), 3810–3814. http://dx.doi.org/10.35940/ijrte.b1298.0982s1119.

Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches. In: Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation. Association for Computational Linguistics, Doha, Qatar, pp. 103–111. http://dx.doi.org/10.3115/v1/w14-4012.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. http://dx.doi.org/10.48550/arxiv.1412.3555, CoRR abs/1412.3555.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Math. Control, Signals Syst. 2 (4), 303–314. http://dx.doi.org/10.1007/bf02551274.

Das, U.K., Tey, K.S., Seyedmahmoudian, M., Mekhilef, S., Idris, M.Y.I., Van Deventer, W., Horan, B., Stojcevski, A., 2018. Forecasting of photovoltaic power generation and model optimization: A review. Renew. Sustain. Energy Rev. 81, 912–928. http://dx.doi.org/10.1016/j.rser.2017.08.017.

Du, S.S., Zhai, X., Póczos, B., Singh, A., 2019. Gradient descent provably optimizes over-parameterized neural networks. In: Proceedings of the 7th International Conference on Learning Representations. OpenReview.net, New Orleans, LA, USA.

Funahashi, K., Nakamura, Y., 1993. Approximation of dynamical systems by continuous time recurrent neural networks. Neural Netw. 6 (6), 801–806. http://dx.doi.org/10.1016/s0893-6080(05)80125-x.

Gaboitaolelwe, J., Zungeru, A.M., Yahya, A., Lebekwe, C.K., Vinod, D.N., Salau, A.O., 2023. Machine learning based solar photovoltaic power forecasting: A review and comparison. IEEE Access 11, 40820–40845. http://dx.doi.org/10.1109/access.2023.3270041.

Gower, R.M., Richtárik, P., Bach, F., 2021. Stochastic quasi-gradient methods: Variance reduction via Jacobian sketching. Math. Program. 188 (1), 135–192. http://dx.doi.org/10.1007/s10107-020-01506-0.

Granas, A., Dugundji, J., 2003. Fixed Point Theory, vol. 14, Springer.

Guermoui, M., Fezzani, A., Mohamed, Z., Rabehi, A., Ferkous, K., Bailek, N., Bouallit, S., Riche, A., Bajaj, M., Dost Mohammadi, S.A., et al., 2024. An analysis of case studies for advancing photovoltaic power forecasting through multi-scale fusion techniques. Sci. Rep. 14 (1), 6653. http://dx.doi.org/10.1038/s41598-024-57398-z.

Guo, X., Mo, Y., Yan, K., 2022. Short-term photovoltaic power forecasting based on historical information and deep learning methods. Sensors 22 (24), 9630. http://dx.doi.org/10.3390/s22249630.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, pp. 770–778. http://dx.doi.org/10.1109/cvpr.2016.90.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Netw. 4 (2), 251–257. http://dx.doi.org/10.1016/0893-6080(91)90009-t.

Hu, Y., Liu, H., Wu, S., Zhao, Y., Wang, Z., Liu, X., 2024. Temporal collaborative attention for wind power forecasting. Appl. Energy 357, 122502. http://dx.doi.org/10.1016/j.apenergy.2023.122502.

Huang, S., Wang, D., Wu, X., Tang, A., 2019. DSANet: Dual self-attention network for multivariate time series forecasting. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. ACM, Beijing, China, pp. 2129–2132. http://dx.doi.org/10.1145/3357384.3358132.

Huang, Y., Zhao, Y., Wang, Z., Liu, X., Fu, Y., 2024. Sparse dynamic graph learning for district heat load forecasting. Appl. Energy 371, 123685. http://dx.doi.org/10.1016/j.apenergy.2024.123685.

Huang, Y., Zhao, Y., Wang, Z., Liu, X., Liu, H., Fu, Y., 2023. Explainable district heat load forecasting with active deep learning. Appl. Energy 350, 121753. http://dx.doi.org/10.1016/j.apenergy.2023.121753.

Hussain, A., Khan, Z.A., Hussain, T., Ullah, F.U.M., Rho, S., Baik, S.W., 2022. A hybrid deep learning-based network for photovoltaic power forecasting. Complexity 2022 (1), 7040601. http://dx.doi.org/10.1155/2022/7040601.

Islam, M.T., Huda, N., Abdullah, A., Saidur, R., 2018. A comprehensive review of state-of-the-art concentrating solar power (CSP) technologies: Current status and research trends. Renew. Sustain. Energy Rev. 91, 987–1018. http://dx.doi.org/10.1016/j.rser.2018.04.097.

Jeffreys, H., Jeffreys, B., 1999. Methods of Mathematical Physics. Cambridge University Press.

Ju, Y., Li, J., Sun, G., 2020. Ultra-short-term photovoltaic power prediction based on self-attention mechanism and multi-task learning. Ieee Access 8, 44821–44829. http://dx.doi.org/10.1109/access.2020.2978635.

Khan, Z.A., Hussain, T., Baik, S.W., 2023. Dual stream network with attention mechanism for photovoltaic power forecasting. Appl. Energy 338, 120916. http://dx.doi.org/10.1016/j.apenergy.2023.120916.

Khan, W., Walker, S., Zeiler, W., 2022. Improved solar photovoltaic energy generation forecast using deep learning-based ensemble stacking approach. Energy 240, 122812. http://dx.doi.org/10.1016/j.energy.2021.122812.

Kharlova, E., May, D., Musilek, P., 2020. Forecasting photovoltaic power production using a deep learning sequence to sequence model with attention. In: Proceedings of the International Joint Conference on Neural Networks. pp. 1–7. http://dx.doi.org/10.1109/ijcnn48605.2020.9207573.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations. OpenReview.net, San Diego, CA, USA, pp. 1–15. http://dx.doi.org/10.1145/1830483.1830503.

Kisvari, A., Lin, Z., Liu, X., 2021. Wind power forecasting–A data-driven method along with gated recurrent neural network. Renew. Energy 163, 1895–1909. http://dx.doi.org/10.1016/j.renene.2020.10.119.

Lai, G., Chang, W., Yang, Y., Liu, H., 2018. Modeling long- and short-term temporal patterns with deep neural networks. In: Proceedings of the 41st International Conference on Research and Development in Information Retrieval. ACM, Ann Arbor, MI, USA, pp. 95–104. http://dx.doi.org/10.1145/3209978.3210006.

Lateko, A.A., Yang, H., Huang, C., 2022. Short-term PV power forecasting using a regression-based ensemble method. Energies 15 (11), 4171. http://dx.doi.org/10.3390/en15114171.

Leshno, M., Lin, V.Y., Pinkus, A., Schocken, S., 1993. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw. 6 (6), 861–867. http://dx.doi.org/10.1016/s0893-6080(05)80131-5.

Li, Y., Xue, C., Zargari, F., Li, Y., 2023. From graph theory to graph neural networks (GNNs): The opportunities of GNNs in power electronics. IEEE Access 11, 145067–145084. http://dx.doi.org/10.1109/access.2023.3345795.

Lin, G., Li, L., Tseng, M., Liu, H., Yuan, D., Tan, R.R., 2020. An improved moth-flame optimization algorithm for support vector machine prediction of photovoltaic power generation. J. Clean. Prod. 253, 119966. http://dx.doi.org/10.1016/j.jclepro.2020.119966.

López Santos, M., García-Santiago, X., Echevarría Camarero, F., Blázquez Gil, G., Carrasco Ortega, P., 2022. Application of temporal fusion transformer for day-ahead PV power forecasting. Energies 15 (14), 5232. http://dx.doi.org/10.3390/en15145232.

Lorentz, G.G., von Golitschek, M., Makovoz, Y., 1996. Constructive Approximation: Advanced Problems, vol. 304, Citeseer.

Malek, S., Melgani, F., Bazi, Y., 2018. One-dimensional convolutional neural networks for spectroscopic signal regression. J. Chemometrics 32 (5), e2977. http://dx.doi.org/10.1002/cem.2977.

Mekhilef, S., Saidur, R., Safari, A., 2011. A review on solar energy use in industries. Renew. Sustain. Energy Rev. 15 (4), 1777–1790. http://dx.doi.org/10.1016/j.rser.2010.12.018.

Mellit, A., Massi Pavan, A., Ogliari, E., Leva, S., Lughi, V., 2020. Advanced methods for photovoltaic output power forecasting: A review. Appl. Sci. 10 (2), 487. http://dx.doi.org/10.3390/app10020487.

Pan, C., Tan, J., Feng, D., Li, Y., 2019. Very short-term solar generation forecasting based on LSTM with temporal attention mechanism. In: Proceedings of the IEEE 5th International Conference on Computer and Communications. pp. 267–271. http://dx.doi.org/10.1109/iccc47050.2019.9064298.

Panda, S., Dhaka, R.K., Panda, B., Pradhan, A., Jena, C., Nanda, L., 2022. A review on application of machine learning in solar energy & photovoltaic generation prediction. In: Proceedings of the International Conference on Electronics and Renewable Systems. pp. 1180–1184. http://dx.doi.org/10.1109/icears53579.2022.9752404.

Park, S., Kim, D., Moon, J., Hwang, E., 2023. Zero-shot photovoltaic power forecasting scheme based on a deep learning model and correlation coefficient. Int. J. Energy Res. 2023 (1), 9936542. http://dx.doi.org/10.1155/2023/9936542.

Peng, Q., Sangwongwanich, A., Yang, Y., Blaabjerg, F., 2020. Grid-friendly power control for smart photovoltaic systems. Sol. Energy 210, 115–127. http://dx.doi.org/10.1016/j.solener.2020.05.001.

Peng, K., Tan, X., Wang, Z., Wu, Y., Cai, Y., 2024. PV power prediction based on feature selection and GMM clustering. J. Phys.: Conf. Ser. 012012. http://dx.doi.org/10.1088/1742-6596/2683/1/012012.

Qian, N., 1999. On the momentum term in gradient descent learning algorithms. Neural Netw. 12 (1), 145–151. http://dx.doi.org/10.1016/s0893-6080(98)00116-6.

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., Cottrell, G.W., 2017. A dual-stage attention-based recurrent neural network for time series prediction. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. ijcai.org, Melbourne, Australia, pp. 2627–2633. http://dx.doi.org/10.24963/ijcai.2017/366.

Rudin, W., 1953. Principles of Mathematical Analysis.

Serre, J., 2009. Lie Algebras and Lie Groups: 1964 Lectures Given at Harvard University. Springer.

Shapiro, J.H., 2012. Composition Operators: and Classical Function Theory. Springer Science & Business Media.

Shi, X., Chen, Z., Wang, H., Yeung, D., Wong, W., Woo, W., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Proceedings of the 28th International Conference on Advances in Neural Information Processing Systems. Montreal, Quebec, Canada, pp. 802–810. http://dx.doi.org/10.48550/arxiv.1506.04214.

Shi, J., Wang, Y., Zhou, Y., Ma, Y., Gao, J., Wang, S., Fu, Z., 2024. Bayesian optimization - LSTM modeling and time frequency correlation mapping based probabilistic forecasting of ultra-short-term photovoltaic power outputs. IEEE Trans. Ind. Appl. 60 (2), 2422–2430. http://dx.doi.org/10.1109/tia.2023.3334700.

Shih, S., Sun, F., Lee, H., 2019. Temporal pattern attention for multivariate time series forecasting. Mach. Learn. 108 (8–9), 1421–1441. http://dx.doi.org/10.1007/s10994-019-05815-0.

Sun, Y., Yu, H., Geng, G., Chen, C., Jiang, Q., 2023. Scalable multi-site photovoltaic power forecasting based on stream computing. IET Renew. Power Gener. 17 (9), 2379–2390. http://dx.doi.org/10.1049/rpg2.12766.

Trivedi, R., Patra, S., Khadem, S., 2022. A data-driven short-term pv generation and load forecasting approach for microgrid applications. IEEE J. Emerg. Sel. Top. Ind. Electron. 3 (4), 911–919. http://dx.doi.org/10.1109/jestie.2022.3179961.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Proceedings of the 31th International Conference on Neural Information Processing Systems. Curran Associates, Inc., Long Beach, CA, USA, pp. 5998–6008. http://dx.doi.org/10.5555/3295222.3295349.

Venkatraman, D., Pitchaipillai, V., 2024. Deep learning-based auto-LSTM approach for renewable energy forecasting: A hybrid network model. Trait. Signal 41 (1), http://dx.doi.org/10.18280/ts.410148.

Wang, Z., Cai, B., 2022. COVID-19 cases prediction in multiple areas via shapelet learning. Appl. Intell. 52 (1), 595–606. http://dx.doi.org/10.1007/s10489-021-02391-6.

Wang, Y., Fu, W., Zhang, X., Zhen, Z., Wang, F., 2024. Dynamic directed graph convolution network based ultra-short-term forecasting method of distributed photovoltaic power to enhance the resilience and flexibility of distribution network. IET Gener., Transm. Distrib. 18 (2), 337–352. http://dx.doi.org/10.1049/gtd2.12963.

Wang, Z., Liu, X., Huang, Y., Zhang, P., Fu, Y., 2023. A multivariate time series graph neural network for district heat load forecasting. Energy 278, 127911. http://dx.doi.org/10.1016/j.energy.2023.127911.

Wu, K., Peng, X., Li, Z., Cui, W., Yuan, H., Lai, C.S., Lai, L.L., 2022. A short-term photovoltaic power forecasting method combining a deep learning model with trend feature extraction and feature selection. Energies 15 (15), 5410. http://dx.doi.org/10.3390/en151554100.

Wu, H., Xu, J., Wang, J., Long, M., 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. Virtual, pp. 22419–22430. http://dx.doi.org/10.48550/arxiv.2106.13008.

Wu, Y., Yang, Y., Nishiura, H., Saitoh, M., 2018. Deep learning for epidemiological predictions. In: Proceedings of the 41st International Conference on Research and Development in Information Retrieval. ACM, Ann Arbor, MI, USA, pp. 1085–1088. http://dx.doi.org/10.1145/3209978.3210077.

Yadav, H.K., Pal, Y., Tripathi, M.M., 2015. Photovoltaic power forecasting methods in smart power grid. In: Proceedings of the Annual IEEE India Conference. pp. 1–6. http://dx.doi.org/10.1109/indicon.2015.7443522.

Zeng, A., Chen, M., Zhang, L., Xu, Q., 2023. Are transformers effective for time series forecasting? In: Proceedings of the 37th AAAI Conference on Artificial Intelligence. AAAI Press, pp. 11121–11128. http://dx.doi.org/10.1609/aaai.v37i9.26317.

Zhou, T., Ma, Z., Wang, X., Wen, Q., Sun, L., Yao, T., Yin, W., Jin, R., 2022a. FiLM: Frequency improved Legendre memory model for long-term time series forecasting. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. New Orleans, LA, USA, pp. 12677–12690. http://dx.doi.org/10.48550/arxiv.2109.03254.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R., 2022b. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: Proceedings of the 39th International Conference on Machine Learning. PMLR, Baltimore, Maryland, USA, pp. 27268–27286. http://dx.doi.org/10.48550/arxiv.2201.12740.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W., 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence. AAAI Press, Virtual Event, pp. 11106–11115. http://dx.doi.org/10.1609/aaai.v35i12.17325.

Zhu, T., Li, Y., Li, Z., Guo, Y., Ni, C., 2022. Inter-hour forecast of solar radiation based on long short-term memory with attention mechanism and genetic algorithm. Energies 15 (3), 1062. http://dx.doi.org/10.3390/en15031062.

Zhu, H., Sun, Y., Jiang, T., Zhang, X., Zhou, H., Hu, S., Kang, M., 2024. An FCM based weather type classification method considering photovoltaic output and meteorological characteristics and its application in power interval forecasting. IET Renew. Power Gener. 18 (2), 238–260. http://dx.doi.org/10.1049/rpg2.12917.

Zsiborács, H., Pintér, G., Vincze, A., Birkner, Z., Baranyai, N.H., 2021. Grid balancing challenges illustrated by two European examples: Interactions of electric grids, photovoltaic power generation, energy storage and power generation forecasting. Energy Rep. 7, 3805–3818. http://dx.doi.org/10.1016/j.egyr.2021.06.007.