Multi-modal Spatio-temporal Meteorological Forecasting with Deep Neural Network

Xinbang Zhang[a,b], Jinqi Zhao[a,b], Tingzhao Yu[c], Shiming Xiang[a,b], Qiuming Kuang[c], Véronique Prinet[a], Chunhong Pan[a]

[a] *The Department of National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, 95 Zhongguancun East Road, Beijing, 100190, Beijing, China*
[b] *The School of Artificial Intelligence, University of Chinese Academy of Sciences, 1 Yanqihu East Road, Beijing, 100049, Beijing, China*
[c] *The Public Meteorological Service Center, China Meteorological Administration, 46 Zhongguancun South Street, Beijing, 100081, Beijing, China*

Abstract

Meteorological forecasting is a typical and fundamental problem in the remote sensing field. Although many brilliant forecasting methods have been developed, long-term (a few days ahead) meteorological prediction still relies on traditional Numerical Weather Prediction (NWP) that is not competent for the oncoming flood of meteorological data. To improve the forecasting ability faced with meteorological big data, this article adopts the Automated Machine Learning (AutoML) technique and proposes a deep learning framework to model the dynamics of multi-modal meteorological data along spatial and temporal dimensions. Spatially, a convolution based network is developed to extract the spatial context of multi-modal meteorological data. Considering the complex relationship between different modalities, the Neural Architecture Search (NAS) method is introduced to automate the designing procedure of the fusion network in a purely data-driven manner. As for the temporal dimension, an encoder-decoder structure is built to exhaustively model the temporal dynamics of the embedding sequence. Specializing for the numerical sequence representation transformation, the multi-head attention module endows the proposed model with the ability to forecast future data. Generally speaking, the whole framework could be optimized with the standard back-propagation, yielding an end-to-end learning mechanism. To investigate its feasibility, the proposed model is evaluated with four typical meteorological modalities including temperature, relative humidity, and two

components of wind, which are all restricted under the region whose latitude and longitude range from 0° to 55° N and 70° E to 140° E, respectively. Experiments on two datasets with different resolutions verify that deep learning is effective as an operational technique for the meteorological forecasting task.

Keywords: Meterological forecasting, deep learning, neural architecture search, AutoML.

1. Introduction

Playing an essential role in transportation [1, 2], manufacture [3, 4, 5], and human safety [6, 7], multi-modal meteorological forecasting has been attaching growing attention in recent years [8, 9]. Thanks to the development of remote-sensing technologies, a variety of sensors measuring high precision meteorological status continuously provide high-precision observation of multiple atmospheric data. As a consequence, a deluge of data that describes the whole environment along spatial and temporal dimensions is available with increasing transmission rates exceeding petabytes per day [10]. Based on the flood of meteorological data, humans keep making new achievements with various kinds of remote sensors including radar, infrared sensor, multispectral scanner, and their combinations [11, 12, 13, 14].

Notwithstanding the success of the collection of meteorological data, the assimilation and prediction ability in the last few decades has not increased apace with the acquisition ability of meteorological data [15]. Nowadays, weather prediction relies extensively on massive numerical simulation systems, which consist of complex coupled partial differential equations describing the atmosphere in terms of momentum, mass, and enthalpy [16]. To obtain a higher resolution of weather prediction, the dynamical core of weather prediction models has witnessed several re-formulations. For example, to meet the demand of a highly parallelizable algorithm, icosahedral [17] and cubed-sphere [18] grids have been developed to realize finite-difference and finite-volume discretization. Taking conservation laws into consideration, remarkable achievements have also been achieved in designing discretization approaches [19]. Although these methods are brilliant, the complexity of physical models and required expert knowledge make it almost impossible to exploit and adapt to the burst of meteorological data and be applied in real-time. Another promising direction is model ensemble. Consider-

ing the nonlinear complexity of the meteorological system, the ensemble of different complete physical nonlinear models provides better prediction results [9, 20, 21]. Unfortunately, the number of ensemble members is restricted to a relatively small number subjecting to the computational cost [22]. In summary, it remains an issue to make use of the flood of meteorological data effectively and efficiently.

On the other hand, thanks to the advances in statistical modeling, machine learning methods have achieved remarkable results and provide a new perspective on geoscience and remote sensing problems [23, 24]. To explore the potential of machine learning for the meteorological service field, various machine techniques have been applied, including Support Vector Machine (SVM) [25, 26], random forecast [27, 28, 29], cluster [30], and neural network [31, 32]. Specifically, in the urban air temperature estimation task, considering the Land Surface Temperature (LST) is affected by many factors and the correlation is hard to model with numerical simulation system, Yoo. *et. al* [27] adopted a random forest model to model the multi-factor correlation and estimate daily maximum and minimum air temperatures based on different climate characteristics. For the flood monitoring task, Tong. *et. al* [33] proposed a monitoring approach based on optical imagery and radar imagery. To effectively extract the underlying mapping between multi-modal information and flood inundation, an SVM method was applied and demonstrated remarkable effectiveness in terms of noisy reduction, computation efficiency, and effect. To reconstruct the Surface Air Temperature (SAT) with a high spatio-temporal resolution, Zhang. *et. al* [28] proposed to blend the geostationary information captured by satellites. In this work, the random forest model is utilized to model the relationship between relationships between the input variables and large-scale SAT observations. In reality, LST is hard to derive accurate estimation directly due to the complexity of the energy processes and the massive parameters involved. To address this issue, Weng. *et. al* proposed a least square support vector machine method to fuse the information from different sources in a data-driven method. Although all these methods are brilliant, the required expert experience and prior knowledge for extracting handcrafted features prevent them from being more wildly applied in practice.

Recently, benefiting from the development of Graphics Processing Units (GPUs) that allow massive computation, deep learning techniques and Deep Neural Networks (DNNs) have become the engine for Artificial Intelligence (AI) [34]. In the last decade, DNN based methods have refreshed the records

for many traditional challenging tasks including visual perception [35, 36, 37], speech recognition [38, 39], and Go Game [40, 41]. Combined with the deluge of high-precision remote sensing data, deep learning has also demonstrated remarkable performances in many traditional remote sensing tasks, such as the retrieval of atmospheric profiles [42], atmospheric correction [43], remote sensing image classification [44, 45, 46] and flood detection [47], demonstrating the superiority of deep learning for multisource meteorological data [48, 22]. Considering the convolution operation is essentially a spatial operation and incompetent for extracting temporal information, ConvLSTM [49] and its variants [50, 51, 52] incorporate the convolution operation into the LSTM framework, rendering deep learning models with the ability to extract context from both spatial and temporal dimensions. For a representation with global receptive filed, Yao *et. al* [53] applied the attention module to extract the context along both spatial and temporal dimensions. To capture the spatio-temporal dependency, Li *et. al* [54] proposed a spatio-temporal network based on the 3D convolution that extends the traditional convolution network by appending an extra channel along the temporal dimension. Similarly, to explicitly model the correlation of different locations and modalities, Huang *et. al* [55] proposed the context-LSTM module and pattern-fusion attention module to capture the inter-region and cross-category correlations. Although these methods are brilliant, they focus on spatio-temporal dynamics and do not take the essential physical characteristics into consideration. Besides, they all adopt handcraft architectures that could be far from optimal for specific tasks. Designing a suitable neural network also requires considerable amounts of expertise and experience, leading to another tedious work - "network engineering".

To democratize the technique, Neural Architecture Search (NAS), or more broadly, AutoML has been proposed to automatize the design of DNN architecture in a purely data-driven manner. NAS methods could roughly be divided into two categories, the sampling-based NAS method and the gradient-based NAS method [56]. Sampling-based NAS methods sample child architectures from designed search spaces and apply nested optimization based on the performance of sampled architectures. A well-investigated algorithm is evolution that has been applied in generating neural networks [57, 58] and CNNs [59, 60]. Another stream of research utilizes an RNN as the agent to generate architectures and applies reinforcement learning to train the agent [61, 62, 63]. Despite their high interpretability and feasibility, sampling-based methods are computationally expensive as numerous archi-

tectures are required to be validated during searching. To this end, gradient-based NAS methods are subsequently introduced with the aim to improve the searching efficiency. Though relaxing the discrete search space to be continuous, DARTS [64] builds a supernet with architecture parameters and optimizes them with back-propagated gradients. Considering the discrete essence of architecture, several works attempt to reduce the impact brought by discretizing architecture parameters. Some researchers formulate NAS as a pruning process and apply compression methods such as sparse regulation [65] and binarization [66]. Others combine the search process with discrete gradient estimators such as Gumbel-Softmax [67, 68]. Benefiting from such efforts, searching cost is reduced remarkably and significant successes have been achieved in a multitude of fields [69, 70, 71].

In this paper, a deep learning based framework that respects meteorological laws is presented for the meteorological forecasting task. Though exploiting multi-modal meteorological data (humidity, wind, temperature), the proposed method suffices to model the dynamics along spatial and temporal dimensions. Spatially, a convolution based network is proposed to extract the spatial information. Inspired by the recently thriving NAS technique, we propose an architecture searching framework to learn the optimal scheme to fuse multi-modal information in a purely data-driven manner. As for the temporal dimension, we formulate the meteorological forecasting problem as a sequence-to-sequence prediction problem and apply an encoder-decoder framework to capture the temporal context. Moreover, the periodicity of meteorological data is explicitly modeled with a physical indicator, which further improves the long-term forecasting ability of our model. Experiments on two datasets with different resolution ratios demonstrate the effectiveness of the proposed method.

This article is organized as follows. Section 2 introduces our problem formulation. Section 3 presents the study area and data sets. Section 4 recalls some background knowledge relevant to this work. Section 5 describes our methodology, which includes a general overview and details regarding every component. Section 6 introduces the experimental design, illustrates results, and provides some experimental analyses. Finally, this article is concluded in Section 7.

2. Problem Formulation

Due to the resolution and interval of meteorological observation and reanalysis, the meteorological observation could be formulated as a sequence of meteorological data with specific spatial and temporal resolution [72]. For the i-th meteorological modality *i.e.*, temperature, humidity, and wind, the meteorological observation over a spatial region of temporal index t could be represented by a grid matrix $\mathbf{G}_i^t \in \mathbb{R}^{P\times H\times W}$ that consists of H rows and W columns divided according to the latitude-longitude resolution. Every grid contains P variables for the observation of P pressure levels. The whole observation could be represented by the concatenation of all meteorological modalities $\mathcal{M}^t = (\mathbf{G}_1^t, \mathbf{G}_2^t, ..., \mathbf{G}_E^t) \in \mathbb{R}^{E\times P\times H\times W}$, where E is the number of modalities. Benefiting from this modeling, the meteorological forecasting task can be formulated as a sequence prediction problem [73], which aims to generate the sequence of prospective meteorological states $\tilde{\boldsymbol{\mathcal{Y}}} = (\tilde{\mathcal{M}}^{T+1}, \tilde{\mathcal{M}}^{T+2}, ..., \tilde{\mathcal{M}}^{T+T_f}) \in \mathbb{R}^{T_f\times E\times P\times H\times W}$ over the next T_f time steps based on recent historical observed meteorological data $\boldsymbol{\mathcal{X}} = (\mathcal{M}^{T-T_p+1}, \mathcal{M}^{T-T_p+2}, ..., \mathcal{M}^{T}) \in \mathbb{R}^{T_P\times E\times P\times H\times W}$ over previous T_p time slices, where T is the current temporal index. It should be noted that the input sequence and output sequence are of the same spatio-temporal resolutions. Specifically, the length of input sequence T_p and the length of output sequence T_f are set to 6, the time interval is set to 12 hours in this work, which means the proposed model targets to forecast as long as three days' meteorological state based on previous three days' observation.

Empirically, the characteristics of meteorological big data could be summarized as the following four aspects:

Multi-modal relationship The relationship between meteorological modalities is extremely complex. For example, the formation of wind is mainly caused by the difference of temperature, while heat can also be transferred through the convection caused by wind. Generally speaking, different meteorological modalities contribute to the forecasting task in particular methods, the fusion mechanism should be deliberately designed in the prediction framework.

Spatial correlation Spatially, meteorological data at a particular location have an obvious correlation with its neighbors. The absolute correlation commonly decreases with the distance between two points. A prediction framework is expected to explicitly and effectively model this correlation.

Temporal dependency Observed data at a specific point in time is con-

ditioned by earlier meteorological states at the same location. As this dependency may span over a long period, explicit modeling along the temporal dimension is essential for long-term meteorological prediction.

Periodicity Driven by the cyclical solar action, meteorological data essentially follows a regular cycle [74]. Figure 3 clearly shows the meteorological data periodical pattern aligned with the cyclical changes of solar altitude. As an indicator of the state of solar action, solar altitude (the angle between the sun's rays and the horizontal plane) could then be exploited to drive the forecasting task. Driven by the cyclical solar action, meteorological data essentially follows a regular cycle.

To elaborately exploit the dependencies between meteorological data and to facilitate long-term forecasting, these main characteristics are taken into consideration in the proposed forecasting model. Specifically, the spatial correlation is captured by a convolution based architecture with various receptive fields [75, 76]. The multi-modal dependencies are accounted for in a fusion scheme, optimized in a NAS framework. The temporal dynamic is modeled by an encoder-decoder structure. As for periodicity, the solar altitude is integrated into the position encoding of every element in the sequence. All in all, the whole model is differentiable and could be optimized end-to-end with back-propagation gradients.

3. Study Area and Data Sets

3.1. Study Area

As shown in Figure 1, we study an area that covers China and South-East Asia, from 0° to 55° N and 70° E to 140° E in latitude and longitude respectively. In this region, the altitude is high in the West and low in the East, producing a three-step elevation distribution with the highest altitude at 8848.86 meters and the lowest altitude point at -154.31 meters. Due to the complex terrain and huge span of latitude, the area is featured with diverse climatic conditions, including cold plateau climate, temperate continental climate, monsoon climate of medium latitudes, subtropical monsoon climate, and tropical monsoon climate. Typically, a visualization of the meteorological observation of a specific time, *i.e.*, June 1^{st}, 2020 00:00 AM GMT, is illustrated in Figure 2. It could be observed that the variety of climatic conditions and complex geography patterns lead to a huge distinction between different areas, bringing considerable challenges to the meteorological forecasting task.

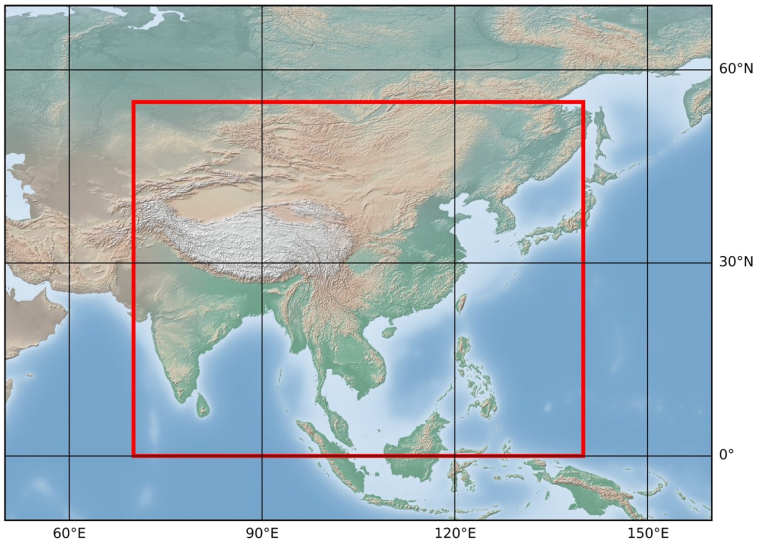


Figure 1: Visualization of the study area, which covers China and South-East Asia.

3.2. DataSets

Based on our formulation of the meteorological forecasting problem, we evaluated the proposed method on two meteorological datasets with hourly gridded meteorological observation.

3.2.1. LAPS Fusion Data

Local Analysis and Prediction System (LAPS) [77] is a local analysis and forecasting system developed by NOAA. It aims at analyzing three-dimensional and high-resolution grid meteorological data from different data sources[1]. We use a meteorological dataset based on LAPS, provided by the China Meteorological Administration (CMA). It covers a period from 2018 to 2021. Data have originally been pre-processed and discretized on a spatial grid of 3 km $\times$ 3 km resolution (around $0.03° \times 0.03°$ for latitude-longitude resolution) and temporal resolution of 1 hour. As for vertical resolution, all modalities are stratified by atmospheric pressure into 37 layers.

3.2.2. ERA5

ERA5 [72] is a new-generation atmospheric reanalysis of the global climate developed by the European Center for Medium-Range Weather Forecasts (ECMWF). The spatial resolution and temporal resolution of the ERA5

[1]more detailed can be found at `http://laps.noaa.gov/`.

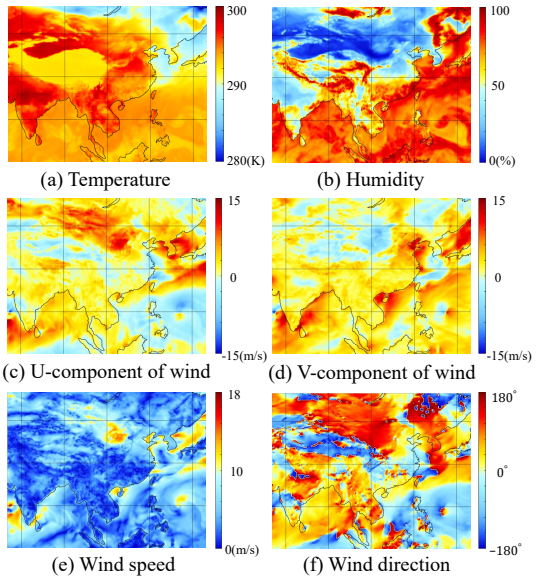


Figure 2: Visualization of the four meteorological modifies, *i.e.*, (a) temperature, (b) relative humidity, (c) U-component of wind, and (d) V-component of wind. For better visualization of wind, wind speed and wind direction are also illustrated in (e) and (f), respectively.

dataset are $0.25° \times 0.25°$ for latitude-longitude resolution and 1 hour respectively[2]. Similar to the LAPS dataset, all meteorological modalities in the ERA5 datasets are stratified into 37 layers according to atmospheric pressure. Although the spatial resolution of the ERA5 dataset is lower than the LAPS dataset, the time span of the ERA5 dataset covers 1979 to 2021, a time range that is much larger compared with the LAPS dataset.

In this work, four classical meteorological modalities are applied, *i.e.*, temperature, relative humidity, U-component of wind, and V-component of wind. As for the vertical dimension, we select three typical atmospheric pressure levels 500, 850, 925 hPa. As a supplement of meteorological data, terrain data from the digital elevation model [78] are also exploited in our experiments to facilitate the prediction by providing geographical information. In experiments, all data are normalized by their corresponding mean and standard deviation.

[2]Available at the Climate Data Store (CDS) `https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=overview`.

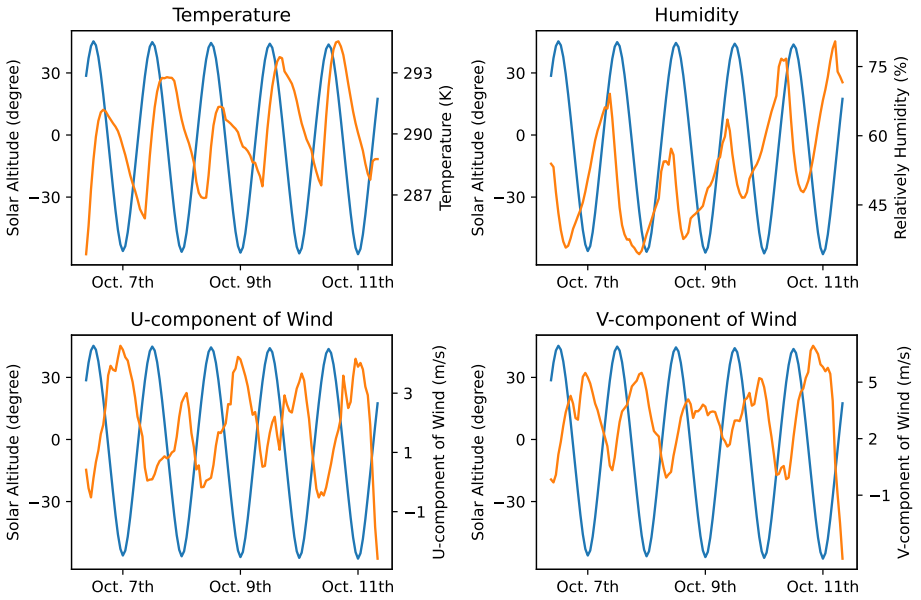


Figure 3: The relationship between solar altitude and the four meteorological modalities, *i.e.*, temperature, humidity, two component of the wind. The solar altitude is shown in blue and the meteorological modalities are shown in yellow. It could be noticed that trends of different meteorological modalities roughly follow the trend of solar altitude.

4. Preliminaries

4.1. Neural Architecture Search

Signifying structure components and connections as nodes and edges respectively, the topology of architecture could be represented as a Directed Acyclic Graph (DAG). Representing the whole search space as a super-graph, the process of NAS can be regraded as obtaining a sub-graph from the super-graph. To make the procedure differentiable, the search process is typically conducted in a continuous space. To achieve this goal, the architecture search parameter α that scales the information flow in the super-graph is introduced [64]. For a specific architecture, it always corresponds to an architecture coding α and could be represented as $\mathcal{N}(\alpha, \mathbf{w})$ with network weights $\mathbf{w}$. Consequently, the NAS optimization is significantly simplified: it boils down to estimating the optimal α^* with minimum validation loss $\mathcal{L}_{val}$, while the corresponding convolutional layers weights $\mathbf{w}$ are optimized from a training loss $\mathcal{L}_{train}$. Formally, the NAS optimization can be formulated as a bilevel

optimization problem [64]:

$$\begin{aligned} &\min_{\alpha \in \mathcal{A}} \ \mathcal{L}_{val}(\mathcal{N}(\alpha, \mathbf{w}^*)), \\ &s.t. \ \mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}_{train}(\mathcal{N}(\alpha, \mathbf{w})). \end{aligned} \tag{1}$$

This nested optimization problem could be computationally complex due to its expensive inner optimization. To solve this problem, a popular solution is to apply the weight sharing strategy [63] and a simple approximation scheme to approximate $\mathbf{w}^*$ with a single training step [64, 65]:

$$\mathbf{w}^* \approx \mathbf{w} - \xi \nabla_{\mathbf{w}} \mathcal{L}_{train}(\mathcal{N}(\alpha, \mathbf{w})). \tag{2}$$

Technically, the training dataset is split into two parts, one is adopted to optimize architecture weights $\mathbf{w}$ and the other performs as the validation set for α. Combined with the one-step approximation, architecture parameters α and architecture weights $\mathbf{w}$ are optimized in an iterative fashion. After optimization, the final discrete architecture is derived by selecting the edge with the highest α for every node [64].

Although brilliant, this modeling fails to bridge the gap between searching and training. As the architecture is essentially sparse, directly discretizing α may destroy the completeness of the original network, leading to the discrepancy of performances between the searching and validating process. To address this problem, DSO-NAS [65] applies the L1-regulation on the architecture parameters to obtain a spare representation of the architecture. Taking L1-regulation for α and L2-regulation for $\mathbf{w}$ into consideration, the objective function can be transformed to:

$$\begin{aligned} &\min_{\alpha \in \mathcal{A}} \ \mathcal{L}_{val}(\mathcal{N}(\alpha, \mathbf{w}^*)) + \gamma \|\alpha\|_1, \\ &s.t. \ w^* = \arg\min_{\mathbf{w}} \mathcal{L}_{train}(\mathcal{N}(\alpha, \mathbf{w})) + \delta \|\mathbf{w}\|_2, \end{aligned} \tag{3}$$

where γ and δ are the weights of the L1 and L2 regulations that prevent overfitting and determine the sparsity of connections, respectively. These hyperparameters could be roughly estimated referring to previous methods [65, 76] and tuned empirically based on the performance on the validation set. After optimization, the architecture parameters are discretized and the final architecture is constructed using the edges with nonzero α^*. Intuitively, an edge whose corresponding architecture parameter α^* is zero can be pruned safely after the search process as it brings no contribution. Once the architecture is learned, the obtained architecture is re-trained on the target task.

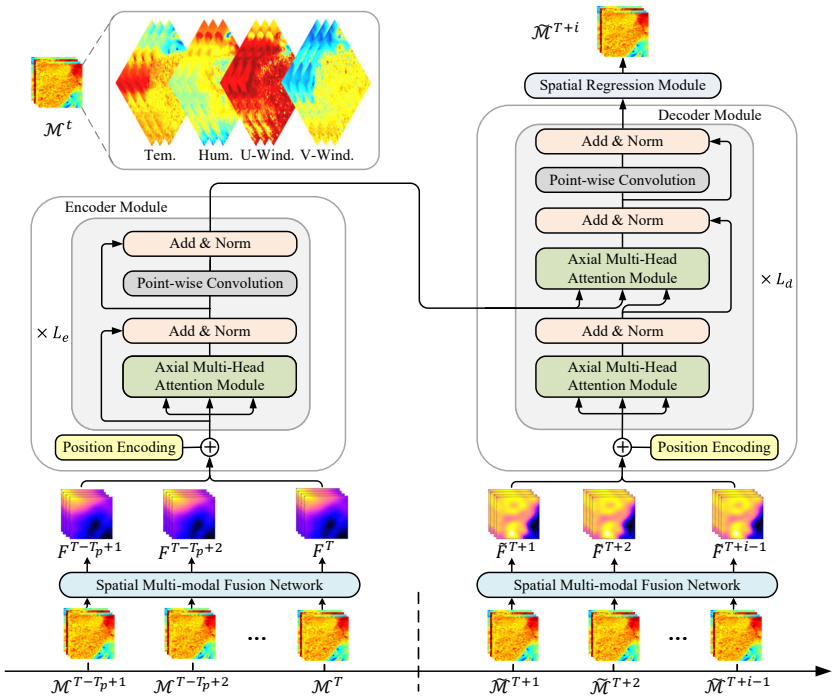


Figure 4: The overall structure of the proposed model. Meteorological data at every time slice is represented by the concatenation of all meteorological modalities from different pressure levels. The meteorological prediction problem is formulated as a sequence-to-sequence prediction problem. Firstly, every element in the sequence is embedded by the spatial multi-modal fusion network. Then, an encoder-decoder framework based on the transformer is applied to capture temporal dynamics and generate the embedding of the target slice. Finally, the spatial regression module upsamples and output the final prediction. In every step, the whole model generates meteorological data of the next slice with the previously generated data as additional input, yielding an auto-regressive manner.

4.2. Attention Mechanism with Transformer

Another fundamental operation in the proposed model is the attention module [79]. The aim of the attention is to weight the contribution of different 'elements/tokens' before combining them and generating an 'output'. This is done by computing akin of correlation factors between these tokens. Specifically, based on a given query, as well as keys and values of these tokens, the output of the attention module is calculated by a weighted sum of all values, where the weight assigned to each value is determined by the correlation between the query and corresponding keys. Formally, a set of queries, keys, and values are packed together into matrices $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ to

compute the attention weights according to the following equation:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \tag{4}$$

where $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{N \times d_k}, \mathbf{V} \in \mathbb{R}^{N \times d_v}$ are all vectors that denote queries, keys, and values, respectively, N represents the number of tokens, d_k and d_v are the feature dimensions of each individual query/key and value. Based on the attention mechanism, multi-head attention [80] is one of the wildest applied attention modules in practice due to its ability to jointly attend to information from different representation subspaces. Technically, the multi-head attention first projects the queries, keys, and values into different representation subspaces and calculates attention value parallelly. Finally, the obtained values are concatenated and further projected linearly. Formally,

$$\begin{aligned} \text{MH}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{Concat}(head_1, head_2, ...head_h)\mathbf{W}^O, \\ \text{where } head_i &= \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \end{aligned} \tag{5}$$

where h is the number of attention heads, $\mathbf{W}_i^Q$, $\mathbf{W}_i^K \in \mathbb{R}^{d_k \times d'_k}$ and $\mathbf{W}_i^V \in \mathbb{R}^{d_v \times d'_v}$ are projection matrices for queries, keys, and values respectively, $\mathbf{W}^O \in \mathbb{R}^{hd'_v \times d_o}$ is the final linearly transform matrix that projects the concatenated representation into final embedding with dimension d_o. Through calculating the weighted sum of all values according to corresponding distances between keys and queries, the multi-head attention module is capable of modeling the dependencies without regard to their distance. As a consequence, the multi-head attention module is wildly applied in the encoder-decoder framework to build the correlation between the input and output sequence for the sequence-to-sequence forecasting task [80, 81].

5. Method

5.1. Overview

The proposed model consists of three parts, the multi-modal fusion network, the encoder-decoder network, and the spatial regression module, as shown in Figure 4. The pipeline works as follows and detailed information could be found in the following subsections. Denoting current time slice as T, to forecast the meteorological data of the i-th time step in the future, *i.e.* $\tilde{\mathcal{M}}^{T+i} \in \mathbb{R}^{E \times P \times H \times W}$, every meteorological raw data $\mathcal{M}^t \in \mathbb{R}^{E \times P \times H \times W}$ in the

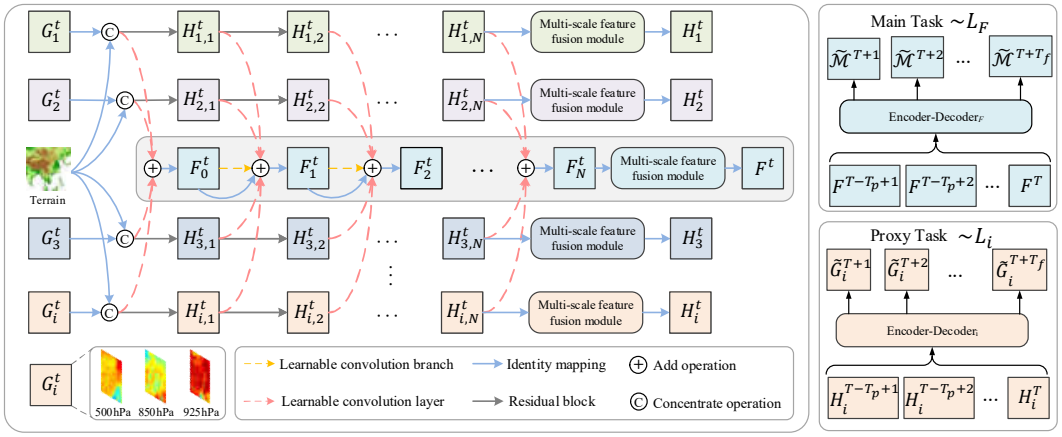


Figure 5: The multi-modal fusion network is based on the residual block and consists of N modality branches for N meteorological modalities and a fusion branch (highlighted with a gray box) to aggregate the information from each layer. The input of every modality branch is the corresponding modality data of different pressure levels. Before being sent to the modality branch, the meteorological data is concatenated with the terrain data along the channel dimension. As for the fusion branch, to decide which modalities to be applied for every layer, a NAS method is adopted to decide the suitable connection (shown as dash arrows) for the fusion branch. At the end of each branch, the multi-scale feature fusion module is applied to fuse the information with different receptive fields. Before the searching process, every modality branch is pre-trained by a proxy task that aims to predict the corresponding modality.

observed sequence $(\mathcal{M}^{T-T_p+1}, \mathcal{M}^{T-T_p+2}, ..., \mathcal{M}^{T})$ and generated data $\tilde{\mathcal{M}}^t \in \mathbb{R}^{E\times P\times H\times W}$ in the previous prediction sequence $(\tilde{\mathcal{M}}^{T+1}, \tilde{\mathcal{M}}^{T+2}, ..., \tilde{\mathcal{M}}^{T+i-1})$ are projected into their respective latent representations by the multi-modal fusion network. Considering the output stride [37, 82] s and output dimension C_f of the multi-modal fusion network, the latent representations of $\mathcal{M}^t$ and $\tilde{\mathcal{M}}^t$ can be denoted as $\mathbf{F}^t, \tilde{\mathbf{F}}^t \in \mathbb{R}^{C_f\times\frac{H}{s}\times\frac{W}{s}}$, receptively. Based on the latent representations, the encoder-decoder structure is adopted to capture the dynamic along the temporal dimension. Specifically, the input observed meteorological sequence $\mathcal{F} = (\mathbf{F}^{T-T_p+1}, \mathbf{F}^{T-T_p+2}, ..., \mathbf{F}^{T})$ is passed into the encoder that is constructed with a stack of L_e identical attention layers, generating the encoded representation $\mathcal{U} = (\mathbf{U}^{T-T_p+1}, \mathbf{U}^{T-T_p+2}, ..., \mathbf{U}^{T}) \in \mathbb{R}^{T_p\times C_e\times\frac{H}{s}\times\frac{W}{s}}$, where C_e is the embedding channel of the transformer module in the encoder network. Along with $\mathcal{U}$, the decoder takes the representations of all previously generated meteorological data $\tilde{\mathcal{F}} = (\tilde{\mathbf{F}}^{T+1}, \tilde{\mathbf{F}}^{T+2}, ..., \tilde{\mathbf{F}}^{T+i-1})$ as inputs to generate the final representation of target data at time step $T + i$,

$\mathbf{Z}^{T+i} \in \mathbb{R}^{C_d \times \frac{H}{s} \times \frac{H}{s}}$, where C_d is the embedding channel of the transformer module in the decoder network. Finally, a spatial regression module transforms the representation $\mathbf{Z}^{T+i}$ into the predicted result $\tilde{\mathcal{M}}^{T+i} \in \mathbb{R}^{E \times P \times H \times W}$.

The training process consists of two stages. Firstly, the NAS allows elaborately exploring the relationship between different meteorological modalities and to search for the architecture of the multi-modal fusion network. Secondly, along with the encoder-decoder network and spatial regression module, the whole network is optimized end-to-end.

5.2. Searching for The Multi-modal Fusion Network

In order to capture the relationship between different modalities, we propose a convolution network that fuses multi-modal meteorological data at each time step independently. As shown in Figure 5, the fusion network consists of (i) E independent modality branches for E input modalities and (ii) a fusion branch to fuse the obtained features. Every modality branch is a ResNet-18 network. For specific time slice t, the i-th modality branch takes the data of the i-th meteorological modality, $\mathbf{G}_i^t \in \mathbb{R}^{P \times H \times W}$, as input. Considering the effect of terrain information, we fuse the meteorological data with terrain information through concatenating them along the channel dimension, yielding a new representation $\mathbf{G'}_i^t \in \mathbb{R}^{(P+1) \times H \times W}$. After that, $\mathbf{G'}_i^t$ is delivered to the modality branch to obtain the embedding of the corresponding modality.

Similarly, the fusion branch is also based on a ResNet-18 network while its target is to combine the features issued from different modalities. Specifically, the output of a specific layer in the fusion branch contains three parts: a skip connection, the convolution projection of the previous layer, and the issued features from all modality branches. Obviously, determining the optimal way of fusing features at different levels is a challenge. Different layers in the DNN network capture varying levels of semantic information [35] and do not necessarily contribute in the same way from one layer to another. Besides, the relationship between different meteorological modalities could be extremely complicated, it requires massive experiments and empirical practice to manually fix the optimal way to fuse the information extracted by different layers.

To effectively exploit the relationship between different meteorological modalities and effectively decide which modalities to be applied in the fusion network for every layer, we propose to apply a NAS technique to search for the optimal fusion scheme depending on the input modalities. Following the

methodology proposed by [65], we apply a group of architecture parameters α to scale the fusion connections and the feature from the previous layer, yielding a flexible search space containing various fusion methods and depths for the fusion branch. Formally, the output of the n-th layer of fusion branch $\mathbf{F}_n^t$ can be defined by the following equation:

$$\mathbf{F}_n^t = \mathbf{F}_{n-1}^t + \alpha_0^n \mathcal{O}(\mathbf{F}_{n-1}^t) + \sum_{c=1}^{E} \alpha_c^n \mathcal{C}(\mathbf{H}_{c,n}^t), \tag{6}$$

where $\mathcal{O}$ signifies the convolution branch of the residual block, $\mathcal{C}$ represents a convolution layer with kernel size 1×1, $\mathbf{H}_{c,n}^t$ is the output of the n-th layer of the c-th modalities branch. Specifically, the original data of the c-th modality is represented by $\mathbf{G}_c^t$. Following [65], we divide the training dataset into two equal parts, one to optimize weights $\mathbf{w}$ and the other to optimize architecture parameters α. Then, the APG-NAG optimizing algorithm [65] is applied to iteratively and alternatively optimizes the weights $\mathbf{w}$ and α on two independent datasets. Specifically, α is optimized under the sparse regulation, as shown in Equation 3. After the searching process, the final architecture is generated by deleting the connections whose α is zero and setting non-zero α to 1.

Prior to applying NAS, we pre-train the whole network, excluding the fusion branch to improve the expressive ability of the multi-modal network, following [65]. To achieve this goal, we define a proxy task: to learn to forecast each modality independently of others. Specifically, the i-th modality branch is equipped with an independent encoder-decoder described in the following section, the whole network is then optimized in a classical way, with auxiliary losses L_i - the error between the predicted of the i-th modality $\tilde{\mathcal{G}} = (\tilde{\mathbf{G}}_i^{T+1}, \tilde{\mathbf{G}}_i^{T+2}, ..., \tilde{\mathbf{G}}_i^{T+T_f})$ and the ground-truth $\mathcal{G} = (\mathbf{G}_i^{T+1}, \mathbf{G}_i^{T+2}, ..., \mathbf{G}_i^{T+T_f})$. This constitutes our pre-training step. We observe experimentally that the pre-training step is essential for the stability of the searching process and contributes to better performance [65].

To account for the large scale spatial dependencies existing in meteorological data, we propose to fuse embeddings at different spatial distances. To this end, a multi-scale feature fusion module is applied behind the fusion branch and every modality branch. Illustrated in Figure 6, the multi-scale feature fusion module consists of a series of atrous convolutions with different atrous rates. As the relevance with different spatial distance become prominent for the atrous convolution layer with a specific receptive field,

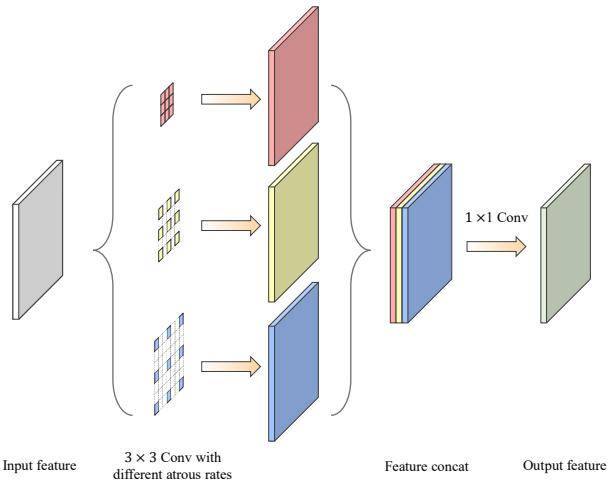


Figure 6: In the multi-scale fusion module, atrous convolution layers with different atrous factors are applied to extract information with different receptive fields, the obtained features are then concatenated along the channel dimension and transformed by an extra 1×1 convolution layer to retain the number of channels.

the multi-scale feature fusion module is in a position to fuse the long-range meteorological features into the local representation.

5.3. Encoder-decoder Structure Based on Transformer

Based on spatial features extracted by the multi-modal fusion network, we further model the temporal dependency between data with attention based encoder-decoder architecture. The model works in an auto-regressive manner, every data in the target sequence $(\tilde{\mathcal{M}}^{T+1}, \tilde{\mathcal{M}}^{T+2}, ..., \tilde{\mathcal{M}}^{T+T_f})$ is generated iteratively. In the i-th forecasting step where $\tilde{\mathcal{M}}^{T+i}$ in generated, the encoder takes the embedding of observed data $\mathcal{F} = (\mathbf{F}^{T-T_p+1}, \mathbf{F}^{T-T_p+2}, ..., \mathbf{F}^{T})$ as input, while the decoder is fed with the embedding of previously generated meteorological data, namely, $\tilde{\mathcal{F}} = (\tilde{\mathbf{F}}^{T+1}, \tilde{\mathbf{F}}^{T+2}, ..., \tilde{\mathbf{F}}^{T+i-1})$. As shown in Figure 4, the encoder's and decoder's structures are composed of two basic elements: (i) the multi-head attention module that extracts temporal dependencies in the sequence and (ii) the point-wise convolution layer that integrates local information at every temporal slice individually. Around each of these two basic operators, a residual connection is attached, followed by a layer normalization [80]. The inputs of the multi-head attention module consist of three parts, query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$. Specifically, for the multi-head attention module in the decoder, $\mathbf{K}$ and $\mathbf{V}$ are features generated

by the encoder module, *i.e.*, $\mathcal{U} = (\mathbf{U}^{T-T_p+1}, \mathbf{U}^{T-T_p+2}, ..., \mathbf{U}^{T})$, while $\mathbf{Q}$ comes from the previous decoder layer. In this way, the information extracted by the encoder module could attend over every position in the decoder. On the contrary, as the purpose of the encoder is to extract the temporal dynamic from the observed input, the query, key, and value vectors in the temporal module all come from the previous encoder layer, *i.e.*, $\mathbf{Q} = \mathbf{K} = \mathbf{V}$, yielding the typical self-attention mechanism.

Without loss of generality, denoting the inputs of the multi-head attention module could be represented as four-dimension tensors $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{l \times c \times h \times w}$, where l, c, h, w represent the temporal length, feature dimension, height, and width respectively. Specifically, as the forecasting module works in an auto-regressive manner, the temporal length l varies according to the specific future time step in the decoder module. In the multi-head attention module, the tensor is firstly permuted to fit a shape of $l \times h \times w \times c$ and then flattened to a two-dimension representation with shape $N \times c$, where $N = l \times h \times w$. Subsequently, the attention function is performed parentally as defined in Equation 5. Finally, the obtained results are reshaped and permuted with inverse operations to the original size $l \times c \times h \times w$. Although effective, the computational cost of the attention mechanism could be extremely high, as the attention weights are calculated on every pair of elements in the target tensor, resulting in $O(N) = O(l \times h \times w)$ multiplications for every grid point. To tackle this issue, instead of applying the traditional attention module to a flattened string of tensor elements, axial attention mechanism [83] where the attention computation is performed along a single axis of the tensor is applied. As shown in Figure 7, in the axial attention mechanism, the attention operation is performed along each of the three dimensions independently, yielding to $O(l + h + w)$ multiplications for every grid. Compared with the computation cost $O(l \times h \times w)$ of the traditional attention model, the axial attention enjoys a significant saving in computation complexity while maintaining a global receptive field.

Despite its capacity to model interdependencies, including along the temporal domain, the attention mechanism is essentially agnostic to the order of data in the sequence since the dependency between inputs and output is built entirely with a weighted sum function. However, accounting for the order of the sequence is essential for any forecasting task, especially for meteorological data which are highly periodic and spatial correlated. In order to equip our model with the ability to properly exploit the order information, every element representation $\mathbf{F}^t$ is augmented with a position encoding,

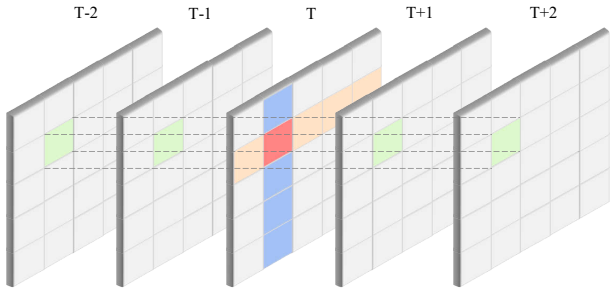


Figure 7: Illustration of the axial attention mechanism. As for a single grid (shown in red) in the three-dimension-tensor, the attention computation is performed along different dimensions and the results are summed up together.

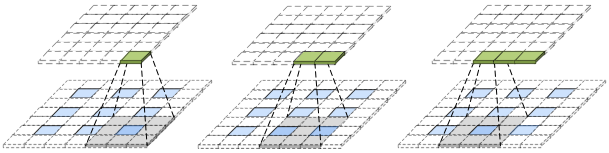


Figure 8: Illustration of the computation of deconvolution operation with input feature size 3×3 and output feature size 5×5.

where t is the temporal slice. This guarantees that different elements that are spatially close or at the same *time-point* in different periods tend to be closer. As changes of meteorological modalities are intrinsically caused by the solar action, a position encoding method based on the solar altitude that could be regarded as a natural indicator for the solar action is applied to every feature. For specific element with location (x, y) in $\mathbf{F}^t$, the position encodings for features with even dimension index $(2i)$ and odd dimension index $(2i + 1)$ are presented as follows:

$$PE(x, y, t, 2i) = sin(SA(x, y, t)/1000^{2i/d_{feat}}), \tag{7}$$

$$PE(x, y, t, 2i + 1) = cos(SA(x, y, t)/1000^{2i/d_{feat}}), \tag{8}$$

where d_{feat} is the dimension of features, $SA(x, y, t)$ represents the solar altitude calculated based on the latitude, the longitude, and actual time which are inferred with x, y, and t. Finally, the resulting position encoding PE is added to the original representation $\mathbf{F}^t$.

5.4. Spatial Regression Module

At the last stage of our framework, a spatial regression module adopted to convert the coarse map generated by the decoder to a dense and fine

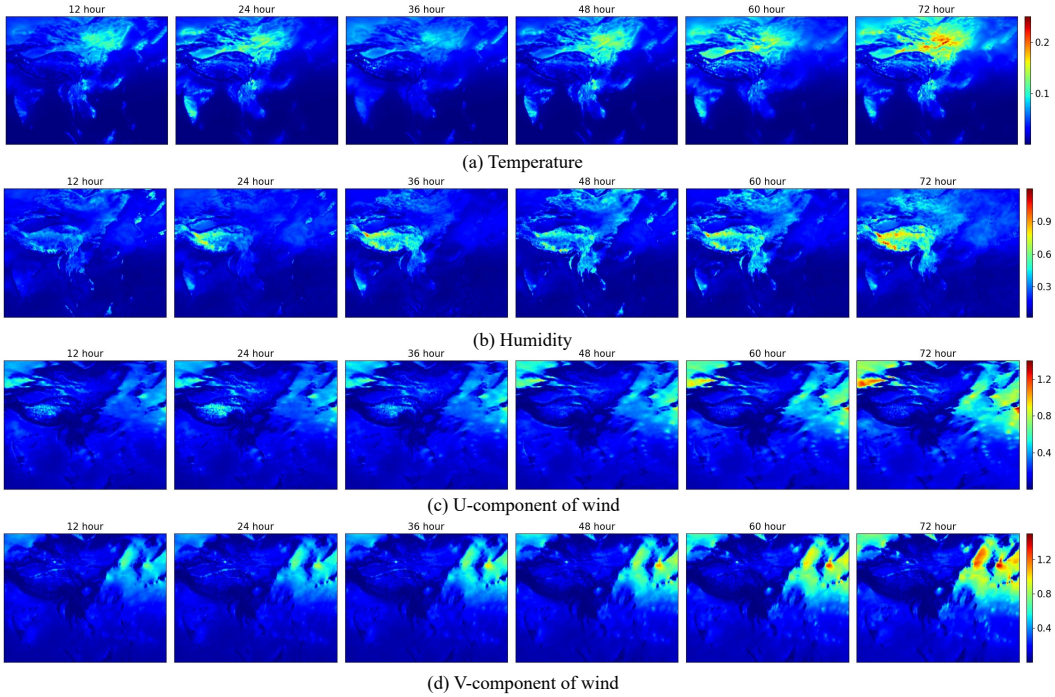


Figure 9: Spatial distribution of the MSE loss of the prediction of different meteorological modalities, *i.e.* (a) Temperature (b) Humidity (c) U-component of wind (d) V-component of wind. The experiments are conducted on the ERA5 dataset.

output. The spatial regression module consists of a series of deconvolution layers [82] and ReLU activation functions. Mathematically, the deconvolution operation is a local computation operation inverse to ordinary convolution operation since it simply reverses the forward and backward passes of convolution. Therefore, the upsampling operation is performed in-network for end-to-end learning by backpropagated gradients from the pixelwise loss. The calculation process of deconvolution is illustrated in Figure 8. With a stack of deconvolution layers and activation functions, the spatial regression decoder is capable of reconstructing the predicted embedding into a larger spatial ratio both effectively and efficiently. To generate the forecast result $\mathcal{M}_t \in \mathbb{R}^{E\times P\times H\times W}$ at moment t, the spatial regression module upsamples the output of decoder module $\mathbf{Z}_t \in \mathbb{R}^{C_z\times \frac{H}{s}\times \frac{H}{s}}$ spatially to $\mathbf{Z}'_t \in \mathbb{R}^{C_o\times H\times W}$, where $C_o = E \times P$. Finally, the forecast output $\tilde{\mathcal{M}}_t$ is obtained through reshaping $\mathbf{Z}'_t$ to the original shape $E \times P \times H \times W$.

After obtaining the predicting result, the loss function could be defined

by the L2 distance between the forecasting sequence $\tilde{\boldsymbol{\mathcal{I}}}$ and ground truth sequence $\boldsymbol{\mathcal{I}}$. Formally:

$$L = \frac{1}{N} \sum_{t,e,p,h,w} (\tilde{\boldsymbol{\mathcal{I}}}_{t,e,p,h,w} - \boldsymbol{\mathcal{I}}_{t,e,p,h,w})^2, \\ where \quad N = T_f \times E \times P \times H \times W, \tag{9}$$

where T_f is the length of forecasting time, E is the number of modalities, P represents the number of atmosphere pressure levels, H and W signify height and width, respectively. $\tilde{\boldsymbol{\mathcal{I}}}, \boldsymbol{\mathcal{I}} \in \mathbb{R}^{T_f \times E \times P \times H \times W}$ are the predicted and ground-truth data respectively. For the proxy task used to initialize the network before applying the architecture search process, the modality dimension is set to $E = 1$ as only one modality is applied in the forecasting task at this stage. Based on the loss function, the whole model could be optimized with backpropagated gradients.

6. Experiments

6.1. Experiment Design

In this work, a series of experiments are conducted to verify the performance of the proposed method, which can be mainly classified into the following categories:

Overall Performance of Our Method Firstly, we evaluate our model's performance to long-term weather forecasting. Data prior to 2020 are used for training, data of 2020 is applied for validating and the resulting model is tested on the data of 2021.

Transferability of Our Method Secondly, we investigate the transferability of our method to different seasons and regions. Specifically, our model is optimized on the data of a specific season or region while evaluated on another one.

Investigation on Major Elements of Our Method Thirdly, experiments are conducted to explore the effectiveness of major components, including the application of NAS method, the multi-scale feature fusion module, and the pre-training strategy in the architecture searching process.

Ablation Study Finally, extensive ablation studies are conducted to systemically and comprehensively analyze the major hyper-parameters of the proposed method, *i.e.*, the number of training epochs, the scale of training data, the effectiveness of solar altitude, terrain information, and the length of the input sequence.

Table 1: Overall performance of the proposed method. For specific forecast time, the mean value of the metric of all meteorological modalities is reported.

Dataset	Method	12 h		24 h		36 h		48 h		60 h		72 h	
		LAPS	EC	LAPS	EC	LAPS	EC	LAPS	EC	LAPS	EC	LAPS	EC
MSE	Pers.	0.50	0.48	0.63	0.60	0.77	0.76	0.86	0.84	0.93	0.91	0.98	0.96
	Clim.	1.12	1.10	1.12	1.10	1.12	1.10	1.12	1.10	1.12	1.10	1.12	1.10
	W-Clim.	0.80	0.77	0.80	0.77	0.80	0.77	0.80	0.77	0.80	0.77	0.80	0.77
	Conv-LSTM	0.25	0.24	0.38	0.38	0.52	0.50	0.61	0.60	0.69	0.66	0.73	0.72
	ST-A3DNet*	0.23	0.20	0.36	0.33	0.49	0.45	0.58	0.56	0.65	0.63	0.71	0.68
	MVSTGN	0.22	0.20	0.36	0.33	0.49	0.45	0.58	0.55	0.64	0.62	0.70	0.67
	PredRNN	0.24	0.22	0.38	0.35	0.50	0.48	0.59	0.58	0.66	0.64	0.72	0.70
	TrajGRU	0.23	0.21	0.36	0.34	0.49	0.47	0.58	0.56	0.65	0.63	0.71	0.69
	MIM	0.23	0.20	0.37	0.33	0.49	0.46	0.59	0.56	0.65	0.62	0.71	0.68
	Ours.	0.14	0.11	0.27	0.25	0.39	0.37	0.49	0.47	0.55	0.53	0.62	0.59
MAE	Pers.	0.51	0.50	0.57	0.56	0.62	0.62	0.65	0.65	0.67	0.66	0.68	0.67
	Clim.	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74
	W-Clim.	0.62	0.60	0.62	0.60	0.62	0.60	0.62	0.60	0.62	0.60	0.62	0.60
	Conv-LSTM	0.36	0.35	0.44	0.44	0.51	0.51	0.56	0.54	0.58	0.58	0.59	0.59
	ST-A3DNet*	0.35	0.32	0.42	0.41	0.50	0.48	0.55	0.53	0.56	0.55	0.58	0.57
	MVSTGN	0.34	0.31	0.42	0.41	0.50	0.48	0.54	0.53	0.58	0.55	0.58	0.56
	PredRNN	0.36	0.34	0.44	0.41	0.50	0.50	0.55	0.54	0.57	0.57	0.59	0.58
	TrajGRU	0.35	0.32	0.43	0.42	0.49	0.48	0.54	0.54	0.57	0.56	0.58	0.57
	MIM	0.35	0.32	0.44	0.41	0.50	0.47	0.55	0.53	0.57	0.56	0.58	0.57
	Ours.	0.27	0.24	0.37	0.36	0.45	0.44	0.49	0.48	0.53	0.52	0.54	0.53
RMSE	Pers.	0.84	0.83	0.89	0.88	0.94	0.93	0.96	0.96	0.98	0.98	0.99	0.99
	Clim.	1.06	1.05	1.06	1.05	1.06	1.05	1.06	1.05	1.06	1.05	1.06	1.05
	W-Clim.	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
	Conv-LSTM	0.50	0.49	0.62	0.62	0.72	0.71	0.78	0.77	0.83	0.81	0.85	0.85
	ST-A3DNet*	0.48	0.45	0.60	0.57	0.70	0.67	0.76	0.75	0.81	0.79	0.84	0.82
	MVSTGN	0.47	0.45	0.60	0.57	0.70	0.67	0.76	0.74	0.80	0.79	0.84	0.82
	PredRNN	0.49	0.47	0.62	0.59	0.71	0.69	0.77	0.76	0.81	0.80	0.85	0.84
	TrajGRU	0.48	0.46	0.60	0.58	0.70	0.69	0.76	0.75	0.81	0.79	0.84	0.83
	MIM	0.48	0.45	0.61	0.57	0.70	0.68	0.77	0.75	0.81	0.79	0.84	0.82
	Ours.	0.37	0.33	0.52	0.50	0.62	0.61	0.70	0.69	0.74	0.73	0.79	0.77

* Our reimplementation.

Considering the target of the meteorological forecasting task is to regress meteorological data in the future, we use standard metrics for quantitative evaluation, such as the Mean Square Error (MSE), the Root Mean Square Error (RMSE), and the Mean Absolute Error (MAE). Given prediction result $\hat{y}$ and ground truth y, these metrics are calculated according to the following equations:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{10}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|. \tag{11}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{12}$$

A lower error value on each of these three metrics represents better performance. To validate the effectiveness of our model, all experiments are conducted on the LPAS and ERA5 datasets introduced in Section 3, error values of all meteorological modalities are reported. In the architecture searching process, the training set is divided into two equal parts: for the optimization of network weights and for the architecture parameters. Every modality branch is firstly equipped with an encoder-decoder structure and pre-trained for 10 epochs with a learning rate 0.01 and weight decay 3×10^{-4}. After that, the network weights and architecture parameters in the multi-modal fusion network are optimized iteratively on two divided datasets for 20 epochs. In this phase, the learning rate is set to 0.01 and weight decay is set to 3×10^{-4}. It should be noted that in the pre-train stage, only the data for optimizing network weights is applied. After the architecture searching process, the connections with $\alpha = 0$ are pruned, all others are set to 1, and the whole architecture is optimized from scratch for $n_{epoch} = 20$ epochs. The weight decay is fixed to 3×10^{-4} while the learning rate is initialized to 0.01 and follows a linear decay scheduler with the minimum value 1×10^{-4}. In our experiments, two forecast baselines are applied for comparison following [84]. a) Persistence forecast where recently observed data are directly applied as forecasts ("tomorrow's weather is today's weather"), b) climatological forecast where two different climatologies are computed from the training data, *i.e.*, a single mean overall times and mean values computed for each of the 52 calendar weeks. Besides, several spatio-temporal forecasting methods like Conv-LSTM [49], ST-A3DNet [54], MVSTGN [53], PredRNN [50], TrajGRU [52] and MIM [85] are compared with the proposed method.

Table 2: Performance of different meteorological modalities. For a specific meteorological modality, the error metrics of the prediction of 72 hours are reported.

Metric	Method	Tem.		Hum.		U-wind.		V-wind.	
		LAPS	EC	LAPS	EC	LAPS	EC	LAPS	EC
MSE	Pers.	0.35	0.32	0.86	0.83	1.06	1.05	1.64	1.63
	Clim.	1.01	0.97	1.02	1.00	1.17	1.11	1.39	1.33
	W-Clim.	0.25	0.21	0.75	0.72	0.94	0.91	1.26	1.25
	Conv-LSTM	0.21	0.20	0.70	0.69	0.87	0.86	1.15	1.13
	ST-A3DNet*	0.19	0.15	0.67	0.64	0.85	0.82	1.12	1.09
	MVSTGN	0.18	0.14	0.68	0.65	0.86	0.83	1.08	1.06
	PredRNN	0.19	0.18	0.70	0.68	0.88	0.84	1.11	1.10
	TrajGRU	0.19	0.18	0.70	0.66	0.85	0.83	1.09	1.07
	MIM	0.18	0.14	0.68	0.66	0.87	0.83	1.11	1.09
	Ours	0.14	0.10	0.62	0.58	0.77	0.76	0.93	0.92
MAE	Pers.	0.43	0.41	0.64	0.63	0.72	0.72	0.87	0.86
	Clim.	0.69	0.67	0.70	0.70	0.75	0.74	0.80	0.79
	W-Clim.	0.36	0.32	0.61	0.61	0.68	0.66	0.78	0.77
	Conv-LSTM	0.33	0.31	0.59	0.59	0.65	0.65	0.75	0.74
	ST-A3DNet*	0.32	0.28	0.58	0.57	0.64	0.63	0.73	0.73
	MVSTGN	0.30	0.27	0.59	0.58	0.64	0.63	0.72	0.72
	PredRNN	0.32	0.30	0.59	0.59	0.65	0.64	0.74	0.74
	TrajGRU	0.31	0.30	0.59	0.57	0.64	0.63	0.73	0.73
	MIM	0.31	0.27	0.58	0.57	0.65	0.65	0.73	0.73
	Ours	0.26	0.23	0.56	0.54	0.61	0.62	0.67	0.67
RMSE	Pers.	0.59	0.57	0.93	0.91	1.03	1.02	1.28	1.28
	Clim.	1.00	0.98	1.01	1.00	1.08	1.05	1.18	1.15
	W-Clim.	0.50	0.46	0.87	0.85	0.97	0.95	1.12	1.12
	Conv-LSTM	0.46	0.45	0.84	0.83	0.93	0.93	1.07	1.06
	ST-A3DNet*	0.44	0.39	0.82	0.80	0.92	0.91	1.06	1.04
	MVSTGN	0.42	0.37	0.82	0.81	0.93	0.91	1.04	1.03
	PredRNN	0.44	0.42	0.84	0.82	0.94	0.92	1.05	1.05
	TrajGRU	0.44	0.42	0.84	0.81	0.92	0.91	1.04	1.03
	MIM	0.42	0.37	0.82	0.81	0.93	0.91	1.05	1.04
	Ours	0.37	0.32	0.79	0.76	0.88	0.87	0.96	0.96

* Our reimplementation.

6.2. Overall Performance of the Proposed Method

First, the overall performance of the proposed method is evaluated. All models are trained on the meteorological data from 2016 to 2019 and validated on the data of 2020. After that, the obtained model is tested on the data of 2021. Performances of different forecasting times and different meteorological modalities are reported in Table 1 and Table 2 respectively. For convenience, the results of persistence forecast, overall climatological forecast, and weekly climatological forecast are simplified as "Pers.", "Clim.", and "W-Clim." respectively. We also report the parameters of all deep learning

based models in Table 3. It could be noted that the proposed method outperforms other methods consistently on the meteorological prediction task with different forecast periods, demonstrating the effectiveness of the proposed method. Compared with other spatio-temporal forecasting methods, the proposed method also achieves better results with comparable number of parameters, owing to the explicitly modeling of the characteristic of meteorological data. Figure 9 shows the spatial distribution of the MSE metric on different meteorological modalities. It could be noted that the difficulty of forecasting a specific modality varies from region to region. Specifically, considering that temperatures change more dramatically than that over the ocean as water has a larger specific heat capacity, it is more difficult to forecast the temperature on the land. On the contrary, forecasting the wind over the ocean is harder as influencing factors are more complex [86].

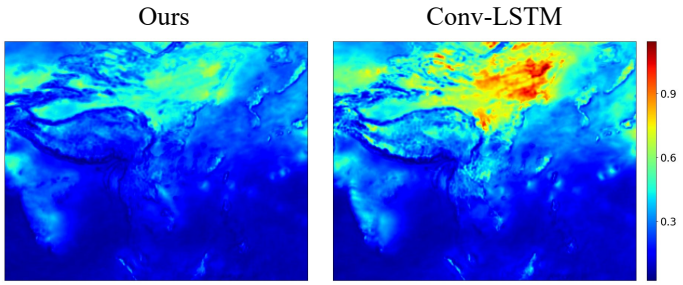
Table 3: The model size of the spatio-temporal forecasting models.

Method	Model Size (MB)
Conv-LSTM	53.47
ST-A3DNet*	52.16
MVSTGN	58.24
PredRNN	64.56
TrajGRU	54.72
MIM	73.55
Ours	62.72

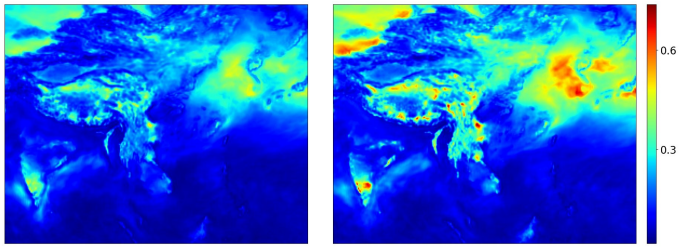
* Our reimplementation.

6.3. Transferability of the Proposed Method

Considering that the difference of meteorological state between different regions or seasons is unnegligible, the transferability of the meteorological prediction model is important in reality. To investigate the transferability of the proposed method, we conducted experiments to explore whether the proposed method is capable of generalizing to different seasons or regions. Empirically, the meteorological difference between the source data (training data) and target data (validation data) should be large enough to guarantee the credibility of experiments. Therefore, summertime (from June to August) and wintertime (from December to February) are selected in the experiments that explore seasonal transferability. As for regional transferability, the North China area (NCN) whose latitude and longitude range from 31° N to 43° N and 109° E to 124° E, and the Central China area (CCN) whose latitude and



(a) Evaluated on the summertime.



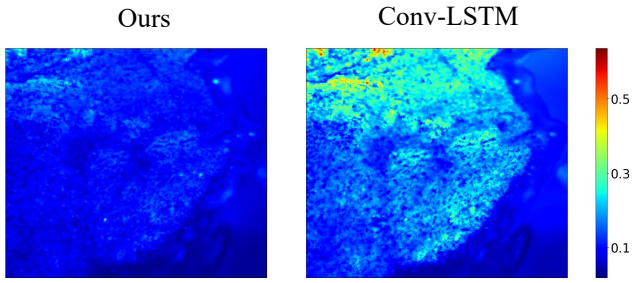
(b) Evaluated on the wintertime.

Figure 10: Spatial distribution of MSE loss of the evaluating models on (a) summertime and (b) wintertime while optimizing on the other season.

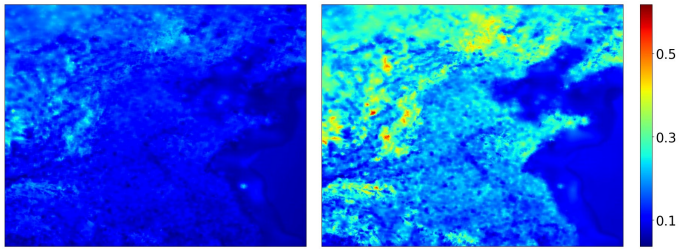
longitude range from 22° N to 36° N and 108° E to 124° E are selected. For simplicity, the spatial distribution of the MSE loss of all combined modalities, computed for the seasonal transfer and the regional transfer, is illustrated in Figure 10 and Figure 11 respectively. It could be observed that the proposed method exhibits better performance than Conv-LSTM when evaluated on independent seasons and regions, indicating the proposed method is in a position to handle the distribution bias and generalize better when faced with severe data distribution gaps between the training dataset and validation dataset.

6.4. Investigation on The Elements of Our Method

To verify the effectiveness of every element of our approach, *i.e.*, the application of the NAS technique, the multi-scale feature fusion module, and the pre-training strategy in the searching process, a series of experiments with and without these elements are conducted. All experiments applied the same hyperparameters directly inherited from the statements in Section 6.1 if not stated otherwise. For every component we firstly illustrate the overall



(a) Evaluated on the CCN region.



(b) Evaluated on the NCN region.

Figure 11: Spatial distribution of MSE loss of the evaluating models on (a) CCN region and (b) NCN region while optimizing on the other region.

performance which is calculated by averaging the performances of all modalities, then we show visualizations of the metrics of every single modality.

6.4.1. The application of NAS technique

To explore the effectiveness of the NAS technique, we compare the proposed method with two baselines: a) random fusion where every learnable connection is kept randomly, b) full fusion where all learnable connections are kept. The results are shown in Figure 12. Note that the fusion network obtained with the NAS technique surpasses other baselines by a significant margin. The obtained architecture with the NAS technique is also informative. Every modality branch is connected with the fusion branch only 4 to 5 times while the two components of wind are always incorporated. Besides, in the first and last layers, most modalities are fused. As for the depth of the fusion branch, all the optional convolution layers are preserved in the fusion branch, owing to the consensus that deeper networks typically have better expressive ability [37].

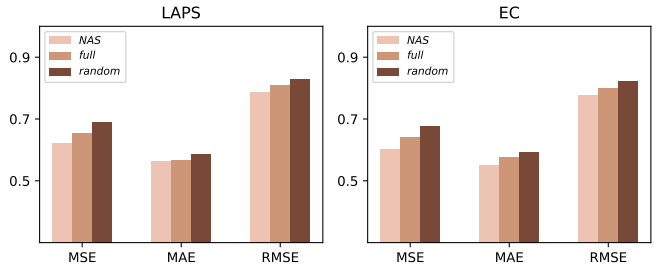


Figure 12: The effectiveness of the NAS technique. The experiments are carried on (a) the LAPS dataset and (b) the EC dataset.

6.4.2. The multi-scale feature fusion module

To investigate the effectiveness of the multi-scale feature fusion module, experiments are conducted to evaluate models with and without the multi-scale feature fusion module. The results are shown in Table 4. It is notable that The multi-scale feature fusion module improves the forecasting results. As the traditional convolution operation is in essence a local operation with a limited receptive field, the multi-scale feature fusion module introduces a learnable method to model the long-distance relevance, which contributes to the long-term forecasting task.

Table 4: The effectiveness of the multi-scale fusion module.

Metric	Method	Tem.		Hum.		U-wind.		V-wind.	
		LAPS	EC	LAPS	EC	LAPS	EC	LAPS	EC
MSE	With	0.14	0.10	0.62	0.58	0.77	0.76	0.93	0.92
	Without	0.16	0.13	0.65	0.62	0.82	0.81	0.98	0.96
MAE	With	0.26	0.23	0.56	0.54	0.61	0.62	0.67	0.67
	Without	0.28	0.25	0.57	0.56	0.63	0.63	0.70	0.69
RMSE	With	0.37	0.32	0.79	0.76	0.88	0.87	0.96	0.96
	Without	0.40	0.36	0.81	0.79	0.91	0.90	0.99	0.98

6.4.3. The effect of the pre-training strategy

As shown in [65], a good initialization is essential for the searching process. To explore the effect of the pre-training strategy, we conduct the fusion network searching process with and without the pre-training step and evaluate the obtained architecture. As the results in Tab. 5 show, equipped with the proxy task that aims to forecast specific modality for every modality

Table 5: The effectiveness of the pre-training strategy in the network searching process.

Metric	Method	Tem.		Hum.		U-wind.		V-wind.	
		LAPS	EC	LAPS	EC	LAPS	EC	LAPS	EC
MSE	Pre-train	0.14	0.10	0.62	0.58	0.77	0.76	0.93	0.92
	Random	0.15	0.12	0.62	0.59	0.81	0.78	1.01	0.99
MAE	Pre-train	0.26	0.23	0.56	0.54	0.61	0.62	0.67	0.67
	Random	0.28	0.26	0.55	0.54	0.64	0.62	0.70	0.70
RMSE	Pre-train	0.37	0.32	0.79	0.76	0.88	0.87	0.96	0.96
	Random	0.39	0.35	0.79	0.77	0.90	0.88	1.00	0.99

branch, the pre-training strategy provides a better initialization, which is favorable to obtain a suitable architecture.

6.5. Ablation Study

As the proposed method is in essence a data-driven method that may be sensitive to the training procedure, extensive ablation studies are conducted to analyze some typical designs of the proposed method. Specifically, all hyper-parameters inherit directly from the statements in Section 6.1, if not stated otherwise. Similar to experiments in Section 6.4, for every element, we compare the overall forecasting performance as well as the forecasting metrics of every modality.

6.5.1. Number of training epochs

In order to analyze the sensitivity with respect to the number of training epochs, we vary the number of training epochs n_{epoch} and visualize the performance of the proposed model in Figure 13 (a). It could be noted that a larger number of training epochs contribute to higher performance, but the effect becomes less apparent as the number of training epochs increases. Inadequate searching epochs may cause under-fitting of architecture distribution while excessive training epochs may lead to over-fitting and damage the generalize ability of the proposed method.

6.5.2. Scale of training dataset

To explore the influence of the scale of training data, the proposed method is optimized on the training dataset with different numbers of years n_{year}, and results are visualized in Figure 13 (b). It could be observed that the performance of the proposed model improves steadily as the scale of the training

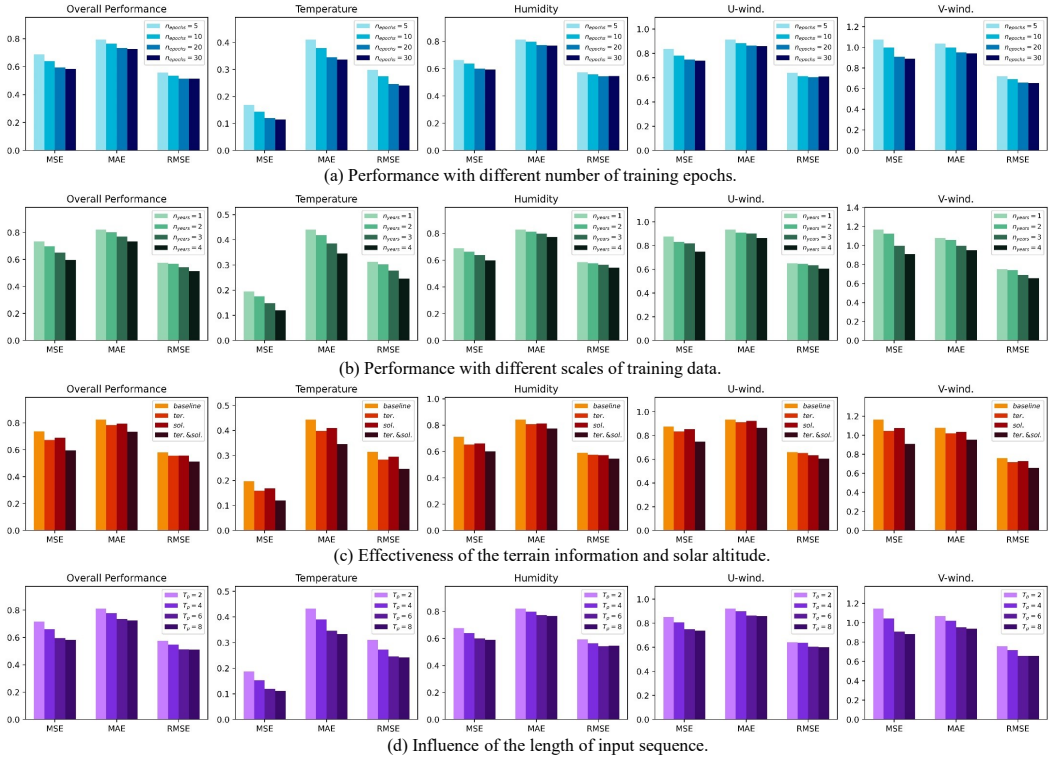


Figure 13: The ablation studies of major hyperparameters in the proposed method, *i.e.*, (a) the number of training epochs, (b) scale of the training dataset, (c) whether terrain and solar altitude are applied, (d) the length of the input sequence. For every experiment, we first visualize the overall performance in the first column and then illustrate three metrics of every meteorological modality. For simplicity, "U-wind" and "V-wind" signify the U-component of the wind and the V-component of the wind receptively.

data increases, indicating that large training data is crucial for improving the generalization ability of the proposed method.

6.5.3. Terrain information and solar altitude

To investigate the effect of terrain information and solar altitude, we carry out experiments with or without these two variables. Results are shown in Figure 13 (c). Obviously, as terrain information and solar altitude explicitly model the geographical condition and the periodicity of meteorological data, they endow the proposed method with higher representative ability and benefit the improvement of performance.

6.5.4. Length of input sequence

To investigate the influence of the length of the input sequence, we vary the number of input time slices T_p and conduct experiments to observe their influences. Specifically, the time interval is changed correspondingly to maintain the overall period. Results shown in Figure 13 (d) indicate that the proposed method achieves better performance as the length of the input sequence increases. It demonstrates that the proposed method is capable of making more accurate predictions based on more detailed information contained in the longer input sequence.

7. Conclusions

To tackle the multi-modal meteorological forecasting task with the oncoming meteorological big data, this article adopts the AutoML technique and proposes a deep learning framework to model the dynamics of meteorological data along spatial and temporal dimensions. In this framework, a deep learning based model is developed to capture the spatial correlations of meteorological data. To model the relationship between different modalities, the NAS technique is applied to search for the optimum fusion network. As for the temporal dimension, we design an encoder-decoder structure for the sequence-to-sequence prediction task to adaptively capture the relation between every sample. Moreover, the periodicity of the meteorological data is elaborately modeled according to physical prior knowledge for better performance. The proposed method is evaluated on the forecasting task of four meteorological modalities, *i.e.*, temperature, humidity, U-component of wind, and V-component of wind. With the China area whose latitude and longitude range from 0° - 55° N, 70° E - 140° E respectively as the study area, experiments on the ERA5 dataset and the LAPS fusion dataset demonstrate the effectiveness of the proposed method. Generally speaking, the proposed method exhibits potential for the application of deep learning methods in meteorological service fields and provides a new perspective for the meteorological forecasting task. Considering its effectiveness and representational ability, the proposed model exhibits the capacity to perform the meteorological forecasting task based on more meteorological modalities on the global scale, which could be left for future work.

8. Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grants 62076242 and 91646207.

References

[1] G. Konapala, S. V. Kumar, S. K. Ahmad, Exploring sentinel-1 and sentinel-2 diversity for flood inundation mapping using deep learning, ISPRS J. Photogramm. Remote Sens. 180 (2021) 163–173.

[2] X. Jiang, S. Liang, X. He, A. D. Ziegler, P. Lin, M. Pan, D. Wang, J. Zou, D. Hao, G. Mao, et al., Rapid and large-scale mapping of flood inundation via integrating spaceborne synthetic aperture radar imagery with unsupervised deep learning, ISPRS J. Photogramm. Remote Sens. 178 (2021) 36–50.

[3] J. A. C. Martinez, L. E. C. La Rosa, R. Q. Feitosa, I. D. Sanches, P. N. Happ, Fully convolutional recurrent networks for multidate crop recognition from multitemporal image sequences, ISPRS J. Photogramm. Remote Sens. 171 (2021) 188–201.

[4] S. Wang, J. Chen, Y. Rao, L. Liu, W. Wang, Q. Dong, Response of winter wheat to spring frost from a remote sensing perspective: Damage estimation and influential factors, ISPRS J. Photogramm. Remote Sens. 168 (2020) 221–235.

[5] D. X. Tran, F. Pla, P. Latorre-Carmona, S. W. Myint, M. Caetano, H. V. Kieu, Characterizing the relationship between land use land cover change and land surface temperature, ISPRS J. Photogramm. Remote Sens. 124 (2017) 119–132.

[6] I. K. Lee, A. Shamsoddini, X. Li, J. C. Trinder, Z. Li, Extracting hurricane eye morphology from spaceborne sar images using morphological analysis, ISPRS J. Photogramm. Remote Sens. 117 (2016) 115–125.

[7] A. Boluwade, Remote sensed-based rainfall estimations over the east and west africa regions for disaster risk management, ISPRS J. Photogramm. Remote Sens. 167 (2020) 305–320.

[8] X. Zhang, J. Zhou, S. Liang, L. Chai, D. Wang, J. Liu, Estimation of 1-km all-weather remotely sensed land surface temperature based on reconstructed spatial-seamless satellite passive microwave brightness temperature and thermal infrared data, ISPRS J. Photogramm. Remote Sens. 167 (2020) 321–344.

[9] F.-G. Ulmer, N. Adam, A synergy method to improve ensemble weather predictions and differential sar interferograms, ISPRS J. Photogramm. Remote Sens. 109 (2015) 98–107.

[10] A. Agapiou, Remote sensing heritage in a petabyte-scale: satellite data and heritage earth engine© applications, Int. J. Digit. Earth 10 (2017) 85–102.

[11] K. Zakšek, M. Schroedter-Homscheidt, Parameterization of air temperature in high temporal and spatial resolution from a combination of the seviri and modis instruments, ISPRS J. Photogramm. Remote Sens. 64 (2009) 414–421.

[12] F. Chen, S. Yang, Z. Su, B. He, A new single-channel method for estimating land surface temperature based on the image inherent information: The hj-1b case, ISPRS J. Photogramm. Remote Sens. 101 (2015) 80–88.

[13] C. Maffei, R. Lindenbergh, M. Menenti, Combining multi-spectral and thermal remote sensing to predict forest fire characteristics, ISPRS J. Photogramm. Remote Sens. 181 (2021) 400–412.

[14] R. Atlas, A. Y. Hou, O. Reale, Application of seawinds scatterometer and tmi-ssm/i rain rates to hurricane analysis and forecasting, ISPRS J. Photogramm. Remote Sens. 59 (2005) 233–243.

[15] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, Prabhat, Deep learning and process understanding for data-driven earth system science, Nat. 566 (2019) 195–204.

[16] P. Bauer, A. Thorpe, G. Brunet, The quiet revolution of numerical weather prediction, Nat. 525 (2015) 47–55.

[17] H. Tomita, M. Satoh, A new dynamical framework of nonhydrostatic global model using the icosahedral grid, Fluid. Dyn. Res. 34 (2004) 357.

[18] P. A. Ullrich, C. Jablonowski, Mcore: A non-hydrostatic atmospheric dynamical core utilizing high-order finite-volume methods, J. Comput. Phys. 231 (2012) 5078–5108.

[19] T. D. Ringler, J. Thuburn, J. B. Klemp, W. C. Skamarock, A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured c-grids, J. Comput. Phys. 229 (2010) 3065–3090.

[20] E. S. Epstein, Stochastic dynamic prediction, Tellus 21 (1969) 739–759.

[21] C. E. Leith, Theoretical skill of monte carlo forecasts, Mon. Weath. Rev. 102 (1974) 409–418.

[22] M. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. Leufen, A. Mozaffari, S. Stadtler, Can deep learning beat numerical weather prediction?, Philo. T. R. Soc. A 379 (2021) 20200097.

[23] S. Sunder, R. Ramsankaran, B. Ramakrishnan, Machine learning techniques for regional scale estimation of high-resolution cloud-free daily sea surface temperatures from modis data, ISPRS J. Photogramm. Remote Sens. 166 (2020) 228–240.

[24] M. Bayad, H. W. Chau, S. Trolove, K. Müller, L. Condron, J. Moir, L. Yi, Time series of remote sensing and water deficit to predict the occurrence of soil water repellency in new zealand pastures, ISPRS J. Photogramm. Remote Sens. 169 (2020) 292–300.

[25] J. Lai, W. Zhan, J. Quan, B. Bechtel, K. Wang, J. Zhou, F. Huang, T. Chakraborty, Z. Liu, X. Lee, Statistical estimation of next-day nighttime surface urban heat islands, ISPRS J. Photogramm. Remote Sens. 176 (2021) 182–195.

[26] Q. Weng, P. Fu, Modeling diurnal land temperature cycles over los angeles using downscaled goes imagery, ISPRS J. Photogramm. Remote Sens. 97 (2014) 78–88.

[27] C. Yoo, J. Im, S. Park, L. J. Quackenbush, Estimation of daily maximum and minimum air temperatures in urban landscapes using modis time series satellite data, ISPRS J. Photogramm. Remote Sens. 137 (2018) 149–162.

[28] Z. Zhang, Q. Du, Hourly mapping of surface air temperature by blending geostationary datasets from the two-satellite system of goes-r series, ISPRS J. Photogramm. Remote Sens. 183 (2022) 111–128.

[29] Y. Zhang, S. Wistar, J. Li, M. A. Steinberg, J. Z. Wang, Severe thunderstorm detection by visual learning using satellite images, IEEE Trans. Geosci. Remote. Sens. 55 (2017) 1039–1052.

[30] B. P. Shukla, C. M. Kishtawal, P. K. Pal, Prediction of satellite image sequence for weather nowcasting using cluster-based spatiotemporal regression, IEEE Trans. Geosci. Remote. Sens. 52 (2014) 4155–4160.

[31] R. Kovordányi, C. Roy, Cyclone track forecasting based on satellite images using artificial neural networks, ISPRS J. Photogramm. Remote Sens. 64 (2009) 513–521.

[32] Q. Vanhellemont, Combined land surface emissivity and temperature estimation from landsat 8 oli and tirs, ISPRS J. Photogramm. Remote Sens. 166 (2020) 390–402.

[33] X. Tong, X. Luo, S. Liu, H. Xie, W. Chao, S. Liu, S. Liu, A. Makhinov, A. Makhinova, Y. Jiang, An approach for flood monitoring by the combined use of landsat 8 optical imagery and cosmo-skymed radar imagery, ISPRS J. Photogramm. Remote Sens. 136 (2018) 144–153.

[34] Y. LeCun, Y. Bengio, G. E. Hinton, Deep learning, Nat. 521 (2015) 436–444.

[35] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), NeurIPS, 2012, pp. 1106–1114.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: CVPR, 2015, pp. 1–9.

[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.

[38] Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Process. Mag. 29 (2012) 82–97.

[39] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6645–6649.

[40] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, Nat. 529 (2016) 484–489.

[41] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., Mastering the game of go without human knowledge, Nat. 550 (2017) 354–359.

[42] D. Malmgren-Hansen, V. Laparra, A. A. Nielsen, G. Camps-Valls, Statistical retrieval of atmospheric profiles with deep convolutional neural networks, ISPRS J. Photogramm. Remote Sens. 158 (2019) 231–240.

[43] J. Sun, F. Xu, G. Cervone, M. Gervais, C. Wauthier, M. Salvador, Automatic atmospheric correction for shortwave hyperspectral remote sensing data using a time-dependent deep neural network, ISPRS J. Photogramm. Remote Sens. 174 (2021) 117–131.

[44] L. Zhang, H. Dong, B. Zou, Efficiently utilizing complex-valued pol-sar image data via a multi-task deep learning framework, ISPRS J. Photogramm. Remote Sens. 157 (2019) 59–72.

[45] T. Liu, A. Abd-Elrahman, Deep convolutional neural network training enrichment using multi-view object-based analysis of unmanned aerial systems imagery for wetlands classification, ISPRS J. Photogramm. Remote Sens. 139 (2018) 154–170.

[46] C. Qiu, L. Mou, M. Schmitt, X. X. Zhu, Local climate zone-based urban land cover classification from multi-seasonal sentinel-2 images with a recurrent residual network, ISPRS J. Photogramm. Remote Sens. 154 (2019) 151–162.

[47] Y. Li, S. Martinis, M. Wieland, Urban flood mapping with an active self-learning convolutional neural network based on terrasar-x intensity and interferometric coherence, ISPRS J. Photogramm. Remote Sens. 152 (2019) 178–191.

[48] Y. Liu, B. Fan, L. Wang, J. Bai, S. Xiang, C. Pan, Semantic labeling in very high resolution images via a self-cascaded convolutional neural network, ISPRS J. Photogramm. Remote Sens. 145 (2018) 78–95.

[49] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, W. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, 2015, pp. 802–810.

[50] Y. Wang, M. Long, J. Wang, Z. Gao, P. S. Yu, Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms, in: NeurIPS, 2017, pp. 879–888.

[51] Y. Wang, Z. Gao, M. Long, J. Wang, P. S. Yu, Predrnn++: Towards A resolution of the deep-in-time dilemma in spatiotemporal predictive learning, in: ICML, volume 80, 2018, pp. 5110–5119.

[52] X. Shi, Z. Gao, L. Lausen, H. Wang, D. Yeung, W. Wong, W. Woo, Deep learning for precipitation nowcasting: A benchmark and A new model, in: NeurIPS, 2017, pp. 5617–5627.

[53] Y. Yao, B. Gu, Z. Su, M. Guizani, Mvstgn: A multi-view spatial-temporal graph network for cellular traffic prediction, IEEE Trans. Mob. Comput. (2021).

[54] H. Li, X. Li, L. Su, D. Jin, J. Huang, D. Huang, Deep spatio-temporal adaptive 3d convolutional neural networks for traffic flow prediction, ACM Trans. Intell. Syst. Technol. 13 (2022) 1–21.

[55] C. Huang, C. Zhang, J. Zhao, X. Wu, D. Yin, N. Chawla, Mist: A multiview and multimodal spatial-temporal learning framework for citywide abnormal event forecasting, in: WWW, 2019, pp. 717–728.

[56] T. Elsken, J. H. Metzen, F. Hutter, Neural architecture search: A survey, J. Mach. Learn. Res. 20 (2019) 55:1–55:21.

[57] P. J. Angeline, G. M. Saunders, J. B. Pollack, An evolutionary algorithm that constructs recurrent neural networks, IEEE Trans. Neural Networks 5 (1994) 54–65.

[58] R. Józefowicz, W. Zaremba, I. Sutskever, An empirical exploration of recurrent network architectures, in: ICML, volume 37, 2015, pp. 2342–2350.

[59] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, A. Kurakin, Large-scale evolution of image classifiers, in: ICML, volume 70, 2017, pp. 2902–2911.

[60] E. Real, A. Aggarwal, Y. Huang, Q. V. Le, Regularized evolution for image classifier architecture search, in: AAAI, 2019, pp. 4780–4789.

[61] B. Zoph, Q. V. Le, Neural architecture search with reinforcement learning, in: ICLR, 2017.

[62] B. Zoph, V. Vasudevan, J. Shlens, Q. V. Le, Learning transferable architectures for scalable image recognition, in: CVPR, 2018, pp. 8697–8710.

[63] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, J. Dean, Efficient neural architecture search via parameter sharing, in: J. G. Dy, A. Krause (Eds.), ICML, 2018, pp. 4092–4101.

[64] H. Liu, K. Simonyan, Y. Yang, DARTS: differentiable architecture search, in: ICLR, 2019.

[65] X. Zhang, Z. Huang, N. Wang, S. Xiang, C. Pan, You only search once: Single shot neural architecture search via direct sparse optimization, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 2891–2904.

[66] H. Cai, L. Zhu, S. Han, Proxylessnas: Direct neural architecture search on target task and hardware, in: ICLR, 2019.

[67] S. Xie, H. Zheng, C. Liu, L. Lin, SNAS: stochastic neural architecture search, in: ICLR, 2019.

[68] X. Zhang, J. Chang, Y. Guo, G. Meng, S. Xiang, Z. Lin, C. Pan, DATA: differentiable architecture approximation with distribution guided sampling, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 2905–2920.

[69] C. Liu, L. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, L. Fei-Fei, Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation, in: CVPR, 2019, pp. 82–92.

[70] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, J. Sun, Detnas: Backbone search for object detection, in: NeurIPS, 2019, pp. 6638–6648.

[71] G. Ghiasi, T. Lin, Q. V. Le, NAS-FPN: learning scalable feature pyramid architecture for object detection, in: CVPR, 2019, pp. 7036–7045.

[72] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, et al., The era5 global reanalysis, Q. J. R. Meteorol. Soc. 146 (2020) 1999–2049.

[73] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), NeurIPS, 2014, pp. 3104–3112.

[74] A. Moreira, D. C. Fontana, T. M. Kuplich, Wavelet approach applied to evi/modis time series and meteorological data, ISPRS J. Photogramm. Remote Sens. 147 (2019) 335–344.

[75] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 834–848.

[76] L. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, CoRR abs/1706.05587 (2017).

[77] S. C. Albers, J. A. McGinley, D. L. Birkenheuer, J. R. Smart, The local analysis and prediction system (laps): Analyses of clouds, precipitation, and temperature, Weather Forecast. 11 (1996) 273–287.

[78] H. I. Reuter, A. Nelson, A. Jarvis, An evaluation of void-filling interpolation methods for SRTM data, Int. J. Geogr. Inf. Sci. 21 (2007) 983–1008.

[79] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Y. Bengio, Y. LeCun (Eds.), ICLR, 2015.

[80] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NeurIPS, 2017, pp. 5998–6008.

[81] S. Guo, Y. Lin, H. Wan, X. Li, G. Cong, Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting, IEEE Trans. Knowl. Data Eng. (2021) 1–1.

[82] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 640–651.

[83] J. Ho, N. Kalchbrenner, D. Weissenborn, T. Salimans, Axial attention in multidimensional transformers, CoRR abs/1912.12180 (2019).

[84] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, N. Thuerey, Weatherbench: A benchmark data set for data-driven weather forecasting, J. Adv. Model. Earth Syst. 12 (2020) e2020MS002203.

[85] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, P. S. Yu, Memory in memory: A predictive neural network for learning higher-order nonstationarity from spatiotemporal dynamics, in: CVPR, 2019, pp. 9154–9162.

[86] M. Denbina, M. J. Collins, Wind speed estimation using c-band compact polarimetric sar for wide swath imaging modes, ISPRS J. Photogramm. Remote Sens. 113 (2016) 75–85.