TemDeep: A Self-Supervised Framework for Temporal Downscaling of Atmospheric Fields at Arbitrary Time Resolutions

Liwen Wang[1], Qian Li[12], Qi Lv[12], Xuan Peng[12] and Wei You[1]

[1]College of Meteorology and Oceanography, National University of Defense Technology, Changsha, China.

[2]High Impact Weather Key Laboratory of CMA, Changsha, China.

Correspondence to: Qian Li (public_liqian@163.com)

Abstract

Numerical forecast products with high temporal resolution are crucial tools in atmospheric studies, allowing for accurate identification of rapid transitions and subtle changes that may be missed by lower-resolution data. However, the acquisition of high-resolution data is limited due to excessive computational demands and substantial storage needs in numerical models. Current deep learning methods for statistical downscaling still require massive ground truth with high temporal resolution for model training. In this paper, we present a self-supervised framework for downscaling atmospheric variables at arbitrary time resolutions by imposing a temporal coherence constraint. Firstly, we construct an encoder-decoder structured temporal downscaling network, and then pretrain this downscaling network on a subset of data that exhibit rapid transitions and is filtered out based on a composite index. Subsequently, this pretrained network is utilized to downscale the fields from adjacent time periods and generate the field at the middle time point. By leveraging the temporal coherence inherent in meteorological variables, the network is further trained based on the difference between the generated field and the actual middle field. To track the evolving trends in meteorological system movements, a flow estimation module is designed to assist with generating interpolated fields. Results show that our method can accurately recover evolution details superior to other methods, reaching 53.7% in the restoration rate on the test set. In addition, to avoid generating abnormal values and guide the model out of local optima, two regularization terms are integrated into the loss function to enforce spatial and temporal continuity, which further improves the performance by 7.6%.

1 Introduction

In the field of meteorology, temporal downscaling refers to the enrichment of time-series data by filling in the time gaps in observations or numerical products, which can provide a more continuous and comprehensive understanding of geophysical phenomena. Temporal downscaling in atmospheric fields holds considerable importance, given its extensive applications across a wide range of domains. In climate research, accurate temporal interpolation plays a vital role in understanding long-term climate variations and assessing the impacts of climate change (Hawkins & Sutton, 2011; Michel et al., 2021; Papalexiou et al., 2018). By enriching historical climate records with temporally enhanced data, researchers gain a more detailed depiction of past climatic events (Barboza et al., 2022; Neukom et al., 2019). For example, the analysis of high-resolution data has revealed the relationship between global temperature rise and the frequency and intensity of extreme weather events, such as heatwaves and heavy rainfall (Kajbaf et al., 2022; Seneviratne et al., 2012). In the field of weather forecasting, accurate temporal downscaling significantly enhances the quality of short-term weather predictions (McGovern et al., 2017; Requena et al., 2021). Filling gaps between discrete atmospheric observations allows for accurate tracking and prediction of various meteorological phenomena (Dong et al., 2013). For instance, the ability to capture rapid changes in wind patterns using high-resolution temporal data enables more accurate forecasting of severe storms, hurricanes, and their paths. This information is critical for issuing timely warnings, facilitating evacuations, and minimizing the potential damage caused by such weather events (Raymond et al., 2017). Furthermore, high-resolution time-series data aid in optimizing agricultural practices, optimizing energy production from renewable sources, and improving transportation planning by considering detailed weather patterns (Lawrimore et al., 2011; Lobell & Asseng, 2017).

Current methods for temporally downscaling atmospheric fields mainly fall into two categories: dynamical downscaling and statistical downscaling. Starting from a specific initial condition, dynamical downscaling methods can interpolate or extrapolate atmospheric fields to a finer time scale by integrating governing equations over time. Early pioneering work by Lorenz (1963) established the basic framework of using governing equations of fluid

dynamics and thermodynamics to predict future atmospheric states. Since then, models such as the Weather Research and Forecasting (WRF) model (Skamarock et al., 2008) and the Community Earth System Model (CESM) (Hurrell et al., 2013) have been developed, incorporating advanced physical parameterizations and data assimilation techniques. These models have been widely used in producing high-temporal-resolution datasets, such as the European Centre for Medium-Range Weather Forecasts' Integrated Forecast System (ECMWF IFS) updates (Bauer et al., 2015) and the High-Resolution Rapid Refresh (HRRR) forecasts (Benjamin et al., 2016). However, the computational expense of these models is a significant barrier, especially for high-resolution, long-term, or global-scale studies (Maraun, 2010). In addition, these models require highly accurate initial conditions. Studies by Lorenz (1969) and Palmer et al. (2005) demonstrate how uncertainties in initial conditions and model parameters can lead to significant prediction errors over time, referred to as the 'butterfly effect'.

The limitations of dynamical downscaling methods have prompted research into statistical alternatives, as they are computationally less expensive and can be easily applied across different spatial and temporal scales (Fowler et al., 2007). These methods, often employing regression techniques or machine learning algorithms, aim to identify and exploit statistical relationships between low-resolution and high-resolution data, such as weather generators (Gutmann et al., 2011; Lee et al., 2012), naïve (Chen et al., 2011; Jia-hong, 2006) and autocorrelation (Mendes & Marengo, 2010). However, as discussed by Maraun (2010), these methods often assume linear or local relationships in consecutive fields and may oversimplify complex atmospheric dynamics.

In recent years, deep learning has been widely applied to meteorology for its potentials to extract complex patterns from large amounts of data (Reichstein et al., 2019). For example, Kajbaf et al. (2022) conducted temporal downscaling with artificial neural networks on precipitation time-series with a 3-hour time step. However, deep learning applications in meteorology so far have generally relied on supervised learning, requiring large amounts of high-resolution ground truth data for training, which could be difficult to acquire due to limited observation intervals, excessive computational demands and high cost of data restoration (Bolton and Zanna, 2019).

In summary, although advancements have been made in temporal downscaling, there still exist significant demands for methods that can provide high temporal resolution with better physical consistency, improved computational efficiency and most importantly, less reliance on high-resolution ground truth data. This motivates our study, which aims to explore self-supervised learning as a potential solution to these challenges. As a form of unsupervised learning, self-supervised learning is a machine learning method that does not rely on supervision but leverages supervisory signals from the structure or properties inherent in data to train deep neural networks (Liu et al., 2020). This approach can leverage vast amounts of unlabeled data for training, thereby significantly enhancing the model's generalization capabilities. It has been applied in diverse fields, including meteorology science (Eldele et al., 2022; Pang et al., 2022; Wang et al., 2022).

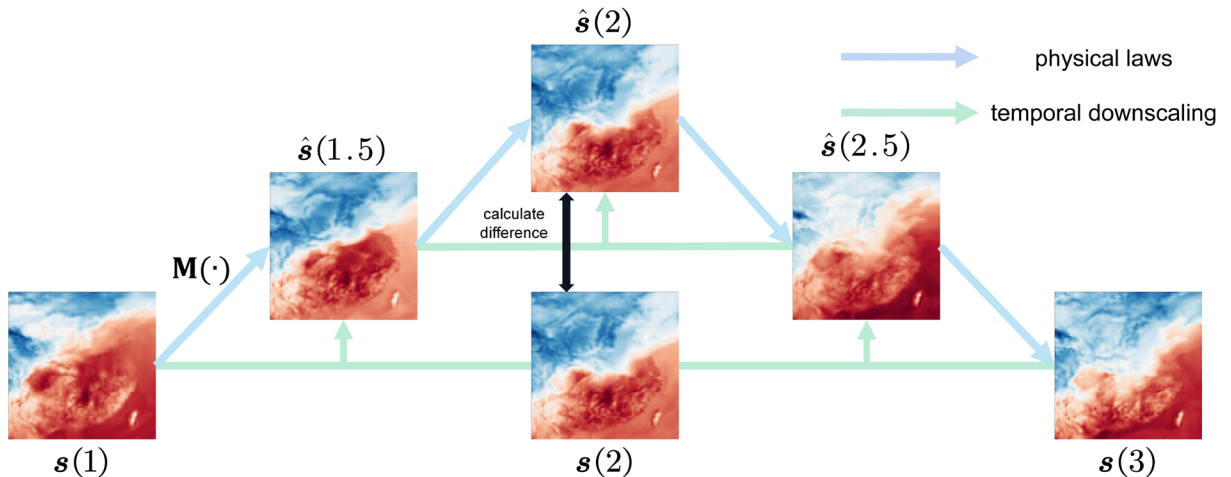



Figure 1. Illustration of time-evolving atmospheric fields. The green arrows represent the evolutions of an atmospheric field guided by physical laws with temporal coherence. The blue arrows represent the outputs of a temporal downscaling model, which seeks to approximate the physics-guided evolutions.

83   In fact, although atmospheric variables do not change linearly at different time steps, it is commonly believed
84 that their evolutions are consistently guided by the same physical laws and thus exhibit temporal coherence over time
85 (Lorenz, 1969). In other words, for the state $s(t)$ of any atmospheric variable at any time t, it will transition from
86 $s(t-1)$ to $s(t)$ following the mapping $\mathbf{P}$ guided by a set of physical laws, expressed as $\mathbf{P}: s(t-1) \rightarrow s(t)$.
87 Based on this invariant mapping constraint, time-series data themselves can be used as supervision information to
88 train the deep learning model. To be specific, at any moment t, $s(t)$ can be taken as the truth value to train the
89 mapping relationship from $s(t-1)$ to $s(t)$. As shown in the example in Fig. 1, for three consecutive fields $s(1)$,
90 $s(2)$ and $s(3)$ with an interval of 1 hour, if the goal is to train a downscaling model $\mathbf{M}$ to fill the gaps at 1.5h and
91 2.5h and obtain $\hat{s}(1.5) = \mathbf{M}(s(1), s(2))$, $\hat{s}(2.5) = \mathbf{M}(s(2), s(3))$, after generating
92 $\hat{s}(2) = \mathbf{M}(\hat{s}(1.5), \hat{s}(2.5))$, the existing $s(2)$ can serve as supervision and the errors between $\hat{s}(2)$ and
93 $s(2)$ be utilized as loss to train $\mathbf{M}$. Therefore, it is clear that continuous atmospheric variables inherently contain
94 sufficient information, which can be utilized as supervision for self-supervised temporal downscaling.

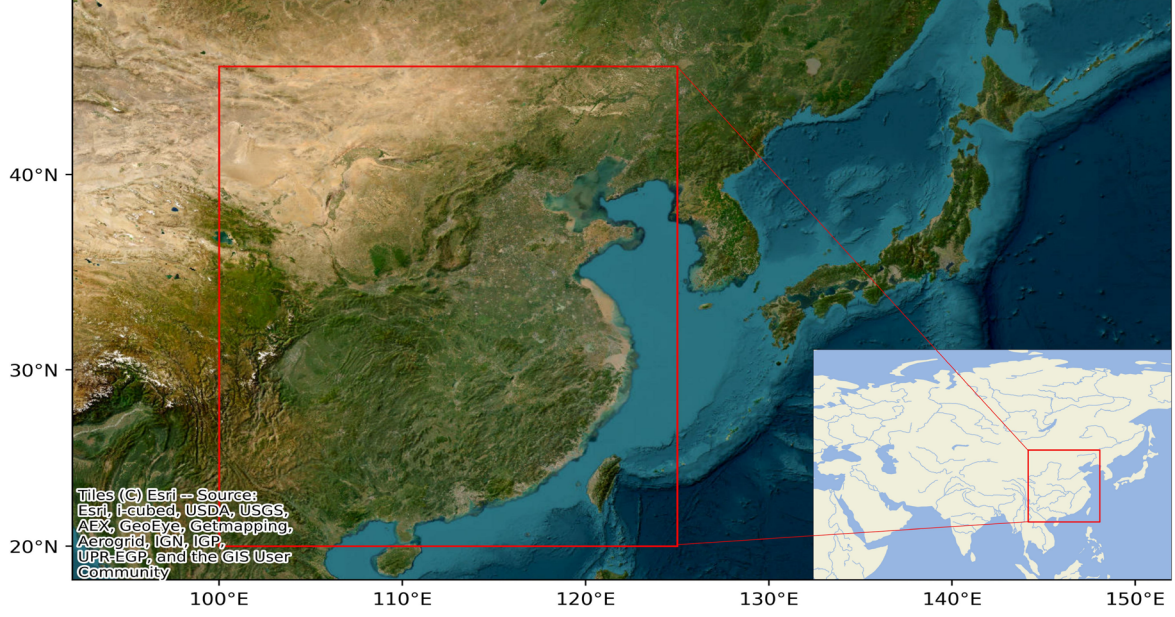
95   In light of this consideration, we present TemDeep, the first self-supervised framework for downscaling
96 atmospheric fields at arbitrary Temporal resolutions based on Deep learning. This framework addresses this issue by
97 imposing a temporal coherence constraint across time-series fields, which means multiple consecutive fields
98 themselves are leveraged as supervision information to train the model. Firstly, we construct an encoder-decoder
99 structured temporal downscaling network, which is capable of performing interpolation at any resolution (see Section
100 2.6), and pretrain this downscaling network by designing a composite index to filter out a subset of data with rapid
101 changes (see Section 2.3). The pretraining stage allows the model to initially capture general patterns and features
102 present in the atmospheric data. In the next step, we utilize this pretrained model to downscale the fields from adjacent
103 time periods and subsequently infer the field at the middle time point (see Section 2.4). Then, the model is further
104 trained based on the difference between the inferred field and the actual middle field, according to the temporal
105 coherence inherent in atmospheric variables. To effectively track the evolving trends in meteorological system
106 movements, the network adopts a flow estimation module to assist with synthesizing fields. We have also designed a
107 module to process terrain data, which enables the model to better perceive the prior information of the underlying
108 surface. In experiments, our method demonstrates effectiveness in accurate downscaling various atmospheric variables
109 at different temporal resolutions, reaching over 53.7% in the restoration rate, superior to other existing unsupervised
110 methods.

111   The structure of this paper is as follows: Section 2 presents the details of the study area and data sources used
112 in our study. Further, we explain our methodology, specifically detailing the entire training process and network
113 architecture. In Section 3, we conduct extensive experiments to assess the model's effectiveness. Finally, Section 5
114 summarizes the methods and contributions made in this study and points out possible future works and applications.

115   2 Data and Methods

116   2.1 Study area and dataset

117   Our study focuses on the geographic area bounded by longitude 100°E to 125°E and latitude 20°N to 45°N
118 with a spatial resolution of 0.25°×0.25° (see Fig. 2), and data for this region were downloaded from the European
119 Centre for Medium-Range Weather Forecasts (ECMWF) ERA5 reanalysis dataset. The dataset comprising 87,660
120 two-hour-interval samples from 2001 to 2020 is used as the training dataset. The testing dataset consists of 8,760 one-
121 hour-interval samples in 2021. To evaluate the generalization performance of the TemDeep method, experiments were
122 conducted on three atmospheric variables: 2-meter air temperature (*t2m*), 850*hPa* geopotential height (*z*) and 850*hPa*
123 relative humidity (*rh*). Horizontal and vertical wind components are utilized to calculate wind speed as part of a
124 composite index (see Section 2.3). Recognizing the influence of topography on local climate and weather patterns, we
125 have also included terrain data with a resolution of 15*km*, sourced from NASA's Shuttle Radar Topography Mission
126 (Hennig et al., 2001). This resolution is sufficient to represent major terrain influences on atmospheric processes
127 without significantly increasing computational demands.

128

129 **Figure 2. Satellite image of the study area.** The study area is outlined by the red rectangle (base map imagery
130 provided by Esri WorldImagery).

131 2.2 Problem definition and overview

132   Given the initial atmospheric fields $\{\boldsymbol{p}\}_T\left(\boldsymbol{p}\in\mathbb{R}^{n_x\times n_y}\right)$ represented as a continuous gridded dataset with a

133 temporal resolution of τ, our goal is to achieve temporal downscaling at any resolution $\theta\tau\left(\theta\in(0,1),\mathbf{R}^+\right)$. Here,

134 n_x and n_y denote the number of grid points in the horizontal and vertical directions, respectively. That is, for a given

135 period of weather process occurring between the interval $\left[T_0,T_0+\tau\right]$, we aim to accurately generate the

136 interpolated field at any time point $T_0+\theta\tau$. To achieve this goal, a self-supervised framework is presented for

137 temporal downscaling (see Fig. 3), in which the training procedure consists of two primary stages. In the first stage,
138 we pretrain our model on a subset of data to simulate the training process on a real high-resolution dataset by selecting
139 scenarios with rapid transitions. Then, the model is further trained under guidance of a temporal coherence constraint,
140 leveraging supervision information inherent in the low-temporal-resolution time-series. In addition, two regularization
141 terms are utilized in the loss function to guide the model out of local optima and prevent abnormal values.

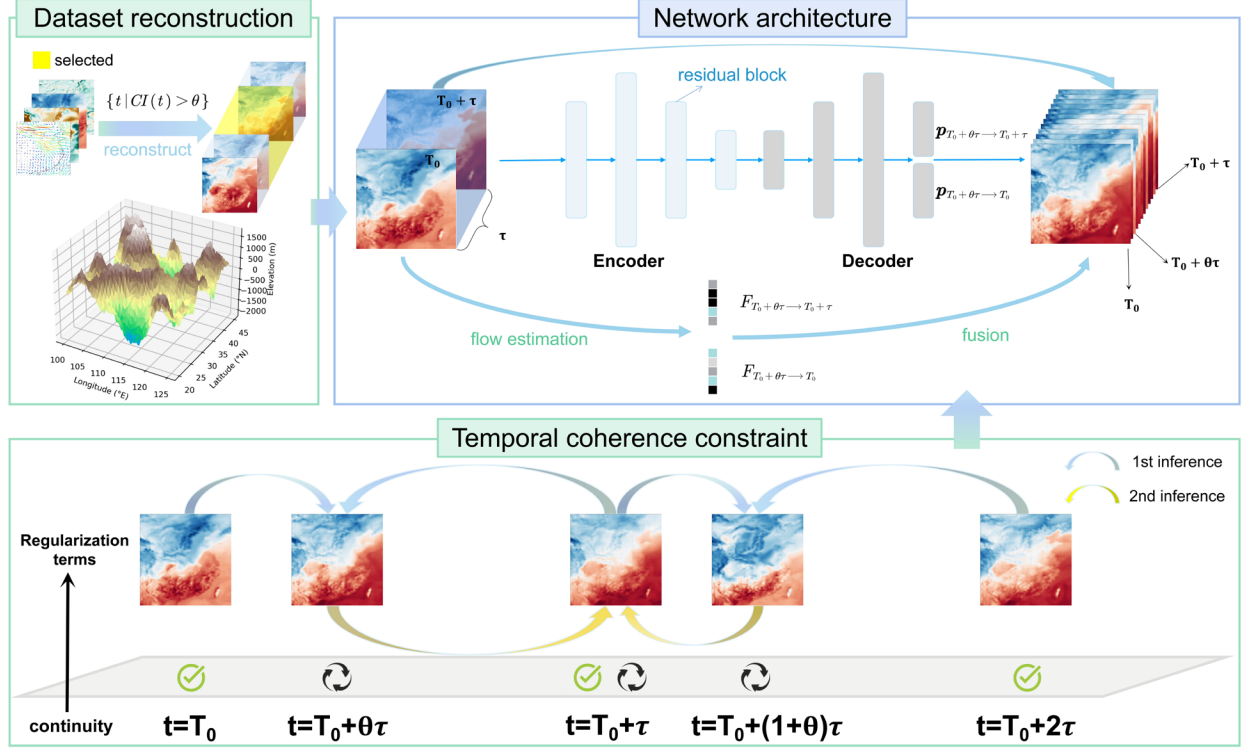



Figure 3. Overview of the proposed TemDeep framework for self-supervised temporal downscaling. The overall network structure for temporal downscaling is depicted in the top right portion of the figure, which is composed of an encoder-decoder structured field prediction network and a flow estimation module, taking consecutive fields and terrain data as input.

2.3 Reconstructing a pretraining dataset through self-similarity

It is easily understood that scenarios with rapid transitions could reflect a condensed evolution of atmospheric processes, where changes that might typically occur over longer durations are instead experienced in a compressed time period(Davis, 1994; Stanley, 1997). Therefore, these scenarios occurring within shorter time intervals in low-temporal-resolution data can potentially serve as 'pseudo labels' for scenarios within longer time intervals in high-temporal-resolution data.

Based on this kind of self-similarity across time scales, we propose to reconstruct a pretraining dataset by establishing a composite index to filter out scenarios with rapid transitions. This composite index is designed based on four physical variables that are indicative of weather system transformations, respectively *rh*, *t2m*, 850*hPa* wind speed (*v*) and 850*hPa* vertical velocity (*w*). Rapid changes in wind speed can indicate major weather phenomena, and similarly, humidity changes are key to atmospheric stability and sudden shifts can trigger severe convection. *t2m* gradients drive atmospheric circulation, with steep gradients signifying developing weather fronts. Lastly, vertical velocity indicates vertical air movement and can signal cloud formation or precipitation.

Given an atmospheric variable a, the normalized change for each time step t is defined as $\Delta a(t) = (a(t+1) - a(t))/\sigma(a)$, where $\sigma(a)$ represents the standard deviation of the variable a over the entire period. Let $V(t) = [\Delta rh(t), \Delta t2m(t), v(t), w(t)]$ denote the vector of normalized changes, and $W = [w1, w2, w3, w4]$ be the respective weight vector, the composite index *CI* can be expressed as

$$CI(t) = W^T \cdot V(t) + \eta \sum_{i<j} (w_{ij} \cdot V_i(t) \cdot V_j(t)). \tag{1}$$

165   Here, the superscript T denotes vector transposition, and the summation extends over all unique pairs of variables

166      $(i < j)$. The parameter η is a scaling factor set at 0.02, which can be adjusted to regulate the influence of the

167   interaction terms. w_{ij} represents the weights linked with the interaction terms, ensuring that each variable has the

168   same magnitude before multiplication. The first component $W^T \cdot V(t)$ is a linear combination of normalized

169   changes to quantify individual influence of each variable, while $\sum_{i<j} (w_{ij} \cdot V_i(t) \cdot V_j(t))$ is introduced to

170   account for synergistic effects among variables by measuring the product of changes between pairs of variables.

171   Finally, we empirically set a threshold θ for the composite index at 0.75, and scenarios with a CI value above the

172   threshold are considered to exhibit rapid transitions:

173
$$\mathbf{T} = \{ t \,|\, CI(t) > \theta \}, \tag{2}$$

174   where $\mathbf{T}$ denotes scenarios with rapid transitions. Finally, we obtained a collection of 1,391 scenarios with 12,531

175   consecutive fields and group these samples every 3 fields into 4,177 sets. During the pretraining process, we train the

176   model by providing the model with the two adjacent fields as input and tasking it to generate a result that is close to

177   the middle field in the sequence.

178   2.4 Self-supervised training leveraging temporal coherence

179   In our approach, we propose a self-supervised training process, which leverages temporal coherence within

180   continuous atmospheric fields to generate interpolated fields at arbitrary time resolutions. Taking inspiration from the

181   success of unpaired data to data translation in a variety of fields (Gao et al., 2022; Reda et al., 2019; Zhou et al., 2016;

182   Zhu et al., 2017), we define a time-domain temporal coherence constraint, ensuring that the interpolated data point

183      $\hat{\boldsymbol{p}}_{T_0+\tau}$ created at time $T_0 + \tau$ right between $\boldsymbol{p}_{T_0}$ and $\boldsymbol{p}_{T_0+2\tau}$ must consist with the original middle data point

184      $\boldsymbol{p}_{T_0+\tau}$. That is, as illustrated in Fig. 3, for a given triplet of consecutive data fields, we generate two intermediate

185   data points in the first inference: one between the first two data points $\hat{\boldsymbol{p}}_{T_0+\theta\tau} = \mathbf{M}(\boldsymbol{p}_{T_0}, \boldsymbol{p}_{T_0+\tau}, \theta\tau)$, where $\mathbf{M}$

186   is our downscaling network (see Section 2.6), and the other between the last two data points

187      $\hat{\boldsymbol{p}}_{T_0+(1+\theta)\tau} = \mathbf{M}(\boldsymbol{p}_{T_0+\tau}, \boldsymbol{p}_{T_0+2\tau}, \theta\tau)$. Then in the second inference, we generate an interpolated data point

188   between these newly created intermediate data points, $\hat{\boldsymbol{p}}_{T_0+\tau} = \mathbf{M}(\hat{\boldsymbol{p}}_{T_0+\theta\tau}, \hat{\boldsymbol{p}}_{T_0+(1+\theta)\tau}, (1-\theta)\tau)$. In this case,

189      $\hat{\boldsymbol{p}}_{T_0+\tau}$ should match the original middle input data point $\boldsymbol{p}_{T_0+\tau}$, illustrating the concept of temporal coherence. By

190   changing the time parameter $t = \theta\tau \, (\theta \in (0, 1))$, our method is capable of generating an array of interpolated data

191   points that maintain temporal coherence over time, effectively enriching the temporal resolution of the atmospheric

192   dataset. To enforce temporal coherence, we aim to minimize the difference between $\hat{\boldsymbol{p}}_{T_0+\tau}$ and $\boldsymbol{p}_{T_0+\tau}$, expressed

193   as $\arg\min_{\Phi(M)} \left\| \hat{\boldsymbol{p}}_{T_0+\tau} - \boldsymbol{p}_{T_0+\tau} \right\|_1$, then the coherence loss $L_c(\Phi)$ can be defined in the form of L_1 loss:

194
$$\mathcal{L}_c(\Phi) = \left\| \hat{\boldsymbol{p}}_{T_0+\tau} - \boldsymbol{p}_{T_0+\tau} \right\|_1. \tag{3}$$

195   2.5 Spatio-temporal continuity regularization

196   Despite the application of the temporal coherence constraint to train the model, which allows for the

197   simulation of evolving weather systems, it is still necessary to regulate the model to prevent it from local optima and

198   avoid generating abnormal values. To address this concern, our approach leverages the inherent continuity of

199   atmospheric fields in space and time, which is integrated into our model training process as a regularization term in

200   the loss function. An example of spatial and temporal gradients in *t2m* fields is provided in Fig. 4, and Fig. 5 indicates

201   that 99.59% of the horizontal gradients and 99.31% of the vertical gradients are lower than *3K* respectively. Similarly,

202   in the continuously varying fields, 99.55% of the temporal gradients are lower than *3K*. Therefore, it can be assumed

that the majority of grid points in *t2m* fields exhibit strong spatial and temporal continuity, as well as other densely distributed atmospheric variables, such as geopotential height and relative humidity. Here, spatial continuity implies that nearby locations should share similar atmospheric conditions, and our model incorporates a spatial continuity loss term to ensure smoothness in both horizontal and vertical directions:

$$\mathcal{L}_s = \frac{1}{2}\left(\sum_{(x,y)\subset\Omega}\|\hat{z}(t,x+1,y)-\hat{z}(t,x,y)\|_1 + \sum_{(x,y)\subset\Omega}\|\hat{z}(t,x,y+1)-\hat{z}(t,x,y)\|_1\right),$$

(4)

where $\hat{z}(t,x,y)$ represents the model's prediction at time t and location (x,y). Meanwhile, temporal continuity assumes that the atmospheric conditions do not change abruptly over short periods, and accordingly, our loss function includes a temporal continuity term that penalizes substantial differences between the model's predictions at three consecutive time steps:

$$\mathcal{L}_t = \lambda(\|\hat{z}(t+1)-\hat{z}(t)\|_1 + \|\hat{z}(t)-\hat{z}(t-1)\|_1),$$

(5)

where $\hat{z}(t)$ denotes the model's prediction at time t, and λ is a parameter set at 0.35 to control the weight of temporal continuity in the loss function.

horizontal gradients   vertical gradients   spatial gradients   time gradients

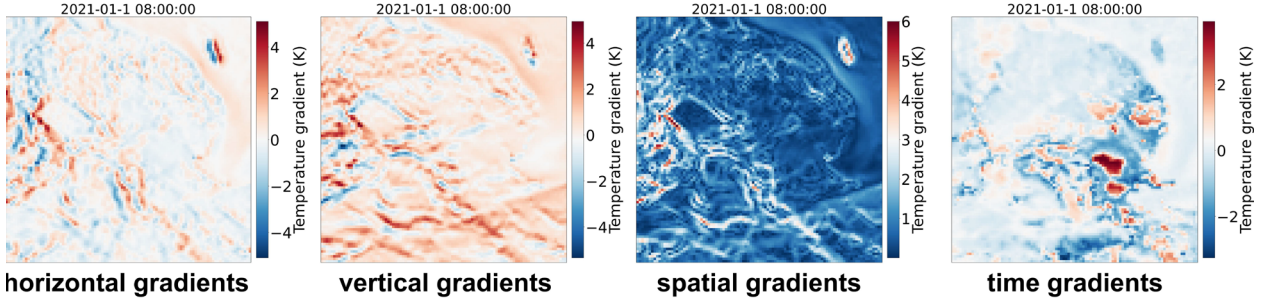



Figure 4. Spatial and temporal gradients in *t2m* fields.

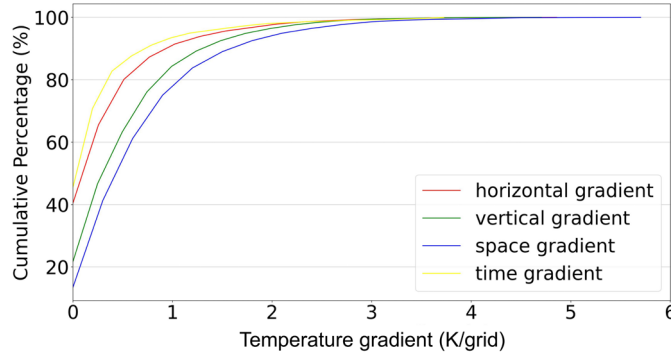



Figure 5. Cumulative percentage of spatial and temporal gradients.

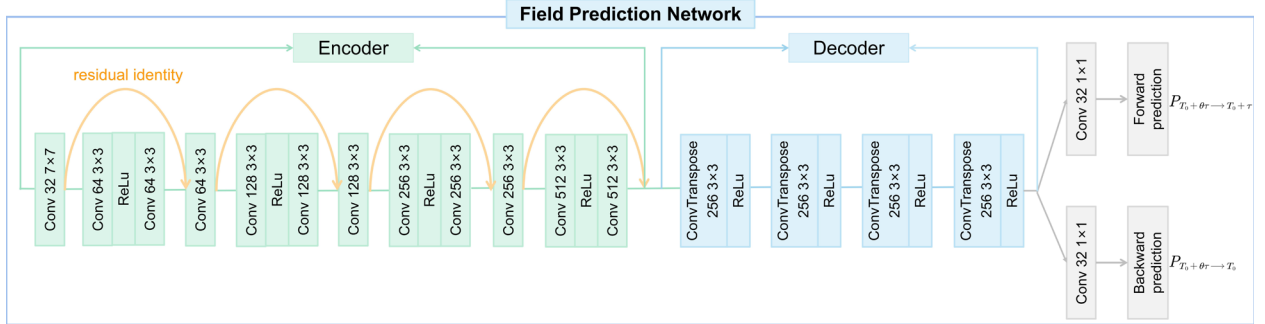
2.6 Network architecture

      In this section, we will introduce the network architecture of TemDeep for generating interpolated fields. As illustrated in Fig. 3, the field prediction network, serving as the backbone network, adopts an encoder-decoder structure to generate intermediate fields (see Fig. 6, Section 2.6.1). Meanwhile, the flow estimation module adopts a unique combination of larger convolutional kernels and Leaky ReLU activations to capture long-range motions (see Fig. 7, Section 2.6.2). Finally, intermediate fields and estimated flow are fused to synthesize interpolated fields (Section 2.6.3).

227   2.6.1 Field prediction network

228   The field prediction network is composed of an encoder-decoder architecture with the inclusion of residual
229   blocks(Azad et al., 2024), as shown in Fig. 6. It takes consecutive single-element fields and terrain data as input, and
230   outputs intermediate fields. The encoder part includes four primary components, each comprised of a convolutional
231   layer and a subsequent residual block. These convolutional layers, coupled with ReLU activation functions, process
232   input data through multiple filter sizes (64, 128, 256, and 512 filters respectively). Notably, the first convolutional
233   layer incorporates a 7×7 kernel with a stride of 2 and padding of 3, enabling more robust feature extraction at the
234   initial stage, while subsequent layers employ 3×3 kernels with a stride of 1 and padding of 1. After each convolutional
235   layer, a corresponding residual block follows, with in-channels and out-channels matching the corresponding
236   convolutional layer's filter size. These residual blocks consist of two convolutional layers and ReLU activation
237   functions, which helps in preserving the identity function and facilitates deeper model learning without the problem
238   of vanishing gradients. The decoder part is designed to upsample and reconstruct the encoded field back to its original
239   resolution. It consists of four deconvolutional layers, each applying the ConvTranspose2d function for upsampling,
240   and these layers upsample the data from 512 filters back to 2 filters, which corresponds to the output flow. Notably,
241   the kernel size used in these layers is 4 with a stride of 2 and padding of 1, which efficiently enlarges the spatial
242   dimensions back to the original size. After a convolutional layer, we obtain forward and backward prediction results:
243    $\vec{p}$ and $\overleftarrow{p}$.

244   Additionally, to process topographic information and integrate it into input, we introduce a Convolutional
245   Terrain Integration Module (CTIM). The CTIM employs a convolutional layer with 3×3 kernels, to create an
246   intermediate feature map topographic information. Subsequent to the convolution operation, batch normalization is
247   applied to accelerate the training process, followed by a ReLU activation function to introduce non-linearity. This
248   output then passes through a second convolutional layer with 3×3 kernels to further refine the feature representation.
249   Once again, we apply batch normalization and ReLU activation to this output. The resulting output from the CTIM is
250   a set of terrain feature maps, ready to be fed into the prediction network.

251

252   **Figure 6. Field prediction network.**

253   2.6.2 Flow estimation module

254   The flow estimation module aims to estimate motion information and calculate forward and backward flow,
255   which is then fused with the intermediate fields from the field prediction network to assist with generating interpolated
256   fields. Fig. 8 provides an example of calculated flow in *t2m* fields.

257   The flow encoder is structured similarly to the encoder of the field prediction network, which comprises four
258   convolutional layers, each followed by a Leaky ReLU activation function. The initial layer utilizes a 7×7
259   convolutional kernel to extract features from the input, stepping down to a stride of 2 and padding of 3. Following
260   this, the subsequent layers use 3×3 convolutional kernels with a stride of 1 and padding of 1, moving from 64 to 128,
261   256, and finally to 512 filters for a more detailed and intricate feature extraction. The subspace features obtained at
262   this layer, after undergoing convolution and ReLU, yield an activation map $\vec{V}$ (see Eq. 6).

263   The flow decoder includes five deconvolution layers that upscale the downsampled encoder outputs. Each
264   layer employs a bilinear upsampling technique to double the spatial dimension, followed by two convolutional layers

and a Leaky ReLU activation. Finally, we obtain forward flow $\mathcal{F}_{T_0+\theta\tau \rightarrow T_0+\tau}$ and backward flow $\mathcal{F}_{T_0+\theta\tau \rightarrow T_0}$ after two convolutional layers.

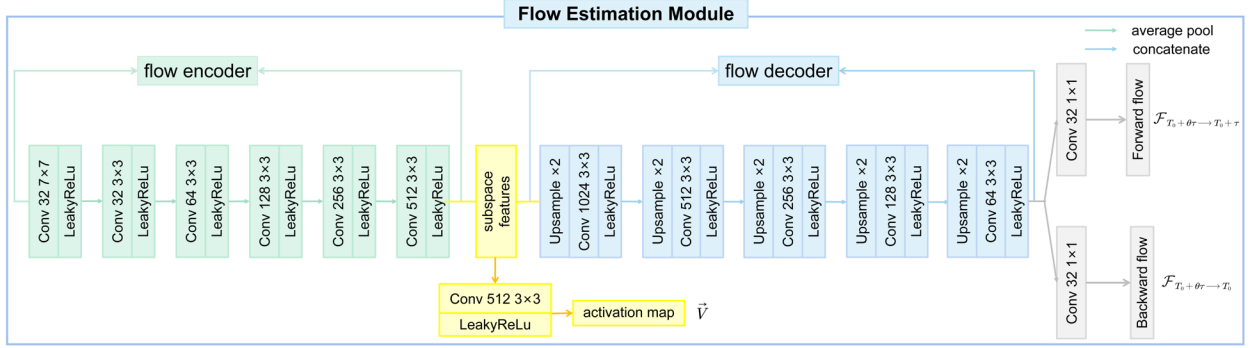



Figure. 7. Flow estimation module.

| 08:00 | 12:00 | 08:00→12:00 | 12:00→08:00 |
| (a) | (b) | (c) | (d) |

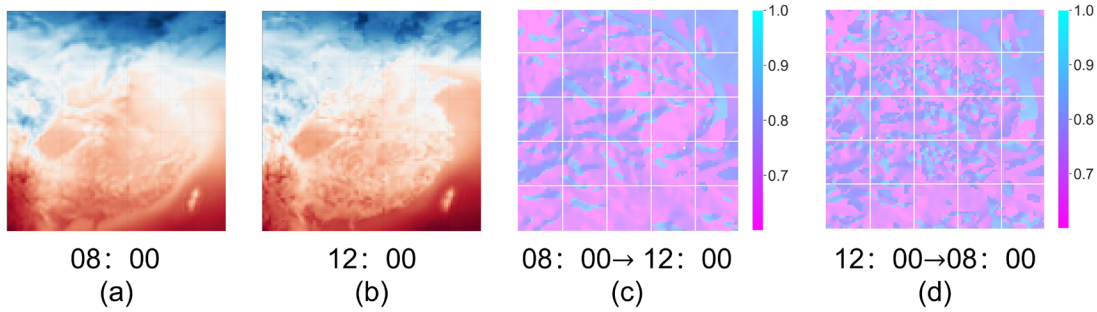



Figure. 8. Forward and backward flow visualization. A and **b** represent the *t2m* fields at 08:00 and 12:00 on January 1, 2021, while **c** and **d** represent the forward flow from 08:00 to 12:00 and backward flow from 12:00 to 08:00, respectively.

2.6.3 Fusion and loss function

We can synthesize the target field $\hat{\boldsymbol{p}}_{T_0+\theta\tau}$ by fusing the outputs from the field prediction network and the flow estimation module as follows:

$$\hat{\boldsymbol{p}}_{T_0+\theta\tau} = (1-\theta)\overleftarrow{V} \odot g\left(\overleftarrow{\boldsymbol{p}}, \mathcal{F}_{T_0+\theta\tau \rightarrow T_0}\right) + \theta\overrightarrow{V} \odot g\left(\overrightarrow{\boldsymbol{p}}, \mathcal{F}_{T_0+\theta\tau \rightarrow T_0+\tau}\right), \tag{6}$$

where $g(\cdot)$ is a warping function (Jiang et al., 2017). $\overrightarrow{V}$ represents the activation map, referring to whether pixels remain activated when moving forward from $t=T_0$ to $t=T_0+\theta\tau$ and $\overleftarrow{V}$ is calculated by $\overleftarrow{V} = \left[1-\overrightarrow{V}(i,j)\right]$.

In order to make the estimated flow resemble the actual flow, we utilize it as motion information to further assist in enhancing the quality of field reconstruction, and accordingly, the flow estimation loss can be defined as

$$\mathcal{L}_{\mathcal{F}} = \left\| \boldsymbol{p}_{T_0} - g\left(\overleftarrow{\boldsymbol{p}}, \mathcal{F}_{T_0+\theta\tau \rightarrow T_0}\right) \right\|_1 + \left\| \boldsymbol{p}_{T_0+\tau} - g\left(\overrightarrow{\boldsymbol{p}}, \mathcal{F}_{T_0+\theta\tau \rightarrow T_0+\tau}\right) \right\|_1 \tag{7}$$

Finally, the loss function to train the model can be expressed by combing the coherence loss $\mathcal{L}_c$ (Section 3.3), flow estimation loss $\mathcal{L}_{\mathcal{F}}$ (Section 3.5.3), and continuity loss $\mathcal{L}_s + \mathcal{L}_t$ (Section 3.4) as

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_{\mathcal{F}} + \alpha(\mathcal{L}_s + \mathcal{L}_t), \tag{8}$$

where α is a parameter set at 0.35 to adjust the weight of continuity regularization.

2.7 Evaluation metrics

In order to evaluate the performance of our model, we propose three metrics: restoration rate (*Re*), consistency degree (*CS*), and continuity degree (*CT*). Among them, *Re* is primarily utilized for the evaluation of the discrepancy between the downscaled results and the true values, while *CS* and *CT* are auxiliary metrics for the analysis and comparison of different methods.

The restoration rate measures the degree to which our model recovers lost information compared to simple linear interpolation, and a larger *Re* indicates a better downscaling performance. Let the restoration rate of linear interpolation as zero, then the formula for calculating *Re* is as follows:

$$Re = 1 - \frac{\dfrac{1}{\Omega} \displaystyle\sum_{(x,y) \subset \Omega} |D^{truth}(x,y) - D(x,y)|^2}{\dfrac{1}{\Omega} \displaystyle\sum_{(x,y) \subset \Omega} |D^{truth}(x,y) - D^{lin}(x,y)|^2}. \tag{9}$$

In this formula, D^{truth} is the ground truth, D are the data generated by our model, D^{lin} is calculated through linear interpolation and Ω represents all pixels in the field.

The consistency degree is a metric used to evaluate the level of consistency in generated fields, and a larger *CS* indicates a smaller discrepancy between the estimated flow $\hat{F}_{T_0 \rightarrow T_0 + \theta\tau}$ and the true flow $F_{T_0 \rightarrow T_0 + \theta\tau}$. It is calculated based on the flow estimation module and can be expressed as

$$CS = 1 - \frac{\left\| \mathcal{F}_{T_0 \rightarrow T_0 + \theta\tau} - \hat{\mathcal{F}}_{T_0 \rightarrow T_0 + \theta\tau} \right\|_1}{\left\| \mathcal{F}_{T_0 \rightarrow T_0 + \theta\tau} \right\|_1}. \tag{10}$$

The continuity degree measures how smoothly the preceding field transitions to the next, and a larger *CT* indicates more smoothness. The mathematical representation is

$$CT = 1 - \frac{\left| \left\| p_{T_0 + \theta\tau} - p_{T_0} \right\|_1 - \left\| \hat{p}_{T_0 + \theta\tau} - p_{T_0} \right\|_1 \right|}{\left\| p_{T_0 + \theta\tau} - p_{T_0} \right\|_1}, \tag{11}$$

where $\hat{p}_{T_0 + \theta\tau}$ is the interpolated field and p_{T_0} is the preceding field.

Table 1. Performance comparison among different methods based on *Re*. According to Eq. 9, the result of linear interpolation is set to 0 as the basis for comparing other methods. Among all unsupervised comparison methods, TemDeep achieved the best performance, approaching the supervised TemDeep*. *rh* and *z* are only downscaled from 2-hourly to 1-hourly intervals.

Methods	*t2m* (2h→1)	*t2m* (3h→1)	*t2m* (4h→1)	*t2m* (5h→1)	*z* (2h→1)	*rh* (2h→1)
Linear	0.000	0.000	0.000	0.000	0.000	0.000
Cubic spline	0.102	0.041	0.019	0.018	0.135	0.074
Optical flow	0.219	0.188	0.102	0.059	0.342	0.236
non-flow	0.462	0.431	0.359	0.307	0.505	0.417
non-regular	0.499	0.470	0.397	0.325	0.525	0.488

non-pretrain	0.528	0.501	0.433	0.372	0.568	0.489
TemDeep	*0.537*	*0.508*	*0.442*	*0.376*	*0.576*	*0.498*
TemDeep*	**0.682**	**0.641**	**0.579**	**0.430**	**0.701**	**0.553**

Note: "non-flow," "non-regular," and "non-pretrain" represent methods that do not include flow estimation, regularization modules, and pretraining steps, respectively.

3 Results and discussion

We conduct experiments on an Ubuntu 20.04 system equipped with eight A100 GPUs. The TemDeep model is trained using the adam optimizer (Kingma & Ba, 2014) with an initial learning rate of 1e-5, and a mini-batch size of 256. Downscaling results of $t2m$, z and rh fields at different time resolutions, respectively 2, 3, 4, 5 and 6 hours, into 1-hour time intervals, are shown in Table 1.

3.1 Quantitative analysis

In order to evaluate the effectiveness of our proposed method on temporal downscaling, we select several methods that do not require supervision information for comparison, namely linear interpolation, cubic spline interpolation and optical flow-based interpolation. The linear interpolation method computes the average value between adjacent fields, while cubic spline interpolation, using four data fields, achieves a smooth curve with cubic polynomials. Additionally, optical flow-based interpolation estimates pixel motion between fields to predict their state at a desired time point. As illustrated in Table 1, for the six tasks of $t2m$ (2h→1), $t2m$ (3h→1), $t2m$ (4h→1), $t2m$ (5h →1), z (2h→1), and rh (2h→1), the TemDeep method scores 0.537, 0.508, 0.442, 0.376, 0.576, and 0.498 in Re, respectively, all considerably higher than the scores achieved by other methods under unsupervised conditions. The approximate time for each inference is 600ms. Without the pretraining stage, Re is relatively lower on all tasks, suggesting that this stage is important in initially capturing general patterns in atmospheric data. The supervised training condition TemDeep* method scores the highest, implying that supervised training can further enhance the downscaling performance of the TemDeep method.

(a) T00 ground truth   (b) T06 ground truth   (c) T11 ground truth   (d) T06 wind

(e) T06 linear   (f) T06 optical flow   (g) T06 TemDeep   (h) T06 non-regular

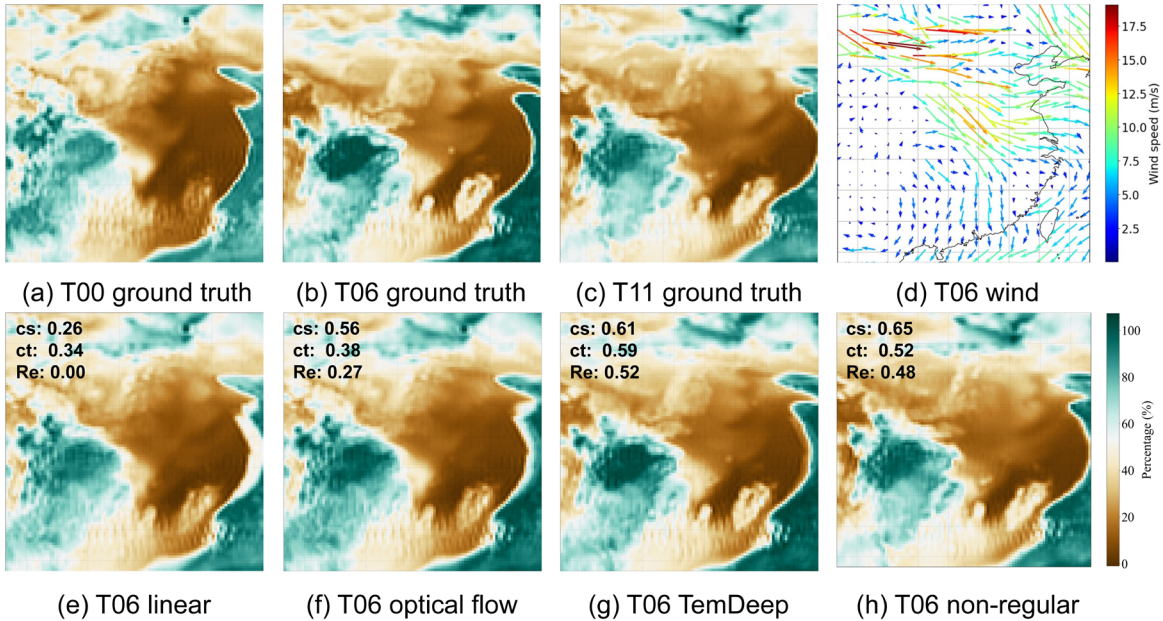



Figure 9. Visualized comparison. a, **b**, and **c** respectively represent the true values of rh fields on January 1, 2021. **d** represents the wind direction and speed at 850 hPa at 06:00, where the wind in the central region points towards the southeast, driving the dry air mass in the same direction, resulting in the expansion of the dry area towards the southeast. **e**, **f**, and **g** display the interpolation results at 06:00 obtained through different methods. **h** represents the result from TemDeep when the spatio-temporal continuity regularization is removed.

336   The flow estimation module provides an improvement of 0.075 in *Re* by guiding the model to learn the
337 movement of weather systems, and the result demonstrates more consistency with the trends of weather system
338 movements, as illustrated in Fig. 9g. In contrast, if completely ignoring the motion of weather systems, the result of
339 time interpolation would simply be an average of the preceding and succeeding fields, leading to significant errors
340 compared to the ground truth, as shown in Fig. 9e. The spatio-temporal continuity regularization also provides an
341 improvement of 7.6% from 0.499 to 0.537 in *Re* by ensuring the generated fields be consistent with the observed
342 patterns in the input data. As depicted in Fig. 9h, without this regularization, the model occasionally produces
343 erroneous estimates of the intensity and direction of motion. Nevertheless, with the inclusion of the regularization
344 term, the results are inevitably constrained to linear changes to a certain degree, which has conflicts with the actual
345 non-linear evolutions.

346

347 **Figure 10. Model performance under the enforcement of spatio-temporal continuity with varying weights. a**
348 shows *Re* of TemDeep trained under self-supervised conditions and supervised conditions (denoted as $\mathrm{TemDeep}^{*}$)
349 at different α. **b** shows consistency degree and continuity degree of TemDeep at different α.

350   To strike a balance between the spatio-temporal continuity regularization and actual non-linear evolutions,
351 we introduce a parameter α in the loss function to adjust the weight for regularization and conduct ablation studies,
352 with the results shown in Fig. 10. A larger α implies that the model emphasizes on regularization, and thus CT
353 increases while CS decreases. Finally, α is set at 0.35 and *Re* reaches a maximum of 0.537.

354   Fig. 11 shows the restoration rate of the test set in these experiments. Increasing the training dataset size
355 consistently improves model performance, but the impact diminishes gradually. Once the amount of training data
356 reaches a critical value (e.g., 8,760), further increases no longer result in significant improvements, suggesting the
357 model is reaching its performance limits. When the data volume reaches 26,280, doubling the data leads to only a
358 modest 1-2% improvement.

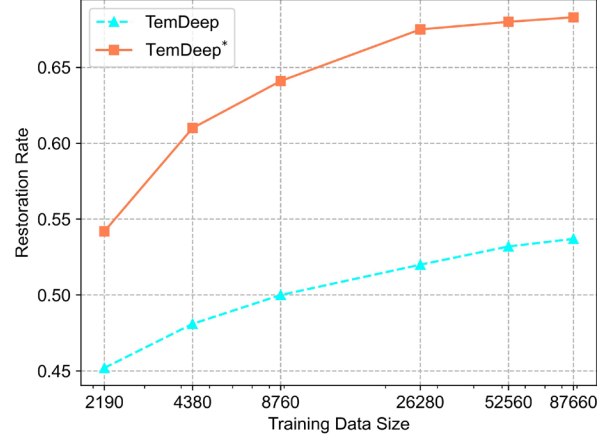



Figure 11. Average test loss on *t2m* fields.

3.2 Case study

In this section, a case study is employed to explore TemDeep's ability in recovering evolving details of *t2m*, *z* and *rh* fields, as shown in Fig. 12. Hourly interpolation is conducted between 08:00 and 12:00 on January 1, 2021, to obtain three interpolated fields at 09:00, 10:00 and 11:00 (UTC).

In the temporal interpolation of *t2m* fields, the selected area in January exhibits a noticeable temperature difference between the sea and the land at 12:00 compared to 08:00, and the gradual changes occurring at 09:00, 10:00, and 11:00 are clearly reproduced by the TemDeep method. Due to the sensitivity of *t2m* to altitude, the temperature gradient near the Sichuan Basin is clearly depicted, closely aligning with the contour of the actual altitude gradient, as marked by the rectangle. Most importantly, at 10:00, regions marked by the triangles exhibit large surrounding gradients and non-linear abrupt changes, resulting in a lower continuity degree of 0.54. In this case, the TemDeep method still achieves a high precision in reproducing the field, with a restoration rate of 0.49, reaching 0.48 and 0.52 at the preceding and following field, respectively.

For the 850hPa *z* fields, their variations are relatively simpler compared to the *t2m* fields, making downscaling easier and leading to less precision fluctuation. The average *Re* over the three-hour period reaches 0.56. At 08:00, there exists a high-pressure region on the western edge, surrounded by low pressure, resulting in a significant gradient. In the generated *z* fields, this gradient gradually diminishes from 09:00 to 11:00, and the central high-pressure region moves northeastward and eventually dissipates, as marked by the ellipse and arrow, which evolves closely in accordance with the ground truth.

Similarly, in the three generated *rh* fields, the drier region on the eastern edge can be observed slowly moving eastward, consistent with the ground truth. At 08:00, the drier region is still located some distance away from the 125°E line, but after four hours of continuous changes, the easternmost part of the dry region has already crossed the 125°E line, and TemDeep has reproduced this movement of dry air mass, rather than simply averaging the fields.

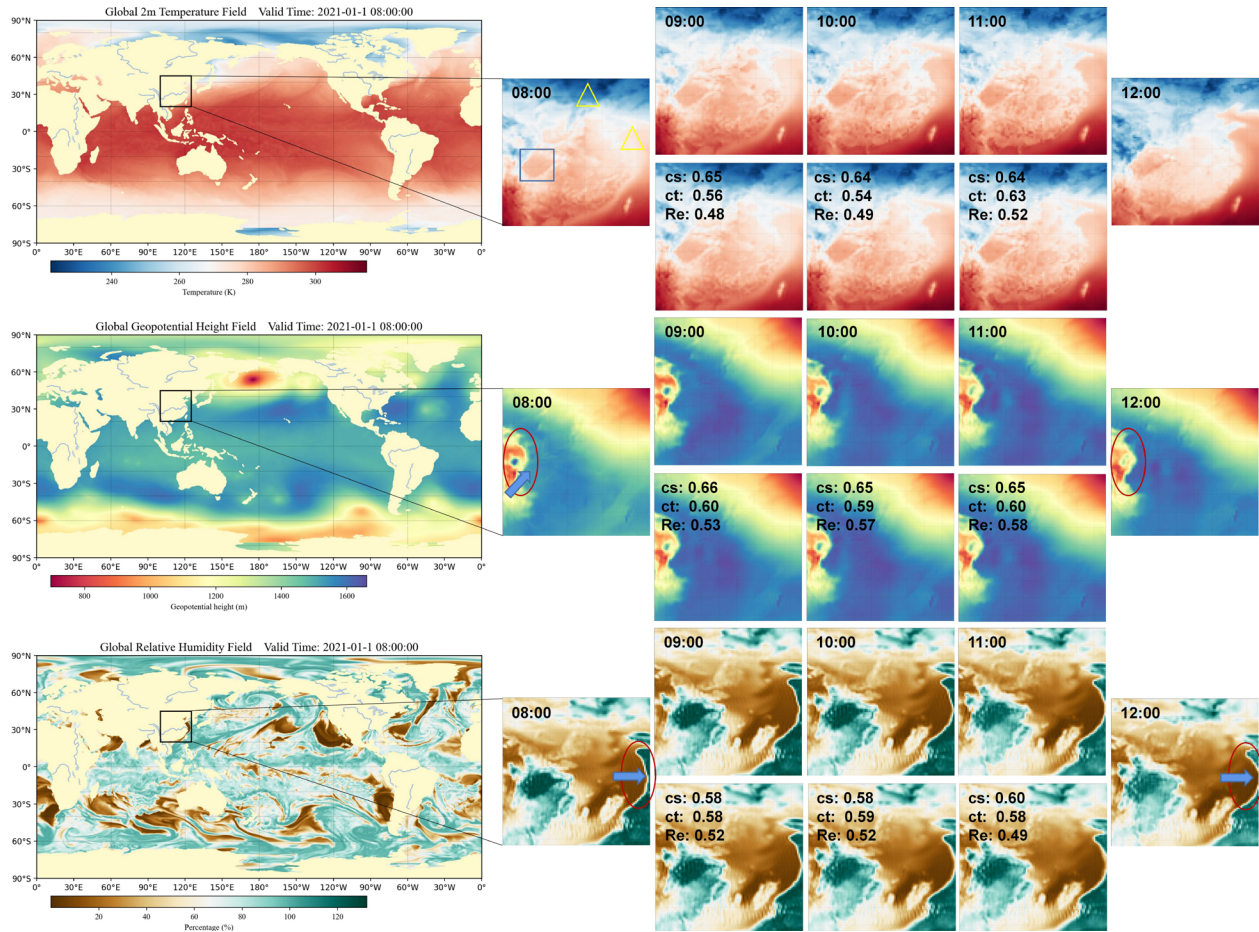



Figure 12. Case study.

4 Conclusions

This paper proposes a self-supervised model for downscaling atmospheric fields at arbitrary time resolutions by leveraging temporal coherence. This model combines an encoder-decoder structured field prediction network with a flow estimation module, fuses intermediate fields and motion information of weather systems and finally synthesizes fields at desired time points. We first pretrain the model based on a reconstructed dataset to initially capture data patterns, and then further utilize existing consecutive fields as supervision for model training. Experiments on three variables ($t2m$, z, rh) indicate that the proposed TemDeep model can accurately reconstruct the evolutionary process of atmospheric variables at the 1-hourly resolution, superior to other unsupervised methods.

As for future research, we will explore multi-modal data fusion to leverage complementary information from various sources. Since ERA5 only provides data at 1-hour temporal resolution, further research will focus on identifying datasets with higher temporal resolution for more accurate downscaling. Further, we plan to extend our downscaling model based on previous work of self-supervised weather system classification (Wang et al., 2022), that is, to downscale temporal and spatial data by referring to similar types of weather systems through similarity search in the historical dataset. To enable real-time downscaling and more refined forecasting, we will also work on simplifying the model architecture to reduce computational complexity, making it more feasible for deployment in operational environments where fast processing times are critical.

Code and Data Availability Statement. All data necessary to reproduce the results of this work can be downloaded at https://doi.org/10.24381/cds.bd0915c6 and https://doi.org/10.24381/cds.adbb2d47. The scripts used for downscaling are freely available at https://github.com/GeoSciLab/TemDeep and have also been permanently archived at Zenodo with the DOI https://zenodo.org/records/14062314.

405 *Author contributions.* Liwen Wang was primarily responsible for the design of the model and conducting the
406 experiments. Qian Li prepared the experimental datasets and organized the entire research project. Qi Lv, Xuan Peng,
407 and Wei You contributed to the optimization of the experimental code.

408 *Competing interests.* The contact author has declared that none of the authors have any competing interests.

409 *Acknowledgments:* This research was funded by the National Natural Science Foundation of China (Grant No.
410 42075139, U2242201, 42105146, 41305138), the China Postdoctoral Science Foundation (Grant No. 2017M621700),
411 Hunan Province Natural Science Foundation (Grant No. 2021JC0009, 2021JJ30773) and Fengyun Application
412 Pioneering Project (FY-APP-2022.0605).

References

414 Azad, R., Aghdam, E. K., Rauland, A., Jia, Y., Avval, A. H., Bozorgpour, A., Karimijafarbigloo, S., Cohen, J. P.,
415   Adeli, E., & Merhof, D. (2024). Medical image segmentation review: The success of u-net. *IEEE*
416   *Transactions on Pattern Analysis and Machine Intelligence*.
417 Barboza, L. A., Chen, S., & Alfaro-Córdoba, M. (2022). Spatio‐temporal downscaling emulator for regional
418   climate models. *Environmetrics*.
419 Bauer, P., Thorpe, A. J., & Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature*, *525*,
420   47-55.
421 Benjamin, S. G., Weygandt, S. S., Brown, J. M., Hu, M., Alexander, C. R., Smirnova, T. G., Olson, J. B., James, E.
422   P., Dowell, D. C., Grell, G., Lin, H., Peckham, S. E., Smith, T. L., Moninger, W. R., Kenyon, J. S., &
423   Manikin, G. S. (2016). A North American Hourly Assimilation and Model Forecast Cycle: The Rapid
424   Refresh. *Monthly Weather Review*, *144*, 1669-1694.
425 Chen, G. F., Qin, D., Ye, R., Guo, Y., & Wang, H. (2011). A new method of rainfall temporal downscaling: a case
426   study on sanmenxia station in the Yellow River Basin. *Hydrology and Earth System Sciences Discussions*,
427   *8*, 2323-2344.
428 Davis, A., A. Marshak, W. Wiscombe, and R. Cahalan. (1994). Multifractal characterizations of nonstationarity and
429   intermittency in geophysical fields: Observed, retrieved, or simulated. *Journal of Geophysical Research:*
430   *Atmospheres*, *99*(D4), 8055-8072. https://doi.org/https://doi.org/10.1029/94JD00219
431 Dong, J., Xiao, X., Chen, B., Torbick, N., Jin, C., Zhang, G., & Biradar, Ç. M. (2013). Mapping deciduous rubber
432   plantations through integration of PALSAR and multi-temporal Landsat imagery. *Remote Sensing of*
433   *Environment*, *134*, 392-402.
434 Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C., Li, X., & Guan, C. (2022). Self-supervised Contrastive
435   Representation Learning for Semi-supervised Time-Series Classification. *ArXiv*, *abs/2208.06616*.
436 Fowler, H. J., Blenkinsop, S., & Tebaldi, C. (2007). Linking climate change modelling to impacts studies: recent
437   advances in downscaling techniques for hydrological modelling. *International Journal of Climatology*, *27*.
438 Gao, L., Han, Z., Hong, D., Zhang, B., & Chanussot, J. (2022). CyCU-Net: Cycle-Consistency Unmixing Network
439   by Learning Cascaded Autoencoders. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1-14.
440 Gutmann, E. D., Rasmussen, R. M., Liu, C., Ikeda, K., Gochis, D. J., Clark, M. P., Dudhia, J., & Thompson, G.
441   (2011). A Comparison of Statistical and Dynamical Downscaling of Winter Precipitation over Complex
442   Terrain. *Journal of Climate*, *25*, 262-281.
443 Hawkins, E., & Sutton, R. T. (2011). The potential to narrow uncertainty in projections of regional precipitation
444   change. *Climate Dynamics*, *37*, 407-418.
445 Hennig, T. A., Kretsch, J. L., Pessagno, C. J., Salamonowicz, P. H., & Stein, W. L. (2001). The Shuttle Radar
446   Topography Mission. Digital Earth Moving,
447 Hurrell, J. W., Holland, M. M., Gent, P. R., Ghan, S. J., Kay, J. E., Kushner, P. J., Lamarque, J. F., Large, W.,
448   Lawrence, D. M., Lindsay, K., Lipscomb, W. H., Long, M. C., Mahowald, N. M., Marsh, D. R., Neale, R.
449   B., Rasch, P. J., Vavrus, S. J., Vertenstein, M., Bader, D. C.,…Marshall, S. J. (2013). The Community
450   Earth System Model: A Framework for Collaborative Research. *Bulletin of the American Meteorological*
451   *Society*, *94*, 1339-1360.
452 Jia-hong, L. (2006). The digital watershed model for the Yellow River basin. *Water Resources and Hydropower*
453   *Engineering*.
454 Jiang, H., Sun, D., Jampani, V., Yang, M.-H., Learned-Miller, E. G., & Kautz, J. (2017). Super SloMo: High Quality
455   Estimation of Multiple Intermediate Frames for Video Interpolation. *2018 IEEE/CVF Conference on*
456   *Computer Vision and Pattern Recognition*, 9000-9008.

Kajbaf, A. A., Bensi, M. T., & Brubaker, K. L. (2022). Temporal downscaling of precipitation from climate model projections using machine learning. *Stochastic Environmental Research and Risk Assessment*, *36*, 2173 - 2194.

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *CoRR*, *abs/1412.6980*.

Lawrimore, J. H., Menne, M. J., Gleason, B. E., Williams, C. N., Wuertz, D. B., Vose, R. S., & Rennie, J. J. (2011). An overview of the Global Historical Climatology Network monthly mean temperature data set, version 3. *Journal of Geophysical Research*, *116*.

Lee, T., Ouarda, T. B. M. J., & Jeong, C. (2012). Nonparametric multivariate weather generator and an extreme value theory for bandwidth selection. *Journal of Hydrology*, *452*, 161-171.

Liu, X., Zhang, F., Hou, Z., Wang, Z., Mian, L., Zhang, J., & Tang, J. (2020). Self-Supervised Learning: Generative or Contrastive. *IEEE Transactions on Knowledge and Data Engineering*, *35*, 857-876.

Lobell, D., & Asseng, S. (2017). Comparing estimates of climate change impacts from process-based and statistical crop models. *Environmental Research Letters*, *12*.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, *20*, 130-141.

Lorenz, E. N. (1969). The predictability of a flow which possesses many scales of motion. *Tellus A*, *21*, 289-307.

Maraun, D. (2010). Precipitation downscaling under climate change.

McGovern, A., Elmore, K. L., Gagne, D. J., Haupt, S. E., Karstens, C. D., Lagerquist, R., Smith, T. M., & Williams, J. K. (2017). Using Artificial Intelligence to Improve Real-Time Decision-Making for High-Impact Weather. *Bulletin of the American Meteorological Society*, *98*, 2073-2090.

Mendes, D., & Marengo, J. A. (2010). Temporal downscaling: a comparison between artificial neural network and autocorrelation techniques over the Amazon Basin in present and future climate change scenarios. *Theoretical and Applied Climatology*, *100*, 413-421.

Michel, A., Sharma, V., Lehning, M., & Huwald, H. (2021). Climate change scenarios at hourly time‐step over Switzerland from an enhanced temporal downscaling approach. *International Journal of Climatology*, *41*, 3503 - 3522.

Neukom, R., Steiger, N. J., Gómez-Navarro, J. J., Wang, J., & Werner, J. P. (2019). No evidence for globally coherent warm and cold periods over the preindustrial Common Era. *Nature*, *571*, 550 - 554.

Palmer, T. N., Doblas-Reyes, F. J., Hagedorn, R., & Weisheimer, A. (2005). Probabilistic prediction of climate using multi-model ensembles: from basics to applications. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *360*, 1991 - 1998.

Pang, Y., Wang, W., Tay, F. E. H., Liu, W., Tian, Y., & Yuan, L. (2022). Masked Autoencoders for Point Cloud Self-supervised Learning. European Conference on Computer Vision,

Papalexiou, S. M., Markonis, Y., Lombardo, F., Aghakouchak, A., & Foufoula‐Georgiou, E. (2018). Precise Temporal Disaggregation Preserving Marginals and Correlations (DiPMaC) for Stationary and Nonstationary Processes. *Water Resources Research*, *54*, 7435 - 7458.

Raymond, C., Singh, D., & Horton, R. M. (2017). Spatiotemporal Patterns and Synoptics of Extreme Wet‐Bulb Temperature in the Contiguous United States. *Journal of Geophysical Research: Atmospheres*, *122*, 13,108 - 113,124.

Reda, F. A., Sun, D., Dundar, A., Shoeybi, M., Liu, G., Shih, K. J., Tao, A., Kautz, J., & Catanzaro, B. (2019). Unsupervised Video Interpolation Using Cycle Consistency. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 892-900.

Requena, A. I., Nguyen, T.-H., Burn, D. H., Coulibaly, P., & Nguyen, V. T. V. (2021). A temporal downscaling approach for sub-daily gridded extreme rainfall intensity estimation under climate change. *Journal of Hydrology: Regional Studies*, *35*, 100811.

Seneviratne, S. I., Nicholls, N., Easterling, D. R., Goodess, C., Kanae, S., Kossin, J. P., Luo, Y., Marengo, J. A., McInnes, K. L., Rahimi, M., Reichstein, M., Sorteberg, A., Vera, C. S., Contributors, A., Alexander, L. V., Allen, S. J., Benito, G., & Cavazos, T. (2012). Changes in climate extremes and their impacts on the natural physical environment.

Skamarock, C., Klemp, B., Dudhia, J., Gill, O., Barker, D. M., Duda, G., Huang, X., Wang, W., & Powers, G. (2008). A Description of the Advanced Research WRF Version 3.

Stanley, H. E. (1997). Turbulence: The legacy of A. N. Kolmogorov. *Journal of Statistical Physics*, *88*(1), 521-523. https://doi.org/10.1007/BF02508484

Wang, L., Li, Q., & Lv, Q. (2022). Self‐Supervised Classification of Weather Systems Based on Spatiotemporal Contrastive Learning. *Geophysical Research Letters*, *49*.

511   Zhou, T., Krähenbühl, P., Aubry, M., Huang, Q., & Efros, A. A. (2016). Learning Dense Correspondence via 3D-
512   Guided Cycle Consistency. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*,
513   117-126.
514   Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent
515   Adversarial Networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2242-2251.

516