Earthformer: Exploring Space-Time Transformers for Earth System Forecasting

Anonymous Author(s)
Affiliation
Address
email

Abstract

It has been studied for centuries to predict the evolution of the Earth system due to its significant impact on human lives. Conventionally, Earth system (e.g., weather and climate) forecasting models rely on numerical simulation of complex physical models and are hence expensive in both computational resources and domain expertise. With the explosive growth of Earth observation data in the past decade, data-driven models that apply Deep Learning (DL) are demonstrating impressive potential for various Earth system forecasting tasks. So far, these DL models mainly use Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) as the basic building blocks. The Transformer architecture, despite its broad success in other domains, has limited adoption for Earth system forecasting. In this paper, we propose *Earthformer*, a space-time Transformer for Earth system forecasting. Earthformer is based on a generic, flexible and efficient space-time attention block, named *Cuboid Attention*, which decomposes the data to cuboids and applies cuboid-level self-attention in parallel. These cuboids are further connected with a collection of global vectors. We conduct experiments on the MovingMNIST dataset and a newly proposed chaotic N-body MNIST dataset to verify the effectiveness of cuboid attention and figure out the best design for Earthformer. Experiments on two real-world benchmarks about precipitation nowcasting and El Niño/Southern Oscillation (ENSO) forecasting show Earthformer achieves state-of-the-art performance.

1 Introduction

Earth is a complex system. Variabilities of the Earth system, ranging from regular events like temperature fluctuation to extreme events like drought, hail storm, and El Niño/Southern Oscillation (ENSO), impact our daily life. Among all the consequences, Earth system variabilities can influence crop yields, delay airlines, cause floods and forest fires. Precise and timely forecasting of these variabilities can help people take necessary precautions to avoid crisis, or better utilize natural resources such as wind and solar energy. Thus, improving forecasting models for Earth variabilities (e.g. weather and climate) has a huge socioeconomic impact. Despite its importance, the operational weather and climate forecasting systems have not fundamentally changed for almost 50 years [30]. These operational models, including the state-of-the-art High Resolution Ensemble Forecast (HREF) rainfall nowcasting model used in National Oceanic and Atmospheric Administration (NOAA) [28], relies on meticulous numerical simulation of complicated physical models based on extensive ground and satellite observations. Such simulation-based systems inevitably fall short in the ability to incorporate signals from newly emerging geophysical observation systems [12], or take advantage of the Petabytes-scale Earth observation data [36].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

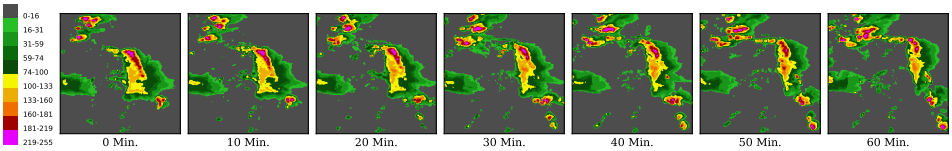


Figure 1: Example VIL observation sequence from the Storm EVent ImagRy (SEVIR) dataset.

As an appealing alternative, deep learning (DL) is offering a new approach for Earth system forecasting [30]. Instead of explicitly incorporating physical rules, DL-based forecasting models are trained from the Earth observation data [31]. By learning from a large amount of observations, the DL model is able to figure out the system's intrinsic physical rules and generate predictions that outperform physics-centric models [9]. Such technique has demonstrated success in several applications, including precipitation nowcasting [28, 6] and ENSO forecasting [15]. Because the Earth system is chaotic [19], high dimensional, and spatiotemporal, designing appropriate DL architecture for modeling the system is particularly challenging. Previous works relied on the combination of Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) [31, 32, 36, 13, 38]. These two architectures impose temporal and spatial inductive biases that help capturing spatiotemporal patterns. However, as a chaotic system, variabilities of the Earth system, such as rainfall and ENSO, are highly sensitive to the system's initial condition and can respond abruptly to internal changes. It is unclear whether the inductive biases in RNN and CNN will still hold for such complex system.

On the other hand, recent years have witnessed major breakthroughs in DL led by the wide adoption of Transformer. The model was originally proposed for natural language processing [35, 7], and has been later extended to computer vision [8, 20], multimodal text-image generation [27], graph learning [42], etc. Transformer relies on the attention mechanism to capture data correlations and is powerful at modeling complex and long-range dependencies, both of which appear in Earth systems (See Fig. 1 for an example of Earth observation data). Despite being suitable for the problem, Transformer sees limited adoption in Earth system forecasting. How to design a space-time Transformer that is good at predicting the future of the Earth systems is largely an open problem to the community. Naively applying the Transformer architecture is infeasible because the $O(N^2)$ attention mechanism is too computationally expensive for the high dimensional Earth observation data.

In this paper, we propose *Earthformer*, a space-time Transformer for Earth system forecasting. To better explore the design of space-time attention, we propose *Cuboid Attention*, which is a generic building block for efficient space-time attention. The idea is to decompose the input tensor to non-overlapping cuboids and apply cuboid-level self-attention in parallel. Different types of correlations can be captured via different cuboid decompositions. Since we limit the $O(N^2)$ self-attention inside the local cuboids, the overall complexity is greatly reduced. By stacking multiple cuboid attention layers with different hyperparameters, we are able to subsume several previously proposed video Transformers [18, 21, 4] as special cases, and also come up with new attention patterns that was not studied before. A limitation of this design is the lack of mechanism for the local cuboids to communicate with each other. Thus, we introduce a collection of global vectors that attend to all the local cuboids, thereby gathering the overall status of the system. By attending to the global vectors, the local cuboids can grasp the general dynamics of the system and share information with each other.

To verify the effectiveness of cuboid attention and figure out the best design under the Earth system forecasting scenario, we conducted extensive experiments on two synthetic datasets: the MovingMNIST [31] dataset and a newly proposed N-body MNIST dataset. Digits in the N-body MNIST follow the chaotic 3-body motion pattern [22], which makes the dataset not only more challenging than MovingMNIST but more relevant to Earth system forecasting. The synthetic experiments reveal the following findings: 1) stacking cuboid attention layers with the Axial attention pattern is both efficient and effective, achieving the best overall performance, 2) adding global vectors provides consistent performance gain without increasing the computational cost, 3) adding hierarchy in the encoder-decoder architecture can improve performance. Based on these findings, we figured out the optimal design for Earthformer and made comparison with other baselines on the SEVIR [36] benchmark for precipitation nowcasting and the ICAR-ENSO dataset [15] for ENSO forecasting. Experiments show that Earthformer achieves the state-of-the-art (SOTA) performance on both tasks.

2 Related Work

Space-time Transformers for video modeling. Inspired by the success of ViT [8] for image classification, space-time Transformer is adopted for improved video understanding. In order to

bypass the huge memory consumption brought by joint spatiotemporal attention, several pioneering work propose efficient alternatives, such as divided attention [4], axial attention [18, 4], factorized encoder [23, 2] and separable attention [44]. Beyond minimal adaptation from ViT, some recent work introduce more vision prior to the design of space-time transformers, including trajectory [25], multi-scale [21, 11] and multi-view [40]. However, no prior work focuses on exploring the design of space-time Transformers for Earth system forecasting.

Deep learning architectures for Earth system forecasting. Conventional DL models for Earth system forecasting are based on CNN and RNN. U-Net with either 2D CNN or 3D CNN have been used for precipitation nowcasting [36], Seasonal Arctic Sea ice prediction [1], and ENSO forecasting [15]. Shi et al. [31] proposed the ConvLSTM network that combines CNN and LSTM for precipitation nowcasting. Wang et al. [38] proposed PredRNN which adds the spatiotemporal memory flow structure to ConvLSTM. To better learn long-term high-level relations, Wang et al. [37] proposed E3D-LSTM that integrates 3D CNN to LSTM. To disentangle PDE dynamics from unknown complementary information, PhyDNet [13] incorporates a new recurrent physical cell to perform PDE-constrained prediction in latent space. Espeholt et al. [9] proposed MetNet-2 that outperforms HREF for forecasting precipitation. The architecture is based on ConvLSTM and dilated CNN. Very recently, there are works that tried to apply Transformer for solving Earth system forecasting problems. Pathak et al. [24] proposed the FourCastNet for global weather forecasting, which is based on Adaptive Fourier Neural Operators (AFNO) [14]. Bai et al. [3] proposed Rainformer for precipitation nowcasting, which is based on an architecture that combines CNN and Swin-Transformer [20]. In the experiments, we can see that Earthformer outperforms Rainformer.

Global and local attention in vision Transformers. To make self-attention more efficient in terms of both memory consumption and speed, recent works have adapted the essence of CNN to perform local attention in transformers [16, 42]. HaloNets [34] develops a new self-attention model family that are simple local self-attention and convolutional hybrids, which outperform both CNN and vanilla ViT on a range of downstream vision tasks. GLiT [5] introduces a locality module and use neural architecture search to find an efficient backbone. Focal transformer [41] proposes focal self-attention that can incorporate both fine-grained local and coarse-grained global interactions. However, these architectures are not directly applicable to spatiotemporal forecasting. Besides, they are also different from our design because we keep K global vectors to summarize the statistics of the dynamic system and connect the local cuboids; experiments show that such a global vector design is crucial for successful spatiotemporal forecasting.

3 Model

Similar to previous works [31, 36, 3], we formulate Earth system forecasting as a spatiotemporal sequence forecasting problem. The Earth observation data, such as radar echo maps from NEXRAD [17] and climate data from CIMP6 [10], are represented as a spatiotemporal sequence $[\mathcal{X}_i]_{i=1}^{T}$, $\mathcal{X}_i \in \mathbb{R}^{H \times W \times C_{\text{in}}}$. Based on these observations, the model predicts the K-step-ahead future $[\mathcal{Y}_{T+i}]_{i=1}^{K}$, $\mathcal{Y}_{T+i} \in \mathbb{R}^{H \times W \times C_{\text{out}}}$. Here, H, W denote the spatial resolution, and $C_{\text{in}}, C_{\text{out}}$ denotes the number of measurements available at each space-time coordinate from the input and the target sequence, respectively. As illustrated in Fig. 2, our proposed *Earthformer* is a hierarchical Transformer encoder-decoder based on *Cuboid Attention*. The input observations are encoded as a hierarchy of hidden states and then decoded to the prediction target. In the following, we present the detailed design of cuboid attention and the hierarchical encoder-decoder architecture in Earthformer.

3.1 Cuboid Attention

Compared with images and text, spatiotemporal data in Earth systems usually have higher dimensionality. As a consequence, applying Transformers to this task is challenging. For example, for a 3D tensor with shape (T, H, W), the complexity of the vanilla self-attention is $O(T^2H^2W^2)$ and can be computationally infeasible. Previous literature proposed various structure-aware space-time attention layers to reduce the complexity [18, 21, 4]. These space-time attention layers share the common design of stacking multiple elementary attention cells that focus on different types of data correlations (e.g., temporal correlation and spatial correlation). Steming from this observation, we propose the generic cuboid attention layer that involves three steps: "decompose", "attend", and "merge".

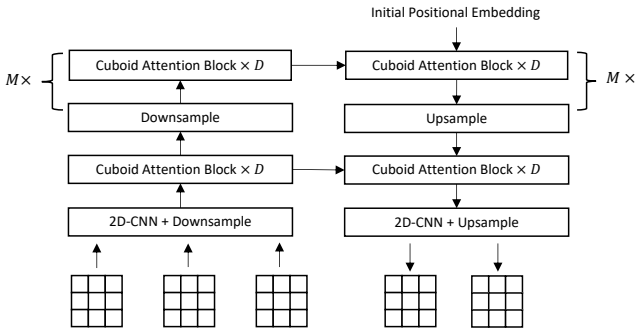


Figure 2: Illustration of the Earthformer architecture. It is a hierarchical Transformer encoder-decoder based on cuboid attention. The input sequence has length 3 and the target sequence has length 2. "$\times D$" means to stack D cuboid attention blocks with residual connection. "$M\times$" means to have M layers of hierarchies.

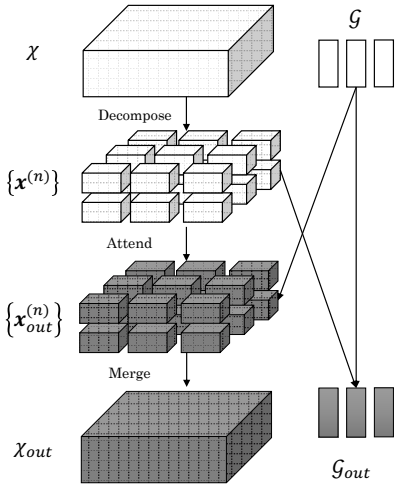
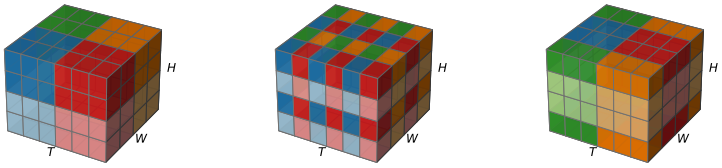


Figure 3: Illustration of the cuboid attention layer with global vectors.



(a) *strategy*="`local`" *shift*=(0, 0, 0) (b) *strategy*="`dilated`" *shift*=(0, 0, 0) (c) *strategy*="`local`" $shift = (0, 1, 1)$

Figure 4: Illustration of cuboid decomposition strategies when the input shape is $(T, H, W) = (6, 4, 4)$, and cuboid size $(b_T, b_H, b_W) = (3, 2, 2)$. Cells that have the same color belong to the same cuboid and will attend to each other. (Best viewed in color).

Decompose. We first decompose the input spatiotemporal tensor $\mathcal{X} \in \mathbb{R}^{T\times H\times W\times C}$ into a sequence of cuboids $\{\boldsymbol{x}^{(n)}\}$.

$$\{\boldsymbol{x}^{(n)}\} = \texttt{Decompose}(\mathcal{X}, cuboid\_size, strategy, shift), \tag{1}$$

where $cuboid\_size = (b_T, b_H, b_W)$ is the size of the local cuboid, $strategy \in \{$"`local`", "`dilated`"$\}$ controls whether to adopt the local decomposition strategy or the dilated decomposition strategy [4], $shift = (s_T, s_H, s_W)$ is the window shift offset [20]. There are a total number of $\lceil \frac{T}{b_T} \rceil \lceil \frac{H}{b_H} \rceil \lceil \frac{W}{b_W} \rceil$ cuboids in $\{\boldsymbol{x}^{(n)}\}$. To simplify the notation, we assume that T, H, W are divisible by b_T, b_H, b_W. In the implementation, we pad the input tensor if it is not divisible.

Assume $\boldsymbol{x}^{(n)}$ is the (n_T, n_H, n_W)-th cuboid in $\{\boldsymbol{x}^{(n)}\}$. The index $[i, j, k]$ of $\boldsymbol{x}^{(n)}$ can be mapped to the index $[i', j', k']$ of $\mathcal{X}$ via Eqn. 2 if the strategy is "local" and Eqn. 3 if the strategy is "dilated".

$$\begin{aligned} i' &\leftrightarrow s_T + b_T(n_T - 1) + i \mod T \\ j' &\leftrightarrow s_H + b_H(n_H - 1) + j \mod H \\ k' &\leftrightarrow s_W + b_W(n_W - 1) + k \mod W \end{aligned} \tag{2}$$

$$\begin{aligned} i' &\leftrightarrow s_T + b_T(i - 1) + n_T \mod T \\ j' &\leftrightarrow s_H + b_H(j - 1) + n_H \mod H \\ k' &\leftrightarrow s_W + b_W(k - 1) + n_W \mod W \end{aligned} \tag{3}$$

Since the mapping is bijective, one can then map indices from $\mathcal{X}$ to $\{\boldsymbol{x}^{(n)}\}$ via the inverse operation. Fig. 4 provides three examples showing how an input tensor will be decomposed following different hyperparameters of $\texttt{Decompose}(\cdot)$.

Attend. After decomposing the input tensor into a set of non-overlapping cuboids, we apply self-attention within each cuboid in parallel:

$$\boldsymbol{x}^{(n)}_{\text{out}} = \texttt{Attention}_\theta(\boldsymbol{x}^{(n)}), 1 \le n \le N. \tag{4}$$

The self-attention parameter θ are shared across all cuboids. The overall complexity of the $\texttt{Attention}_\theta(\cdot)$ step is $O\left(\lceil \frac{T}{b_T} \rceil \lceil \frac{H}{b_H} \rceil \lceil \frac{W}{b_W} \rceil (b_T b_H b_W)^2\right) \approx O(THW \cdot b_T b_H b_W)$, which scales

Table 1: Configurations of the cuboid attention patterns explored in the paper. The input tensor has shape (T, H, W). If the window shift or the Local/Dilated key is not given, we use $(\texttt{0}, \texttt{0}, \texttt{0})$ and "`local`" strategy, respectively, by default. When stacking multiple cuboid attention layers, each cuboid attention layer will be coupled with layer normalization layers and feed-forward network as in the Pre-LN Transformer [39]. The first row shows the configuration of the generic cuboid attention.

Name	Keys	Configuration Values
Generic Cuboid Attention	Cub. Size Window Shift Local/Dilated	$(\texttt{T}_1, \texttt{H}_1, \texttt{W}_1) \rightarrow (\texttt{T}_2, \texttt{H}_2, \texttt{W}_2) \rightarrow \cdots \rightarrow (\texttt{T}_\texttt{L}, \texttt{H}_\texttt{L}, \texttt{W}_\texttt{L})$ $(\texttt{P}_1, \texttt{M}_1, \texttt{M}_1) \rightarrow (\texttt{P}_2, \texttt{M}_2, \texttt{M}_2) \rightarrow \cdots \rightarrow (\texttt{P}_\texttt{L}, \texttt{M}_\texttt{L}, \texttt{M}_\texttt{L})$ "`loc./dil.`" $\rightarrow$ "`loc./dil.`" $\rightarrow \cdots \rightarrow$ "`loc./dil.`"
Axial	Cub. Size	$(\texttt{T}, \texttt{1}, \texttt{1}) \rightarrow (\texttt{1}, \texttt{H}, \texttt{1}) \rightarrow (\texttt{1}, \texttt{1}, \texttt{W})$
Divided Space-Time	Cub. Size	$(\texttt{T}, \texttt{1}, \texttt{1}) \rightarrow (\texttt{1}, \texttt{H}, \texttt{W})$
Video-Swin $P \times M$	Cub. Size Window Shift	$(\texttt{P}, \texttt{M}, \texttt{M}) \rightarrow (\texttt{ P , M , M })$ $(\texttt{0}, \texttt{0}, \texttt{0}) \rightarrow (\texttt{P}/\texttt{2}, \texttt{M}/\texttt{2}, \texttt{M}/\texttt{2})$
Spatial Local-Dilate-M	Cub. Size Local/Dilated	$(\texttt{T}, \texttt{1}, \texttt{1}) \rightarrow (\texttt{1}, \texttt{M}, \texttt{M}) \rightarrow (\texttt{1}, \texttt{M}, \texttt{M})$ "`local`" $\rightarrow$ "`local`" $\rightarrow$ "`dilated`"
Axial Space Dilate-M	Cub. Size Local/Dilated	$(\texttt{T}, \texttt{1}, \texttt{1}) \rightarrow (\texttt{1}, \texttt{H}/\texttt{M}, \texttt{1}) \rightarrow (\texttt{1}, \texttt{H}/\texttt{M}, \texttt{1}) \rightarrow (\texttt{1}, \texttt{1}, \texttt{W}/\texttt{M}) \rightarrow (\texttt{1}, \texttt{1}, \texttt{W}/\texttt{M})$ "`local`" $\rightarrow$ "`dilated`" $\rightarrow$ "`local`" $\rightarrow$ "`dilated`" $\rightarrow$ "`local`"

linearly to the cuboid size. Since the cuboid size can be much smaller than the size of the input tensor, the layer is more efficient than full attention.

Merge. $\texttt{Merge}(\cdot)$ is the inverse operation of $\texttt{Decompose}(\cdot)$. The set of cuboids obtained after the attention step $\{\boldsymbol{x}_{\text{out}}^{(n)}\}$ are merged back to the original input shape to produce the final output of cuboid attention, as shown in Eqn. 5. The mapping follows the same bijections in Eqn. 2 and Eqn. 3.

$$\mathcal{X}_{\text{out}} = \texttt{Merge}(\{\boldsymbol{x}_{\text{out}}^{(n)}\}_n, cuboid\_size, strategy, shift). \tag{5}$$

Explore cuboid attention patterns. By stacking multiple cuboid attention layers with different choices of $cuboid\_size$, $strategy$ and $shift$, we are able to efficiently explore existing and potentially more effective space-time attention. In this paper, we explore the cuboid attention patterns as listed in Table 1. From the table, we can see that cuboid attention subsumes previously proposed space-time attention methods like axial attention, video swin-Transformer, and divided space-time attention. Also, we manually picked the patterns that are reasonable and not computationally expensive as our search space. The flexibility of cuboid attention allows us to conduct Neural Architecture Search (NAS) to automatically pick a pattern but we will leave it as future work.

3.2 Global Vectors

One limitation of the previous formulation is that the cuboids do not communicate with each other. This is sub-optimal because each cuboid is not capable of understanding the global dynamics of the system. Thus, inspired by the CLS token adopted in BERT [7, 43], we propose to introduce a collection of P global vectors $\mathcal{G} \in \mathbb{R}^{P \times C}$ to help cuboids scatter and gather crucial global information. When each cuboid is performing the self-attention, the elements will not only attend to the other elements within the same cuboid but attend to the global vectors $\mathcal{G}$. We revise Eqn. 4 to Eqn. 6 to enable local-global information exchange. We also use Eqn. 7 to update the global vectors $\mathcal{G}$ by aggregating the information from all local vectors.

$$\boldsymbol{x}_{\text{out}}^{(n)} = \texttt{Attention}_\theta(\boldsymbol{x}^{(n)}, \mathcal{G}), 1 \leq n \leq N. \tag{6}$$

$$\mathcal{G}_{\text{out}} = \texttt{Attention}_\phi(\{\boldsymbol{x}^{(n)}\}_n, \mathcal{G}). \tag{7}$$

The additional complexity caused by the global vectors is approximately $O\left(THW \cdot P + P^2\right)$. Given that P is usually small (in our experiments, P is at most 8), computational overhead induced by the global structure is negligible. The overall cuboid attention layer is illustrated in Fig. 3.

3.3 Hierarchical Encoder-Decoder Architecture

Earthformer adopts a hierarchical encoder-decoder architecture illustrated in Fig. 2. Each cuboid attention block in the encoder uses one of the patterns described in Table 1. Each block are repeated

Table 2: Statistics of the datasets used in the experiments.

Dataset	Size			Seq. Len.		Spatial Resolution
	train	val	test	in	out	$H \times W$
MovingMNIST	8,100	900	1,000	10	10	64×64
N-body MNIST	20,000	1,000	1,000	10	10	64×64
SEVIR	35,718	9,060	12,159	13	12	384×384
ICAR-ENSO	5,205	334	1,667	12	26	24×48

for D times. The cuboid blocks in the decoder adopt the "Axial" pattern. The hierarchical architecture gradually encodes the input sequence to multiple levels of representations and generates the prediction via a coarse-to-fine procedure. To reduce the spatial resolution of the input for cuboid attention layers, we include a pair of initial downsampling and upsampling modules that consist of stacked 2D-CNN and Nearest Neighbor Interpolation (NNI) layers. Different from other papers that adopt Transformer for video prediction [18, 26], we generate the predictions in a non-auto-regressive fashion rather than an auto-regressive patch-by-patch fashion. This means that our decoder directly generates the predictions from the initial learned positional embeddings. We also conducted experiments with an auto-regressive decoder based on visual codebook [29]. However, the auto-regressive decoder underperforms the non-auto-regressive decoder in terms of forecasting skill scores. The comparison between non-auto-regressive decoder and auto-regressive decoder are shown in the Appendix.

4 Experiments

We first conducted experiments on two synthetic datasets, MovingMNIST and a newly proposed N-body MNIST, to verify the effectiveness of Earthformer and conduct ablation study on our design choices. Results on these two datasets lead to the following findings: 1) Among all patterns listed in Table 1, "Axial" achieves the best overall performance; 2) Global vectors bring consistent performance gain with negligible increase in computational cost; 3) Using a hierarchical coarse-to-fine structure can boost the performance. Based on these findings, we figured out the optimal design of Earthformer and compared it with other state-of-the-art models on two real-world datasets: SEVIR [36] and ICAR-ENSO[1]. On both datasets, Earthformer achieved the best overall performance. The statistics of all the datasets used in the experiments are shown in Table 2. We normalize the data to the range $[0, 1]$ and trained all models with the Mean-Squared Error (MSE) loss. More implementation details are shown in the Appendix.

4.1 Experiments on Synthetic Datasets

MovingMNIST. We follow [33] to use the public MovingMNIST dataset[2]. The dataset contains 10,000 sequences. Each sequence shows 2 digits moving inside a 64×64 frame. We split the dataset to use 8,100 samples for training, 900 samples for validation and 1,000 samples for testing. The task is to predict the future 10 frames for each sequence conditioned on the first 10 frames.

N-body MNIST. Earth is a complex system where an extremely large number of variables interact with each other. Compared with the Earth system, the dynamics of the synthetic MovingMNIST dataset, in which the digits move independently with constant speed, is over-simplified. Thus, achieving good performance on MovingMNIST does not imply that the model is capable of modeling complex interactions in Earth system. On the other hand, the real-world Earth observation data, though have experienced rapid development, are still noisy and may not provide useful insights for model development. Therefore, we extend MovingMNIST to N-body MNIST, where N digits are moving inside a 64×64 frame. Each digit has its mass and is subjected to the gravity from other digits. We choose $N = 3$ in the experiments so that the digits will follow the chaotic 3-body motion [22]. The highly non-linear interactions in N-body MNIST makes it much more challenging than the original MovingMNIST. We generate 20,000 sequences for training, 1,000 for validation and 1,000 for test. Perceptual examples of the dataset can be found at the first two rows of Fig. 5.

Hierarchical v.s. non-hierarchical. We choose "Axial" without global vectors as our cuboid attention pattern and compare the performance of non-hierarchical and hierarchical architectures on

[1]Dataset available at `https://tianchi.aliyun.com/dataset/dataDetail?dataId=98942`

[2]MovingMNIST: `https://github.com/tychovdo/MovingMNIST`

Table 3: Ablation study of different cuboid attention patterns and the effect of global vectors on MovingMNIST and N-body MNIST. The variant that achieved the best performance is in bold-case while the second best is underscored. We also compared the performance of the cuboid attention patterns with and without global vectors and highlight the better one with grey background.

Model	#Param. (M)	GFLOPS	Metrics on MovingMNIST			Metrics on N-Body		
			MSE ↓	MAE ↓	SSIM ↑	MSE ↓	MAE ↓	SSIM ↑
Axial	6.61	33.7	46.91	101.5	0.8825	15.89	41.38	0.9510
+ global ★	7.61	34.0	**41.79**	**92.78**	**0.8961**	**14.82**	**39.93**	**0.9538**
DST	5.70	35.2	57.43	118.6	0.8623	18.24	45.88	0.9435
+ global	6.37	35.5	52.92	108.3	0.8760	17.77	45.84	0.9433
Video Swin 2x8	5.66	31.1	54.45	111.7	0.8715	19.89	49.02	0.9374
+ global	6.33	31.4	52.70	108.5	0.8766	19.53	48.43	0.9389
Video Swin 10x8	5.89	39.2	63.34	125.3	0.8525	23.35	53.17	0.9274
+ global	6.56	39.4	62.15	123.4	0.8541	22.81	52.94	0.9293
Spatial Local-Global 2	6.61	33.3	59.88	122.1	0.8572	23.24	54.63	0.9263
+ global	7.61	33.7	59.42	122.9	0.8565	21.88	52.49	0.9305
Spatial Local-Global 4	6.61	33.5	58.72	118.5	0.8600	21.02	49.82	0.9344
+ global	7.61	33.9	54.84	115.5	0.8585	19.82	48.12	0.9371
Axial Space Dilate 2	8.59	41.8	50.11	104.4	0.8814	15.97	42.19	0.9494
+ global	10.30	42.4	46.86	98.95	0.8884	15.73	41.85	0.9510
Axial Space Dilate 4	8.59	41.6	47.40	99.31	0.8865	19.49	51.04	0.9352
+ global	10.30	42.2	45.11	95.98	0.8928	17.91	46.35	0.9440

Table 4: Comparison of Earthformer with baselines on MovingMNIST and N-body MNIST.

Model	#Param. (M)	GFLOPS	MovingMNIST			N-body MNIST		
			MSE ↓	MAE ↓	SSIM ↑	MSE ↓	MAE ↓	SSIM ↑
UNet [36]	16.6	0.9	110.4	249.4	0.6170	38.90	94.29	0.8260
ConvLSTM [31]	14.0	30.1	62.04	126.9	0.8477	32.15	72.64	0.8886
PredRNN [38]	23.8	232.0	52.07	108.9	0.8831	21.76	54.32	0.9288
PhyDNet [13]	3.1	15.3	58.70	124.1	0.8350	28.97	78.66	0.8206
E3D-LSTM [37]	12.9	302.0	55.31	101.6	0.8821	22.98	62.52	0.9131
Rainformer [3]	19.2	1.2	85.83	189.2	0.7301	38.89	96.47	0.8036
Earthformer w/o global	6.6	33.7	46.91	101.5	0.8825	15.89	41.38	0.9510
Earthformer	7.6	34.0	**41.79**	**92.78**	**0.8961**	**14.82**	**39.93**	**0.9538**

MovingMNIST. The results are shown in the Appendix. We can find that a hierarchical structure has fewer FLOPS and also performs better. We thus use a hierarchical structure in all other experiments.

Cuboid pattern search. The design of cuboid attention greatly facilitates the search for optimal space-time attention. We compare the patterns listed in Table 1 on both MovingMNIST and N-body MNIST to investigate the effectiveness and efficiency of different space-time attention on spatiotemporal forecasting tasks. Besides the previously proposed space-time attention methods, we also include new configurations that are reasonable and not computationally expensive in our search space. For each pattern, we also compare the variant that uses global vectors. Results are summarized in Table 3. We find that the "Axial" pattern is both effective and efficient and adding global vectors improves performance for all patterns while having similar FLOPS. We thus pick "Axial + global" as the pattern in Earthformer when conducting experiments on real-world datasets.

Comparison to the state of the art. We evaluate six spatiotemporal forecasting algorithms: UNet [36], ConvLSTM [31], PredRNN [38], PhyDNet [13], E3D-LSTM [37] and Rainformer [3]. The results are in Table 4. Note that the MovingMNIST performance on several papers [13] are obtained by training the model with on-the-fly generated digits while we pre-generate the digits and train all models on a fixed dataset. Comparing the numbers in the table with numbers shown in these papers are not fair. We train all baselines from scratch on both MovingMNIST and N-body MNIST using the default hyperparameters and configurations in their officially released code[3].

Qualitative results on N-body MNIST. Fig. 5 shows the generation results of different methods on a sample sequence from N-body MNIST test set. The qualitative example demonstrates that our Earthformer is capable of learning the long-range interactions among digits and correctly predicting

[3]Except for Rainformer which originally has 212M parameters and thus suffers overfitting severely.

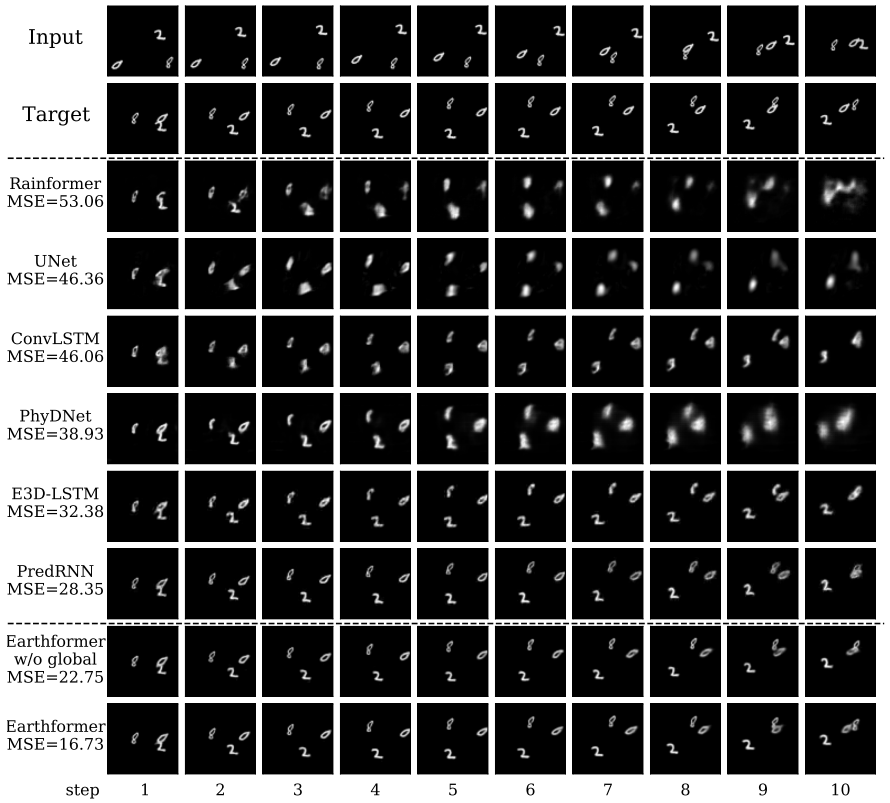


Figure 5: A set of examples showing the perceptual quality of the predictions on the N-body MNIST test set. From top to bottom: input frames, target frames, predictions by Rainformer [3], UNet [36], ConvLSTM [31], PhyDNet [13], E3D-LSTM [37], PredRNN [38], Earthformer without using global vectors, Earthformer. The results are sorted according to the MSE.

their future motion trajectories. Also, we can see that Earthformer is able to more accurately predict the position of the digits with the help of global vectors. On the contrary, none of the baseline algorithms that achieved solid performance on MovingMNIST gives the correct and precise position of the digit "0" in the last frame. They either predict incorrect motion trajectories (PredRNN and E3D-LSTM), or generate highly blurry predictions (Rainformer, UNet and PhyDNet) to accommodate the uncertainty about the future.

4.2 SEVIR Precipitation Nowcasting

Storm EVent ImagRy (SEVIR) [36] is a spatiotemporally aligned dataset containing over 10,000 weather events. Each event consists of 384 km $\times$ 384 km image sequences spanning 4 hours of time. Images in SEVIR were sampled and aligned across five different data types: three channels (C02, C09, C13) from the GOES-16 advanced baseline imager, NEXRAD Vertically Integrated Liquid (VIL) mosaics, and GOES-16 Geostationary Lightning Mapper (GLM) flashes. SEVIR benchmark supports scientific research on multiple meteorological applications including precipitation nowcasting, synthetic radar generation, front detection, etc. We adopt SEVIR for benchmarking precipitation nowcasting, i.e., to predict the future VIL up to 60 minutes (12 frames) given 65 minutes context VIL (13 frames). Fig. 1 shows an example of VIL observation sequences in SEVIR.

Besides MSE, we also include the Critical Success Index (CSI), which is commonly used in precipitation nowcasting and is defined as $\texttt{CSI} = \frac{\#\texttt{Hits}}{\#\texttt{Hits}+\#\texttt{Misses}+\#\texttt{F.Alarms}}$. To count the $\#$Hits (truth=1, pred=1), $\#$Misses (truth=1, pred=0) and $\#$F.Alarms (truth=0, pred=1), the prediction and the ground-truth are rescaled back to the range 0-255 and binarized at thresholds $[16, 74, 133, 160, 181, 219]$. We report CSI at different thresholds and also their mean CSI-M.

SEVIR is much larger than MovingMNIST and N-body MNIST and has higher resolution. We thus slightly adjust the configurations of baselines based on those for MovingMNIST for fair comparison. Detailed configurations are shown in the Appendix. The experiment results are listed in Table 5. Earthformer consistently outperforms baselines on almost all metrics and brings significant performance gain especially at high thresholds like CSI-219, which are more valued by the communities.

Table 5: Performance comparison on SEVIR. We include `Critical Success Index` (`CSI`) besides MSE as evaluation metrics. The `CSI`, a.k.a intersection over union (IOU), is calculated at different precipitation thresholds and denoted as `CSI-`$thresh$.

Model	#Param. (M)	GFLOPS	Metrics							
			CSI-M ↑	CSI-219 ↑	CSI-181 ↑	CSI-160 ↑	CSI-133 ↑	CSI-74 ↑	CSI-16 ↑	MSE (10^{-3}) ↓
Persistence	-	-	0.2613	0.0526	0.0969	0.1278	0.2155	0.4705	0.6047	11.5338
UNet [36]	16.6	33	0.3593	0.0577	0.1580	0.2157	0.3274	0.6531	0.7441	4.1119
ConvLSTM [31]	14.0	527	0.4185	0.1288	0.2482	0.2928	0.4052	0.6793	0.7569	3.7532
PredRNN [38]	46.6	328	0.4080	0.1312	0.2324	0.2767	0.3858	0.6713	0.7507	3.9014
PhyDNet [13]	13.7	701	0.3940	0.1288	0.2309	0.2708	0.3720	0.6556	0.7059	4.8165
E3D-LSTM [37]	35.6	523	0.4038	0.1239	0.2270	0.2675	0.3825	0.6645	0.7573	4.1702
Rainformer [3]	184.0	170	0.3661	0.0831	0.1670	0.2167	0.3438	0.6585	0.7277	4.0272
Earthformer w/o global	13.1	257	0.4356	0.1572	0.2716	0.3138	0.4214	0.6859	**0.7637**	3.7002
Earthformer	15.1	257	**0.4419**	**0.1791**	**0.2848**	**0.3232**	**0.4271**	**0.6860**	0.7513	**3.6957**

Table 6: Performance comparison on ICAR-ENSO. C-`Nino3.4-M` and C-`Nino3.4-WM` are the mean and the weighted mean of the correlation skill $\boldsymbol{C}^{\texttt{Nino3.4}}$ over $K = 12$ forecasting steps. C-`Nino3.4-WM` assigns more weights to longer-term prediction scores. MSE is calculated between the spatiotemporal SST anomalies prediction and the corresponding ground-truth.

Model	#Param. (M)	GFLOPS	Metrics		
			C-Nino3.4-M ↑	C-Nino3.4-WM ↑	MSE (10^{-4}) ↓
Persistence	-	-	0.3221	0.447	4.581
UNet [36]	12.1	0.4	0.6926	2.102	2.868
ConvLSTM [31]	14.0	11.1	0.6955	2.107	2.657
PredRNN [38]	23.8	85.8	0.6492	1.910	3.044
PhyDNet [13]	3.1	5.7	0.6646	1.965	2.708
E3D-LSTM [37]	12.9	99.8	0.7040	2.125	3.095
Rainformer [3]	19.2	1.3	0.7106	2.153	3.043
Earthformer w/o global	6.6	23.6	0.7239	2.214	2.550
Earthformer	7.6	23.9	**0.7329**	**2.259**	**2.546**

4.3 ICAR-ENSO Sea Surface Temperature Anomalies Forecasting

El Niño/Southern Oscillation (ENSO) has a wide range of associations with regional climate extremes and ecosystem impacts. ENSO sea surface temperature (SST) anomalies forecasting for lead times up to one year (12 steps) is a valuable and challenging problem. Nino3.4 index, which is the area-averaged SST anomalies across a certain area (170°-120°W, 5°S-5°N) of the Pacific, serves as a crucial indicator of this climate event. The forecast quality is evaluated by the correlation skill [15] of the three-month-moving-averaged Nino3.4 index $\boldsymbol{C}^{\texttt{Nino3.4}} = \frac{\sum_N (\boldsymbol{X}-\bar{\boldsymbol{X}})(\boldsymbol{Y}-\bar{\boldsymbol{Y}})}{\sqrt{\sum_N (\boldsymbol{X}-\bar{\boldsymbol{X}})^2 \sum_N (\boldsymbol{Y}-\bar{\boldsymbol{Y}})^2}} \in \mathbb{R}^K$ calculated on the whole test set of size N, where $\boldsymbol{Y} \in \mathbb{R}^{N\times K}$ is the ground-truth of K-step Nino3.4 index, $\boldsymbol{X} \in \mathbb{R}^{N\times K}$ is the corresponding prediction of Nino3.4 index.

ICAR-ENSO consists of historical climate observation and stimulation data provided by Institute for Climate and Application Research (ICAR). We forecast the SST anomalies up to 14 steps (2 steps more than one year for calculating three-month-moving-average) given context 12 steps SST anomalies observations. Table 6 compares the performance of our Earthformer with baselines on ICAR-ENSO dataset. We report the mean correlation skill C-`Nino3.4-M` $= \frac{1}{K}\sum_k C_k^{\texttt{Nino3.4}}$ and the weighted mean correlation skill C-`Nino3.4-WM` $= \frac{1}{K}\sum_k a_k \cdot C_k^{\texttt{Nino3.4}}$ over $K = 12$ forecasting steps[4], as well as the MSE between the spatiotemporal SST anomalies prediction and the corresponding ground-truth. We can find that Earthformer consistently outperforms baselines in all concerned evaluation metrics and that using global vectors further improves the performance.

5 Conclusion and Future Work

In this paper, we propose Earthformer, a space-time Transformer for Earth system forecasting. Earthformer is based on a generic and efficient building block called *Cuboid Attention*. It achieves SOTA on MovingMNIST, our newly proposed N-body MNIST, SEVIR, and ICAR-ENSO. For future works, we plan to extend Earthformer with NAS and GAN to further improve the performance and apply it on more Earth system forecasting problems.

[4] $a_k = b_k \cdot \ln k$, where $b_k = 1.5$, for $k \leq 4$; $b_k = 2$, for $4 < k \leq 11$; $b_k = 3$, for $k > 11$.

References

[1] Tom R Andersson, J Scott Hosking, María Pérez-Ortiz, Brooks Paige, Andrew Elliott, Chris Russell, Stephen Law, Daniel C Jones, Jeremy Wilkinson, Tony Phillips, et al. Seasonal arctic sea ice forecasting with probabilistic deep learning. *Nature communications*, 12(1):1–12, 2021.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Luvcić, and Cordelia Schmid. Vivit: A video vision transformer. In *International Conference on Computer Vision (ICCV)*, 2021.

[3] Cong Bai, Feng Sun, Jinglin Zhang, Yi Song, and Shengyong Chen. Rainformer: Features extraction balanced network for radar-based precipitation nowcasting. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095*, 2(3):4, 2021.

[5] Boyu Chen, Peixia Li, Chuming Li, Baopu Li, Lei Bai, Chen Lin, Ming Sun, Junjie Yan, and Wanli Ouyang. Glit: Neural architecture search for global and local image transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–21, 2021.

[6] Christian Schroeder de Witt, Catherine Tong, Valentina Zantedeschi, Daniele De Martini, Freddie Kalaitzis, Matthew Chantry, Duncan Watson-Parris, and Piotr Bilinski. RainBench: towards global precipitation forecasting from satellite imagery. In *AAAI*, 2021.

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

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

[9] Lasse Espeholt, Shreya Agrawal, Casper Sønderby, Manoj Kumar, Jonathan Heek, Carla Bromberg, Cenk Gazen, Jason Hickey, Aaron Bell, and Nal Kalchbrenner. Skillful twelve hour precipitation forecasts using large context neural networks. *arXiv preprint arXiv:2111.07470*, 2021.

[10] Veronika Eyring, Sandrine Bony, Gerald A Meehl, Catherine A Senior, Bjorn Stevens, Ronald J Stouffer, and Karl E Taylor. Overview of the coupled model intercomparison project phase 6 (cmip6) experimental design and organization. *Geoscientific Model Development*, 9(5):1937–1958, 2016.

[11] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, 2021.

[12] Steven J Goodman, Timothy J Schmit, Jaime Daniels, and Robert J Redmon. *The GOES-R series: a new generation of geostationary environmental satellites*. Elsevier, 2019.

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

[14] John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

[15] Yoo-Geun Ham, Jeong-Hwan Kim, and Jing-Jia Luo. Deep learning for multi-year ENSO forecasts. *Nature*, 573(7775):568–572, 2019.

[16] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *NeurIPS*, 2021.

[17] William H Heiss, David L McGrew, and Dale Sirmans. NEXRAD: next generation weather radar (wsr-88d). *Microwave Journal*, 33(1):79–89, 1990.

[18] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

[19] Christophe Letellier. *Chaos in nature*, volume 94. World Scientific, 2019.

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[21] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. arxiv. *arXiv preprint arXiv:2106.13230*, 2021.

[22] Christian Marchal. The three-body problem. 2012.

[23] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.

[24] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[25] Mandela Patrick, Dylan Campbell, Yuki M. Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[26] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. *arXiv preprint arXiv:2006.10704*, 2020.

[27] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[28] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

[29] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

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

[31] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, volume 28, 2015.

[32] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *NeurIPS*, volume 30, 2017.

[33] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, pages 843–852. PMLR, 2015.

[34] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *CVPR*, 2021.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, **L**ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017.

[36] Mark Veillette, Siddharth Samsi, and Chris Mattioli. Sevir: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. *Advances in Neural Information Processing Systems*, 33:22009–22019, 2020.

[37] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3D LSTM: A model for video prediction and beyond. In *International conference on learning representations*, 2018.

[38] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip Yu, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[39] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.

[40] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, 2022.

[41] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. In *NeurIPS*, 2021.

[42] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34, 2021.

[43] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

[44] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13577–13587, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Exploring Earth system forecasting has no negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]