A Heterogeneous Spatiotemporal Network for Lightning Prediction

Yangli-ao Geng, Qingyong Li, Tianyang Lin
and Jing Zhang
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University
Beijing, China
Emails: {gengyla,liqy,tyanglin,j_zhang}@bjtu.edu.cn

Liangtao Xu, Wen Yao, Dong Zheng
and Weitao Lyu
State Key Laboratory of Severe Weather
Chinese Academy of Meteorological Sciences
Beijing, China
Email: {xult,yaowen,zhengdong,wtlyu}@cma.gov.cn

Abstract*—Lightning prediction is a complicated and challenging task requiring that meteorologists integrate information from multiple data sources to make decisions. Although some data-driven models have been proposed to make prediction automatically, most of them are based on a single data source or several basically-homogeneous data sources, making them hard to adapt to complex and diverse data in practice. In this work, we propose a heterogeneous spatiotemporal network (HSTN) for lightning prediction, aiming at mining knowledge from several heterogeneous spatiotemporal (ST) data sources. Specifically, HSTN comprises three modules: Gaussian diffusion module, ST encoder and ST decoder. Noting that most of meteorological data can be formatted into either a dense ST tensor or a sparse ST tensor, the ST encoder, with the help of the Gaussian diffusion module, is designed to extract information from both two types of tensors. On the other hand, ST decoder is responsible for merging all information from the other modules and generate the final prediction. By organically combining the three modules, HSTN can handle complex input with heterogeneity in both space and time domains. Besides, we propose a multi-scale pooling loss to deal with the short-sight problem caused by grid-wise losses. We conduct experimental evaluations on a real-world lightning dataset. The results demonstrate that HSTN achieves state-of-the-art performance compared with several established baselines.

***Index Terms*—heterogeneous spatiotemporal data mining, Gaussian diffusion, Bayesian inference, weather prediction**

I. Introduction

Lightning is a naturally occurring electrostatic discharge with the release of massive energy, often accompanied by heavy rainfall, hail, and strong wind [1]. It is a common but dangerous natural phenomenon, posing huge threats to human life, aviation and electrical infrastructures [2]. The great harm of lightning has driven significant interest in the prediction of this natural hazard [3].

Traditional lightning prediction methods can be roughly divided into two major categories: extrapolation based nowcasting and numerical weather prediction (NWP) based forecasting. The extrapolation based nowcasting [4], [5] first identifies thunderstorm area and tracks their movements from the past observations (e.g. satellite and radar images), and then makes a prediction based on the momentum of the thunderstorm track (i.e. extrapolation). Its prediction accuracy relies heavily on the consistency of the thunderstorm development before and after, prohibiting extrapolation based methods to be applied for a long-term (e.g. more than two hours) prediction. In contrast, the NWP based forecasting can simulate long-term weather evolution by solving complex atmospheric equations with the support of powerful computers. Then the lightning prediction is conducted by a series of empirical functions [6], [7] with the simulated atmospheric parameters as input. However, since these methods are fully based on NWP simulations, their performance is bounded by the simulation quality.

In the past ten years, the success of machine learning has inspired some researchers applying data-driven models to weather predictions. Early works [8], [9] usually treat machine learning as a black box, simply exploiting it to replace empirical functions in the prediction process. However, the performance of traditional machine learning models highly relies on the form of input features, and good results usually require appropriate feature engineering. This limits the application of these models to meteorology considering that meteorologists usually face massive heterogeneous noisy data. In recent years, this situation has improved as the development of deep neural networks (DNNs), whose high flexibility and strong modeling capacity rescue users from intricate feature engineering. Many researchers managed to apply DNNs to weather predictions [10]–[13]. Among these works, Wang et al. [12] and Geng et al. [13] both propose to predict weather via DNN models combining historical observations and NWP simulations, which are meaningful attempts to enhance the weather prediction by fusing dual data sources with heterogeneity in time.

Back to the lightning prediction, we still need to overcome three difficulties. Firstly, given the complex and dynamic nature of thunderstorms [1], the performance of nowcasting methods based on past observations usually drop for a long-period prediction. Secondly, as a basic tool, the NWP system usually produces simulations with deviations in both space and time domains, which introduces irreparable biases to the prediction methods based on them [13]. Thirdly, despite the rich information sources for the lightning prediction, valuable knowledge is scattered among massive heterogeneous spatiotemporal (ST) data (an example is shown in Figure 1), which is hard to extract to enhance the prediction quality.

To meet the above challenges, we propose a heteroge-

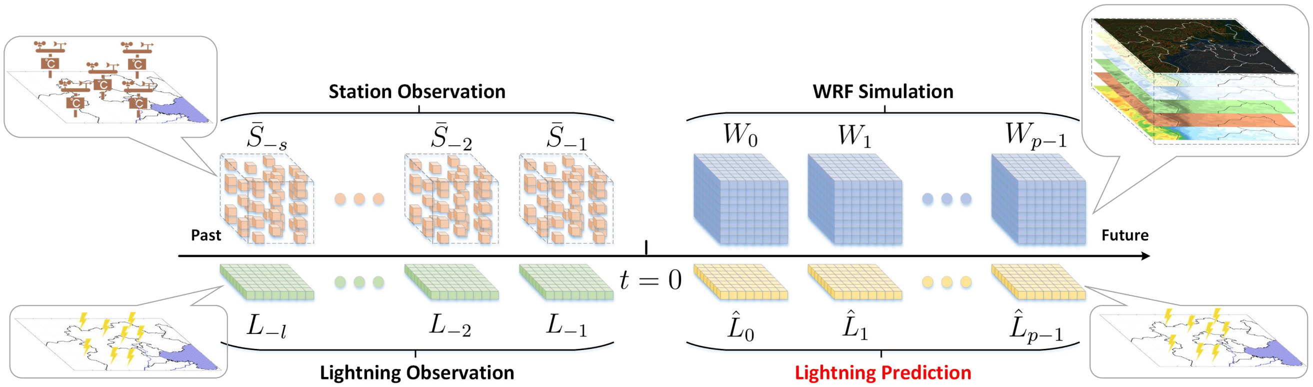


Fig. 1: The heterogeneous spatiotemporal structure of data in our task. On one hand, the (station and lightning) observation data and the (WRF) simulation data describe weather situations for different periods (the past and the future). On the other hand, the sparse structure of the station observation distinguishes it from the other data. We seek to extract information from these heterogeneous data and produce a lightning prediction.

neous spatiotemporal network (HSTN) for lightning predictions. Unlike previous prediction models learning from one data source or several basically-homegeneous data sources, HSTN aims at mining knowledge from several heterogeneous ST data sources, where the ST heterogeneity is an obvious characteristic of meteorological data. This designation helps to not only utilize complementary information in different data sources but also reduce the risk of errors from one single data source. Specifically, HSTN comprises three modules: Gaussian diffusion module, ST encoder and ST decoder. Considering that most of meteorological data can be formatted into either a dense ST tensor or a sparse ST tensor, the Gaussian diffusion module is responsible for converting a sparse ST tensor into a dense form, and then the ST encoder intends to extract task-related information from several dense ST tensors (maybe heterogeneous in time). Finally, the ST decoder seeks to fuse all information and produce the final prediction. By organically combining the three modules, HSTN can handle complex input with heterogeneity in both space and time domains. The contributions of this work are summarized as follows:

1) We propose HSTN for lightning predictions, which can mine knowledge from several heterogeneous ST data sources.
2) A Gaussian diffusion module is proposed to convert a sparse ST tensor into a dense form. It is computation-efficient and completely compatible with the DNN framework.
3) We design a multi-scale pooling loss to deal with the short-sight problem (to be explained in Section IV-D) caused by grid-wise losses for lightning predictions.

The rest of this paper is organized as follows. In Section II, we discuss the related works. Section III introduces the preliminaries. Section IV details the proposed HSTN. Experimental results are demonstrated in Section V. Finally, we conclude this paper in Section VI.

II. Related works

Lightning prediction. NWP based methods are now the mainstream for general lightning predictions. Here we focus on some classical works. As a pioneering work, Price and Rind [14] bridge the relation between lightning frequency and maximum vertical velocity (or convective cloud top height), and propose the well-known PR92 lightning parameterization scheme. Many works have developed from PR92. Michalon et al. [15] propose a new parameterization by combining PR92 and cloud droplet concentration. Mccaul et al. [2] propose two approaches based on the upward fluxes of precipitating ice hydrometeors in the mixed-phase region and the vertically integrated amounts of ice hydrometeors, respectively. Qie et al. [16] utilized an empirical formula with the ice-phase mixing ratio to connect the total lightning flash rate with ice-phase particles. Theoretically, the NWP based methods can achieve accurate long-term predictions provided high-grade NWP simulation. In practice, however, their performance is susceptible to the inevitable errors in the NWP process, which motivates us to synthesize multi-source data to produce a more robust prediction.

DNN based weather pattern mining. As a breakthrough work, Shi et al. [10] develop the conventional LSTM and propose convolutional LSTM (ConvLSTM), with an application to precipitation nowcasting. ConvLSTM now has become a basic tool for ST data mining. Furthermore, they also proposed the Trajectory GRU model [17] that improves ConvLSTM by actively learning the recurrent connection structure. Wang et al. [11] presents a predictive recurrent neural network (PredRNN) in the light of the idea that ST predictive learning should memorize both spatial appearances and temporal variations in a unified memory pool. Schon et al. [1] propose to nowcast lightning based on the error of two-dimensional optical flow algorithms applied to images of meteorological satellites, which provides a new view for nowcasting tasks. Wang et al. [12] and Geng et al. [13] propose to predict future weather by combining information from both historical data and NWP simulation, which shares a similar motivation with us. In our work, however, we consider a more challenging scenario—enhancing the weather prediction by mining knowledge from several heterogeneous ST data sources.

Deep ST model. Recently, some deep models are proposed to handle ST data, such as PredCNN [18], StepDeep [19], Hetero-ConvLSTM [20], DML [21] and CoST-Net [22]. PredCNN [18] designs a cascade multiplicative unit and models the dependencies between the next frame and the ST inputs by an entirely CNN based architecture. StepDeep [19] formulates the problem of mobility event predictions as an ST mining task and then designs a spatial-temporal progress cell to solve it. Hetero-ConvLSTM [20] is the first work seeking to address the spatial heterogeneity of ST data, which introduces spatial graph features and spatial model ensemble on top of the basic ConvLSTM. DML [21] proposes a meta graph attention module and a meta recurrent module to capture diverse spatial and temporal correlations, respectively, based on which a sequence-to-sequence network is built to deal with ST urban traffic data. CoST-Net [22] decomposes each time slice of ST data into a combination of hidden spatial demand bases, and thus the combination weights instead of the raw slice are sent into a heterogeneous LSTM net to make a prediction. Based on the above works, we consider a more general problem—how to handle various ST data with heterogeneity in both space and time domains.

III. Preliminary

We first introduce the heterogeneous ST data used in our work, and then define the problem to be solved. Throughout the paper, we use Italic lowercase (x), Italic capital (X), boldface lowercase ($\mathbf{x}$), boldface capital ($\mathbf{X}$) and Calligraphic ($\mathcal{X}$) symbols to denote scalars, tensors, vectors, matrices and general sets, respectively. $\mathbf{I}$, $\mathbf{0}$ and $\mathbf{1}$ represent the identity matrix, the zero matrix and the all-one vector, respectively, whose dimensions are determined by the context. $\|\mathbf{x}\|$ and $\|\mathbf{X}\|$ denote the ℓ_2 vector norm of $\mathbf{x}$ and the ℓ_2 operator norm of $\mathbf{X}$, respectively.

A. Heterogeneous Spatiotemporal Data

As illustrated in Figure 1, the model inputs comprise several types of data, each of which possesses its own ST structure. Next, we separately introduce them according to their structures.

1) WRF Simulation: The Weather Research and Forecasting (WRF) Model [23] is a next generation numerical weather prediction system designed for both atmospheric research and operational forecasting applications. The WRF model can provide abundant weather-parameter simulations for future several hours, with each parameter characterized by a four-dimensional tensor, where the four dimensions are longitude (x), latitude (y), altitude (z) and time scale (t), respectively. Specifically, for a simulation tensor X, $X_t^{x,y,z}$ stores the simulation value in a grid with an ST coordinate (x, y, z, t), and X_t (a three-dimensional tensor) denotes the slice at the t-th hour. Following [13], we choose simulated micro-physical parameters, radar reflectivity and maximum vertical velocity as our input. We concatenate all parameters along the z direction and form a comprehensive WRF tensor $W = [W_t]_{t=0}^{p-1}$, where p is the number of simulation hours and W_t comprises the simulated parameters at the t-th hour.

2) Lightning Observation: Considering the temporal correlation of thunderstorms, we introduce past lightning observation data [13] into the input. Following the structure of the WRF simulation data, we format the light observation records into a three-dimensional tensor L with indexes (x, y, t), where $L_t^{x,y}$ is a binary value indicating whether lightning happened during the period of $[t, t+1)$ in (x, y). We exploit the past l hours observations (i.e. $L = [L_t]_{t=-l}^{-1}$).

3) Weather Station Observation: We further employ the observation data from weather stations, which are sparsely distributed in the spatial plane. Each station can provide single-point observations of various weather parameters for the past hours. We select three lightning-related parameters [3] (i.e. average temperature, average relative humidity and precipitation) as our input. We use the observation for past s hours and denote it as $\bar{S} = [\bar{S}_t]_{t=-s}^{-1}$. Different from the previous two data (W and L), $\bar{S}$ is a four-dimensional sparse tensor (as illustrated in Figure 1), where the value of most grids is none (unobservable). Moreover, this data also faces a data missing challenge. Specifically, letting $|\cdot|$ denote the number of observable cells for a sparse tensor $\cdot$, we may have $|\bar{S}_{t_1}| \neq |\bar{S}_{t_2}|$ for $t_1 \neq t_2$. This raises the bar for our model.

B. Problem Definition

Suppose the current moment is $t = 0$. Given the WRF simulation $W = [W_t]_{t=0}^{p-1}$ (real four-dimensional tensor) for future p hours, the lightning observation $L = [L_t]_{t=-l}^{-1}$ (binary three-dimensional tensor) for past l hours, and the weather station observation $\bar{S} = \left[\bar{S}_t\right]_{t=-s}^{-1}$ (real four-dimensional sparse tensor) for past s hours, our target is to predict the lightning occurrence $\hat{L} = [\hat{L}_t]_{t=0}^{p-1}$ (binary three-dimensional tensor) for next p hours, where W, L, and $\hat{L}$ share the same x-y scope and resolution. Here we should emphasize the heterogeneous ST structure of our data, as illustrated in Figure 1. On one hand, the simulation data (W) and the observation data (L and $\bar{S}$) describe weather situations for different periods (the past and the future). On the other hand, the sparse structure of ($\bar{S}$) distinguishes it from other dense tensors (W and L). We seek to mining knowledge from these heterogeneous data.

During the training stage, the ground-truth lightning occurrence $L^\star = [L_t^\star]_{t=0}^{p-1}$ is known to us. Our task can be formulated as the following optimization problem:

$$\begin{aligned} \min_{\theta} \quad & \mathbb{E}_{(W,L,\bar{S},L^\star)\sim\mathcal{T}}\, loss(\hat{L}, L^\star) \\ \text{s. t.} \quad & \hat{L} \triangleq f_\theta(W, L, \bar{S}), \\ & \theta \in \Theta, \end{aligned} \tag{1}$$

where $\mathcal{T}$ denotes the training set and f_θ is a function characterized by θ constrained in Θ. In this paper, we exploit

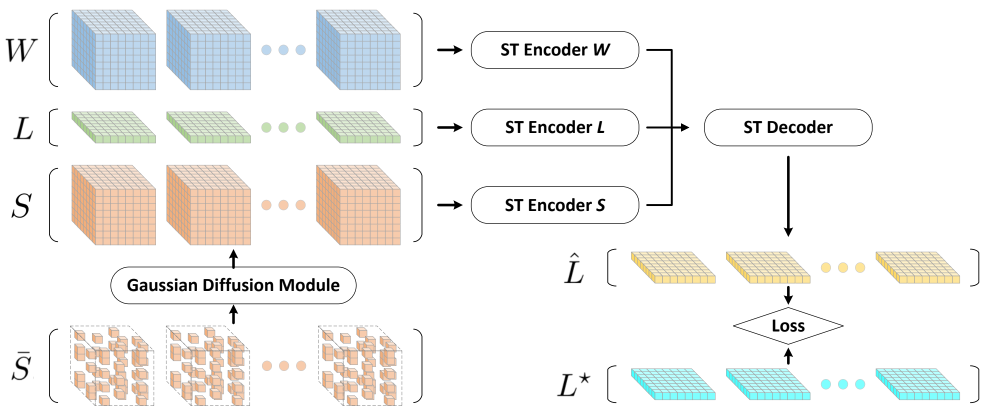


Fig. 2: The architecture of HSTN. The Gaussian diffusion module converts the sparse tensor $\bar{S}$ into a dense form S. Three ST encoders extract information from W, L and S. The ST decoder merges all information and produces lightning prediction $\hat{L}$.

neural networks to model f_θ, and thus θ denotes the network parameters.

IV. Heterogeneous Spatiotemporal Network

This section presents the architecture of the proposed heterogeneous spatiotemporal network (HSTN), as illustrated in Figure 2. Section IV-A introduces the Gaussian diffusion module to convert the sparse tensor $\bar{S}$ into a dense form S. Section IV-B depicts the ST encoder, whose structure is shared by three components to process W, L and S, respectively. Section IV-A displays the ST decoder which merges all information and generates the lightning prediction. Finally, we detail the multi-scale pooling loss in Section IV-D.

A. Gaussian Diffusion Module

This module receives the sparse tensor $\bar{S}$ and outputs a dense tensor S. If we treat this module independently of the others, the task is reduced into a tensor completion (or interpolation) problem. However, this isolation prohibits the module from communicating with the final loss. We design the Gaussian diffusion module to densify $\bar{S}$ in light of the final loss.

To begin with, let us consider a slice $\bar{S}_t^{:,:,k}$ (a sparse matrix, i.e. the k-th parameter of the t-th hour) of $\bar{S}$. As shown in Figure 3, the shaded and the plain grids denote the observable and the unobservable grids, respectively. The intuition is that the value of each grid should be related to its neighbors. Specifically, for some grid s_i (where i uniquely corresponds to a (x, y) and thus s_i uniquely indicates a grid in $\bar{S}_t^{:,:,k}$), we model the relation from s_i to its neighbors $\mathcal{N}(s_i)$ by the following Gaussian conditional probability distribution:

$$p\left(s_i|\mathcal{N}(s_i)\right) \triangleq \frac{1}{\sqrt{2\pi}\sigma_i} \cdot \exp\left\{-\frac{1}{2\sigma_i^2}\left(s_i - \textstyle\sum_{s_j \in \mathcal{N}(s_i)} w_{ij} s_j\right)^2\right\}, \tag{2}$$

where w_{ij} weights the neighbor s_j and σ_i adjusts the strength of the correlation.

Next, considering the whole x-y space of $\bar{S}_t^{:,:,k}$, let $\mathbf{s}_o$ and $\mathbf{s}_u$ be two vectors collecting all observable grids and unobservable grids of $\bar{S}_t^{:,:,k}$, respectively. Following [24], we define the joint pseudo-likelihood of $\mathbf{s}_o$ and $\mathbf{s}_u$ as

$$\begin{aligned}\hat{p}(\mathbf{s}_o, \mathbf{s}_u) &\triangleq \prod_{i=1}^{n} p\left(s_i|\mathcal{N}(s_i)\right) \\ &= \frac{1}{(2\pi)^{n/2}\sqrt{\mathbf{\Sigma}}} \exp\left\{-\frac{1}{2}(\mathbf{Ws} - \mathbf{s})^T \mathbf{\Sigma}^{-1} (\mathbf{Ws} - \mathbf{s})\right\},\end{aligned} \tag{3}$$

where n is the number of total grids of $\bar{S}_t^{:,:,k}$, $\mathbf{s} = [\mathbf{s}_o{}^T, \mathbf{s}_u{}^T]^T \in \mathbb{R}^n$, $\mathbf{W} \in \mathbb{R}^{n \times n}$ collects $\{w_{ij}\}_j$ in its i-th row and $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ collects σ_i^2 as its i-th diagonal elements for $1 \le i \le n$. Although $\hat{p}$ may not be properly normalized, previous work [25] has shown that maximizing $\hat{p}$ is asymptotically consistent with the true maximum likelihood estimator. Based on (3), our goals become

- Estimation:
$$\begin{aligned}\hat{\mathbf{W}}, \hat{\mathbf{\Sigma}} &= \arg\max_{\mathbf{W},\mathbf{\Sigma}} \ln \hat{p}(\mathbf{s}_o|\mathbf{W}, \mathbf{\Sigma}) \\ &= \arg\max_{\mathbf{W},\mathbf{\Sigma}} \ln \int_{\mathbf{s}_u} \hat{p}(\mathbf{s}_o, \mathbf{s}_u|\mathbf{W}, \mathbf{\Sigma}).\end{aligned} \tag{4}$$
- Inference:
$$\hat{\mathbf{s}}_u = \arg\max_{\mathbf{s}_u} \ln \hat{p}(\mathbf{s}_o, \mathbf{s}_u|\hat{\mathbf{W}}, \hat{\mathbf{\Sigma}}). \tag{5}$$

We settle (5) first, and then (4) can be naturally solved by the EM algorithm. Solving (5) is equivalent to maximizing

$$\begin{aligned}\ln \hat{p}(\mathbf{s}_o, \mathbf{s}_u|\hat{\mathbf{W}}, \hat{\mathbf{\Sigma}}) &= -\frac{1}{2}\|\hat{\mathbf{\Sigma}}^{-1/2}(\hat{\mathbf{W}} - \mathbf{I})\mathbf{s}\|^2 + c \\ &= -\frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{s}_u\|^2 + c,\end{aligned} \tag{6}$$

where c is a constant independent of $\mathbf{s}_u$,

$$\mathbf{A} = \begin{bmatrix} \hat{\mathbf{\Sigma}}_1^{-1/2} \hat{\mathbf{W}}_{12} \\ \hat{\mathbf{\Sigma}}_2^{-1/2} (\hat{\mathbf{W}}_{22} - \mathbf{I}) \end{bmatrix} \tag{7}$$

and

$$\mathbf{b} = \begin{bmatrix} \hat{\mathbf{\Sigma}}_1^{-1/2} (\hat{\mathbf{W}}_{11} - \mathbf{I}) \mathbf{s}_o \\ \hat{\mathbf{\Sigma}}_2^{-1/2} \hat{\mathbf{W}}_{21} \mathbf{s}_o \end{bmatrix}. \tag{8}$$

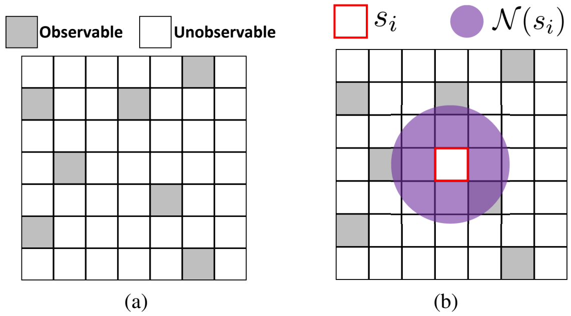


Fig. 3: (a) A sparse matrix $\bar{S}_t^{:,:,k}$. (b) s_i and its neighbors $\mathcal{N}(s_i)$. For simplicity, we only display the 3×3 neighbors, which can be extended according to requirements.

Algorithm 1 Iteration

```
Input: $\mathbf{s}_o$, $\hat{\mathbf{W}}$, $\hat{\mathbf{\Sigma}}$
  Initialize $\mathbf{s}_u^{(0)} = \mathbf{0}$
  for $i = 0, 1, \dots$ do
      $$\begin{aligned} \mathbf{s}_u^{(i+1)} &= \mathbf{s}_u^{(i)} + \alpha^{(i)} [\mathbf{0} \;\; \mathbf{I}] \left(\mathbf{b} - \mathbf{A}\mathbf{s}_u^{(i)}\right) \\ &= \mathbf{s}_u^{(i)} + \alpha^{(i)} [\mathbf{0} \;\; \mathbf{I}] \underbrace{\hat{\mathbf{\Sigma}}^{-1/2} \underbrace{\left(\hat{\mathbf{W}} - \mathbf{I}\right) \mathbf{s}^{(i)}}_{\text{convolution}}}_{\text{locally-connected}} \end{aligned}$$
    where $\mathbf{s}^{(i)} = [\mathbf{s}_o{}^T, \mathbf{s}_u^{(i)}{}^T]^T$.
  end for
  return $[\mathbf{s}_o^T, \hat{\mathbf{s}}_u^T]^T$
```

Maximizing (6) can be further simplified as

$$\min_{\mathbf{s}_u} \|\mathbf{b} - \mathbf{A}\mathbf{s}_u\|^2, \tag{9}$$

which is a least square problem with the optimal solution $\mathbf{s}_u^\star = \mathbf{A}^\dagger \mathbf{b}$. However, calculating the pseudo inverse $\mathbf{A}^\dagger$ is not only computation-consuming but also barely compatible with the DNN framework. Inspired by the Richardson method [26], we propose to solve (9) via an iteration method shown in Algorithm 1. The following proposition presents its convergence characteristics.

Proposition 1. *Suppose that* $0 < \alpha^{(i)} \leq 1$ $(i = 0, 1, \dots)$, $\sum_{l=0}^{\infty} \alpha^{(i)} = a < \infty$, $\|\hat{\mathbf{\Sigma}}^{-1/2}\| \leq 1$ *and* $\|\hat{\mathbf{W}}\| \leq 1 - \epsilon$, *where* ϵ *is a small positive constant. The iteration in Algorithm* (1) *will converge to* $\hat{\mathbf{s}}_u$ *with*

$$\|\mathbf{b} - \mathbf{A}\hat{\mathbf{s}}_u\| \leq (1 + 2a)\|\mathbf{b} - \mathbf{A}\mathbf{s}_u^\star\|, \tag{10}$$

where $\mathbf{s}_u^\star$ *is the optimal solution to* (9).

Proof. See appendix. □

The superiority of Algorithm 1 lies in that all its operations are simple matrix addition and multiplication, and thus completely compatible with the DNN framework. Specific to our task, $\mathbf{W}$ and $\mathbf{\Sigma}^{-1/2}$ are implemented as a 5×5 convolution layer and a 1×1 locally-connected layer with constraints, respectively. The iteration will stop after q times. Based on Algorithm 1, our Gaussian diffusion module is presented in Algorithm 2, in which $\text{vec}(\cdot)$ flattens a matrix $\cdot$ into a vector and $\text{ivec}(\cdot)$ is its inverse operation. Utilizing this module, the weather observation sparse tensor $\bar{S}$ is transformed into a dense tensor S.

Algorithm 2 Gaussian diffusion

```
Input: $\bar{S}$
  for $k = 1, 2, 3$ do
    Initialize $\mathbf{W}^k$ and $\mathbf{\Sigma}^k$
    for $t = -s, -s+1, \dots, -1$ do
      $\mathbf{s}_o = \text{vec}(\bar{S}_t^{:,:,k})$ (only observable grids);
      $\mathbf{s}_t^k = \text{Iteration}(\mathbf{s}_o, \mathbf{W}^k, \mathbf{\Sigma}^k)$ (cf. Algorithm 1)
      $S_t^{:,:,k} = \text{ivec}(\mathbf{s}_t^k)$
    end for
  end for
  return $[S_t]_{t=-s}^{-1}$
```

Now we employ the EM algorithm to solve the remaining (4), which is equivalent to

$$\max_{\mathbf{W}, \mathbf{\Sigma}} \mathbb{E}_{\hat{p}(\mathbf{s}_u | \mathbf{s}_o, \mathbf{W}, \mathbf{\Sigma})} \ln \hat{p}(\mathbf{s}_o, \mathbf{s}_u | \mathbf{W}, \mathbf{\Sigma}). \tag{11}$$

The expectation in (11) can be estimated as [24]:

$$\mathbb{E}_{\hat{p}(\mathbf{s}_u | \mathbf{s}_o, \hat{\mathbf{W}}, \hat{\mathbf{\Sigma}})} \ln \hat{p}(\mathbf{s}_o, \mathbf{s}_u | \mathbf{W}, \mathbf{\Sigma}) \approx \ln \hat{p}(\mathbf{s}_o, \hat{\mathbf{s}}_u | \mathbf{W}, \mathbf{\Sigma}), \tag{12}$$

where $\hat{\mathbf{s}}_u$ is provided by Algorithm 1. Thus we add the following objective function into the final loss:

$$\ell_d = -\sum_{t,k} \ln \hat{p}(\mathbf{s}_t^k | \mathbf{W}^k, \mathbf{\Sigma}^k), \tag{13}$$

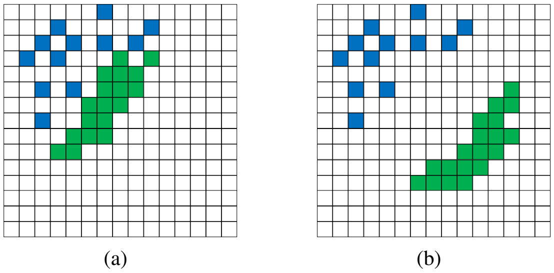
where $\mathbf{s}_t^k$, $\mathbf{W}^k$, $\mathbf{\Sigma}^k$ are defined in Algorithm 2. $\{\mathbf{W}^k\}_{k=1}^3$ and $\{\mathbf{\Sigma}^k\}_{k=1}^3$ will be optimized via backpropagation.

B. ST Encoder

Given a dense ST tensor, the ST encoder aims to extract the task-related knowledge from it. This task fits perfectly with the convolutional long short-term memory (ConvLSTM) [10], which has proven successful in processing ST tensors. Following the conventional notations, let X_t, C_t and H_t denote the input, the cell state and the hidden state of ConvLSTM at the t-th moment, respectively. A ConvLSTM cell is characterized by the following equations:

$$\begin{aligned} Z_t^i &= \sigma(P_{xi} * X_t + P_{ci} \circ C_{t-1} + P_{hi} * H_{t-1} + B_i), \\ Z_t^f &= \sigma(P_{xf} * X_t + P_{cf} \circ C_{t-1} + P_{hf} * H_{t-1} + B_f), \\ C_t &= Z_t^f \circ C_{t-1} + Z_t^i \circ \tanh(P_{xc} * X_t + P_{hc} * H_{t-1} + B_c), \\ Z_t^o &= \sigma(P_{xo} * X_t + P_{ho} * H_{t-1} + P_{co} \circ C_t + B_o), \\ H_t &= Z_t^o \circ \tanh(C_t), \end{aligned}$$

where $*, \circ$ and $\sigma(\cdot)$ denote the convolution operator, the Hadamard product and the logistic sigmoid function, respectively. $P.$ and $B.$ are trainable parameters compatible with the corresponding operator.



(a) (b)

Fig. 4: Two cases of the lightning prediction: (a) and (b). Real-lightning grids and predicted-lightning grids are colored by blue and green, respectively. Intuitively, (a) is better than (b).

Next, we build our ST encoder based on ConvLSTM. Specifically, for a given ST tensor ($X = [X_t]_{t=\tau_1}^{\tau_2}$), we initialize all states of ConvLSTM as zero (i.e. $C_{\tau_1-1} = 0$ and $H_{\tau_1-1} = 0$), and then recurrently feed X_t into the ConvLSTM as input for $\tau_1 \leq t \leq \tau_2$. Finally, we return the cell state (C_{τ_2}) and the hidden state (H_{τ_2}) of the last step as the outputs of the ST encoder.

The ST encoder provides an effective way to summarize information from ST tensors. We exploit it to process the WRF simulation W, the lightning observation L, and the densified weather station observation S. Specifically, their features are extracted by

$$\begin{aligned} C^W, H^W &= \text{ST\_encoder}^W(W), \\ C^L, H^L &= \text{ST\_encoder}^L(L), \\ C^S, H^S &= \text{ST\_encoder}^S(S). \end{aligned}$$

These features will be fed into the ST decoder to produce lightning predictions.

C. ST Decoder

This module summarizes all state information come from the ST encoders and infer the prediction $\hat{L}$. Following the same reason presented in Section IV-B, ConvLSTM also applies to the ST decoder. Before feeding all state tensors into a ConvLSTM, we perform a convolution operation with relu activation to transform them into a uniform state space:

$$\begin{aligned} \tilde{C}^k &= \text{Conv2D}^{C,k}(C^k) \quad (k \in \{\text{“}W\text{”}, \text{“}L\text{”}, \text{“}S\text{”}\}), \\ \tilde{H}^k &= \text{Conv2D}^{H,k}(H^k) \quad (k \in \{\text{“}W\text{”}, \text{“}L\text{”}, \text{“}S\text{”}\}), \\ \tilde{C} &= \text{Concatenate}(\tilde{C}^W, \tilde{C}^L, \tilde{C}^S), \\ \tilde{H} &= \text{Concatenate}(\tilde{H}^W, \tilde{H}^L, \tilde{H}^S), \end{aligned}$$

where $\tilde{C}$ and $\tilde{H}$ encapsulate the information from three ST encoders, which will be set as the initial states of ConvLSTM. After that, we recurrently run ConvLSTM p steps:

$$\begin{aligned} C_{-1} =& \tilde{C}, \\ H_{-1} =& \tilde{H}, \\ C_t, H_t =& \text{ConvLSTM}(C_{t-1}, H_{t-1}) \quad (t = 0, \ldots, p-1). \end{aligned}$$

Algorithm 3 Multi-scale pooling loss

```
Input: $\hat{L}, L^\star$
  for $t = 0, \ldots, p-1$ do
    for $k = 1, 2, 3$ do
      $\ell_t^k = \text{mean\_weighted\_CE}(w^k, \hat{L}_t, L_t^\star)$
      $\hat{L}_t = \text{max\_pool}(\hat{L}_t)$
      $L_t^\star = \text{max\_pool}(L_t^\star)$
    end for
    $\ell_t = r^1\ell_t^1 + r^2\ell_t^2 + r^3\ell_t^3$
  end for
  $\ell_m = \frac{1}{t}\sum_{t=0}^{p-1} \ell_t$
  return $\ell_m$
```

Finally, we employ a convolution layer with sigmoid activation to transform the outputs of ConvLSTM into the lightning prediction:

$$\hat{L}_t = \text{Conv2D}^L(H_t) \quad (t = 0, \ldots, p-1)$$

The difference between $\hat{L}$ and $L^\star$ will be measured by the proposed multi-scale pooling loss, detailed in Section IV-D.

D. Multi-scale Pooling Loss

Since lightning is a rare event compared to no lightning, the spatial distribution of lightning demonstrates sparsity, causing a short-sight problem to grid-wise measures. Taking Figure 4 as an example, the blue and the green grids represent the real-lightning and the predicted-lightning grids, respectively. We intuitively deem that the prediction in Figure 4a is better than that in Figure 4b, since the former is “closer” to the real-lightning area than the latter. However, we get the same grid-wise loss for both cases, i.e. no real-lightning grids are hit for both cases.

We propose the multi-scale pooling loss to meet this challenge. The idea is measuring the difference between the prediction and the groundtruth from multiple scales. A large-scale loss tries to “pull” misplaced predictions closer to real-lightning areas while a small-scale loss is responsible for adjusting the “contour” of predictions to fit the distribution of real-lightning. Furthermore, we choose the weighted cross entropy (mean_weighted_CE) to balance the quantity gap between lightning and no-lightning grids. We detail the multi-scale pooling loss in Algorithm 3, in which $\{w^k\}_{k=1}^3$ and $\{r^k\}_{k=1}^3$ are the weights of positive grids and the ratios for three scales, respectively. Finally, we summarize the data flow of HSTN in Algorithm 4.

V. Experiments

A. Experiments Setup

Parameter setting. We empirically determine the parameters of HSFN. For the parameters of the Gaussian diffusion module (cf. Section IV-A), the iteration number q is set to 20 and the decay factor $\alpha^{(i)}$ is set to 0.95^i for $1 \leq i \leq q$. The convolution kernel corresponding to $\mathbf{W}$ is constrained to be positive and hollow (i.e. $\text{diag}(\mathbf{W}) = \mathbf{0}$), with sum equaling

Algorithm 4 Heterogeneous ST network

```
Input: $W, L, \bar{S}, L^{\star}$
$S = \text{Gaussian\_diffusion}(\bar{S})$
$\ell_d = -\sum_{t,k} \ln \hat{p}(\mathbf{s}_t^k | \mathbf{W}^k, \mathbf{\Sigma}^k)$ (cf. (13))
$C^W, H^W = \text{ST\_encoder}^W(W)$
$C^L, H^L = \text{ST\_encoder}^L(L)$
$C^S, H^S = \text{ST\_encoder}^S(S)$
$\hat{L} = \text{ST\_decoder}(C^W, H^W, C^L, H^L, C^S, H^S)$
$\ell_m = \text{Multi-scale\_pooling\_loss}(\hat{L}, L^{\star})$
return $\ell = \lambda \ell_d + (1 - \lambda) \ell_m$
```

TABLE I: The detailed information for three performance metrics.

Name	Equation	Range	Explanation
POD	$\frac{n_1}{n_1+n_3}$	[0,1]	The ratio of the number of hit lightnings to the number of observed lightnings. Larger is better.
FAR	$\frac{n_2}{n_1+n_2}$	[0,1]	The ratio of the number of false alarm lightnings to the number of foretasted lightnings. Smaller is better.
ETS	$\frac{n_1-r}{n-n_4-r}$	$[-\frac{1}{3}, 1]$	The ratio of the number of hit lightnings to the number of events except for the correct rejections, and removed the contribution from hits by chance in random forecasts. Larger is better.

0.98 (i.e. $\mathbf{W1} = 0.98 \cdot \mathbf{1}$). The diffusion variance $\mathbf{\Sigma}$ is set to the identity matrix. For the parameters related to the loss function (cf. Section IV-D), w^1, w^2, w^3, r^1, r^2, r^3 and λ are set to 18, 6, 2, 0.6, 0.3, 0.1 and 0.05, respectively. We optimize the model by the Adam method [27] with an initial learning rate 1×10^{-4}. The batch size and epoch number are set to 8 and 30, respectively. Our code was released on GitHub.

Dataset. We consider a real lightning dataset of North China. Specifically, this region is a square with a center at 40°N and 116.2°E, which is divided into 159×159 grids and each grid has a resolution of 4km × 4km. The number of weather stations is 237, leading to a sparsity ratio $237/(159 \times 159) \approx 0.001$. The number of input hours p, l and s (cf. Section III-B) are $6, 3$ and 6, respectively. $p = 6$ implies that our task is to predict lightning for next six hours. The periods of data cover June to September in 2015, and May to September in both 2016 and 2017, 14 months in total. Following [13], we chronologically divide the total dataset by a ratio of 11:1:2 for training, validation and testing. Since there are too many samples containing no 318 lightning, which take little effect on our task, we remove them from the original dataset. As a result, the number of samples in the training set, the validation set and the test set becomes 3656, 449 and 236, respectively.

Baselines. We compare the proposed HSTN with the following baseline models: PR92 [28], LF1 [29], LF2 [29], StepDeep [19], StepDeep+, LightNet [13] and LightNet+, where the methods with suffix "+" mean that the bicubic interpolation of weather-station features are added into their input, to distinguish from the results only employing WRF simulation and light observation as reported in [13]. Among these baselines, the first three are traditional lightning prediction methods and the others are DNN-based models. In addition, we also introduce two variants (HSTN-grid and HSTN-interp) of HSTN for comparison. They are formed by replacing the multi-scale pooling loss and the Gaussian diffusion module of HSTN by a grid-wise loss and bicubic interpolation, respectively. We repeatedly train each model three times and select the one with the best validation score.

Performance metric. We evaluate the prediction results by three metrics: probability of detection (POD), false alarm ratio (FAR) and equitable threat score (ETS), which have been widely applied in meteorology [30]. Let n denote the total number of grids. Let n_1, n_2, n_3 and n_4 denote the number of true-positive, false-positive, false-negative and true-negative grids, respectively. Define the expectation of the number of hit lightnings in random forecasts as $r = (n_1 + n_2)(n_1 + n_3)/n$. The equations for the three metrics are detailed in Table I. Following [13], we also calculate eight-neighborhood-based POD, FAR and ETS, where $\{n_i\}_{i=1}^4$ is replaced by the eight-neighborhood-based statistics $\{\tilde{n}_i\}_{i=1}^4$ [30]. We evaluate the prediction results over three periods, i.e. the first three hours, last three hours and all six hours.

B. Effect of data sources

The first experiment is to investigate whether exploiting heterogeneous data sources helps to achieve better prediction accuracy. We evaluate the performance of our HSTN model trained on different combinations of data sources including WRF simulation (WRF), lightning observation (LIG) and weather station observation (STA).

From the upper part of Table II, we notice that all three single data sources obtain positive ETSs over all prediction periods, indicating each of them contains information conducive to the lightning prediction. Among them, LIG displays surprisingly better performance than the other two data sources, owing to the strong correlation between the past lightning and the future lightning. This correlation, however, becomes weaker and weaker as the prediction time goes by, causing a performance drop over last three hours. A similar phenomenon also applies to the STA prediction from first three hours to last three hours. In contrast, for WRF, the performance decay from first three hours to last three hours is much less obvious, since the simulation data is not as dependent on timeliness as the observation data.

As shown in the middle part of Table II, the combinations of different sources boost the prediction accuracy more or less in most circumstances. Among the three combinations, WRF+LIG obtains very promising results, which are close to the best results by employing all data sources. Meanwhile, WRF+STA achieves obvious improvements in comparison with both WRF and STA over all prediction periods. The above two cases validate the effectiveness of the combination of simulation data and observation data, which contain complementary information about thunderstorms. On the other hand, although LIG+STA achieves much better results than

TABLE II: Quantitative results (POD(%), FAR(%) and ETS) of HSTN on combinations of various data sources. The best performance is reported using **bold red**, and the second best is reported using **bold blue**.

Data Source	First Three Hour Score						Last Three Hour Score						Six Hour Score					
	Strict Metric			Neighb.-Based Metric			Strict Metric			Neighb.-Based Metric			Strict Metric			Neighb.-Based Metric		
	POD	FAR	ETS	POD	FAR	ETS	POD	FAR	ETS	POD	FAR	ETS	POD	FAR	ETS	POD	FAR	ETS
WRF	13.7	84.7	0.074	31.0	60.1	0.207	13.1	88.9	0.059	31.1	68.5	0.180	17.4	80.0	0.095	35.1	52.7	0.244
LIG	**64.1**	79.9	0.174	**82.9**	51.3	0.433	18.7	85.0	0.086	37.3	63.3	0.221	**46.1**	75.5	0.180	**67.7**	**44.8**	0.425
STA	13.3	85.0	0.071	26.9	65.4	0.174	1.2	95.4	0.007	4.0	83.8	0.031	9.1	82.5	0.058	19.2	60.3	0.142
WRF+LIG	57.5	**78.3**	**0.181**	77.7	**48.9**	**0.438**	**22.6**	**81.1**	**0.110**	**42.0**	**55.9**	**0.269**	45.0	**72.4**	**0.196**	65.6	**40.4**	**0.443**
WRF+STA	24.9	85.9	0.093	48.8	62.9	0.260	16.6	86.8	0.074	35.5	65.6	0.206	26.1	79.8	0.119	47.5	52.4	0.301
LIG+STA	**62.4**	79.5	0.176	**81.5**	50.7	0.435	17.3	86.8	0.076	36.1	66.2	0.206	45.4	76.1	0.175	66.8	46.0	0.413
All	60.5	**78.1**	**0.185**	80.0	**48.4**	**0.449**	**26.1**	**80.9**	**0.118**	**46.7**	**55.1**	**0.291**	**48.8**	**72.6**	**0.202**	**69.2**	**40.4**	**0.459**

STA, its performance decreases slightly compared with LIG over last three hours. This suggests that information hold by STA may partly overlap with LIG, considering that both of them are observation data.

The last row of Table II demonstrates that the model with all data sources as input achieves the best performance over other models. The above results verify that the aggregation of heterogeneous data sources into our prediction model is helpful to learn more in-depth patterns of the development of thunderstorms.

C. *Quantification Results*

As show in Table III, the upper, middle and lower parts report the results of three meteorologic methods, four DNN-based baselines and HSTN with its variations, respectively. We first notice a huge performance gap between the meteorologic methods and the DNN models. The poor performance of meteorologic baselines can be explained in two ways. Firstly, they are completely based on the WRF simulation, which usually have deviations in both space and time domains, introducing irreparable biases to these methods. Secondly, these methods are essentially some simple functions designed empirically, which can hardly model the complicated relations among various meteorologic parameters and benefit from massive historical data. In contrast, the DNN-based methods own much stronger modeling capabilities and can mine knowledge from training data, helping them achieve significantly higher prediction accuracy.

Among the DNN-based baselines, we observe that the models (StepDeep+ and LightNet+) with the weather-station input achieve overall better performance than those (StepDeep and LightNet) without it, validating the effectiveness of this newly added input. However, these improvements are limited and unstable since they just treat the weather-station observation as a new "channel" of the input tensor, without in-depth exploration of the special structure inside this feature.

In contrast, HSTN obtains overall better results than three interpolation-based methods (StepDeep+, LightNet+ and HSTN-interp), indicating that the proposed Gaussian diffusion module, which considers the prediction loss as a supervisor, extracts more informative representation than the plain unsupervised interpolation. Here we should emphasize that the last-three-hour score is much more important than the first-three-hour score, and improving the former is much harder than the latter. On the other hand, HSTN achieves better performance than HSTN-grid over all prediction periods, implying that multi-scale pooling loss is more effective than the grid-wise loss for the lightning prediction task. Next, we focus on the three representative methods (StepDeep+, LightNet+ and HSTN) and compare their performance for different prediction periods as follows:

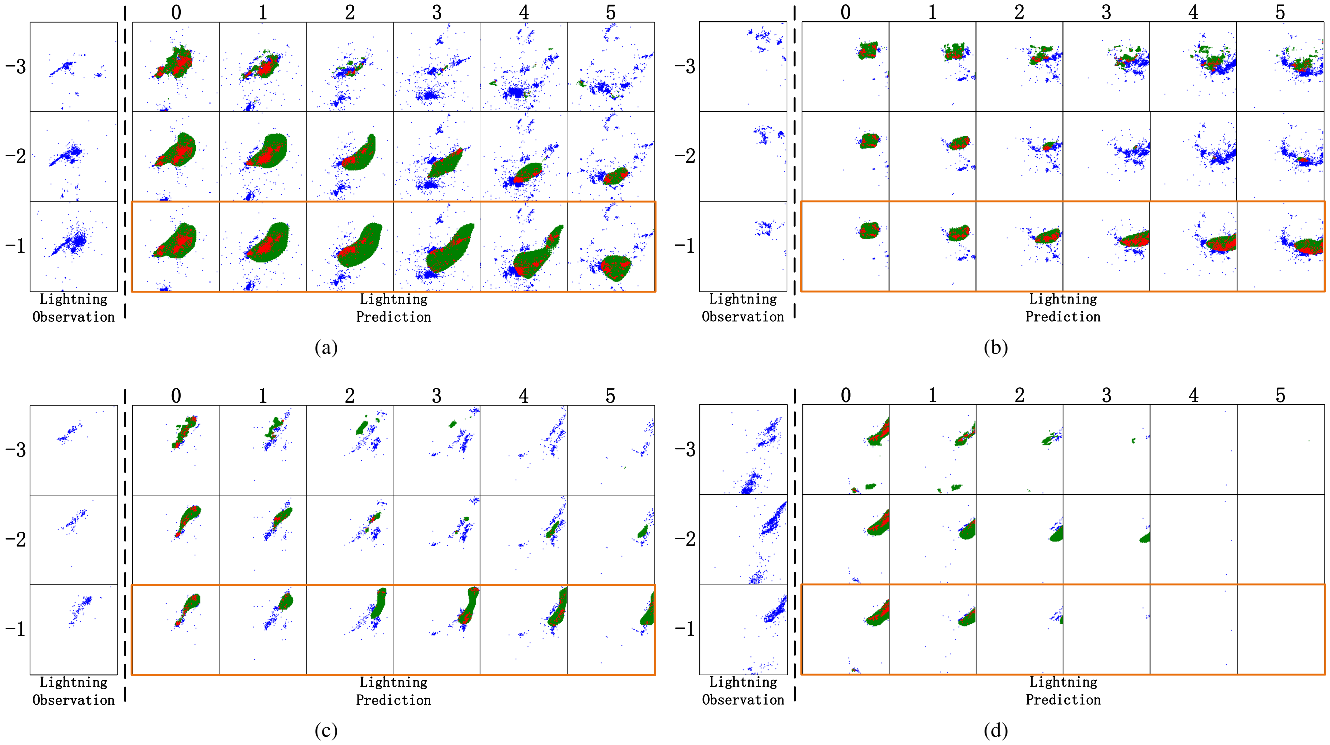
1) First-three-hour prediction. Among the three models, StepDeep+ and LightNet+ share similar FARs, but a higher POD brings LightNet+ a higher ETS than StepDeep+. In contrast, HSTN obtains the highest POD, while a high FAR lowers its ETS. In general, LightNet+ achieves the best performance over this prediction period. But the performance gap is not large between LightNet+ and HSTN.
2) Last-three-hour prediction. Compared with the first three hours, prediction over this period is much harder since the pattern of thunderstorms becomes almost unforeseeable after a three-hour development. As a result, the performance indicators of all methods dropped significantly. In this case, HSTN surprisingly displays a much higher POD while a smaller FAR than the two other baselines. Specifically, the strict POD of HSTN is about 20% and 200% higher than the second place (LightNet+) and the third place (StepDeep+), respectively, which brings HSTN an absolute superiority in ETS over this period. These results indicate that HSTN learns a more long-term pattern of thunderstorms by synthesizing information in multi-source data.
3) Six-hour prediction. HSTN achieves the highest POD and ETS on both strict and neighborhood-based metrics, although its FAR is slightly higher than the second place, LightNet+. LightNet+ obtains a very competitive ETS, but it is at the expense of a lower POD, which is an undesirable thing in the prediction task. For StepDeep+, the poor results over the last three hours cause its unsatisfactory overall performance over the six hours.

In summary, HSTN demonstrates the overall best performance among all models under test. It achieves superior prediction accuracy over last three hours, though at the cost of a little high FAR over first three hours.

TABLE III: Illustration for quantitative results (POD(%), FAR(%), ETS). The best performance is reported using **bold red**, and the second best is reported using **bold blue**. "–" indicates that no reported results are available. The method with suffix "+" means that the bicubic interpolation of weather-station features are added into the input.

Method	First Three Hour Score						Last Three Hour Score						Six Hour Score					
	Strict Metric			Neighb.-Based Metric			Strict Metric			Neighb.-Based Metric			Strict Metric			Neighb.-Based Metric		
	POD	FAR	ETS	POD	FAR	ETS	POD	FAR	ETS	POD	FAR	ETS	POD	FAR	ETS	POD	FAR	ETS
PR92*	18.0	96.9	0.019	52.1	88.5	0.094	–	–	–	–	–	–	23.2	95.2	0.027	54.7	86.1	0.110
LF1*	15.6	97.1	0.018	45.3	86.2	0.100	–	–	–	–	–	–	18.8	93.8	0.037	47.4	88.0	0.109
LF2*	15.0	94.5	0.035	37.7	82.8	0.126	–	–	–	–	–	–	19.1	92.0	0.048	39.5	80.5	0.139
StepDeep*	42.7	**74.1**	**0.187**	67.4	**40.2**	**0.458**	–	–	–	–	–	–	26.3	**70.9**	0.152	48.1	**35.2**	0.373
StepDeep+	48.1	77.4	0.176	73.6	**46.1**	0.445	8.0	88.4	0.046	23.0	64.8	0.157	34.2	74.5	0.162	58.9	41.7	0.404
LightNet*	59.8	77.7	**0.187**	**80.2**	47.4	**0.458**	22.3	82.8	0.102	42.3	59.1	0.257	46.5	73.3	0.194	68.0	41.3	0.449
LightNet+	58.6	**77.3**	**0.189**	78.7	46.3	**0.461**	20.4	81.8	0.102	39.0	57.1	0.252	45.9	**72.2**	0.199	66.6	**39.3**	0.454
HSTN-grid	59.7	78.2	0.184	79.2	48.7	0.444	23.0	**80.8**	**0.112**	42.3	**55.4**	**0.272**	47.3	72.6	**0.200**	67.5	40.9	0.449
HSTN-interp	**59.9**	78.3	0.183	79.8	48.6	0.447	**23.6**	81.9	0.109	**43.7**	56.7	**0.272**	**47.6**	72.8	0.199	**68.4**	40.3	**0.457**
HSTN	**60.5**	78.1	0.185	**80.0**	48.4	0.449	**26.1**	**80.9**	**0.118**	**46.7**	**55.1**	**0.291**	**48.8**	72.6	**0.202**	**69.2**	40.4	**0.459**

* Results are reported in [13].



(a)

(b)

(c)

(d)

Fig. 5: Visualization of four representative cases. In each case, the left part is lightning observations for past three hours, and the right part is predictions for future six hours, where the predictions from top to bottom are made by StepDeep+, LightNet+ and HSTN. The observation, prediction and their intersection are marked by blue (●), dark green (●) and red (●), respectively.

D. Visualization Results

Figure 5 visualizes four representative cases for StepDeep+, LihgtNet+ and HSTN. We observe that all methods make a better prediction for first three hours, which benefits from the trend information provided by the observation data. On the other hand, although the performance of all models drops over last three hours, HSTN displays a higher prediction accuracy than the other two baselines, which is consistent with the quantification results shown in Table III. Specifically, StepDeep+ and LihgtNet+ tend to predict no lightning over last three hours, since for them the gain of POD fails to offset the drawback of FAR in this period. In contrast, HSTN captures the core thunder zone though at risk of a little high FAR. Particularly, Figure 5d shows a case where thunderstorms gradually disappear. We find that HSTN anticipates the disappearing trend earlier than the other two baselines. The above results validate the superiority of HSTN.

VI. Conclusion

This paper investigated to solve the lightning prediction problem via a DNN model (HSTN) extracting information from several heterogeneous ST data sources. Noting that most of meteorological data can be formatted into either a dense ST tensor or a sparse ST tensor, HSTN employs the Gaussian diffusion module to convert a sparse ST tensor into a dense form. Thus, all data sources can be unified into dense ST tensors, which can be effectively processed by the ST encoder and the ST decoder. Besides, we propose a multi-scale pooling loss by measuring the difference between a prediction and the groundtruth from multiple scales, which proves effective in solving the short-sight problem caused by grid-wise losses. Experimental results show state-of-the-art performance on a real-world lightning dataset, which demonstrates the effectiveness of HSTN in mining knowledge from heterogeneous ST data sources.

Acknowledgment

This work is supported by the National Key Research and Development Program of China (No. 2017YFC1501503) and Beijing Natural Science Foundation (No. L191016).

Appendix

Proof of Proposition 1

We first derive the bound of $\|\mathbf{s}_u^\star - \mathbf{s}_u^{(i+1)}\|$:

$$\begin{aligned}
\|\mathbf{s}_u^\star-\mathbf{s}_u^{(i+1)}\| =& \|\mathbf{s}_u^\star-\mathbf{s}_u^{(i)}-\alpha^{(i)}\,[\mathbf{0}\ \ \mathbf{I}]\,(\mathbf{b}-\mathbf{A}\mathbf{s}_u^{(i)})\| \\
\le& \|\mathbf{s}_u^\star+\alpha^{(i)}\,[\mathbf{0}\ \ \mathbf{I}]\,(\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star) \\
&\quad -\mathbf{s}_u^{(i)}-\alpha^{(i)}\,[\mathbf{0}\ \ \mathbf{I}]\,(\mathbf{b}-\mathbf{A}\mathbf{s}_u^{(i)})\| \\
&\quad +\alpha^{(i)}\|\,[\mathbf{0}\ \ \mathbf{I}]\,(\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star)\| \\
=& \|\,[\mathbf{0}\ \ \mathbf{I}]\,(\mathbf{s}^\star-\mathbf{s}^{(i)})+ \\
&\quad +\alpha^{(i)}\,[\mathbf{0}\ \ \mathbf{I}]\,\boldsymbol{\Sigma}^{-1/2}\,(\mathbf{W}-\mathbf{I})\,(\mathbf{s}^\star-\mathbf{s}^{(i)})\| \\
&\quad +\alpha^{(i)}\|\,[\mathbf{0}\ \ \mathbf{I}]\,(\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star)\| \\
\le& \|\,[\mathbf{0}\ \ \mathbf{I}]\,(\mathbf{I}+\alpha^{(i)}\boldsymbol{\Sigma}^{-1/2}\,(\mathbf{W}-\mathbf{I}))(\mathbf{s}^\star-\mathbf{s}^{(i)})\| \\
&\quad +\alpha^{(i)}\|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\| \\
\le& \|\mathbf{I}+\alpha^{(i)}\boldsymbol{\Sigma}^{-1/2}\,(\mathbf{W}-\mathbf{I})\,\|\|\mathbf{s}^\star-\mathbf{s}^{(i)}\| \\
&\quad +\alpha^{(i)}\|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\| \\
=& \beta^{(i)}\|\mathbf{s}^\star-\mathbf{s}^{(i)}\|+\alpha^{(i)}\|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\| \\
=& \beta^{(i)}\|\mathbf{s}_u^\star-\mathbf{s}_u^{(i)}\|+\alpha^{(i)}\|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\|.
\end{aligned}$$

Based on the assumptions $\alpha^{(i)} < 1$, $\|\boldsymbol{\Sigma}^{-1/2}\| \le 1$ and $\|\hat{\mathbf{W}}\| \le 1-\epsilon$, we can infer $0 \le \beta^{(i)} \le 1-\gamma$, where γ is a small positive constant. Then we have

$$\begin{aligned}
\|\mathbf{s}_u^\star-\mathbf{s}_u^{(i+1)}\| \le& \prod_{j=0}^{i}\beta^{(j)}\|\mathbf{s}_u^\star-\mathbf{s}_u^{(0)}\| \\
&+\sum_{j=0}^{i}\alpha^{(j)}\prod_{k=j+1}^{i}\beta^{(k)}\|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\| \\
\le& \prod_{j=0}^{i}\beta^{(j)}\|\mathbf{s}_u^\star-\mathbf{s}_u^{(0)}\|+\sum_{j=0}^{i}\alpha^{(j)}\|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\|,
\end{aligned}$$

implying

$$\|\mathbf{s}_u^\star-\hat{\mathbf{s}}_u\| = \lim_{i\to\infty}\|\mathbf{s}_u^\star-\mathbf{s}_u^{(i+1)}\| \le a\|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\|. \tag{14}$$

(14) paved the way to the desirable conclusion:

$$\begin{aligned}
\|\mathbf{b}-\mathbf{A}\hat{\mathbf{s}}_u\| =& \|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star+\mathbf{A}(\mathbf{s}_u^\star-\hat{\mathbf{s}}_u)\| \\
\le& \|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\|+\|\mathbf{A}(\mathbf{s}_u^\star-\hat{\mathbf{s}}_u)\| \\
\le& \|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\|+2a\|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\| \\
=& (1+2a)\|\mathbf{b}-\mathbf{A}\mathbf{s}_u^\star\|.
\end{aligned} \tag{15}$$

References

[1] C. Schön *et al.*, “The error is the feature: How to forecast lightning using a model prediction error,” in *Proc. ACM KDD*, 2019, pp. 2979–2988.

[2] E. W. McCaul Jr *et al.*, “Forecasting lightning threat using cloud-resolving model simulations,” *Weather and Forecasting*, vol. 24, no. 3, pp. 709–729, 2009.

[3] T. M. Giannaros *et al.*, “Predicting lightning activity in greece with the weather research and forecasting (WRF) model,” *Atmospheric Research*, vol. 156, pp. 1–13, 2015.

[4] H. D. Betz *et al.*, “Cell-tracking with lightning data from LINET,” *Advances in Geosciences*, vol. 17, no. 17, pp. 55–61, 2008.

[5] M. Kohn *et al.*, “Nowcasting thunderstorms in the mediterranean region using lightning data,” *Atmospheric Research*, vol. 100, no. 4, pp. 489–502, 2011.

[6] Y. Yair *et al.*, “Predicting the potential for lightning activity in mediterranean storms based on the weather research and forecasting (WRF) model dynamic and microphysical fields,” *Journal of Geophysical Research: Atmospheres*, vol. 115, no. D4, 2010.

[7] J. Wong *et al.*, “Evaluating a lightning parameterization based on cloud-top height for mesoscale numerical model simulations,” *Geoscientific Model Development*, vol. 6, no. 2, pp. 429–443, 2013.

[8] J. Latham *et al.*, “Field identification of a unique globally dominant mechanism of thunderstorm electrification,” *Quarterly Journal of the Royal Meteorological Society*, vol. 133, no. 627, pp. 1453–1457, 2007.

[9] N. I. Sapankevych and R. Sankar, “Time series prediction using support vector machines: a survey,” *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[10] S. Xingjian *et al.*, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Proc. NeurIPS*, 2015, pp. 802–810.

[11] Y. Wang *et al.*, “Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms,” in *Proc. NeurIPS*, 2017, pp. 879–888.

[12] B. Wang *et al.*, “Deep uncertainty quantification: A machine learning approach for weather forecasting,” in *Proc. ACM KDD*, 2019, pp. 2087–2095.

[13] Y.-a. Geng *et al.*, “Lightnet: A dual spatiotemporal encoder network model for lightning prediction,” in *Proc. ACM KDD*, 2019, pp. 2439–2447.

[14] C. Price and D. Rind, “A simple lightning parameterization for calculating global lightning distributions,” *Journal of Geophysical Research: Atmospheres*, vol. 97, no. D9, pp. 9919–9933, 1992.

[15] N. Michalon *et al.*, “Contribution to the climatological study of lightning,” *Geophysical research letters*, vol. 26, no. 20, pp. 3097–3100, 1999.

[16] X. Qie *et al.*, “Application of total-lightning data assimilation in a mesoscale convective system based on the WRF model,” *Atmospheric research*, vol. 145, pp. 255–266, 2014.

[17] X. Shi *et al.*, “Deep learning for precipitation nowcasting: A benchmark and a new model,” in *Proc. NeurIPS*, 2017, pp. 5617–5627.

[18] Z. Xu *et al.*, “Predcnn: Predictive learning with cascade convolutions.” in *Proc. IJCAI*, 2018, pp. 2940–2947.

[19] B. Shen *et al.*, “Stepdeep: A novel spatial-temporal mobility event prediction framework based on deep neural network,” in *Proc. ACM KDD*, 2018, pp. 724–733.

[20] Z. Yuan *et al.*, “Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data,” in *Proc. ACM KDD*, 2018, pp. 984–992.

[21] Z. Pan *et al.*, “Urban traffic prediction from spatio-temporal data using deep meta learning,” in *Proc. ACM KDD*, 2019, pp. 1720–1730.

[22] J. Ye *et al.*, “Co-prediction of multiple transportation demands based on deep spatio-temporal neural network,” in *Proc. ACM KDD*, 2019, pp. 305–313.

[23] W. C. Skamarock *et al.*, “A description of the Advanced Research WRF Version 3,” National Center for Atmospheric Research, Tech. Rep., 2008.

[24] M. Qu *et al.*, “GMNN: Graph Markov neural networks,” in *Proc. ICML*, vol. 97, 2019, pp. 5241–5250.

[25] R. Xiang and J. Neville, “Pseudolikelihood EM for within-network relational learning,” in *Proc. IEEE ICDM*, 2008, pp. 1103–1108.

[26] L. F. Richardson, “The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam,” *Philosophical Transactions of the Royal Society of London. Series A*, vol. 210, no. 459-470, pp. 307–357, 1911.

[27] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. ICLR*, 2015.

[28] Y. Wang *et al.*, “Evaluation of lightning forecasting based on one lightning parameterization scheme and two diagnostic methods,” *Atmosphere*, vol. 9, no. 3, p. 99, 2018.

[29] K. A. Cummings, “An investigation of the parameterized prediction of lightning in cloud-resolved convection and the resulting chemistry,” Ph.D. dissertation, University of Maryland, College Park, 2017.

[30] A. J. Clark *et al.*, “Neighborhood-based verification of precipitation forecasts from convection-allowing ncar WRF model simulations and the operational nam,” *Weather and Forecasting*, vol. 25, no. 5, pp. 1495–1509, 2010.