Hierarchically Structured Transformer Networks for Fine-Grained Spatial Event Forecasting

Xian Wu[†*]
Chao Huang[§*]
xwu9@nd.edu,chaohuang75@gmail.com
University of Notre Dame[†]
JD Finance America Corporation[§]

Chuxu Zhang
Department of Computer Science and Engineering
University of Notre Dame
czhang11@nd.edu

Nitesh V. Chawla
Department of Computer Science and Engineering
University of Notre Dame
nchawla@nd.edu

ABSTRACT

Spatial event forecasting is challenging and crucial for urban sensing scenarios, which is beneficial for a wide spectrum of spatial-temporal mining applications, ranging from traffic management, public safety, to environment policy making. In spite of significant progress has been made to solve spatial-temporal prediction problem, most existing deep learning based methods based on a coarse-grained spatial setting and the success of such methods largely relies on data sufficiency. In many real-world applications, predicting events with a fine-grained spatial resolution do play a critical role to provide high discernibility of spatial-temporal data distributions. However, in such cases, applying existing methods will result in weak performance since they may not well capture the quality spatial-temporal representations when training triple instances are highly imbalanced across locations and time.

To tackle this challenge, we develop a hierarchically structured Spatial-Temporal ransformer network (STtrans) which leverages a main embedding space to capture the inter-dependencies across time and space for alleviating the data imbalance issue. In our STtrans framework, the first-stage transformer module discriminates different types of region and time-wise relations. To make the latent spatial-temporal representations be reflective of the relational structure between categories, we further develop a cross-category fusion transformer network to endow STtrans with the capability to preserve the semantic signals in a fully dynamic manner. Finally, an adversarial training strategy is introduced to yield a robust spatial-temporal learning under data imbalance. Extensive experiments on real-world imbalanced spatial-temporal datasets from NYC and Chicago demonstrate the superiority of our method over various state-of-the-art baselines.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; **Data mining**; • **Computing methodologies → Neural networks**;

KEYWORDS

Spatial-temporal data mining, Deep neural networks

ACM Reference Format:
Xian Wu, Chao Huang, Chuxu Zhang, and Nitesh V. Chawla. 2020. Hierarchically Structured Transformer Networks for Fine-Grained Spatial Event Forecasting. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380296

1 INTRODUCTION

Spatial-temporal predictive analytics is one of the most important techniques for many geographical data mining and urban sensing applications, such as traffic management [23], spatial-temporal anomaly detection [42], location inference [11] and recommendation [21]. Among various prediction tasks, predicting spatial events is very important and beneficial for smart city applications [1, 33]. For example, an accurate and reliable spatial event forecasting system in our physical environment could help the government for making decisions and provide early warnings for public safety emergency management [12, 27]. In general, spatial event prediction can be formulated as learning a function that maps predictor variables to the target event occurrences based on historical event observations across time and space.

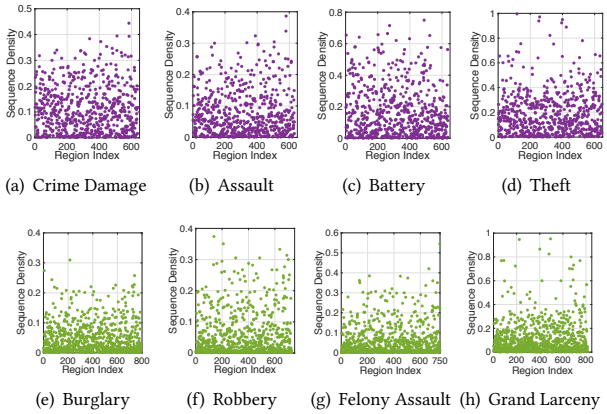
To build predictive models of spatial events at different regions, a common solution is to convert event data observations to a temporally-ordered sequence based on location information. Thereafter, standard time series forecasting techniques such as autoregressive integrated moving average (ARIMA) [26] and support vector machine (SVR) [3] can be applied. However, the spatial-temporal data involves dynamic and non-linear time-ordered dependencies across time steps, which poses difficulties to conventional time series modeling methods–relying on the stationary and linear assumption of time-ordered data [13, 24, 43].

To mitigate this issue, various types of deep neural network models have been introduced to consider the time-varying and non-linear spatial-temporal patterns. For example, recurrent neural network-based approaches have been proposed to model the complicated correlations between past and future states of spatial-temporal data [19, 40]. In addition, several studies model time-stamped data by combining the recurrent neural network (RNN) and convolutional neural network (CNN), to capture both the intra-series and inter-series correlations [34, 36]. Furthermore, another research line, which utilizes attention mechanisms to identify relevance across time steps [7, 15], has also been investigated.

[*]Both authors contributed equally to this work.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380296



(a) Crime Damage (b) Assault (c) Battery (d) Theft

(e) Burglary (f) Robbery (g) Felony Assault (h) Grand Larceny

Figure 1: Data imbalance analysis of each partitioned geographical region in terms of its sequence density. (a)-(d): data from Chicago; (e)-(h): data from NYC.

Although the effectiveness of current neural network architectures in spatial-temporal forecasting task with the consideration of various correlations, we point out that methods only make prediction with a coarse-grained geographical setting and are not sufficient to yield satisfactory spatial-temporal representations for the fine-grained forecasting scenario, *i.e.*, with the aim of providing high discernibility of spatial-temporal data distributions. Hence, the assumption of data sufficiency for training, which is key for deep neural network techniques, is not realistic. Many practical spatial-temporal forecasting scenarios involve imbalanced and high-dimensional data [38]. For example, abnormal events with multi-categories can only happen at a small number of regions across the entire city. The finer the granularity for geographical region discernment, making the already imbalanced spatial-temporal data even worse [45].

To get a better understanding of spatial event distributions across regions of a city, we show the distributions of data imbalance with respect to the sequence density degrees of each partitioned geographical region in Figure 1. In particular, we partition the New York City and Chicago using a $1km \times 1km$ grid map into disjoint 830 and 664 geographical regions, respectively. The smaller sequence density value (*i.e.*, the ratio of non-zero values) indicates that the more sparse the region's event occurrence sequence is. We can observe that most of density degrees belongs to the range of (0.0,0.1], which suggests the highly imbalanced phenomenon (*i.e.*, irregular and rare event occurrences) of predicted sequences of spatial events. In the context of imbalanced spatial-temporal sequences, current deep neural network solutions may behave poorly and fail to recognize limited number of future events, due to the overfitting phenomenon on the test set.

Therefore, when studying the spatial event forecasting problem in a more fine-grained spatial scenario, we end up with a learning scenario under data imbalance. To build effective spatial-temporal predictive models with such imbalanced data, it is crucial to account for mutli-dimensional inter-dependencies across regions (spatial), time steps (temporal) and categories (semantic), to alleviate data imbalance in the forecasting task. While one solution in predicting spatial events could be uncovering effective external factors (*e.g.*, meteorological conditions [46]) based on hand-engineering domain-specific features, it is difficult to discover accurate external data sources which can be generalized and applied to different categories of events without domain-specific expert knowledge. As a result, it is necessary to design an automatic learning framework for all categorical spatial event data, significantly reducing the effort of hand-crafted feature engineering.

With the motivations above, this work proposes a general and flexible framework–Dual-Stage Spatial-Temporal Transformer network (STtrans)–for uncovering cross-modal correlation structures from imbalanced multi-dimensional spatial-temporal data. In particular, at the first stage, we develop a spatial-temporal transformer network to jointly preserve the inter-region correlations and intra-region dependencies. Furthermore, at the second stage, to capture the inherent influences among categories, we propose a cross-category transformer network to promote the collaboration of different semantic views. Our STtrans model is able to automatically capture the contribution of correlated regions, time steps and categories from the spatial-temporal-semantic view in the predictive framework on imbalanced spatial-temporal data.

The main contributions can be summarized as follows:

- This work explores the problem of fine-grained spatial event prediction from the viewpoint of hierarchical structure relation learning under data imbalance, empowering it to effectively model time-evolving multi-dimensional spatial-temporal data.
- We develop a hierarchically structured spatial-temporal transformer network to learn the latent region-time-category interactions shared in multi-dimensional spatial-temporal latent space, and effectively leverage cross-modal knowledge to guide the spatial-temporal embedding process.
- To integrate dynamic spatial, temporal and semantic dependencies, we devise a dual-stage transformer network in which: i) the first-stage transformer architecture discriminates which types of spatial-temporal correlations affect more on the target region and time steps; ii) the second-stage transformer fusion network estimates the contribution of each event category in assisting the forecasting task. Finally, an adversarial training strategy is introduced for robust spatial-temporal learning.
- Through extensive experiments performed on real-world spatial-temporal data collected from NYC and Chicago, our results demonstrate that STtrans achieves better performance as compared to state-of-the-art methods with different experimental settings.

The remainder of this work is presented with the following organization. We first introduce some preliminaries and formally define the studied problem in Section 2. We describe the details of our developed STtrans framework in Section 3. The evaluation results are presented in Section 4. In Section 5, we discuss the related work. Finally, we conclude this paper in Section 6.

2 PRELIMINARY AND PROBLEM STATEMENT

In this section, we first introduce key preliminary definitions and then present the goal of our work.

Definition 1. ***Spatial Region.*** *We consider a city as an area of interest and divide it into a $M \times N$ grid map based on the geographical coordinates (longitude and latitude), where each grid is regarded as an individual spatial region. In our prediction scenario, we target at fine-grained spatial resolution for geographical region scale (i.e., large M and N) to obtain a more detailed spatial-temporal data map.*

Definition 2. ***Spatial Event Tensor.*** *After the grid-based city partition, each spatial region is our target geographical unit for making prediction. Suppose there are $|R|$ regions, we could represent their spatial event occurrence observations during past T time steps (e.g.., days) with a generated four-way tensor Y, i.e., 1^{th} and 2^{nd} dimension–region (m, n); 3^{rd} dimension–C categories indexed by c; 4^{th} dimension–time step (e.g.., days). In Y, each element $y^{c,t}_{m,n} = 1$ if there exist c-th categorical data observations (e.g., crimes) from region $r_{m,n}$ at t-th time step and $y^{c,t}_{m,n} = 0$ otherwise.*

With the above settings of fine-grained spatial resolution, we note that the generated spatial event tensor Y is very imbalanced–time steps without any spatial-temporal data observations (*i.e.*, $y^{c,t}_{m,n} = 0$) occupy the majority of time periods, when data imbalance phenomenon exists.

Problem Statement. Given the aforementioned definitions, we formulate the problem of fine-grained spatial event prediction as follows. **Input**: The spatial event tensor Y from all geographical regions $r_{m,n} \in R$ during past T time steps. **Output**: A predictive model to estimate the likelihood that a region has the c-th category of event at the future time step:

$$\hat{y}^{c,(T+1)}_{m,n} = f(Y)\ r_{m,n} \in R; c \in C \tag{1}$$

where $\hat{y}^{c,(T+1)}_{m,n} \in [0, 1]$ denotes the probability of event occurrences with the category c at region $r_{m,n}$ in the future $(T + 1)$ step.

3 METHODOLOGY

In this section, we describe the technical details of our STtrans method which is a hierarchically structured transformer network (as shown in Figure 2), *i.e.*, (i) spatial-temporal transformer network to jointly capture complex inter-region spatial correlations and dynamic intra-region temporal dependencies; (ii) cross-category transformer network that is capable of adaptively uncovering dynamic dependencies across different categories. Each component of STtrans is elaborated in the following subsections.

3.1 Spatial-Temporal Transformer

The spatial-temporal transformer network aims to aggregate contextual signals with the representation transformation process from both the location and time dimensions.

Definition 3. **Spatial-Temporal Embedding Tensor $\mathcal{A}_{m,n}$.** *We define a three-way tensor $\mathcal{A}_{m,n}$ to describe the spatial event occurrence distributions of each region $r_{m,n}$ across all C categories. $\mathcal{A}_{m,n} \in \mathbb{R}^{L \times L \times T}$, where L and T denotes the number of grids and time slots, respectively. Hence, the generated tensor $\mathcal{A}_{m,n}$ includes the information from $L \times L$ spatially nearby regions (L grid rows and L grid columns) of region $r_{m,n}$. Particularly, each element in $\mathcal{A}^{c,t}_{m,n}$ is the concatenation of region $r_{m,n}$'s embedding vector $E_{r_{m,n}}$ and the embedding vector E_t corresponds to the occurrence time information of t-th observation during previous T time steps.*

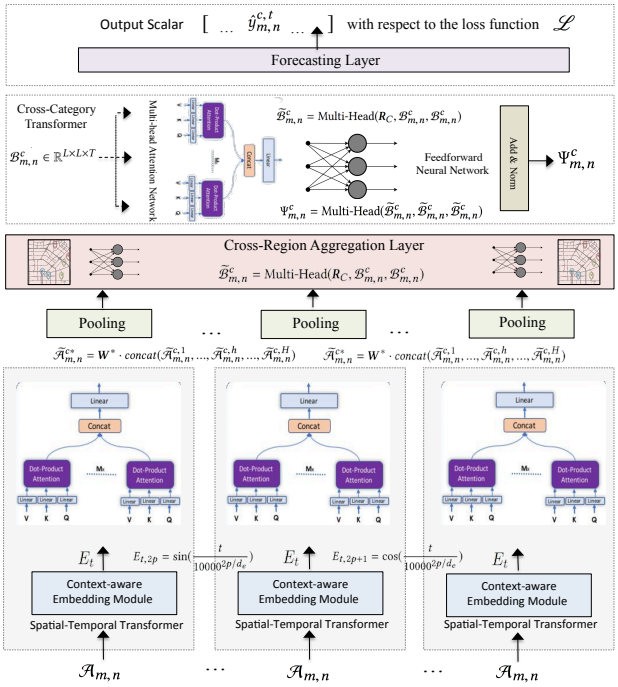


Figure 2: The STtrans Framework.

In order to consider temporal contextual signals in our STtrans framework, we devise a embedding component to generate a embedding vector for each time slot. Specifically, our embedding module leverages the relative time difference between each previous time step (corresponding to the t-th data point) and the current one. Each element of this embedding vector is associated with a non-trainable date embedding with the utilization of the timing signal method [32]. Formally, we present the specific derivations of the embedding vector E_t for the t-th spatial event as below:

$$E_{t,2p} = \sin(\frac{t}{10000^{2p/d_e}})$$
$$E_{t,2p+1} = \cos(\frac{t}{10000^{2p/d_e}}) \tag{2}$$

where d_e and t denotes the embedding dimensionality and relative time value, respectively. We define $2p$ and $2p + 1$ to respectively represent the embedding element index with the even and odd position.

3.1.1 **Multi-Head Transformation Network.** Attention-based neural network models have been shown to provide effective performance in capturing correlations between latent representations without the consideration of sequentially propagate relationships. In this work, we propose to jointly model the spatial and temporal

inter-dependencies among regions and time slots by performing the attentive learning within multiple representation learning subspaces. We aim to jointly attend to relevant signals from other geographical regions at different positions and time steps in the temporally-ordered sequences.

Towards the above goal, during the representation transformation process, we propose a multi-head attention mechanism to automatically learn the quantitative relevance across different regions (*i.e.*, $r_{m,n}$) and occurrence time t (*i.e.*, $t \in [1, ..., T]$). Specifically, given the generated spatial-temporal embedding tensor $\mathcal{A}^c_{m,n}$ of the target region $r_{m,n}$ with respect to the c-th category, the designed M-head attention mechanism performs the aggregation learning process with H times which is indexed by h. For each attentive learning component, we apply the self-attention mechanism on $\mathcal{A}^l_{i,j}$, which can be formally represented with the following formulas:

$$\widetilde{\mathcal{A}}^c_{m,n} = \text{softmax}(\frac{\boldsymbol{W}_Q \cdot \mathcal{A}^c_{m,n}(\boldsymbol{W}_K \cdot \mathcal{A}^c_{m,m})^T}{\sqrt{d_Q}})\boldsymbol{W}_V \cdot \mathcal{A}^c_{m,n} \quad (3)$$

where $\widetilde{\mathcal{A}}^c_{m,n}$ represents the recalibrated representation of region $r_{m,n}$ for the c-th category. $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, and $\boldsymbol{W}_V$ represent the trainable matrices of h-th head attention mechanism corresponding to the queries, keys, and values in the attention mechanism [32]. d_Q is defined as the embedding dimension size of $\boldsymbol{W}_Q$ and the scale factor $\sqrt{d_Q}$ is used to avoid overly large values of the inner product.

After the pattern encoding process of each self-attention mechanism, we concatentate each recalibrated embedding vector $\widetilde{\mathcal{A}}^{c,h}_{m,n}$ from each h-th learning subspace. Then, the multi-head learning process can be formally given with the following formula:

$$\widetilde{\mathcal{A}}^{c*}_{m,n} = \boldsymbol{W}^* \cdot concat(\widetilde{\mathcal{A}}^{c,1}_{m,n}, ..., \widetilde{\mathcal{A}}^{c,h}_{m,n}, ..., \widetilde{\mathcal{A}}^{c,H}_{m,n}) \quad (4)$$

$W^* \in \mathbb{R}^{d_e \times d_e}$ is the learnable parameter. By doing so, we could jointly embed inter-dependency units into the shared spatial-temporal space under the multi-head attention architecture. The fused embedding $\widetilde{\mathcal{A}}^{c*}_{m,n}$ is then fused into a feed-forward neural network to generate the output $\mathcal{B}^c_{m,n} \in \mathbb{R}^{L \times L \times T}$ as follows:

$$\mathcal{B}^c_{m,n} = \mathbf{W}^f_2 \cdot \text{ReLU}(\mathbf{W}^f_1 \cdot \widetilde{\mathcal{A}}^{c*}_{m,n} + \mathbf{b}^f_1) + \mathbf{b}^f_2, \quad (5)$$

where $\mathbf{W}^f_1$, $\mathbf{W}^f_2$ and $\mathbf{b}^f_1$, $\mathbf{b}^f_2$ are the learnable parameters in the feed-forward network. The multi-head transformation network learns the embedding of a region by comparing the closeness between pairwise neighboring regions. For simplicity, we denote the multi-head attention as: Multi-Head(·) in the following subsections.

3.2 Cross-Category Fusion Transformer

In this subsection, we show how to encode the semantic signals in our predictive solution STtrans, by modeling the category-specific inter-dependencies with a cross-category fusion transformer network. Our cross-category transformer aims to model the inter-dependencies between category-specific spatial event occurrence patterns. In particular, we generate a latent representation $\mathcal{B}^c_{m,n}$ for the c-th category learned from our spatial-temporal transformer module. We first perform the average pooling on $\mathcal{B}^c_{m,n}$ over the time dimension, and further aggregate the learned representation $\mathcal{B}^c_{m,n}$ over the region dimension with a multi-head attention operation, which is formally presented as follows:

$$\widetilde{\mathcal{B}}^c_{m,n} = \text{Multi-Head}(\boldsymbol{R}_C, \mathcal{B}^c_{m,n}, \mathcal{B}^c_{m,n}) \quad (6)$$

where $\boldsymbol{R}_C$ denotes the set of C embedding vectors in which each represents the embedding concatenation of region $r_{m,n}$ and c-th category. After that, we could generate C embedding vectors (corresponding to C categories) for the target region $r_{m,n}$. Then, we feed $\widetilde{\mathcal{B}}^c_{m,n}$ into another multi-head attention to capture the cross-category dependencies for learning conclusive representations as follows:

$$\Psi^c_{m,n} = \text{Multi-Head}(\widetilde{\mathcal{B}}^c_{m,n}, \widetilde{\mathcal{B}}^c_{m,n}, \widetilde{\mathcal{B}}^c_{m,n}) \quad (7)$$

After the transformer layers that hierarchically exchange information across all regions, time steps and categories, we obtain the final latent representations $\Psi^c_{m,n}$ which correspond to the region $r_{m,n}$ and c-th category. The learned representations preserve implicit dynamic dependencies between region $r_{m,n}$ and other relevant regions with respect to the category-specific spatial event occurrence in a fully time-evolving environment.

3.3 Model Optimization

During the forecasting phase, we integrate Multilayer Perceptron (MLP) with dropout technique [31] to generate the occurrence probability by capturing the element-wise non-linear dependencies prevent the overfitting while training neural networks. Our prediction module randomly drops out units (hidden and visible) in our neural networks, along with all its incoming and outgoing connected layers. After the utilization of dropout technique, we reduce the inter-dependent learning amongst the neurons, and the dropout operation offers a remarkably effective regularization method to improve the generalization error in our neural network architecture. In our forecasting component, we utilize the *ReLU* and *Sigmod* as the activation function for the fully connected layers and the output layer, respectively. We finally output the event occurrence probability corresponds to specific region, category and time step.

Based on the cross entropy-based metric, we define our loss function as follows:

$$\mathscr{L} = -\sum_{(m,n,c,t) \in S} y^{c,t}_{m,n} \log \hat{y}^{c,t}_{m,n} + (1 - y^{c,t}_{m,n}) \log(1 - \hat{y}^{c,t}_{m,n}) \quad (8)$$

where $\hat{y}^{c,t}_{m,n}$ denotes the estimated probability of the c-th category of spatial-temporal event occur at region $r_{m,n}$ in t-th time slot. By minimizing the defined loss function, we can infer the model parameters (model optimization is performed with the Adam optimizer [17]). We choose Adam optimizer because of its advantage for the convergence property in a faster way when compared with other optimization schemes. The general learning process of our STtrans is summarized in Algorithm 1.

3.4 Adversarial Training

The goal of employing adversarial training is to make the prediction system not only suitable for prediction, but also robust to adversarial perturbations, so as to further improve the model robustness on the imbalanced spatial-temporal data. To ensure the high-quality prediction, we utilize cross-validation, shown in Equation 8, as the building block. To enhance the robustness, we enforce the model to perform consistently well even when the adversarial perturbations

Algorithm 1: The Learning Process of STtrans Model.

Input: Spatial-Temporal Embedding Tensor $\mathcal{A}_{m,n}$, Neighbor Region Set G, Category Set L, Sequence Length T, and Batch Size b_{size}.
Paras: Embedding Matrices $\mathbf{e_r} \in \mathbb{R}^{(M*N)\times d_e}$, $\mathbf{e_c} \in \mathbb{R}^{C\times d_e}$, and Other Hidden Parameters θ.

Initialize all parameters;
// Sample a minibatch of size b_{size}.
foreach $\mathsf{T}_{\text{batch}} = \mathtt{sample}(\mathbf{X}, b_{\text{size}})$ **do**
  foreach $\langle m, n, c, t\rangle \in \mathsf{T}_{\text{batch}}$ **do**
    foreach $\langle m', n'\rangle \in G[(m, n)]$ **do**
      $\mathbf{R}_{m,n} = \mathbf{e_r}[(m, n) :]$;
      // Lookup region embeddings
      foreach $c' \in C$ **do**
        $\mathbf{C}_c = \mathbf{e_c}[c, :]$;
        // Lookup category embeddings
        $c^0_{m',n',c'} = c^0$, $h^0_{m',n',c'} = h^0$// init. hidden states
        $\widetilde{\mathcal{A}}^{c*}_{m,n} = \boldsymbol{W}^* \cdot concat(\widetilde{\mathcal{A}}^{c,1}_{m,n}, \ldots, \widetilde{\mathcal{A}}^{c,h}_{m,n}, \ldots, \widetilde{\mathcal{A}}^{c,H}_{m,n})$
        $\widetilde{\mathcal{B}}^{c}_{m,n}$ = Multi-Head($\boldsymbol{R}_C$, $\mathcal{B}^c_{m,n}$, $\mathcal{B}^c_{m,n}$)
        $\Psi^c_{m,n}$ = Multi-Head($\widetilde{\mathcal{B}}^c_{m,n}$, $\widetilde{\mathcal{B}}^c_{m,n}$, $\widetilde{\mathcal{B}}^c_{m,n}$)
      end
    end
    $\hat{y}^{c,t}_{m,n} = \mathtt{MLP}(\Psi^c_{m,n})$ $y^{c,t}_{m,n} = \mathbf{X}[m, n, c, t]$ Update loss $\mathcal{L}$
  end
  Update all parameters w.r.t $\mathcal{L}$;
end

are presented. To achieve this goal, we additionally optimize the model to minimize the objective function with the perturbed parameters. Formally, we define the objective function with adversarial examples incorporated as below:

$$L_{adv}(D|\Theta) = L(D|\Theta) + \lambda L(D|\Theta + \Delta_{adv}),$$
$$\text{where } \Delta_{adv} = \arg\max_{\Delta, \|\Delta\|\leq\epsilon} L(D|\hat{\Theta} + \Delta), \tag{9}$$

where Δ denotes the perturbations on model parameters, $\epsilon \geq 0$ controls the magnitude of the perturbations, and $\hat{\Theta}$ denotes the current model parameters. In this formulation, the adversarial term $L(D|\Theta + \Delta_{adv})$ can be treated as a model regularizer, which stabilizes the ranking performance. We use λ to control the strength of the adversarial regularizer, where the intermediate variable Δ maximizes the objective function to be minimized by Θ. The training process can be expressed as playing a minimax game:

$$\Theta_{opt}, \Delta_{opt} = \arg\min_{\Theta} \max_{\Delta, \|\Delta\|\leq\epsilon} L(D|\Theta) + \lambda L(D|\Theta + \Delta) \tag{10}$$

where the learning algorithm for model parameters Θ is the minimizing player, and the procedure to derive perturbations Δ acts as the maximizing player, which aims to identify the worst-case perturbations against the current model. The two players alternately play the game until convergence.

3.4.1 **Adversarial Perturbation Construction.** Given a training instance (E_{r_j}, L), the problem of constructing adversarial perturbations Δ_{adv} is formulated as maximizing the following function:

$$\ell_{adv}(E_{r_j}, L|\Delta) = \log(1 + \exp(y_{b,u^-}(\hat{\Theta} + \Delta) - t_{b,u^+}(\hat{\Theta} + \Delta))), \tag{11}$$

where $\hat{\Theta}$ denotes a set of current model parameters. As it is difficult to get the exact optimal solutions of Δ_{adv}, we employ the fast gradient method proposed in [9], to estimate Δ_{adv}. The idea is to approximate the objective function around Δ as a linear function. To maximize the approximated linear function, we need to move towards the gradient direction of the objective function with respect to the Δ. With the max-norm constraint $\|\Delta\| \leq \epsilon$, we approximate Δ_{adv} with the following operation:

$$\Delta_{adv} = \epsilon \frac{\tau}{\|\tau\|}, \text{ where } \tau = \frac{\partial \ell_{adv}(E_{r_j}, L|\Delta)}{\partial \Delta}. \tag{12}$$

3.4.2 **Learning Model Parameters.** We now consider how to learn model parameters Θ. The local objective function to minimize for a training instance (b, u^+, u^-) is shown as follows:

$$\begin{aligned}\ell_{adv}(b, u^+, u^-|\Theta) &= \log(1 + \exp(t_{b,u^-}(\Theta) - t_{b,u^+}(\Theta))) \\ &+\lambda \log(1 + \exp(t_{b,u^-}(\Theta + \Delta_{adv}) - t_{b,u^+}(\Theta + \Delta_{adv})))\end{aligned} \tag{13}$$

where Δ_{adv} is obtained from Equation 12. We can obtain the SGD update rule for Θ as follows:

$$\Theta = \Theta - \eta \frac{\partial \ell_{adv}(E_{r_j}, L|\Theta)}{\partial \Theta}, \tag{14}$$

where η denotes the learning rate.

Alg 2 summarizes the adversarial training process of STtrans. In each training step, we first randomly draw an instance (b, u^+, u^-). We then construct adversarial perturbations and optimize model parameters in a sequential order. The training phase involves multiple training steps and stops until reaching a certain number of training epochs. The parameters achieving the best performance on the validation dataset are utilized for evaluations.

Algorithm 2: Parameter Optimizations.

Input: Training instances D, max iteration $iter_{\max}$;
Output: Model parameters Θ
Initialization: initialize Θ with Normal distribution $N(0,0.01)$, $iter = 0$, $\Theta_{opt} = \Theta$, $L_{opt} = L_{vali}$;
repeat
  foreach *training instance* $(b, u^+, u^-) \in D$ **do**
    // Constructing adversarial perturbations;
    $\Delta_{adv} \leftarrow$ Equation 12;
    // Updating model parameters;
    $\Theta \leftarrow$ Equation 14;
  end
  if $L_{vali} < L_{opt}$ **then**
    $L_{opt} = L_{vali}$;
    $\Theta_{opt} = \Theta$;
  end
  $iter++$;
until $iter > iter_{max}$;
Return Θ_{opt};

4 EVALUATION

In this section, we evaluate the proposed *STtrans* framework on two real-world datasets for fine-grained spatial event forecasting, and present the experimental results as compared with different categories of competitive techniques. Particularly, the key research questions we aim to answer are shown as below:

- **RQ1**: Compared with state-of-the-art predictive models, how does *STtrans* perform in spatial event prediction?
- **RQ2**: How does *STtrans* perform *w.r.t* different spatial resolutions in the prediction scenario?

Table 1: Statistics of Experimented Datasets.

Data Source	**Crimes in NYC**	
Category	Burglary	Robbery
# of Instances	31,799	33,453
Category	Felony Assault	Grand Larceny
# of Instances	40,429	85,899
Data Source	**Crimes in Chicago**	
Category	Theft	Battery
# of Instances	124,630	99,389
Category	Criminal Damage	Felony Assault
# of Instances	59,886	37,972

- **RQ3**: Does *STtrans* consistently outperform other baselines in terms of prediction accuracy *w.r.t* different time windows with different training and testing time periods?
- **RQ4**: How does our *STtrans* model work for forecasting different categories of spatial events?
- **RQ5**: How do different modules (*e.g.*, cross-region aggregation component and category-wise transformer module) affect the prediction performance of *STtrans*?
- **RQ6**: What is the influence of hyperparameters settings in the developed *STtrans* framework.

In the following subsections, we first describe the experimental settings. We then report results by answering the above research questions in turn.

4.1 Experimental Settings

4.1.1 **Data Description.** To validate the effectiveness of *STtrans*, we utilize two collected abnormal event datasets in the urban space which provide us a good opportunity to investigate the model performance in a real-world fine-grained spatial-temporal prediction scenario, due to the low occurrence of abnormal events. We further present the details of those datasets as follows:

- **(NYC-Data)**: This data is collected from the crime report service in New York. Wherein, the crime observations are splited into 4 different categories (*i.e.*, Burglary, Robbery, Assault and Grand Larceny). Every record in the data consists of crime category, timestamp and GPS coordinates. We normalize the timestamp information into one day with keeping the original order of crime occurrences.
- **(Chicago-Data)**: This is another crime dataset collected from Chicago. Each crime category is regarded as an individual target class to fit our prediction scenario. Similarly, we use the day as the time interval to map the original crime data into occurrence sequences.

To comprehensively evaluate the prediction performance with various spatial settings in urban space, We adopt the grid-based mapping strategy with two different spatial resolutions to study the spatial event prediction task, in which grids are regarded as regions. In particular, we partition the New York City with $1km \times 1km$ and $2km \times 2km$ grid into 830 and 248 spatial regions, respectively. Similarly, the Chicago geographical area is divided into disjoint 664 and 189 regions accordingly. We show the distributions of different categorical abnormal events over different time periods in Figure 3.

4.1.2 **Evaluation Protocols.** For the data partition during the performance comparison part, we constitute the training and validation set with the ratio of 5:1 from spatial-temporal sequence data of six months in chronological order, where validation set is utilized for tune hyper-parameters. The spatial-temporal data of one month just after the validation set is treated as the test set. For fair consideration and avoiding the evaluation bias, we target at consecutive days in the test time period and the reported performance is averaged across all days.

4.1.3 **Evaluation Metrics.** We leverage two types of evaluation metrics for the performance validation (*i.e.*, performance across categories and category-specific performance) which can be summarized as below:

Performance across Categories: To evaluate the overall prediction accuracy of all compared approaches, we use *Micro-F1* and *Macro-F1* [37] as the performance metrics across all data categories. Formally, *Micro-F1* and *Macro-F1* can be formally represented as:

$$\begin{aligned} \text{Micro-F1} &= \frac{1}{C} \cdot \sum_{c=1}^{C} \frac{2 \cdot TP_c}{2 \cdot TP_c + FN_c + FP_c} \\ \text{Macro-F1} &= \frac{2 \cdot \sum_{c=1}^{C} TP_c}{2 \cdot \sum_{c=1}^{C} TP_c + \sum_{c=1}^{C} FN_c + \sum_{c=1}^{C} FP_c} \end{aligned} \tag{15}$$

where TP, FP, FN indicates true positive, false positive and false negative instances. Note that a higher Micro-F1 and Macro-F1 score indicates better prediction performance.

Category-specific Performance. To investigate the performance in forecasting each individual category of all the models, we employ a variety of evaluation metrics, including F1-score and Area Under the ROC Curve (AUC).

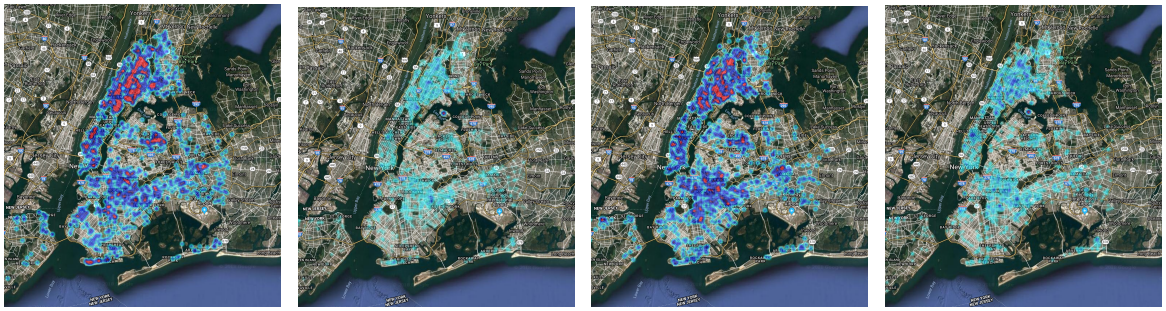
4.1.4 **Methods for Comparison.** We compared it with the following state-of-the-art methods from various research lines:

Conventional Time Series Modeling Techniques. In the experiments, we first make comparison with two representative conventional time series forecasting techniques (*i.e.*, SVR and ARIMA).

- **Support Vector Regression (SVR)** [3]: it has been used for time series prediction via mapping the input data into a high dimensional feature space with a nonlinear function.
- **Auto-Regressive Integrated Moving Average (ARIMA)** [26]: it is a widely used time series analysis model which gauges the relation strength of one dependent variable relevant to other varying variables from the time series data.

RNN-based Methods for Spatial-Temporal Forecasting. RNNs have been introduced to deal with sequential data, we consider the following RNN-based learning approaches in spatial-temporal data forecasting.

- **Spatial-Temporal Recurrent Neural Network (ST-RNN)** [24]: this method models the spatial and temporal contextual information from the geo-tagged sequence data with the recurrent neural network architecture.



(a) Robbery–Aug (b) Felony Assault–Aug (c) Robbery–Nov (d) Felony Assault–Nov

Figure 3: Distributions of spatial events across NYC.

- **Deep Recurrent Networks (DRN)** [40]: DRN proposes to stack several LSTM layers to predict the future values of spatial-temporal data with the consideration of both the normal and extreme sequential transitional conditions.

Attentive Spatial-Temporal Data Analytic Techniques. We also consider another type of models that encodes the sequential spatial-temporal information through combining the recurrent neural network and attentive learning framework.

- **Spatial-Temporal Fusion Forecasting (DCrime)** [15]: a fusion framework which leverages attention mechanisms to consider correlations across time, and uses the recurrent neural network to encode temporal dependencies of the target spatial-temporal sequences.
- **Attentive Recurrent Model (ARM)** [7]: it is an attention-based recurrent network to model of periodicity information from historical spatial-temporal data.

Convolutional Recurrent Learning Framework: In the performance comparison, we also include the deep spatial-temporal predictive approaches with the integration of convolutional and recurrent neural networks.

- **Convolutional Recurrent Networks (Conv-R)** [44]: it employs convolution-based networks to model nearby spatial dependencies and encodes the temporal signals with recurrent networks.

Neural Matrix Factorization. Finally, we compare our *STtrans* with a matrix factorization model based on the neural network architecture for multi-dimensional interaction prediction.

- **Neural Matrix Factorization (NeurMF)** [10]: We extend this Matrix Factorization scheme with neural network architecture to handle multi-dimensional spatial-temporal data.

4.1.5 **Parameter Settings and Reproductivity.** In our experiments, we leverage Adam [20] as the optimizer for all neural network models and implement the proposed *STtrans* framework with TensorFlow architecture. The sequence length and hidden state dimension is set to 10 and 64, respectively. Additionally, the batch size and learning rate is set to 64 and $1e^{-3}$, respectively. SVR and ARIMA are implemented based on the statsmodels library [29]. The early stopping [28] is adopted to terminate the training process based on the validation performance. We perform training process on all the models from scratch without pre-training on a single NVIDIA GeForce GTX 1080.

Table 2: Prediction results across different categories in terms of *Macro-F1* and *Micro-F1*.

Dataset	NYC-Data		Chicago-Data	
Granularity	$1km \times 1km$ grid-based partition			
Metrics	Micro-F1	Macro-F1	Micro-F1	Macro-F1
ARIMA	0.1614	0.2138	0.1953	0.2585
SVR	0.1619	0.2090	0.1934	0.2535
ST-RNN	0.2323	0.2579	0.2589	0.3196
DRN	0.2012	0.2521	0.2406	0.3050
Conv-R	0.2346	0.2608	0.2844	0.3257
NeurMF	0.2322	0.2567	0.2517	0.3184
DCrime	0.2377	0.2636	0.2642	0.3280
ARN	0.2328	0.2617	0.2633	0.3207
STtrans	**0.2562**	**0.2943**	**0.3211**	**0.3580**
Dataset	NYC-Data		Chicago-Data	
Granularity	$2km \times 2km$ grid-based partition			
Metrics	Micro-F1	Macro-F1	Micro-F1	Macro-F1
ARIMA	0.3293	0.4156	0.3382	0.5162
SVR	0.3143	0.4100	0.3225	0.5101
ST-RNN	0.4428	0.4954	0.3968	0.4956
DRN	0.4548	0.5008	0.3686	0.5238
Conv-R	0.4501	0.5002	0.3962	0.5061
NeurMF	0.4376	0.4698	0.3722	0.5258
DCrime	0.4576	0.4832	0.3751	0.5280
ARN	0.3147	0.4418	0.3698	0.5236
STtrans	**0.4901**	**0.5207**	**0.4660**	**0.5543**

4.2 Performance Comparison

In this subsection, we make comparison between our *STtrans* framework with other baselines.

4.2.1 **Overall Performance (RQ1).** Table 2 shows the spatial event forecasting accuracy across different categories on two datasets in terms of Macro-F1 and Micro-F1. The reported results are averaged on all days in the predicted months (*i.e.*, Jan, Mar, May, Jul, Sep and Nov). We can observe that *STtrans* achieves the best performance and obtains high improvements over different types of baselines in all cases. This sheds lights on the benefit of our model

Table 3: Forecasting results v.s. different time periods in terms of *Micro-F1* and *Macro-F1*.

City	Month	Metrics	ARIMA	SVR	ST-RNN	DRN	DCrime	Conv-R	NeurMF	ARN	*STtrans*
NYC Data	Jan	Micro-F1	0.1652	0.1631	0.2236	0.1998	0.2517	0.2470	0.2409	0.2225	**0.2632**
		Macro-F1	0.2175	0.2157	0.2578	0.2469	0.2667	0.2627	0.2563	0.2620	**0.2905**
	Mar	Micro-F1	0.1567	0.1533	0.2239	0.2017	0.2362	0.2291	0.2345	0.2231	**0.2318**
		Macro-F1	0.2087	0.2054	0.2519	0.2504	0.2619	0.2538	0.2588	0.2616	**0.2873**
	May	Micro-F1	0.1656	0.1576	0.2349	0.1899	0.2389	0.2442	0.2255	0.2287	**0.2443**
		Macro-F1	0.2182	0.2108	0.2566	0.2477	0.2609	0.2584	0.2472	0.2586	**0.2884**
	Jul	Micro-F1	0.1601	0.1586	0.2429	0.2022	0.2552	0.2474	0.2359	0.2387	**0.2672**
		Macro-F1	0.2124	0.2113	0.2667	0.2533	0.2683	0.2622	0.2613	0.2595	**0.3001**
	Sep	Micro-F1	0.1586	0.1525	0.2286	0.2001	0.2166	0.2079	0.2286	0.2398	**0.2561**
		Macro-F1	0.2113	0.2049	0.2585	0.2509	0.2659	0.2582	0.2584	0.2638	**0.2933**
	Nov	Micro-F1	0.1626	0.1867	0.2397	0.2139	0.2079	0.2218	0.2286	0.2442	**0.2747**
		Macro-F1	0.2148	0.2065	0.2560	0.2637	0.2582	0.2696	0.2584	0.2652	**0.3063**
CHI Data	Jan	Micro-F1	0.1924	0.1713	0.2397	0.2357	0.2527	0.2539	0.2418	0.2541	**0.3274**
		Macro-F1	0.2577	0.2318	0.3052	0.3017	0.3177	0.3096	0.3111	0.3110	**0.3596**
	Mar	Micro-F1	0.1918	0.1804	0.2489	0.2332	0.2491	0.2483	0.2305	0.2543	**0.3021**
		Macro-F1	0.2484	0.2360	0.3055	0.2902	0.3136	0.3062	0.3004	0.3112	**0.3397**
	May	Micro-F1	0.1946	0.2160	0.2634	0.2484	0.2660	0.2628	0.2596	0.2625	**0.3277**
		Macro-F1	0.2529	0.2775	0.3223	0.3099	0.3336	0.3284	0.3279	0.3218	**0.3693**
	Jul	Micro-F1	0.1960	0.2123	0.2573	0.2565	0.2881	0.2809	0.2676	0.2707	**0.3303**
		Macro-F1	0.2614	0.2734	0.3229	0.3219	0.3438	0.3425	0.3300	0.3169	**0.3723**
	Sep	Micro-F1	0.2023	0.1861	0.2637	0.2398	0.2613	0.2641	0.2564	0.2630	**0.3212**
		Macro-F1	0.2672	0.2494	0.3261	0.3059	0.3293	0.3206	0.3218	0.3272	**0.3547**
	Nov	Micro-F1	0.1949	0.1946	0.2806	0.2305	0.2680	0.2691	0.2545	0.2753	**0.3183**
		Macro-F1	0.2636	0.2529	0.3359	0.3007	0.3305	0.3305	0.3196	0.3362	**0.3524**

which jointly captures spatial-temporal-categorical dependencies and effectively alleviates the data imbalance issue in the generated spatial event tensor Y.

4.2.2 **Performance Gain Analysis.** Given the significant performance improvement between *STtrans* and other compared baselines, we present the following summarization:

The evaluation results shed light on the limitations of conventional time series forecasting methods (*i.e.*, ARIMA and SVR) which assume fixed temporal patterns in modeling time-ordered spatial-temporal sequences. The performance gain between *STtrans* and matrix factorization based method (*i.e.*, NeurMF) stem from: (i) explicitly modeling temporal dynamics of latent factors underlying spatial event occurrences; (ii) generating better hidden representations to encode the non-linear interactions between different dimensions (*i.e.*, spatial-temporal-semantic).

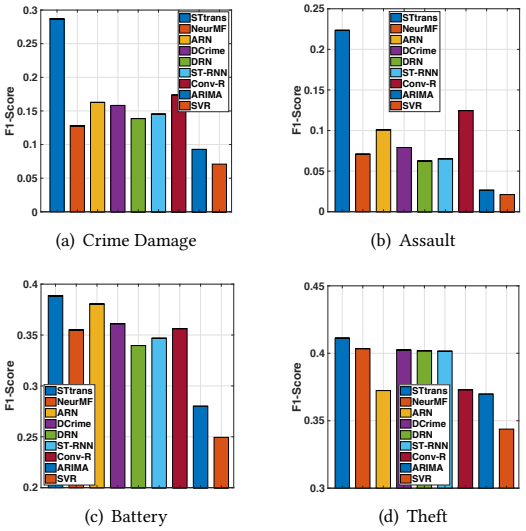
The attentive recurrent models (ARN and DCrime) and convolution based recurrent networks (Conv-R) perform better than recurrent neural network predictive models, which suggests that these recurrent neural architectures (ST-RNN and DRN) only emphasize the temporal dimension and ignore the dynamic region-time-category inter-dependencies. Furthermore, *STtrans* performs distinctly better than attentive recurrent predictive methods and convolutional recurrent networks, suggesting that the deep self-attention mechanism under multiple learning space is a more powerful tool for capturing the spatial-temporal regularities over time, and the designed adversarial training strategy is beneficial for modeling highly imbalanced geographic sequences.

4.2.3 **Performance *w.r.t* Spatial Resolution (RQ2).** Furthermore, we perform experiments to evaluate the performance of predicting spatial event occurrence with respect to different geographical resolutions (*i.e.*, fine-grained and high-level region scale). The evaluation results are presented in Table 2. Note that the prediction task becomes more challenging when we map each spatial event record into a specific fine-grained region (out of 830 and 664 regions) compared to high-level region (368 and 286 regions), since the generated spatial event tensor Y will become more imbalanced by including more zero values, *i.e.*, there are fewer event occurrences when mapping anomaly reports into more fine-grained geographical regions. From the results, we can observe that improvements can still be obtained by *STtrans* with different geographical region granularity as compared to competing baselines.

4.2.4 **Performance *w.r.t* Predicted Periods (RQ3).** Table 3 lists the evaluation results of all compared methods with respect to different training and test time windows using the $1km \times 1km$ grid map partition. Although different time windows reflect a spectrum of temporal diversity which is maintained by month and season variation (*i.e.*, Mar–Spring; May,Jul–Summer; Sep–Fall; Nov,Jan–Winter), our proposed *STtrans* method consistently achieves the best performance by capturing such subtle temporal dynamics of spatial-temporal data. Therefore, the evaluation results across different time frames demonstrate the effectiveness of *STtrans* in modeling time-evolving dependencies in time-stamped anomaly sequences and reasonably interpret the importance of past anomaly occurrences when predicting anomalies in future time slots.

4.2.5 **Category-Specific Performance (RQ4).** We further perform experiments to evaluate *STtrans* in predicting individual spatial event categories. The results are shown in Figure 4 (on Chicago

Crime data with $1km \times 1km$ grid map measured by F1), and in Figure 5 (on NYC-Data with $2km \times 2km$ grid map measured by AUC). Overall, our proposed *STtrans* predictive system outperforms state-of-the-art methods in most cases. The advantage of the proposed *STtrans* lies in its joint consideration and accommodation of dynamic intra-series and inter-series dependencies within the context of spatial and temporal signals. Most importantly, the performance gap between *STtrans* and other baselines increases as the data becomes sparser, which implies that the cross-modal knowledge is more helpful for handling sparse data. Due to space limit, we only present the forecasting results for Chicago-Data and NYC-Data with $2km \times 2km$ spatial resolution. Similar results could be observed for other time periods and datasets.



(a) Crime Damage (b) Assault

(c) Battery (d) Theft

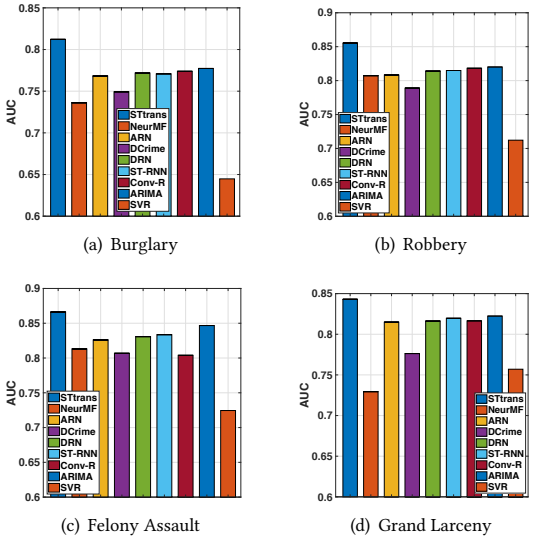
Figure 4: Prediction results for individual category on Chicago.

4.3 Model Abalation Study of *STtrans* (RQ6)

We further perform ablation experiments for the key components of *STtrans*, namely, cross-region aggregation module, time-wise aggregation layer and category-wise transformer fusion network, with the following model variants:

- **Efficacy of cross-region aggregation module**. *STtrans*-s: a simplified architecture of *STtrans* without the cross-region aggregation layer to capture the correlations among geographical regions.
- **Influence of time-wise aggregation layer**. *STtrans*-t: another variant of *STtrans* without performing pooling operation over time dimension before feeding the recalibrated representations into the cross-category transformer.
- **Impact of category-wise transformer network**. *STtrans*-c: this variant does not include the cross-category transformer to capture the latent dependencies among categories.

Figure 6 list the evaluation results of all compared variants in predicting spatial event of individual category with $2km \times 2km$ grid



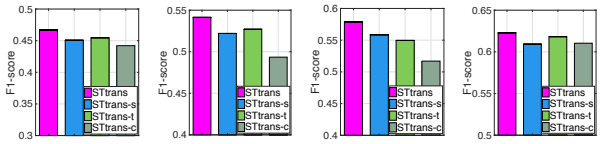
(a) Burglary (b) Robbery

(c) Felony Assault (d) Grand Larceny

Figure 5: Prediction results of individual category on NYC.

granularity. We can observe that the best performance is achieved by our proposed method *STtrans-F* (with the integration of the above modules) in most cases, and analyze their effects respectively as below:

- (1) Overall, *STtrans* achieves better performance as compared to the variant STtrans-s. This observation justifies the effectiveness of our interaction attention mechanism in capturing region-wise influence with respect to spatial event distributions.
- (2) The performance gain between *STtrans* and *STtrans*-t suggests the rationality of our designed temporal aggregation mechanism to encode the unknown relevance of past event occurrences in forecasting future events.
- (3) *STtrans-F* outperforms the variant STtrans-c (without both categorical relation transformer network) in all cases, which further demonstrates the efficacy of our developed fusion model in assisting spatial event prediction task with the careful consideration of global contextual information.



(a) Burglary (b) Robbery (c) Felony Assault (d) Grand Larceny

Figure 6: Model Abalation Study of STtrans.

4.4 Hyperparameter Studies of *STtrans* (RQ7)

Figure 7 shows the evaluation results (measured by Macro-F1 and Micro-F1) as a function of one selected hyperparameter while keeping other optimal hyper-parameters unchanged. Overall, *STtrans* is

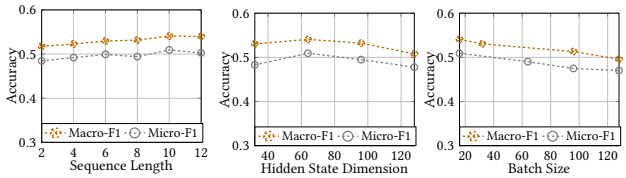


Figure 7: Hyperparameter Studies of *STtrans*.

not very sensitive to these parameters. We first study how the hidden dimensionality (varying from 32 to 128) affect the forecasting performance. We can notice that a larger hidden dimensionality does not always result in better model accuracy, especially on sparse and imbalanced spatial-temporal data, thus marks as the overfitting phenomenon. Additionally, as show in Figure 7, the prediction results of *STtrans* becomes more accurate with the longer encoded sequence (larger value of length T) and tends to saturate when $T = 10$. As the encoded sequence length increases, more extra information and noise is simultaneously introduced into the data modeling process. We can see that *STtrans* performs very stably as the sequence length becomes larger. In our experiments, we set the batch size as 64 due to the balance between efficacy and computational cost, *i.e.*, the larger the batch size is, the less time the training process might take.

5 RELATED WORK

In this section, we briefly review two lines of research works closely related to our studied problem. In particular, we first discuss the work which focuses on forecasting for various spatial-temporal data. Then, we introduce the work which utilizes attention networks on different applications.

5.1 Spatial-Temporal Prediction

Deep learning techniques have been widely used in spatial-temporal mining tasks, such as user traffic data modeling [5], sensor data fusion [25] and air quality data prediction [18]. These applications focus on modeling data at different geographical locations and temporal points and their proposed solutions often rely on sufficient data points collected in chronological order. However, the generated spatial-temporal data can be sparse and of high di when considering a fine-grained spatial resolution. In this work, we effectively transfer knowledge from different data views to alleviate data imbalance issue in spatial-temporal data mining tasks.

Additionally, there exists a rich amount of work on the topic of spatial-temporal prediction with recurrent neural network or its variants, such as traffic prediction [39], location prediction [24], taxi demand forecasting [35] and mobility prediction [30]. These models aim to explore temporal correlations of the generated non-stationary spatial-temporal sequences by learning latent representations based on recurrent frameworks. For example, Yu *et al.* [39] proposed a LSTM-based framework to model the joint effects of normal traffic and accidents. A Recurrent model was developed to capture the temporal and spatial contextual information in user generated geographic sequence [24]. . In addition, many efforts have been devoted to developing hybrid models by integrating RNNs with other advanced deep neural network techniques (*e.g.*, attention mechanisms [7, 14], graph neural network [8] and convolutional diffusion networks [22, 36]).

5.2 Attention Networks

Attention networks have been shown effective in various nature language processing and sequence data modeling tasks, such as text generation [16], machine translation [41], speech recognition [2] and user-item interaction learning [4]. For these attention-based applications, relations between different entities can be aggregated with the learned relevance weights. To learn long-term dependencies of sequence data, self-attention mechanism has been introduced to model sequential patterns directly from any pair of positions in the input sequence data [32]. Later on, BERT [6] was proposed as a pre-training model for sentence representation for various text analysis task based on the text contextual information. However, the singular dimension multi-head self-attention network requires sufficient and balanced training instances and can hardly to be directly applied to encode the occurrence patterns of infrequent spatial events. Our proposed STtrans method is motivated by the self-attention architecture in a sense that a hierarchically structured transformer network is designed to model multi-view relation structures from spatial-temporal sequences under data imbalance.

6 CONCLUSION

In this paper, we develop an effective learning framework, Spatial-Temporal Transformer Network (STtrans), for fine-grained spatial event forecasting. Specifically, we develop a spatial-temporal transformer to jointly preserve the correlations across regions and time. In addition, a cross-category transformer is developed to further augment STtrans with the cross-modal dependency modeling in handing imbalanced spatial-temporal sequences. We perform extensive experiments on real-world spatial-temporal datasets and experimental results show that the proposed model significantly outperforms the state-of-the-art methods across various settings.

One possible direction for future work is to adapt STtrans model to an online learning framework in a dynamic environment where imbalanced spatial-temporal sequential data arrives in a timely manner. One key challenge is to design an effective optimization strategy to achieve the balance between the parameter inference and computation complexity of the streaming algorithm. For example, how to update the model parameters efficiently, rather than performing the training process from scratch, when new spatial-temporal data arrives.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their provided constructive feedback. This work was supported in part by National Science Foundation (NSF) grant IIS-1447795. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Laura Alfers, Phumzile Xulu, and Richard Dobson. 2016. Promoting workplace health and safety in urban public space: reflections from Durban, South Africa. *Environment and Urbanization* 28, 2 (2016), 391–404.

[2] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 4945–4949.

[3] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *Transactions on Intelligent Systems and Technology (TIST)* 2, 3 (2011), 27.

[4] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 335–344.

[5] Quanjun Chen, Xuan Song, Harutoshi Yamada, and Ryosuke Shibasaki. 2016. Learning deep representation from big and heterogeneous data for traffic accident inference. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*.

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

[7] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of International Conference on World Wide Web (WWW)*. ACM, 1459–1468.

[8] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal Multi-Graph Convolution Network for Ride-hailing Demand Forecasting. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*.

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of International Conference on World Wide Web (WWW)*. ACM, 173–182.

[11] Chao Huang, Dong Wang, and Shenglong Zhu. 2017. Where are you from: Home location profiling of crowd sensors from noisy and sparse crowdsourcing data. In *International Conference on Computer Communications (Infocom)*. IEEE, 1–9.

[12] Chao Huang, Xian Wu, and Dong Wang. 2016. Crowdsourcing-based urban anomaly prediction system for smart cities. In *International on Conference on Information and Knowledge Management (CIKM)*. 1969–1972.

[13] Chao Huang, Chuxu Zhang, Peng Dai, and Liefeng Bo. 2019. Deep Dynamic Fusion Network for Traffic Accident Forecasting. In *International Conference on Information and Knowledge Management (CIKM)*. 2673–2681.

[14] Chao Huang, Chuxu Zhang, Jiashu Zhao, Xian Wu, Nitesh Chawla, and Dawei Yin. 2019. MiST: A Multiview and Multimodal Spatial-Temporal Learning Framework for Citywide Abnormal Event Forecasting. In *The World Wide Web Conference (WWW)*. ACM, 717–728.

[15] Chao Huang, Junbo Zhang, Yu Zheng, and Nitesh V Chawla. 2018. DeepCrime: Attentive Hierarchical Recurrent Networks for Crime Prediction. In *Proceedings of International Conference on Information and Knowledge Management (CIKM)*. ACM, 1423–1432.

[16] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2016. Summarizing source code using a neural attention model. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. 2073–2083.

[17] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] İbrahim Kök, Mehmet Ulvi Şimşek, and Suat Özdemir. 2017. A deep learning model for air quality prediction in smart cities. In *Proceedings of International Conference on Big Data (Big Data)*. IEEE, 1983–1990.

[19] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. 2017. Time-series extreme event forecasting with neural networks at uber. In *Proceedings of International Conference on Machine Learning (ICML)*. 1–5.

[20] Quoc V Le, Jiquan Ngiam, Adam Coates, Abhik Lahiri, et al. 2011. On optimization methods for deep learning. In *Proceedings of International Conference on Machine Learning (ICML)*. 265–272.

[21] Ruirui Li, Jyun-Yu Jiang, Chelsea J-T Ju, and Wei Wang. 2019. CORALS: Who Are My Potential New Customers? Tapping into the Wisdom of Customers' Decisions. In *International Conference on Web Search and Data Mining (WSDM)*. 69–77.

[22] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *ICLR*.

[23] Yuxuan Liang, Kun Ouyang, Lin Jing, Sijie Ruan, Ye Liu, Junbo Zhang, David S Rosenblum, and Yu Zheng. 2019. UrbanFM: Inferring Fine-Grained Urban Flows. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. ACM, 3132–3142.

[24] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts.. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*. 194–200.

[25] Zuozhu Liu, Wenyu Zhang, Shaowei Lin, and Tony QS Quek. 2017. Heterogeneous sensor data fusion by deep multimodal encoding. *Journal of Selected Topics in Signal Processing* 11, 3 (2017), 479–491.

[26] Bei Pan, Ugur Demiryurek, et al. 2012. Utilizing real-world transportation data for accurate traffic prediction. In *Proceedings of International Conference on Data Mining (ICDM)*. IEEE, 595–604.

[27] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban Traffic Prediction from Spatio-Temporal Data Using Deep Meta Learning. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM.

[28] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. 2014. Early stopping and non-parametric regression: an optimal data-dependent stopping rule. *Journal of Machine Learning Research (JMLR)* 15, 1 (2014), 335–366.

[29] Skipper Seabold and Josef Perktold. 2010. Statsmodels: Econometric and statistical modeling with python. In *Python in Science Conference*, Vol. 57. SciPy society Austin, 61.

[30] Xuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. 2016. DeepTransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level.. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 16. 2618–2624.

[31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)* 15, 1 (2014), 1929–1958.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of International Conference on Neural Information Processing Systems (NIPS)*. 5998–6008.

[33] Xian Wu, Yuxiao Dong, Chao Huang, Jian Xu, Dong Wang, and Nitesh V Chawla. 2017. Uapd: Predicting urban anomalies from spatial-temporal data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*. Springer, 622–638.

[34] Xian Wu, Baoxu Shi, Yuxiao Dong, Chao Huang, Louis Faust, and Nitesh V Chawla. 2018. RESTFul: Resolution-Aware Forecasting of Behavioral Time Series Data. In *Proceedings of International Conference on Information and Knowledge Management (CIKM)*. ACM, 1073–1082.

[35] Jun Xu, Rouhollah Rahmatizadeh, Ladislau Bölöni, and Damla Turgut. 2017. Real-time prediction of taxi demand using recurrent neural networks. *Transactions on Intelligent Transportation Systems* 19, 8 (2017), 2572–2581.

[36] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, and Jieping Ye. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*. 2588–2595.

[37] Chih-Kuan Yeh, Wei-Chieh Wu, Wei-Jen Ko, and Yu-Chiang Frank Wang. 2017. Learning deep latent space for multi-label classification. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*.

[38] Rose Yu, Dehua Cheng, and Yan Liu. 2015. Accelerated online low rank tensor learning for multivariate spatiotemporal streams. In *Proceedings of International Conference on Machine Learning (ICML)*. 238–247.

[39] Rose Yu, Yaguang Li, Ugur Demiryurek, Cyrus Shahabi, and Yan Liu. 2017. Deep learning: a generic approach for extreme condition traffic forecasting. In *Proceedings of SIAM International Conference on Data Mining (SDM)*. SIAM, 777–785.

[40] Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, and Yan Liu. 2017. Deep learning: A generic approach for extreme condition traffic forecasting. In *Proceedings of SIAM International Conference on Data Mining (SDM)*. SIAM, 777–785.

[41] Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Neural machine translation with deep attention. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)* (2018).

[42] Chao Zhang, Liyuan Liu, Dongming Lei, Quan Yuan, Honglei Zhuang, Tim Hanratty, and Jiawei Han. 2017. Triovecevent: Embedding-based online local event detection in geo-tagged tweet streams. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 595–604.

[43] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 33. 1409–1416.

[44] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[45] Liang Zhao, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2016. Multi-resolution spatial event forecasting in social media. In *Proceedings of International Conference on Data Mining (ICDM)*. IEEE, 689–698.

[46] Guanjie Zheng, Susan Brantely, Thomas Lauvaus, and Li Zhenhui. 2017. Contextual spatial outlier detection with metric learning. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2161–2170.