SpatialRank: Urban Event Ranking with NDCG Optimization on Spatiotemporal Data

Bang An
Department of Business Analytics
University of Iowa
Iowa City, IA 52242
bang-an@uiowa.edu

Xun Zhou *
Department of Business Analytics
University of Iowa
Iowa City, IA 52242
xun-zhou@uiowa.edu

Yongjian Zhong
Department of Computer Science
University of Iowa
Iowa City, IA 52242
yongjian-zhong@uiowa.edu

Tianbao Yang
Department of Computer Science and Engineering
Texas A&M University
College Station, TX 77843
tianbao-yang@tamu.edu

Abstract

The problem of urban event ranking aims at predicting the top-k most risky locations of future events such as traffic accidents and crimes. This problem is of fundamental importance to public safety and urban administration especially when limited resources are available. The problem is, however, challenging due to complex and dynamic spatio-temporal correlations between locations, uneven distribution of urban events in space, and the difficulty to correctly rank nearby locations with similar features. Prior works on event forecasting mostly aim at accurately predicting the actual risk score or counts of events for all the locations. Rankings obtained as such usually have low quality due to prediction errors. Learning-to-rank methods directly optimize measures such as Normalized Discounted Cumulative Gain (NDCG), but cannot handle the spatiotemporal autocorrelation existing among locations. In this paper, we bridge the gap by proposing a novel spatial event ranking approach named SpatialRank. SpatialRank features adaptive graph convolution layers that dynamically learn the spatiotemporal dependencies across locations from data. In addition, the model optimizes through surrogates a hybrid NDCG loss with a spatial component to better rank neighboring spatial locations. We design an importance-sampling with a spatial filtering algorithm to effectively evaluate the loss during training. Comprehensive experiments on three real-world datasets demonstrate that SpatialRank can effectively identify the top riskiest locations of crimes and traffic accidents and outperform state-of-the-art methods in terms of NDCG by up to 12.7%.

1 Introduction

Given a risk score defined based on historical urban events (e.g., crimes, traffic accidents) in a study area as well as the socio-environmental attributes (e.g., travel demands, weather, road network) associated with the events, the goal of the *urban event ranking problem* is to learn a model that can predict the ranking of the top-k riskiest locations of future events.

*corresponding author

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

The urban event ranking problem is a widely observed spatiotemporal prediction problem, which has critical applications in public safety, traffic management, and urban planning. For example, the National Highway Traffic Safety Administration [30], estimated a total of 42,939 deaths in motor vehicle traffic crashes in the United States in 2021. Crime control and property loss cost over $2.5 trillion in the United States in 2017 [28]. Chicago Police Department has utilized criminal intelligence analysis and data science techniques to help command staff determine where best to deploy resources [1]. However, according to the Police Executive Research Forum, there were 42.7% more resignations among law enforcement but a 3.9% decrease in hiring new officers in 2021 compared to 2019 [3]. Meanwhile, the Federal Bureau of Investigation confirms that violent crime in 2020 has surged nearly 30% over 2019 [2]. Therefore, given such growth of crimes and accidents, predicting the riskiest locations of traffic accidents or crimes helps law enforcement stakeholders allocate their limited resources strategically to reduce injuries, deaths, and property losses.

Despite its importance, the urban event ranking problem is technically challenging. **First**, there exist complex and dynamic spatiotemporal correlations between locations in terms of events and attributes such as traffic conditions and weather changes. Capturing such dynamic relationships is non-trivial. **Second**, due to the presence of spatial autocorrelation, nearby locations may share very similar socio-environmental attributes. It is thus very challenging to learn a good model that can correctly rank neighboring locations. **Finally**, urban events are usually sparse in space. Making a prediction with a low average error in count does not ensure a good ranking of the top-k locations.

Prior works on urban event prediction employed either classic machine learning methods [16] [4] [7] [9] [11] [12] [20] or deep learning techniques including recurrent neural networks [33] and convolutional neural networks [29][39]. Recently, Li et al. [23] proposed a cross-region hypergraph structure network to address the data sparsity issues. Besides, Yuan et al. [43] [5] proposed spatial ensembles to address spatial heterogeneity. The objective of the above methods is to make accurate predictions of the risk or count for all locations. However, urban events are often very sparse in space, which might guide the model to avoid predicting accidents in most locations to obtain a low average error. Such prediction does not truly benefit the users such as police officers. The top-k locations derived from such predictions are naturally inaccurate.

Our problem is also relevant to the learning-to-rank problem frequently studied in the field of recommendation systems. State-of-the-art methods in this area typically predict the top-k items most likely to be chosen by users through optimizing ranking-based metrics such as the Normalized Discounted Cumulative Gain (NDCG) [8]. The rank operator of NDCG is non-differentiable, and previous studies have made noticeable progress in approximating NDCG by surrogate functions [35] [31] [37] [32]. In our problem, a straightforward adaptation would be to consider the locations as "items" and each time slot as a "user". However, NDCG is not a perfect objective function for our problems as it neither measures the local ranking quality nor considers spatial autocorrelation among locations. Instead, the existing approaches assume items are independent. Therefore, directly applying existing NDCG optimization solutions might not be the best solution to our problem.

In this paper, we bridge the gaps in both fields by formulating the urban event ranking problem as a spatial learning-to-rank problem and solving it by directly optimizing a "spatial" version of the NDCG measure. We propose **SpatialRank**, our deep learning model with three novel designs. To efficiently capture the spatial and temporal dependencies, we design an adaptive graph convolution layer that learns the correlations between locations dynamically from features and historical events patterns; to ensure the model balances both the global ranking quality on all the locations and the local ranking quality on subsets of locations, we propose a hybrid loss function combining both NDCG loss and a novel local NDCG loss; to improve the effectiveness of optimizing NDCG surrogates in the spatiotemporal setting of our problem, we design an importance-based location sampling with spatial filtering algorithm to iteratively adjust the weights of each location considered in the objective function, thereby guiding the model to concentrate on learning for more important locations. We conduct comprehensive experiments on three real-world datasets collected from Chicago and the state of Iowa. The results demonstrate that SpatialRank can substantially outperform baselines and achieve better ranking quality.

Our contributions are summarized below:

- To the best of our knowledge, this is the first paper to formulate urban event forecasting as a location ranking problem and learn a ranking model by optimizing its ranking quality.
- We propose SpatialRank with adaptive graph convolution layers learning from historical event patterns and spatiotemporal features to capture dynamic correlations over time and space.

- We propose a hybrid objective function to leverage the trade-off between global ranking quality and local ranking quality.
- We propose a ranking-based importance sampling algorithm to adaptively adjust the weights of different locations considered in the objective function of the prediction results from the last training epoch to help the model focus on important locations.

2 Preliminaries

2.1 Formulation of the event ranking problem

A spatio-temporal field $\mathcal{S} \times T$ is a three-dimensional partitioned space, where $T = \{t_1, t_2, ..., t_T\}$ is a study period divided into equal-length intervals (e.g., hours, days) and $\mathcal{S} = \{s_{(0,0)}, ..., s_{(M,N)}\}$ is a $M \times N$ two-dimension spatial grid partitioned from the study area. A set of socio-environmental features F are observed over $S \times T$, which include temporal features $F_t \in \mathbb{R}^T$ (e.g., day of week, holiday), spatial features $F_s \in \mathbb{R}^{M \times N}$ (e.g., total road length, speed limit), and spatiotemporal features $F_{st} \in \mathbb{R}^{M \times N \times T}$ (e.g., traffic volume, rainfall amount). A risk score $y \in \mathbb{R}^{M \times N \times T}$ is a user-defined attribute over $\mathcal{S} \times T$ measuring the risk level of each spatiotemporal location (e.g., number of crimes, injuries of traffic accidents). $y(s,t) > 0$ when any events occurred in (s,t), and equals 0 when no events occurred.

Given the socio-environmental features F and risk score y for all the locations in time window $\{t_1, t_2, .., t_n\}$, the urban event ranking problem is to predict the ranking of the top-k locations $\{s_1, ..., s_K\} \in S$ in the next time interval t_{n+1} with the highest risk scores. The objective is to prioritize the ranking quality on the top-k riskiest locations. As a basic assumption of our problem, there exists spatial and temporal autocorrelation among locations in F, meaning nearby locations tend to have more correlated values. In addition, events are sparse so y is 0 for the majority of the locations. A detailed example of data attributes and feature generation steps can be found in the supplementary materials Appendix A.

2.2 Stochastic Optimization of NDCG in Event Ranking Problem

In the field of recommendation systems, learning to rank is substantially studied, and NDCG is widely used as the metric to measure the ranking quality of the foremost importance. In the following part, we use the terminologies from the event forecasting problem to define NDCG in our problem setting. For a ranked list of locations $s \in \mathcal{S}$ in a period $t \in T$, the NDCG score is computed as by:

$$\text{NDCG}_t = \frac{1}{Z_t} \sum_{s \in \mathcal{S}_t} \frac{2^{y_s} - 1}{\log_2(1 + \text{r}(s))}, \tag{1}$$

where y_s is the risk score of the location s, $\text{r}(s)$ denotes the rank of location s in the studied spatial domain $\mathcal{S}_t$, and Z_t is the Discounted Cumulative Gain (DCG) score [19] of the perfect ranking of locations for time period t. However, the rank operator in NDCG is non-differentiable in terms of model parameters, and thus cannot be optimized directly. A popular solution is to approximate the rank operator with smooth functions and then optimize its surrogates [37][31][32] as shown in Eq. 2.

$$\bar{g}(\mathbf{w}; \mathbf{x}, \mathcal{S}_t) = \sum_{s' \in \mathcal{S}_t} \ell(h_t(s'; \mathbf{w}) - h_t(\mathbf{s}; \mathbf{w})), \tag{2}$$

where rank operator $r(s)$ in NDCG is approximated by a differentiable surrogate function $\ell()$, and the squared hinge loss $\ell(x) = \max(0, x + c)^2$ is commonly used [41]. In this way, the model parameters $\mathbf{w}$ can be updated by a gradient-based optimizer. We can maximize over $L(\mathbf{w})$:

$$\max_{\mathbf{w} \in \mathbb{R}^d} L(\mathbf{w}) := \frac{1}{|T|} \sum_{t=1}^{T} \sum_{s \in \mathcal{S}_t^+} \frac{2^{y_s^t} - 1}{Z_t \log_2(\bar{g}(\mathbf{w}; \mathbf{x}_s^t, \mathcal{S}_t) + 1)}. \tag{3}$$

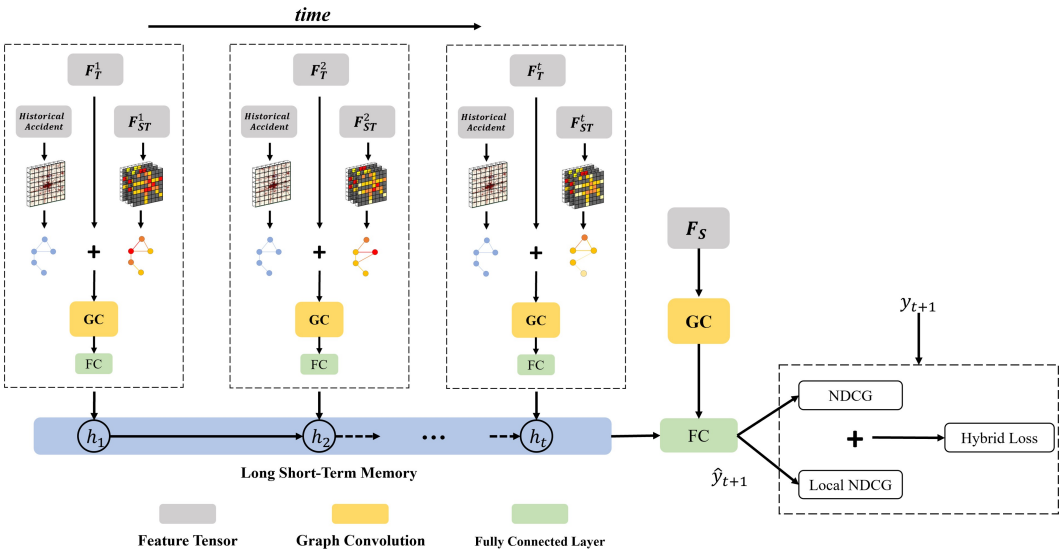

Figure 1: SpatialRank Architecture. The spatiotemporal features are used to generate adjacency matrices and then embedded by graph convolution layers. We use a fully connected layer to make final predictions. The hybrid objective function is combined with NDCG loss and local NDCG loss.

where $\mathbf{s}^t \in S_t^+$ denotes a set of locations with positive risk scores to be considered in the objective function. In this way, the optimization solution on NDCG can be directly applied to the event ranking problem.

3 Methodology

Directly optimizing NDCG as described above might provide a solution to our problem but fails to capture the spatiotemporal autocorrelation in the data. Also, since NDCG is a global measure, the model might not be able to learn how to rank nearby locations with highly correlated features correctly. In this section, we present our SpatialRank method to address these limitations. Figure. 1 demonstrates the proposed model architecture.

3.1 The Deep Learning Model

Many recent deep learning models for spatiotemporal could be used as the backbone architecture of our SpatialRank method. A key idea in these methods is to model the correlations between locations using a graph, with locations as nodes and edge weights representing the strength of correlations, and extract latent spatial dependency information through graph convolution layers [40] [5]. We follow a similar idea to build the overall network architecture, where spatial and spatiotemporal features are fed into graph convolutional layers. Then the extracted latent representations are concatenated with the temporal features and fed into LSTM layers to capture temporal representations before the final output layer.

A key novelty in our SpatialRank network is the design of a time-guided and traffic-aware graph-generating process for the graph convolution layers. Prior works typically use Pearson's correlation coefficients or similar measures of features (e.g., traffic volume, accident counts) between locations as their correlation strengths and pre-compute a **time invariant** adjacency matrix of locations [44]. In fact, studies [10][27] demonstrate that the influence of traffic conditions on events varies over different periods. Inspired by a related work [42] on a different problem, we use historical events to generate a static graph and learn a time-variant graph from F_{ST} (e.g., traffic volume) in each time interval. Intuitively, we use F_T (e.g. hour of the day) to learn the weights of dynamic graph vs. static graph to be considered in the graph convolution. The key equations of generating the adaptive graph

adjacency are shown below:

$$Z_1 = \tanh(\alpha E_1 W_1) \tag{4}$$

$$Z_2 = \tanh(\alpha E_2 W_2) \tag{5}$$

$$\mathcal{A}_{dynamic} = \text{ReLU}(\tanh(\alpha(Z_1 Z_2^T - Z_2 Z_1^T))) \tag{6}$$

$$\beta = \text{sigmoid}(F_T W_3) \tag{7}$$

$$\mathcal{A} = \beta \mathcal{A}_{dynamic} + (1 - \beta)\mathcal{A}_{static} \tag{8}$$

Where W_1, W_2, and W_3 are learnable parameters. E_1 and E_2 are randomly initialized node embeddings that can be learned during training. We represent those embeddings by the spatiotemporal features F_{ST} (e.g., traffic volume) to reveal the underlying dynamic connections between nodes. The subtraction and ReLU activation function in Eq. (6) lead to the asymmetric property of $\mathcal{A}_{dynamic}$. α is a hyperparameter to control the saturation rate. $\mathcal{A}_{static}$ is a pre-calculated adjacency matrix before training the model by computing the Pearson correlation coefficient between the risk scores of locations (nodes), where the correlation between node i and node j, $a_{ij} = \frac{\sum_t (y_i - \bar{y_i})(y_j - \bar{y_j})}{\sqrt{\sum_t (y_i - \bar{y_i})^2 (y_j - \bar{y_j})^2}}$. The final adjacency matrix $\mathcal{A}$ is the weighted sum of $\mathcal{A}_{dynamic}$ and $\mathcal{A}_{static}$, and the weight β is learned by a sigmoid activation function in Eq. 7 from the linear transformation of temporal feature F_T. Intuitively, the static adjacency matrix indicates a baseline correlation between different locations and it is pre-computed before training. This is also what most of the related work has been done. However, inspired by many observations from related studies [10][43], it is evident that we think this correlation is not always constant. Therefore, we use locations' time-variant features to construct a new adjacency matrix, and this dynamic adjacency matrix varies with time. We learn the parameters in this dynamic matrix during the network training process. Finally, we combine the static and the learned dynamic matrices through a learned weight β. In this way, a combined adjacency matrix can be treated as an adaptation from a static adjacency matrix considering the influence of other features during different periods. Finally, extracted embeddings are fed into LSTM layers and make final predictions of the risk scores y for each location through a fully connected layer.

3.2 Local Ranking Metric and Hybrid Loss function

As previously mentioned, ranking-based metrics such as NDCG are not designed to handle spatial correlations. According to the first law of geography [38], nearby locations may share very similar socio-environmental attributes, thus it is challenging to rank neighboring locations correctly. However, the locations nearby with uncertain event patterns are worthy to be focused on so that more potential events can be discovered. Moreover, using NDCG on event ranking problems can cause over-concentrating on top-ranked locations and sacrificing prediction accuracy on other locations due to lower priority. To address those issues, we design a novel local ranking measurement named Local Normalized Discounted Cumulative Gain (L-NDCG) to measure spatially local ranking quality over every sub-region of the study area. The L-NDCG is calculated as:

$$\text{L-NDCG} = \frac{1}{T|\mathcal{S}_t|} \sum_{t=1}^{T} \sum_{s^t \in \mathcal{S}_t} \sum_{\hat{s} \in \mathcal{N}(s^t)} \frac{2^{y_{\hat{s}}^t} - 1}{Z_{\mathcal{N}(s^t)}^t \log_2(r(\hat{s}, \mathcal{N}(s^t)) + 1)}. \tag{9}$$

We use the same terminologies from Eq. 3. The unique part is that we compute an NDCG score for every location in the study area based on the local ranking of a subset of locations $\mathcal{N}(s^t)$, where $\mathcal{N}$ is a neighborhood of location s^t. We define the $\mathcal{N}$ as the Euclidean distance of coordinates smaller than R in this work. $\mathcal{N}$ can be defined in other ways depending on the need of the problem. Essentially, L-NDCG is the average of NDCG scores for all subsets of locations. In this way, a few stationed hot-spot locations only take considerably large weight in their own NDCG scores and cannot over-influence the overall L-NDCG score. L-NDCG emphasizes ranking correctly on each subset of locations, and a greater L-NDCG score indicates that relatively more important locations can be distinguished from their nearby locations in a small region. Similar to optimizing NDCG, the ranking operator of L-NDCG is non-differentiable, thus we optimize its surrogates instead.

$$\max_{\mathbf{w} \in \mathbb{R}^d} L(\mathbf{w}) := \frac{1}{T|\mathcal{S}_t^+|} \sum_{t=1}^{T} \sum_{s^t \in \mathcal{S}_t^+} \sum_{\hat{s} \in \mathcal{N}(s^t)} \frac{2^{y_{\hat{s}}^t} - 1}{Z_{\mathcal{N}(s^t)}^t \log_2(\bar{g}(\mathbf{w}; \mathbf{x}_{\hat{s}}^t, \mathcal{N}(s^t)) + 1)}. \tag{10}$$

Where $g(\mathbf{w}; \mathbf{x}_{\hat{s}}^t, \mathcal{N}(s^t))$ is a surrogate loss function similar to Eq. 2, and $\mathcal{N}(s^t)$ is a set of locations. Finally, to learn a trade-off between locally ranking quality and globally ranking quality we design a hybrid objective function consisting of both NDCG and L-NDCG

$$Loss = (1 - \sigma)\text{NDCG} + \sigma\text{L-NDCG} \tag{11}$$

Where σ is a hyperparameter controlling the preference between NDCG and L-NDCG.

3.3 Importance-based Location Sampling with Spatial Filtering

To bridge the gap between capturing spatial correlations in a list of locations and optimizing the quality of predicted ranking, we propose a novel importance-based location sampling strategy with spatial filtering. Specifically, we first design an importance measure function to assign higher weights to locations with larger errors in predictions and higher ranking priority. Secondly, we sample locations based on the weights assigned by importance-measuring, so that important locations have a higher probability to be sampled. Afterward, only losses from sampled locations will be calculated in the objective function and considered during the optimization. In this way, the model pays attention to more important locations. Thirdly, we adjusted the importance scores every epoch, so that the model learns to focus on different locations adaptively. Lastly, spatial filters are applied to smooth the importance scores in each epoch to achieve spatial-aware sampling. This allows nearby locations to be sampled in the same batch with high probability to help the model learn how to rank them. The training process is shown in algorithm 1

Algorithm 1: SpatialRank Training

Input: feature tensor F, event risk scores y, hyperparameter σ, standard deviation λ
Output: learned model f_θ

1 Initialize probability set $P = \{p_1, p_2, ..., p_l\} \in L$;
2 **for** *each epoch* **do**
3 Compute $\hat{y} = f_\theta(x)$
4 Compute $loss_{NDCG}$ = **WeightedLoss**$(y, \hat{y}, P)$
5 Compute $loss_{local}$ = **LocalWeightedLoss**$(y, \hat{y}, P)$
6 $loss_{hybrid} = (1 - \sigma) \cdot loss_{NDCG} + \sigma \cdot loss_{local}$
7 **for** $s \in \mathcal{S}$ **do**
8 $E_s = \frac{1}{T} \sum_t \frac{2^{|y_s^t - \hat{y_s^t}|} - 1}{log_2(1 + r(y_s^t))}$
9 $E'_{s'} = \sum_s \frac{E_s}{2\pi\lambda^2} e^{-\frac{x^2}{2\lambda^2}}$ for $s \in \mathcal{S}$, $x = dist(s', s)$
10 Update P = **normorlize**(E')
11 Compute gradient $\nabla f(\theta)$ by $loss_{hybrid}$
12 Update θ by $\nabla f(\theta)$
13 **return** f_θ

Algorithm 1 shows the details of the importance-based sampling mechanism. The inputs include feature tensor F, event risk scores y, hyperparameter σ, and Gaussian standard deviation λ. The output is a learned model. The algorithm starts with initializing a set of equal-importance scores, which means the probabilities of locations being sampled are equal in the first epoch. Line 3 is forward propagation with current model parameters. Line 4 calculates the weighted NDCG losses given true label y, predicted labels $\hat{y}$, and current importance scores for locations. $Loss_{NDCG}$ is computed by Eq. 9, where importance scores $\{p_1, p_2, ..., p_s\} \in \mathcal{S}$ are normalized and treated as weights. Line 6 is our novel hybrid loss function discussed in Eq 10 to leverage the local ranking quality. The key step is to update the importance scores set based on current prediction errors and true ranking in Lines 7-8. We design a score function shown in Line 9 to leverage the errors made in predictions and the priority of true ranking for each location.

$$E_s = \frac{1}{T} \sum_t \frac{2^{|y_s^t - \hat{y_s^t}|} - 1}{log_2(1 + r(y_s^t))} \tag{12}$$

where $r()$ denotes a ranking function of the i-th location in the study area, and $|y_s^t - \hat{y_s^t}|$ is the absolute difference between predicted injuries and true injuries. Basically, a higher score E_l indicates that

there are larger prediction errors and higher true ranking in this location s. Note that smaller $r(y_s^t)$ denotes a higher true ranking. To capture their geographical connections, we map the list of locations to their original locations on the grids, and then we apply a Gaussian filter to smooth the score distribution spatially in Line 9. λ is the standard deviation. In this way, the model can learn better spatially correlated patterns, and avoid over-focusing on a few standalone locations but ignoring the the area nearby. Next, the smoothed scores E^* are normalized into P, where $\sum_s P_s = 1$. Finally, we compute the gradient $\nabla f(\theta)$ by $loss_{hybrid}$ and update model parameters θ. After a few iterations, we obtain the learned model f_θ.

3.4 Complexity Analysis

In each iteration complexity of SpatialRank, we need to conduct forward propagation $h_t(s; \mathbf{w}), \forall s \in \mathcal{S}_t^+ \cup \mathcal{S}_t$ and back-propagation for computing $\nabla h_t(s; \mathbf{w}), \forall s \in \mathcal{S}_t^+ \cup \mathcal{S}_t$. The complexity of forwarding is $S = \sum_{t \in T}(|\mathcal{S}_t^+|\mathcal{N}(s) + |\mathcal{S}_t|)d \leq O(TSd)$, where d is the model size. To compute $local_{NDCG}$ part, an extra cost on neighbor querying is needed to replace location list size on its complexity. The cost of computing $\bar{g}(\mathbf{w}; \mathbf{x}, \mathcal{S}_t)$ is $\sum_{t \in T} |\mathcal{S}_t^+||\mathcal{S}_t| \leq O(TS^2)$ for NDCG and $\sum_{t \in T} |\mathcal{S}_t^+|\mathcal{N}^2(s) \leq O(TS\mathcal{N}^2(s))$ for local NDCG. The size of $\mathcal{N}^2(s)$ is small, therefore the total cost is reduced to $O(TSd + TS^2)$.

4 Experiments

We perform comprehensive experiments on three real-world traffic accident and crime datasets from Chicago[2] and the State of Iowa[3]. Experiment results show that our proposed approach substantially outperforms state-of-art baselines on three datasets by up to 12.7%, 7.8%, and 4.2% in NDCG respectively. The case study and results on the state of Iowa are shown in Appendix C.

Data. In the Chicago dataset, we collect data from the year 2019 to the year 2021. The first 18 months of this period are used as the training set, and the last 6 months of 2020 are used as the validating set. The year 2021 is used as a testing set. The area of Chicago is partitioned by 500 m $\times$ 500 m square cells and converted to a grid with the size of 64×80. For the crime dataset, we use total crimes as the risk score. For accident datasets, we use the number of injuries as the risk score.

Baselines. First, we use daily **Historical Average (HA)**. Next, we consider popular machine-learning methods including **Long Short-term Memory (LSTM)** [18], and **Convolutional LSTM Network (ConvLSTM)** [34]. Thirdly, we compare with recent methods such as **GSNet** [40], **Hetero-ConvLSTM**[43], and **HintNet** [5]. Moreover, we compared our optimization approach with other NDCG optimization solutions including **Cross Entropy (CE)**, **ApproxNDCG** [31], and **SONG** [32]. The details of the baselines are described in Appendix C.

Metrics. To measure the ranking quality of foremost locations, we test $K \in [30, 40, 50]$ on the test data. We use the metrics including NDCG, L-NDCG, and top-K precision (Prec)[26]. We report the average performance and standard deviation over 3 runs for three datasets.

4.1 Performance Comparison

In table 4, we can observe that our SpatialRank significantly outperforms other compared baselines in both datasets. We observe a similar trend among all metrics. On both datasets, Hetero-ConvLSTM and HintNet achieve similar results and outperform general machine learning methods such as LSTM and ConvLSTM. GSNet is designed on a dataset with a smaller grid size, thus performing worse on our larger grid. To study the effectiveness of learning the appropriate graph, we set the learning parameter β as a fixed ratio of 0.5 in SpatialRank#. Oppositely, β is a parameter to be learned based on temporal features in SpatialRank. In the Table 4 and Table. 5, each * indicates that the performance improvement of the proposed method over this baseline is statistically significant based on the student t-test with $\alpha = 0.05$ over three runs. The results show that the design of the time-aware graph convolution is able to improve performance and capture more dynamic variations in the graph, therefore performing better over different top-k rankings.

[2] https://data.cityofchicago.org/Transportation/Traffic-Crashes-Crashes/85ca-t3if

[3] https://icat.iowadot.gov/

TABLE 1: PERFORMANCE COMPARISON

CHICAGO ACCIDENT	K=30			K=40			K=50		
	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
HA	.214±0	.332±0	.502±0	.225±0	.322±0	.497±0	.235±0	.316±0	.493±0
LSTM	.215±2‰*	.392±2‰*	.519±3‰*	.225±1‰*	.380±2‰*	.543±3‰*	.249±1‰*	.368±2‰*	.544±3‰*
CONVLSTM	.225±5‰*	.410±4‰*	.558±3‰*	.236±1‰*	.388±4‰*	.563±8‰*	.252±1‰*	.366±2‰*	.540±8‰*
GSNET	.194±1‰*	.371±2‰*	.493±5‰*	.201±2‰*	.371±2‰*	.517±5‰*	.231±1‰*	.337±3‰*	.499±3‰*
HETERO-CONVLSTM	.229±2‰*	.401±1‰*	.557±2‰*	.240±1‰*	.395±4‰*	.564±3‰*	.255±3‰*	.375±2‰*	.551±3‰*
HINTNET	.228±1‰*	.400±2‰*	.555±3‰*	.238±2‰*	.390±3‰*	.569±3‰*	.256±1‰*	.373±4‰*	.561±8‰*
SPATIALRANK #	.250±2‰	.438±3‰	.591±3‰	.256±1‰	.409±2‰	.593±2‰	.271±2‰	.394±2‰	.585±1‰
SPATIALRANK	**.257**±1‰	**.444**±3‰	**.621**±4‰	**.268**±1‰	**.420**±1‰	**.614**±2‰	**.278**±3‰	**.403**±1‰	**.599**±1‰
CHICAGO CRIME	K=30			K=40			K=50		
	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
HA	.237±0	.348±0	.514±0	.250±0	.333±0	.506±0	.259±0	.322±0	.449±0
LSTM	.246±1‰*	.327±2‰*	.517±3‰*	.257±1‰*	.329±2‰*	.521±3‰*	.262±3‰*	.314±5‰*	.512±3‰*
CONVLSTM	.313±2‰*	.415±4‰*	.617±4‰*	.325±2‰*	.404±1‰*	.607±6‰*	.333±2‰*	.387±4‰*	.599±3‰*
GSNET	.283±3‰*	.388±2‰*	.584±3‰*	.296±2‰*	.374±2‰*	.565±3‰*	.299±2‰*	.335±4‰*	.568±3‰*
HETERO-CONVLSTM	.346±3‰*	.468±3‰*	.657±4‰	.365±1‰*	.452±3‰*	.642±6‰	.374±4‰*	.433±4‰*	.638±4‰*
HINTNET	.342±3‰*	.468±3‰*	.661±4‰	.358±3‰*	.448±4‰*	.631±6‰*	.369±2‰*	.434±2‰*	.628±4‰*
SPATIALRANK #	.361±2‰	.484±1‰	**.670**±4‰	.376±4‰	.463±3‰	**.655**±7‰	.387±1‰	**.446**±1‰	**.651**±7‰
SPATIALRANK	**.373**±2‰	**.491**±3‰	.665±4‰	**.380**±2‰	**.467**±5‰	.647±6‰	**.392**±2‰	**.446**±3‰	.644±6‰

$\beta = 0.5$ ‰: $\times 10^{-3}$

TABLE 2: OPTIMIZATION COMPARISON

CHICAGO ACCIDENT	K=30			K=40			K=50		
	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
CE	.232±1‰	.424±1‰	.588±2‰	.253±1‰	.415±2‰	.588±2‰	.266±1‰	.400±1‰	.580±2‰
APPROXNDCG	.240±2‰	.426±2‰	.597±3‰	.255±1‰	.415±1‰	.591±6‰	.264±1‰	.398±2‰	.575±3‰
SONG	.240±1‰	.438±2‰	.610±6‰	.254±2‰	.417±1‰	.600±11‰	.267±2‰	.400±4‰	.575±1‰
SPATIALRANK	**.257**±1‰	**.444**±3‰	**.621**±4‰	**.268**±1‰	**.420**±1‰	**.614**±2‰	**.278**±3‰	**.403**±1‰	**.599**±1‰
CHICAGO CRIME	K=30			K=40			K=50		
	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
CE	.362±1‰	.475±1‰	.653±3‰	.378±1‰	.455±1‰	.646±1‰	.386±1‰	.441±2‰	.644±2‰
APPROXNDCG	.344±4‰	.455±3‰	.637±10‰	.353±1‰	.435±2‰	**.658**±5‰	.365±3‰	.421±4‰	.619±6‰
SONG	.364±2‰	.484±3‰	.660±5‰	.379±4‰	.466±3‰	.456±1‰	.390±2‰	**.450**±3‰	**.649**±6‰
SPATIALRANK	**.373**±2‰	**.491**±3‰	**.665**±4‰	**.380**±2‰	**.467**±5‰	.647±6‰	**.392**±2‰	.446±3‰	.644±6‰

‰: $\times 10^{-3}$

4.2 Optimization Comparison

To evaluate the effectiveness of our proposed hybrid objective function and importance-based location sampling, we perform experiments on the same network architecture but with different optimization solutions including Cross Entropy (CE), ApproxNDCG, and SONG. The results are shown in table 5. Methods designed to optimize NDCG consistently perform better than Cross Entropy. SpatialRank substantially outperforms SONG and ApproxNDCG and made a noticeable improvement on L-NDCG as it is considered in the objective function.

4.3 Ablation Study

We examine the effects of tuning hyper-parameter σ in the hybrid loss function. We present the results in table 8. Recall that σ decides the ratio of L-NDCG takes in the objective function. $\sigma = 0$ means L-NDCG is not considered in the objective function. Starting from $\sigma = 0$, we observe a trend that the overall performance improves steadily while σ goes up, and it reaches the best performance

TABLE 3: ABLATION STUDY

CHICAGO ACCIDENT	K=30			K=40			K=50		
	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
$\sigma = 0.0$	0.251	0.439	0.610	0.263	0.407	**0.612**	0.274	0.289	0.605
$\sigma = 0.05$	0.239	0.416	0.559	0.254	0.394	0.600	0.266	0.386	0.597
$\sigma = 0.1$	**0.256**	0.443	**0.621**	**0.268**	**0.421**	0.608	**0.276**	**0.400**	0.599
$\sigma = 0.2$	0.250	**0.444**	0.616	0.261	0.411	0.605	0.271	0.394	**0.603**
$\sigma = 0.3$	0.234	0.422	0.606	0.243	0.388	0.59	0.253	0.374	0.591
CHICAGO CRIME	K=30			K=40			K=50		
	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
$\sigma = 0.0$	0.366	0.489	0.661	0.378	0.464	0.644	0.385	0.445	0.642
$\sigma = 0.05$	0.366	0.489	0.659	0.378	**0.467**	**0.649**	0.382	0.447	**0.643**
$\sigma = 0.1$	**0.371**	**0.494**	**0.665**	**0.383**	**0.467**	0.644	**0.391**	**0.450**	0.642
$\sigma = 0.2$	0.356	0.474	0.654	0.367	0.448	0.643	0.372	0.429	0.623
$\sigma = 0.3$	0.345	0.457	0.658	0.361	0.445	0.641	0.373	0.431	0.631

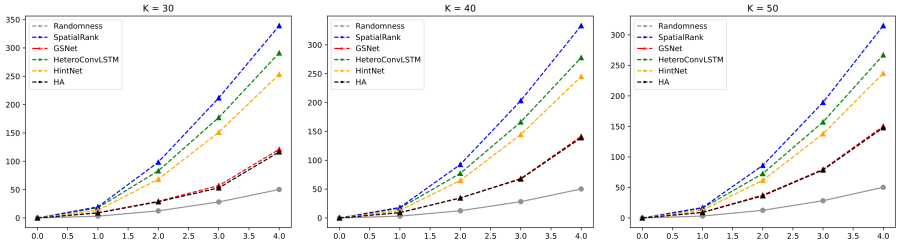

Figure 2: Comparison of cross-K function in Chicago Accident.

at σ equals 0.1 in the accident and crime dataset. It indicates that considering a reasonable weight of L-NDCG in the objective function boosts overall performance. These results demonstrate the importance of considering local ranking because NDCG, L-NDCG, and precision can be all improved.

4.4 Cross-K function

We use the Cross-K function[15] with Monte Carlo Simulation[36] to evaluate the accuracy of predicted locations. The Cross-K function measures the spatial correlation between the predicted locations and true locations. Specifically, we calculate the average density of predictions within every distance d of a true event in each day as shown in Eq. 14:

$$\hat{K}(d) = \lambda_j^{-1} \sum_{i \neq j} I(d_{ij} \leq d)/n, \tag{13}$$

where λ is global density of event j, and $I()$ is an identity function which equals one if real distance d_{ij} is smaller than d, else equals zero. n is the number of events i. The results are shown in Figure 2. The grey curve represents the complete spatial randomness and we use it as a reference baseline. Higher is better. Our SpatialRank achieves the best predictions in both datasets, which indicates that the predictions of SpatialRank are significantly spatially correlated with ground truth. The similar results on the other two datasets are shown in Appendix C in the supplementary materials.

Additional Results and a case study to demonstrate successful prediction examples are included in the Supplementary materials. We also include our code and sample data with the supplementary materials.

5 Related Work

Urban Event forecasting has been widely studied in the past few decades. Most early studies [14] [24] [6][21] rely on small-scale and single-source datasets with limited types of features, thus the prediction accuracy is limited. Notably, Zhou et al. [45] proposed a differential time-varying graph convolution network capturing traffic changes and improving prediction accuracy. Similarly, Wang et al. [40] proposed GSNet with a geographical module and a weighted loss function to capture semantic spatial-temporal correlations among regions and solve data sparsity issues. Addressing the issue of spatial heterogeneity, Yuan et al. [43] proposed Hetero-ConvLSTM to leverage an ensemble of predictions from models learned from pre-selected sub-regions. Furthermore, An et al. [5] proposed HintNet partitions the study area hierarchically and transfers learned knowledge over different regions to improve performance. However, most existing models rely on optimizing cross-entropy and the objective is to make accurate predictions on every location. **Learning to rank** is an extensively studied area in recommendation systems and search engines [25]. NDCG is a widely adopted metric to measure ranking quality. The prominent class of methods involves approximating ranks in NDCG using smooth functions and subsequently optimizing the resultant surrogates. Taylor et al. [37] tries to use rank distributions to smooth NDCG, but suffers from high computational cost. Similarly, Qin et al. [31] approximates the indicator function by a generalized sigmoid function with a top-k variant. Noticeably, Qiu et al. [32] develop stochastic algorithms optimizing the surrogates for NDCG and its top-K variant. Although it is possible to directly apply the existing ranking methods to the urban event ranking problem, the performance tends to be unsatisfactory as these methods commonly assume independence between items and queries and lack the ability to handle spatiotemporal autocorrelation in the data.

6 Conclusion and Limitations

In this work, we formulate event forecasting as a location ranking problem. We propose a novel method SpatialRank to learn from spatiotemporal data by optimizing NDCG surrogates. To capture dynamic spatial correlations, we design an adaptive graph convolution layer to learn the graph from features. Furthermore, we propose a hybrid loss function to capture potential risks around hot-spot regions, and a novel ranking-based importance sampling mechanism to leverage the importance of each location considered during the model training. Extensive experimental results on three real-world datasets demonstrate the superiority of SpatialRank compared to baseline methods.

Limitations: the model's performance might be affected by other properties of data, such as spatial heterogeneity and sparsity. We observe less improvement over baselines on the Iowa dataset, partially due to that the data is sparser over a large area with heterogeneity. These are issues addressed by some of the prior work and can be addressed in our future work. In addition, the new algorithm increases the training time complexity due to the sampling steps. This is acceptable due to the relatively small number of locations and time periods in urban event datasets but may require extra work to generalize to large datasets.

Acknowledgments and Disclosure of Funding

T. Yang was partially supported by NSF Career Award 18844403, NSF Fair AI Grant 2147253, NSF RI Grant 2110545.

References

[1] *Chicago Police Department, https://home.chicagopolice.org/cpd-expands-smart-policing-technology-to-support-strategic-deployment-and-cta-safety/.* 2020.

[2] *Fox News, https://www.foxnews.com/us/us-murder-rate-continued-grim-climb-i2021-new-fbi-estimates-show*. 2022.

[3] *Police Executive Research Forum, https://www.policeforum.org/workforcemarch2022*. 2022.

[4] J. Abellán, G. López, and J. De Oña. Analysis of traffic accident severity using decision rules via decision trees. *Expert Systems with Applications*, 40(15):6047–6054, 2013.

[5] B. An, A. Vahedian, X. Zhou, W. N. Street, and Y. Li. Hintnet: Hierarchical knowledge transfer networks for traffic accident forecasting on heterogeneous spatio-temporal data. *Proceedings of the 2022 SIAM International Conference on Data Mining*, 2022.

[6] L. Barba, N. Rodríguez, and C. Montt. Smoothing strategies combined with arima and neural networks to improve the forecasting of traffic accidents. *The Scientific World Journal*, 2014, 2014.

[7] R. Bergel-Hayat, M. Debbarh, C. Antoniou, and G. Yannis. Explaining the road accident risk: Weather effects. *Accident Analysis & Prevention*, 60:456–465, 2013.

[8] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, volume 29, pages 730–738, 2015.

[9] C. Caliendo, M. Guida, and A. Parisi. A crash-prediction model for multilane roads. *Accident Analysis & Prevention*, 39(4):657–670, 2007.

[10] R. N. Carey and K. M. Sarma. Impact of daylight saving time on road traffic collision risk: a systematic review. *BMJ Open*, 7(6), 2017.

[11] L.-Y. Chang. Analysis of freeway accident frequencies: negative binomial regression versus artificial neural network. *Safety science*, 43(8):541–557, 2005.

[12] L.-Y. Chang and W.-C. Chen. Data mining of tree-based models to analyze freeway accident frequency. *Journal of safety research*, 36(4):365–375, 2005.

[13] N. Chicago. Winter storm warning issued for chicago area ahead of forecasted snow storm. In *media*, 2021.

[14] M. M. Chong, A. Abraham, and M. Paprzycki. Traffic accident analysis using decision trees and neural networks. *ArXiv*, cs.AI/0405050, 2004.

[15] P. M. Dixon. *Ripley's K Function*. John Wiley & Sons, Ltd, 2014.

[16] N. Dong, H. Huang, and L. Zheng. Support vector machine in crash prediction at the level of traffic analysis zones: Assessing the spatial proximity effects. *Accident Analysis & Prevention*.

[17] P. et al. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019.

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[19] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[20] A. Khezerlou, X. Zhou, L. Li, Z. Shafiq, A. Liu, and F. Zhang. A traffic flow approach to early detection of gathering events: Comprehensive results. *ACM transactions on intelligent systems and technology*, 8(6):1–24, 2017.

[21] A. V. Khezerlou, X. Zhou, X. Li, W. N. Street, and Y. Li. Dilsa+: Predicting urban dispersal events through deep survival analysis with enhanced urban features. *ACM transactions on intelligent systems and technology*, 12(4):1–25, 2021.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. 2014.

[23] Z. Li, C. Huang, L. Xia, Y. Xu, and J. Pei. Spatial-temporal hypergraph self-supervised learning for crime prediction. In *arXiv.org*, Ithaca, 2022. Cornell University Library, arXiv.org.

[24] L. Lin, Q. Wang, and A. W. Sadek. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. *Transportation Research Part C*, 55, 2015.

[25] T.-Y. Liu. *Learning to Rank for Information Retrieval.* Springer, 2011.

[26] J. Lu, C. Xu, W. Zhang, L. Duan, and T. Mei. Sampling wisely: Deep image embedding by top-k precision optimization. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7960–7969. IEEE, 2019.

[27] X. Mao, C. Yuan, J. Gan, and S. Zhang. Risk factors affecting traffic accidents at urban weaving sections: Evidence from china. *International journal of environmental research and public health*, 16(9):1542, 2019.

[28] T. R. Miller, M. A. Cohen, D. I. Swedler, B. Ali, and D. V. Hendrie. Incidence and costs of personal and property crimes in the usa, 2017. *Journal of benefit-cost analysis*, 12(1):24–54, 2021.

[29] A. Najjar, S. Kaneko, and Y. Miyanaga. Combining satellite imagery and open data to map road safety. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[30] NHTSA. 2021 fatality data show increased traffic fatalities during pandemic. `https://www.nhtsa.gov/press-releases/early-estimate-2021-traffic-fatalities`, Jun 2021.

[31] T. Qin, T.-Y. Liu, and H. Li. A general approximation framework for direct optimization of information retrieval measures. *Information Retrieval*, 13(4):375–397, 2010.

[32] Z. Qiu, Q. Hu, Y. Zhong, L. Zhang, and T. Yang. Large-scale stochastic optimization of ndcg surrogates for deep learning with provable convergence. In *International Conference on Machine Learning*, 2022.

[33] H. Ren, Y. Song, J. Wang, Y. Hu, and J. Lei. A deep learning approach to the citywide traffic accident risk prediction. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3346–3351, 2018.

[34] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. 2015.

[35] R. Swezey, A. Grover, B. Charron, and S. Ermon. Pirank: Scalable learning to rank via differentiable sorting. *Advances in Neural Information Processing Systems*, 34, 2021.

[36] R. Tao and J.-C. Thill. Flow cross k-function: a bivariate flow analytical method. *International journal of geographical information science : IJGIS*, 33(10):2055–2071, 2019.

[37] M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Web Data Mining*, pages 77–86, 2008.

[38] W. R. Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46:234–240, 1970.

[39] A. Vahedian, X. Zhou, L. Tong, W. N. Street, and Y. Li. Predicting urban dispersal events: A two-stage framework through deep survival analysis on mobility data. 2019.

[40] B. Wang, Y. Lin, S. Guo, and H. Wan. Gsnet: Learning spatial-temporal correlations from geographical and semantic aspects for traffic accident risk forecasting. In *2021 AAAI Conference on Artificial Intelligence (AAAI'21)*, 2021.

[41] M. Wu, Y. Chang, Z. Zheng, and H. Zha. Smoothing dcg for learning to rank: a novel approach using smoothed hinge functions. In *Proceedings of the 18th ACM conference on information and knowledge management*, CIKM '09, pages 1923–1926. ACM, 2009.

[42] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 753–763, 2020.

[43] Z. Yuan, X. Zhou, and T. Yang. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 984–992, 2018.

[44] Y. Zhang, Y. Li, X. Zhou, X. Kong, and J. Luo. Trafficgan: Off-deployment traffic estimation with traffic generative adversarial networks. *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1474–1479, 2019.

[45] Z. Zhou, Y. Wang, X. Xie, L. Chen, and C. Zhu. Foresee urban sparse traffic accidents: A spatiotemporal multi-granularity perspective. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.

7 Appendix A Feature Engineering

In this section, we explain how we generate features. Features are generated on partitioned grid cells and at different time intervals.

Temporal Features F_T Such calendar features and Weather features are generated from the date of Vehicle Crash or Crime Records, where all grid cells share a vector of temporal features in a time interval. calendar features include the day of the year, the month of the year, holidays, and so on. Weather features include temperature, precipitation, snowfall, wind speed, etc.

Spatial Features F_S are generated based on each grid cell and remain the same over different time intervals. First, POI features are the number of POI data in each grid cell for different categories. For example, one of the POI types is shopping, we count the number of shopping instances in each grid cell. Second, basic road condition features are extracted from road network data, in which we calculate the summation or average of provided data for road segments in each grid cell. Third, we use top eigenvectors of the Laplacian matrix of road networks as spatial graph features[43], which represent the topological information for each grid cell.

Spatio-Temporal Features F_{ST} such as real-time traffic conditions are estimated by taxi GPS data and Bus GPS data. Spatio-temporal features include pick-up volumes, drop-off volumes, traffic speed, etc.

Feature Summary In total, 36 features are extracted, including 12 temporal features, 18 spatial features, and 6 spatial-temporal features for each location s and time interval t.

8 Appendix B Methodology

8.1 Symbol Table

Symbol Table	
Symbol	Explainations
$\mathcal{S}$	Spatial filed, study area
s	A partitioned location, grid cell
T	Temporal filed, study period
t	Time interval (e.g. hours, days)
F_T	Temporal features (weather, time)
F_S	Spatial features (e.g. POI)
F_ST	Spatiotemporal features (e.g. traffic conditions)
Z	Discounted Cumulative Gain (DCG) score
a	Pearson correlation coefficient
NDCG	Normalized Discounted Cumulative Gain
L-NDCG	Local Normalized Discounted Cumulative Gain
Prec	top-K precision
r()	Ranking function
$\mathcal{N}$	Neighbour querying

9 Appendix C Experiments

Parameter Configuration. For each method, we train the network for 100 epochs and save the model with the best performance on validating set. We use the Adam optimizer [22] with settings $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We tune trade-off hyperparameter σ with value 0, 0.05, 0.1, 0.2, and 0.3. The R in the spatial neighbor definition is set as 2 grid cells for computation efficiency. We use an initial warm-up strategy by optimizing cross-entropy at the first 20 epochs to obtain a good initial solution because merely optimizing NDCG might land in the local minimum if a poor initial solution is given. The same warm-up process is also applied to optimization baselines. The learning rate is set at 0.001 in the warm-up and changed to 0.0001 afterward. The batch size is 64.

Data. For the state of Iowa, we collect data from 2016 to 2018. 80% of data from the year 2016 and year 2017 is randomly selected as a training set, and the remaining data is used as the validating set. The data from the year of 2018 is used as the testing set. The area is partitioned by 5 km $\times$ 5 km cells. The grid size is 128×64. Normalization is used to transform data into the range $[0, 1]$.

TABLE 4: PERFORMANCE COMPARISON

IOWA	K=30			K=40			K=50		
	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
HA	.314±0	.167±0	.326±0	.326±0	.134±0	.261±0	.333±0	.111±0	.231±0
LSTM	.503±3‰*	.278±3‰*	.573±3‰*	.522±1‰*	.209±1‰*	.518±3‰*	.519±5‰*	.187±1‰*	.474±1‰*
CONVLSTM	.490±3‰*	.282±2‰*	.583±1‰*	.507±3‰*	.207±1‰*	.513±4‰*	.511±3‰*	.189±3‰*	.474±8‰*
GSNET	.493±2‰*	.265±1‰*	.569±3‰*	.509±3‰*	.222±3‰*	.527±3‰*	.526±5‰*	.207±1‰*	.510±3‰*
HETERO-CONVLSTM	.518±1‰*	.289±2‰*	.617±4‰	.523±5‰*	.258±1‰*	.589±5‰*	.543±3‰*	.226±1‰*	.534±5‰*
HINTNET	.512±5‰	.289±3‰*	.617±1‰	.542±5‰	.243±4‰*	.590±9‰	.556±3‰*	.209±2‰*	.534±8‰*
SPATIALRANK #	.530±3‰	.300±2‰	.617±4‰	.556±3‰	.264±2‰	.594±3‰	.571±3‰	.223±1‰	**.552**±4‰
SPATIALRANK	**.540**±3‰	**.309**±2‰	**.618**±3‰	**.563**±6‰	**.268**±3‰	**.600**±3‰	**.585**±3‰	**.232**±2‰	.550±6‰

$\beta = 0.5$ ‰: $\times 10^{-3}$

TABLE 5: OPTIMIZATION COMPARISON

IOWA	K=30			K=40			K=@50		
	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
CE	.531±4‰	.306±4‰	.554±2‰	.555±1‰	.246±2‰	.556±2‰	.548±2‰	.205±1‰	.502±1‰
APPROXNDCG	.528±4‰	.304±3‰	.561±1‰	.551±3‰	.250±4‰	.557±6‰	.554±6‰	.212±2‰	.508±3‰
SONG	.529±3‰	.308±2‰	**.618**±1‰	**.563**±3‰	.264±4‰	.584±1‰	.581±3‰	.227±4‰	.536±6‰
SPATIALRANK	**.540**±3‰	**.309**±2‰	**.618**±3‰	**.563**±6‰	**.268**±3‰	**.600**±3‰	**.585**±3‰	**.232**±2‰	**.550**±6‰

‰: $\times 10^{-3}$

Platform. We run the experiments on a High-Performance Computer system where each node runs has an Intel Xeon E5 2.4 GHz and 256 GB of Memory. We use a GPU node with Nvidia Tesla V100 Accelerator Cards with the support of Pytorch library [17] to train the deep learning models. Our code is available at: https://github.com/BANG23333/SpatialRank

Baselines. First, we compare our methods with daily **Historical Average (HA)**. Next, we consider popular machine-learning methods. **Long Short-term Memory (LSTM)** [18] is a recurrent neural network architecture with feedback connections, we stack two fully-connected LSTM layers. **Convolutional LSTM Network (ConvLSTM)** [34] is a recurrent neural network with convolution layers for spatial-temporal prediction. We use a stacked two-layer network. Thirdly, we compare with recent methods designed to tackle the traffic accident occurrence prediction problem. **GSNet** [40] is a deep-learning method utilizing complicated graph information. **Hetero-ConvLSTM** [43] is an advanced deep-learning framework to address spatial heterogeneity. It applies multiple ConvLSTM on pre-defined sub-regions with size 32×32. We use the same parameter setting in the experiments. **HintNet** [5] is a recent work capturing heterogeneous accident patterns by a hierarchical-structured learning framework. Finally, using our proposed network, we compared our optimization approach with other NDCG optimization solutions. **Cross Entropy (CE)** is a well-accepted objective function. **ApproxNDCG** [31] approximates the indicator function in the computation of ranks. **SONG** [32] is an efficient stochastic method to optimize NDCG.

9.1 Performance Comparison

We compare the results between SpatialRank and the baselines on the Iowa traffic accident dataset and observe that our full version of SpatialRank still outperforms other compared baselines in all the three metrics. The results are shown in Table 4. SpatialRank$^{\#}$ with β pre-defined (as 0.5) rather than automatically learned is the second best. Consistent with the results on the other two datasets, SparkRank outperforms all the baselines and the dynamic convolution layers are effective in improving the model's performance.

9.2 Optimization Comparison

We perform the comparison of optimization methods on the Iowa traffic accident data. We use the same network architecture but with different optimization solutions including Cross Entropy (CE), ApproxNDCG, and SONG. The results are shown in Table 5. Methods designed to optimize NDCG consistently perform better than Cross Entropy. SpatialRank substantially outperforms SONG and ApproxNDCG and made a noticeable improvement on L-NDCG as it is considered in the objective function.

TABLE 6: COMPUTATION COST

TRAINING TIME	COST IN SECONDS			
	SPATIALRANK	HINTNET	HETEROCONVLSTM	GSNET
CHICAGO	88.2	132.1	**47.7**	98.8
IOWA	76.5	117.5	**41.6**	83.5
INFERENCE TIME	SPATIALRANK	HINTNET	HETEROCONVLSTM	GSNET
CHICAGO	**5.6**	51.2	41.4	21.1
IOWA	**5.1**	41.7	12.3	16.2

TABLE 7: ABLATION STUDY ON WEIGHTING

CHICAGO ACCIDENT	K=30			K=40			K=50		
	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
NO-WEIGHT	0.255	0.441	**0.622**	0.265	0.417	0.613	0.274	0.401	0.595
SPATIALRANK	**0.257**	**0.444**	0.621	**0.268**	**0.420**	**0.614**	**0.278**	**0.403**	**0.599**
IOWA	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
NO-WEIGHT	0.531	0.304	0.617	0.557	0.264	0.591	0.573	0.225	0.546
SPATIALRANK	**0.540**	**0.309**	**0.618**	**0.563**	**0.268**	**0.600**	**0.585**	**0.232**	**0.550**
CHICAGO CRIME	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
NO-WEIGHT	0.364	0.484	0.660	0.379	0.466	0.642	0.390	**0.450**	**0.649**
SPATIALRANK	**0.373**	**0.491**	**0.665**	**0.380**	**0.467**	**0.647**	**0.392**	0.446	0.644

9.3 Computation Cost Comparison

We conduct comparisons with SOTA methods on average training time in seconds per epoch and inference time on the testing dataset in Table 6. The Chicago crime dataset and the Chicago accident dataset have the same input feature; thus, have equivalent training costs. In summary, SpatialRank trains faster than two SOTA baselines HintNet and GSNet on both datasets. It is only slower than HeteroConvLSTM but the training times of the two are on the same order of magnitude. The training phase of SpatialRank is slow because of computing nested L-NDCG loss function. Without extra cost on proposed optimization techniques, SpatialRank is significantly faster than all baselines in the inference phase. Given the improvement in prediction performance, we believe the cost of training time is acceptable, which will not affect the predicting efficiency of the proposed method.

9.4 Ablation study

We perform an ablation study on the parameter σ on the Iowa traffic accident dataset. The results are shown in Table 8. For K=30, 40, and 50, σ = 0.05 always gives the best performance. For K=50 there is a tie in Precision@K between $\sigma = 0.05$ and $\sigma = 0.1$. Compared with the Chicago datasets, the best σ changed from 0.1 to 0.05, suggesting that local ranking might be less challenging in the Iowa data. This makes sense as the Iowa data has a coarser resolution (5km), making it easier to separate potential hotspots from surrounding grid cells.

9.5 Cross-K function

We use the Cross-K function[15] with Monte Carlo Simulation[36] to evaluate the accuracy of predicted locations. The Cross-K function measures the spatial correlation between the predicted accident locations and true locations in our case. Specifically, we calculate the average density of predictions within every distance d of a true event in each day as shown in Eq.14:

$$\hat{K}(d) = \lambda_j^{-1} \sum_{i \neq j} I(d_{ij} \leq d)/n, \tag{14}$$

where λ is global density of event j, and $I()$ is an identity function which equals one if real distance d_{ij} is smaller than d, else equals zero. n is the number of events i. The results on the Chicago crime data and Iowa accident datasets are shown in Figure 4 and Figure 5, respectively. The grey curve represents the complete spatial

TABLE 8: ABLATION STUDY

IOWA	K=30			K=40			K=50		
	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG	NDCG	PREC	L-NDCG
$\sigma = 0.0$	0.532	0.305	0.610	0.560	0.261	0.597	0.578	0.228	0.554
$\sigma = 0.05$	**0.539**	**0.311**	**0.626**	0.559	**0.267**	**0.597**	**0.579**	**0.231**	**0.560**
$\sigma = 0.1$	0.532	0.307	0.603	**0.560**	0.264	0.572	0.578	**0.229**	0.555
$\sigma = 0.2$	0.528	0.303	0.585	0.561	0.262	0.566	0.580	0.227	0.526
$\sigma = 0.3$	0.523	0.300	0.579	0.559	0.263	0.572	0.580	0.227	0.530

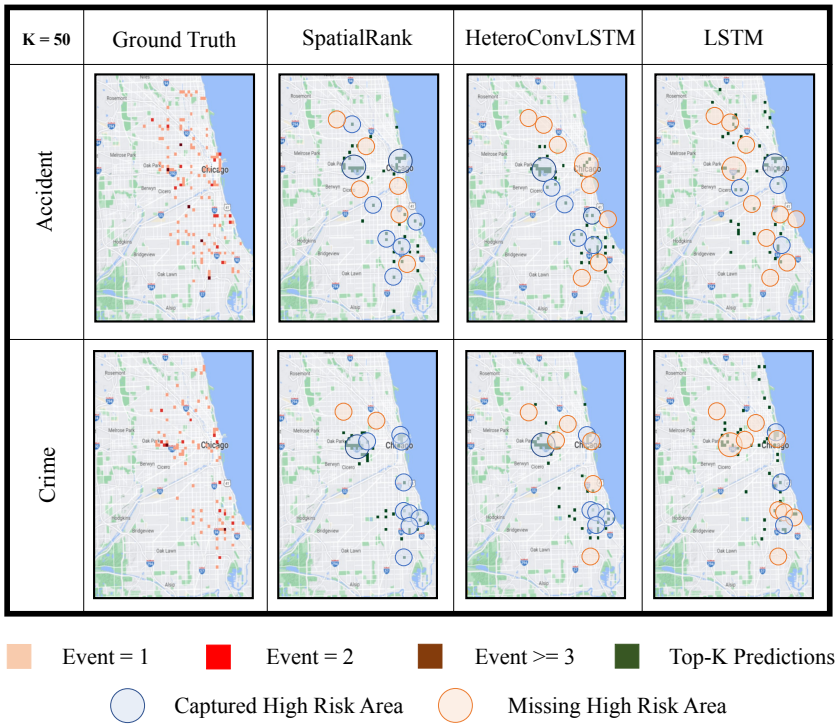

Figure 3: Case Study on Chicago Jan. 25^{th}, 2021.

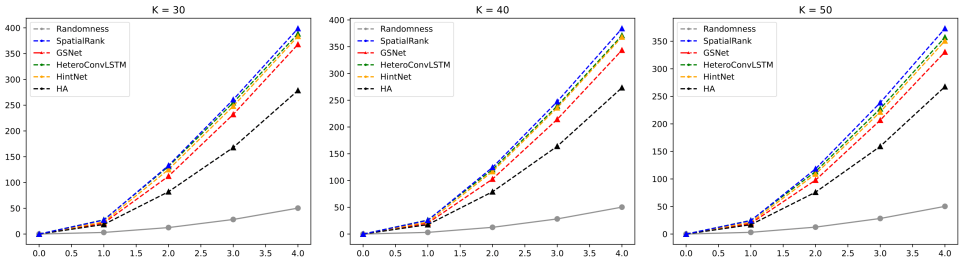

Figure 4: Comparison of Cross-K function for Chicago Crime.

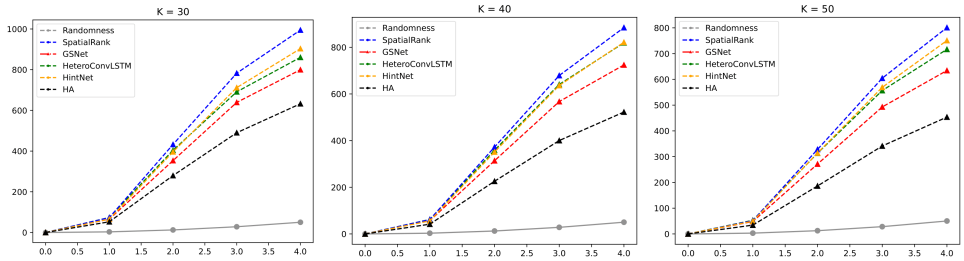

Figure 5: Comparison of Cross-K function for Iowa accident data.

randomness and we use it as a reference baseline. Higher curves are better. Our SpatialRank (blue) achieves the best predictions in both datasets, which indicates that the predictions of SpatialRank are significantly spatially correlated with ground truth.

9.6 Case Study

We present a successful prediction on Chicago using SpatialRank on Jan 25^{th} 2021 in Figure 3. There was a severe winter storm in the area of Chicago and 147 people were injured [13] and 99 severe crimes occurred. We compare predictions of SpatialRank with ground truth, Hetero-ConvLSTM, and LSTM. The top 50 riskiest locations in the predictions are chosen and labeled on the map. For easier visualization, the number of events greater than three in the ground truth map is changed to three, representing the riskiest location. To understand how those methods prioritize the riskiest locations, We define locations with events great than 1 as high-risk areas. The blue circle indicates this high-risk location is predicted correctly. Missing High-risk location is indicated as an orange circle. We can observe that SpatialRank captures most High-risk locations, while others are not able to find the potential risk nearby the downtown area. As a result, the overall NDCG score is improved in SpatialRank.