WhereTo: a Travel Recommendation System

Course project for CSE 6240: Web Search and Text Mining, Spring 2023

Divya Umapathy*
dumapathy3@gatech.edu
Georgia Institute of Technology

Harshvardhan Baldwa*
hbaldwa@gatech.edu
Georgia Institute of Technology

Mansi Bhandari*
mbhandari6@gatech.edu
Georgia Institute of Technology

Pankhuri Singh*
pankhuri.s@gatech.edu
Georgia Institute of Technology

ABSTRACT

Personalization is an important aspect in terms of Recommendation Systems, especially in the tourism domain. Tourism is heavily characterized by a tourist's preferences and better recommendations are made with more knowledge about the tourist's personality. Inspired by this, we aim to build a personalized travel recommendation system for the users and understand the impact of different features on the recommendations. In order to achieve this, we will be using two different publicly available datasets, Gowalla and Foursquare, and comparing our findings across the proposed collaborative filtering-based and spatio-temporal based recommendation systems. We have used the Blurring-Sharpening Process Model (**BSPM**) for collaborative filtering and the Spatio-Temporal Transformer Recommender (**STTR**) for sequential recommendations. We processed the datasets to get the information and model it accordingly as an input for both the methods. Results from both the methods are nearly similar to what was presented in the original papers.

KEYWORDS

Recommendation Systems, Travel Tourism, Spatio-Temporal, Collaborative Filtering, Transformers

1 INTRODUCTION

There is a plethora of data available with more and more tourists sharing their travel ratings and real experiences on the Internet. Generating big data for tourism, the industry has been utilizing this vast amount of information to give recommendations for travelers utilizing Recommendation Systems. By analyzing users' check-in travel locations and historical trajectories, we aim to build a recommendation system that incorporates these preferences in providing future travel destination recommendations for users. We use two datasets, Gowalla and Foursquare, which contain spatio-temporal information over the users' visited locations (check-ins) and other information like friendship links. Our first model, the Blurring-Sharpening Process Model (**BSPM**) for collaborative filtering, involves using a combination of a blurring operator and a sharpening operator to smooth out and enhance user-item preference matrices for better recommendation performance. Our second model, the Spatio-Temporal Transformer Recommender (**STTR**), is deep learning-based model that incorporates spatial and temporal information to make sequential recommendations by utilizing multi-head self-attention and positional encodings. For results, BSPM showed that as the SVD top k vectors for the Ideal low pass filter increase and as the IDL factor increases, the recall values increases; however, the time for training also increases. For STTR, we noticed an increase in recall with the increase in embedding dimension and size of historical trajectories. We also noted an optimal value in the number of negative samples for recall@5. However, comparison between these two models is difficult since BSPM is trying to provide a personalized set of new locations recommendations for the users, while STTR method is predicting what the next best location would be for any given user. Thus, both method have significantly different recall calculations and it would not be ideal to compare the values directly. The impact our project makes in its final form is that we follow and highlight the importance in the use of historical data to give accurate suggestion to users, leading to a state-of-the-art travel recommendation system.

2 LITERATURE SURVEY

Recent work in travel recommendations primarily focuses on understanding users' behavior and pattern in order to give more personalized travel recommendations. Some of the non-DL-based approaches included matrix factorization and collaborative filtering. Some of the DL approaches of spatial-temporal models involve using Recurrent Neural Network (RNN) since the data is sequential in nature.

The Score-Based Generative Model (SGM) [7] is a deep generative model that uses a score function of a diffusion process to generate samples. The score function provides the direction and magnitude of the diffusion at each point in space. The model is trained by maximizing the likelihood of observed data using the adjoint method to efficiently compute gradients with respect to the model parameters. One limitation is that it requires a large number of model parameters, which can make the training process slow and memory-intensive. Additionally, the SGM is sensitive to the choice of diffusion process and noise distribution, which can affect the quality of the generated samples. Finally, the SGM may not be suitable for tasks that require precise control over the generated samples, such as conditional generation or interpolation, as it tends to produce samples that are more diverse than specific. Our critique considers these limitations and the fact that it is a novel and exciting approach; however, we acknowledge that it is unclear how well these techniques scale to larger datasets. Nonetheless, this is related to our project scope as SGMs can be trained on a dataset of travel itineraries to learn the underlying patterns and regularities in the data. It can then be used to generate new travel itineraries

*All authors contributed equally to this research.

that are similar to the user's rating preferences, while still being diverse and novel.

The DeepMove model [5] is a prediction model that uses Recurrent Neural Networks (RNN) and an attention module to predict the next point of interest (POI) for a user. RNN is used to identify long-term patterns in user behavior and an attention module is sued to capture the most relevant POIs and long-term dependencies from the user history trajectories. What sets this model apart is its ability to identify meaningful information from sparse trajectory data and convert it into dense embeddings. It also worked well in identifying spatial-temporal relationships in incomplete and sequential data. Overall our critique is that the DeepMove model is a novel approach to predicting human mobility that effectively leverages both long-term and short-term patterns in user trajectories. However, a major drawback of this model is its incapability to identify dependencies between non-sequential check-ins or distant locations. Our project scope is related to this model as we intend to build a similar model using a sequence-based deep learning approach with an attention mechanism that can identify long and short-term preferences of users and locations, as well as dependencies between non-consecutive user check-ins and distant locations.

The Long and Short-Term Preference Model (LSTPM) [8] is used to predict the next point of interest (POI). This method considers both the user's long-term and short-term preferences as well as the geographic relationships between locations. The LSTPM includes two Long Short-Term Memory (LSTM) models to capture the user's long and short-term preferences. It also uses a geo-dilated RNN to capture the non-sequential geographic relationships among locations. The novel factor is that this approach considers the user's past behavior, general interests, and preferences, and takes into account both geographical and temporal factors to aggregate spatial preferences of recently visited locations. Our critique is that it is important to consider the geographical distance between two locations in POI recommendation, even if it does not appear sequentially in the user's history. Overall, the LSTPM model provides a comprehensive approach by combining long-term and short-term preferences, considering nearby POI visits, and finding relationships from non-adjacent locations. The model has a limitation in that it only aggregates locations for spatial preferences and does not consider temporal preferences, such as information from non-sequential data. However, non-contiguous visits are equally important in understanding users' behavior. Our project scope is related to this model as we intend to build a similar model that can identify both spatial and temporal patterns between users and locations, as well as dependencies between non-consecutive user check-ins and distant locations.

3 DATASET DESCRIPTION

Gowalla [3] and Foursquare [6] are the two publicly available large datasets that we will have used. Both of them were crawled from their respective platforms and are available publicly for usage. The datasets contain spatio-temporal check-ins for each user across different locations and their friendship networks, which were collected using their public API. As for the pre-processing part, data cleaning involved dropping duplicate records, converting all the timestamps to the same timezone and same format for ease of usage, and treating the outliers.

The descriptive statistics for both Gowalla and Foursquare datasets have been given in the table 1, and we can see that they show similar trends in terms of 'Check-ins per Location' metric. However, the two datasets vastly differ in terms of 'Check-ins per User' with Foursquare capturing more than Gowalla.

To feed data into the models, specific pre-processing was required. For model 1, we capped users at a minimum of 5 check-ins and a maximum of 30/35 check-ins for Foursquare/Gowalla datasets and then aggregated for user-location pairs due to memory constraints. Whereas, for model 2 we selected users with at least 5 check-ins and locations with at least 10 visits and then sequentially sorted data with timestamp to consider users' latest check-ins.

Table 1: Dataset Raw Statistics

Metric	Gowalla	Foursquare
No of Users	196,591	114,324
No of Check-ins	6,442,890	22,809,624
No of Social Links	950,327	607,333
No of Locations	1,280,969	3,820,891
Timeframe	Feb. 2009 - Oct. 2010	Apr. 2012 - Jan. 2014
Check-ins per Location		
Mean	5.03	5.97
Median	2	2
Range	1-5,811	1-23,520
Check-ins per User		
Mean	60.16	199.52
Median	25	142
Range	1-2,175	73-6,029

4 EXPERIMENTAL SETTINGS

We will use the user's travel history and interests to find the user score for each location. Overall, the system will perform a prediction task, predicting user scores for each location. After predicting the scores for each location, the highest score location will be the most recommended location for the user.

For evaluating the recommendation lists, we are using the Recall metric which by definition measures the proportion of relevant location recommendations among the total recommendations given. This metric is suitable for travel recommendation tasks as it measures relevance, accuracy, and order of recommendations, and is sufficient for evaluating travel recommendation system effectiveness. One thing to note is that the recall measure for method 1 is not just the ability to give relevant recommendations, but to give relevant recommendations which haven not been seen during the training.

For method 1, we use a 70-30 train-test split and for method 2, we do a split by interaction as 98-1-1 train-val-test split. And we utilize a MacBook Pro with 8GB RAM, Google Colab, Nvidia K80 GPU, and CoC Pace clusters for the code execution.

5 METHODS

5.1 Method 1: Blurring-Sharpening Process Models for Collaborative Filtering (BSPM)

We implemented a framework for collaborative filtering that uses blurring and sharpening processes to model user preferences [4].

This framework is based on the same idea as score-based generative models (SGMs) with some minor differences being that SGMs work on multiple images, while BSPM works on a single interaction matrix. The blurring model is based on the matrix factorization technique, which decomposes the user-item interaction matrix into two lower-rank matrices. The lower-rank matrices capture the underlying patterns and regularities in the data, which are then used to produce the blurred version of the original matrix. The sharpening model is based on the deep autoencoder architecture, which is a neural network that learns to compress and decompress data. The autoencoder takes the blurred data as input and tries to reconstruct the original matrix by learning a compressed representation of the data. The blurring-sharpening process models are trained using a maximum likelihood estimation approach, which minimizes the difference between the reconstructed data and the original data. The model also incorporates regularization techniques to avoid overfitting and improve generalization performance. We can see the workflow of this framework in the bottom of Figure 1. And, the additional data generated is the recommendation. Experimental results on several real-world datasets show that the BSPM framework outperforms several state-of-the-art collaborative filtering algorithms and GCNs in terms of Recall and NDCG and robustness to noise in the three benchmark datasets, Gowalla, Yelp2018, and Amazon-book. In addition, the processing time of this method is comparable to other fast baselines. Hence, we selected this to be one of our models for this project, in addition to it being a new and novel approach, with the paper being published in 2023. We referenced the authors' GitHub repository [1] aligned with our reference paper. Our hyperparameters used are SVD top k vectors and the Ideal low pass filter factor.

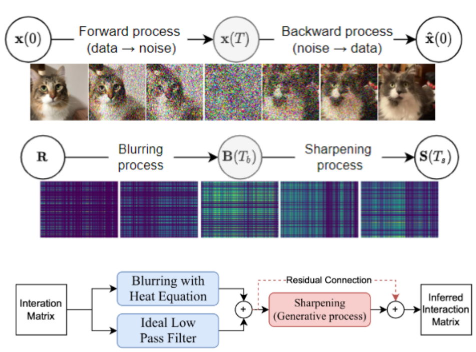

Figure 1: Comparison of SGMs and BSPMs; BSPM workflow

5.2 Method 2: Spatio-temporal Transformer Recommender (STTR)

Our second method is called Spatio-Temporal Transformer Recommender (STTR) [9]. This is a sequential deep learning-based recommendation system that analyses and identifies the spatio-temporal patterns between the users' trajectory sequences and their visited locations to predict the user's next Point of Interest (POI). Our dataset comprises user check-ins in the format of (userId, locationId, timestamp), with each userid and locationId being mapped to further user and location features. User trajectories are created based on their check-ins, providing valuable information regarding their daily movements, visiting frequencies of locations, and temporal visit patterns. This allows us to capture users' behavior and identify long-term preferences and dependencies of users and visited locations. Model architecture consists of multiple layers:

5.2.1 Spatio-Temporal Embedding Layer: Historical trajectories are constructed for each user based on their check-ins and fed into a multi-modal embedding layer. For our model, we have chosen the most recent 100 check-ins for every user. This layer generates user embeddings by encoding user, location, and time from the trajectories as latent factors into the embeddings. The resulting embeddings capture the user's spatial preferences and temporal periodicity and are fed into the transformer layer.

5.2.2 Transformer aggregation layer: This layer is used to process the user embeddings and extract deep complex spatio-temporal patterns between the visited locations. It collects information from relevant locations within the user trajectory and updates the location embeddings of each check-in interaction. The layer is composed of multiple transformer units, each with a Multi-Head Self-Attention mechanism that computes attention functions for all positions simultaneously, capturing both long-term and short-term dependencies from different locations at different timestamps in the sequence. This enables the model to capture information for non-adjacent locations and non-contiguous check-ins. The Position-wise Feed-Forward Network is used to introduce nonlinearity to the model.

5.2.3 Output Layer: An attention matcher computes the softmax probability of each location becoming the next POI for a user based on the location-hidden embedding and check-in representations. A balanced sampler is used for the model optimization, and it uses one positive sample and several negative samples to compute the error.

For training, we used Adam optimizer and cross-entropy as the loss function. Finally, after training the model, we made top-k predictions on each test and validation data point. We set k values of 1, 5, 10, and 20. Some of the hyperparameters are embedding dimensions of user embeddings, batch size, time scale (conversion of timestamp into standardization form), number of negative samples, and user history length. We referenced the authors' GitHub repository [2] aligned with our reference paper.

We selected the STTR model because it can solve the limitations of existing travel recommendation models, specifically the ability to capture spatial and temporal patterns in user trajectories and dependencies for nonconsecutive sequences and non-adjacent locations. It can also identify dependencies and patterns when data is spatial distant and temporal regular. Additionally, the model has been shown to outperform other state-of-the-art models for spatio-temporal data in terms of recall and robustness, making it a suitable choice for answering our research question.

6 EXPERIMENTS AND RESULTS

This section presents the experimental settings and the evaluation results for the mentioned methods BSPM and STTR on both Gowalla and Foursquare.

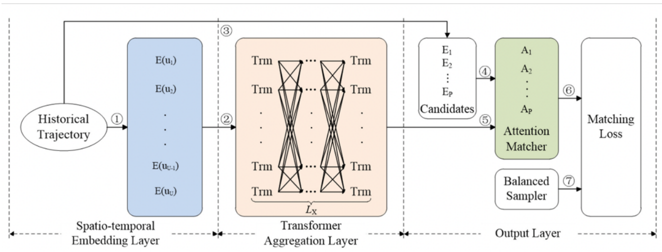

Figure 2: STTR Model Architecture

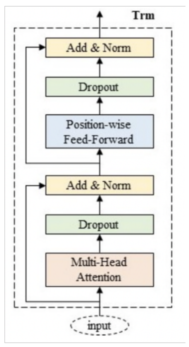

Figure 3: Transformer Unit Architecture

6.1 Method 1: BSPM

In order to find the best hyper-parameters for the method, we tuned our model by checking the recall values for different values of said hyperparameters. The hyperparameters and results obtained for them are listed below.

6.1.1 Top SVD vectors for Ideal low pass filter. Calculating these SVD vectors takes a lot of time, especially for larger datasets. We played around with many different algorithms and implementations of calculating top SVD vectors. The reference paper used a Python wrapper around the SVDLIBC library by Doug Rohde (sparsesvd), but we quickly realised that it is very slow for a big interaction matrix and hence shifted to TruncatedSVD function present in sci-kit learn package. Instead of using the standard ARPACK library to compute these vectors, we used the randomized algorithm which provides close enough vectors in a fraction of time. We could not find any python library which computes the truncated SVD decomposition on a GPU, which would have further reduced the training time. But after dealing with the implementation logistics, we noticed that increasing the number of top SVD vectors, we were getting better results, which is expected. Although, it meant trading off between training time and recall values. And since, we are using the same train-split to train our model, we also saved the SVD vectors as to save time when we wanted to keep this hyperparameter constant and change other hyperparameters. This saved us a lot of time during hyperparameter tuning. The three values we looked for were $[200, 448, 600]$

6.1.2 β factor. As evident in Figure 1, we are using two different methods to introduce noise in our data, namely Heat equation blurring and Ideal low pass filter. The ratio in which we add the results of these two methods is the second hyperparameter we dealt with. We checked the results for the following values of β: $[0.1, 0.2, 0.4, 0.5, 0.8]$

6.1.3 Residual Connection. After sharpening the data matrix, we tested another parameter, residual connection, which adds the interaction matrix before and after the sharpening process. Although, we realised that this hyperparameter for any value between 0 and 1 would give more preference to the training locations and hence we decided to drop the results generated by using this parameter.

After completing this study, we came to the conclusion that our model works best with the following configuration present in Table 2 The impact of hyper-parameters is present in Figure 4a and Figure 4b

Table 2: BSPM Final Configuration

Hyper-parameter	Value
Top **k** SVD vectors	600
β factor	0.8
Residual connection	No (0)

After this, we wanted to compare our model with the original reference paper. Although there was no information provided about the train-test split, we decided to model our own data (using 70/30 split) and tried that as an input. But due to memory issues (original BSPM model was storing the data in a matrix format, and not in a sparse matrix format), we were only able to run a small subset of the data. The original model was also using a different implementation of Recall@20, which we changed accordingly. The results for the same are provided in Table 3 using the best configuration on the Gowalla dataset. One reason, we suspect that BSPM paper is generating better results is because they are calculating the real SVD decomposition, while we, on the other hand, are working with approximated SVD vectors which can be calculated quite fast.

Table 3: Performance Comparison with Reference

Implementation	Recall@20
BSPM Paper	0.068379
Our Implementation	0.041916

6.2 Method 2: STTR

As part of developing and understanding the best possible configuration of STTR system, we performed hyperparameter tuning across three different metrics and evaluated the stability and performance of the model using Recall@5 and Normalized Discounted Cumulative Gain (NDCG@5). NDCG measures the effectiveness of the ranking system depending on the position of correct destination recommendations in the top recommendations. Note that NDCG is used only to understand the STTR model performance across different hyperparameters.

6.2.1 Embedding Dimension. We vary the dimension of user embedding which is fed into the transformer layer for a series of values $emb\_dim = [64, 128, 256, 512]$. As can be seen from the trajectories of Figure 5a, increasing the dimension leads to a relatively significant increase in the performance of the model for both Gowalla and Foursquare datasets. This indicates that a higher embedding size captures more user information which is expected. However,

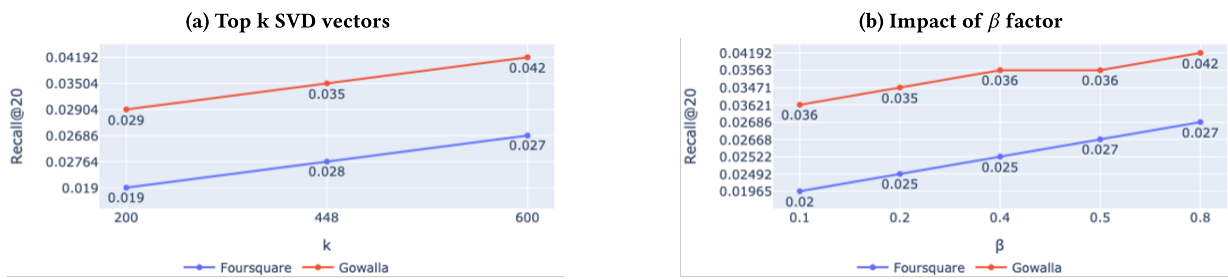

Figure 4: BSPM: Hyper-Parameter's Impacts

higher embedding size has a trade-off with training time of the model.

6.2.2 Number of Negative Samples. A series of number of negative samples $num\_neg = [1, 10, 15, 20, 30]$ were experimented with to understand the impact of negative samples in the balanced sampler. Figure 5b shows the trends observed across both the Foursquare and Gowalla datasets. Even though Foursquare suggests that 30 negative samples give the highest Recall@5 and NDCG@5, Gowalla observes a 17% decrease in Recall@5 and 24% decrease in NDCG from its best possible value at 15 samples. Thus, best performance is observed at 15 samples, and any value lower or higher leads to a drop in the performance.

6.2.3 Length of User History. As our model takes into account the user history to mine long-term preferences, length of history becomes an important metric. We observe the performance across different historical trajectories of $hist = [5, 10, 20, 40, 60, 80, 100]$ for metrics Recall@k and NDCG@k with $k \in \{5, 10, 20\}$ in Figure 6. Even though lower history length gives higher values there is a high possibility of overfitting as it takes time to reveal user preferences. Post history length of 10, we see a dip and then a consecutively increasing trend with increasing user history for both datasets.

Based on the above experiments, we tried various configurations and concluded that the configuration in Table 4 gave the best results and this is what we compared with the results of our reference paper across Recall@5, Recall@10 and Recall@20. As can be seen in Table 5, our configuration of parameters performs better than the reference paper for Gowalla dataset and also NYC[1] dataset (which we downloaded, pre-processed and used to compare with the reference paper). Note that NYC is a subset of Foursquare dataset filtered only for the New York City area. It is observed that pre-processing has an impact on results as our pre-processing differed from the reference paper a little in terms of feature selection.

6.3 Comparison between BSPM and STTR

Method 1 (BSPM) is observed to be significantly faster as it allows us to save the trained matrix and then tune it during testing. Also, multiple recommendations can be generated with reasonable recall. Method 2 (STTR), on the other hand, captures complex spatio-temporal patterns for user interactions and also captures long-term dependencies for visited locations. Although the data is sequential, it can capture the spatial-temporal patterns between non-adjacent locations and non-consecutive check-ins. This allows for more personalized recommendations. However, due to the significant difference in the nature of the two methods, it is difficult to compare them and their evaluation metrics. Method 1 generates a new interaction matrix by blurring and sharpening the input interactions to generate new possible interactions for the users. Whereas method 2 is a sequential recommendation system that understands the existing patterns within the user trajectories to understand where the user will go next. Although, as per Tables 3 and 5 the recall values for BSPM are one order smaller than STTR method, we can not quantitatively compare these two methods since both are solving a slightly different problem. BSPM is trying to provide users with a set of new locations that they might be interested in, while the STTR method is predicting what the next-best location would be for any given user. Thus, both methods have significantly different recall calculations and it would not be ideal to compare the values directly.

Table 4: STTR Final Configuration

Hyper-parameter	Value
Embedding Dimension	512
Number of Negative Samples	10
Length of Historical Trajectory	100
Learning Rate	3e-3
Dropout	0.1
Epochs	100

7 CONCLUSION

We proposed two methods in order to recommend the next travel destination for the user utilizing two spatiotemporal datasets. However, due to the limitation of computation resources, the current models have been trained on smaller versions of the datasets and thus there may have been a bias introduced during the experimentation phase. Also, embedding generation and train time for STTR model is quite significant, and there may be a potential for optimizing both.

Several future research directions can be explored with a focus on including psychological parameters such as user personality and emotions, making the recommendations more personalized for each user. The exploitation of social links of the users can also give insights into personalization of the travel recommendations along with potential user emotion extraction via their tweets. A potential area of exploration is the inclusion of spatial distance

[1]http://www-public.imtbs-tsp.eu/~zhang_da/pub/dataset_tsmc2014.zip

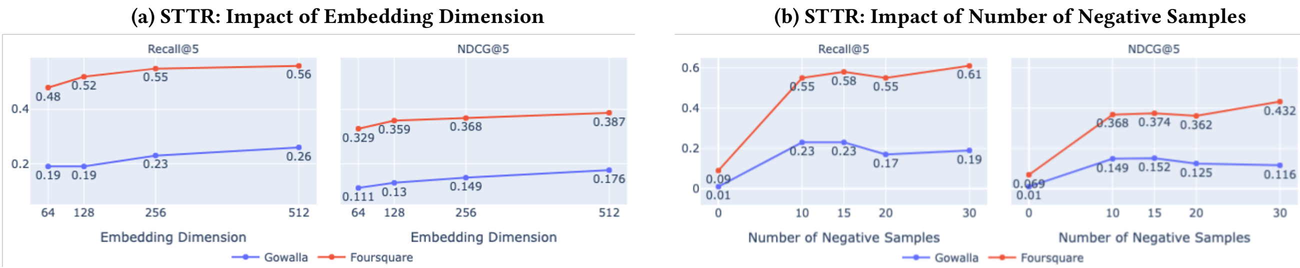

Figure 5: STTR: Hyper-Parameter's Impacts

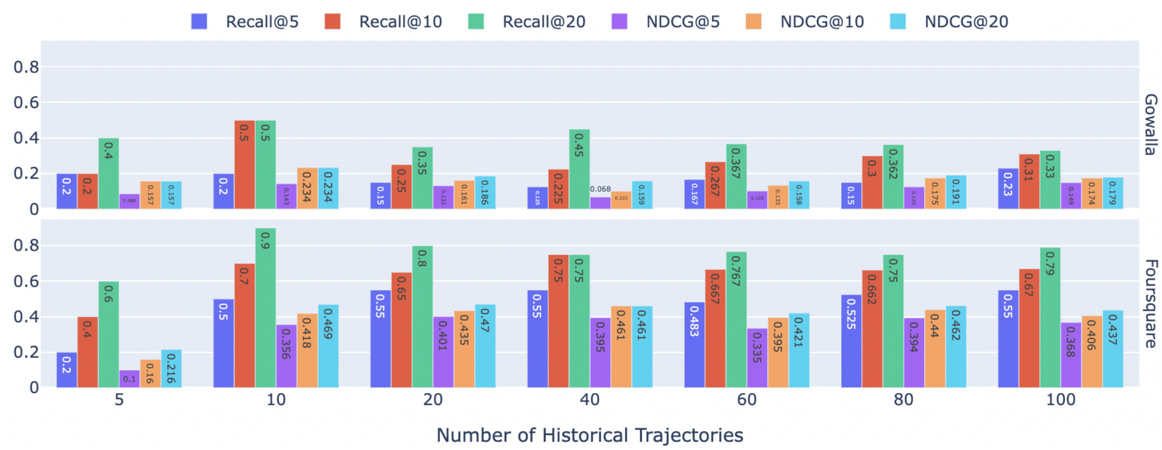

Figure 6: STTR: Impact of User History Length

Table 5: Method 2: Recommendation Performance Comparison with Reference

	Gowalla			NYC		
	Recall@5	Recall@10	Recall@20	Recall@5	Recall@10	Recall@20
STTR Paper	0.35	0.43	0.53	0.53	0.65	0.73
Our Implementation	0.39	0.47	0.58	0.56	0.7	0.8

introducing time and spatial preferences of users. Also, method 1 has the potential of improving it's current blurring and sharpening process to better learn from the data which in turn can improve the performance.

8 CONTRIBUTION

All team members have contributed equally for the project.

REFERENCES

[1] [n. d.]. GitHub - jeongwhanchoi/BSPM: Blurring-Sharpening Process Models for Collaborative Filtering, SIGIR'23 — github.com. https://github.com/jeongwhanchoi/BSPM. [Accessed 28-Apr-2023].

[2] [n. d.]. GitHub - skmxu/STTF-Rec: Space-Time Transformer Recommender: Next Location Recommendation with Attention Mechanism by Mining the Space-Time Relationship between Visited Locations — github.com. https://github.com/skmxu/STTF-Rec. [Accessed 28-Apr-2023].

[3] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and Mobility: User Movement in Location-Based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Diego, California, USA) *(KDD '11)*. Association for Computing Machinery, New York, NY, USA, 1082–1090. https://doi.org/10.1145/2020408.2020579

[4] Jeongwhan Choi, Seoyoung Hong, Noseong Park, and Sung-Bae Cho. 2023. Blurring-Sharpening Process Models for Collaborative Filtering. arXiv:2211.09324 [cs.IR]

[5] 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 the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1459–1468. https://doi.org/10.1145/3178876.3186058

[6] Huiji Gao, Jiliang Tang, and Huan Liu. 2012. GSCorr: Modeling Geo-Social Correlations for New Check-Ins on Location-Based Social Networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management* (Maui, Hawaii, USA) *(CIKM '12)*. Association for Computing Machinery, New York, NY, USA, 1582–1586. https://doi.org/10.1145/2396761.2398477

[7] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. 2021. Maximum Likelihood Training of Score-Based Diffusion Models. arXiv:2101.09258 [stat.ML]

[8] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to Go Next: Modeling Long- and Short-Term User Preferences for Point-of-Interest Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 01 (Apr. 2020), 214–221. https://doi.org/10.1609/aaai.v34i01.5353

[9] Shuqiang Xu, Qunying Huang, and Zhiqiang Zou. 2023. Spatio-Temporal Transformer Recommender: Next Location Recommendation with Attention Mechanism by Mining the Spatio-Temporal Relationship between Visited Locations. *ISPRS International Journal of Geo-Information* 12, 2 (2023). https://doi.org/10.3390/ijgi12020079