Predicting Human Mobility via Attentive Convolutional Network

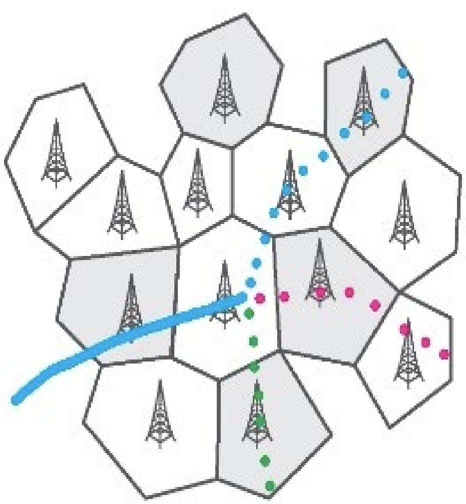
Congcong Miao, Ziyan Luo, Fengzhu Zeng, Jilong Wang

Department of Computer Science and Technology, Tsinghua University

1. Introduction
2. Related Work
3. Solution
4. Experiment Result
5. Conclusion

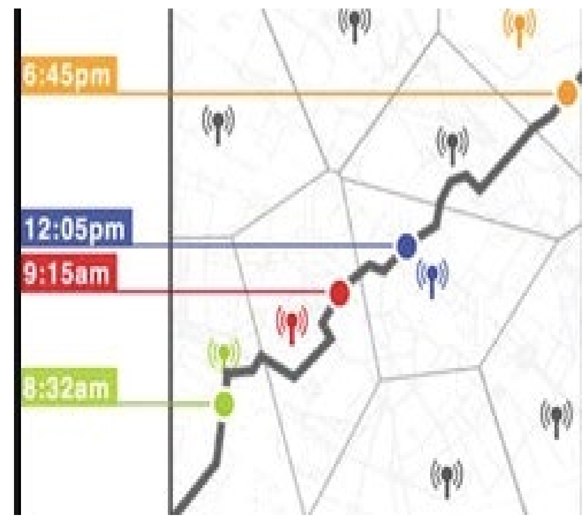
1. Introduction

- Human mobility prediction is of great importance for various applications.
  - Intelligent traffic management
  - Smart city planning
  - Personalized recommendation



Mobility management

Estimating travel demand



Recommendation

1. Introduction

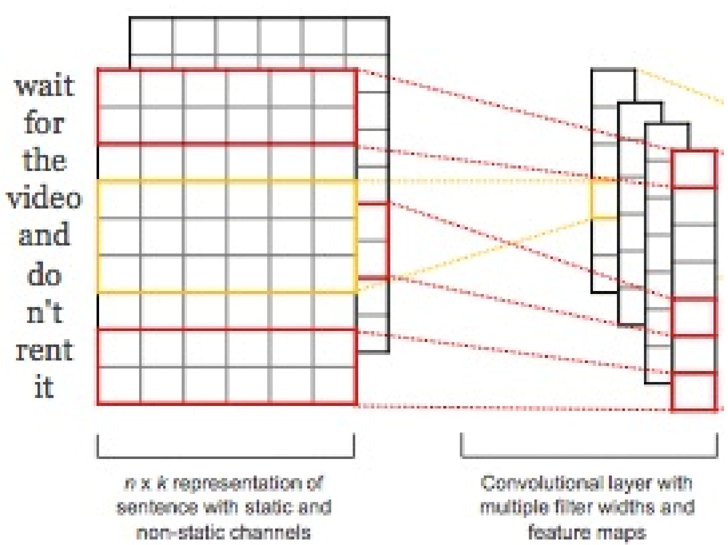
- Three unique characteristics on geo-tagged social media (GTSM) data
  - Extreme data sparsity: low-sampling and generated only when the users want to share their locations
  - High order sequential patterns: containing complex dependency relationships of human mobility and not all adjacent GTSM data has dependency relationships.
  - Evolving preference: human taste (i.e., long-term preference) for tagging is changing over time

2. Related work

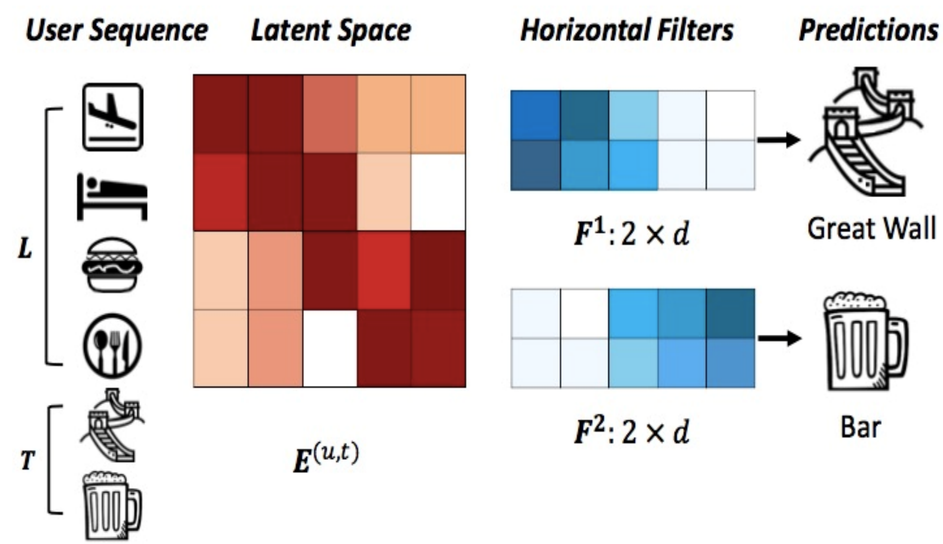
- Human mobility prediction
  - Pattern based approach:
    - Matrix factorization (non-negative MF, WMF)
    - Tensor factorization (TF)
  - Ignore sequential transition regularities and long-term preference
  - Model based approach:
    - Markov models (MC, HMM)
    - Recurrent neural network (ST-RNN, DeepMove)
  - Unable to model high-order sequential pattern

2. Related work

- Convolutional Neural Network (CNN)
  - Sequential modeling
    - Natural language processing (NLP)
    - Item recommendation



NLP



Recommendation

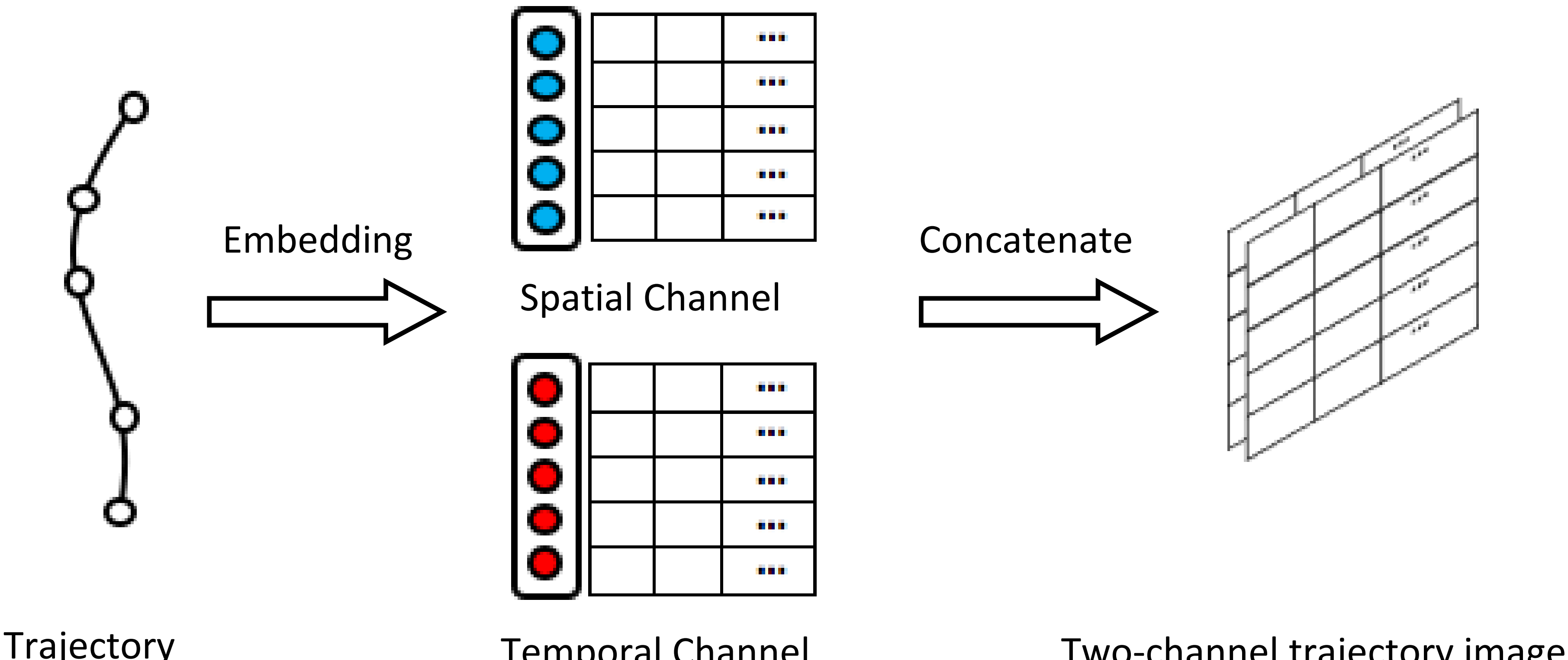
3. Solution

- DEFINITION 1 (Trajectory Sequence)
  - We define a spatio-temporal point **q** as a tuple of location **p** and time **t**, **e.g. q = (p,t)**. For a user ID **u**, trajectory sequence **T** is the aggregation of spatio- temporal points, i.e., $\mathbf{T_u = q_1 q_2 \cdots q_n}$.
- DEFINITION 2 (Trajectory)
  - Given a trajectory sequence $\mathbf{T_u}$ for a user **u**, trajectory is a subsequence of $\mathbf{T_u}$. The **k**-th trajectory with length **L** can be represented as $\mathbf{T_{u,k} = q_k q_{k+1} \cdots q_{k+L-1}}$.

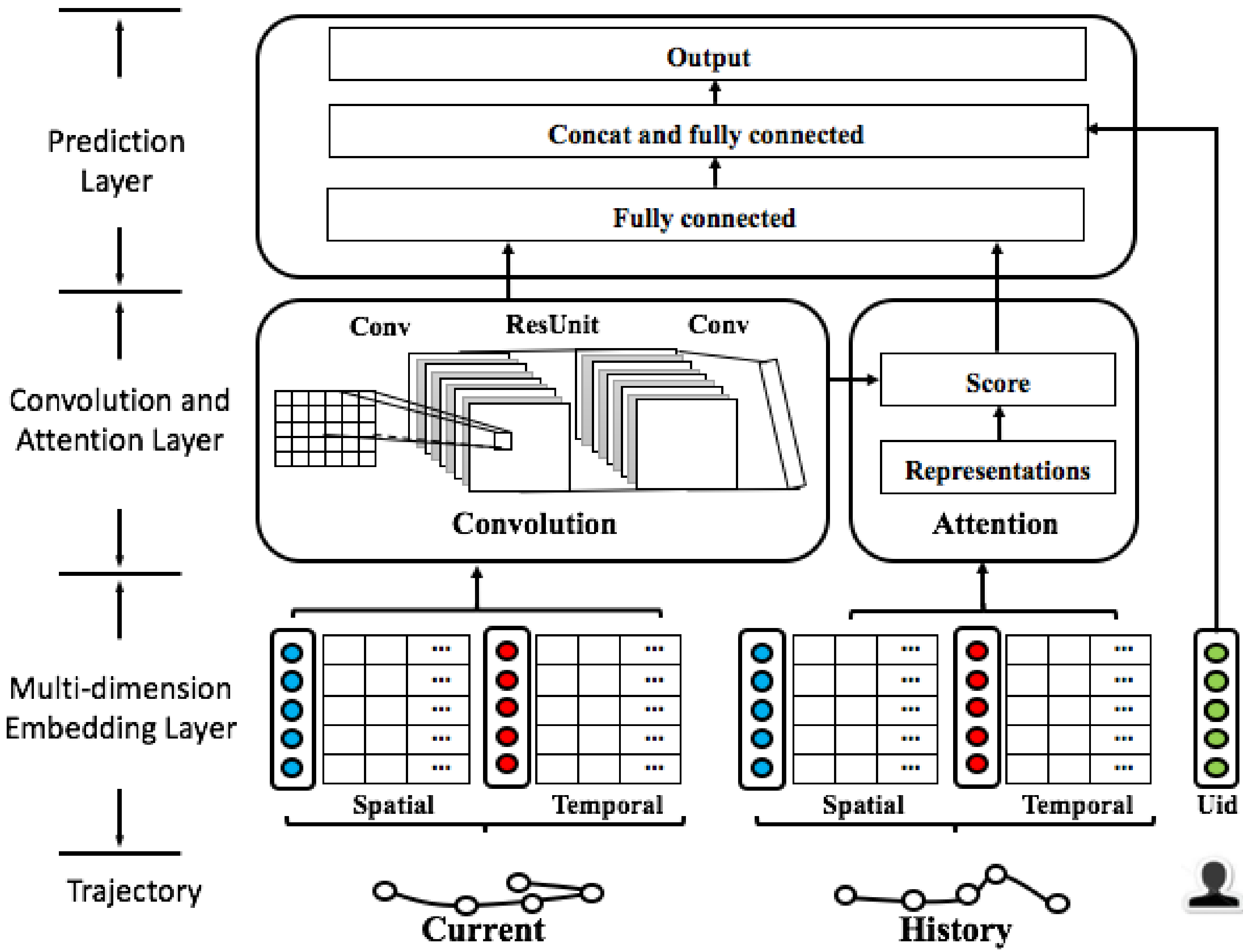
3. Solution

- Problem description
  - Given the trajectory $\mathbf{T}_{u,k}$, predict the **next spatial context: Location.**
  - The trajectory sequence of each person is divided into two parts: current trajectory and up-to-date historical trajectory
    - **predict the next location** of the current trajectory with the help of current trajectory and historical trajectory.

- CNN based mobility prediction
  - Trajectory embedding → Trajectory map
  - Use convolution operation to search for sequential patterns as local features of the image.

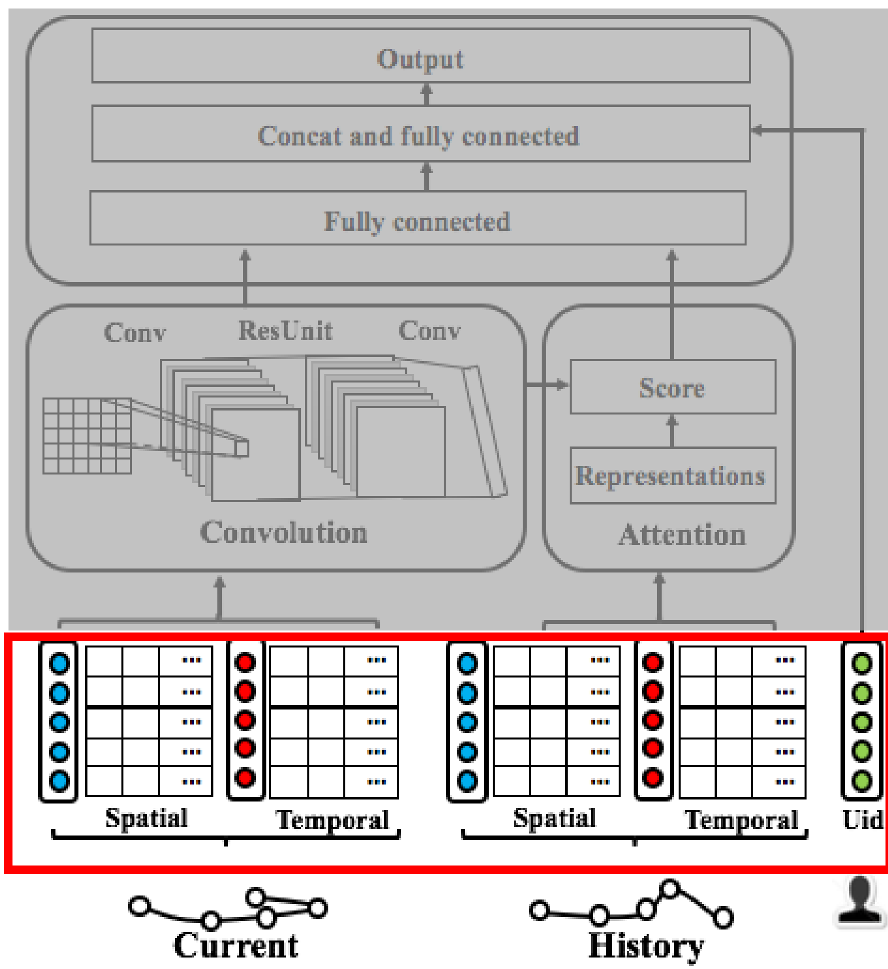


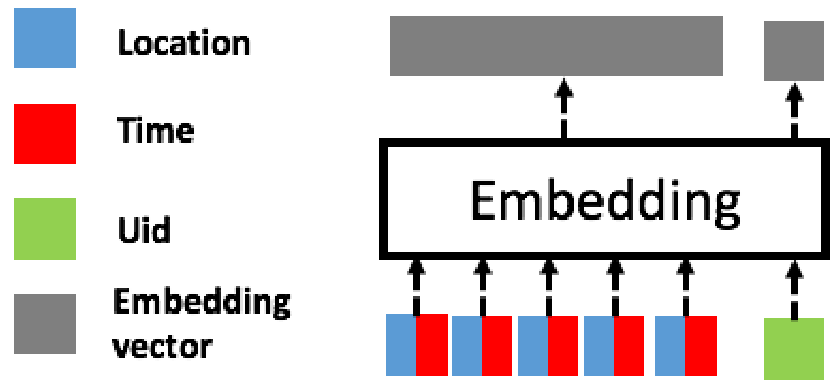
3. Solution



Architecture of attentive convolutional network (ACN)

ACN—Multi-dimension Embedding

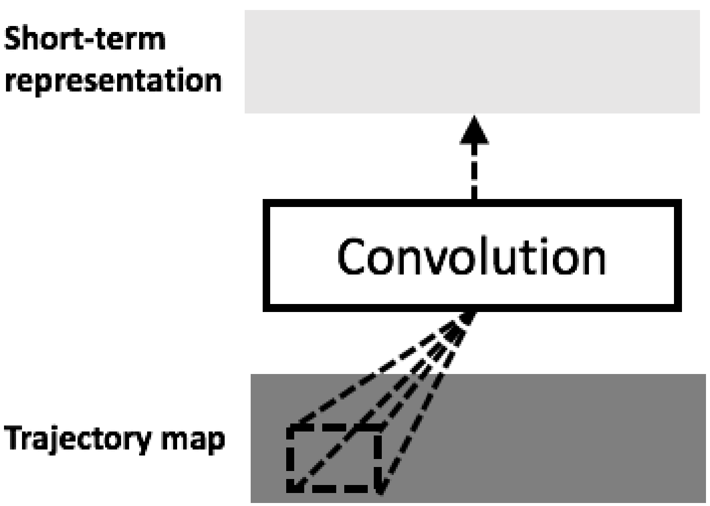




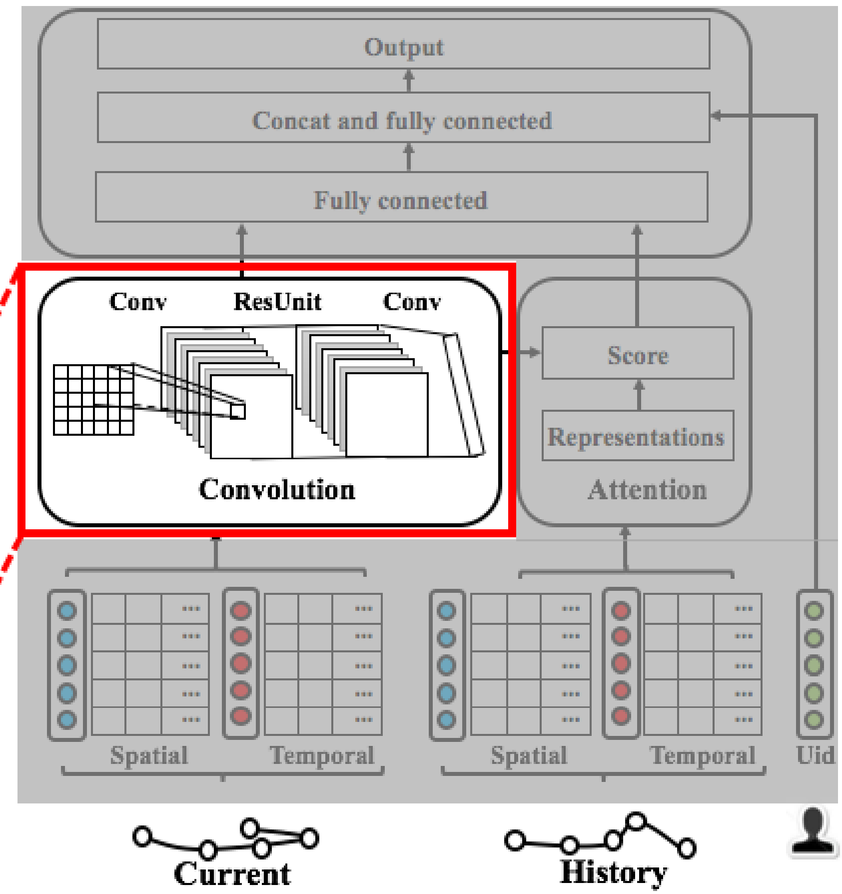
- **Multi-dimension embedding**
  - One-hot encoding
    Location, time, uid
  - Transformation matrix
  - Concatenate layer

3. Solution

ACN—Convolution module

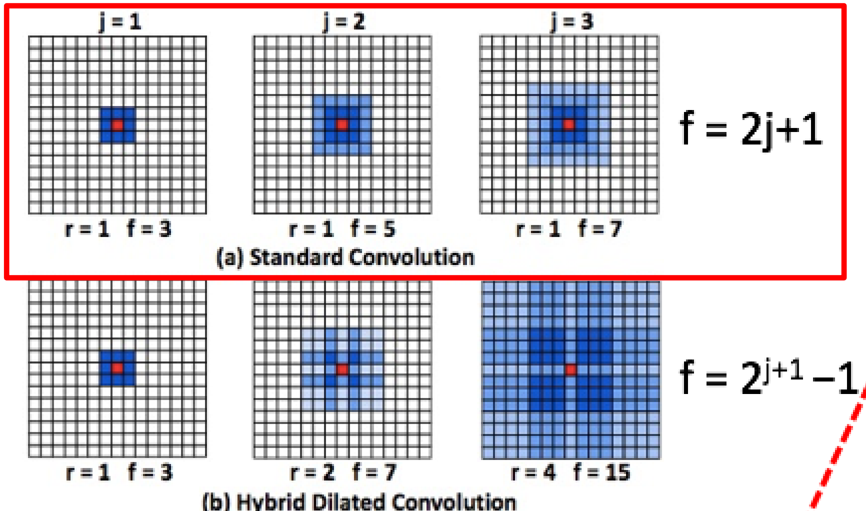


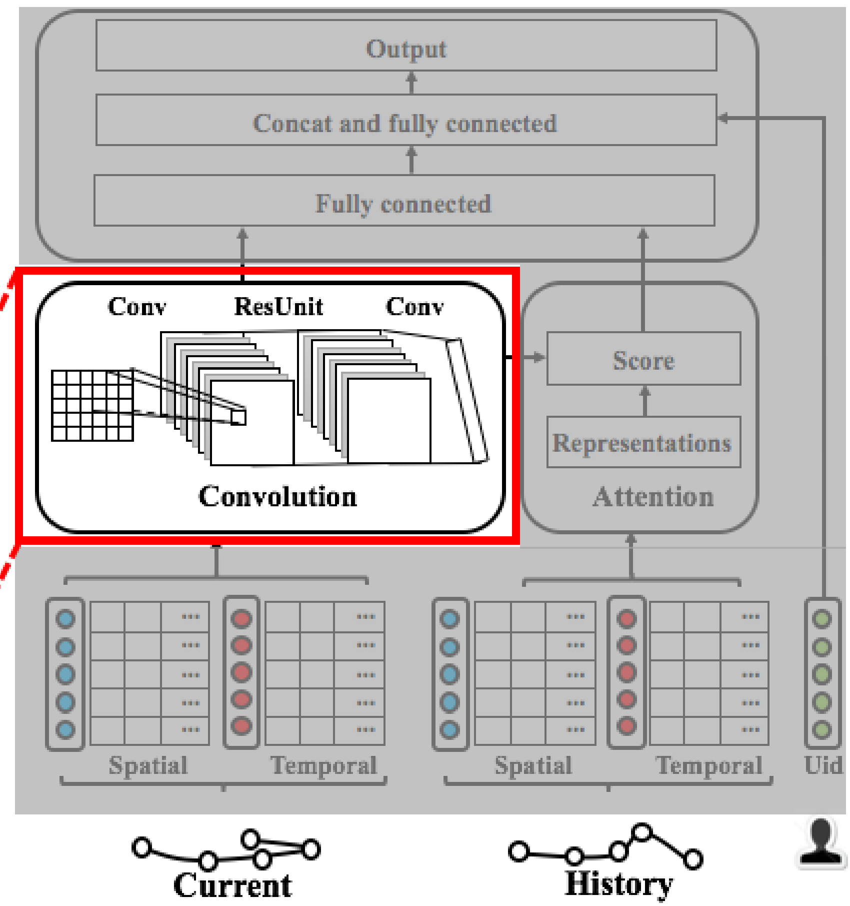
- **Convolution module**
  - Hybrid dilated convolution
  - Separable and shared convolution
  - Residual network



3. Solution

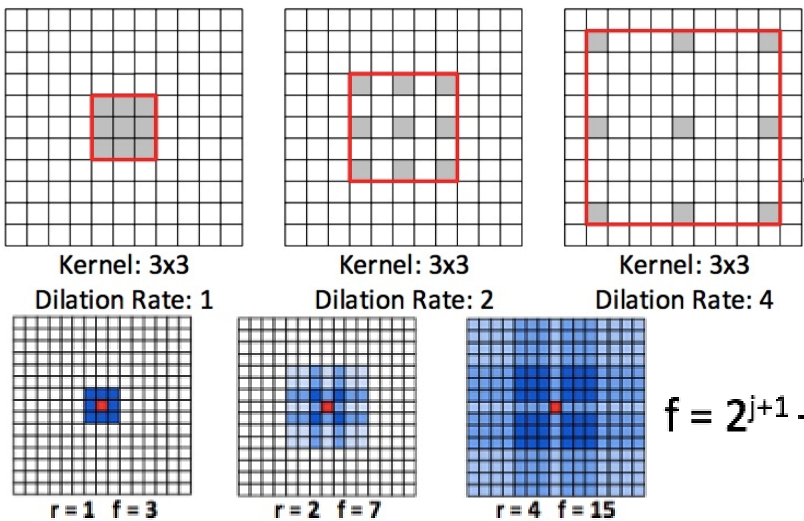
ACN—Convolution module





- **Convolution module**
  - **Hybrid dilated convolution**
  - Separable and shared convolution
  - Residual network

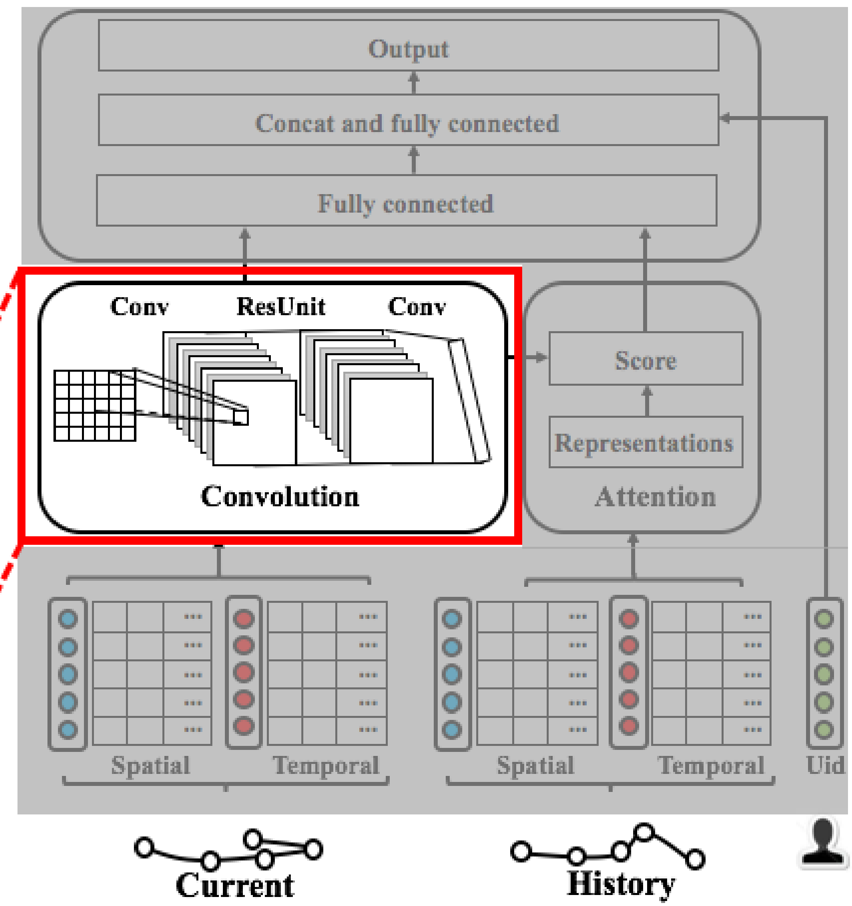
ACN—Convolution module



(b) Hybrid Dilated Convolution

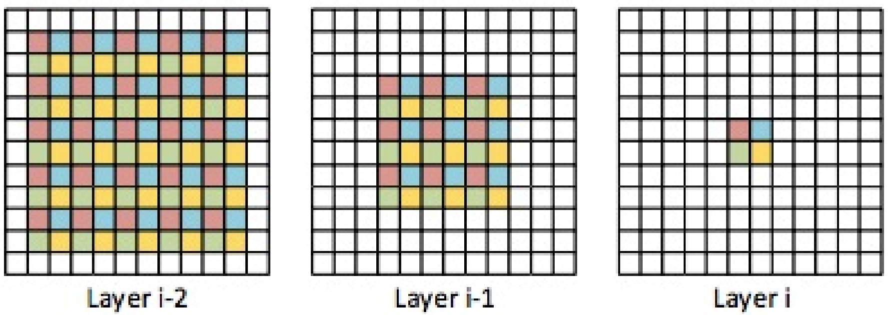
$$f = 2^{j+1} - 1$$

- **Convolution module**
  - **Hybrid dilated convolution**
  - Separable and shared convolution
  - Residual network



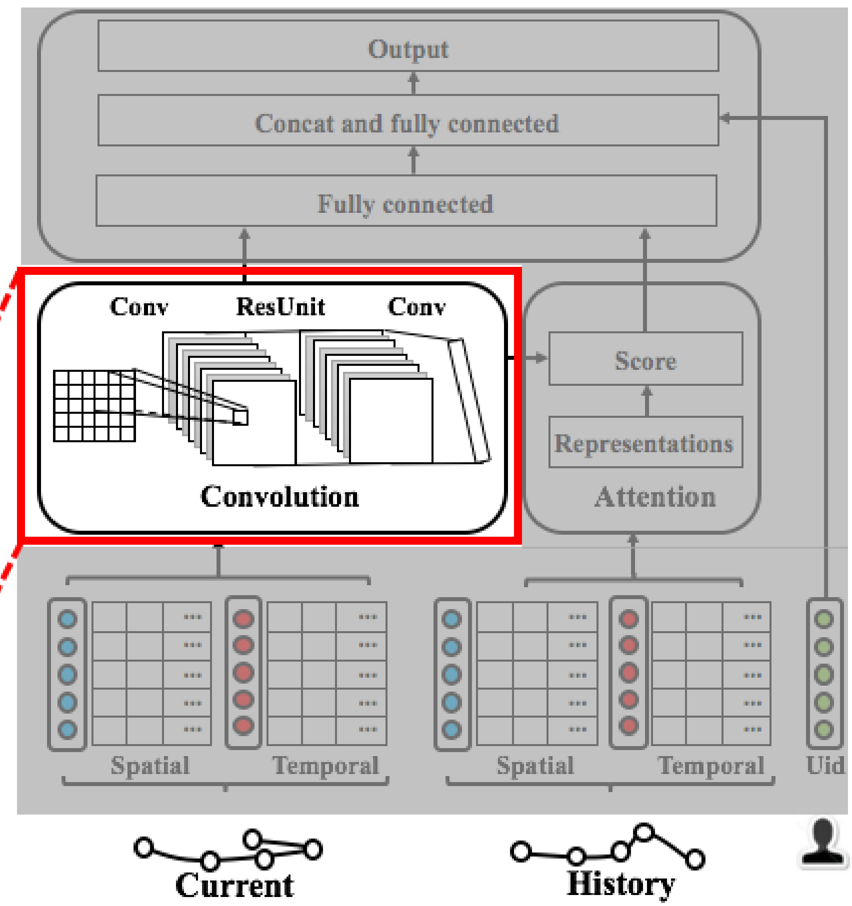
3. Solution

ACN—Convolution module

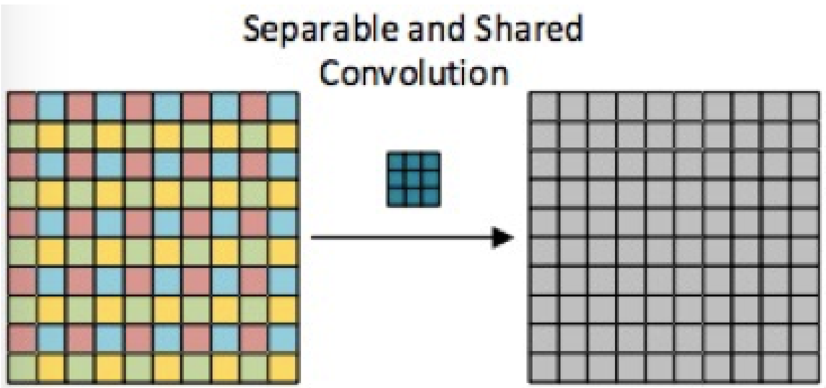


An illustration of gridding artifacts. Dilated convolutions with kernel size of 3 × 3 and a dilation rate of r = 2

- **Convolution module**
  - Hybrid dilated convolution
  - **Separable and shared convolution**
  - Residual network

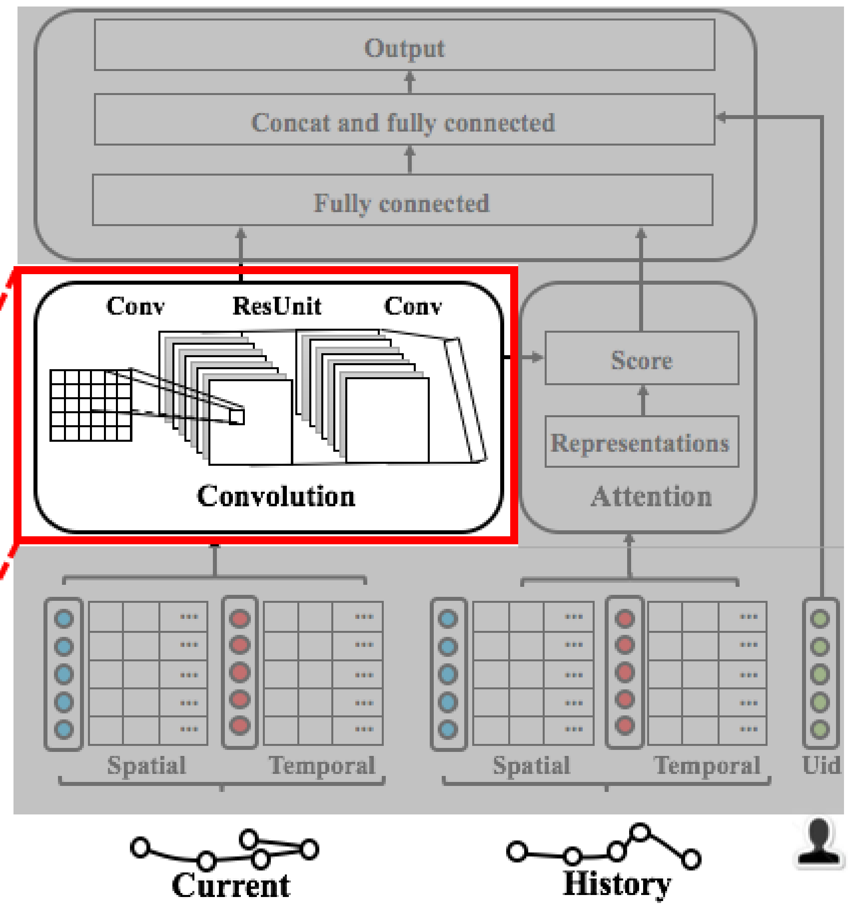


ACN—Convolution module



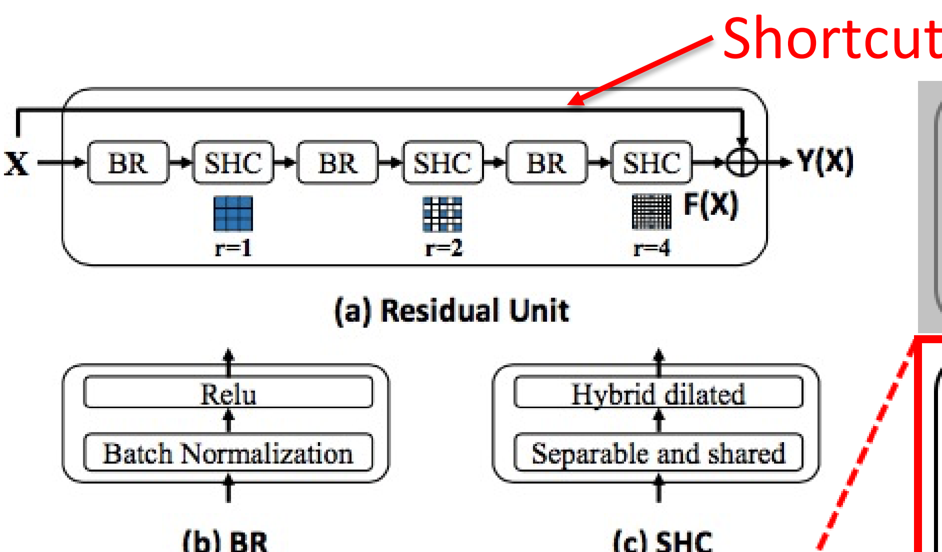
Degridding method to improve the consistency of dilated convolution

- **Convolution module**
  - Hybrid dilated convolution
  - **Separable and shared convolution**
  - Residual network



3. Solution

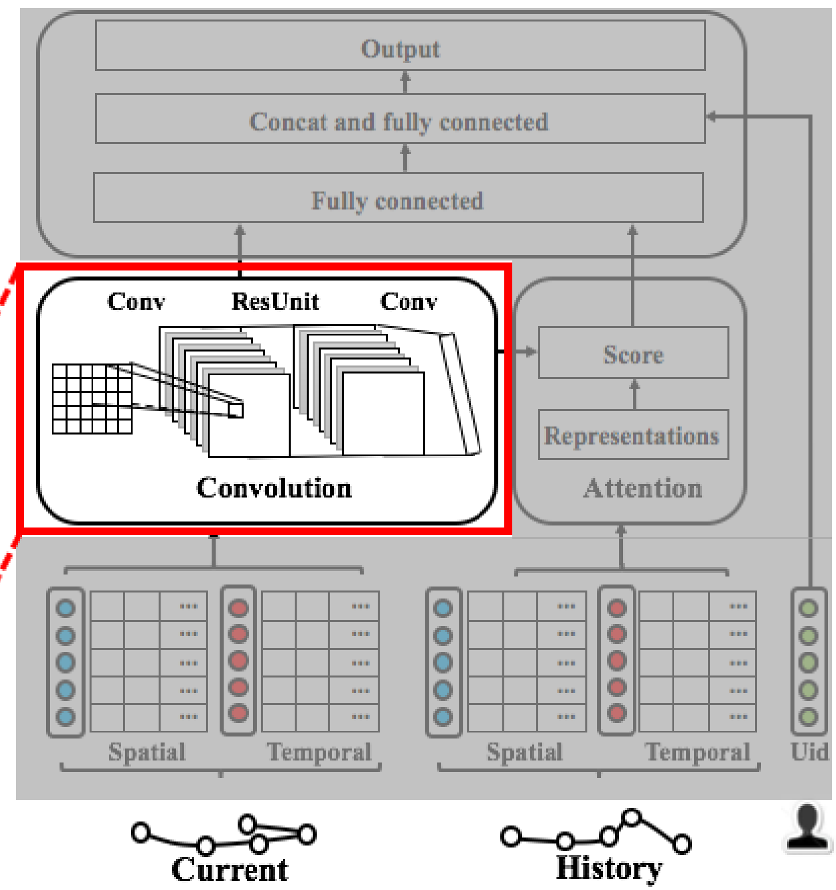
ACN—Convolution module



(a) Residual Unit

(b) BR

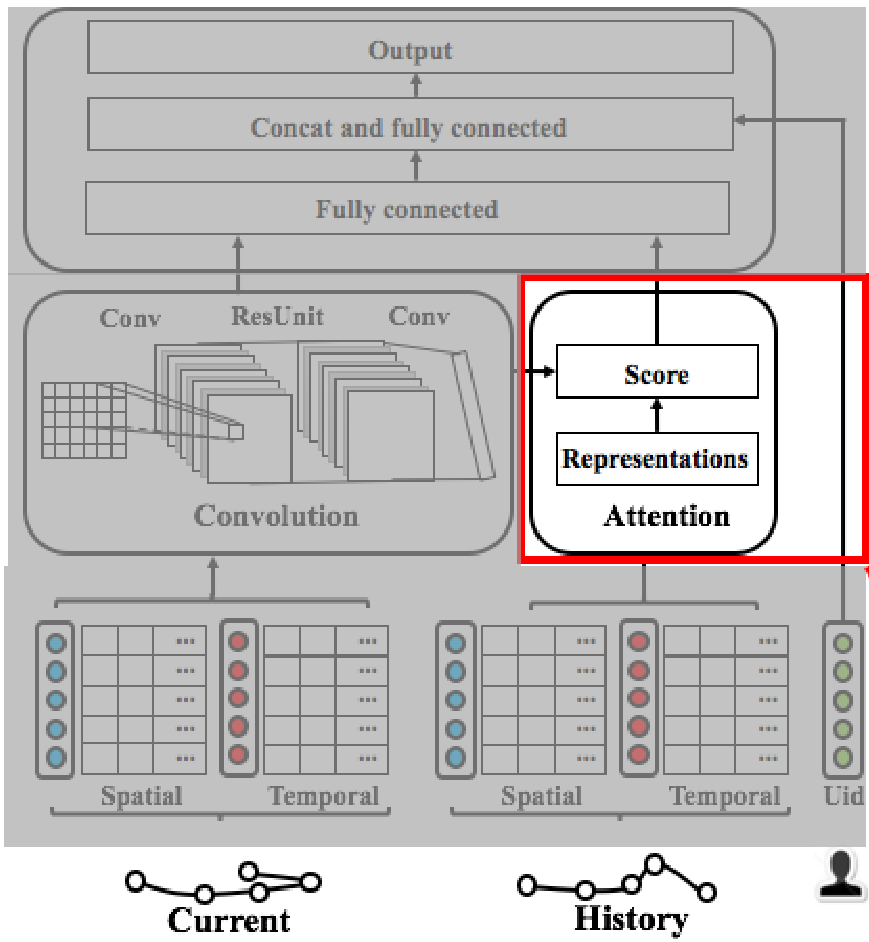
(c) SHC

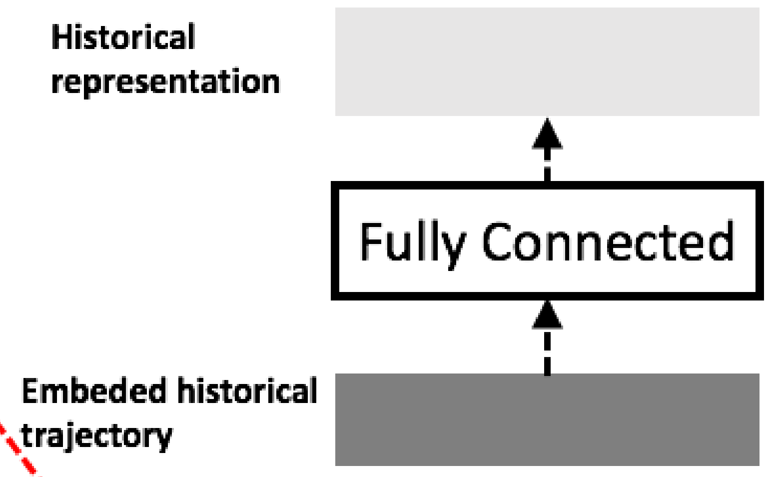


- **Convolution module**
  - Hybrid dilated convolution
  - Separable and shared convolution
  - **Residual network**

3. Solution

ACN—Attention module

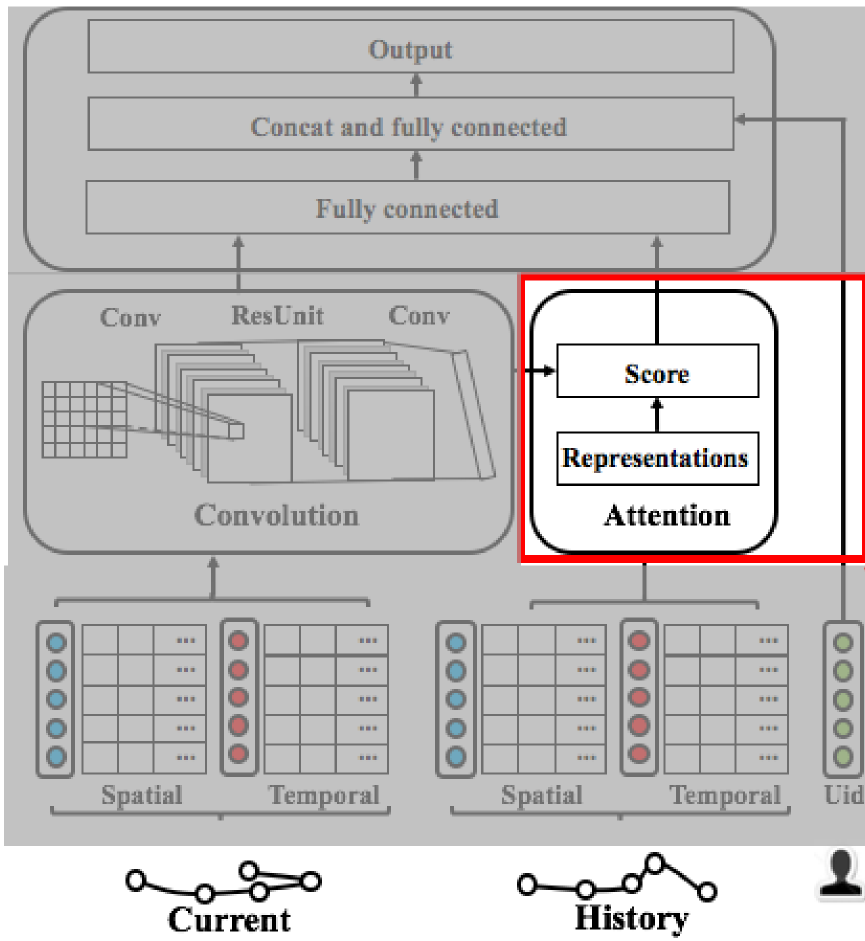


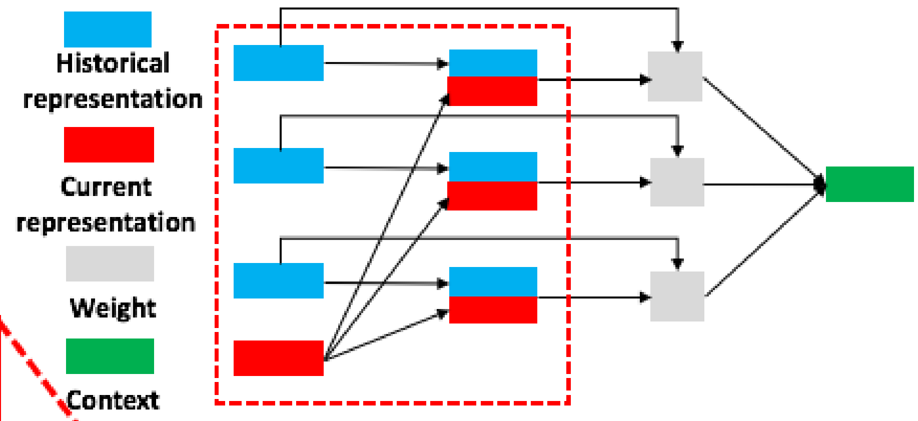


- **Attention module**
  - **Historical representation layer**
  - Score function
    - Calculating similarity
    - Soft-max
    - Weighted sum

3. Solution

ACN—Attention module



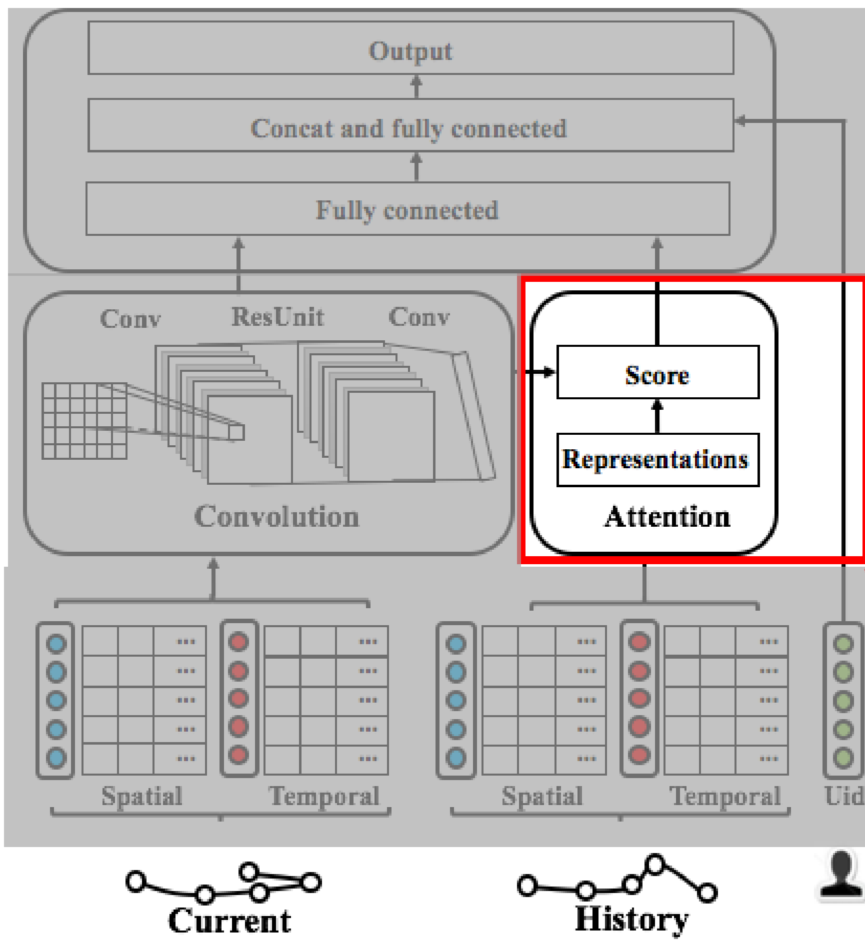


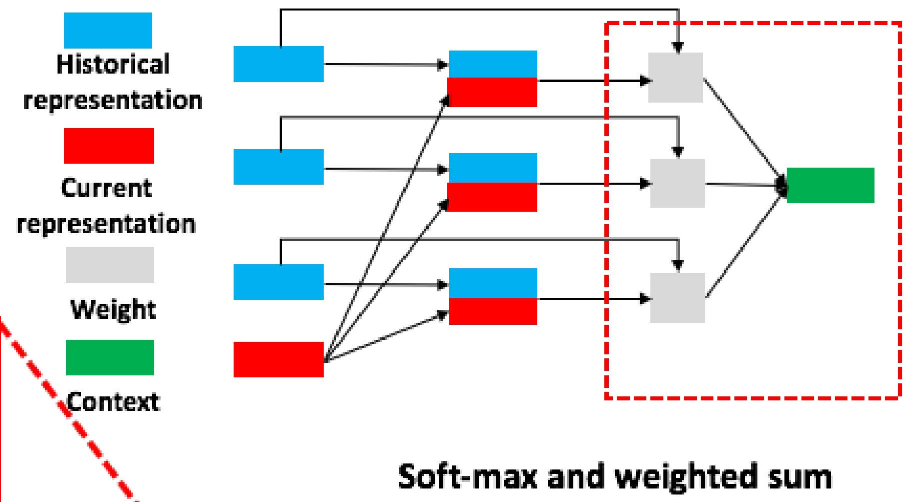
Calculating similarity

- **Attention module**
  - Historical representation layer
  - **Score function**
    - Calculating similarity
    - Soft-max
    - Weighted sum

3. Solution

ACN—Attention module

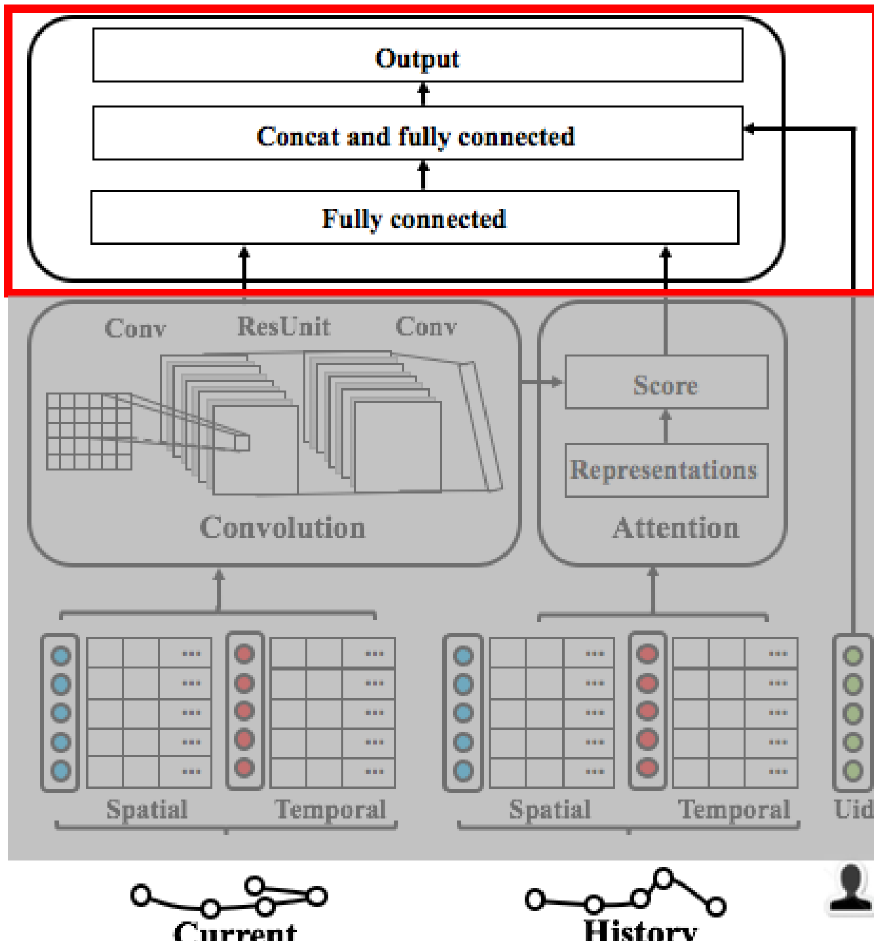




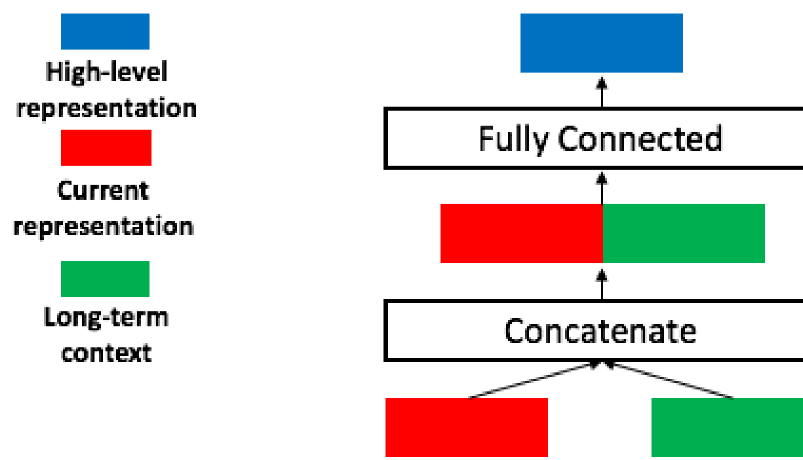
- **Attention module**
  - Historical representation layer
  - **Score function**
    - Calculating similarity
    - Soft-max
    - Weighted sum

3. Solution

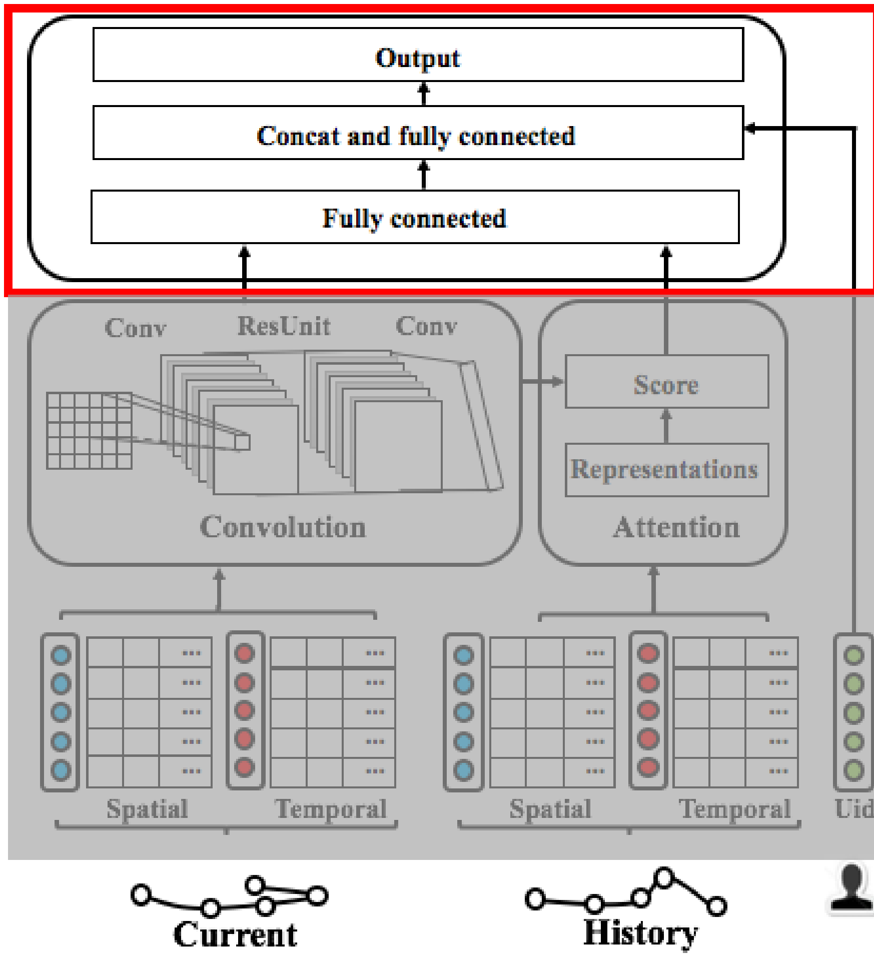
ACN—Attention module



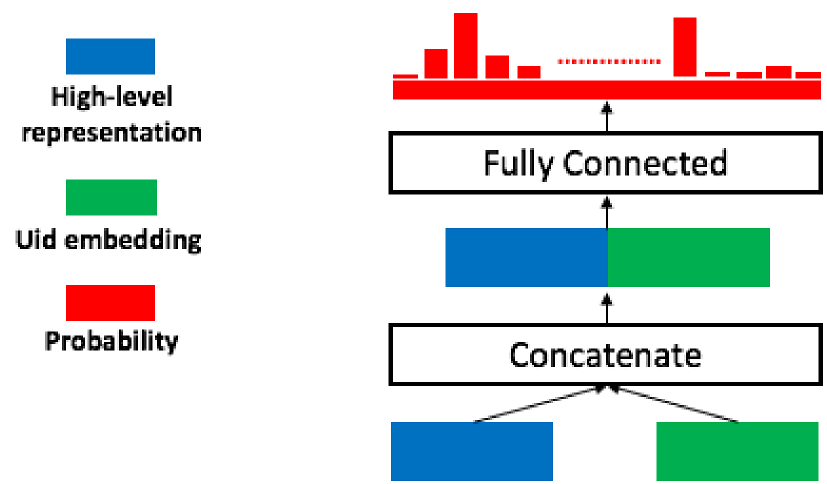
- **Prediction layer**
  - **Current representation and long-term context**
    concatenate + fc
  - **Human personal preference**
    concatenate + fc + soft-max



ACN—Attention module

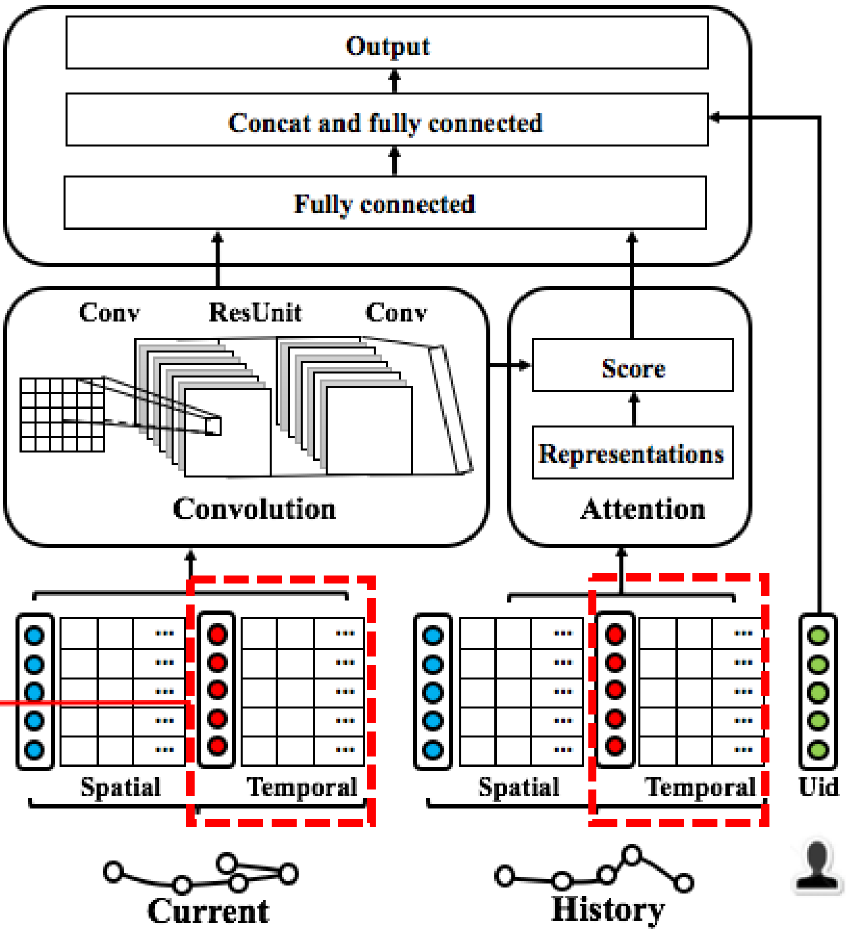


- **Prediction layer**
  - Current representation and long-term context
    concatenate + fc
  - **Human personal preference**
    concatenate + fc + soft-max



3. Solution

ACN



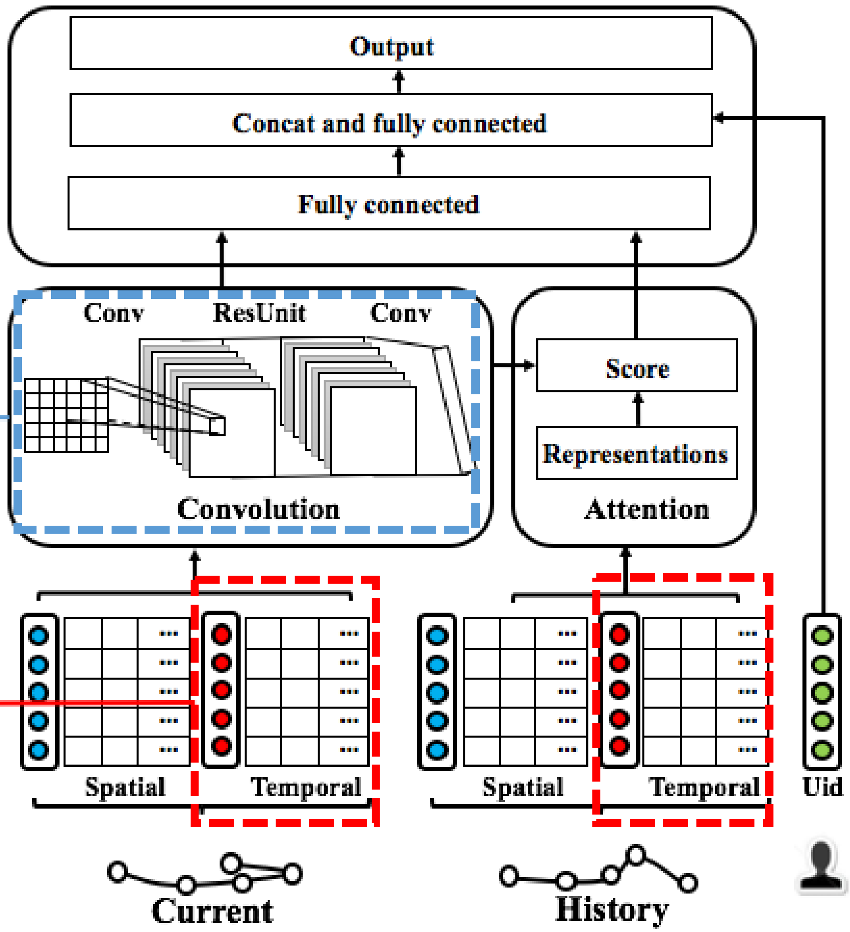
The time embedding captures time-dependent relations.

3. Solution

ACN

The convolution module captures high-order sequential pattern.

The time embedding captures time-dependent relations.

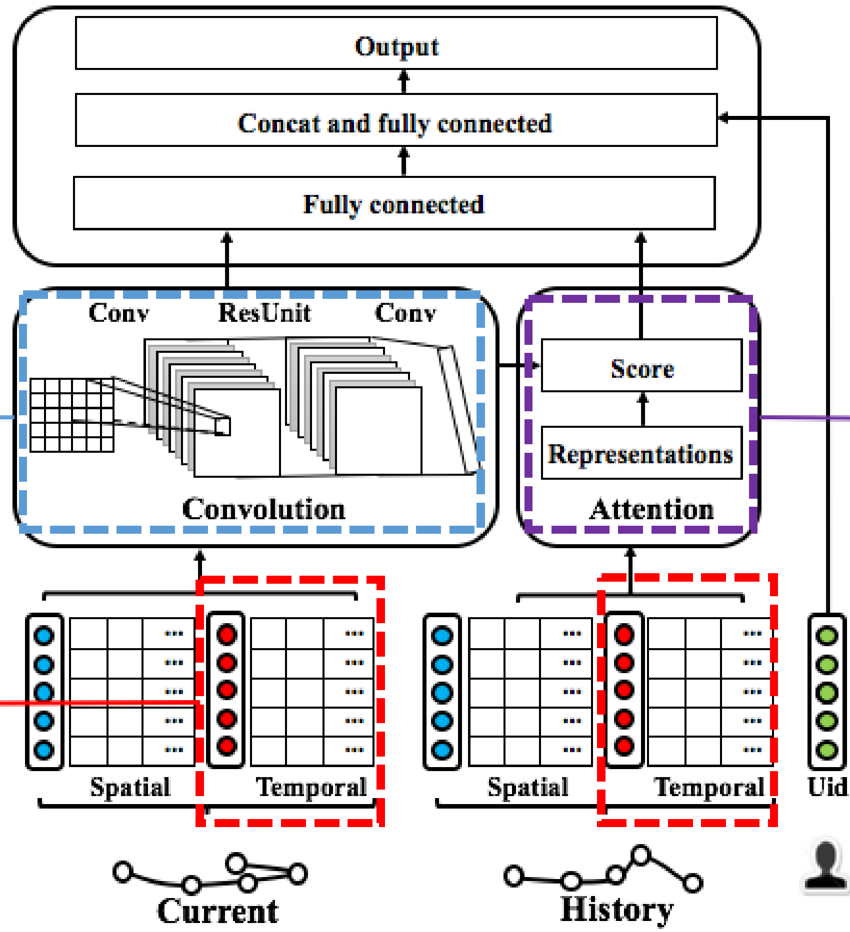


3. Solution

ACN

The convolution module captures high-order sequential pattern.

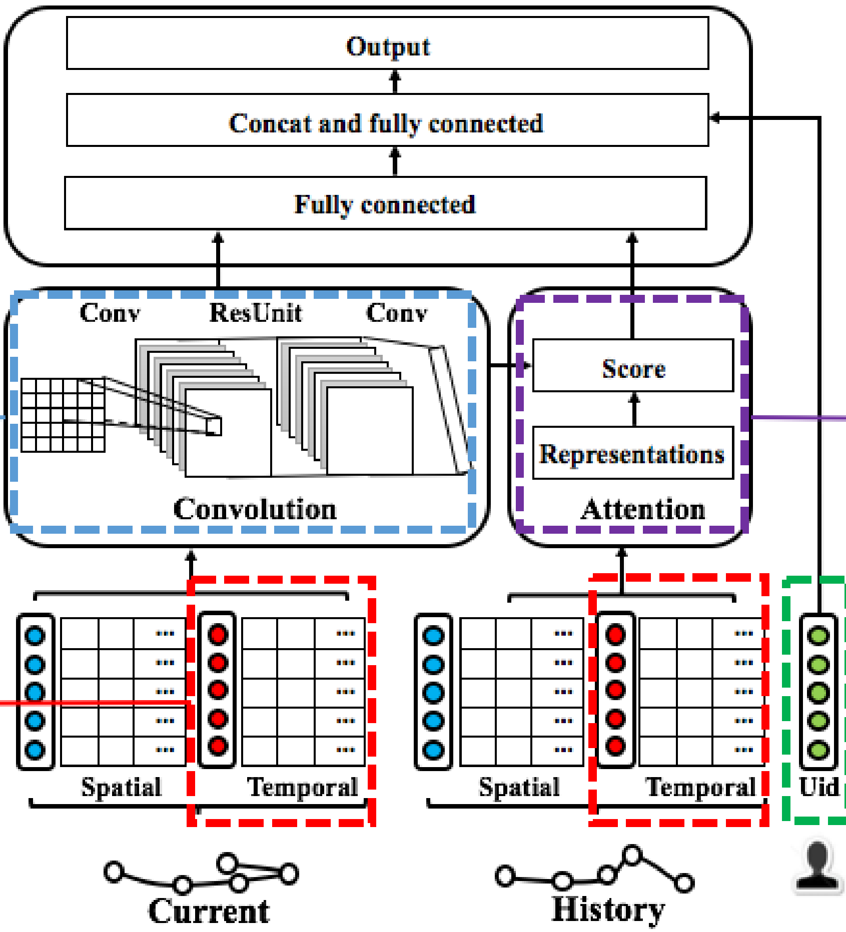
The time embedding captures time-dependent relations.



The historical attention module learns from historical data to 1) alleviate data sparsity; 2) evolving long-term preference

3. Solution

ACN



The convolution module captures high-order sequential pattern.

The historical attention module learns from historical data to 1) alleviate data sparsity; 2) evolving long-term preference

The time embedding captures time-dependent relations.

Uid embedding captures user personal preference

4. Experiment Results

- Dataset:

Table 1: The description and statistics of three datasets. $|\mathcal{U}|$: number of users; $|\mathcal{P}|$: number of locations; $|\mathcal{R}|$: average length of trajectory sequence; $|\mathcal{S}|$: sparsity.

Datasets	$\|\mathcal{U}\|$	$\|\mathcal{P}\|$	$\|\mathcal{R}\|$	$\|\mathcal{S}\|$
Gowalla	1989	40121	134	0.9966
Foursquare-TKY	2293	24321	229	0.9906
Foursquare-NYK	1083	15624	183	0.9883

- Evaluation metric:

$$Acc@K = \frac{|\{s \in S : l^*(s) \in L_K(s)))\}|}{|S|}$$

$$\text{macro-F1} = \frac{2 \times \text{macro-P} \times \text{macro-R}}{\text{macro-P} + \text{macro-R}}$$

4. Experiment Results

- Baselines:
  - Traditional:
    - MC: widely used mobility model working with state transition matrix
    - MF: factorizes users-locations matrix to generate user general preferences
    - FPMC: subsumes both MC and MF for mobility prediction.
  - RNN-based:
    - RNN: a basic deep neural network for sequential modeling
    - ST-RNN: extends RNN to model continuous spatio-temporal contexts
    - Deepmove: an enhanced version of RNN with history attention mechanism

- Experiment design:
  - Question1: what is the performance of our model as compared to other state-of-art methods?
  - Question2: what is the effect of the key hyper-parameters, such as length of trajectory and embedding size?
  - Question3: what is the influence of each of ACN’ s components?

4. Experiment Results

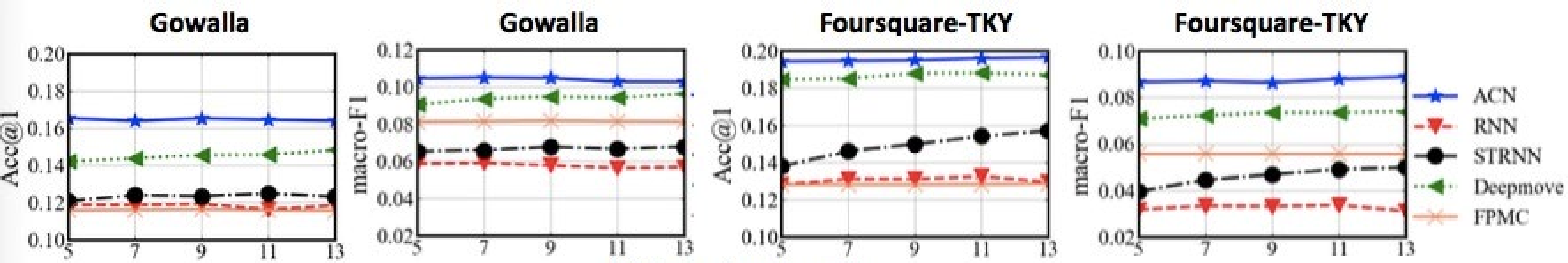
- Question1:

Table 2: Performance comparison on three public GTSM datasets.

Dataset	Metric	Traditional			RNN-based			Ours	Improv.
		MC	MF	FPMC	RNN	ST-RNN	Deepmove	ACN	
Gowalla	Acc@1	0.1151	0.0555	0.1163	0.1191	0.1249	0.1480	**0.1668**	12.70%
	Acc@5	0.2381	0.1227	0.2377	0.2596	0.2848	0.3097	**0.3247**	4.84%
	Acc@10	0.2701	0.1446	0.2707	0.3112	0.3464	0.3759	**0.3854**	2.53%
	macro-F1	0.0806	0.0223	0.0819	0.0601	0.0666	0.0964	**0.1047**	8.61%
Foursquare-TKY	Acc@1	0.1281	0.1299	0.1281	0.1325	0.1572	0.1881	**0.1966**	4.52%
	Acc@5	0.2758	0.2460	0.2761	0.3059	0.3435	0.3906	**0.4002**	2.46%
	Acc@10	0.3345	0.2793	0.3369	0.3724	0.4102	0.4624	**0.4698**	2.03%
	macro-F1	0.0555	0.0360	0.0560	0.0337	0.0499	0.0735	**0.0888**	14.40%
Foursquare-NYK	Acc@1	0.1242	0.1225	0.1265	0.1570	0.1634	0.1907	**0.2173**	13.95%
	Acc@5	0.2594	0.2292	0.2604	0.3489	0.3551	0.3926	**0.4131**	5.22%
	Acc@10	0.3024	0.2624	0.3027	0.4192	0.4251	0.4731	**0.4855**	3.49%
	macro-F1	0.0646	0.0677	0.0648	0.0814	0.0841	0.1140	**0.1302**	14.21%

- Traditional < RNN-based < CNN

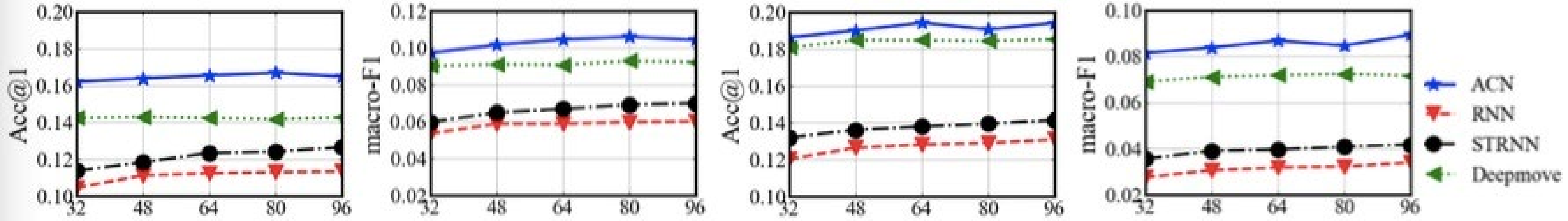
- Question2:



(a) Trajectory Length

  - Our model outperforms all other baselines on all lengths of trajectory.
  - The metrics Acc@1 and macro-F1 increase when the length of trajectory increases, however decrease when the trajectory becomes longer. This can be explained by the reason that for extremely sparse dataset, a longer trajectory tends to introduce extra information and more noises.

- Question2:



(b) Embedding Size

  - Our proposed model consistently outperforms all other baselines on all embedding sizes.
  - A larger embedding size does not necessarily lead to better performance because of overfitting issue. A model achieves its best performance when dimension size is properly chosen.

- Question3:
  - For x ∈ {**no, a, r , ar** }, ACN-x denotes ACN with component x enabled where **a** denotes attention mechanism and **r** denotes residual network.

Table 3: Acc@1 and macro-F1 vs. ACN components

Component	Gowalla		Foursquare-TKY	
	Acc@1	macro-F1	Acc@1	macro-F1
ACN-no	0.1603	0.0970	0.1903	0.0840
ACN-r	0.1641	0.1003	0.1924	0.0864
ACN-a	0.1650	0.1037	0.1928	0.0883
ACN-ar	0.1668	0.1047	0.1966	0.0888

  - ACN-ar achieves the best performance by jointly using all parts of ACN.

- Question3:
  - For x ∈ {**no, d, s, ds** }, ACN-x denotes ACN with component x enabled where **d** denotes hybrid dilated convolution and **s** denotes separable and shared convolution.

Table 3: Acc@1 and macro-F1 vs. ACN components

Component	Gowalla		Foursquare-TKY	
	Acc@1	macro-F1	Acc@1	macro-F1
ACN-no	0.1563	0.0944	0.1914	0.0866
ACN-d	0.1563	0.0946	0.1931	0.0876
ACN-s	0.1568	0.0947	0.1936	0.0875
ACN-ds	0.1668	0.1047	0.1966	0.0888

  - ACN-ds achieves the best performance by jointly using all parts of ACN.

1. Introduction
2. Related Work
3. Solution
4. Experiment Result
5. Conclusion

5. Conclusion

- We are firstly to propose a novel attentive convolutional network on sparse GTSM data
  - Regard the embedded trajectory as an image, using convolution filters to search for sequential patterns as local features of the image.
  - Design HSC which is combined of Hybrid dilated convolutions and Separable Convolutions to model high-order sequential patterns.
  - Use an attention mechanism to learn long-term preferences of users from history trajectory.
- Interesting future directions
  - Consider external feature like Point of interest and tweets to conduct semantic mobility prediction.

Thanks！