Toward Automated Pattern Discovery: Deep Representation Learning with Spatial-Temporal-Networked Data

—Collective, Dynamic, and Structured Analysis

Yanjie Fu

MISSOURI S&T

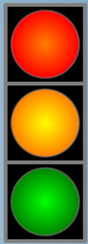
Outline

- **Background and Motivation**
- Collective Representation Learning
- Dynamic Representation Learning
- Structured Representation Learning
- Conclusions and Future Work

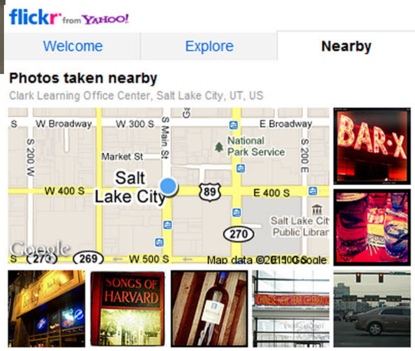
Human-Social-Technologic Systems

Physical World

IoT, GPS, wireless sensors, mobile Apps



Cyber World



airbnb

foursquare
CHECK-IN HERE

Human Activities in Human-Social-Technologic Systems

- **Spatial, Temporal, and Networked (STN) data can be**
  - Spatial: Point-of-Interests, blocks, zones, regions
  - Spatiotemporal: Taxi trajectories, bus trips, bike traces
  - Spatiotemporal-networked: Geo-tagged twitter posts, power grid netload
- **from a variety of sources**
  - Devices: phones, WIFIs, network stations, RFID
  - Vehicles: bikes, taxicabs, buses, subways, light-rails
  - Location based services: geo-tweets (Facebook, Twitter), geo-tagged photos (Flickr), check-ins (Foursquare, Yelp)

Taxicab GPS Traces

Bus Traces

Phone Traces

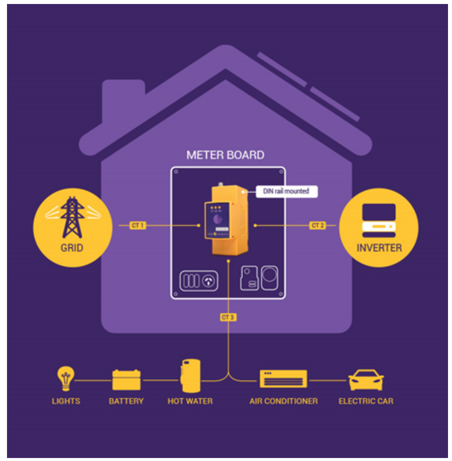
Mobile Check-ins

Represent the spatial, temporal, social, and semantic contexts of dynamic human/systems behaviors within and across regions

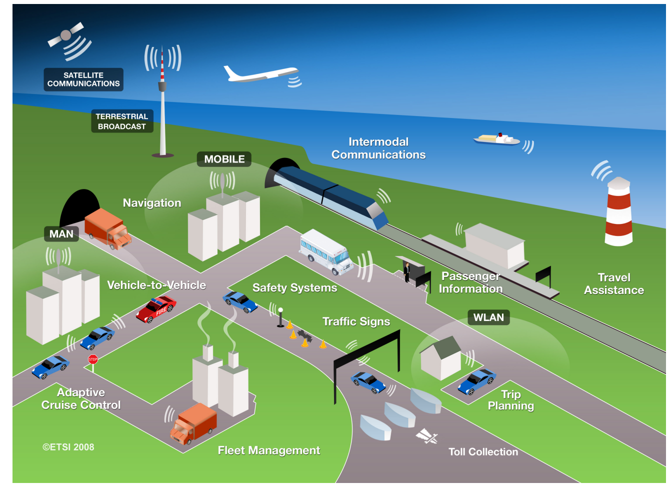
Important Applications

MISSOURI S&T

User Profiling & Recommendation Systems



Solar Analytics for Energy Saving



Intelligent Transportation Systems

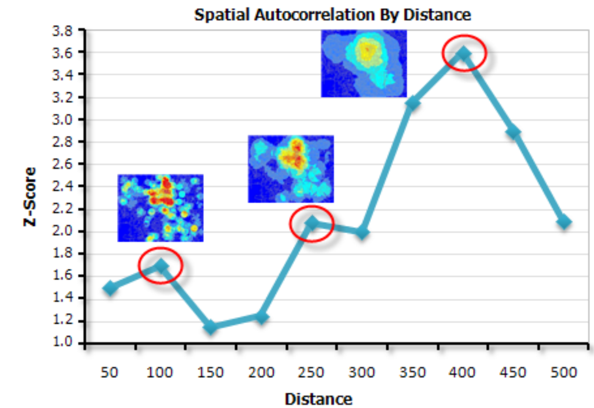
Personalized and Intelligent Education

Smart Heath Care

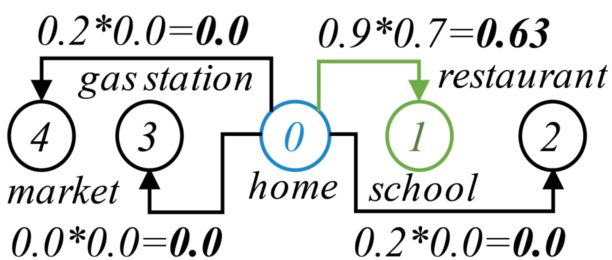
City Governance and Emergency Management

Unprecedented and Unique Complexity

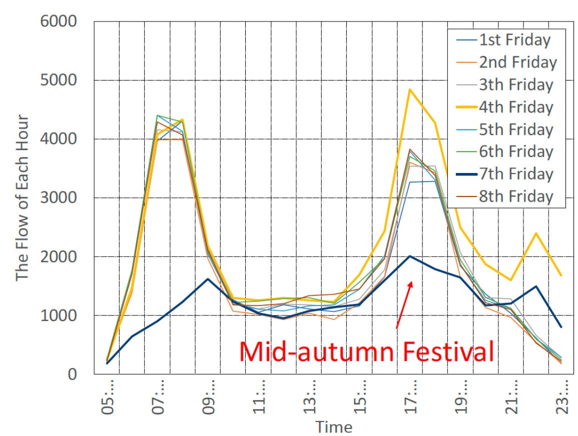
- **Spatiotemporallly non-i.i.d.**
  - Spatial autocorrelation
  - Spatial heterogeneity
  - Sequential asymmetric patterns
  - Temporal periodicity and dependency



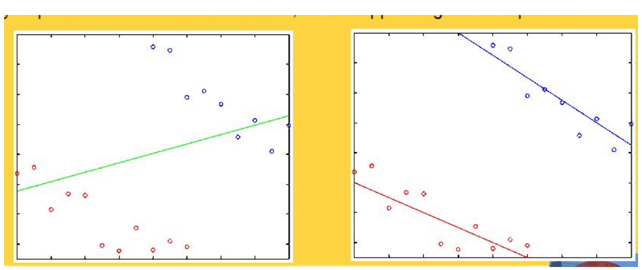
Spatial autocorrelations



Sequential asymmetric transitions



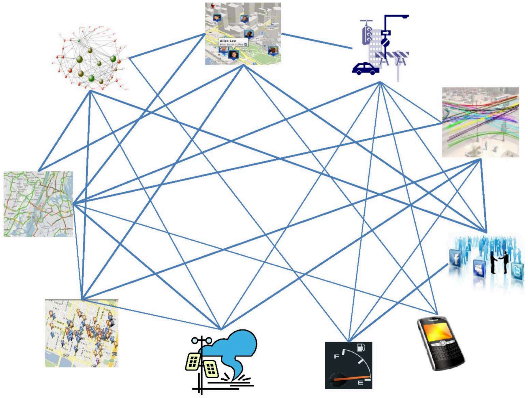
Temporal periodical patterns



Spatial heterogeneity

Unprecedented and Unique Complexity

- **Networked over time**
  - Collectively-related
- **Heterogeneous**
  - Multi-source
  - Multi-view
  - Multi-modality
- **Semantically-rich**
  - Trajectory semantics
  - User semantics
  - Event semantics
  - Region semantics

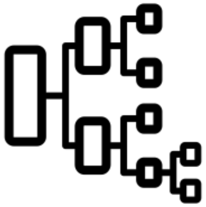


Technical Pains in Pattern Discovery (1)

Classic Machine learning

Input

Pattern/Feature extraction



Classification / Clustering

Car
Not car

Output

- **Feature identification and quantification**
  - Traditional method: Find domain experts to hand-craft features
  - **Can we automate feature/pattern extraction?**

Technical Pains in Pattern Discovery (2)

Classic Machine learning

Input

Pattern/Feature extraction

Classification / Clustering

Car
Not car

Output

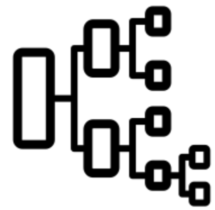
- **Multi-source unbalanced data fusion**
  - Traditional method: Extract features, weigh features, weighted combination
  - **Can we automatically extract features from multi-source unbalanced data?**

Technical Pains in Pattern Discovery (3)

Classic Machine learning

Input

Pattern/Feature extraction



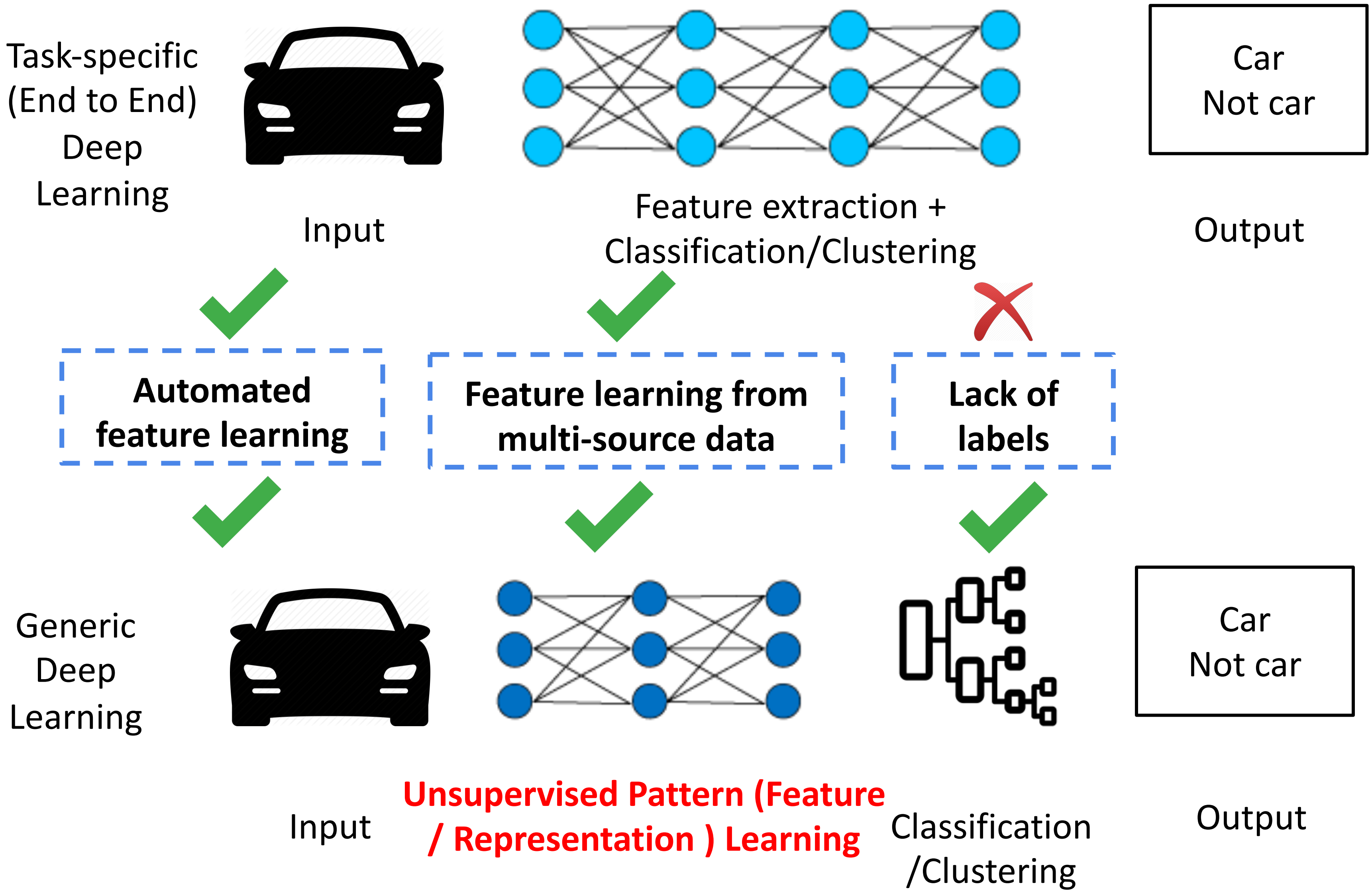
Classification / Clustering

Car
Not car

Output

- **Field data/real-world systems are usually lack of benchmark labels (i.e., y, responses, targets)**
  - Example: Netload in power grids: behind-the-meter gas-generated electricity and solar-generated electricity are unknown
  - **Can we learn features without labels (unsupervised)?**

Deep Learning Can Help

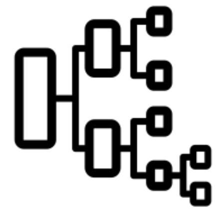


Technical Pains in Pattern Discovery (4)

Classic Machine learning

Input

Pattern/Feature extraction



Classification / Clustering

Car
Not car

Output

- **Classic algorithms are not directly available in spatiotemporal networked data**
  - Traditional method: revised classic algorithms + spatiotemporal networked data regularities
    - Regression + spatial properties = spatial autoregression method
    - Clustering + spatial properties = spatial co-location method
  - **Can we learn features while maintaining the regularities of spatiotemporal networked data?**

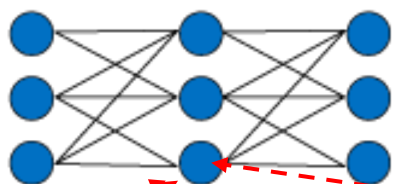
Data Regularity-aware Unsupervised Representation Learning

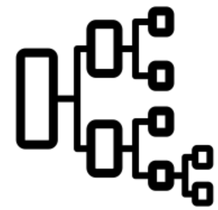
Human and system behaviors have spatiotemporally socially regularities

Regularities of spatiotemporal networked data

Data Regularity-aware representation learning

Generic Deep Learning





Car
Not car

Automated feature learning

Feature learning from multi-source data

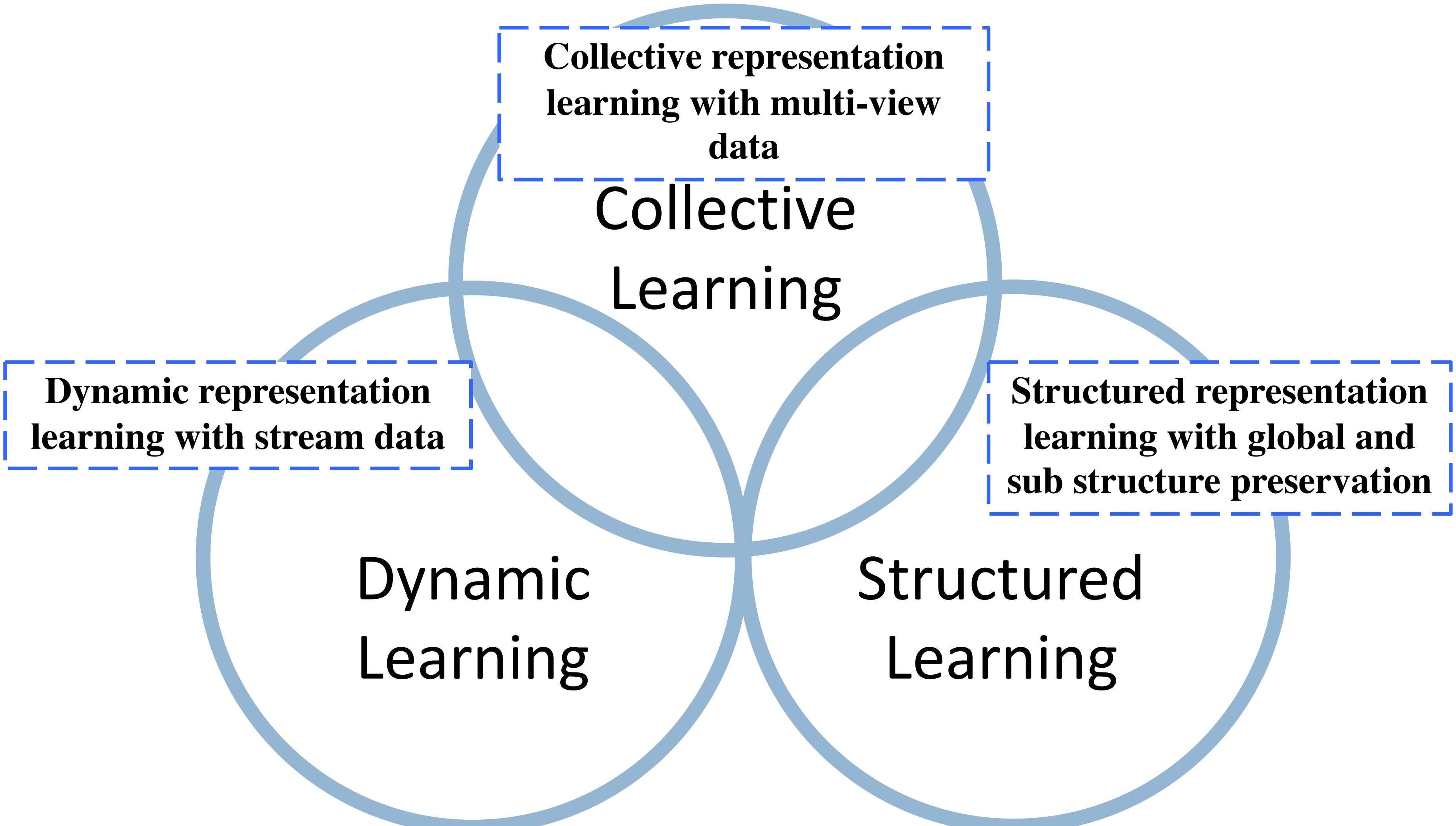
- Lack of labels (unsupervised)
- Multi-source multi-view multi-modality
- Spatial autocorrelation (peer)
- Spatial heterogeneity (clustering)
- Temporal dependencies (current-past)
- Periodical patterns
- Sequential asymmetric transition
- Spatial hierarchy (hierarchical clustering)
- Hidden semantics
- Spatial locality
- Global and sub structural patterns in behavioral graphs

Data regularities

Lack of labels

The Overview of The Talk

Automated Feature Learning from Spatial-Temporal-Networked Data



Outline

- Background and Motivation
- **Deep Collective Representation Learning**
- Deep Dynamic Representation Learning
- Deep Structured Representation Learning
- Conclusion and Future Work

The Rising of Vibrant Communities

- **Consumer City Theory, Edward L. Glaeser (2001), Harvard University.**
  - More by Nathan Schiff (2014), University of British Columbia. Victor Coutour (2014), UC Berkeley. Yan Song (2014), UNC Chapel Hill.
  - **Spatial Characters**: walkable, dense, compact, diverse, accessible, connected, mixed-use, etc.
  - **Socio-economic Characters:** willingness to pay, intensive social interactions, attract talented workers and cutting-edge firms, etc.

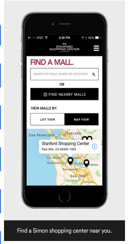
Supported by NSF CISE pre-Career award (III-1755946)

What are the underlying driving forces of a vibrant community?

Measuring Community Vibrancy

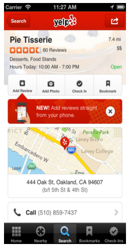
- Mobile checkin data

Urban vibrancy is reflected by the **frequency** and **diversity** of user activities.

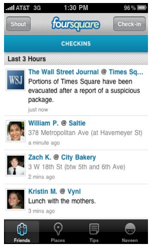


Shopping

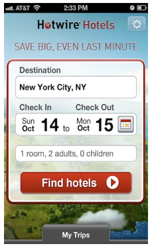
Transport



Dinning



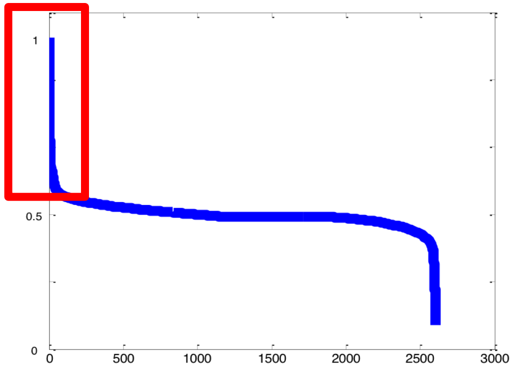
Travel



Lodging

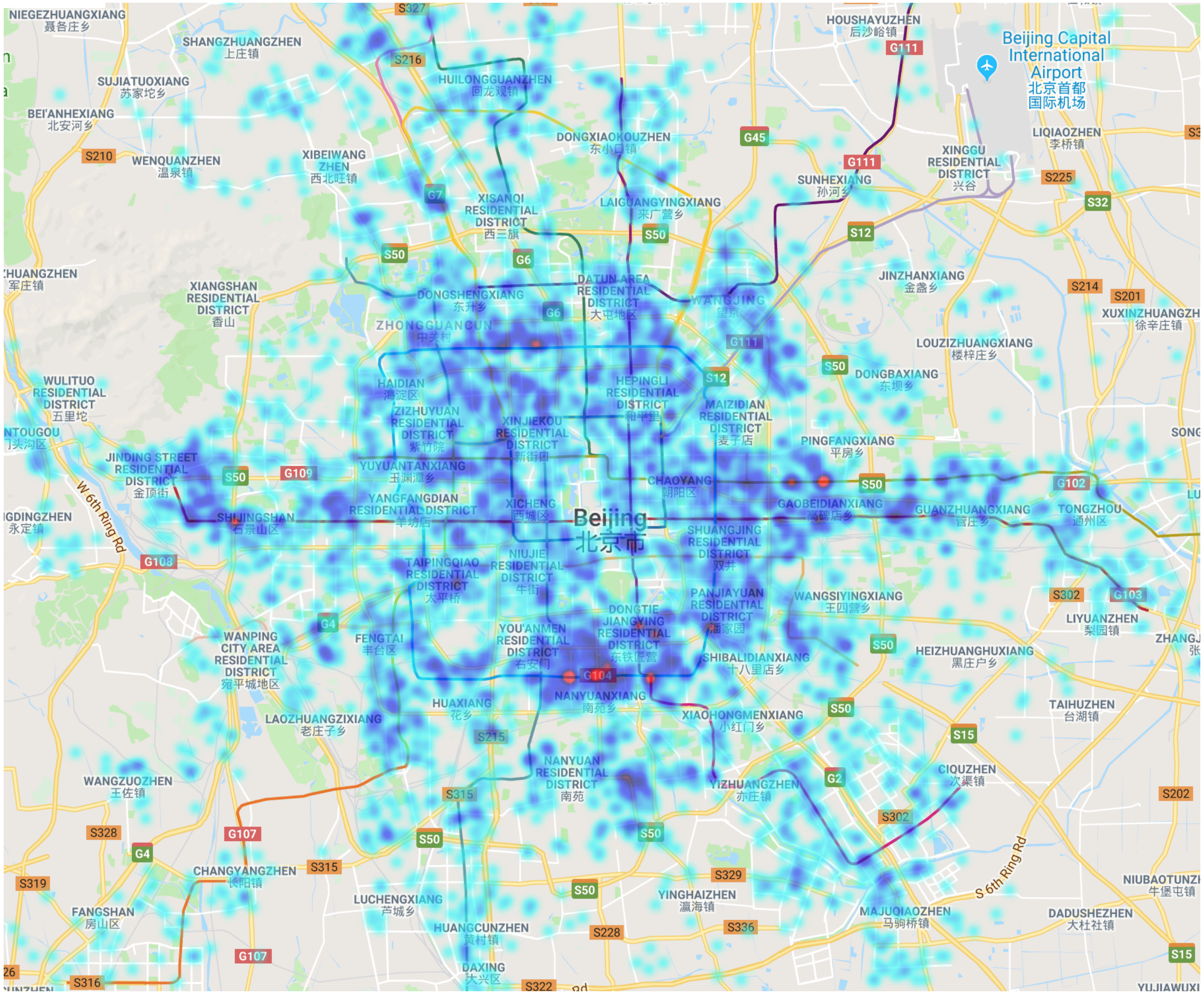
- Frequency and diversity of mobile checkins
  - Frequency: $\text{fre} = \#(checkin)$
  - Diversity: $\text{div} = -\sum_{type} \frac{\#(checkin,type)}{\#(checkin)} \log \frac{\#(checkin,type)}{\#(checkin)}$, where ***type*** denotes the activity type of mobile users
- Fused scoring
  - $Vibrancy = (1+\beta^2)\frac{fre*div}{(\beta^2*fre+div)}$
  - β controls the weights of fre and div
  - Power-law distributed
  - Some are highly vibrant while most are somewhat vibrant

Vibrancy Score



Community rankings

Spatial Unbalance of Urban Community Vibrancy

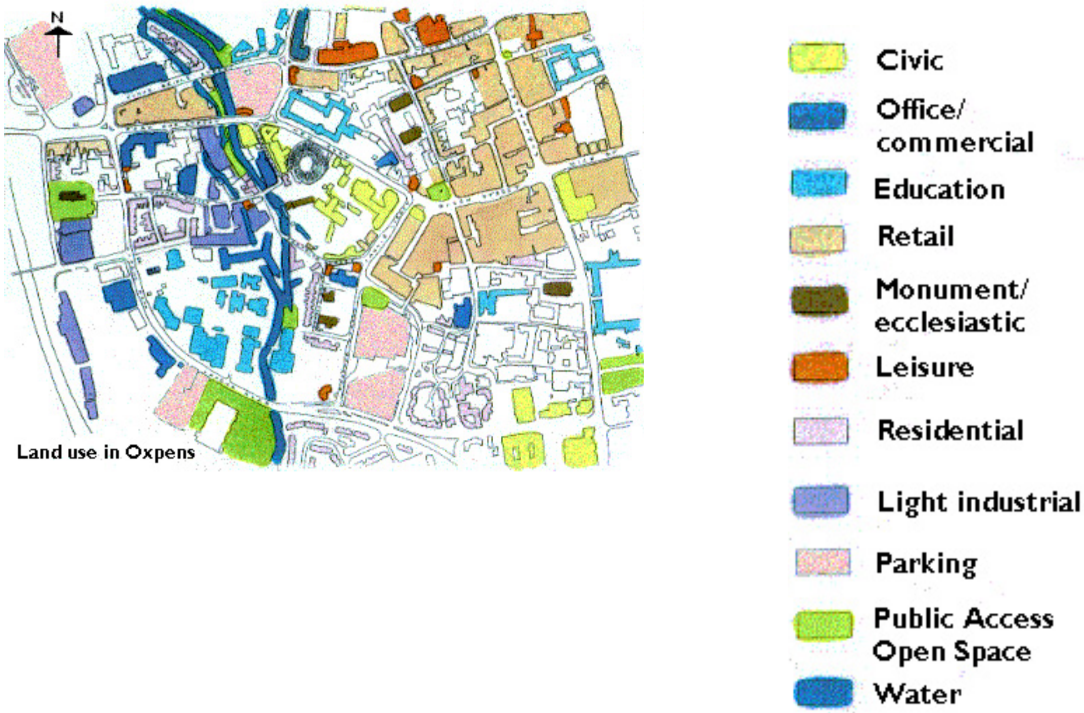


Motivation Application: How to Quantify Spatial Configurations and Social Interactions

Static Element

Dynamic Element

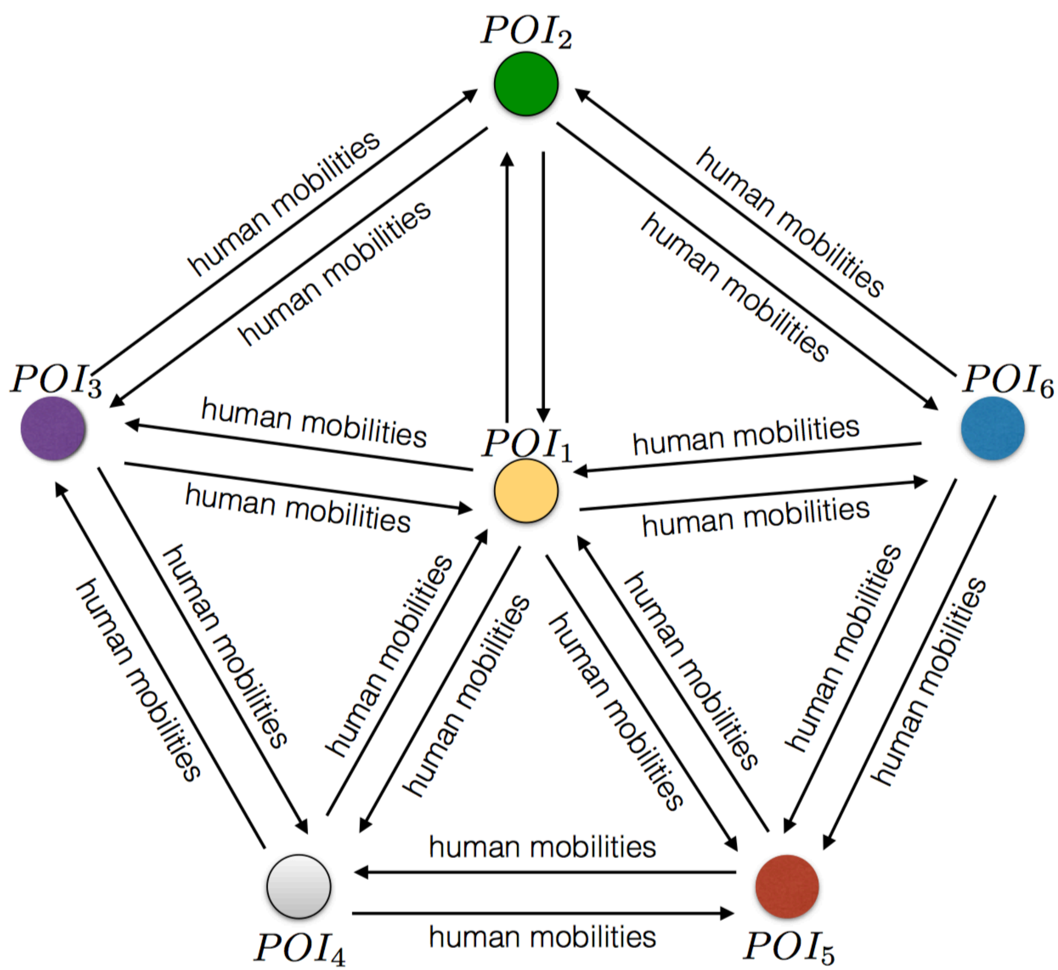
Urban Community =Spatial Configuration + Social Interactions



From Regions to Graphs

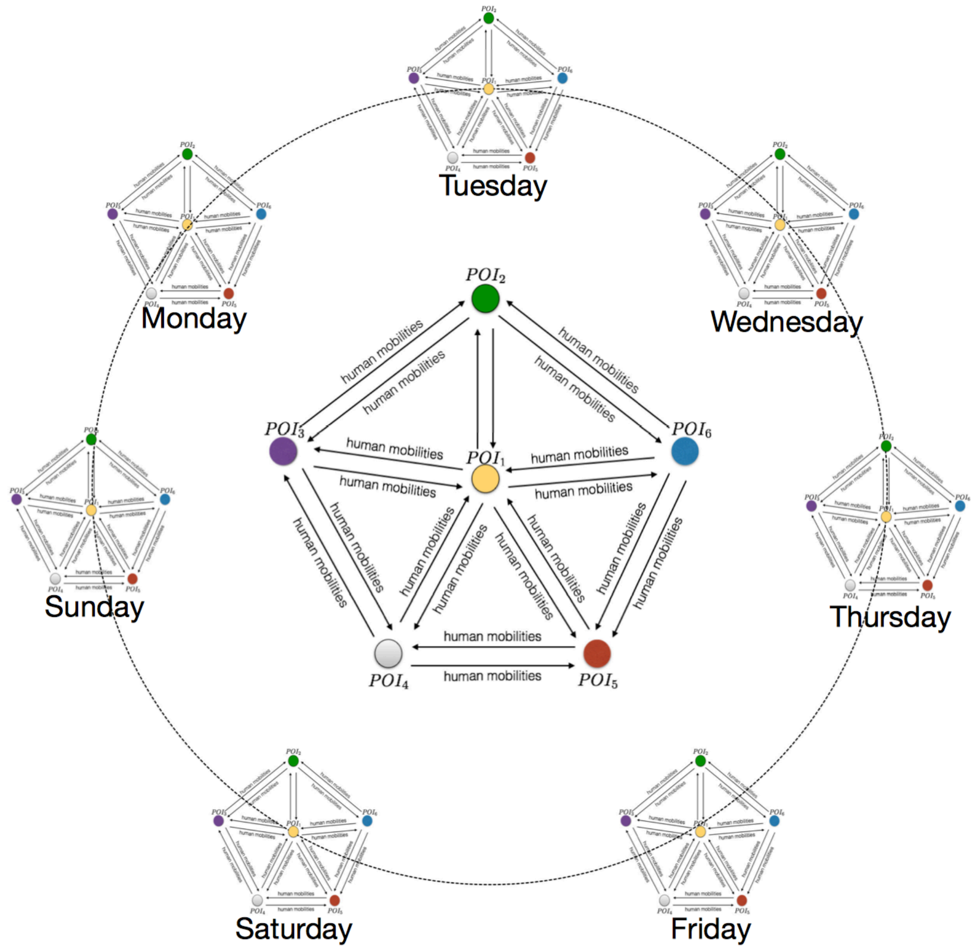
Spatial Regions as Human Mobility Graphs

- POIs → nodes
- Human mobility connectivity betwee two POIs → edge weights
- Edge weights are asymmetric



Periodicity of Human Mobility

- Different days-hours → different periodic mobility patterns → different graph structures

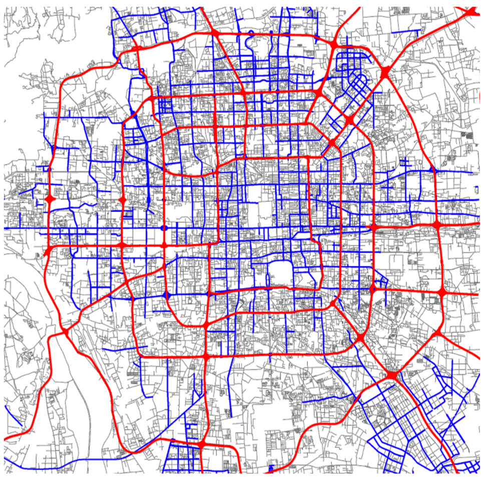
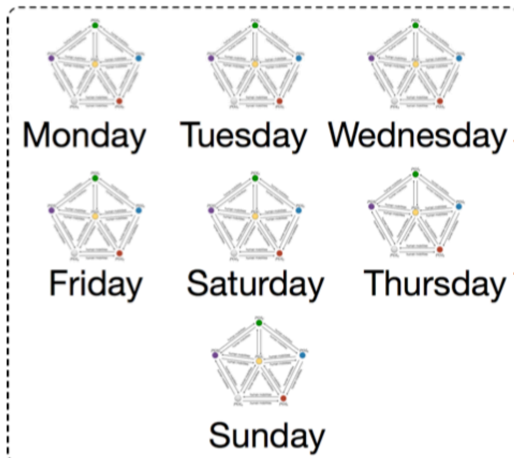


Collective Representation Learning with Multi-view Graphs

Spatial Objects (e.g., Regions)

Multiple Graphs

Feature Vector Representations

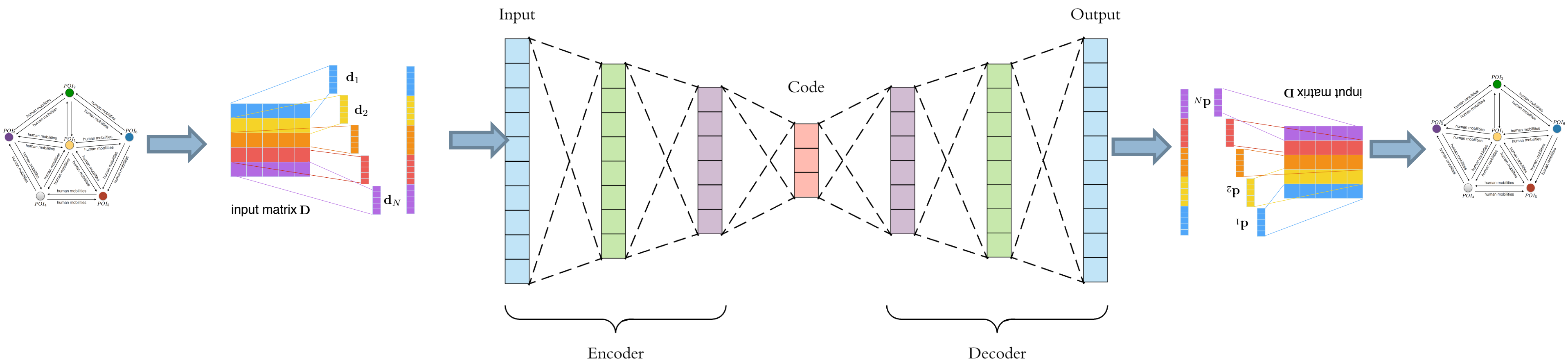
f( , 

) = 

Constraint: the multi-view graphs are collaboratively related

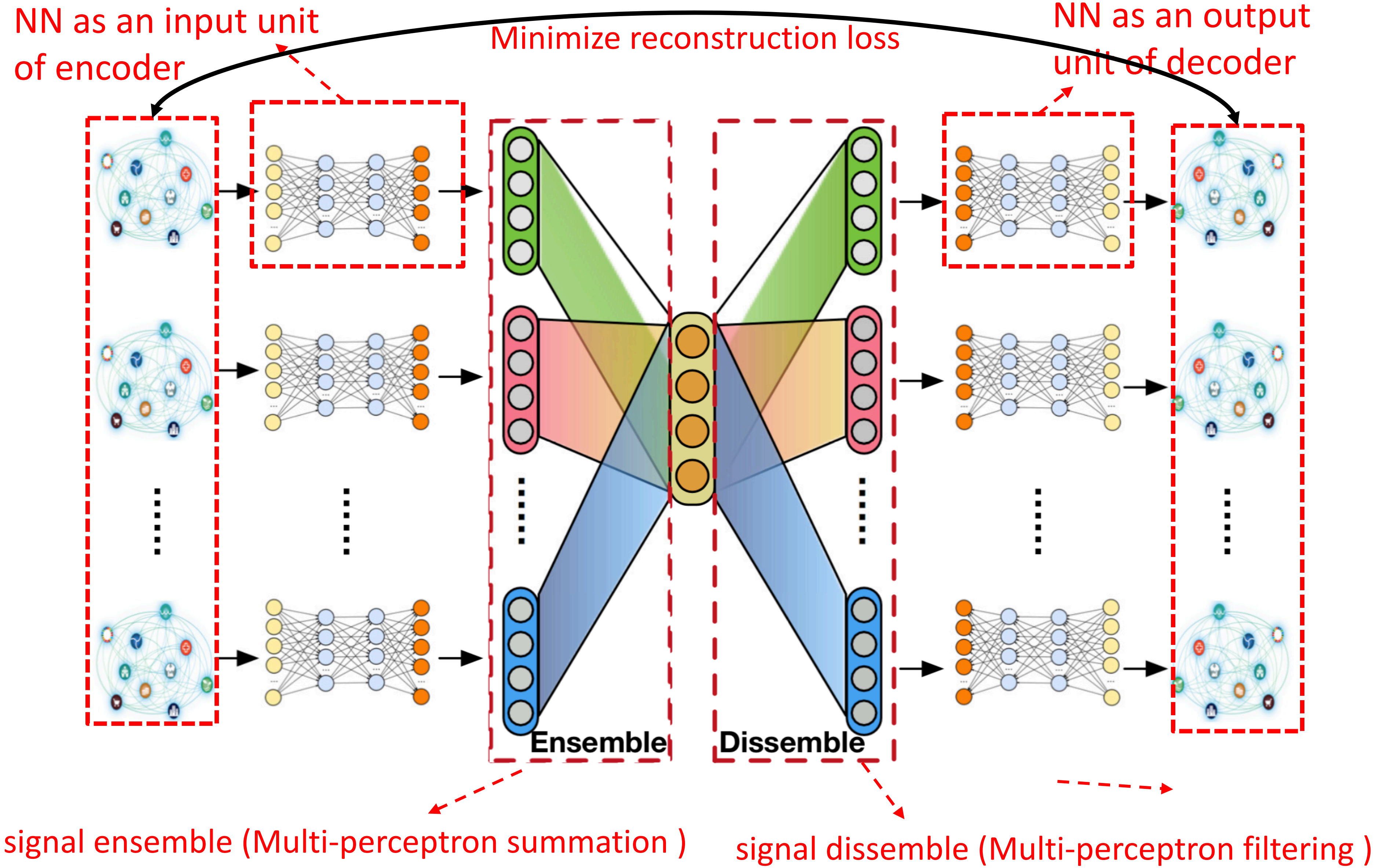
Solving Single-Graph Input

- The encoding-decoding representation learning paradigm
  - Encoder: compress a graph into a latent feature vector
  - Decoder: reconstruct the graph based on the latent feature vector
  - Objective: minimizing the difference between original and reconstructed graphs



- Unsupervised (label-free): doesn't require labels
- Generic: not specific for single application
- Intuitive: a good representation can be used to reconstruct original signals

Solving Multi-graph Inputs: An Ensemble-Encoding Dissemble-Decoding Method



Solving the Optimization Problem

1. Multi-graph Ensemble Encoding

$$\begin{cases} \mathbf{y}_{i,t}^{(k),1} & = \sigma(\mathbf{W}_{i,t}^{(k),1}\mathbf{p}_{i,t}^{(k)} + \mathbf{b}_{i,t}^{(k),1}), \forall t \in \{1,2,\cdots,7\}, \\ \mathbf{y}_{i,t}^{(k),r} & = \sigma(\mathbf{W}_{i,t}^{(k),r}\mathbf{p}_{i,t}^{(k)} + \mathbf{b}_{i,t}^{(k),r}), \forall r \in \{2,3,\cdots,o\}, \\ \mathbf{y}_{i}^{(k),o+1} & = \sigma(\sum_t \mathbf{W}_{t}^{(k),o+1}\mathbf{y}_{i,t}^{(k),o} + \mathbf{b}_{t}^{(k),o+1}), \\ \mathbf{z}_{i}^{(k)} & = \sigma(\mathbf{W}^{(k),o+2}\mathbf{y}_{i}^{(k),o+1} + \mathbf{b}^{(k),o+2}), \end{cases}$$

Ensemble multi-graphs

2. Multi-graph Dissemble Decoding

$$\begin{cases} \hat{\mathbf{y}}_{i}^{(k),o+1} & = \sigma(\hat{\mathbf{W}}^{(k),o+2}\mathbf{z}_{i}^{(k)} + \hat{\mathbf{b}}^{(k),o+2}), \\ \hat{\mathbf{y}}_{i,t}^{(k),o} & = \sigma(\hat{\mathbf{W}}_{t}^{(k),o+1}\hat{\mathbf{y}}_{i}^{(k),o+1} + \hat{\mathbf{b}}_{t}^{(k),o+1}), \\ \hat{\mathbf{y}}_{i,t}^{(k),r-1} & = \sigma(\hat{\mathbf{W}}_{i,t}^{(k),r}\hat{\mathbf{y}}_{i,t}^{(k),r} + \hat{\mathbf{b}}_{i,t}^{(k),r}), \forall r \in \{2,3,\cdots,o\}, \\ \hat{\mathbf{p}}_{i,t}^{(k)} & = \sigma(\hat{\mathbf{W}}_{i,t}^{(k),1}\hat{\mathbf{y}}_{i,t}^{(k),1} + \hat{\mathbf{b}}_{i,t}^{(k),1}), \end{cases}$$

Dissemble multi-graphs

3. Objective Function

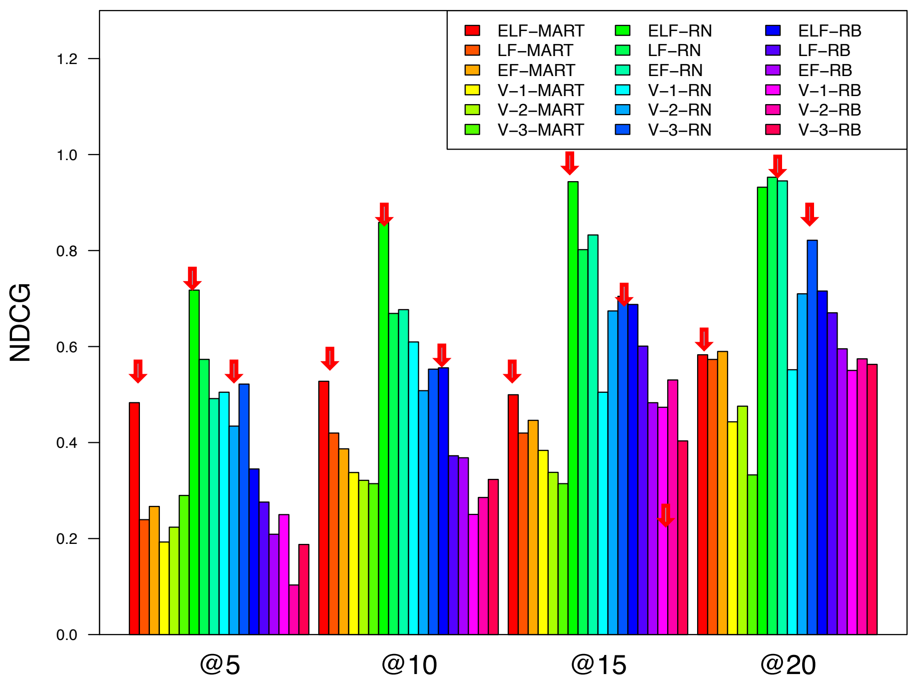
$$\mathcal{L}^{(k)} = \sum_{t \in \{1,2,\ldots,7\}} \sum_{i} \|(\mathbf{p}_{i,t}^{(k)} - \hat{\mathbf{p}}_{i,t}^{(k)}) \odot \mathbf{v}_{i,t}^{(k)}\|_2^2$$

Reconstruction loss

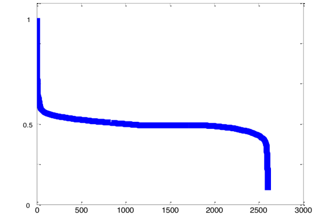
Sparsity regularization: If mobility connectivity = 0, weight=1 to penalize the loss

If mobility connectivity >0, weight>1

Comparisons with Features Generated By Different Methods

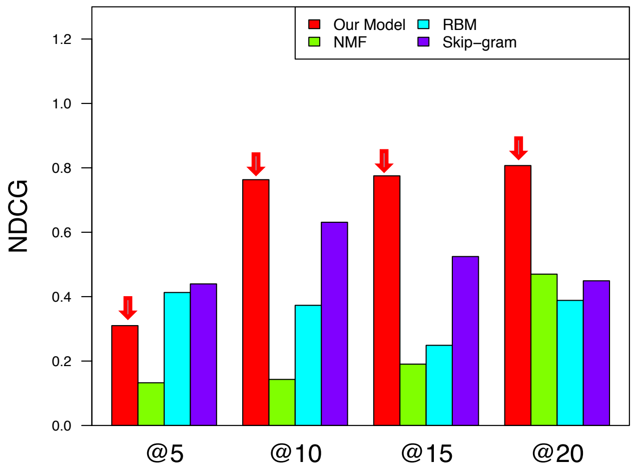


- **Data**
  - Beijing Checkin Data

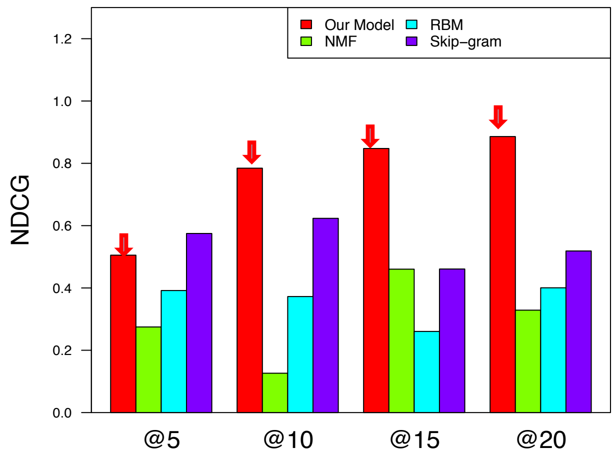


- **Ranking Models**
  - MART: it is a boosted tree ranking model
  - RankBoost (RB): it is a boosted pairwise ranking method, which trains multiple weak rankers and combines their outputs as final ranking.
  - RankNet (RN): it uses a neural network to model the underlying probabilistic cost function.
- **Feature Sets**
  - Explicit Features (EF)
  - Latent features (LF)
  - Explicit & Latent features (ELF)
  - Features generated by variation 1 of our method: distance graphs not mobility graphs
  - Features generated by variation 2 of our method: average not collective
  - Features generated by variation 3 of our method: non-weighted not unsupervised weighted.
- **Evaluation Criteria**
  - NDCG: Evaluate the ranking performance at Top N

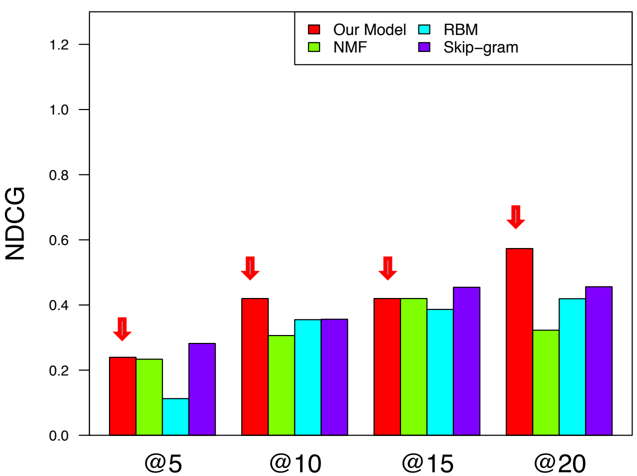
Comparison with Baseline Representation Learning Algorithms



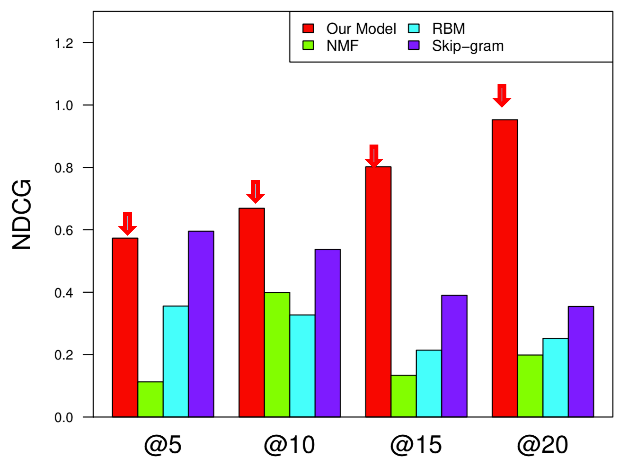
NDCG@N comparisons over LambdaMART



NDCG@N comparisons over ListNe



NDCG@N comparisons over MART



NDCG@N comparisons over RankBoost

- Ranking Models
  - LAMBDAMART
  - ListNet
  - MART
  - RankBoost
- Baseline Methods
  - RBM: restricted Boltzmann machine
  - NMF: non-negative matrix factorization
  - Skip-gram
- Evaluation Criteria
  - NDCG: Evaluate the ranking performance at Top N

Summary

- **Task**
  - Collective representation learning with multi-view graphs
- **Modeling**
  - Develop an ensemble-dissemble encoding-decoding approach
  - multi-graph ensemble encoding and multi-graph dissemble decoding
- **Application**
  - Quantifying urban communities for understanding urban vibrancy

Outline

- Background and Motivation
- Collective Representation Learning
- **Dynamic Representation Learning**
- Structured Representation Learning
- Conclusion and Future Work

Social Fairness in Insurance Sector

CNN Money Companies Markets Tech Media U.S.

Auto insurers charge (some) safe drivers higher rates

by Melanie Hicken @melhicken

January 28, 2013: 4:48 PM ET

Recommend 0

When setting rates, insurers often put more weight on income-related fact than factors like driving history, according to a consumer watchdog report

Consumers Union finds Auto Insurers Penalize

What can we do to defend social fairness on insurance rates?

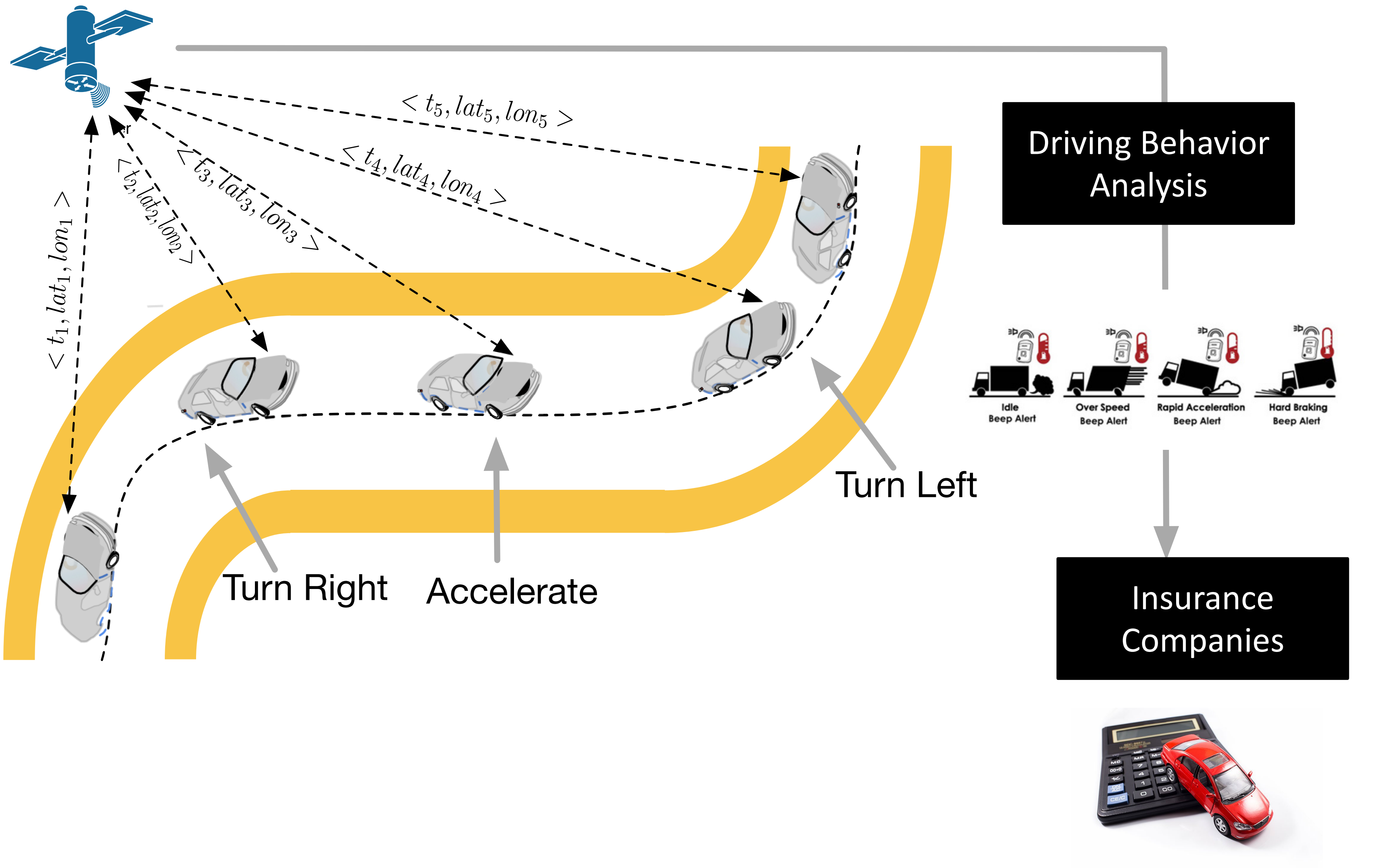
Consumers Union

Nonprofit Publisher of Consumer Reports

Washington, D.C. – Many good drivers pay higher insurance premiums because of their credit history and other factors that have nothing to do with their driving record, according to Consumers Union, the policy and advocacy division of Consumer Reports.

The consumer group urged regulators to ban the use of credit histories and some other non-driving factors for setting premiums at a National Association of Insurance Commissioners (NAIC) hearing on November 19th.

Motivation Application: Machine-Learning Based Driving Behavior Analysis



Defining Driving Operations & States

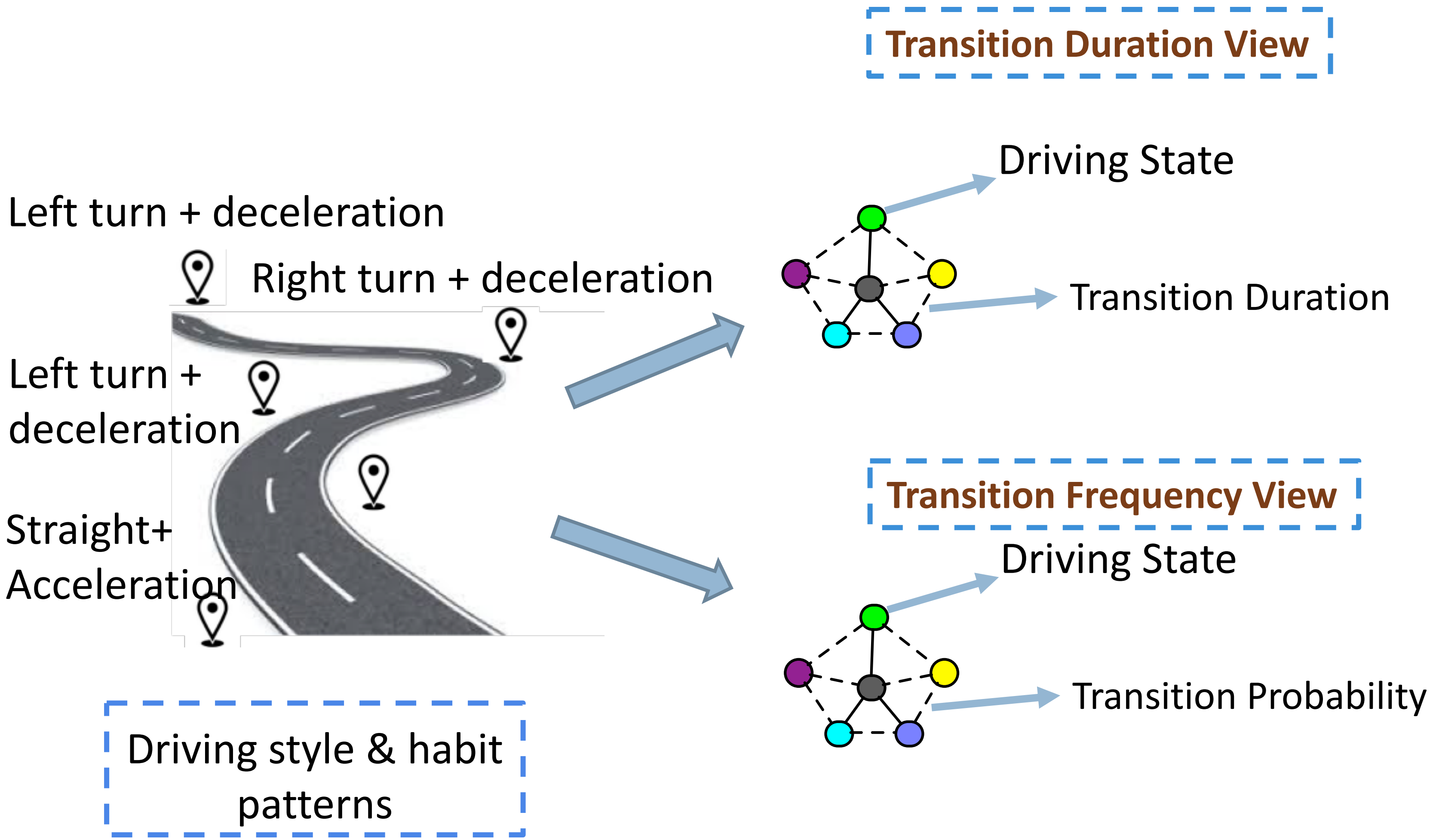
- **Driving Operations**
  - Speed-related:
    acceleration, deceleration, constant speed
  - Direction-related:
    Turning right, left, moving straight
- **Driving States**
  - Definition: speed operation + direction operation

Speed Operation
Acceleration
Deceleration
Constant Speed

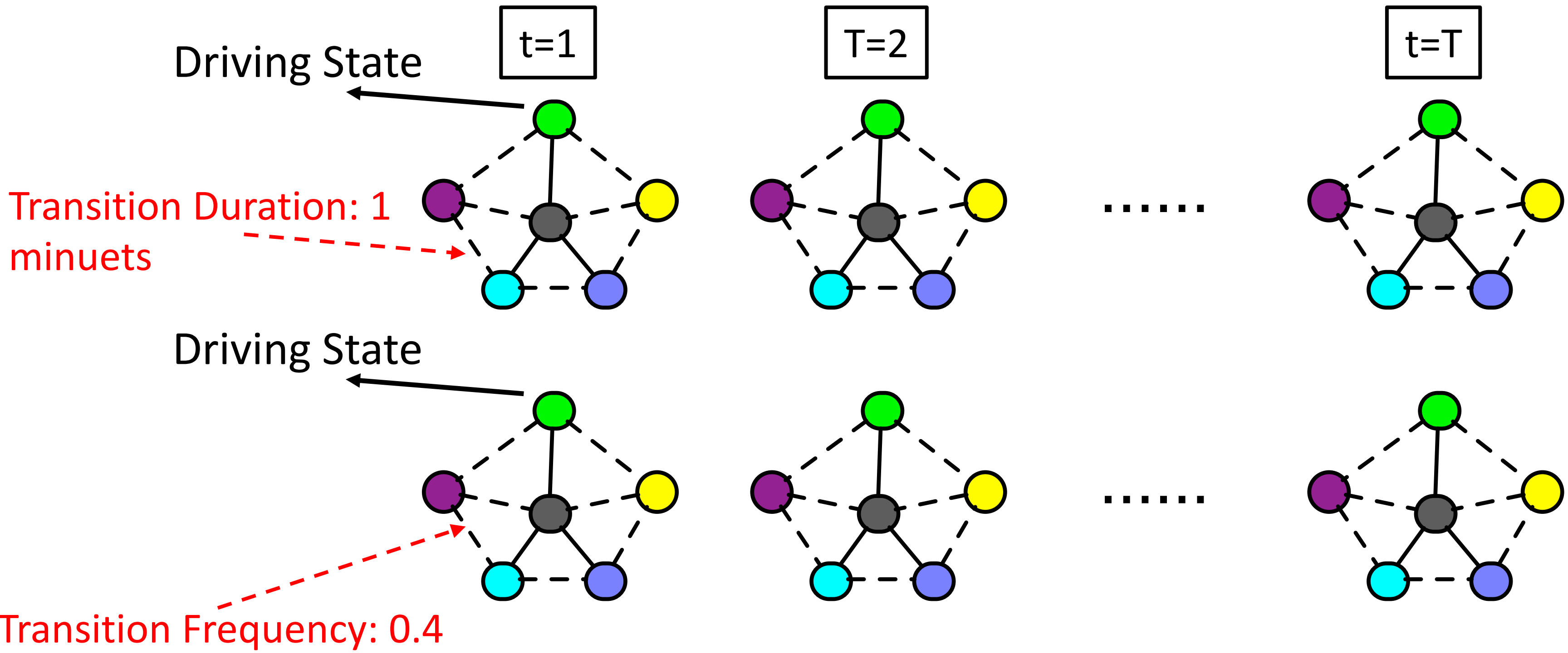
+

Direction Operation
Turning Right
Turning Left
Moving Straight

Quantifying Driving Habits with Driving State Transition Graphs

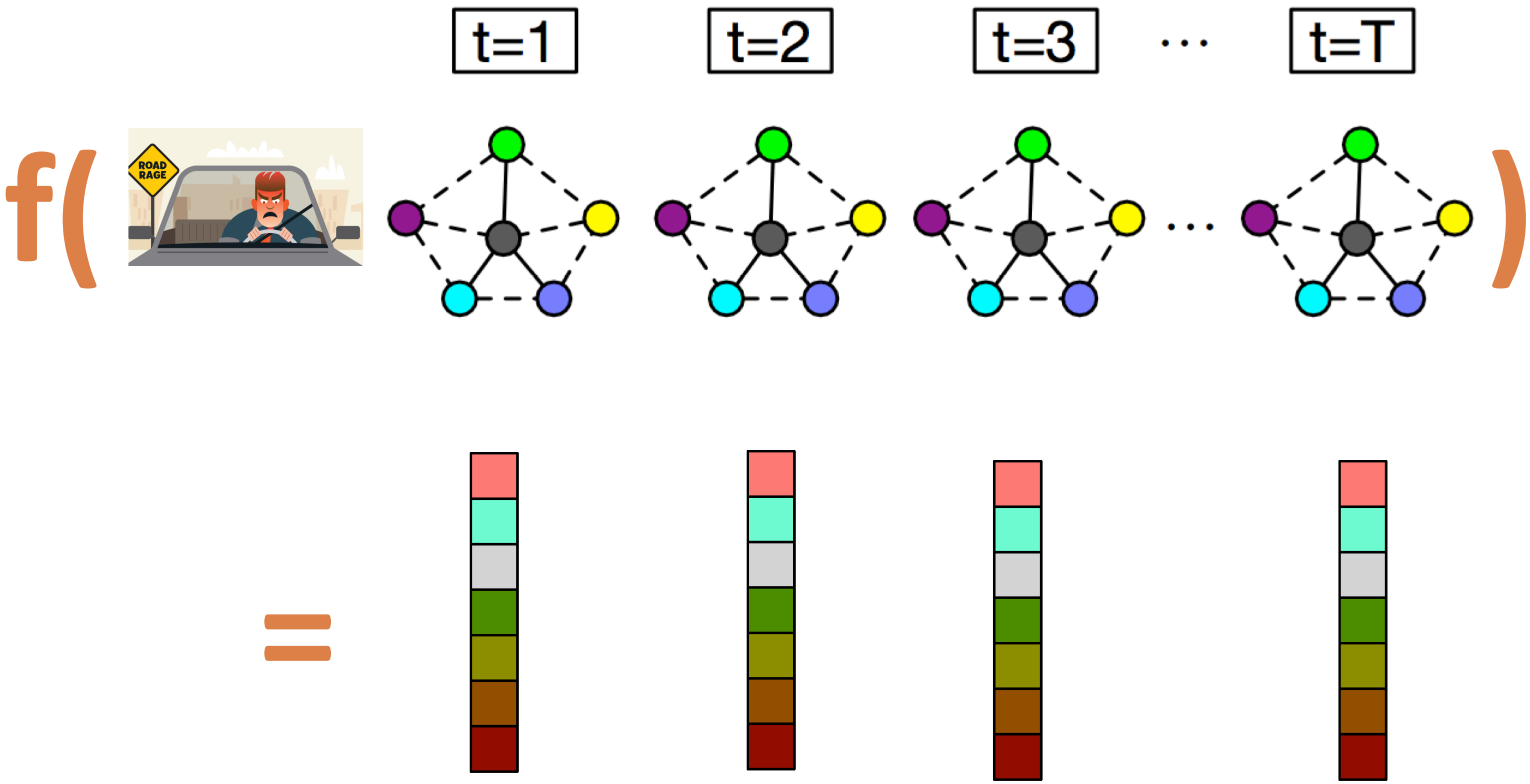


Driving State Transition Graph Sequence



- Transition frequency: how frequently a driver changes his/her driving state from one to another **(unusual high-frequency: drunk?)**
- Transition duration: how quickly a driver changes his/her driving state from one to another **(unusually fast: non-comfortable driving habits)**

Dynamic Representation Learning with Graph Stream



- Map a sequence of time-varying yet relational graphs to a sequence of time-varying yet relational vectors
- s. t. spatial and temporal dependencies

Three Modeling Constraints

- **Structural Reservation**
  - If two graphs' structures are similar, their feature vectors are similar
- **Temporal Dependency**
  - Current driving operations are related to previous driving operations
- **Peer Dependency**
  - Drivers with similar driving behaviors should share similar feature vectors

Modeling Structural Reservation

Structural Reservation: Minimizing reconstruction loss

Input vector

$$\begin{cases} \mathbf{y}_i^1 & = \sigma(\mathbf{W}^1 \mathbf{x}_i + \mathbf{b}^1), \\ \mathbf{y}_i^k & = \sigma(\mathbf{W}^k \mathbf{y}_i^{k-1} + \mathbf{b}^k), \forall k \in \{2, 3, \cdots, o\}, \\ \mathbf{z_i} & = \sigma(\mathbf{W}^{o+1} \mathbf{y}_i^o + \mathbf{b}^{o+1}). \end{cases}$$

Encoded vector

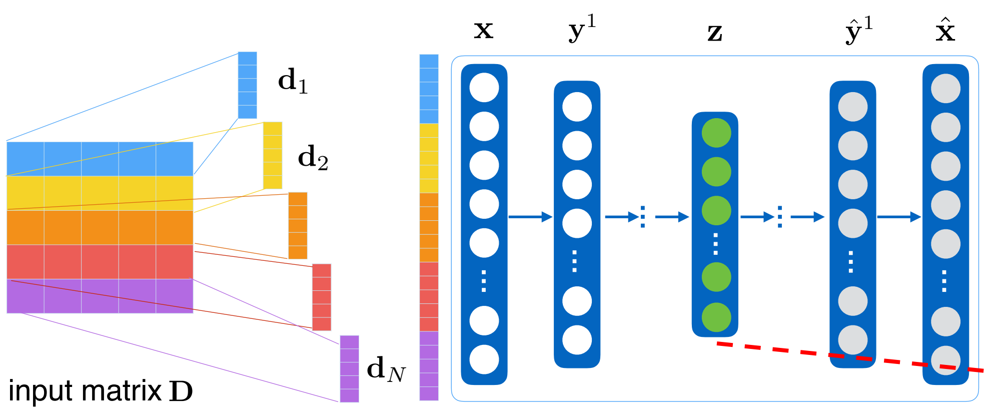
Embedding

Embedding

$$\begin{cases} \hat{\mathbf{y}}_i^o & = \sigma(\hat{\mathbf{W}}^{o+1} \mathbf{z}_i + \hat{\mathbf{b}}^{o+1}), \\ \hat{\mathbf{y}}_i^{k-1} & = \sigma(\hat{\mathbf{W}}^k \hat{\mathbf{y}}_i^k + \hat{\mathbf{b}}^k), \forall k \in \{2, 3, \cdots, o\}, \\ \hat{\mathbf{x}}_i & = \sigma(\hat{\mathbf{W}}^1 \hat{\mathbf{y}}_i^1 + \hat{\mathbf{b}}^1). \end{cases}$$

Input vector

Decoded vector

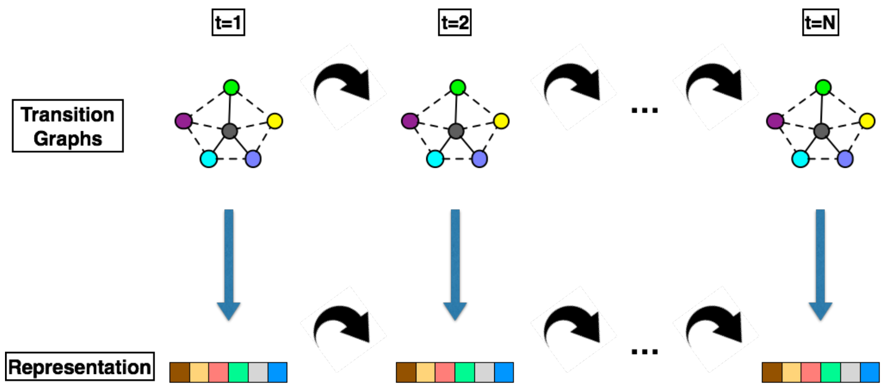


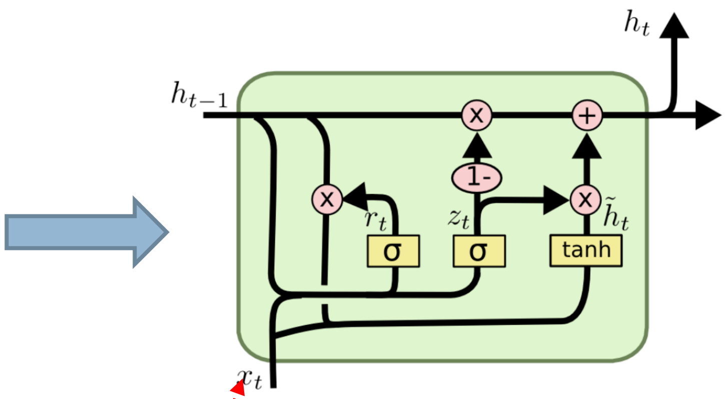
Learned Representation

The encoding phrase: encode input vector into embedding;
The decoding phrase: decode the embedding to recover input.

Modeling Temporal Dependency

- Temporal Dependency: Current driving operations are related to previous driving operations





Feed the output of Autoencoder's hidden layer into Gated Recurrent Unit

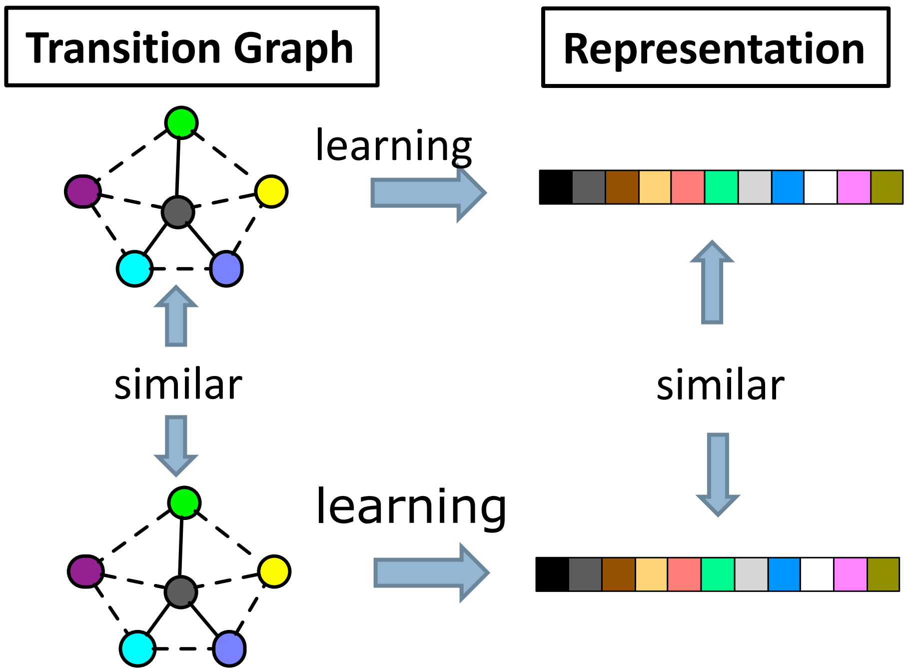
$$
\begin{cases}
 & \#Sequential \quad Encode \quad Step \\
(\mathbf{y}_i^1)^\tau & = \sigma(\mathbf{W}^1 \mathbf{x}_i^\tau + \mathbf{b}^1), \\
(\mathbf{y}_i^k)^\tau & = \sigma(\mathbf{W}^k (\mathbf{y}_i^{k-1})^\tau + \mathbf{b}^k), \forall k \in \{2, 3, \cdots, o\}, \\
\mathbf{z}_i^\tau & = (1 - c^\tau)\mathbf{z}_i^{\tau-1} + c^\tau \tilde{\mathbf{z}}_i^\tau .
\end{cases}
$$

Current hidden layer depends on previous hidden layer

$$
\begin{cases}
 & \#Sequential \quad Decode \quad Step \\
(\hat{\mathbf{y}}_i^o)^\tau & = \sigma(\hat{\mathbf{W}}^{o+1} \mathbf{z}_i^\tau + \hat{\mathbf{b}}^{o+1}), \\
(\hat{\mathbf{y}}_i^{k-1})^\tau & = \sigma(\hat{\mathbf{W}}^k (\hat{\mathbf{y}}_i^k)^\tau + \hat{\mathbf{b}}^k), \forall k \in \{2, 3, \cdots, o\}, \\
\hat{\mathbf{x}}_i^\tau & = \sigma(\hat{\mathbf{W}}^1 (\hat{\mathbf{y}}_i^1)^\tau + \hat{\mathbf{b}}^1).
\end{cases}
$$

Modeling Peer Dependency

- **Peer Dependency: Drivers with similar driving behaviors should share similar latent representations**



Graphical regularization: if a spatial item i and a spatial item j are similar at time T, the representation Zi and Zj are similar; punished otherwise.

$$\mathcal{H}_c(G^\tau) = \sum_{u_i \in \mathcal{U}} \sum_{u_j \in \mathcal{U}, u_i \neq u_j} s_{i,j}^{\tau} \cdot \left\| \mathbf{z}_i^{\tau} - \mathbf{z}_j^{\tau} \right\|_2^2$$

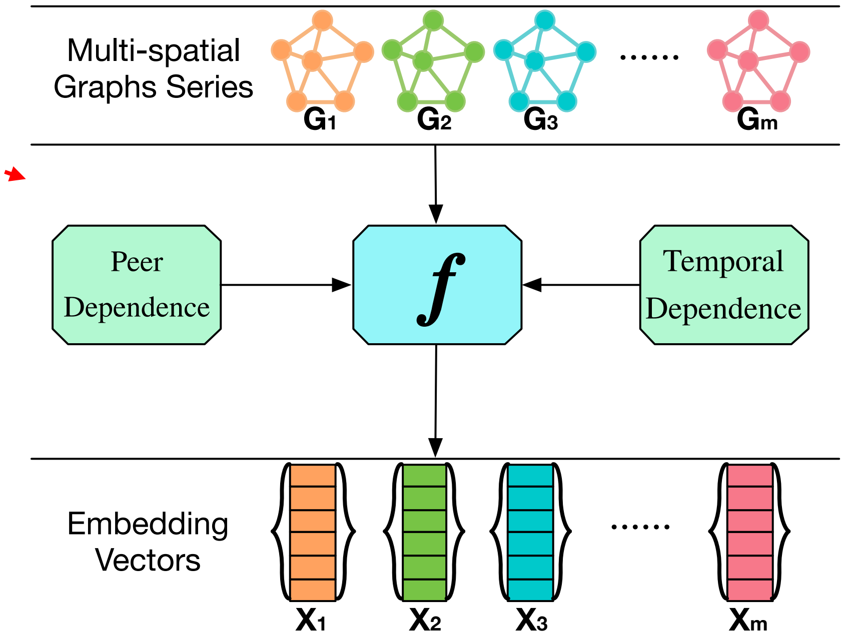
The similarity of driving behavior between the driver u_i and u_j at the time slot τ

$$s_{i,j}^{\tau} = \cos(\boldsymbol{x}_i^{\tau}, \boldsymbol{x}_j^{\tau})$$

using descriptive statistics of various historical driving operations

A Joint Optimization Objective

Model Structure



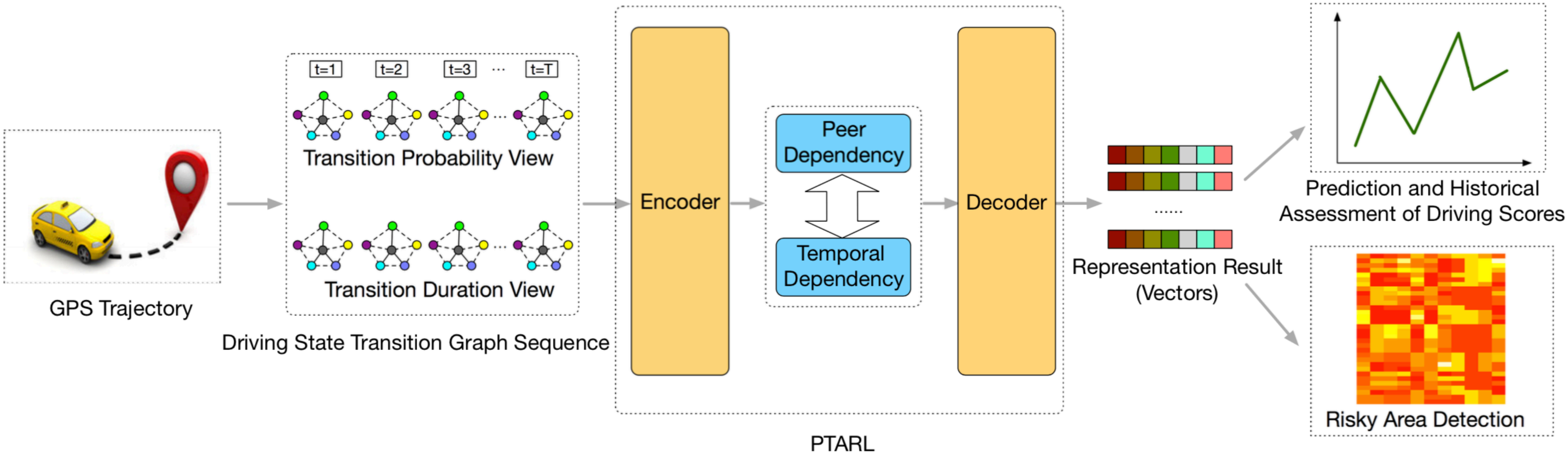
Structural reservation: the representation that is encoded from input can be decoded to recover input

$$\min \frac{1}{2} \sum_{\tau \in \mathcal{T}} \Big\{ \sum_{u_i \in \mathcal{U}(n)} \left\| (\mathbf{x}_i^{\tau} - \hat{\mathbf{x}}_i^{\tau}) \right\|_2^2 + \alpha \cdot \mathcal{H}_c(G^{\tau}) \Big\}$$

Temporal dependency: current embedding is related to past embedding

Peer dependency: the similar graph streams from two similar drivers share similar representations

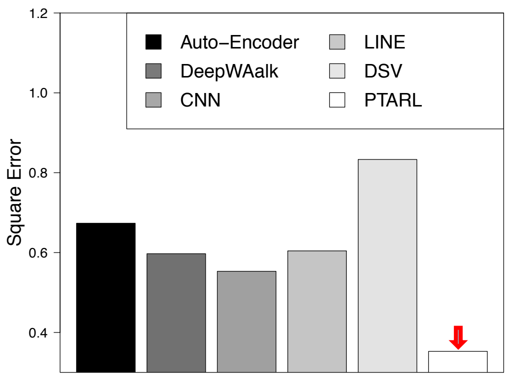
Applications: Driving Performance Scoring and Risky Area Detection

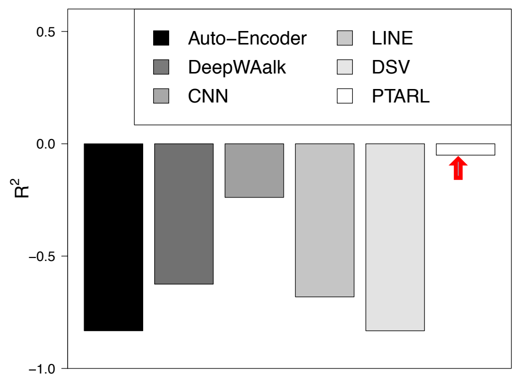


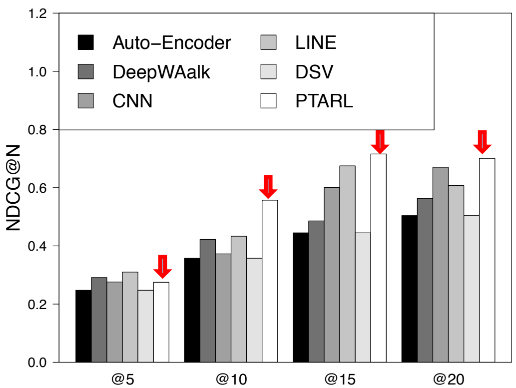
1. Learn driving behavior profiles from driving state transition graphs
2. Use driving behavior profiles to automatically score driving performances and detect risky areas

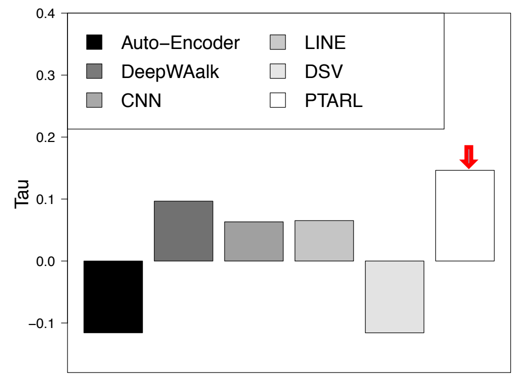
Comparison with Baseline Methods

Apply the learned representations to predict driving scores



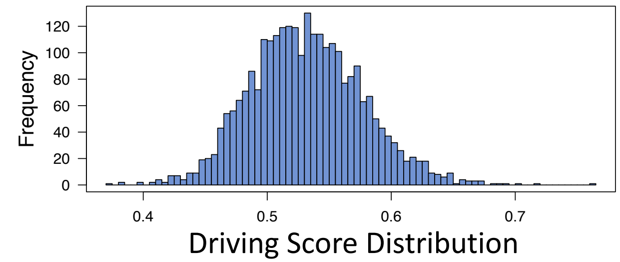






- Our model achieves the best performances
- Peer and temporal dependencies are essential for representing driving behavior

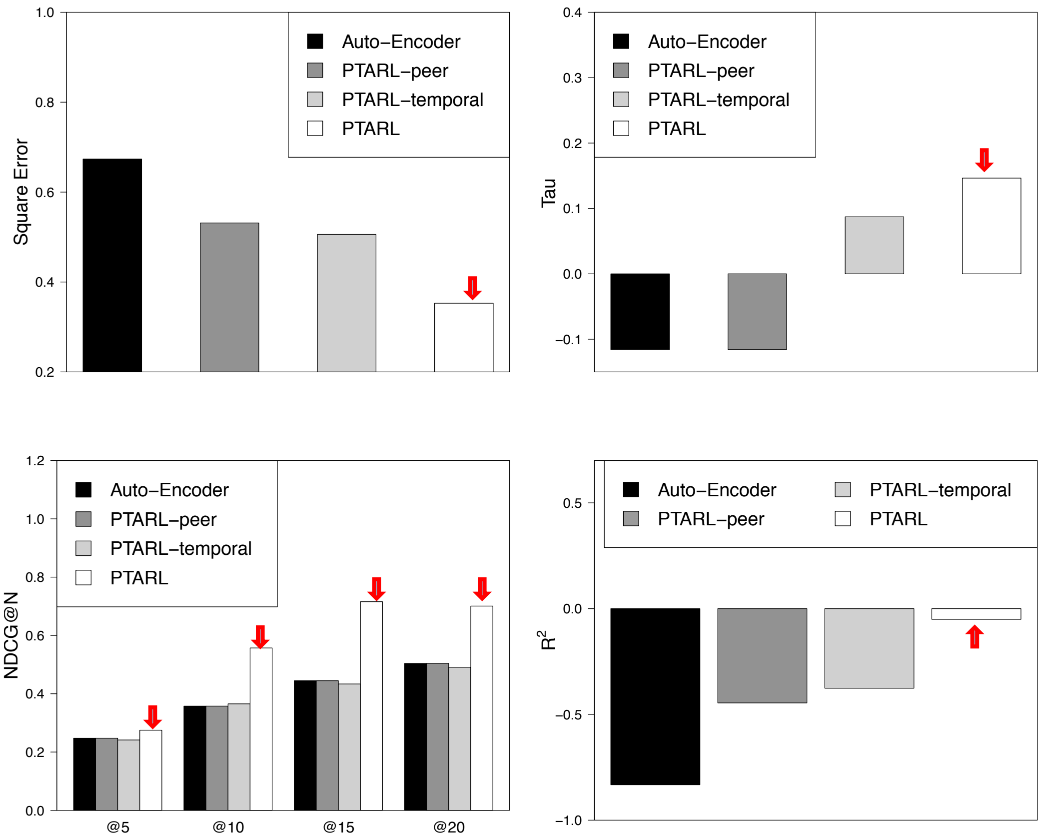
- **Data**
  - T-drive (Beijing GPS trajectories of volunteer drivers)



- **Evaluation Metrics**
  - Square Error
  - Coefficient of Determination (R^2)
  - Normalized Discounted Cumulative Gain (NDCG@N)
  - Kendall Tau Coefficient (Tau)
- **Baselines**
  - Autoencoder
  - DeepWalk: use truncated random walks to learn latent representations
  - LINE: preserve both local and global network structures with an edge-sampling algorithm
  - CNN: Convolutional Neural Network
  - Driving State Vector (DSV) – a traditional transportation approach
  - PTARL—Our model

Study of Peer and Temporal Dependencies

PTARL: -Our model

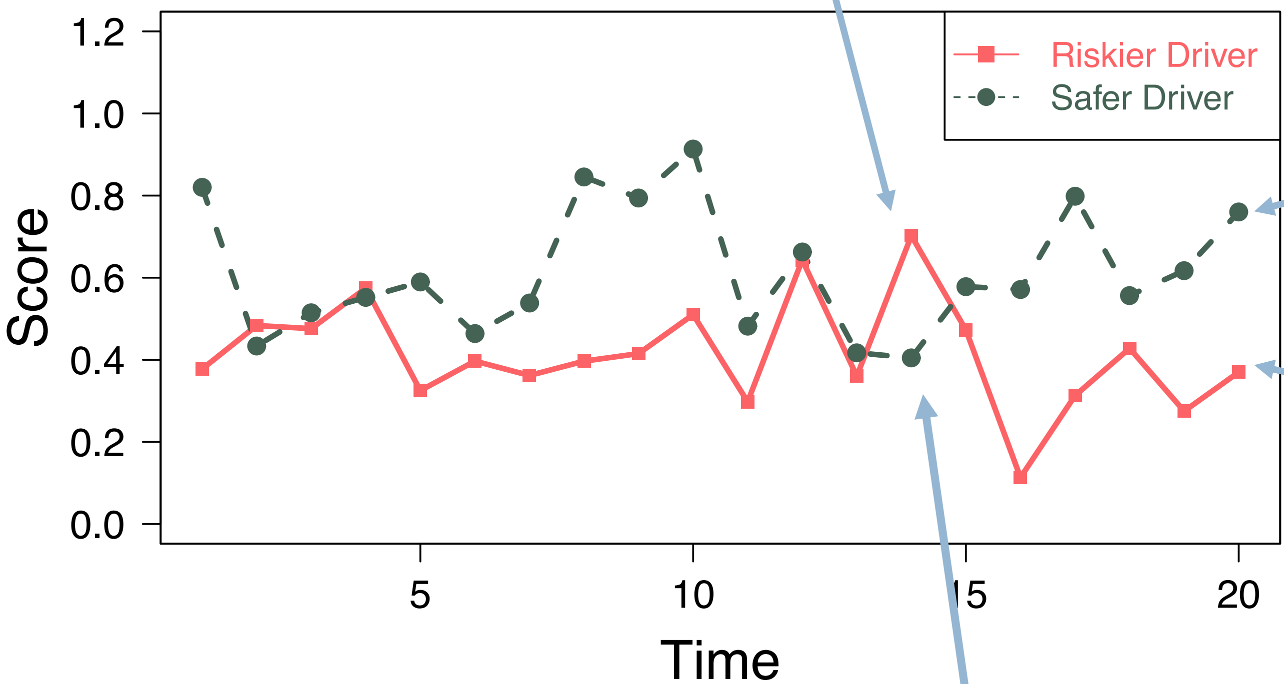


- Two variants of our model
  - PTARL-peer that only considers the peer dependency.
  - PTARL-temporal that only considers the temporal dependency.

- The Autoencoder that ignores both dependencies performs the worst
- The temporal dependency is more significant in profiling driving behavior than the peer dependency

Historical Assessment of Driving Scores

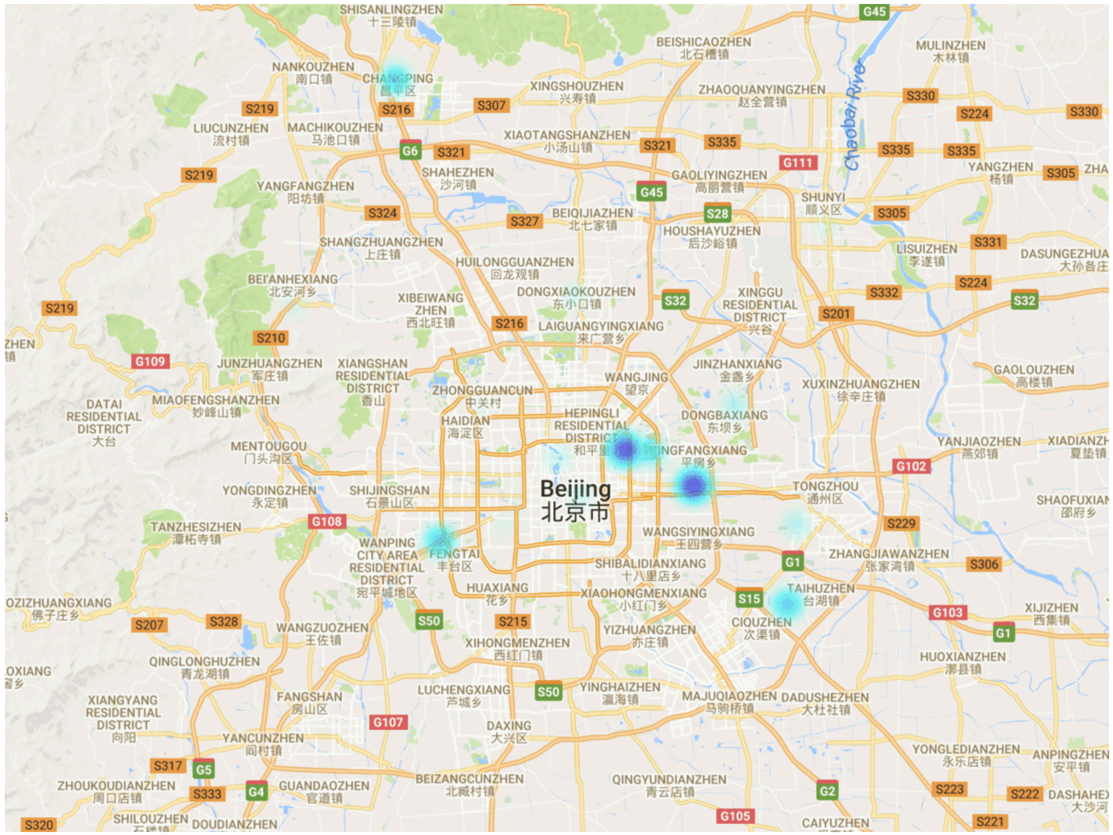
A "Riskier Driver" is not always risky



Scores of the "Safer Driver" are relatively higher at most time, while the scores of the "Riskier Driver" are relatively lower at most time

A "Safer Driver" is not always safe

Risky Area Detection



Dynamic evolution of the distribution of risky areas in 12 hours

Summary

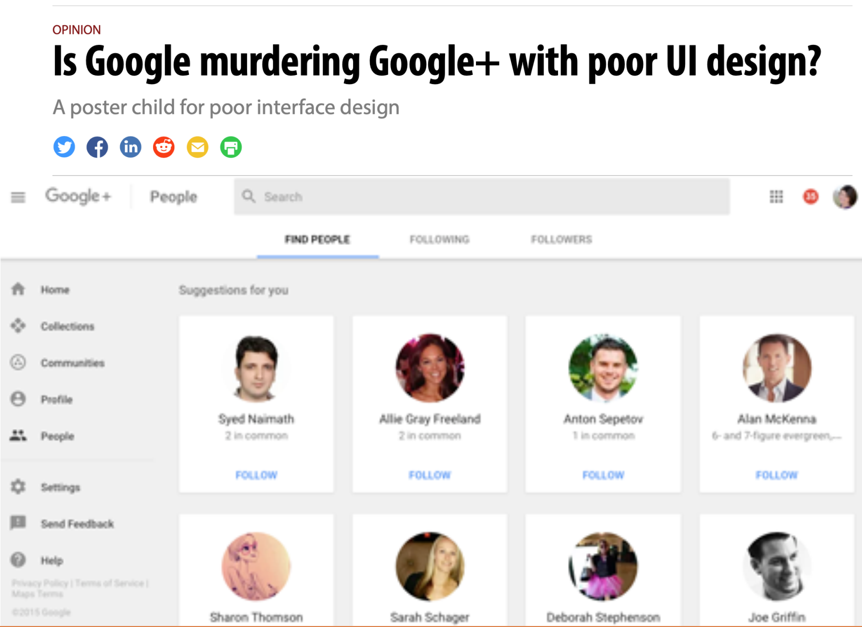
- **Task**
  - Dynamic representation learning with graph streams
- **Modeling**
  - Develop a temporal and peer-aware dynamic representation learning approach
  - Robustness checks over structural preservation, temporal dependency, and peer dependency
- **Application**
  - Driving behavior analysis for inferring driving scores and risk area detection

Outline

- Background and Motivation
- Deep Collective Representation Learning
- Deep Dynamic Representation Learning
- **Deep Structured Representation Learning**
- Conclusion and Future Work

Less Matches Between Human and Technologies

Non-personalized news feeds



Non personalized education

Too hard

Suitable

Too easy

Learning Resources

Personalized Learning

What can we do to improve user performance and engagement in human-technological systems?

Motivation Application: Precision User Profiling

Webpage = Contents + Structure

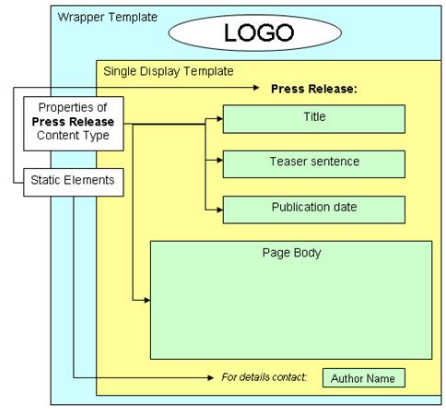
Webpage Content Template

Pre-Writing Questions:

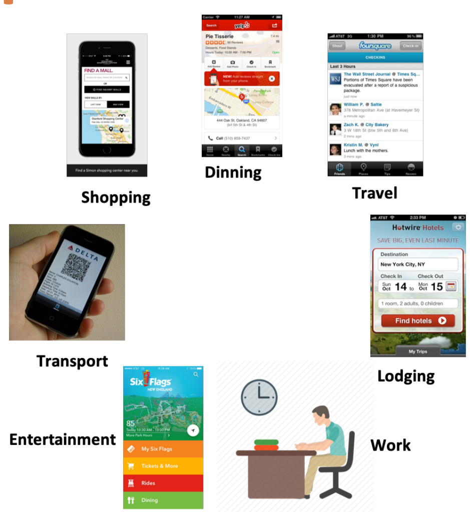
- What is the goal of this page?
- Which audience/persona is this page targeting?
- Which phase of their buying cycle is this page addressing?
- Based on the topic you're covering, what are the 3 primary benefits you want to communicate?
- What keywords and phrases do you need to include for SEO?

Page Headline / Title *(should be <h1>, clear and catchy, include primary keyword/phrase if possible)*

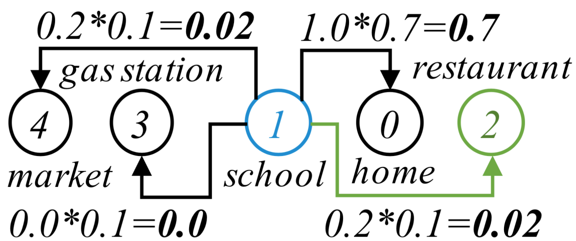
First paragraph – What is the ONE thing you want the



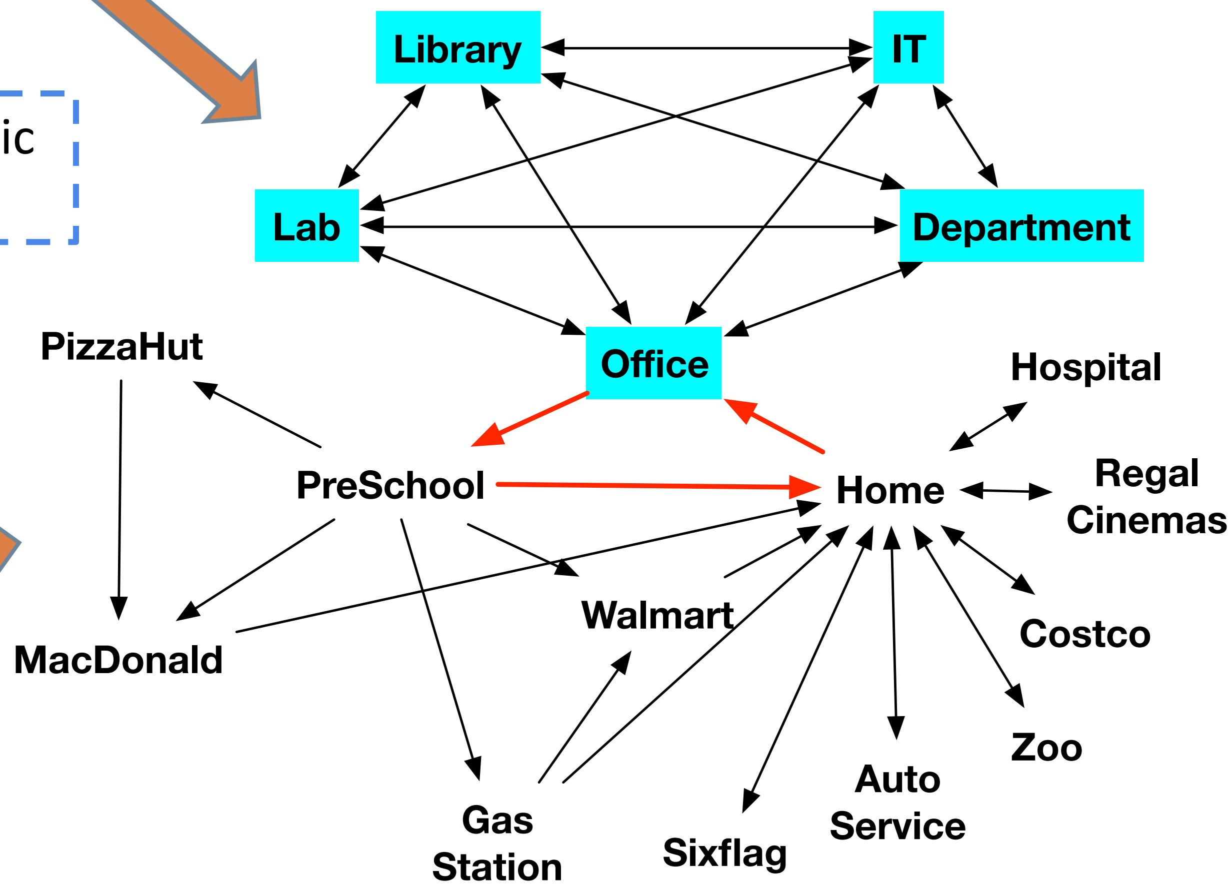
User = Explicit Activities + Latent Behavioral Structure ?



From Users To Activity Graphs

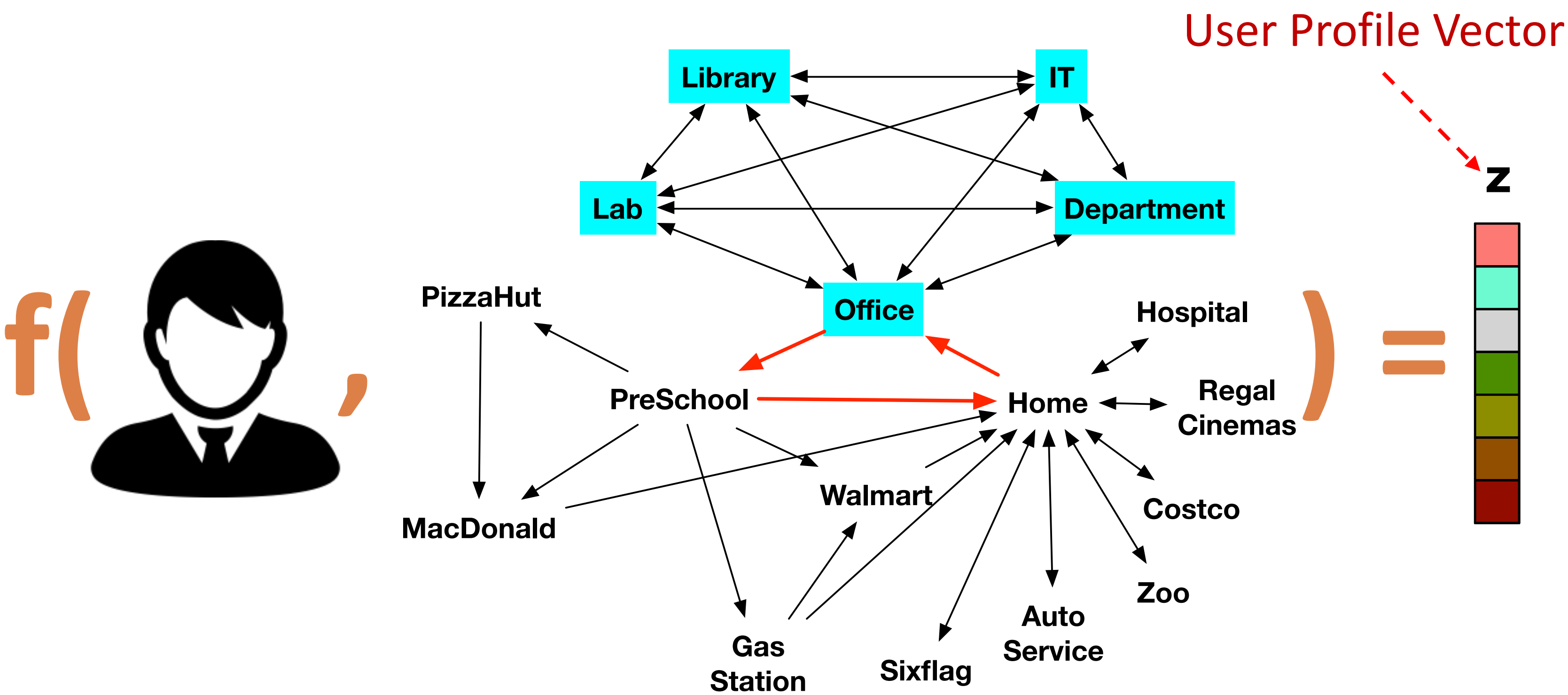


Spatial-temporal asymmetric transition patterns



User Activity Graph

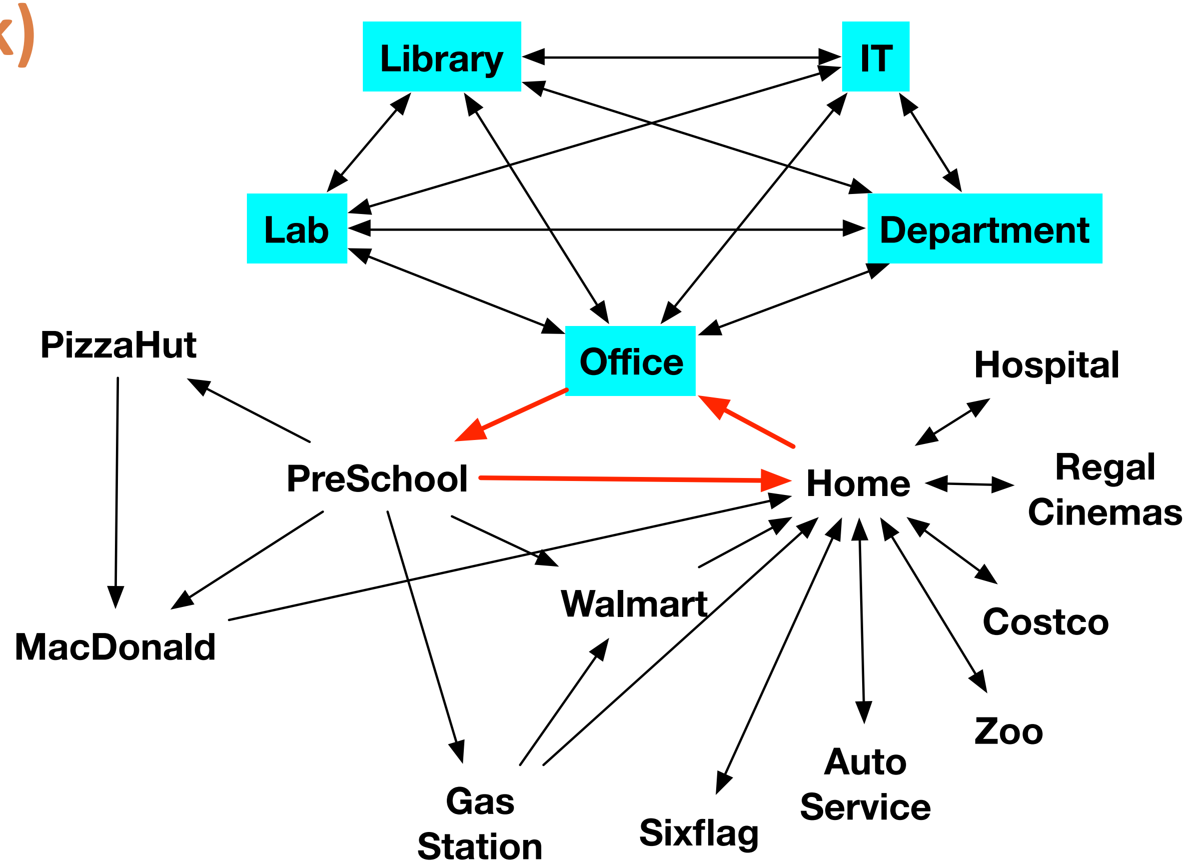
Problem Reformulation: Representation Learning with Activity Graphs



- Given a user and corresponding user activity graph, we aim to map the user to a profile vector

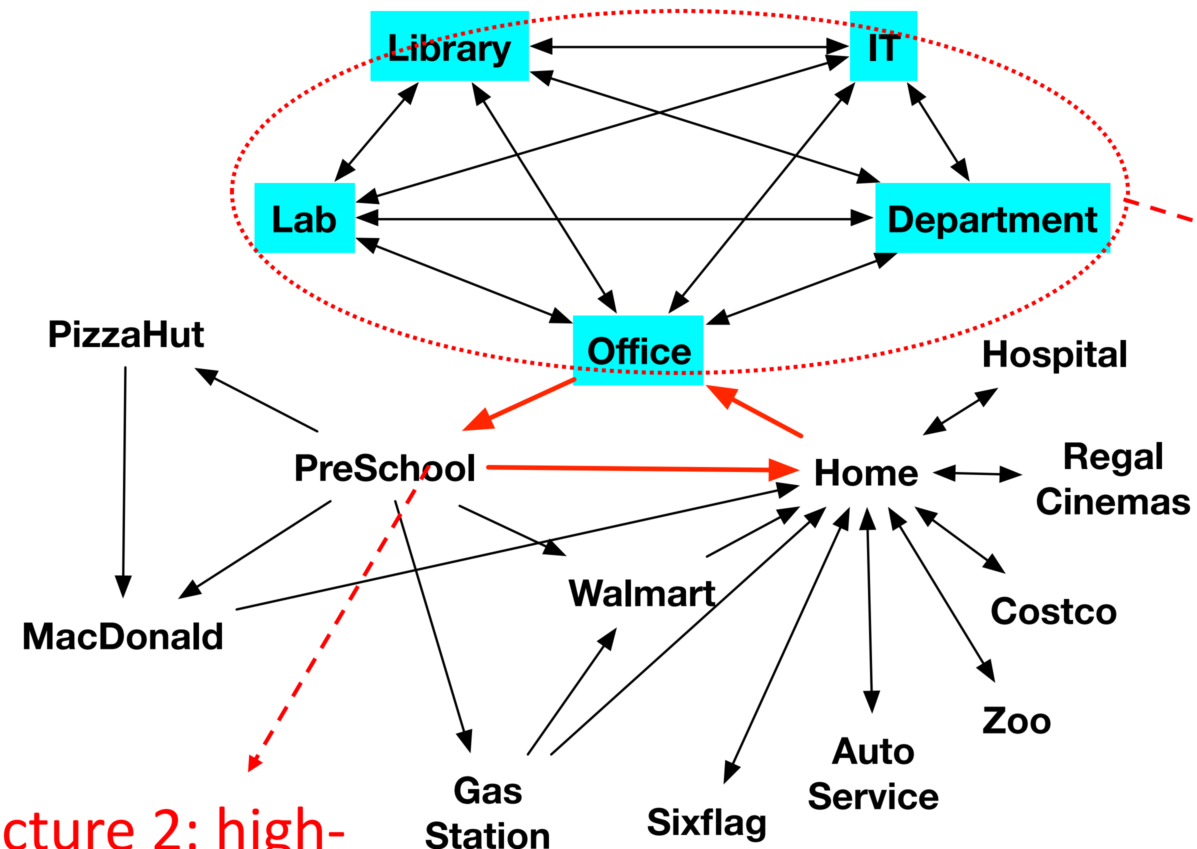
Global Behavioral Patterns

- Global structures: how a user' activities globally interact with each other (strongly link, weakly link, no link)



Substructure Behavioral Patterns

- **Substructures:** topology of subgraphs that feature the unique behavioral patterns of a user's activities

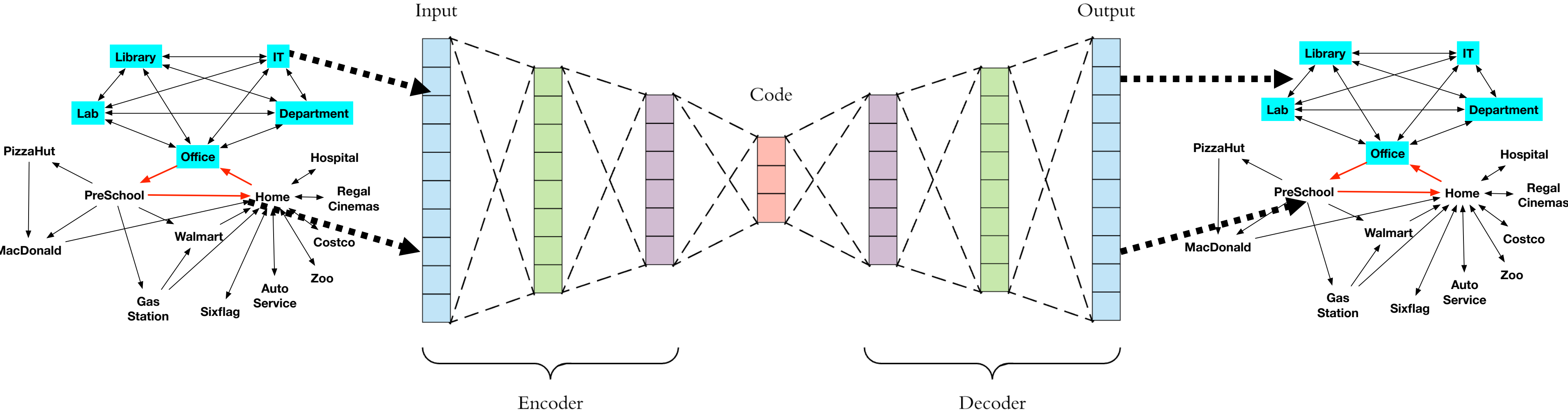


Substructure 1: high-frequency discrete nodes

Substructure 2: high-frequency circle

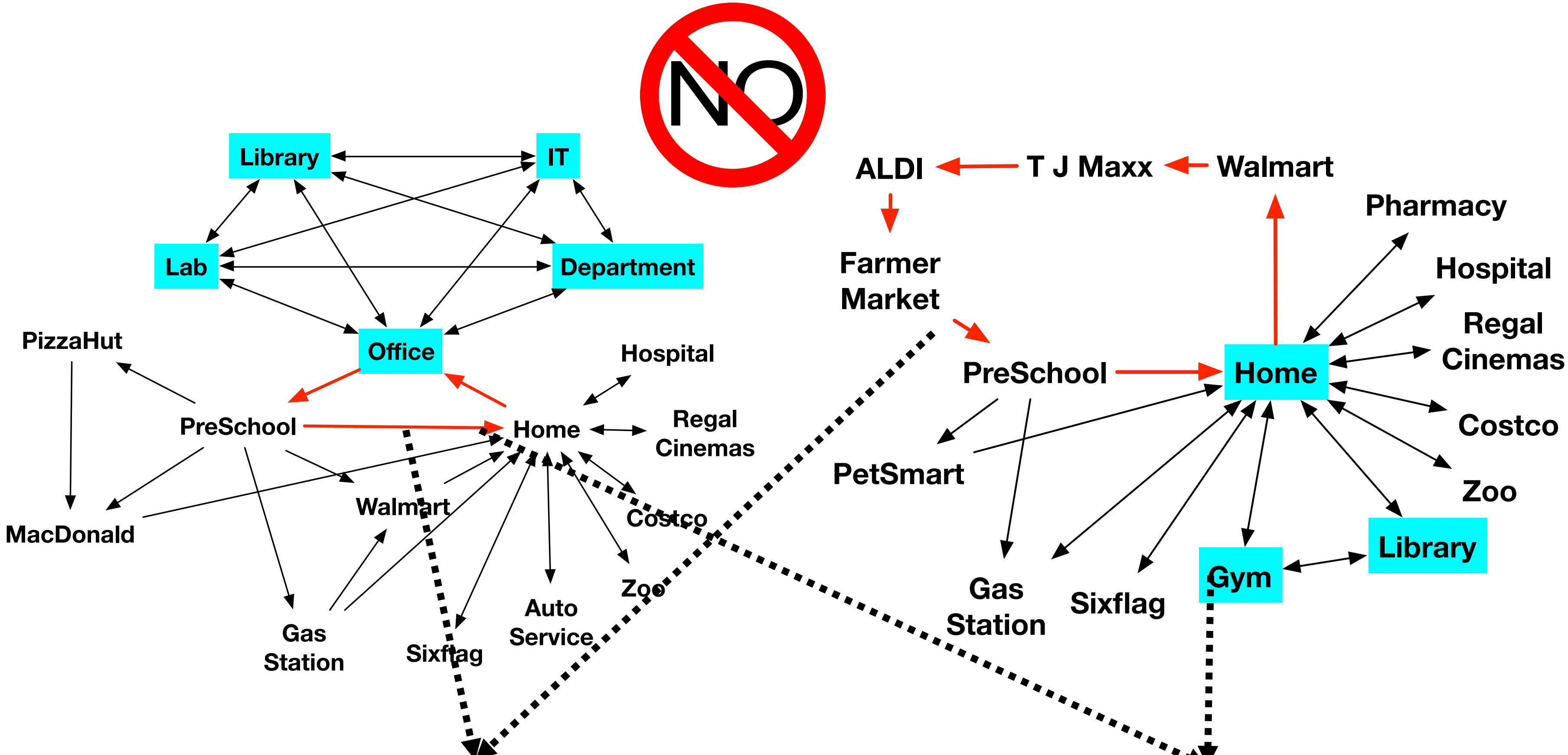
Representation Learning with Behavioral Global and Substructure Preservation

- Traditional solution: global structure (encoding-decoding) + substructure (loss regularization)



- Global structure:
  - Minimize the loss between the input graph and the reconstructed graph
- Substructure preservation:
  - Strongly penalize the loss if the model cannot accurately reconstruct substructures

Will The Traditional Solution Work?



- Different users show different substructure topology and contents

- Substructure are dynamically distributed in different locations of graphs

Adversarial Substructured Learning

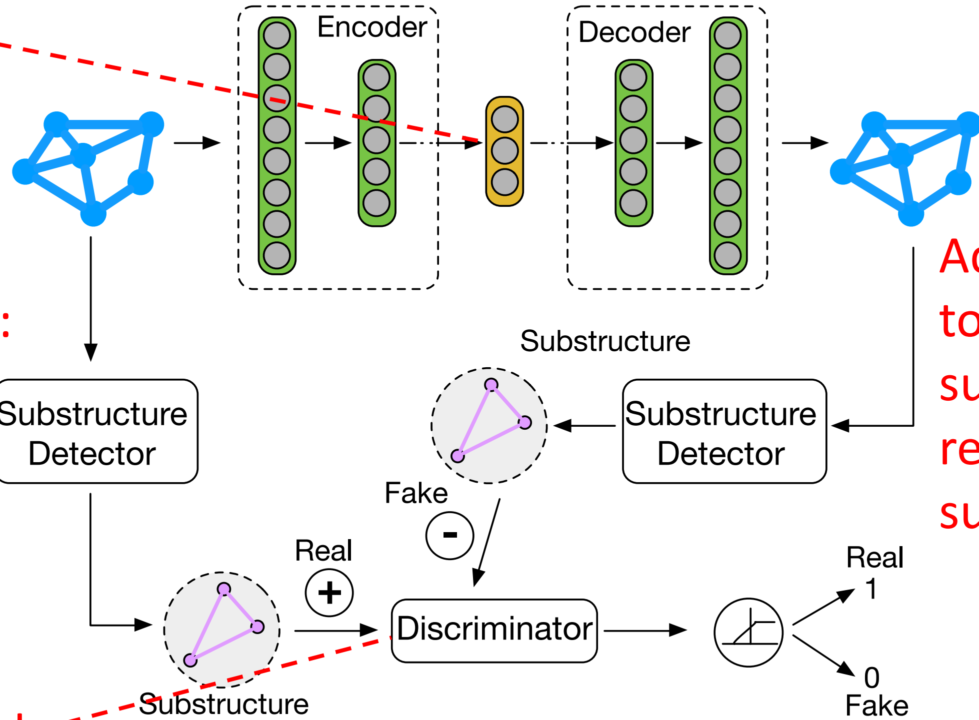
- Translate substructure-aware representation learning into an adversarial substructured learning problem

An encoder-decoder network: learn the representations of a graph

A substructure detector: detect substructure patterns

A discriminator: classify original substructure and reconstructed substructure

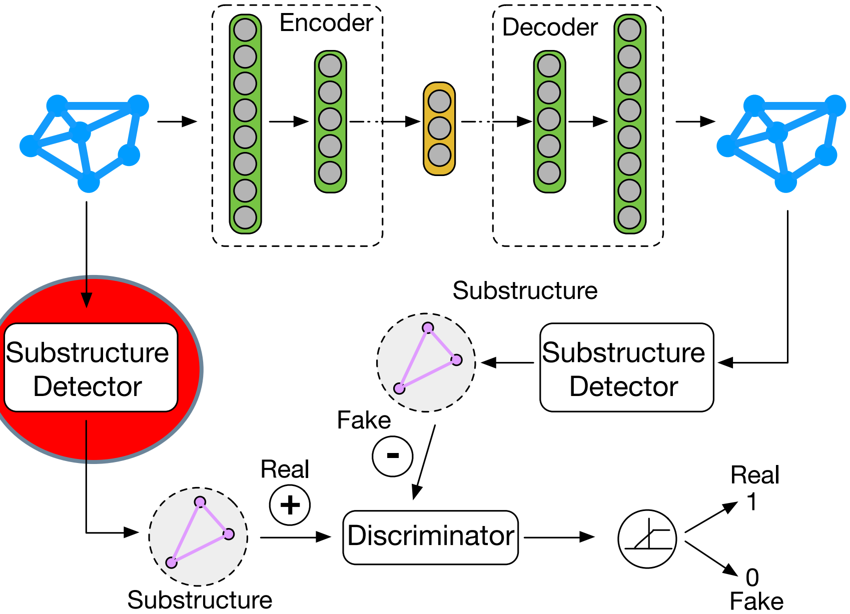
Adversarial training: to match original substructures with reconstructed substructures



Will The New Formulation Work?

deep first search based subgraph detection

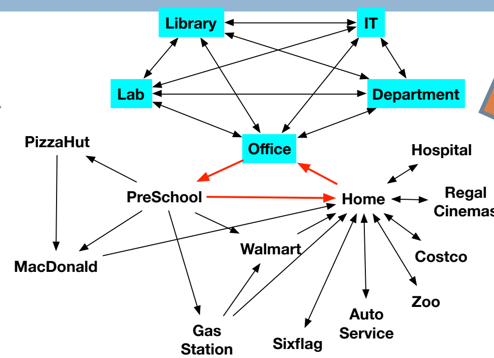
```
Procedure DFS(input: graph G)
begin
   Stack S;
   Integer s,x;
   while (G has an unvisited node) do
      s := an unvisited node;
      visit(v);
      push(v,S);
      While (S is not empty) do
         x := top(S);
         if (x has an unvisited neighbor y) then
            visit(y);
            push(y,S);
         else
            pop(S);
         endif
      endwhile
   endwhile
end
```



- Traditional subgraph detection algorithms are usually not differentiable
- Impossible to backpropagate gradience for optimization

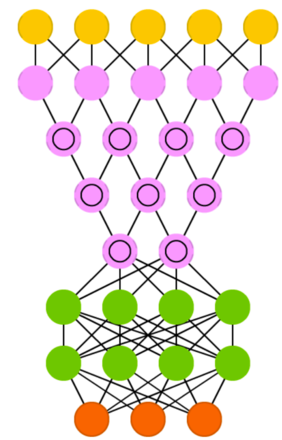
How to Approximate Substructure Detector?

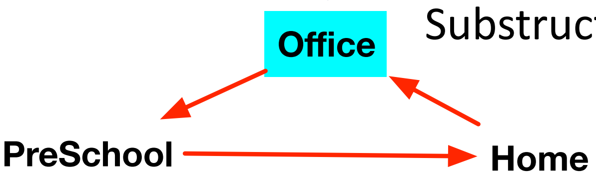
Non-differentiable substructure detection algorithm



Differentiable CNN

```
Procedure DFS(input: graph G)
begin
   Stack S;
   Integer s,x;
   while (G has an unvisited node) do
      s := an unvisited node;
      visit(v);
      push(v,S);
      While (S is not empty) do
         x := top(S);
         if (x has an unvisited neighbor y) then
            visit(y);
            push(y,S);
         else
            pop(S);
         endif
      endwhile
   endwhile
end
```





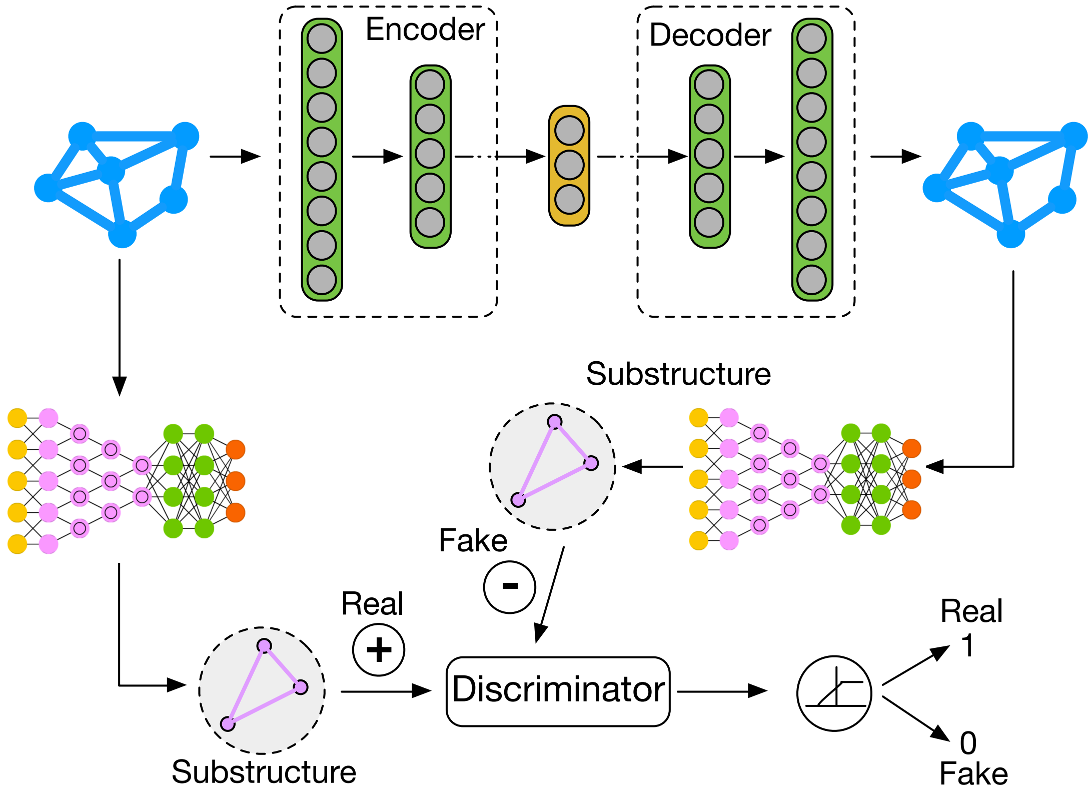
Substructure

Latent embedding

- Use CNN to replace substructure detection algorithms
- Use an embedding vector to replace a subgraph

Approximated Adversarial Substructured Learning

CNN-based detector: detect and output a substructure feature vector



The Mini-Max Game in Optimization

- Discriminator: is trained to maximize the accuracy of classifying detected and generated substructures
- Generator: is trained to minimizing the probability that Discriminator correctly classify generated substructures

Solving The Min-Max Game

1. Minimizing Objective Function

Discriminator accuracy

Reconstruction loss

$$\mathcal{L} = -\lambda_D \mathcal{L}_D + \lambda_G \mathcal{L}_G + \lambda_{AE} \mathcal{L}_{AE}$$

Train G to minimize D's accuracy on generated substructures

2. Update discriminator to maximize accuracy

$$-\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(\mathbf{s}^i) + \log(1 - D(G(\mathbf{x}^i)))]$$

Maximize likelihood

Classify ground true substructure to 1

Classify generated substructure to 0

3. Update generator to confuse discriminator

$$\nabla_{\theta_{AE}} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(\mathbf{x}^i)))$$

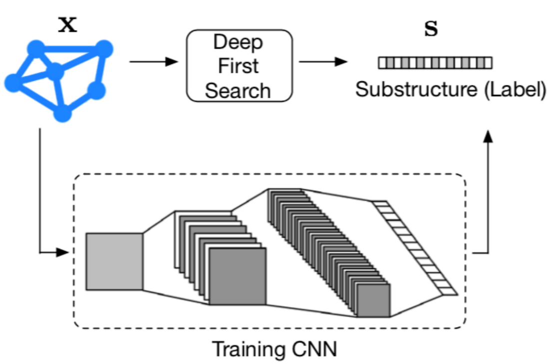
Minimize likelihood

Classify generated substructure to 0

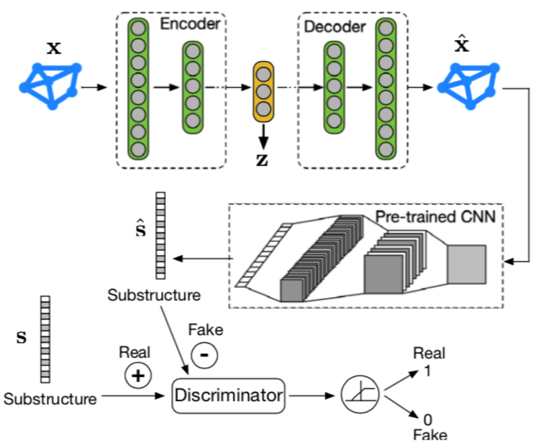
4. Minimize reconstruction loss

$$\nabla_{\theta_{AE}} \|(\mathbf{x}_i - \hat{\mathbf{x}}_i)\|_2^2$$

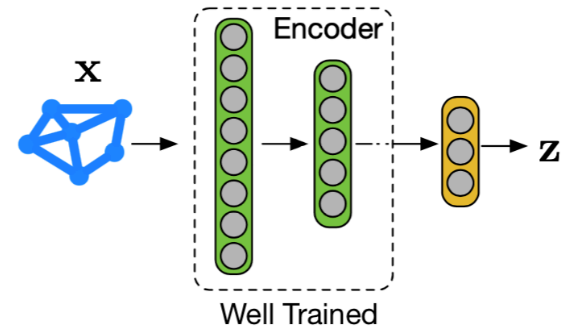
Recap: Training and Testing of Adversarial Substructured Learning



(a) Pre-train the CNN to approximate the substructure detector.

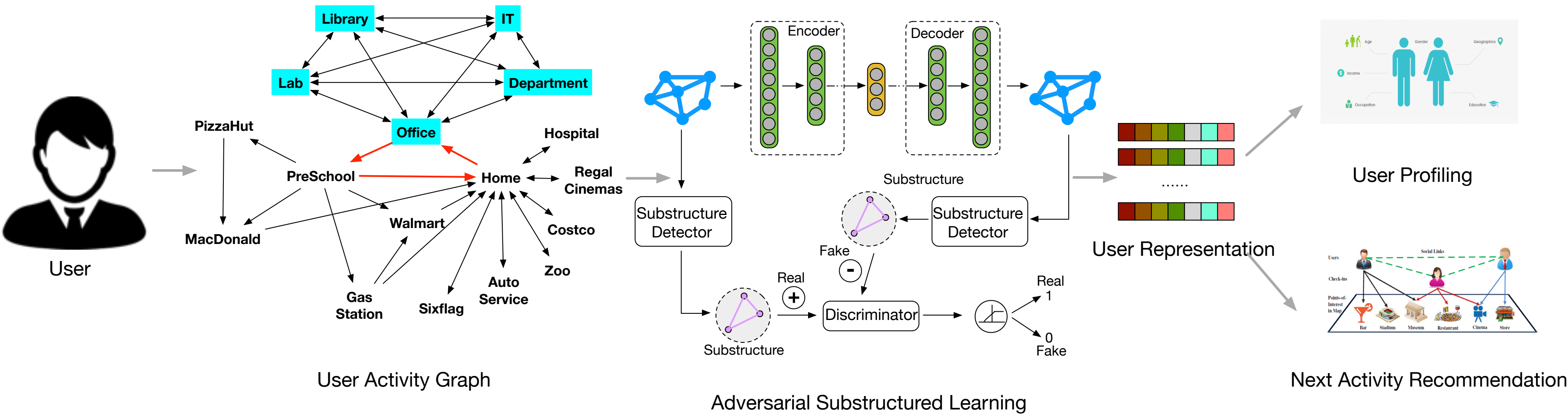


(b) Adversarial training process to integrate the substructure.



(c) Utilize the well-trained model to generate representations of mobile user profiles.

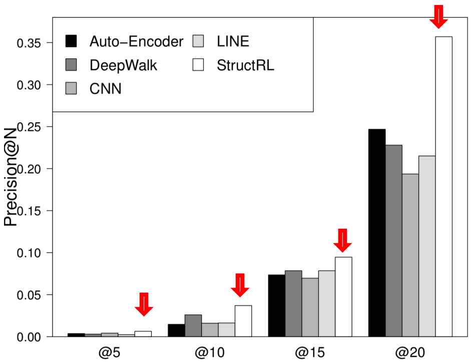
What To Do Next: Inferring Next Activity for POI Recommendations



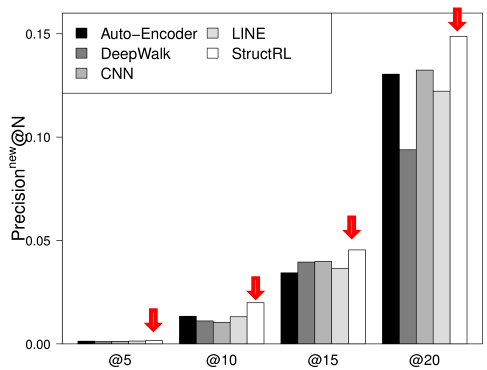
1. **Given a time period, learn a user's profiles from corresponding user activity graph**
2. **Exploit user profiles to forecast next activity category**

Performance Comparisons on New York and Tokyo Activity Checkin Data

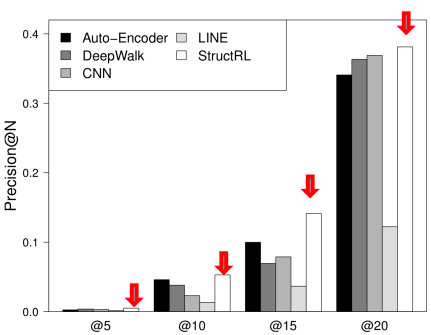
Apply the learned representations to predict next activity type (next POI category)



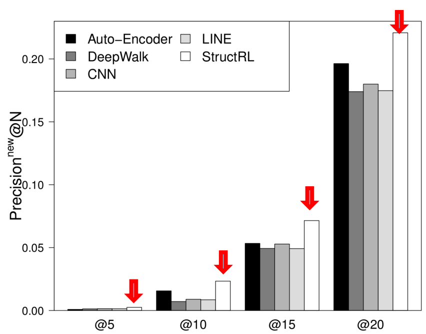
(a) Precision@N with New York dataset



(b) Precision^{New}@N with New York dataset



(c) Precision@N with Tokyo dataset



(d) Precision^{New}@N with Tokyo dataset

- Our model achieves the best performances on user profiling
- Substructures in a graph are essential for user behavior patterns

Data

- Mobile activity checkin data of NYC and Tokyo

City	# Check-ins	# POI Categories	Time Period
New York	227428	251	12 April 2012 to 16 February 2013
Tokyo	573703	247	12 April 2012 to 16 February 2013

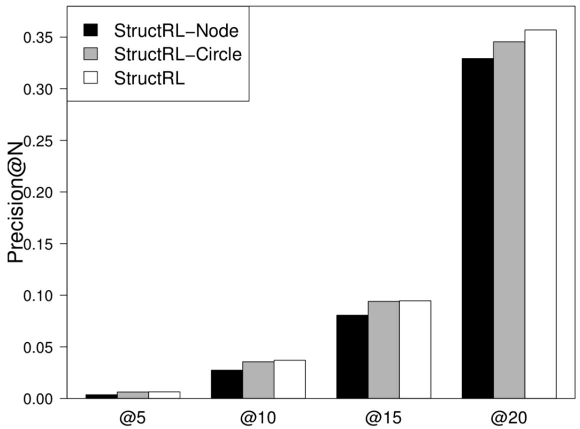
Evaluation Metrics

- The precision@N of activity category prediction
- The precision@N of new activity recommendation

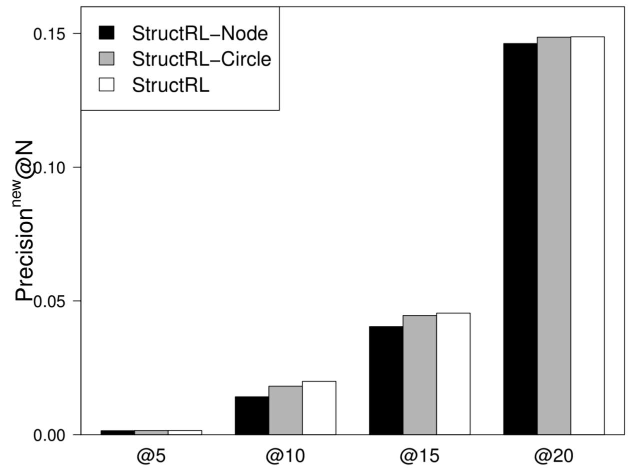
Baselines

- Autoencoder
- DeepWalk: use truncated random walks to learn latent representations
- LINE: preserve both local and global network structures with an edge-sampling algorithm
- CNN: Convolutional Neural Network

Study of Node and Circle Substructures

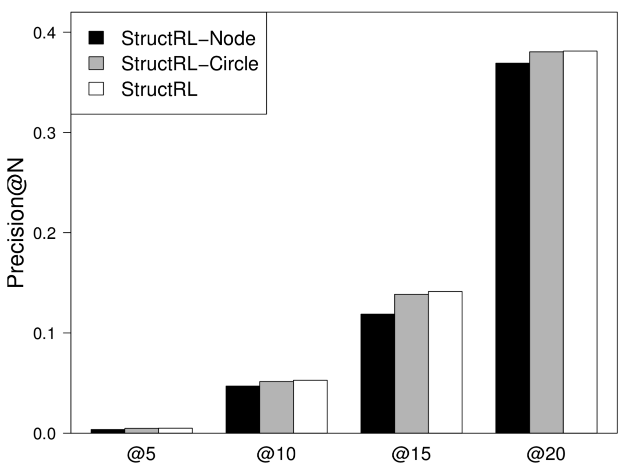


(a) Precision@N with New York dataset

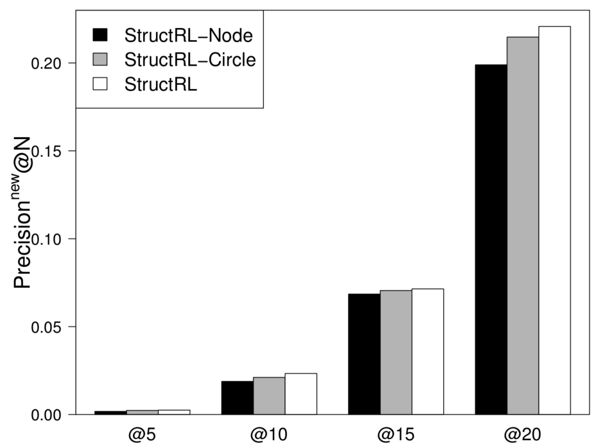


(b) Precision^{New}@N with New York dataset

- StructRL: consider both nodes and circles
- StructRL-Node: only consider node based substructure
- StructRL-Circle: only consider circle based substructure



(c) Precision@N with Tokyo dataset



(d) Precision^{New}@N with Tokyo dataset

Summary

- **Task**
  - Structured representation learning with global and sub structure preservation
- **Modeling**
  - Develop an adversarial substructured learning approach
  - Preserving global and sub structures via solving the mini-max game
- **Application**
  - Precision user profiling and quantification for personalization and recommender systems