Selective Cross-City Transfer Learning for Traffic Prediction via Source City Region Re-Weighting

Yilun Jin[1], Kai Chen[1], Qiang Yang[1, 2]

[1] HKUST, [2] WeBank

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

WeBank

Outline

- Background & Motivation
- Problem Definition
- Proposed Method: CrossTReS
- Experiments
- Conclusion

Background: Traffic Prediction

- Traffic Prediction
  - Forecasting future human flows, traffic speeds, travel demands, etc.

Traffic flow:
How many cars passed the intersection?

Travel demand:
How many taxis stopped for passengers?

Traffic speed:
What is the average speed of this lane?

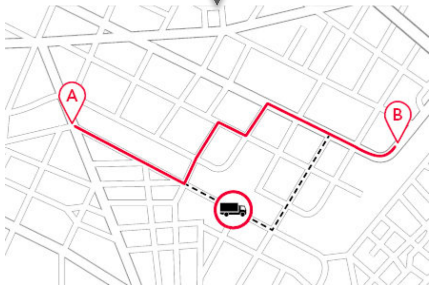
Background: Traffic Prediction

- Traffic Prediction
    - Forecasting future human flows, traffic speeds, travel demands, etc.
    - Foundations for smart transportation tasks, e.g. route planning, vehicle dispatching

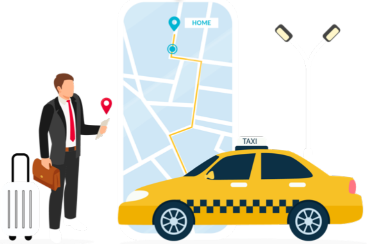
Traffic flow:
How many cars passed the intersection?

Travel demand:
How many taxis stopped for passengers?

Traffic speed:
What is the average speed of this lane?



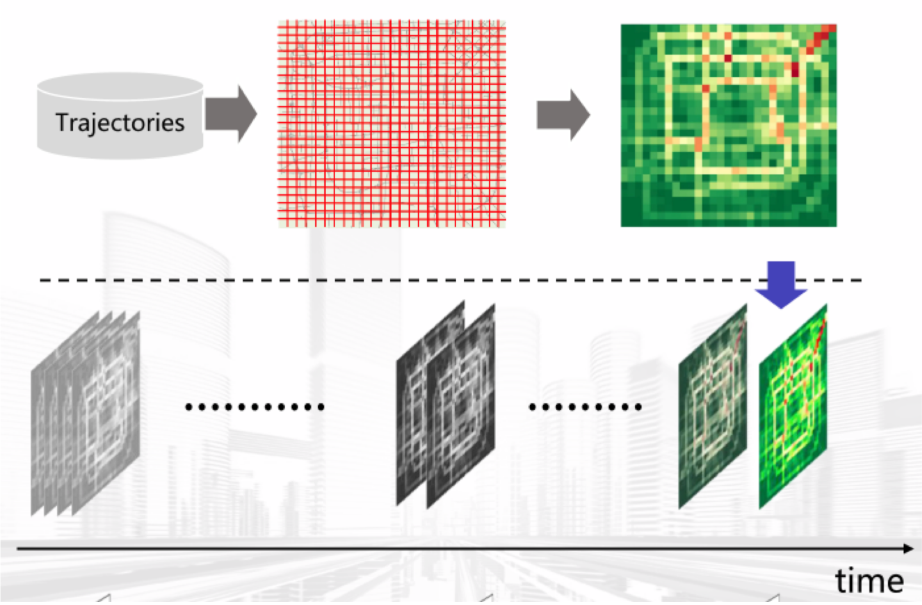
Route Planning



Taxi Dispatching

Background: Deep Learning for Traffic Prediction

- Deep learning models achieve success in traffic prediction.
  - e.g. CNN [Zhang et al. 2017], RNN [Yao et al. 2019b], GCN [Li et al. 2018],

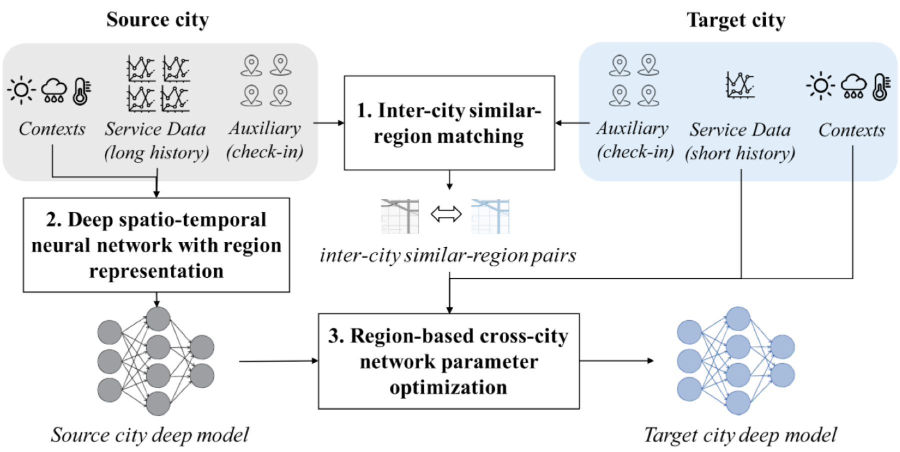


Example: CNN-based models
- Represent a city with grids
- Deep CNN models to learn spatio-temporal features.

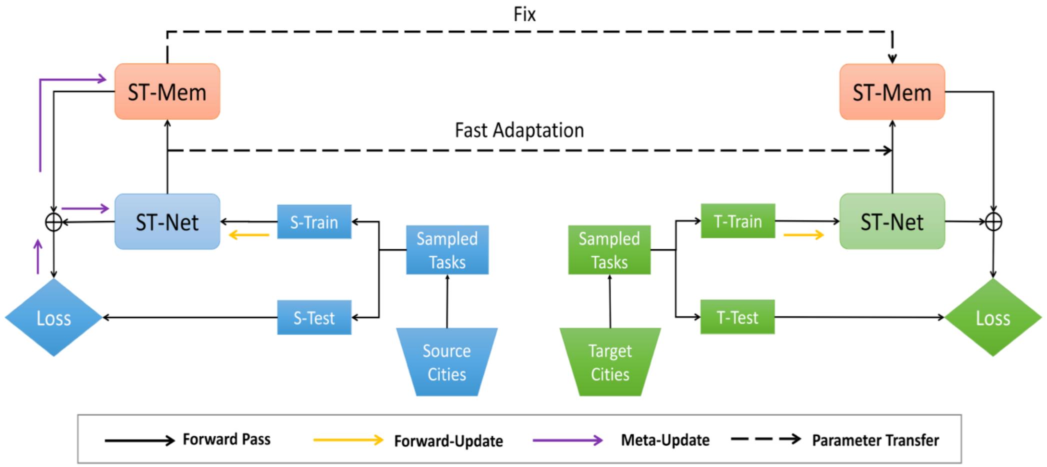
  - **Drawback:** Require large-scale traffic data (e.g. a year)
  - **Question:** What if we only have limited data?
    - e.g. Under-developed cities.

Background: Transfer Learning for Traffic Prediction

- **Cross-city transfer learning** for traffic prediction:
  - Transfers knowledge from data-rich cities to data-scarce cities.
  - **Examples:** RegionTrans [Wang et al. 2019], MetaST [Yao et al. 2019a]
  - **Main Methods:** Fine-tuning



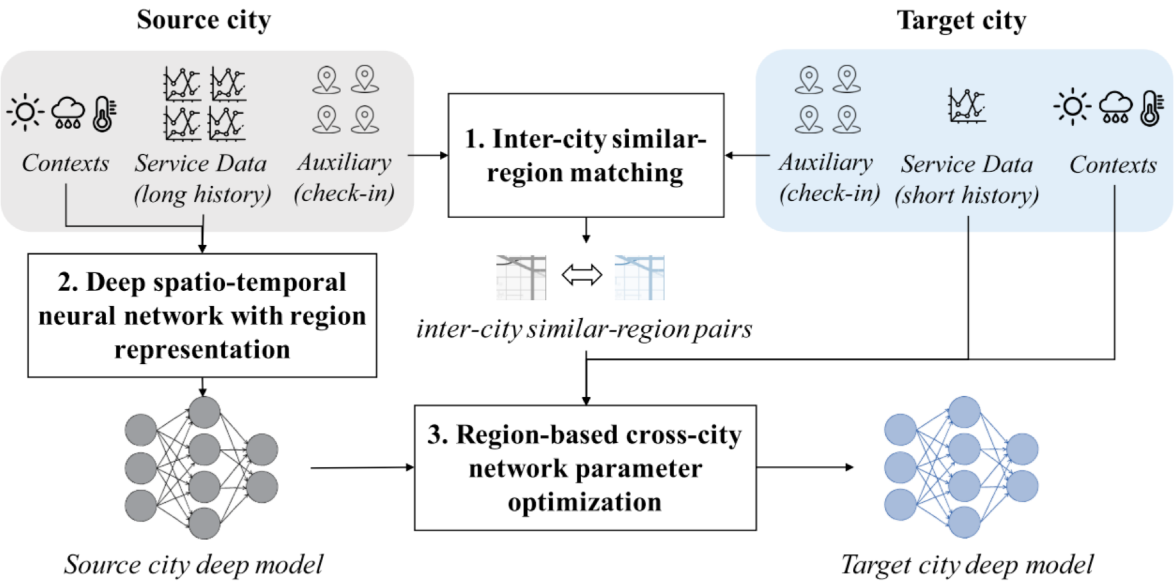
RegionTrans: Fine-tuning
with auxiliary data similarity



MetaST: Fine-tuning +
Long-term memory

Background: Fine-Tuning Solutions

- **Example:** RegionTrans [Wang et. al, 2019]
  - **Step 1:** Finding similar cross-city region pairs.
  - **Step 2 (Source Training):** Train the model on abundant data from **source city.**
  - **Step 3 (Fine-Tuning):** Fine-tune the model with target data & region similarity.



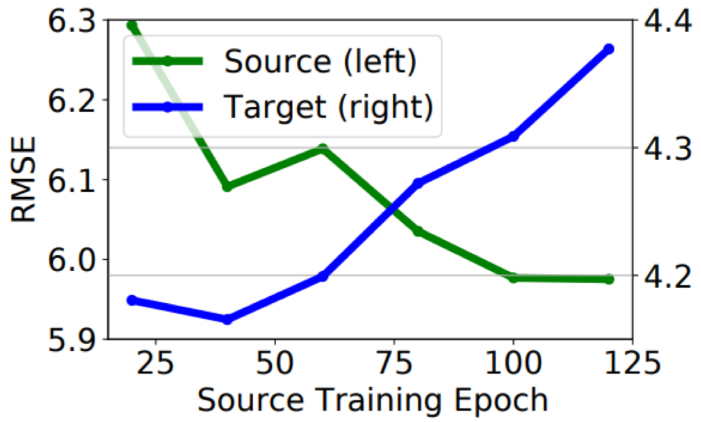
Motivation

- **Common drawback** of fine-tuning-based methods:
  - Focus on designing novel fine-tuning methods.
  - Ignore the impact of source training: may learn irrelevant source knowledge.

- **Our observation:**
  - Inadequate source training is **harmful**.

Motivation: Experiments

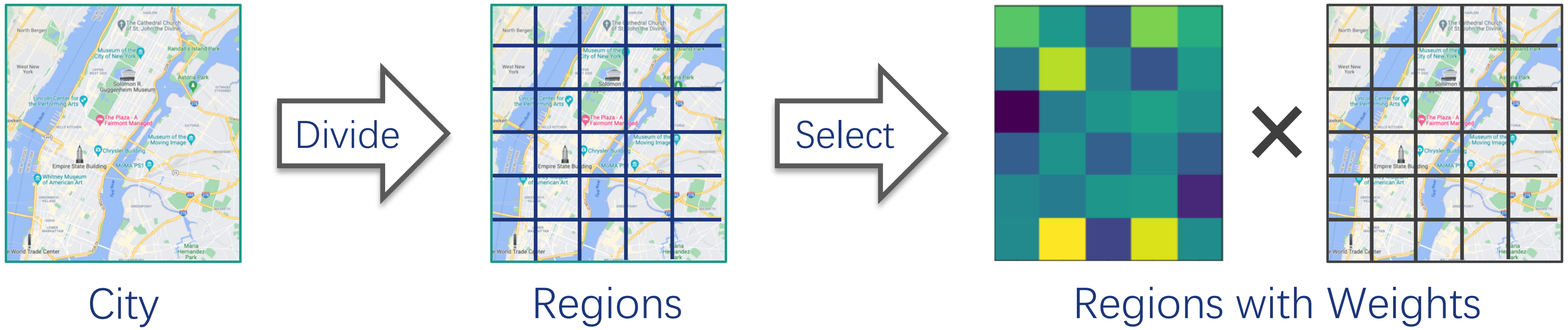
- Real-world taxi data
  - **Source City**: Chicago;
    Target City: Washington DC (7 days)
- Vary number of epochs for source training.
- Results:
  - More source training
    ➔ **Lower** source error
    ➔ **Higher** target error.
  - **Source training learns harmful knowledge!**



(a) Supervised Source Training

Problem Definition

- **Goal:** Selective Transfer Learning
  - Select relevant knowledge, rule out harmful knowledge.
- **How?**
  - **Common Practice:** Divide cities to regions [Wang et al. 2019]
  - We select knowledge by **re-weighting regions**.
  - **Advantages:**
    - Better transfer learning performance.
    - Better interpretability (by visualization).



Problem Definition

- **When?**
  - **Selective Source Training:**
    Source knowledge is learned during source training, not fine-tuning.
- **Problem Definition:**
  - For each region r_S in the source city S, learn weights $\lambda_{r_S} > \mathbf{0}$, such that after
    1. Source training with weights $\lambda_{r_S} > \mathbf{0}$, and
    2. Target fine-tuning,

    error on the target city is minimized.

Proposed Work

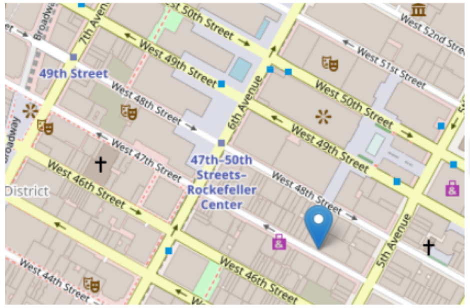
- CrossTReS (Cross-City Transfer Learning with Region Selection)
  - **General** framework for selective source training.
  - **Agnostic** to fine-tuning methods.
  - Up to **8% error reduction** on real-world taxi and bike datasets.
  - **Interpretable** cross-city knowledge transfer.

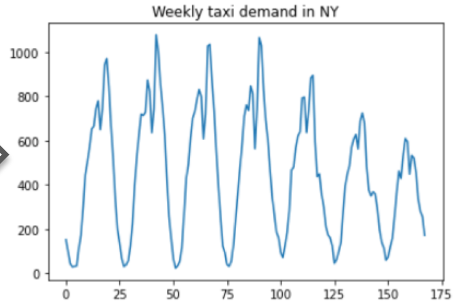
Main Ideas

- **Idea 1**: Regional **urban features** shed light on traffic patterns.
  - e.g. Industrial areas → morning and evening rush hours
    Business centers → traffic flows peak during weekends

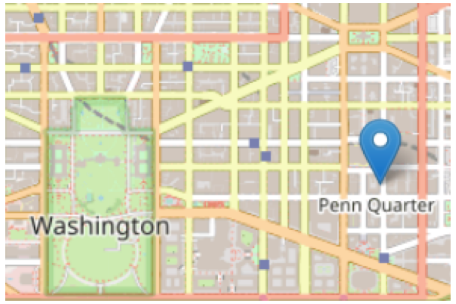
  - **Cross-city regions** with similar features → similar traffic patterns.

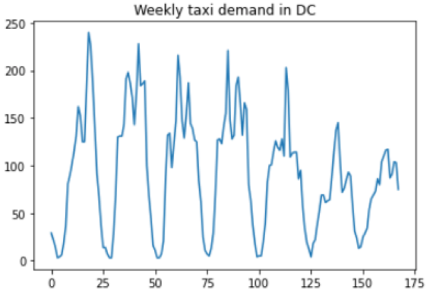
  - **Challenge 1:** How to learn generalizable region features in both cities?





Downtown NY: Rockefeller Center





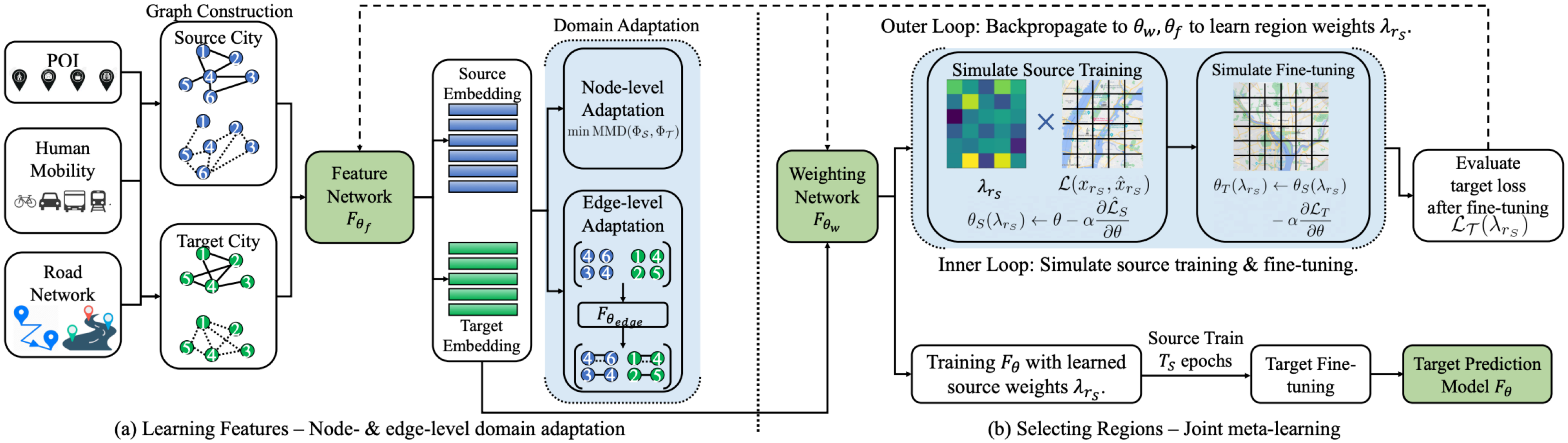
Downtown DC: Penn Quarter

Main Ideas

- **Idea 2**: 'Helpful' source regions should be assigned high weights, and vice versa.
    - e.g. Target city enjoys smooth traffic flows → Source regions with heavy congestion should be given low weights.
    - **Challenge 2:** How to quantify such 'helpfulness'?

Components

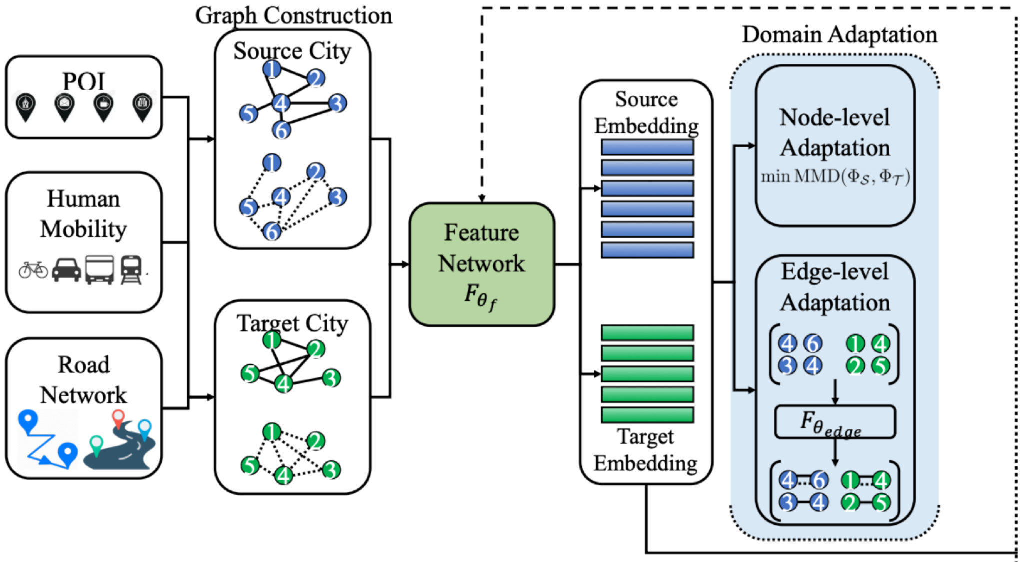
- Feature network F_{θ_f}: Graph-based models to learn region features
- Weighting network F_{θ_w}: Learns weights λ_{r_S} for source regions
- Prediction model F_θ : Performs traffic prediction.



Learning Generalizable Region Features

- Regional Feature Learning
  - **Common Practice:** Build multi-view graphs within a city [Zhang et al. 2020]
    - Nodes (regions) linked by various relations, e.g. similar POI, similar human mobility, road connections, etc.
  - **City-specific**: only reflects intra-city relations.

- **Generalizable** Region Feature Learning
  - **Goal**: Similar regions across cities have similar features.
  - How?
    - **Node** and **edge-level domain adaptation**

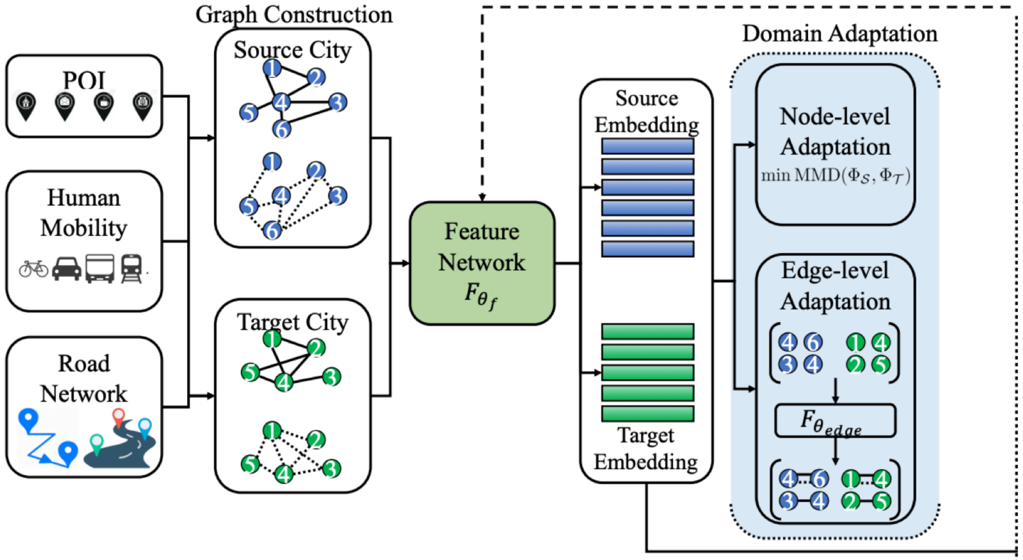


(a) Learning Features – Node- & edge-level domain adaptation

Learning Generalizable Region Features

- Node-level:
  - Maximum mean discrepancy (MMD)
  - Aligns distribution of node features.

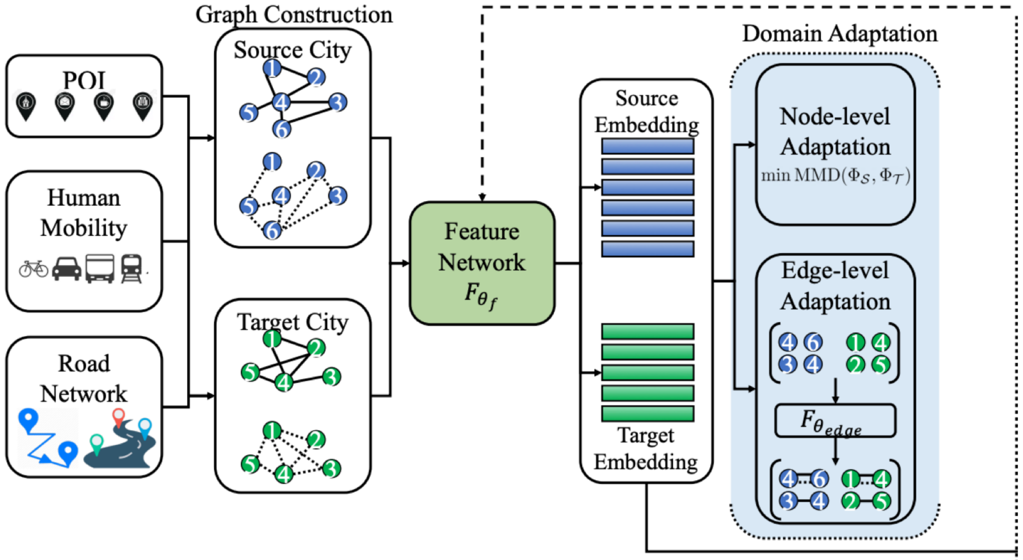
- Edge-level:
  - **Intuition: Use edge types to link cities.**
    1. Different types of edges ➔ **separable** edge features.
    2. Different cities, same edge types ➔ **similar** edge features
  - **Method:** Shared edge classifier.



(a) Learning Features – Node- & edge-level domain adaptation

Learning Generalizable Region Features

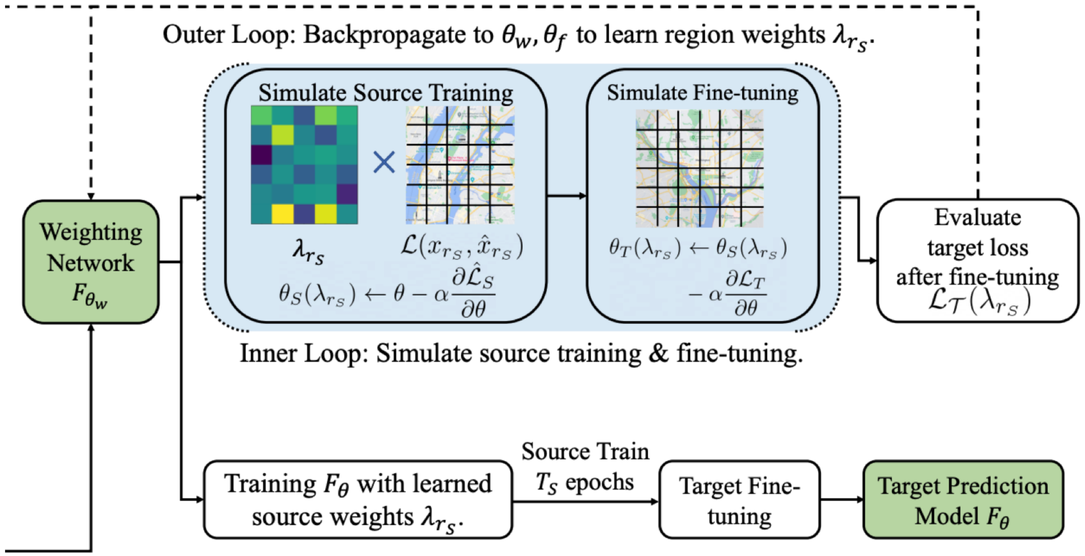
- Edge classifier $F_{\theta_{edge}}$:
  - **Input:** edge features
    (concat. of node features)
  - **Predict:** edge type
    (Intuition 1)
  - **Shared** between source & target cities
    (Intuition 2)



(a) Learning Features – Node- & edge-level domain adaptation

Quantifying Helpfulness of Source Regions

- **Intuition:**
  - After selective source training with λ_{rs} and fine-tuning, target error is low.
- **Solution:**
  Source-target joint meta-learning
  1. **Simulate source training**: K_S steps of SGD on prediction model θ with source data X_S **and weights** λ_{rs}.
  2. **Simulate fine-tuning:** K_T steps of SGD on θ with target data X_T**.**
  3. **Optimize weights:** Compute loss on X_T. Backpropagate to θ_f, θ_w to optimize λ_{rs}



(b) Selecting Regions – Joint meta-learning

Overall Algorithm

- Source Training (Lines 2-8):
  - Train feature and weighting network to adjust weights (Lines 3-4).
  - Selectively train on source data (Line 5-6).

- Fine-tuning (Lines 9-12)
  - Agnostic to fine-tuning methods, e.g. Naïve fine-tuning, RegionTrans, etc.

Algorithm 1 Selective Cross-City Transfer Learning for Traffic Prediction with CrossTReS

Input: Source and target traffic data $\mathcal{X}_\mathcal{S}, \mathcal{X}_\mathcal{T}$,
Multi-view urban data $\{\mathcal{G}_v^C\}, v \in \{prox, road, poi, s, d\}, C \in \{\mathcal{S}, \mathcal{T}\}$,
Output: A deep prediction model $\theta_\mathcal{T}$ for $\mathcal{T}$

1: Set $s\_epoch = 0, tune\_epoch = 0$.
2: **while** $s\_epoch < T_s$ **do**
3: Train feature network θ_f via Eqn. 11.
Region Feature Learning
4: Train $\theta_s = \{\theta_f, \theta_w\}$ via Eqn. 12, 13, and 14.
Joint Meta-Learning
5: Obtain source weights $\lambda_{r_\mathcal{S}}$.
6: Train θ on source data $\mathcal{X}_\mathcal{S}$ via Eqn. 5 with weights $\lambda_{r_\mathcal{S}}$.
7: $s\_epoch = s\_epoch + 1$.
8: **end while**
9: **while** $tune\_epoch < T_{tune}$ **do**
10: Train model θ on target data $\mathcal{X}_\mathcal{T}$.
11: $tune\_epoch = tune\_epoch + 1$.
12: **end while**
13: **return** Trained model $\theta_\mathcal{T}$.

Experiments

- **Datasets**: Taxi & Bike data, pickup & dropoff
- **Source Cities:** New York (NY), Chicago (CHI)
  Target City: Washington (DC)
- **Base Model:** ST-Net [Yao et al. 2019b]
- **Data Amount:**
  - Source: 1 year; Target: 30, 7, 3 days

- **Result Highlights:**
  - Up to **8% error reduction** compared to SOTA baselines.
  - **Good compatibility** with general fine-tuning methods.
  - Source region weights λ_{r_s} provide **interpretable visualizations**.

Quantitative Results: Bike

Target Data	30 Days		7 Days		3 Days	
Method/ Source City	**NY**	**CHI**	**NY**	**CHI**	**NY**	**CHI**
ARIMA	3.44		3.46		3.48	
ST-Net	2.49		2.73		3.14	
Best Transfer	2.293	2.339	2.453	2.529	2.535	2.653
CrossTReS	2.187	2.244	2.300	2.349	2.397	2.449
CrossTReS-RT	**2.177**	**2.211**	2.315	2.315	**2.377**	2.419
CrossTReS-Mem	2.179	2.231	**2.299**	**2.313**	2.391	**2.414**

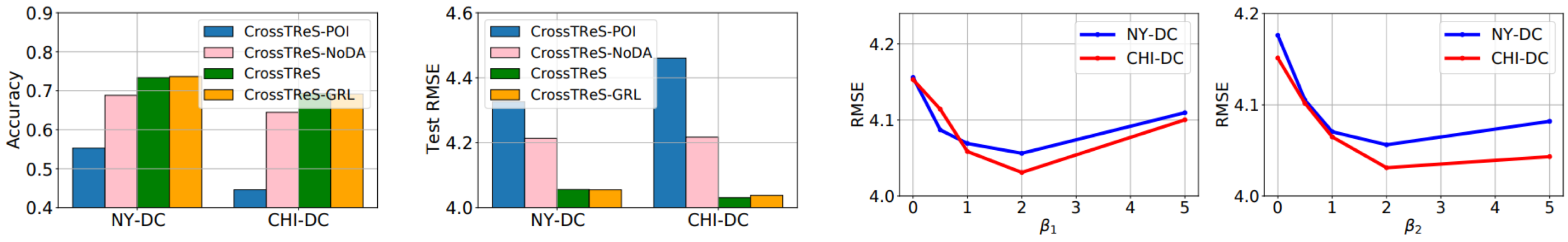
- CrossTReS-RT and –Mem use RegionTrans and STMem [Yao et al. 2019] for fine-tuning.
- **Metric:** RMSE.

Quantitative Results: Taxi

Target Data	30 Days		7 Days		3 Days	
Method/ Source City	**NY**	**CHI**	**NY**	**CHI**	**NY**	**CHI**
ARIMA	5.18		5.19		5.20	
ST-Net	4.85		5.74		6.83	
Best Transfer	4.097	4.077	4.411	4.347	4.672	4.544
CrossTReS	3.885	3.869	4.056	**4.031**	4.326	4.271
CrossTReS-RT	**3.880**	**3.867**	**4.052**	4.064	4.230	**4.235**
CrossTReS-Mem	3.883	3.873	4.053	4.048	**4.211**	4.241

- CrossTReS reduces error by **up to 8%**.
- CrossTReS is **compatible with general fine-tuning methods**, e.g. –RT, -Mem.

Model Analysis: Region Feature Learning



(a) Region Classification (b) Target Performance (c) Parameter β_1 (d) Parameter β_2

Figure 3: Analysis results on node- and edge-level domain adaptations for spatial feature learning.

- With **learned region features** and **domain adaptation**, CrossTReS achieves the best results.
- Removing either level of domain adaptation ($\beta_1 = 0$ or $\beta_2 = 0$) leads to larger error.

Model Analysis: Joint Meta-Learning

- Removing the weighting network θ_w leads to larger error.
- The additional simulation of target fine-tuning ($K_T = 1$) leads to better knowledge transfer.

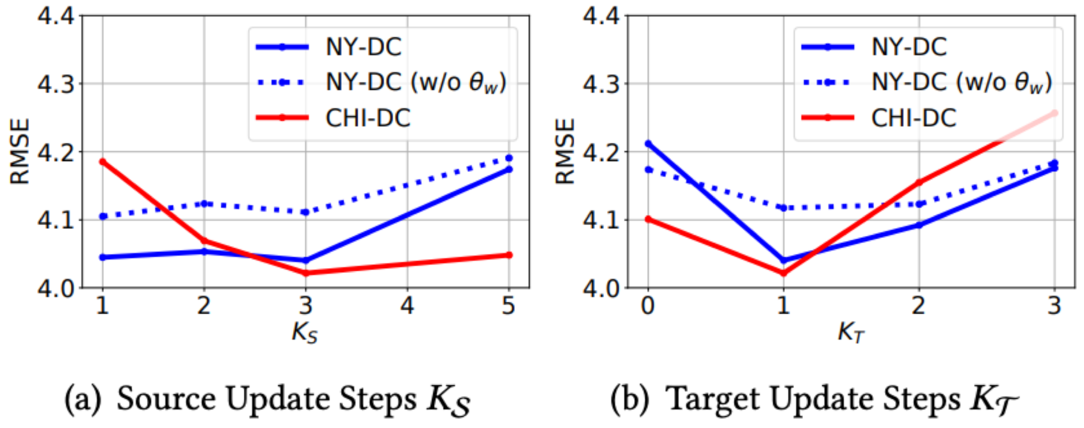
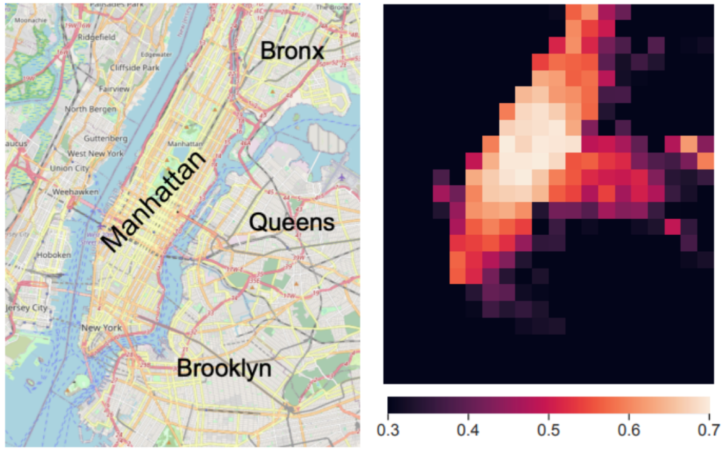


Figure 4: Analysis results on the joint meta-learning for region re-weighting.

Case Study: Visualization

- For the NY-DC transfer learning task, CrossTReS selects Manhattan over Bronx, Queens, and Brooklyn.
- Indeed, DC is most similar to Manhattan:
    - High economic development.
    - Popular tourist destinations.



(a) Map of NY (b) Visualization of λ_{r_S}

Figure 6: Visualization of source region weights λ_{r_S} over NY.

Conclusion

- CrossTReS: **selective transfer learning** for traffic prediction.
  - Selects helpful source regions to improve target fine-tuning.
  - Learns generalizable region features via **bi-level domain adaptation**.
  - Re-weights source regions via **joint meta-learning**.
  - Achieves up to **8% error reduction** and interpretable visualization on real-world data.

Reference

- Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. ICLR 2018
- Zhang, J., Zheng, Y., & Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. In Thirty-first AAAI conference on artificial intelligence.
- Yao, H., Tang, X., Wei, H., Zheng, G., & Li, Z. (2019). Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 5668-5675).
- Wang, L., Geng, X., Ma, X., Liu, F., & Yang, Q. (2019). Cross-city transfer learning for deep spatio-temporal prediction. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (pp. 1893-1899).
- Yao, H., Liu, Y., Wei, Y., Tang, X., & Li, Z. (2019). Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In The World Wide Web Conference (pp. 2181-2191).
- Zhang, M., Li, T., Li, Y., & Hui, P. (2020). Multi-view joint graph representation learning for urban region embedding. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (pp. 4431-4437).
- Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning (pp. 1126-1135). PMLR.

Thanks
Q & A

Project: https://turing.ust.hk/

Author: https://kl4805.github.io/
yilun.jin@connect.ust.hk