Graph Neural Networks for Road Safety Modeling: Datasets and Evaluations for Accident Analysis

Abhinav Nippani‡, **Dongyue Li**‡, Haotian Ju, Haris N. Koutsopoulos, Hongyang R. Zhang

Email addresses: {nippani.a, li.dongyu, ju.h, h.koutsopoulos, ho.zhang}@northeastern.edu

A Unified Dataset of Traffic Accidents

We construct a unified dataset of **9 million** accident records spanning **eight states of US**, the longest spanning **twenty years**.

Accident Records from official reports of Department of Transportation.

- Each accident is associated with a road where the accident happened and the timestamp during which the accident happened.

	Start	End	Crash Rate	# Nodes	# Roads	# Accidents/Month
Delaware	2009	2022	3.27	49,023	116,196	26,725
Iowa	2013	2022	4.92	253,623	707,072	49,495
Illinois	2012	2021	36.7	627,661	1,647,614	230,666
Massachusetts	2002	2022	24.48	285,942	706,402	70,640
Maryland	2015	2022	11.44	250,565	580,526	87,079
Minnesota	2015	2023	5.39	383,086	979,259	48,963
Montana	2016	2020	1.69	145,525	351,516	17,576
Nevada	2016	2020	5.42	121,392	292,674	35,121

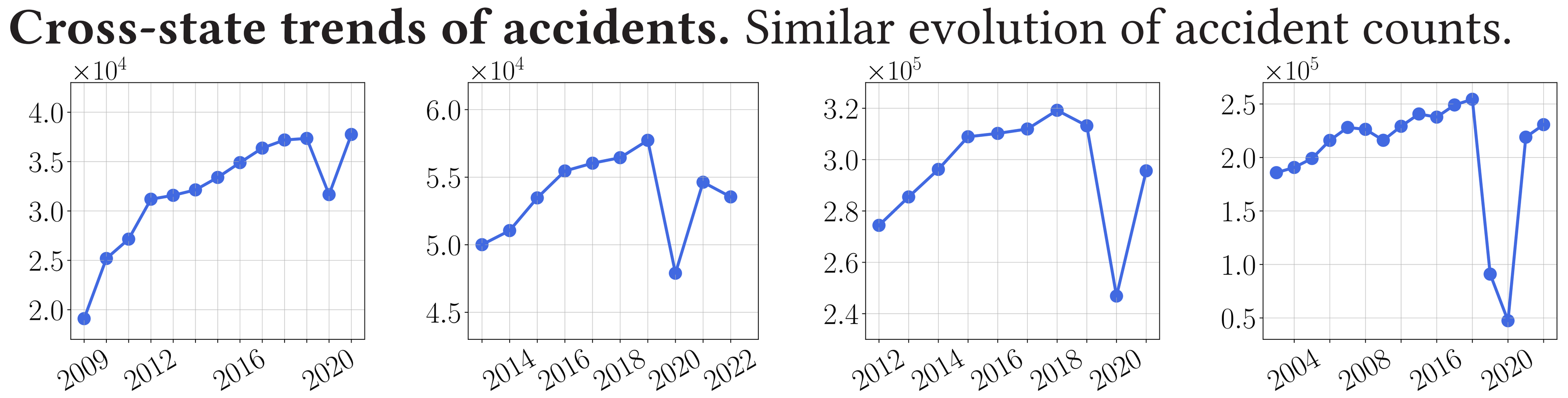
	d_{avg}	d_{max}	Centrality (10^{-3})	Avg Length (m)	Volume (%)
Delaware	2.4	6	5.7	213	3.14
Iowa	2.8	7	1.4	532	-
Illinois	2.6	8	0.8	307	-
Massachusetts	2.5	8	0.9	188	1.34
Maryland	2.3	8	1.0	211	1.76
Minnesota	2.5	8	0.6	474	-
Montana	2.4	7	0.02	859	-
Nevada	2.4	6	0.4	280	1.38

Road Features: We generate road networks from OpenStreetMap

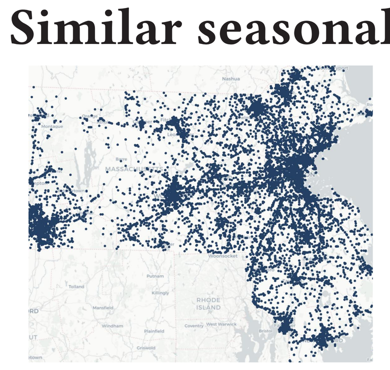
- Road-level static features: road category and length information.
- Node-level static features: in- & out-degrees, and betweenness centrality.
- Road-level temporal features: annual average daily traffic (AADT).
- Node-level temporal features: daily weather information including temperature, rainfall, wind speed, etc.

Observations

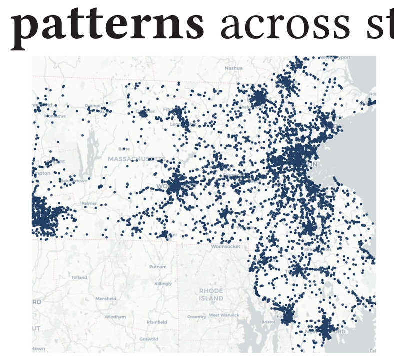
Cross-state trends of accidents. Similar evolution of accident counts.



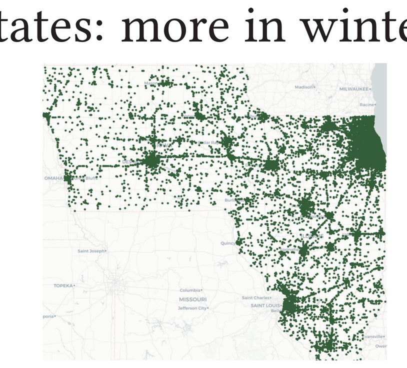
Similar seasonal patterns across states: more in winter than in spring.



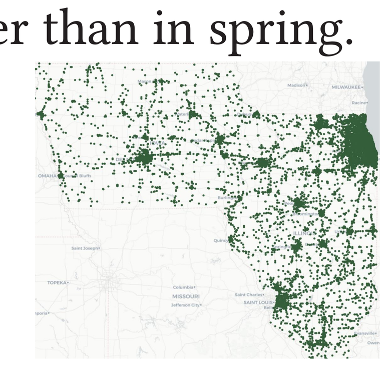
MA, Winter



MA, Spring

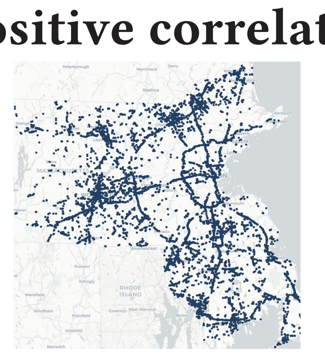


IA & IL, Winter

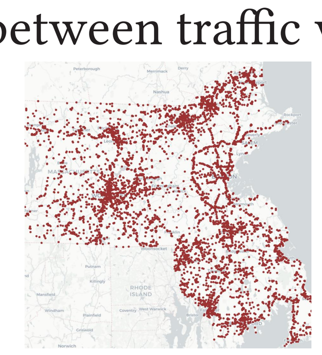


IA & IL, Spring

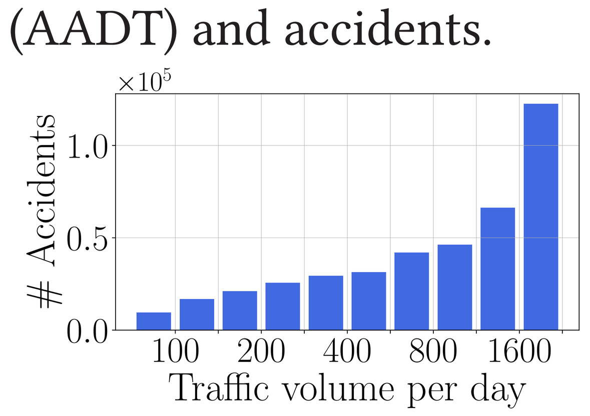
Positive correlation between traffic volume (AADT) and accidents.



MA, Accidents



MA, Traffic



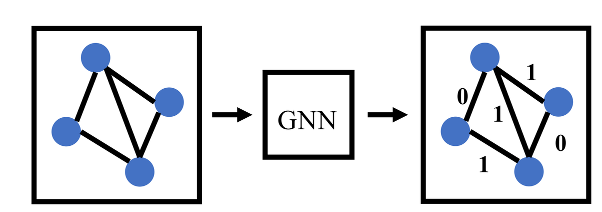
Problem Statement

Predicting road accidents as **an edge-level prediction problem**.

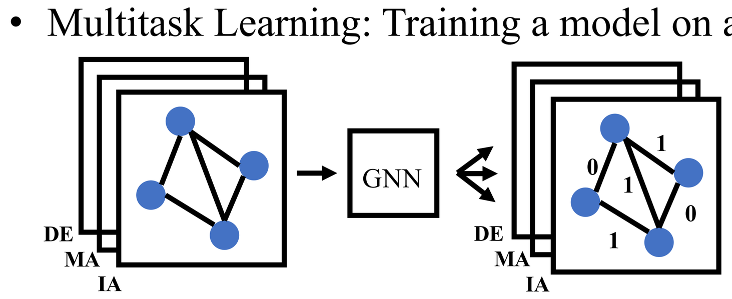
- Model a road network as a directed graph $G = (V, E)$
- Train a model with accidents up to a certain period (e.g., month)
- Evaluate the model to predict accidents for the remaining periods

Baselines:

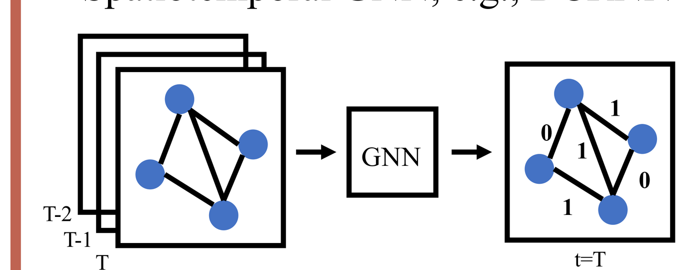
- Graph Neural Network, e.g., GCN



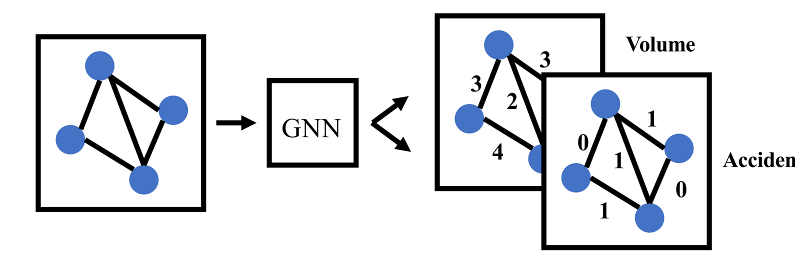
- Multitask Learning: Training a model on all states



- Spatiotemporal GNN, e.g., DCRNN



- Transfer Learning: Predicting volume and accident



Two evaluations:

- **Regression**: predicting the number of accidents per edge, evaluated by Mean Absolute Error
- **Classification**: predicting if one (or more) accidents will occur on a road, evaluated by AUROC score.

Experimental Results

Graph neural networks can accurately predict accident labels.

- GNNs predict accident counts with **0.3 MAE**, 22% relative to the absolute accident count and predict whether an accident occurs on a road with **87% AUROC score** on average.
- Multitask learning on the combined data of all states improves learning on a single state data by relatively **8.4% for MAE** and **0.9% for AUROC**.
- Transfer learning (training jointly with traffic volume prediction) brings a relative improvement of **7.9% for MAE** and **1.1% for AUROC**.

MAE (↓)	DE	MA	MD	AUROC (↑)	DE	MA	MD
Avg Count	1.23	2.27	1.22	Positive Rate	0.23	0.10	0.15
GraphSAGE	0.3±0.01	0.8±0.02	0.4±0.01	GraphSAGE	87.6±0.1	81.8±0.1	87.6±0.1
DCRNN	0.3±0.01	0.9±0.06	0.3±0.01	DCRNN	81.2±1.2 0	70.5±0.1	84.5±0.3
MTL	**0.2**±0.01	0.7±0.01	**0.3**±0.00	MTL	**87.8**±0.3	81.9±0.3	**88.1±0.1**
TL	0.2±0.01	**0.6**±0.02	0.3±0.01	TL	87.3±0.2	**82.6**±0.2	87.9±0.4

Key observations:

- MTL on two states mostly improves over learning on a single state data.
- Comparing GraphSAGE with spatiotemporal GNNs: none of them dominating each other. The reason may be low labeling rates ($\leq 0.25\%$).
- Removing graph-structural features reduces the performance by 6.9%.

Target \ Source	DE	IA	IL	MA	MD	MN	MT	NV
DE	0.0	0.2	0.6	0.4	0.2	0.0	0.0	0.3
IA	0.9	0.0	1.2	1.0	0.6	0.8	0.8	2.0
IL	0.6	-0.1	0.0	0.4	0.3	-0.1	0.1	0.1
MA	0.9	0.3	0.7	0.0	-0.5	-0.3	0.0	0.3
MD	0.3	0.1	0.6	0.7	0.0	0.1	0.3	0.3
MN	3.0	3.0	3.5	2.9	3.4	0.0	3.0	3.1
MT	0.1	0.2	-0.2	0.0	0.1	0.3	0.0	0.2
NV	1.0	-4.4	-3.9	1.2	1.1	0.5	0.2	0.0

Pairwise MTL vs STL

ML4RoadSafety Package

Traffic Accident Dataset: Automatic data loading

```
>>> from ml_for_road_safety import TrafficAccidentDataset
# Dataset as PyTorch Geometric dataset object
>>> dataset = TrafficAccidentDataset(state_name = "MA")
# Loading the accident records of a particular month
>>> data = dataset.load_monthly_data(year = 2022, month = 1)
# The edges of accidents and accident counts
>>> data["accidents"], data["accident_counts"]
# Edge list, node features, and edge features
>>> data["edge_index"], data["x"], data["edge_attr"]
```

Trainer: Easy training and evaluation of graph neural networks

```
>>> from ml_for_road_safety import Trainer, Evaluator
# Creating the dataset
>>> dataset = TrafficAccidentDataset(state_name = "MA")
# Create a GNN model, e.g., GCN
>>> model = GNN(encoder = "gcn", **kwargs)
# Get an evaluator for the classification task.
>>> evaluator = Evaluator(type = "classification")
# Initialize a trainer with a GNN, dataset, and evaluator
>>> trainer = Trainer(model, dataset, evaluator, ...)
# Conduct training and evaluation inside the trainer
>>> log = trainer.train()
```

Multitask and transfer learning: Capturing cross-sectional trends

```
# Create a trainer for every task
>>> self.task_to_trainers = {}
>>> for task_name in tasks:
>>>     self.task_to_trainers[task_name] = Trainer(...)
# Optimize the average loss of all tasks.
>>> for epoch in range(1, 1 + epochs):
>>>     for task_name in task_list:
#           Each task trainer optimizes the loss of one task
>>>         task_trainer = self.task_to_trainers[task_name]
>>>         task_trainer.train_epoch()
```

Including advanced training techniques:

- Sharpness-aware minimization
- Graph contrastive learning
- Task affinity grouping
- Spatiotemporal graph neural networks: GCRNN, STGCN, etc.

Conclusion

- We construct a large-scale, unified dataset of over 9M traffic accident records from eight states, accompanied by road network features.
- Existing graph neural networks can accurately predict both the number of accidents on roads and whether an accident will occur or not.
- Improved results are achieved by multitask learning across states and transfer learning that combines traffic volume with accident prediction.

Lab website: virtuosoresearch.github.io

Code: github.com/VirtuosoResearch/ML4RoadSafety

Paper: https://arxiv.org/abs/2311.00164