Deep Learning based Scalable Inference of Uncertain Opinions

University at Albany - SUNY, Virginia Tech

Xujiang Zhao
PhD Student
Department of Computer Science, University at Albany - SUNY

Feng Chen
Assistant Professor
Department of Computer Science, University at Albany - SUNY

Jin-Hee Cho
Associate Professor
Department of Computer Science, Virginia Tech

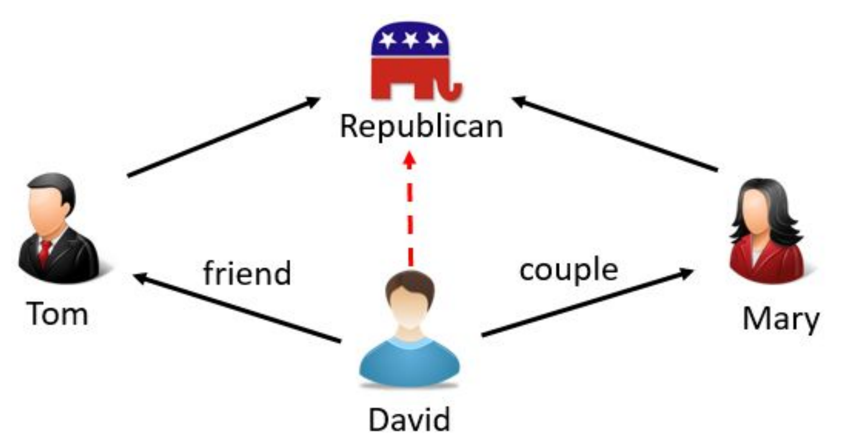
I. Research Goal

Motivation: Decision making with subjective, uncertain opinions is an important and challenging problem. Subjective Logic (SL) is one of well-known belief models explicitly dealing with uncertain opinions. However, SL is not scalable for a large-scale network data and incapable to handle heterogeneous opinions.

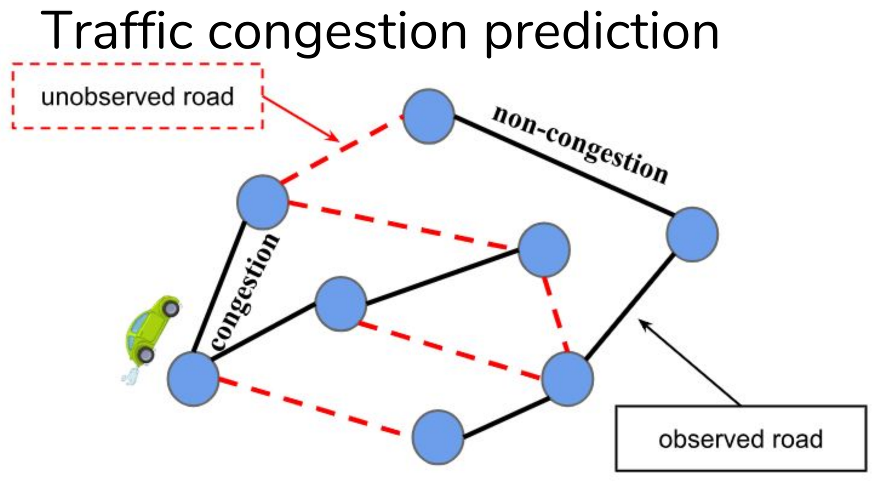
Goal: Develop a DL-based opinion inference model handles node-level opinions explicitly in a large-scale network using *graph convolutional network* (GCN) and *variational autoencoder* (VAE) techniques.

II. Application

- Trust in social network



- Traffic congestion prediction



- Opinion diffusion
- Graph summarization

III. Problem Formulation

Problem 1 (Uncertainty-based opinion inference in network data): *Let us define the following notations:*

- *Let* $\mathbb{G} = (\mathbb{V}, \mathbb{E}, y)$ *be an input network as defined above.*
- *Let* $\omega_i = (b_i, d_i, u_i)$ *be node* i*'s subjective opinion of variable* y_i *where node* $i \in \mathbb{V}$*. Let* $\mathbb{L} \subseteq \mathbb{V}$ *be a subset of edges whose opinions are denoted by* $\boldsymbol{\omega}_{\mathbb{L}} = [\omega_i]_{i \in \mathbb{L}}$*.*

Given

- $\mathbb{G} = (\mathbb{V}, \mathbb{E}, y)$*, an input network;*
- $\boldsymbol{\omega}_{\mathbb{L}} = [\omega_i]_{e_i \in \mathbb{L}}$*, a vector of subjective opinions on* $\{y_i\}_{i \in \mathbb{L}}$*.*

Predict $\boldsymbol{\omega}_{\mathbb{V}\setminus\mathbb{L}} = [\omega_i]_{i \in \mathbb{V}\setminus\mathbb{L}}$*, unknown opinions on* $\{y_i\}_{i \in \mathbb{V}\setminus\mathbb{L}}$

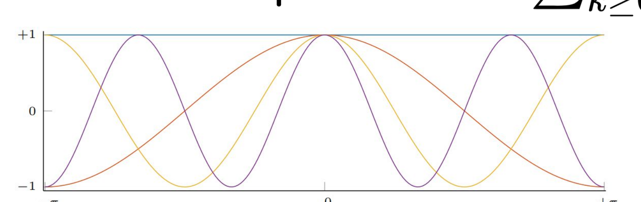
Question?

How can we accurately and efficiently predict unknown opinions with a large, heterogeneous, uncertain network data?

IV. Graph Convolutional

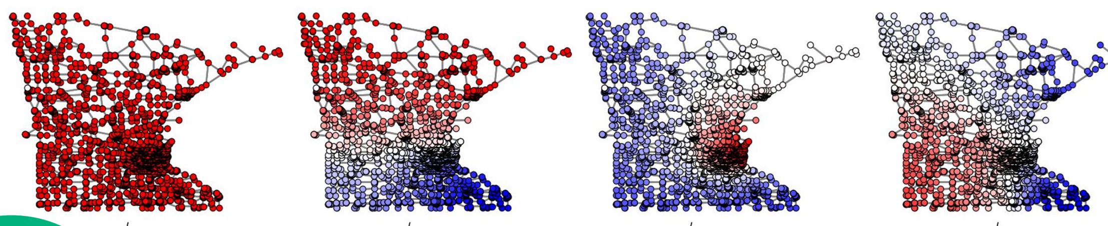
- ***Graph Fourier Transform***

Euclidean spaces: $\mathbf{r} = \sum_{k \geq 0} \hat{\mathbf{r}}_k e^{ik}$



Graph Laplacian $L = \Phi \Lambda \Phi^T$

non-Euclidean spaces: $\mathbf{r} = \sum_{k \geq 0} \hat{\mathbf{r}}_k \phi_k = \phi^T \phi \mathbf{r}$



ϕ_1 ϕ_2 ϕ_3 ϕ_4

- Graph Convolution in Fourier domain

$$\mathbf{r} \star \mathbf{b} = \Phi^T(\Phi^T \mathbf{r}) \circ (\Phi^T \mathbf{b}) = \Phi diag(\hat{r}_1, \cdots, \hat{r}_n)\hat{\mathbf{b}}$$

element-wise product

- Graph convolutional layer: $g_\theta \star \mathbf{r} = \Phi g_\theta \Phi^T \mathbf{r}$

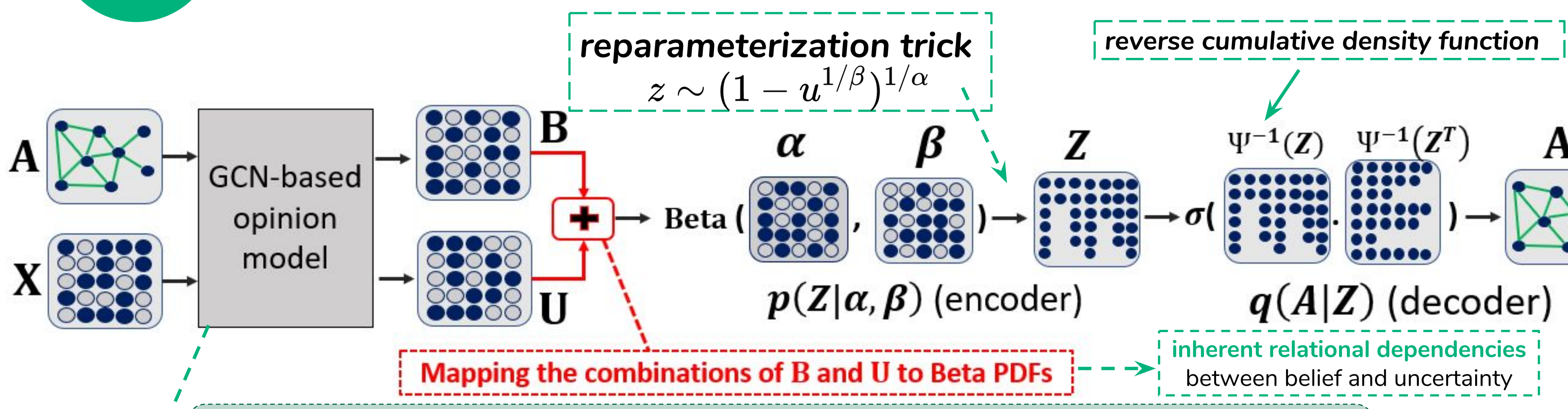
computationally **expensive** of Φ is $O(n^2)$

- Chebyshev polynomials: $g_\theta(\Lambda) \approx \sum_{k=1}^{K} \theta_k T_k(\tilde{\Lambda})$

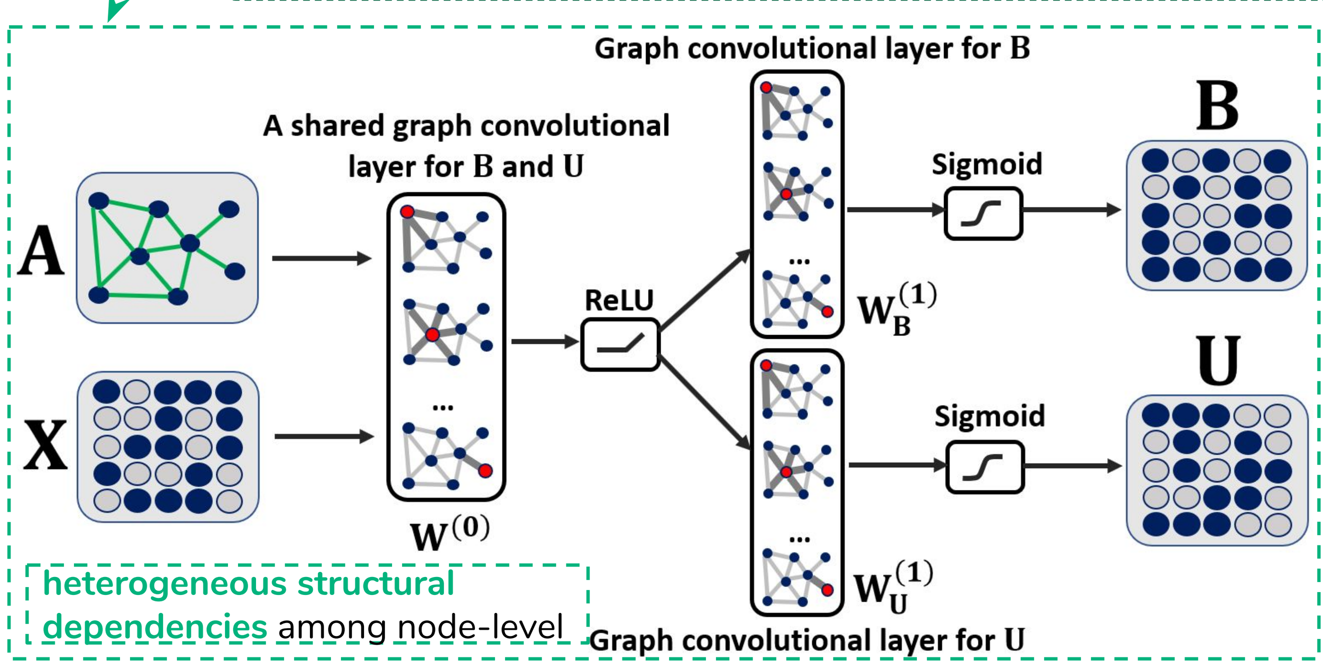
$$T_k(r) = 2xT_{k-1}(r) - T_{k-2}(r), T_0(r) = 1, T_1(r) = r$$

$$g_\theta \star \mathbf{r} \approx \sum_{k=1}^{K} \theta_k T_k(\tilde{L})\mathbf{r}$$

V. Our Solution: GCN-AVE-opinion



$$\min_{\theta} \lambda \tilde{\mathcal{L}}(\theta) + \mathcal{L}_{\mathbf{B}}(\theta) + \mathcal{L}_{\mathbf{U}}(\theta), \ \tilde{\mathcal{L}}(\theta) = -\frac{1}{K}\sum_{n=1}^{K}\left[\log p(\mathbf{A}|\mathbf{Z}^{(n)})\right] + \mathrm{KL}\left[q(\mathbf{Z}|\mathbf{X}, \mathbf{A}) \| p(\mathbf{Z})\right]$$



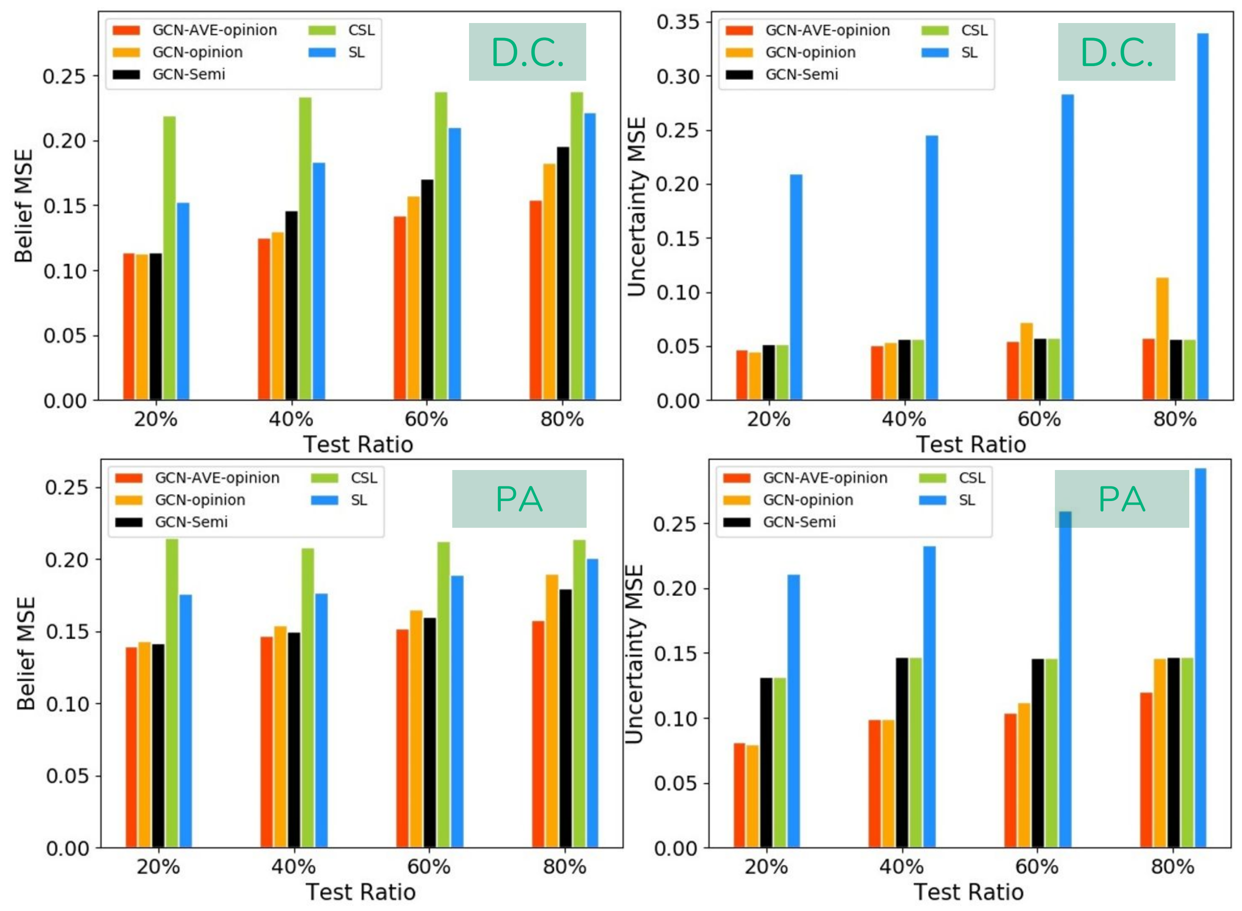
Algorithm 1: GCN-AVE based Opinion Prediction

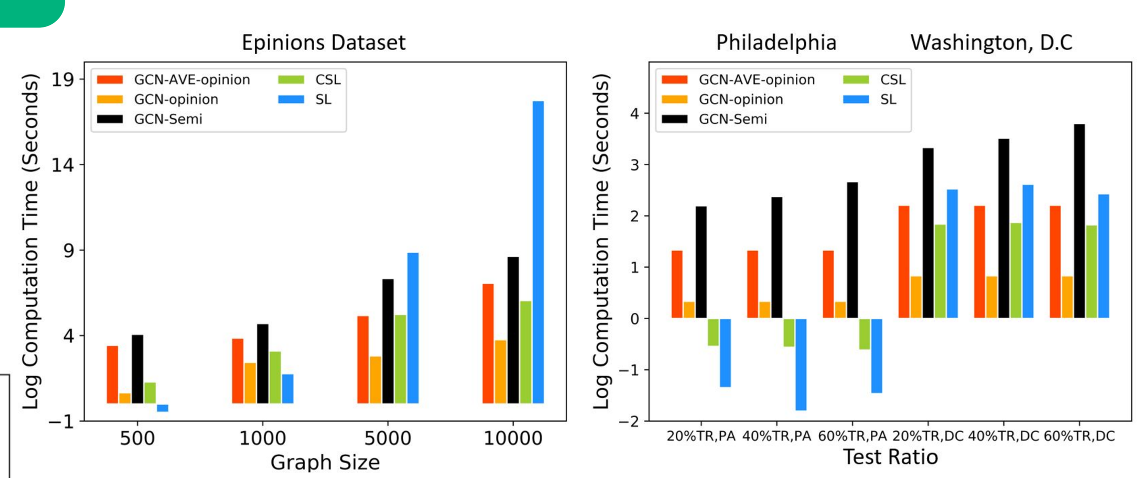
Input: $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbf{A}, y)$ and $\{\omega_i\}_{i \in \mathbb{L}}$
Output: $\{\omega_i\}_{i \in \mathbb{V}\setminus\mathbb{L}}$

```
ℓ = 1;
K = 16; (Set the mini-batch size)
η = 0.001; (Set the learning rate)
Estimate the initial θ^(ℓ) by solving Problem 1;
repeat
    Sample {u_{i,j}^(n)} ~ Unif(0,1), for i = 1,···,N,
      j = 1,···,P, and n = 1,···,K;
    Forward pass to compute B, U, Calculate {Z^(n)}_{n=1}^K
    Backward pass via the chain-rule for gradient
      g^(ℓ) = ∇_θ[L̃(θ^(ℓ)) + L_B(θ^(ℓ)) + L_U(θ^(ℓ))]
    Update parameters using step size η via
      θ^(ℓ+1) = θ^(ℓ) − η · g^(ℓ)
    ℓ = ℓ + 1;
until convergence
[B, U] = f(X, A; θ^(ℓ+1))
Calculate {ω_i}_{i∈V\L} based on B and U, return {ω_i}_{i∈V\L}
```

VI. Experimental Results

Dataset	# nodes	# edges	# of snapshots
Epinions	47,676	477,468	-
Washington, D.C	1,383	1,878	3440
Philadelphia	603	708	3440





B-MSE: **GCN-AVE** > GCN > GCN-Semi > SL > CSL

U-MSE: **GCN-AVE** > GCN > GCN-Semi > CSL > SL

Time Cost: **GCN** < CSL< GCN-AVE < GCN-Semi < SL

Conclusions

- ★ GCN-AVE method **outperforms** among all in both B-MSE and U-MSE
- ★ GCN-AVE method shows **less sensitivity** over a wide range of test ratios
- ★ GCN-AVE scales almost **linearly** in proportion to the network size