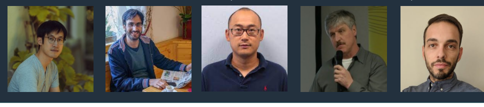
SEIGN: A Simple and Efficient Graph Neural Network for Large Dynamic Graphs

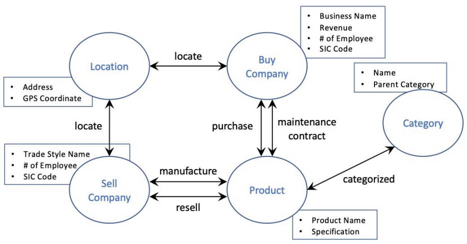
Xiao Qin[1], Nasrullah Sheikh[2], Chuan Lei[1], Berthold Reinwald[2] and Giacomo Domeniconi[3]
AWS AI Research and Education[1], IBM Research – Almaden[2], U.S. Bank[3]

ICDE 2023
Anaheim, CA April 3 – 7, 2023

amazon | science



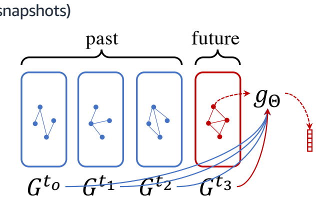
Motivating Example



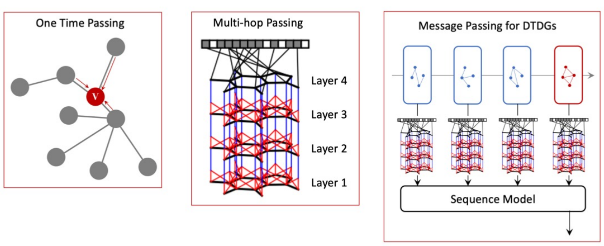
Sheikh, Nasrullah, et al. Distributed Training of Knowledge Graph Embedding Models using Ray. EDBT 2022.

Discrete Time Dynamic Graphs (DTDGs)

- **Discrete-time Dynamic Graphs (DTDGs)**
  - $G^T = (G^{t_0}, \ldots, G^{t_n})$ # a sequence of static graphs (snapshots)
  - $G^{t_i} = \{V^{t_i}, E^{t_i}, X^{t_i}\}$ # vertex, edge and feature sets
  - $V^{t_i} = \{v_1, \ldots v_{|V^{t_i}|}\}$ # a set of nodes
  - $E^{t_i} = \{e_1, \ldots, e_{|E^{t_i}|}\}$ # a set of edges
  - $X^{t_i} = \{x_1, \ldots, x_{|V^{t_i}|}\}$ # a set of node features
- **Problem**
  Learn the future node representation
  -- predicting the future based on the past.



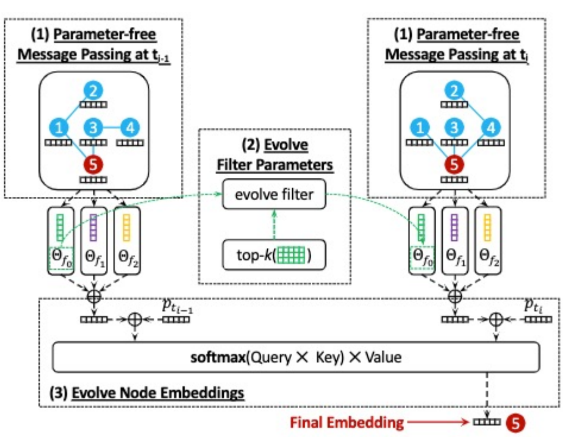
Existing Message Passing-based Solutions



Question 1. How can the model be efficiently trained on large-scale DTDGs to exploit hardware accelerators with small memory footprint?

Question 2. How can the model effectively capture the changing dynamics of the graphs?

Our Approach – Scalable Evolving Inception Graph Neural Network (SEIGN)



Parameter-free Message Passing

$$\mathcal{Z}^{t_i} = \xi\left(\sigma\left(\bigoplus_{j=0}^{k} \mathcal{A}_j^{t_i}\mathcal{X}^{t_i}\Theta_{f_j}^{t_i}\right)\Theta_o\right)$$

$\mathcal{A}_j^{t_i}$ is a linear diffusion operator.

$\mathcal{A}_j^{t_i}\mathcal{X}^{t_i}$ computes the node aggregated messages for j^{th} hop. The computation does not involve learnable parameters. Therefore, such a message passing can be done in a preprocessing step.

Benefit: 1) less node dependencies 2) less parameters to train

The embeddings $\mathcal{Z}^{t_i}$ for a snapshot are generated based on a set of graph diffusion operations covering multiple neighborhood sizes, analogous to the popular Inception network.

Benefit: better embedding expressivity

Evolving Parameters over Time

$\Theta_{f_j}^{t_i}$ is a learnable matrix that transforms the node aggregated messages for j^{th} hop. Sharing these parameters across time results in poor performance. We propose to use a sequence model to regulate them over time.

Benefit: better embedding expressivity

Evolving Node Embedding over Time

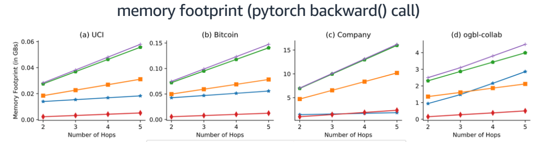
$\mathcal{Z}_v = \{z_v^{t_0}+p_{t_0},\cdots,z_v^{t_{|\mathcal{T}|}}+p_{t_{|\mathcal{T}|}}\}$ is the concatenated node and positional embeddings for node v over time. We propose to use the self-attention model to generate the final embedding for v from its embeddings over time.

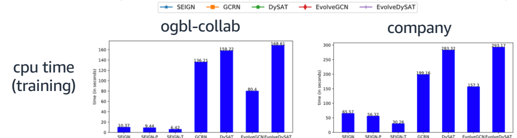
Benefit: better embedding expressivity

Experimental Evaluation

Methods	UCI			Bitcoin			Company			ogbl-collab		
	AUC↑	AP↑	F1↑(0.5)	AUC↑	AP↑	F1↑(0.5)	AUC↑	AP↑	F1↑(0.5)	AUC↑	AP↑	F1↑(0.5)
GCN	74.35±1.8	74.96±1.2	68.90±0.9	83.58±1.2	84.68±1.1	77.86±0.8	88.75±0.3	85.15±0.4	82.14±0.3	93.45±0.1	93.29±0.0	87.98±0.1
GAT	68.62±3.2	67.75±3.0	61.93±2.8	84.25±0.6	83.62±0.6	79.80±0.4	84.10±1.2	81.84±0.9	77.82±1.0	92.79±0.3	92.88±0.3	88.59±0.2
SIGN	74.98±0.3	69.92±0.4	74.54±0.2	87.03±0.1	85.70±0.1	**84.89**±0.2	92.72±0.1	89.11±0.1	86.22±0.2	97.07±0.0	96.10±0.1	82.80±0.1
GCRN	77.21±0.7	75.28±0.8	74.19±0.7	87.81±0.2	88.02±0.2	80.21±0.2	91.65±0.2	87.05±0.2	84.71±0.2	96.47±0.0	96.18±0.1	84.54±0.2
DySAT	77.46±0.6	76.54±0.6	73.06±0.7	88.82±0.1	88.19±0.2	82.01±0.2	92.45±0.0	88.07±0.1	86.10±0.0	95.18±0.2	95.05±0.3	87.34±0.1
EvolveGCN	76.09±1.0	75.65±0.9	71.07±0.9	85.53±0.4	84.82±0.5	79.23±0.3	89.22±0.1	86.40±0.2	79.84±0.1	96.06±0.2	95.61±0.2	**91.01**±0.3
EvolveDySAT	80.49±0.6	78.08±0.7	74.02±0.4	89.38±0.1	90.12±0.0	76.32±0.1	92.78±0.0	88.78±0.0	86.25±0.1	96.09±0.0	96.13±0.1	87.40±0.0
SEIGN-P	79.82±0.5	78.55±0.4	**75.44**±0.4	88.44±0.1	87.04±0.1	83.61±0.2	93.35±0.1	89.25±0.0	87.35±0.0	97.50±0.1	97.34±0.1	86.78±0.2
SEIGN-T	76.86±0.3	72.49±0.2	73.08±0.2	88.51±0.2	86.56±0.2	84.41±0.1	93.26±0.2	88.28±0.2	86.86±0.1	97.35±0.1	96.62±0.1	82.20±0.2
SEIGN	**80.83**±0.3	**81.36**±0.2	71.81±0.2	**90.12**±0.0	**91.13**±0.1	81.89±0.1	**94.34**±0.1	**90.09**±0.1	**88.39**±0.1	**98.33**±0.1	**98.49**±0.0	86.85±0.1

memory footprint (pytorch backward() call)





Conclusion

- We introduce a simple and efficient representation learning method for DTDGs.
- We propose to simultaneously evolve the graph model parameters and the representations from each graph snapshot to effectively capture the changing dynamics of the DTDGs.
- We further introduce an evolving graph inception architecture in SEIGN for DTDGs which enables efficient graph mini-batch construction and incurs a considerably lower training memory footprint.
- We evaluate the effectiveness of SEIGN in a position study and an ablation study. The position study compares our method against state-of-the-art approaches for an unsupervised training task on publicly available benchmarks as well as a real industrial dataset. We also demonstrate the efficiency of training SEIGN by providing various compute and memory profiles.

*Corresponding author: drxqin@amazon.com