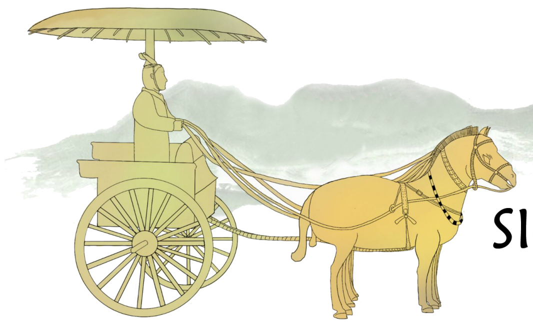


Xi'an, Shaanxi, China
SIGMOD/PODS 2021

LightNE: A Lightweight Graph Processing System for Network Embedding

Jiezhong Qiu, Laxman Dhulipala, Jie Tang, Richard Peng, Chi Wang

https://github.com/xptree/LightNE.

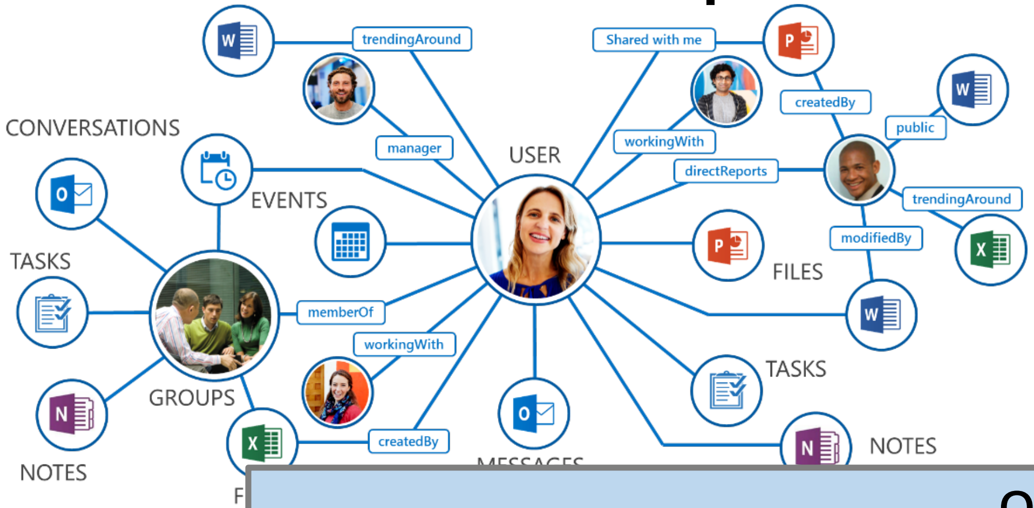
Roadmap

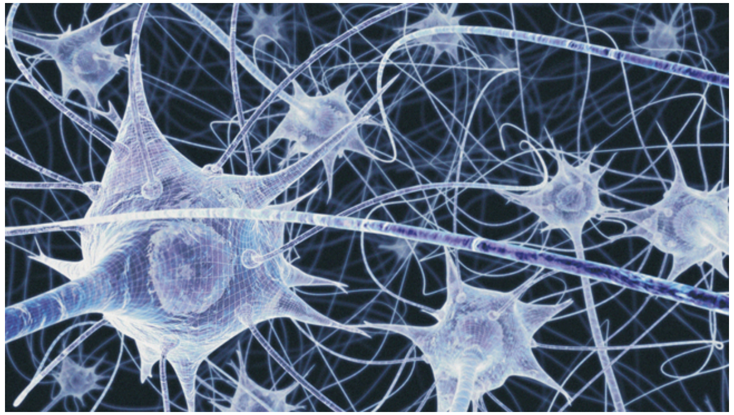
- **Introduction to Network Embedding**

- LightNE: Co-design of Algorithm and System

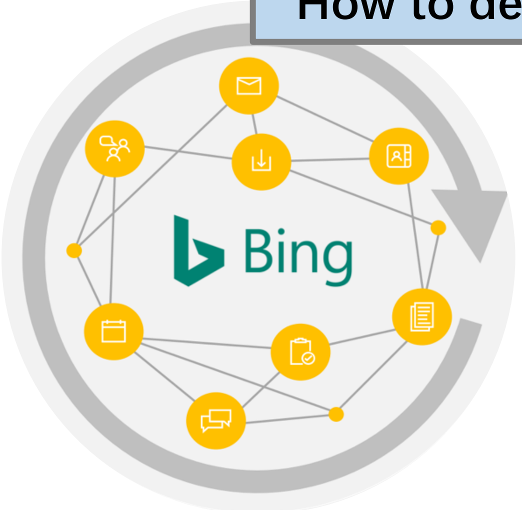
- Experiments on graphs with billions of edges.

Real-world Graphs

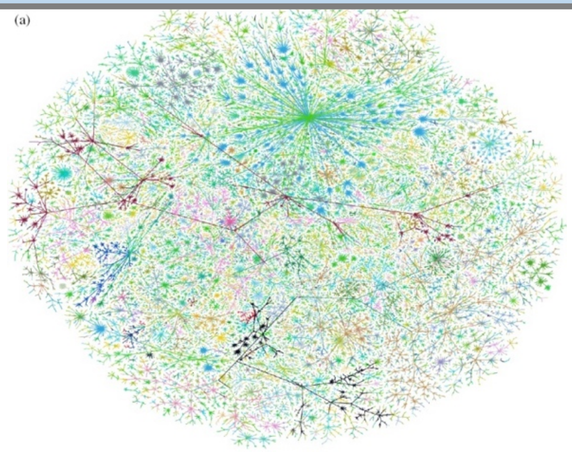




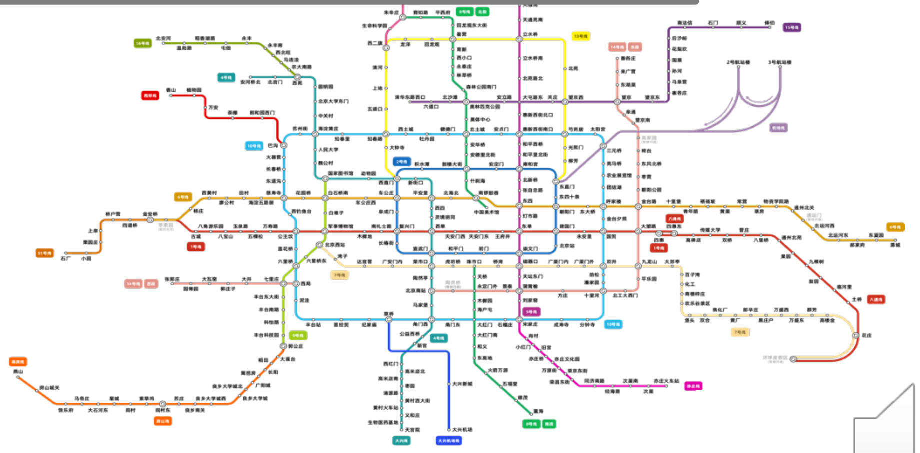
Question:
How to design machine learning models for large-scale real-world graphs?



Knowledge Graph



Internet Graph

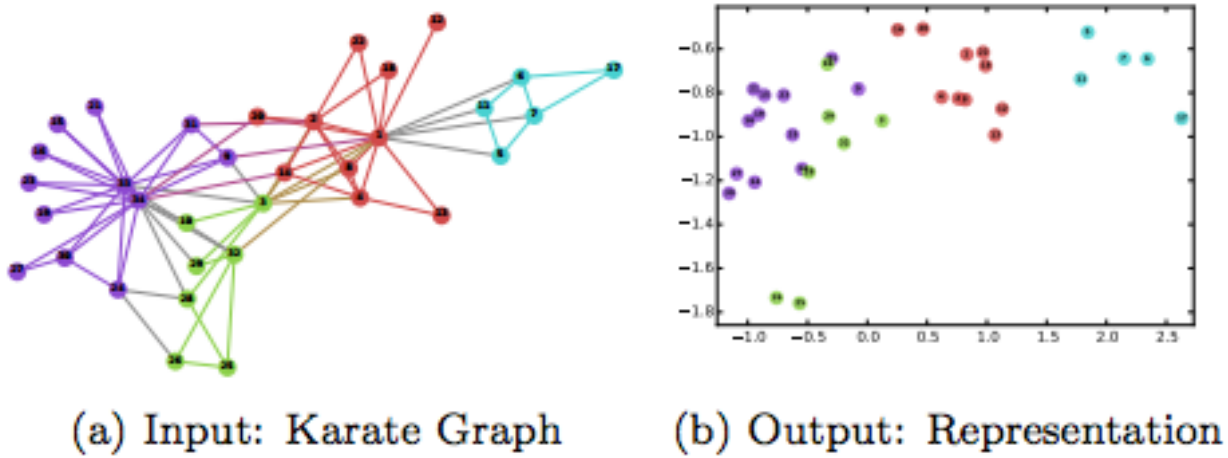


Transportation Graph

figure credit: Web

Background: Network Embedding

- Given a graph $G = (V, E)$, aim to learn a function $f\colon V \rightarrow R^d$ to capture neighborhood similarity and community membership.

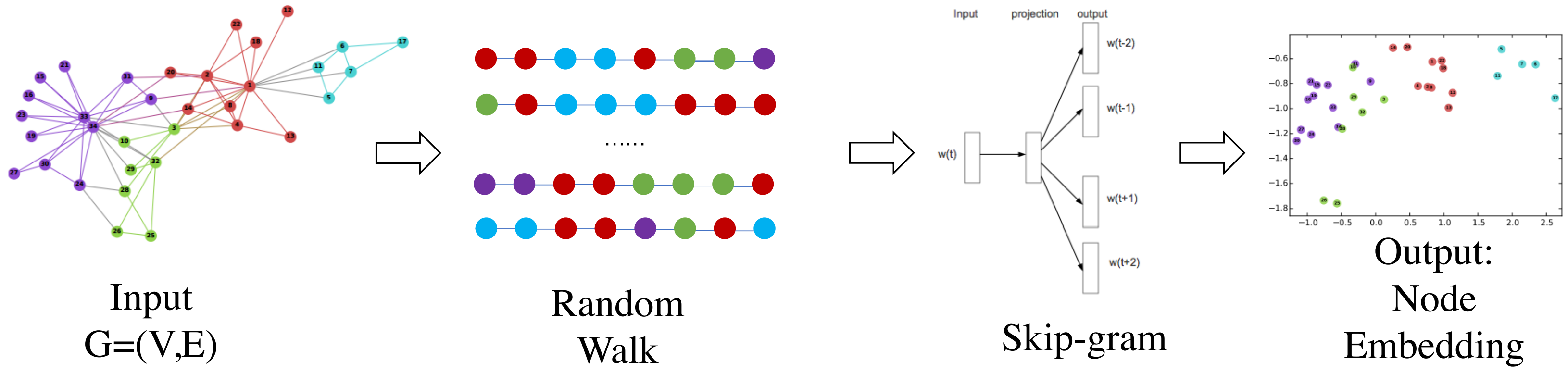


A toy example from DeepWalk [1]

[1] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In KDD ' 14. ACM, 701–710.

Background: DeepWalk

- Sampling random walk sequences on the input graph
- Train a skip-graph model (word2vec) on the sampled sequences



- Scalability issue:
  - Alibaba embeds a 600-billion-node commodity graph by first partitioning it into 50-million-node subgraphs, and then embedding each subgraph separately with 100 GPUs running DeepWalk [1]

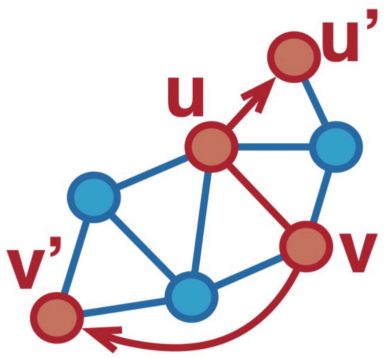
[1] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In KDD ' 18. 839–848.

Background: Network Embedding as Matrix Factorization

- NetMF[1]: DeepWalk is implicitly and asymptotically factorizing:

$$\boldsymbol{M} \triangleq \mathrm{trunc\_log}^{\circ}\left(\frac{\mathrm{vol}(G)}{b}\frac{1}{T}\sum_{r=1}^{T}(\boldsymbol{D}^{-1}\boldsymbol{A})^{r}\boldsymbol{D}^{-1}\right)$$

- NetSMF[2]:
  - Sparisify r-step random walk matrix $(D^{-1}A)^r$ with PathSampling.
  - Need $O(mlogn)$ samples



Algorithm 1: PathSampling.

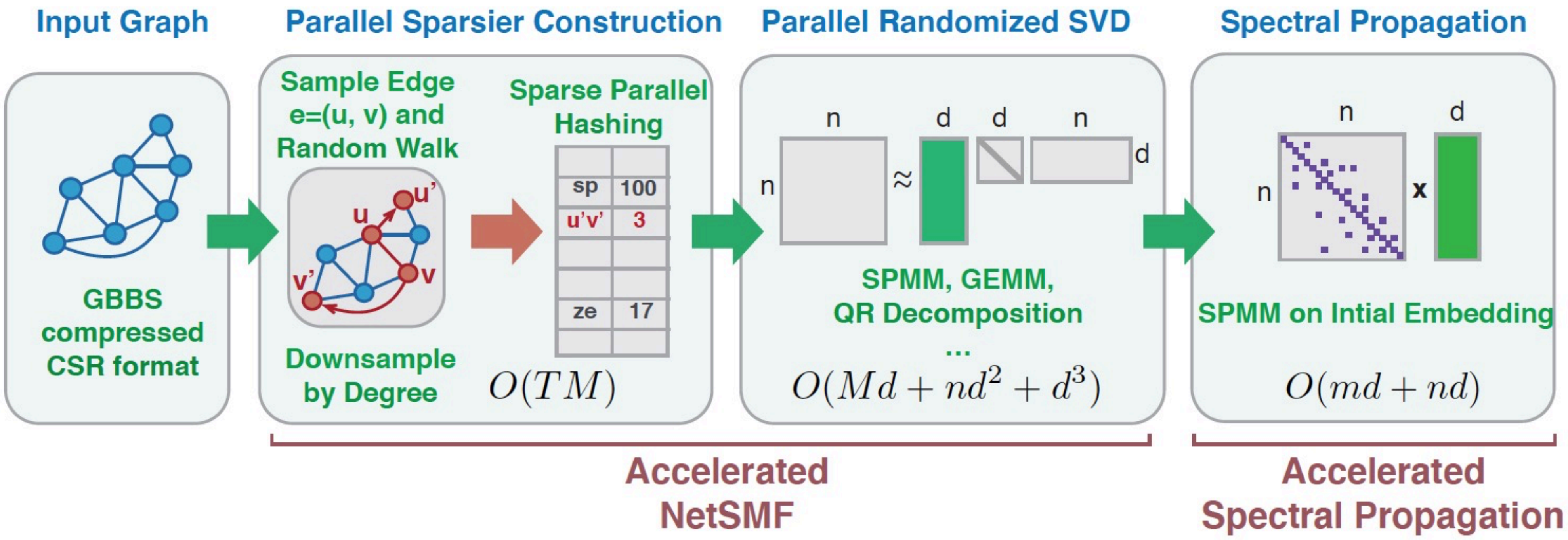
Procedure PathSample(G, u, v, r)
  Let a random edge (u, v) be given.
  Sample a random number s uniformly in $[0, r-1]$.
  $u' \leftarrow$ random walk u for s steps on graph G
  $v' \leftarrow$ random walk v for $r-1-s$ steps on graph G.
  return edge (u', v')

[1] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network embedding as matrix factorization: Unifying deepw k, line, pte, and node2vec. In WSDM ' 18.
[2] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Chi Wang, Kuansan Wang, and Jie Tang. 2019. Netsmf: Large-scale network embedding as sparse matrix factorization. WWW' 19.

Roadmap

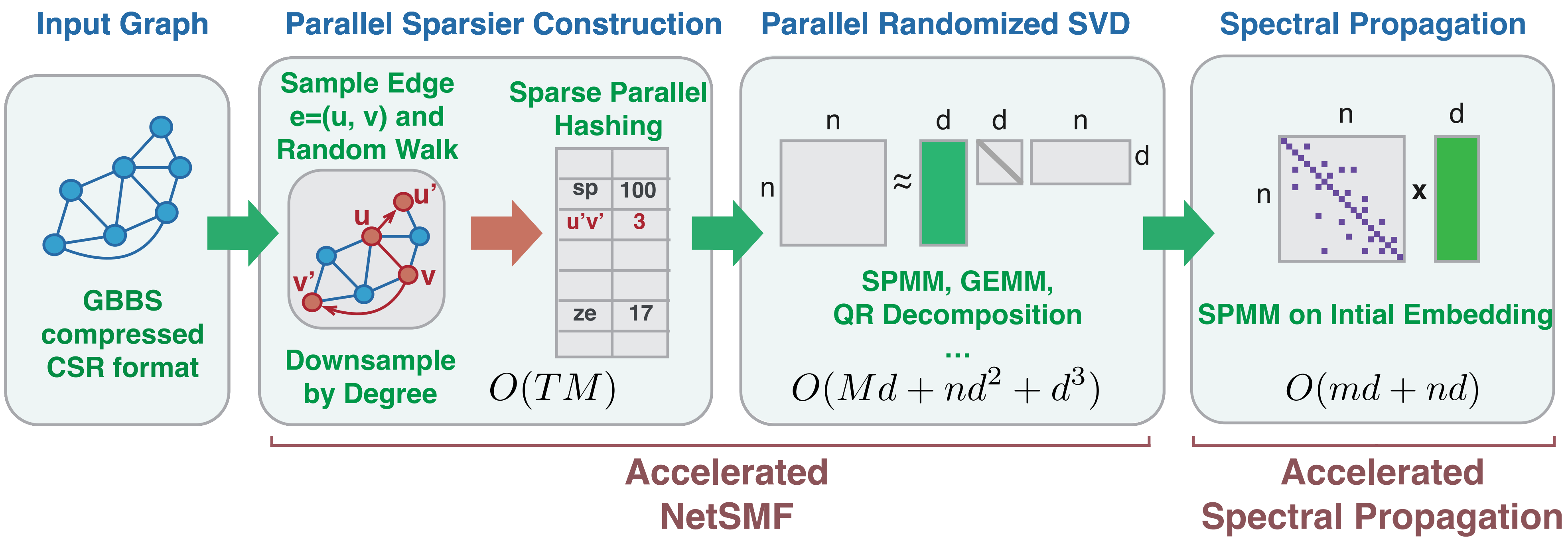
- Introduction to Network Embedding
- **LightNE: Co-design of Algorithm and System**
- Experiments on graphs with billions of edges.

LightNE: Design Goal



- **Scalable:** Embed graphs with 1B edges within 1.5 hours.
- **Lightweight:** Occupy hardware costs below 100 dollars measured by cloud rent to process 1B to 100B edges.
- **Accurate:** Achieve the highest accuracy in downstream tasks under the same time budget and similar resources.

LightNE: Algorithm and System Co-design



- Store input graph in GBBS[1] with compressed CSR format
- Highly optimized parallel graph processing

- **Parallel Sampling**: ... down-sample
- **Sparse Parallel Hashing**: collect sampled edges in parallel
- Downsample edge (u, v) with prob. $P_e = 1/d_u + 1/d_v$.
- Reduce #samples M from $O(mlogn)$ to $O(nlogn)$

- **Parallel Randomized SVD** implemented by Intel MKL

- **Spectral Propagation[3]**: enhance the embedding by $X \leftarrow \sum_{r=0}^{k} c_r (I - D^{-1}A)^r X$
- c_r's are chosen to be coefficients of Chebyshev polynomials

[1] Laxman Dhulipala, Guy E. Blelloch, and Julian Shun. 2018. Theoretically Efficient Parallel Graph Algorithms Can Be Fast and Scalable. In ACM Symposium on Parallelism in Algorithms and Architectures (SPAA). 393–404.
[2] Jiezhong Qiu, Yuxiao Dong, ... NetSMF ... as sparse matrix factorization. In WWW '19.
[3] Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. 2019. ProNE: fast and scalable network representation learning. In IJCAI '19. 4278–4284.

Roadmap

- Introduction to Network Embedding
- LightNE: Co-design of Algorithm and System
- **Experiments on graphs with billions of edges.**
    - Baselines: GraphVite, Pytorch-Big-Graph, **NetSMF, ProNE**

Comparison to NetSMF and ProNE

- Open Academic Graph (67,768,244 nodes, 895,368,962 edges)
- LightNE-small (#samples=m) and LightNE-large (#samples=200m)

Table 4: Comparison on OAG with label ratio 0.001%, 0.01%, 0.1% and 1%.

Metric	Method	Time	0.001%	0.01%	0.1%	1%
Micro	NetSMF (M=8Tm)	22.4 h	30.43	31.66	35.77	38.88
	ProNE+	21 min	23.56	29.32	31.17	31.46
	LightNE-Small	20.9 min	23.89	30.23	32.16	32.35
	LightNE-Large	1.53 h	**44.50**	**52.89**	**54.98**	**55.23**

- LightNE-Large achieves **15x speedup** (1.53h v.s. 22.4h) and **significant performance gain**, comparing to NetSMF.
- Not only does LightNE-Small run **faster** than ProNE+ (20.9 min v.s. 21 min), but also **outperforms ProNE+ significantly.**
- **Estimated price of LightNE-Large: 1.53h * 13$/h =20$**

Comparison to NetSMF and ProNE

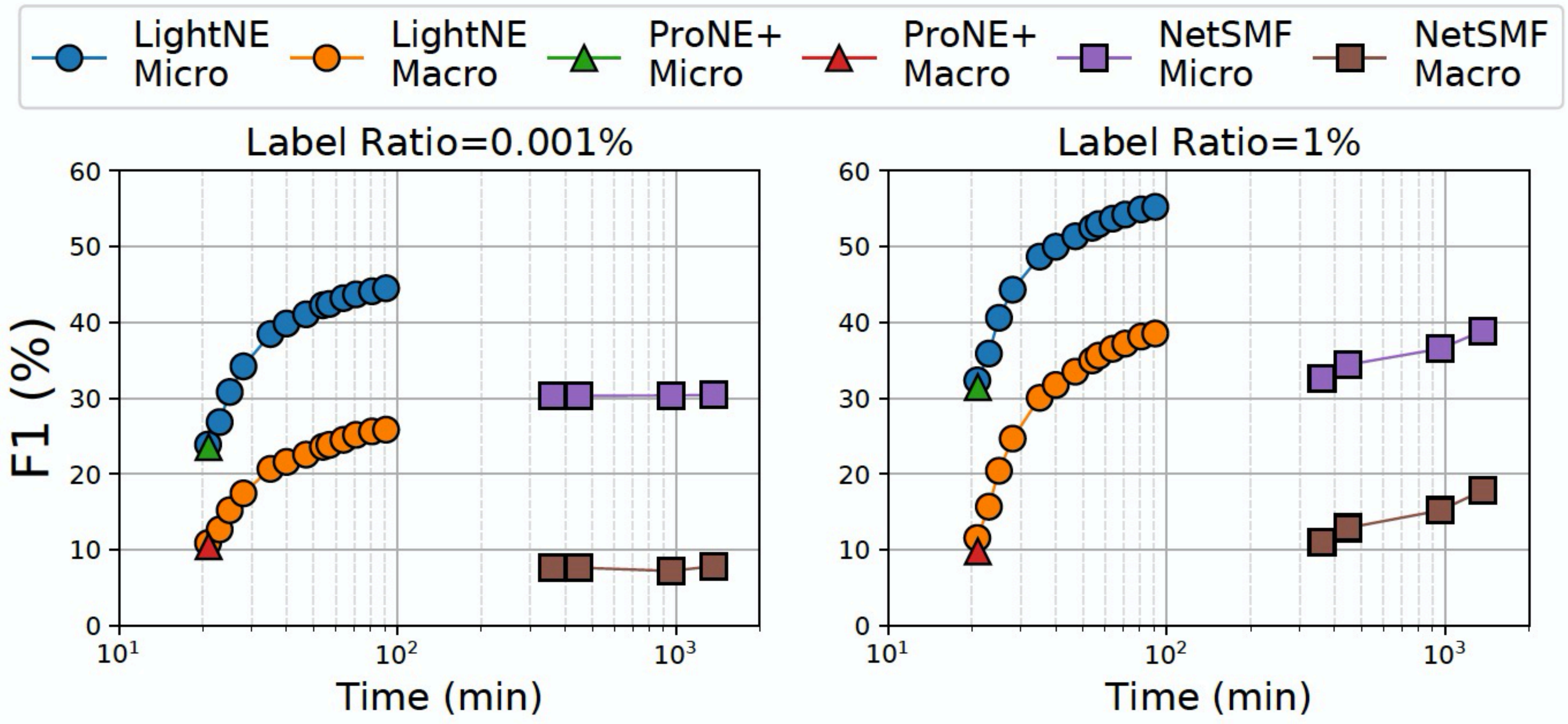
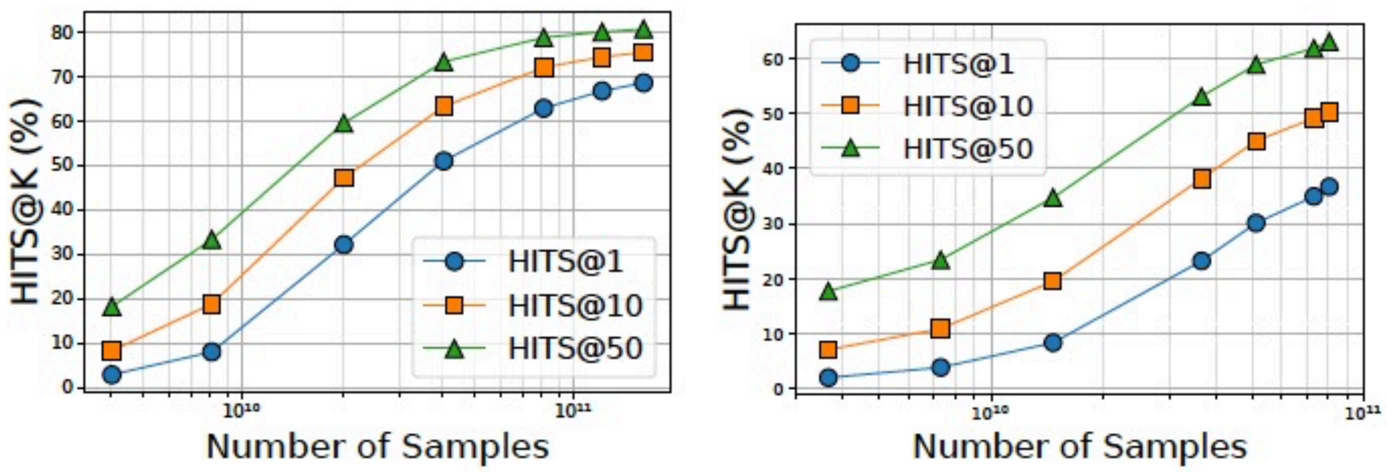


Figure 2: Efficiency-effectiveness trade-off curve of LIGHTNE.

Very Large Graphs

None of the existing network embedding systems can handle such large graphs in a single machine!

	ClueWeb-Sym	Hyperlink2014-Sym
n	978,408,098	1,724,573,718
m	74,744,358,622	124,141,874,032



(a) ClueWeb-Sym (b) Hyperlink2014-Sym

Figure 3: HITS@K ($K = 1, 10, 50$) of LightNE w.r.t. the number of samples.

Conclusion

- Propose LightNE, a **cost-effective, scalable, and high quality** network embedding system that scales to graphs with hundreds of billions of edges on a **single machine**.
- Introduce 4 techniques to network embedding for the first time:
  1. A new downsampling method to reduce the sample complexity of NetSMF.
  2. A parallel graph processing stack GBBS for memory efficiency and scalability;
  3. Sparse parallel hash table to maintain the matrix sparsifier in memory
  4. Intel MKL for efficient randomized SVD and spectral propagation.

Xi'an, Shaanxi, China
SIGMOD/PODS 2021

Thanks!

LightNE: A Lightweight Graph Processing System for Network Embedding

Jiezhong Qiu, Laxman Dhulipala, Jie Tang, Richard Peng, Chi Wang

https://github.com/xptree/LightNE.