

S M L 4 0 5

Deep Graph Library: Deep Graph learning at scale

George Karypis

Senior Principal Scientist, AI/ML
AWS



Agenda

Learning on graphs

Graph neural networks

Deep Graph Library (DGL)

Scaling training to large graphs

Recommending a show



Chinnu

Will Chinnu like Goliath?



Occupation: Artist
Education: MS
Country: Canada
Age: 25
Sex: Female
Device: Android

Storyline 

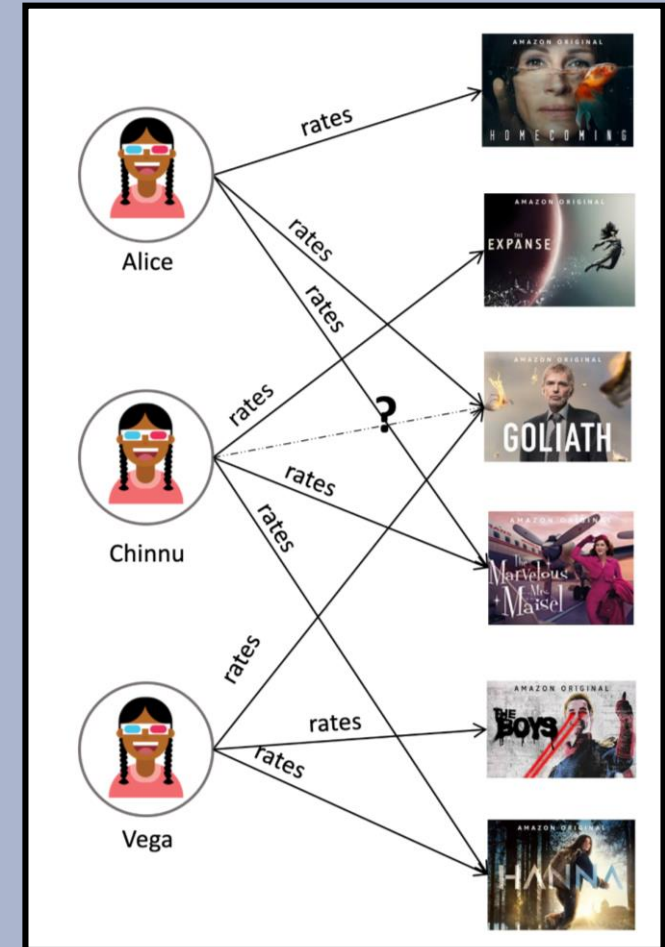
Billy McBride is going through a rough patch. He has been fired from the law firm he helped build, his wife has left him, and he's now a down on his luck ambulance chaser. A lady (Patty) approaches him to represent her in a wrongful death case. After reluctantly accepting to take on the case, a series of strange events befall Billy. Through death threats, harassment, and trumped up arrests, Billy embarks on obtaining justice, and it is one hell of a ride. Additional unique cases eventually come Billy and Patty's way, making the ride even more entertaining.

[Plot Summary](#) | [Add Synopsis](#)

Plot Keywords: [law firm](#) | [lawyer](#) | [nickname](#) | [name calling](#) | [reference to david and goliath](#) | [See All \(24\)](#) »

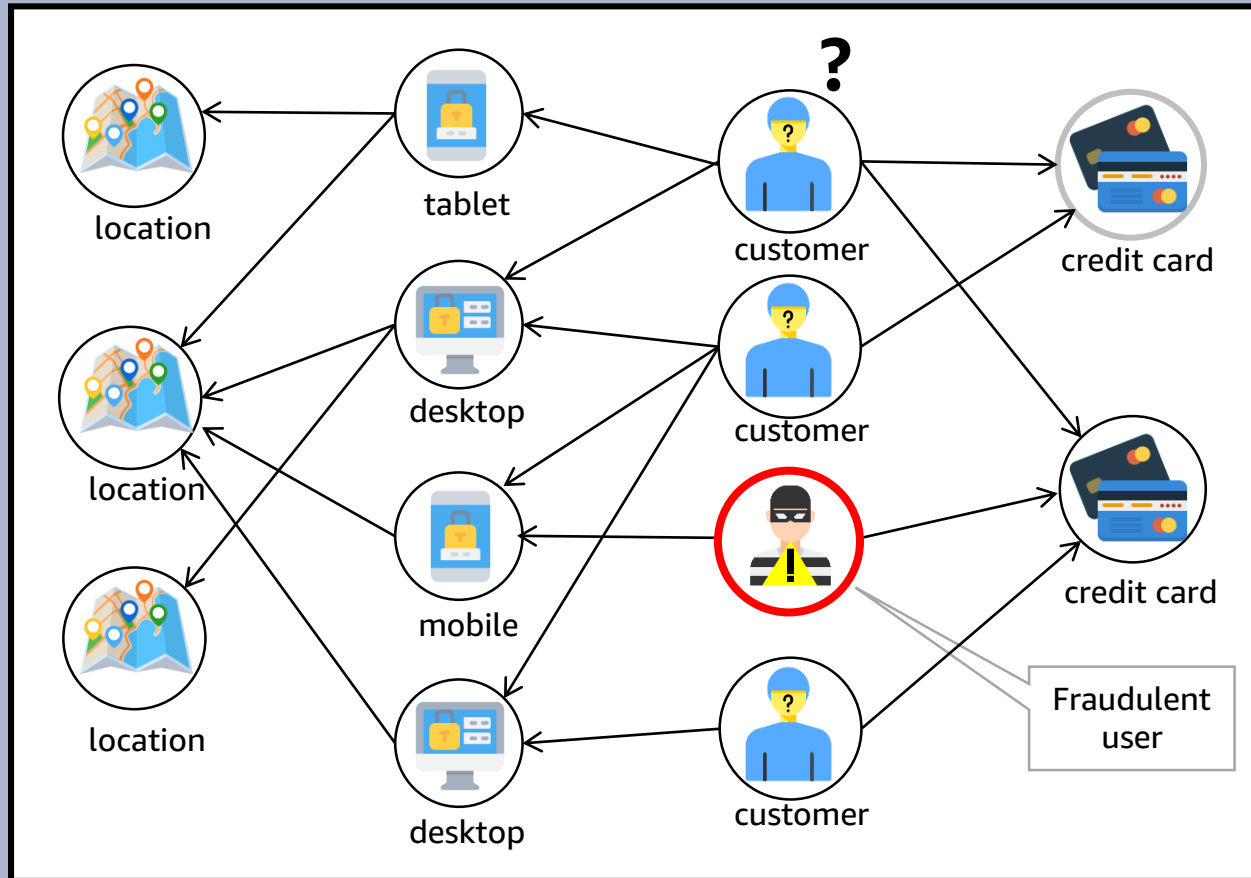
Genres: [Drama](#)

Certificate: TV-MA | [See all certifications](#) »
Parents Guide: [View content advisory](#) »



Collaborative (knowledge) graph

Detecting a fraudulent user



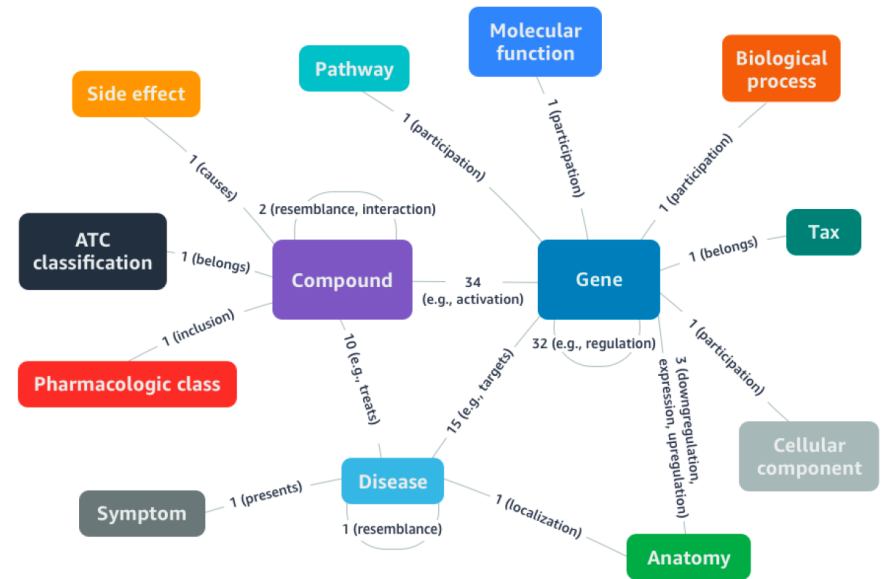
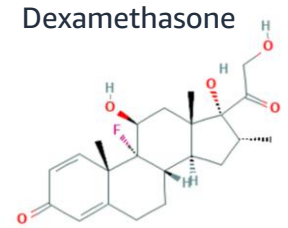
Deciding which drug to use for a new disease



The screenshot shows an Amazon Science article from June 2, 2020, by Cyrus Vahid. The headline is "Amazon Web Services open-sources biological knowledge graph to fight COVID-19". The article describes a knowledge graph combining data from six public databases and including machine learning tools. Below the article is a table of the top-100 drugs for COVID-19 treatment.

Drug name	Score	Ranking in top-100
Ribavirin	-0.21	0
Dexamethasone	-1.00	4
Colchicine	-1.08	8
Methylprednisolone	-1.16	16
Oseltamivir	-1.39	49
Deferoxamine	-1.51	87

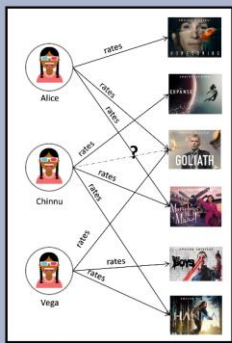
How can I predict which drugs are good candidates for a disease?



Drug repurposing knowledge graph

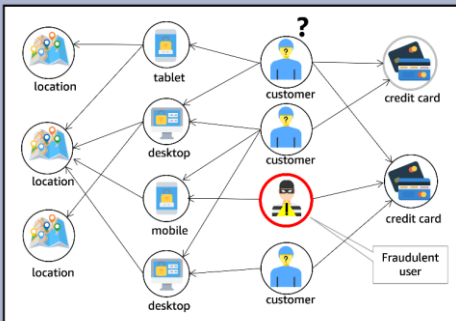


<https://github.com/gnn4dr/DRKG>
<https://arxiv.org/abs/2007.10261>

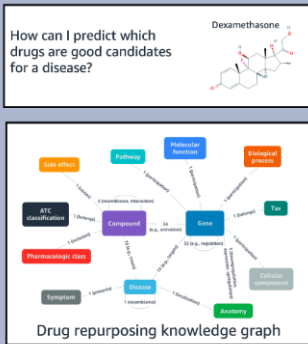
[illegible]

Collaborative (knowledge) graph

Detecting a fraudulent user

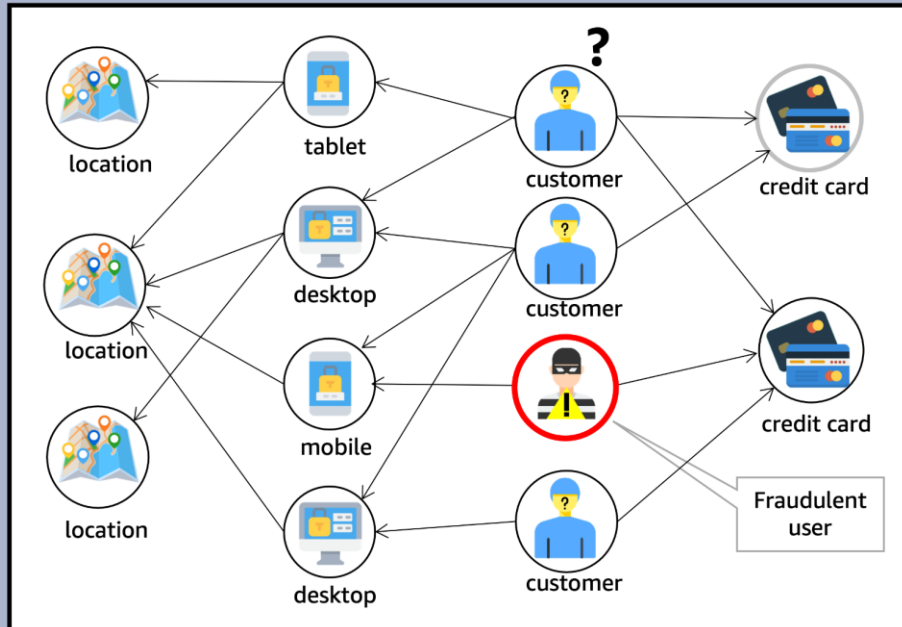


Deciding which drug to use for a new disease

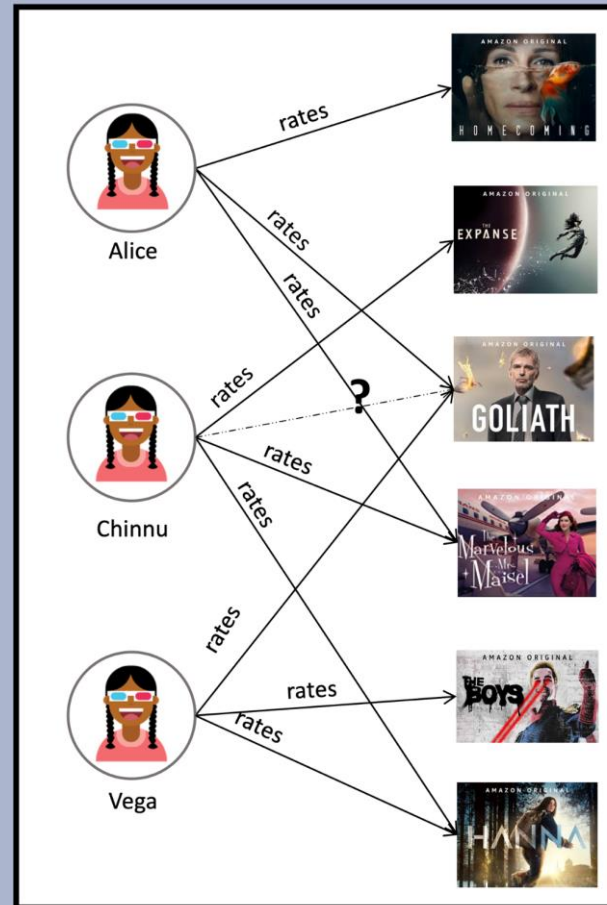


Tasks in graph learning

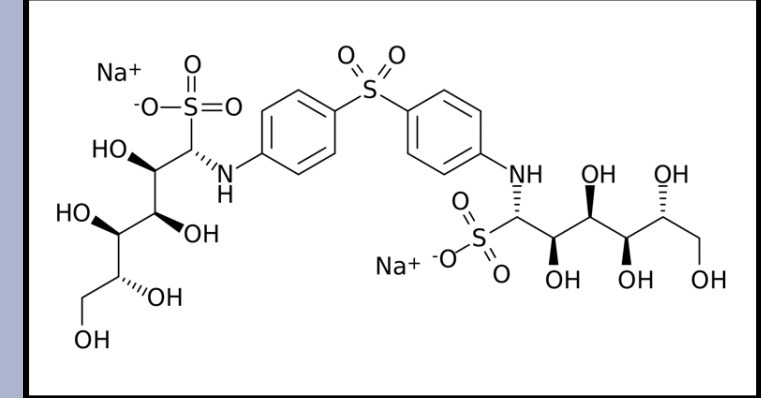
Node-level prediction



Edge-level prediction

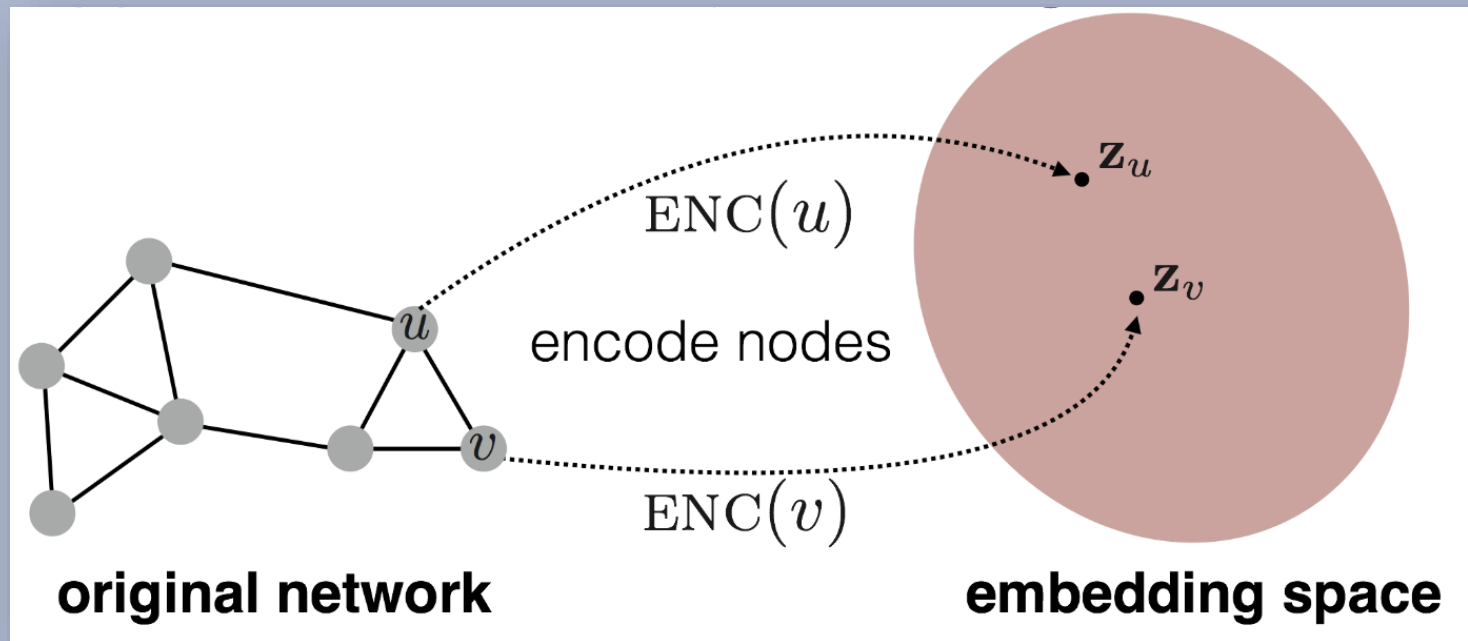


Graph-level prediction



Graph learning and node embeddings

Embed nodes to a low-dimension space by capturing the essential task-specific information and use them to train off-the-self classifiers.



<http://snap.stanford.edu/proj/embeddings-www/>

Graph embedding approaches

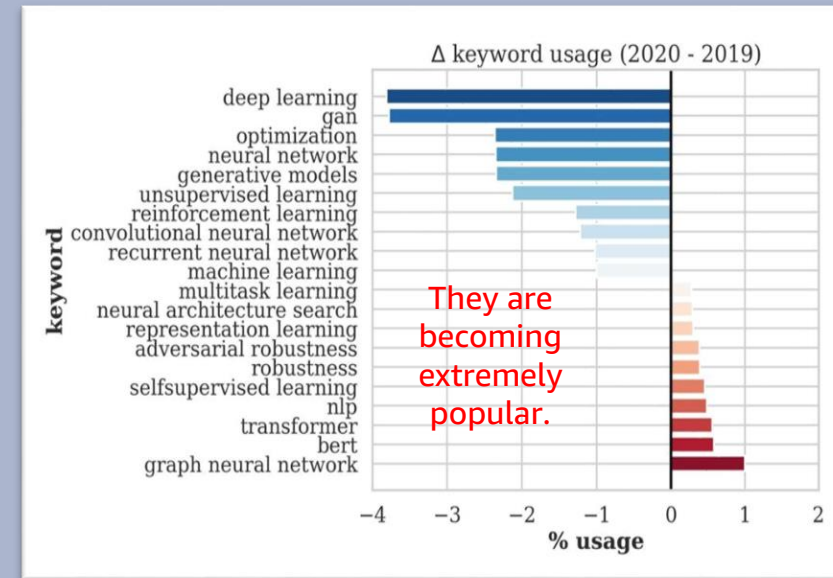
Generate embeddings using manual feature engineering.

Generate embeddings automatically using self-supervised dimensionality reduction approaches.

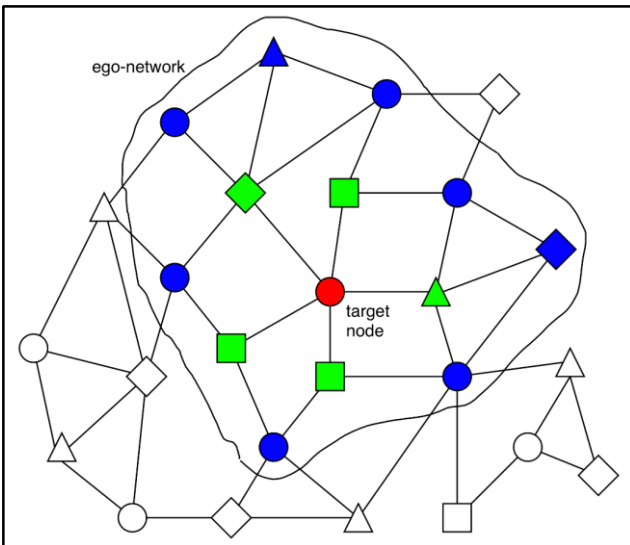
Can we do better?

Graph neural network (GNN)

A family of (deep) neural networks that learn node, edge, and graph embeddings



How do GNNs work?



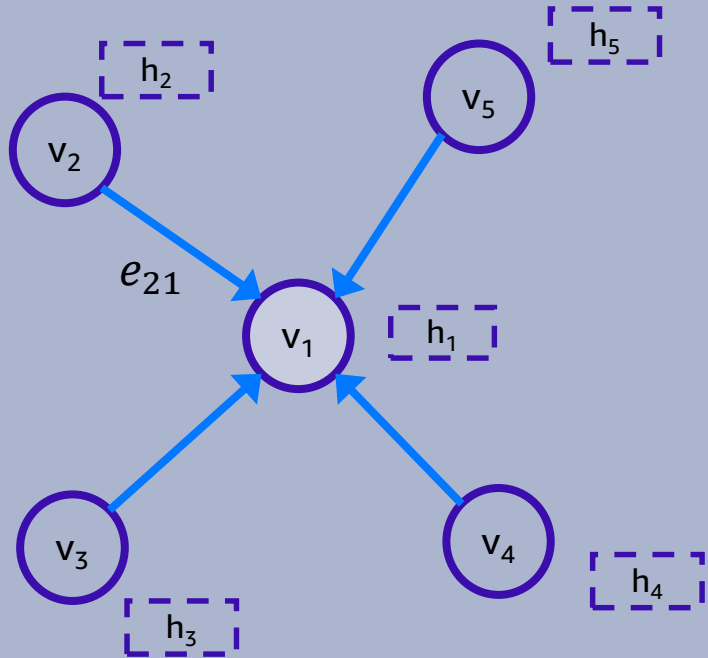
An ego-network around each node is used to learn an embedding that captures task-specific information.

The embeddings use both the structure of the graph and the features of the nodes and edges.

The embeddings are learned in an end-to-end fashion; thus, the predictions are a function of the target node's ego-network.

A general graph neural network formalism

Graph neural networks are based on **message-passing**.



Reduce/aggregate

$$m_v^{(l)} = \sum_{w \in N(v)} M^{(l)}(h_v^{(l-1)}, h_w^{(l-1)}, e_{vw})$$

Message

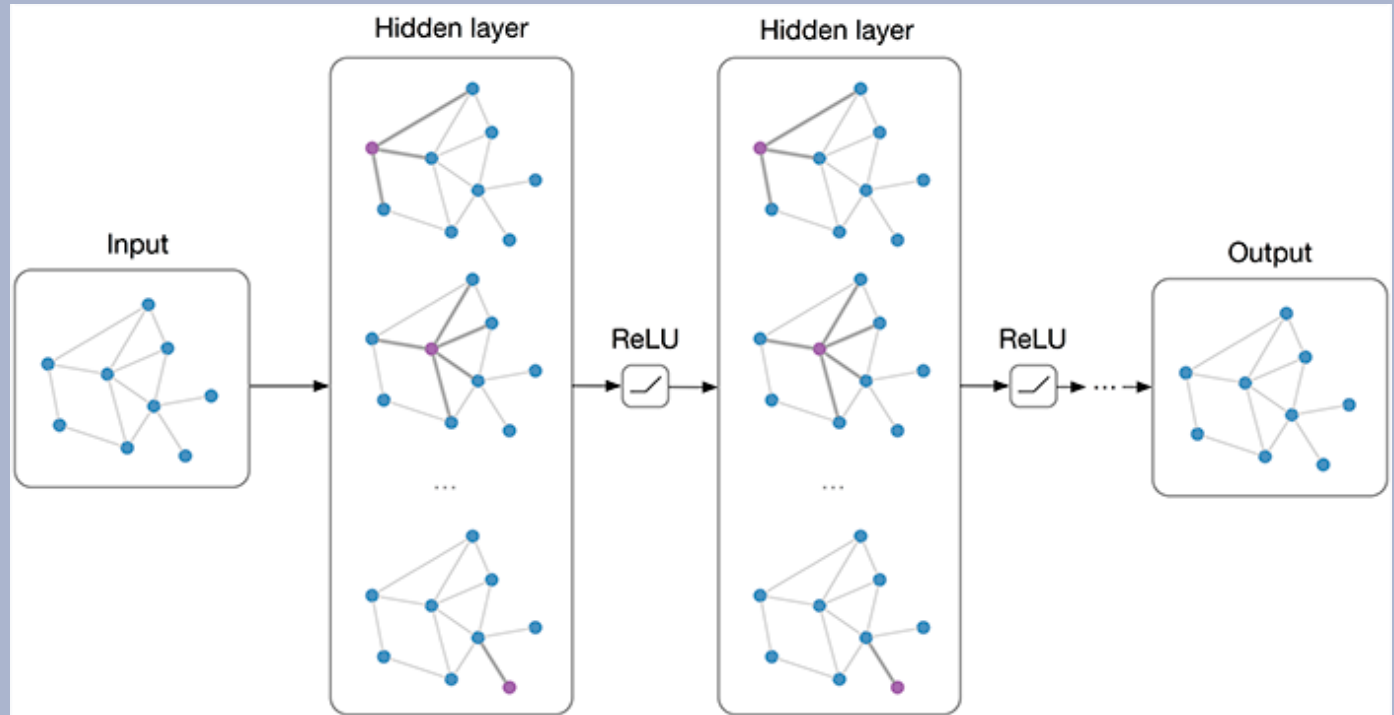
$$h_v^{(l)} = U^{(l)}(h_v^{(l-1)}, m_v^{(l)})$$

Update

GCN: Graph convolution network

$$m_v^{(l)} = \frac{1}{1 + |N(v)|} \sum_{w \in N(v) \cup \{v\}} h_w^{(l-1)}$$

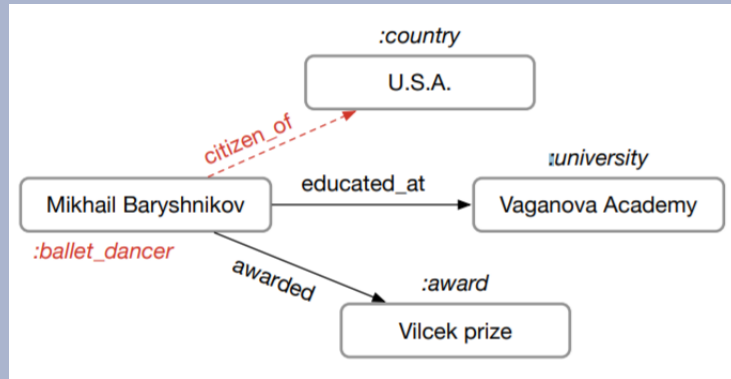
$$h_v^{(l)} = \phi(W^{(l)} m_v^{(l)})$$



<https://tkipf.github.io/graph-convolutional-networks/>

RGCN: Relational graph convolution networks

Handles graphs whose nodes are connected with different relations

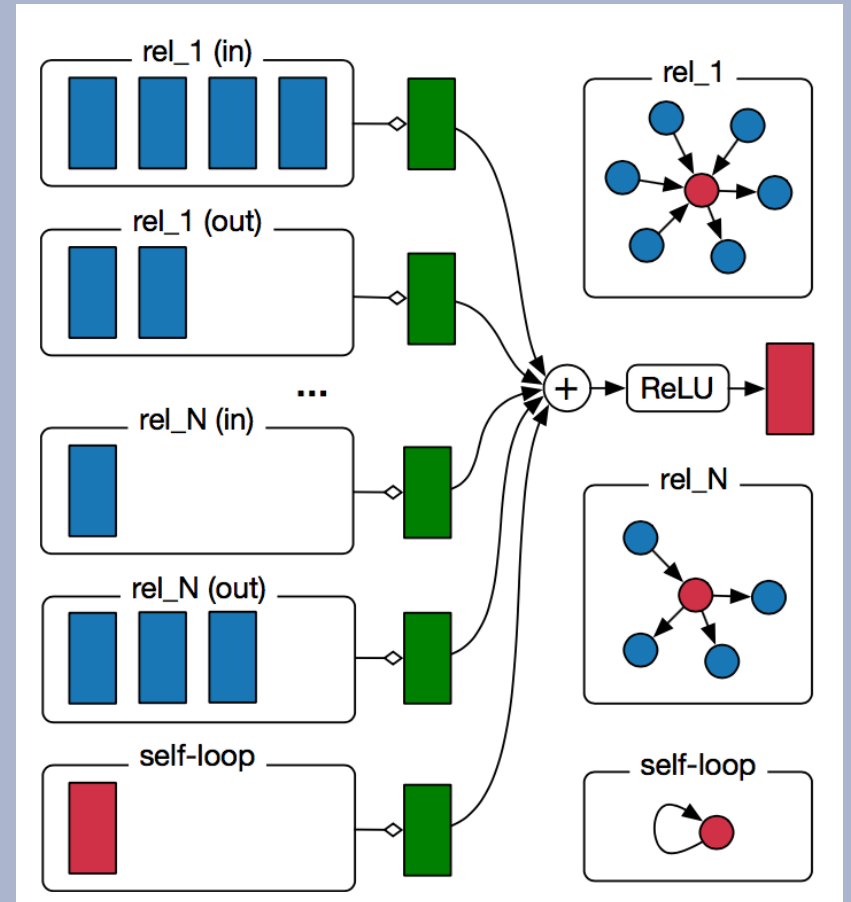


<https://arxiv.org/abs/1703.06103>

$$M_{vw}^{(l)} = \frac{1}{c_{v,r}} W_r^{(l)} h_w^{(l-1)}, r \text{ is the relation of } e_{vw}$$

$$m_v^{(l)} = \sum_{w \in N(v) \cup \{v\}} M_{vw}^{(l)}$$

$$h_v^{(l)} = \phi(W^{(l)} m_v^{(l)})$$



<https://arxiv.org/abs/1703.06103>

How to train GNN on large graphs

Many applications have extremely large graphs—millions of nodes and billions of edges:

- Social networks

- Recommendation

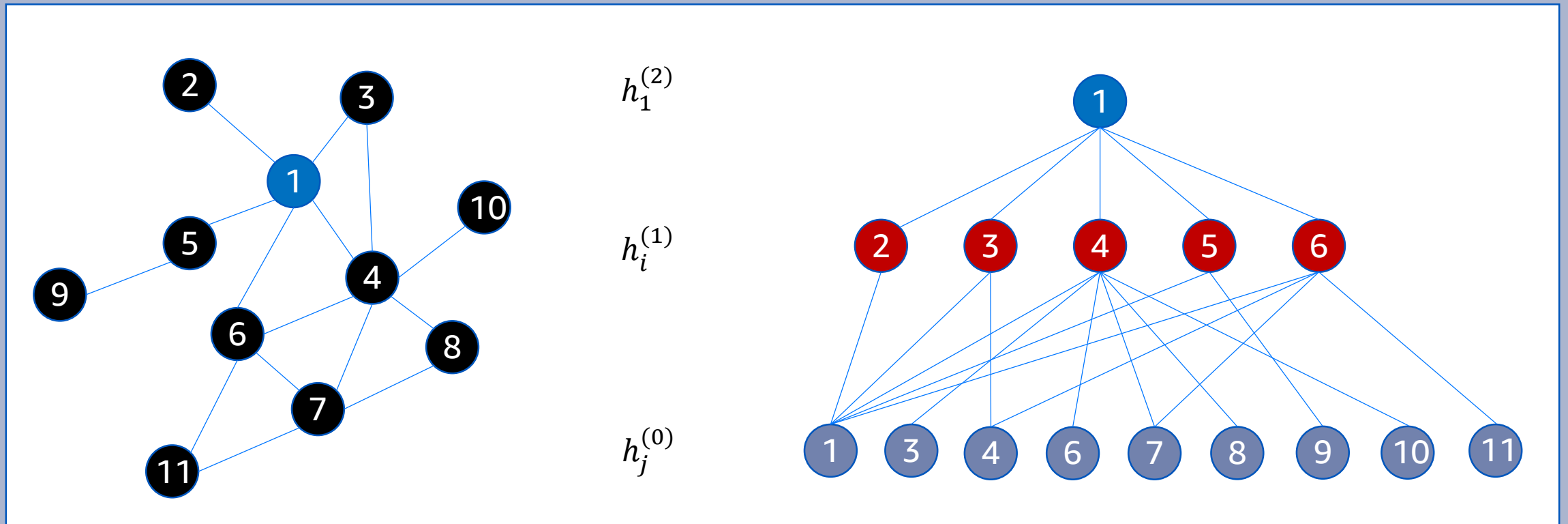
- Knowledge graph

- ...

A typical training method: mini-batch training

A glance of mini-batch training on graphs

A mini batch represents the computation graph for target nodes.

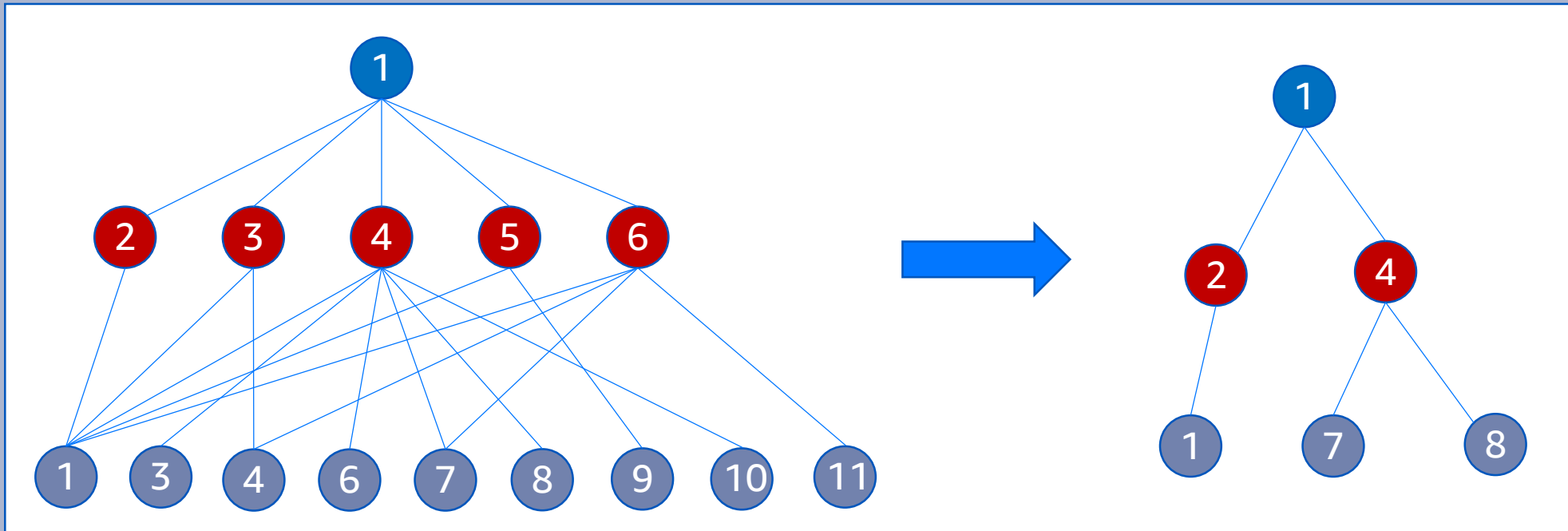


Mini-batch with neighbor sampling

Small-world graphs lead to a huge computation graph.

Prune the computation graph:

Sample neighbors from a neighbor list of a vertex.



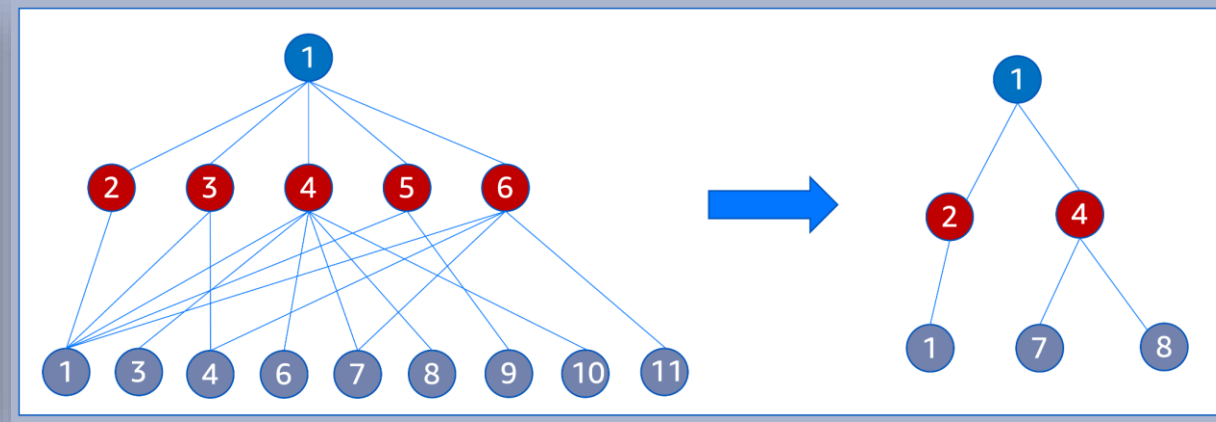
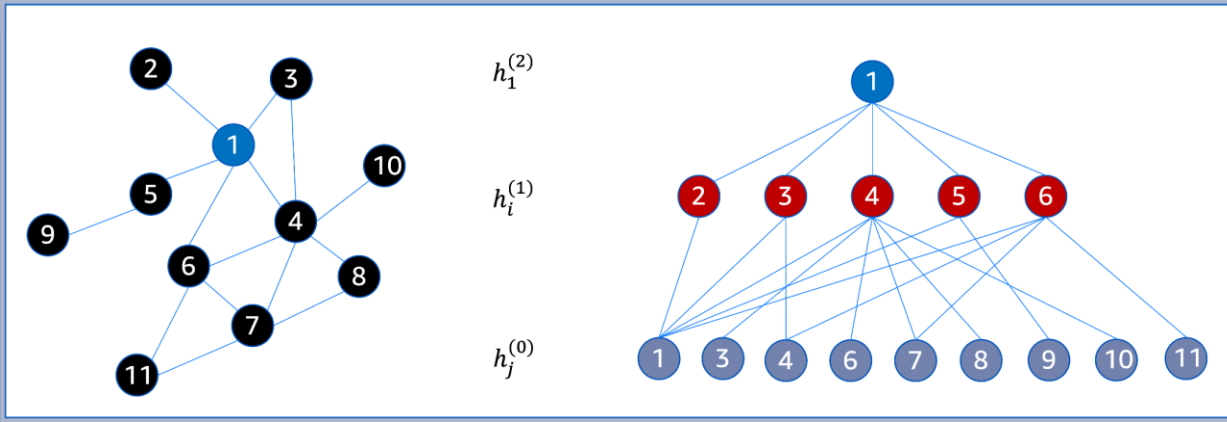
Deep Graph Library

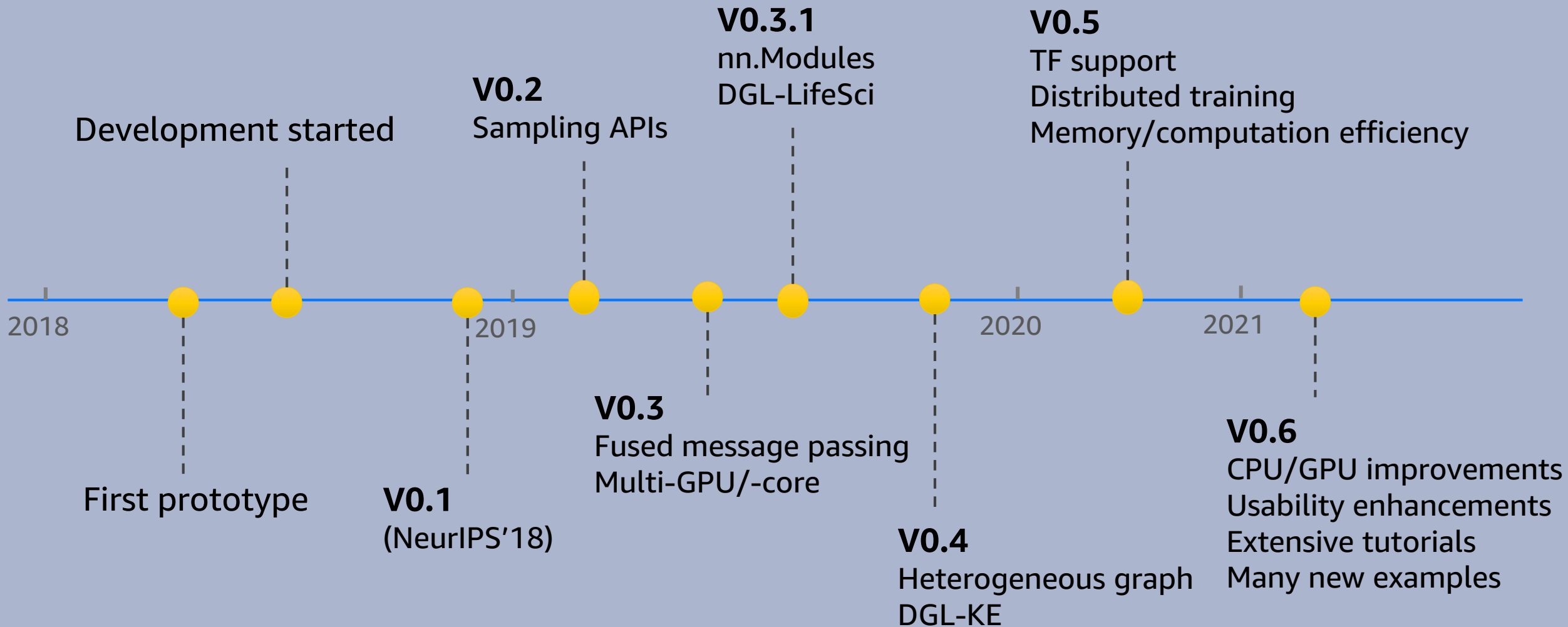


Why do we need something special for GNNs?

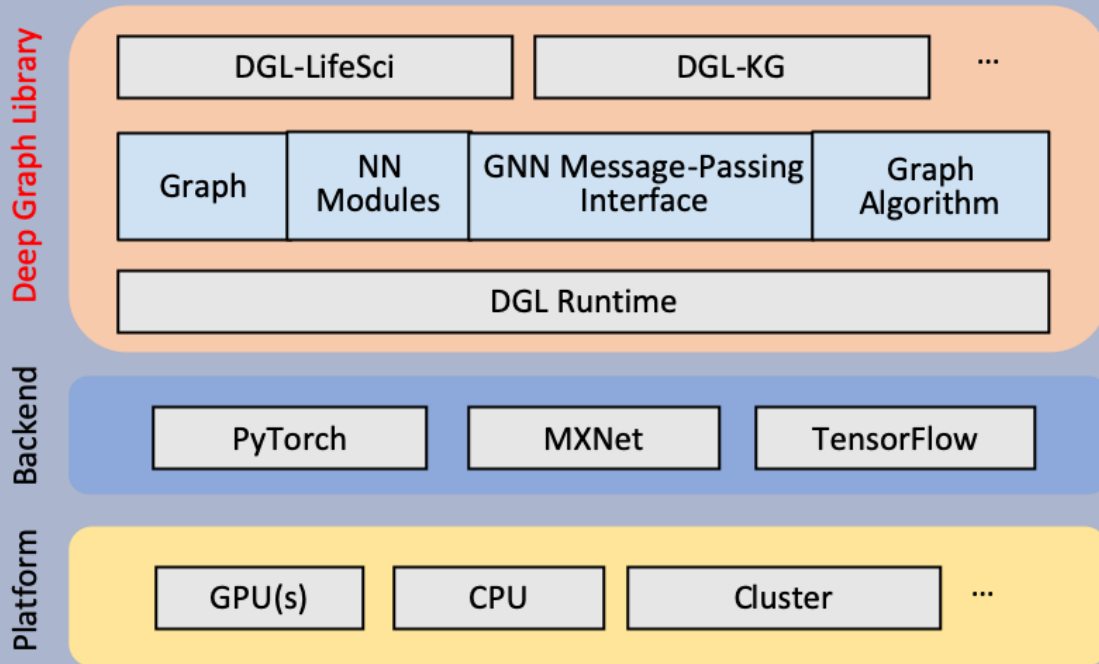
Deep learning frameworks are optimized for work-loads whose computational graphs are fixed and for calculations involving dense tensors.

Both do not hold for GNNs.





DGL's architecture and meta-objective



Forward and backward compatible

Forward: easy to develop new models

Backward: seamless integration with existing frameworks

Fast and scalable

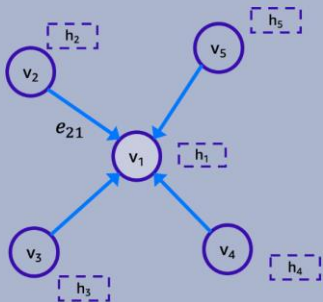
DGL programming interface

Graph as the core abstraction

DGLGraph

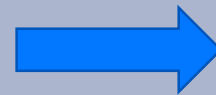
`g.ndata['h']`

Simple but versatile message passing APIs



$$m_v^{(l)} = \sum_{w \in N(v)} \phi^e(h_v^{(l-1)}, h_w^{(l-1)}, e_{wv})$$

$$h_v^{(l)} = \phi^v(h_v^{(l-1)}, m_v^{(l)})$$



`update_all(ϕ^e , Σ , ϕ^v)`

ϕ^e , ϕ^v , Σ can be user-defined functions (**UDFs**) or **built-in** symbolic functions.

DGL's API functionalities

Graph queries

Message passing API

Mini-batch generation

Distributed training (distributed graph, distributed tensors, etc.)


nn.Modules for popular GNN models

Graph transformations

Multi-graph operations (e.g., batching and readout)

Graph datasets



 DGL

0.6.x

Search docs

GET STARTED

Install and Setup

A Blitz Introduction to DGL

ADVANCED MATERIALS

User Guide

用户指南

Stochastic Training of GNNs

Paper Study with DGL

API REFERENCE

dgl

dgl.data

dgl.dataloading

dgl.DGLGraph

dgl.distributed

dgl.function

dgl.nn

dgl.nn.functional

dgl.ops

dgl.optim

dgl.sampling

User-defined Functions


DEVELOPER NOTES


Contribute to DGL

DGL Foreign Function Interface (FFI)

MISC

Frequently Asked Questions (FAQ)

 » Welcome to Deep Graph Library Tutorials and Documentation

 [Edit on GitHub](#)

Welcome to Deep Graph Library Tutorials and Documentation

Deep Graph Library (DGL) is a Python package built for easy implementation of graph neural network model family, on top of existing DL frameworks (currently supporting PyTorch, MXNet and TensorFlow). It offers a versatile control of message passing, speed optimization via auto-batching and highly tuned sparse matrix kernels, and multi-GPU/CPU training to scale to graphs of hundreds of millions of nodes and edges.

Getting Started

For absolute beginners, start with the [Blitz Introduction to DGL](#). It covers the basic concepts of common graph machine learning tasks and a step-by-step on building Graph Neural Networks (GNNs) to solve them.

For acquainted users who wish to learn more advanced usage,

- [Learn DGL by examples](#).
- Read the [User Guide \(中文版链接\)](#), which explains the concepts and usage of DGL in much more details.
- Go through the tutorials for [Stochastic Training of GNNs](#), which covers the basic steps for training GNNs on large graphs in mini-batches.
- [Study classical papers](#) on graph machine learning alongside DGL.
- Search for the usage of a specific API in the [API reference manual](#), which organizes all DGL APIs by their namespace.

Contribution

DGL is free software; you can redistribute it and/or modify it under the terms of the Apache License 2.0. We welcome contributions. Join us on [GitHub](#) and check out our [contribution guidelines](#).

Index

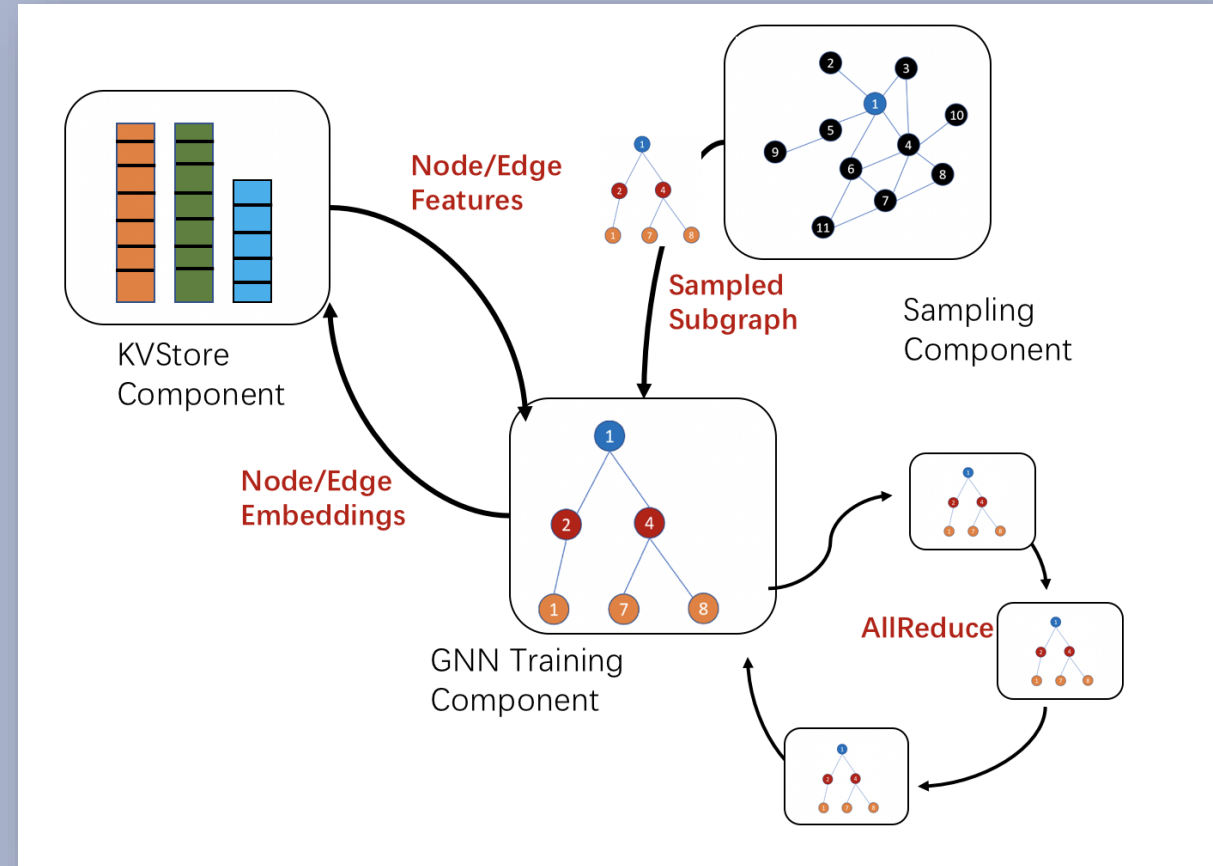
- [Index](#)

Scaling training to large graphs

Distributed training

Design

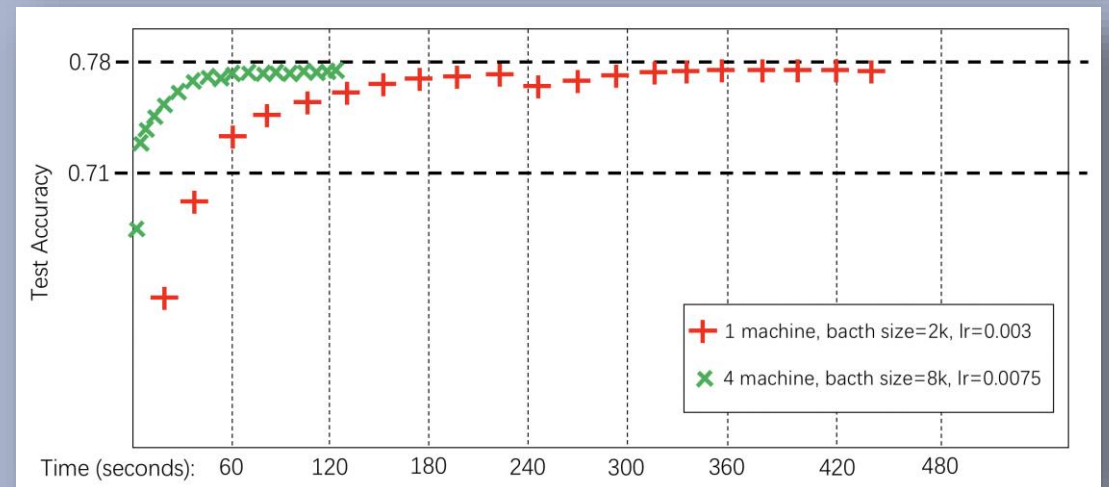
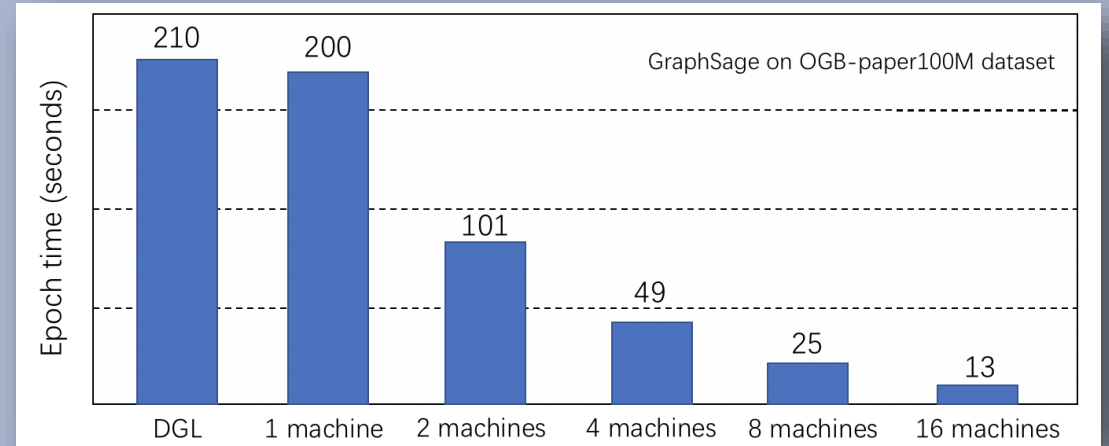
- Partitions the graph across the machines.
- Co-locates data with computations.
- Data parallel mini-batch training.
- Synchronous GNN model parameter updates.
- Asynchronous sparse embedding updates.



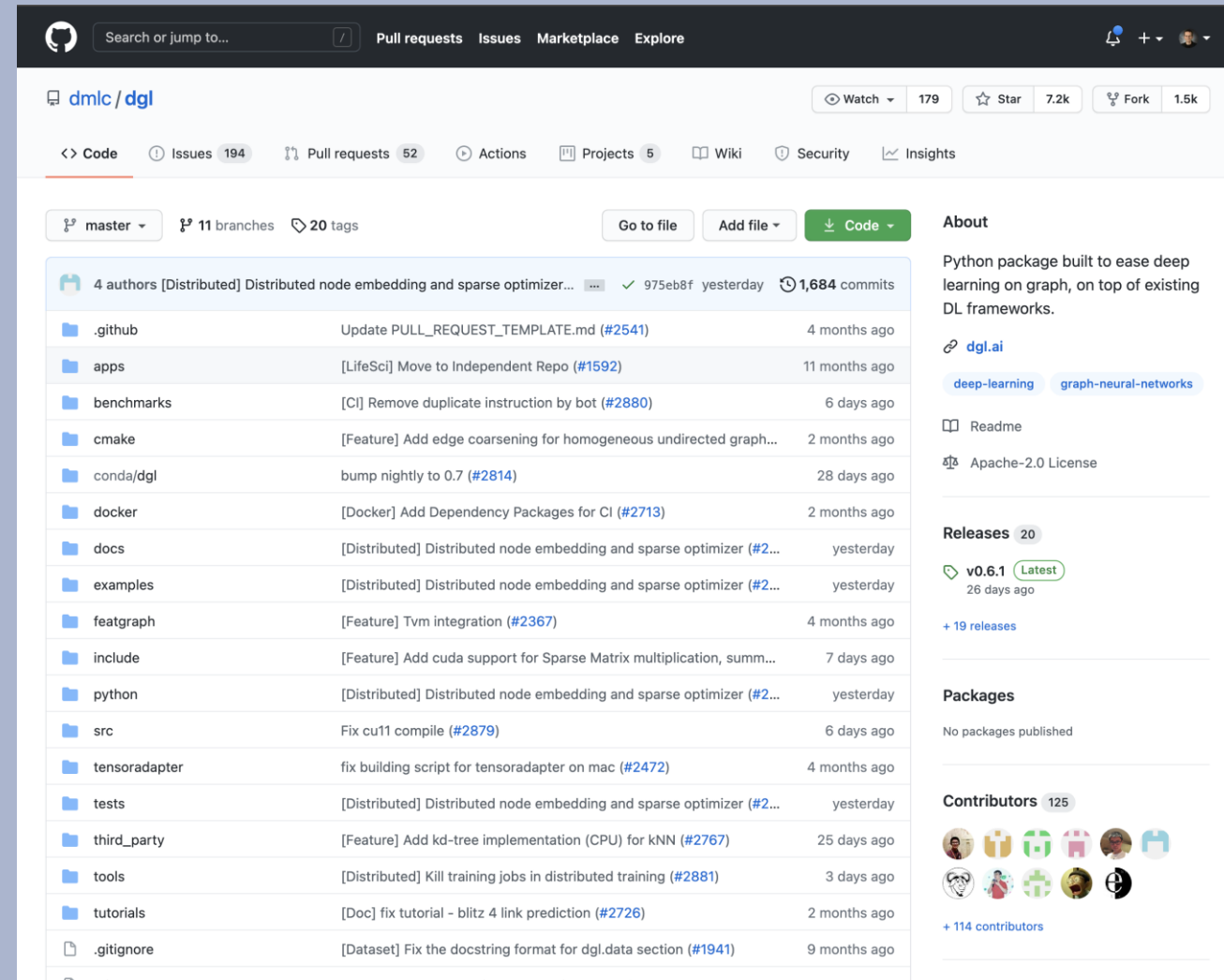
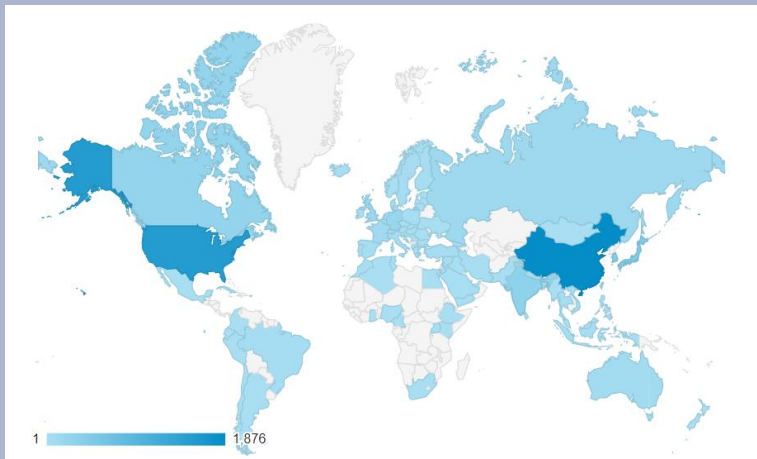
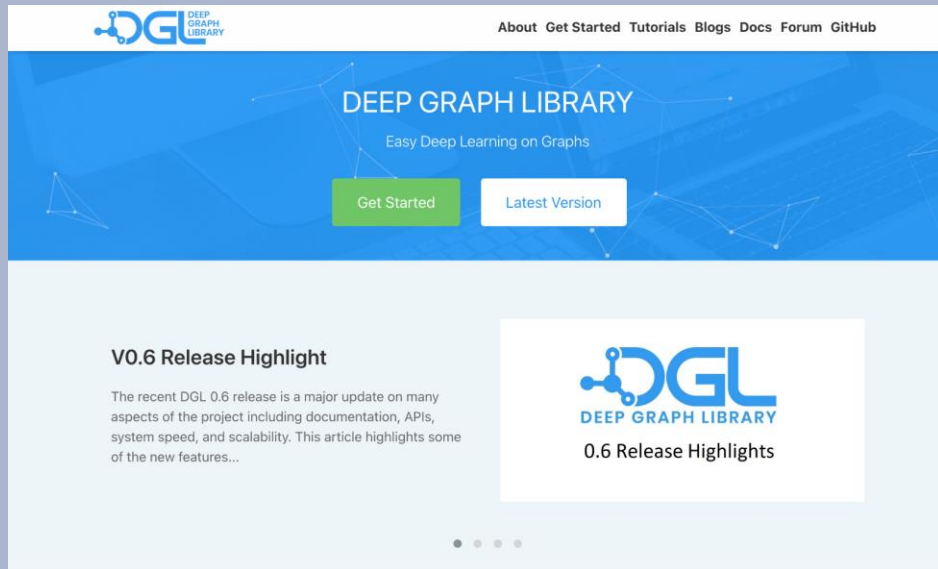
Distributed training

Scale to a graph with 110 million nodes, 3.2 billion edges

Linear speedup with no compromise in model accuracy



DGL is a vibrant open-source effort



<https://www.dgl.ai>

Learn ML with AWS Training and Certification

LEARN TO APPLY MACHINE LEARNING TO YOUR BUSINESS, UNLOCKING NEW INSIGHTS AND VALUE



Courses appropriate for your role, including developers, data scientists, data platform engineers, and business decision-makers



65+ [free digital machine learning courses](#) from AWS experts let you learn from real-world challenges solved by Amazon



Build credibility and confidence with AWS Certification, including [AWS Certified Machine Learning – Specialty](#)



Go deeper with labs, white papers, tech talks, and more by accessing the [AWS Ramp-Up Guide](#)



Visit aws.training/MachineLearning

Thank you!