

# Predicting Tweet Popularity in Twitter GeoCoV19 Dataset

Chris Winsor  
Update 4/11/2023

# From Prior Weeks:

## GeoCoV19: A Dataset of Hundreds of Millions of Multilingual COVID-19 Tweets with Location Information

Umair Qazi, Muhammad Imran, Ferda Ofli  
{uqazi, mimran, ofli}@hbku.edu.qa

Qatar Computing Research Institute, Hamad Bin Khalifa University, Doha, Qatar

### Abstract

*The past several years have witnessed a huge surge in the use of social media platforms during mass convergence events such as health emergencies, natural or human-induced disasters. These non-traditional data sources are becoming vital for disease forecasts and surveillance when preparing for epidemic and pandemic outbreaks. In this paper, we present GeoCoV19, a large-scale Twitter dataset containing more than 524 million multilingual tweets posted over a period of 90 days since February 1, 2020. Moreover, we employ a gazetteer-based approach to infer the geolocation of tweets. We postulate that this large-scale, multilingual, geolocated social media data can empower the research communities to evaluate how societies are collectively coping with this unprecedented global crisis as well as to develop computational methods to address challenges such as identifying fake news, understanding communities' knowledge gaps, building disease forecast and surveillance models, among others.*

### 1 Introduction

Social media platforms such as Twitter receive an overwhelming amount of messages during emergency events, including disease outbreaks, natural and human-induced disasters. In particular, the information shared on Twitter during disease outbreaks is a goldmine for the field of epidemiology. Research studies show that Twitter provides timely access to health-related data about chronic disease, outbreaks, and epidemics [4]. In this paper, we present GeoCoV19, a large-scale Twitter dataset about the COVID-19 pandemic. Coronavirus disease 2019 or COVID-19 is an infectious disease that was first identified in December 2019 and has spread globally since then. The World Health Organization (WHO) declared the COVID-19 outbreak a pandemic on March 11, 2020. As of May 6, 2020, more than 263K fatalities have been recorded, and more than 3.8 million people have been infected globally. Twitter has seen a massive surge in the daily traffic amid COVID-19. People try to connect with their family and friends, look for the latest information about the pandemic, report their symptoms, and ask questions. Moreover, many conspiracies, rumors, and misinformation began to surface on social media, e.g., drinking bleach can cure it, Bill Gates is behind it, etc. Furthermore, both benefits and unanticipated consequences of lockdowns and closure of businesses around the globe and social distancing are among the top topics on social media.

In this dataset collection, we aimed at covering several different perspectives related to the COVID-19 pandemic ranging from social distancing to food scarcity and symptoms to treatments and shortage of supplies and masks. The dataset contains more than 524 million multilingual tweets collected over a period of 90 days starting from February 1, 2020 till May 1, 2020, using hundreds of multilingual hashtags and keywords. Various public health and disease surveillance applications that use social media rely on broad coverage of location information. However, only a limited number of tweets contain geolocation information (i.e., 1-3%). To increase

- Q: What are we predicting?
- A: Number of re-tweets given text of tweet.

Q: What attributes are available?

- Tweet ID and full tweet text
- User IDs (sending), follower/friend counts
- Lots of relevant tweet info (retweet, favorite counts)
- Target:
  - Number of re-tweets given text of a tweet.

# Prior Weeks (Tools and Process)

additional detail in slides from 4/4

## PyG: Pytorch Geometric

- Getting Started: [https://pytorch-geometric.readthedocs.io/en/latest/get\\_started/colabs.html](https://pytorch-geometric.readthedocs.io/en/latest/get_started/colabs.html)



## Supporting Materials

- Stanford CS224 (Machine Learning with Graphs)
- <http://web.stanford.edu/class/cs224w/index.html>
- Lots of good references including:

## "Self-Supervised Learning For Graphs"

<https://medium.com/stanford-cs224w/self-supervised-learning-for-graphs-963e03b9f809>

**CS224W**



CS224W: Machine Learning with Graphs

Stanford / Winter 2023

**Logistics**

**Lectures:** are on Tuesday/Thursday 3:00-4:20pm **in person** in the [NVIDIA Auditorium](#).

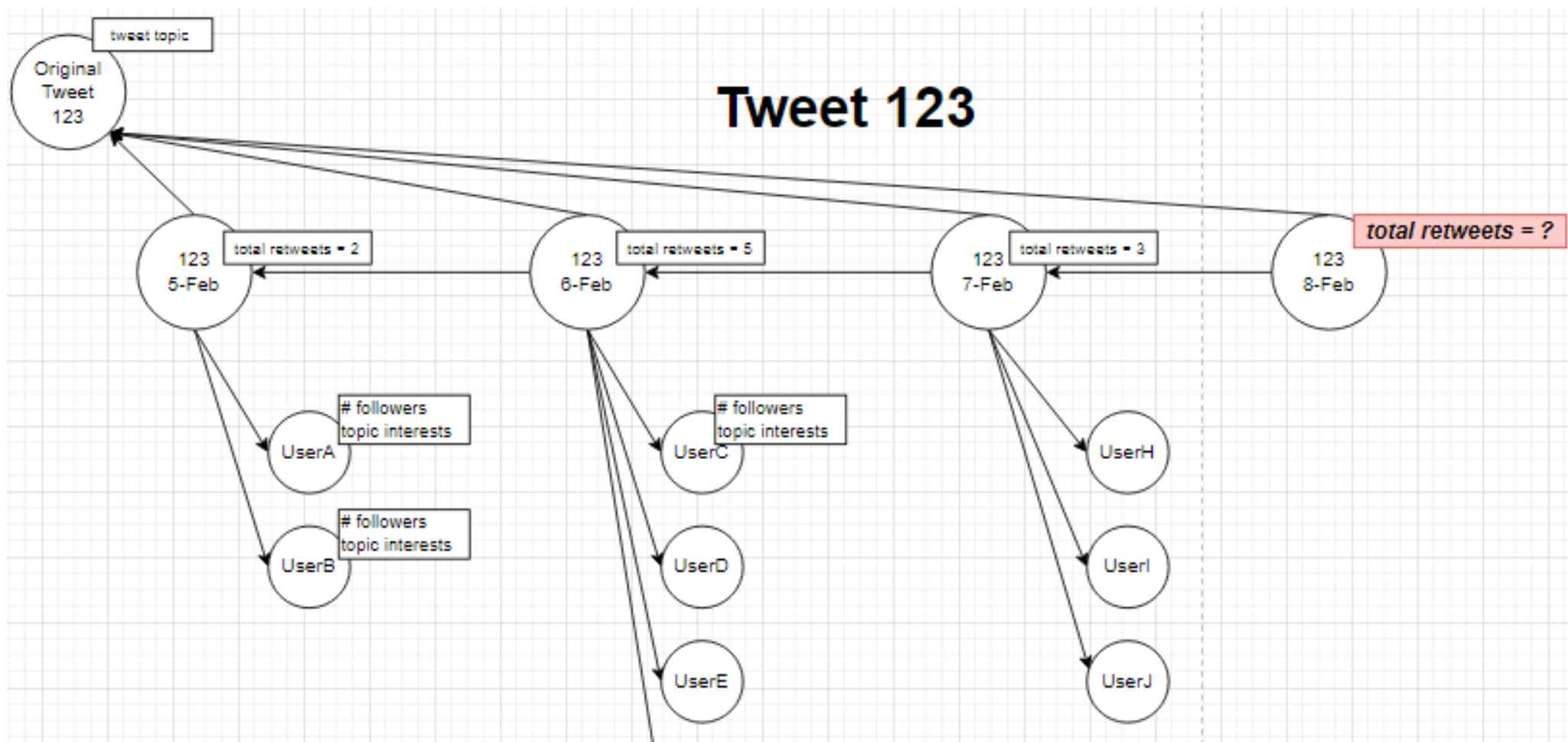
**Lecture Videos:** are available on [Canvas](#) for all the enrolled Stanford students.

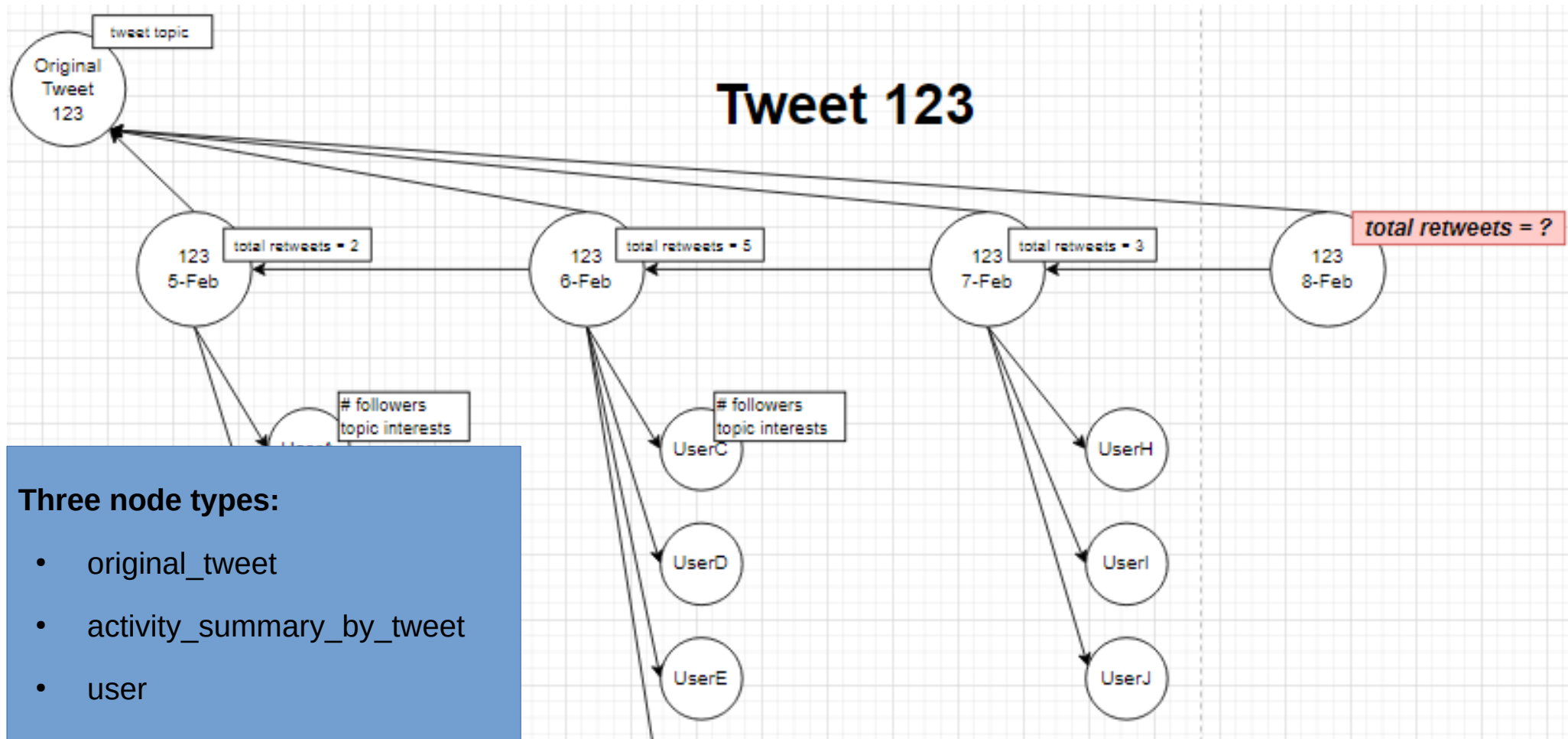
**Public resources:** The lecture slides and assignments will be posted online as the course progresses. We are happy for anyone to use these resources, but we cannot grade the work of any students who are not officially enrolled in the class.

**Contact:** Students should ask *all* course-related questions on Ed (accessible from

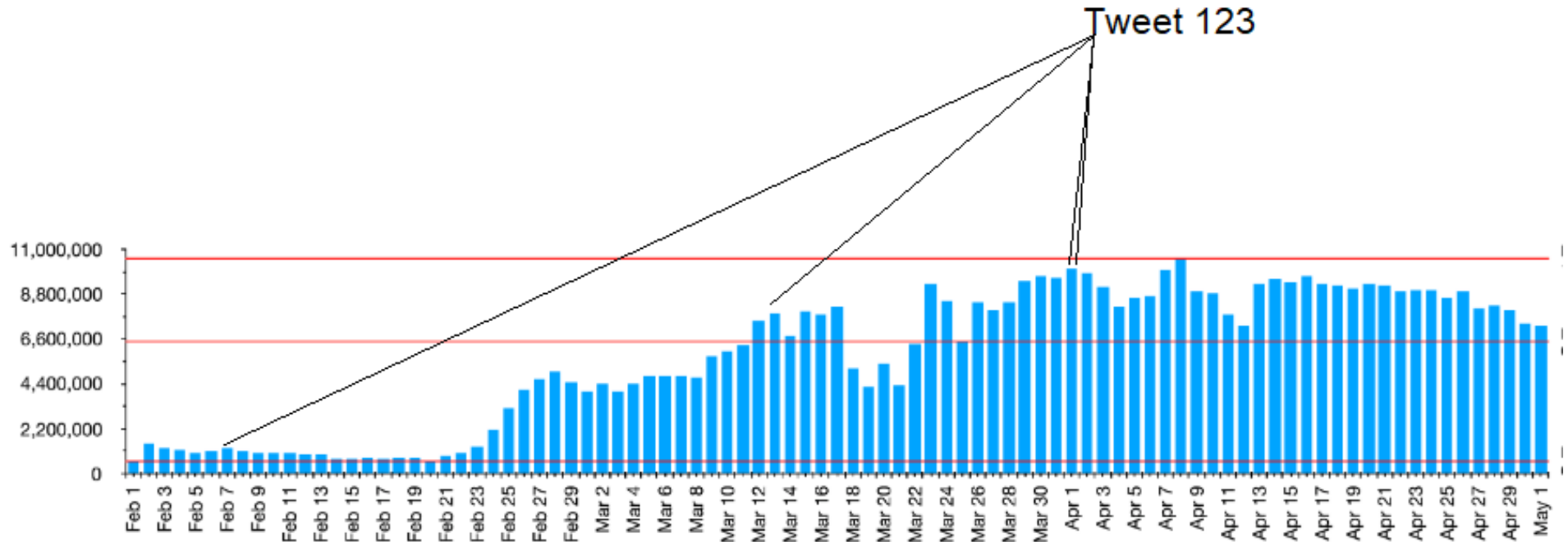
# This week

- Infrastructure: self-supervised learning for heterogenous graph
- Preprocessing





# Preprocessing – the “why”



# Preprocessing

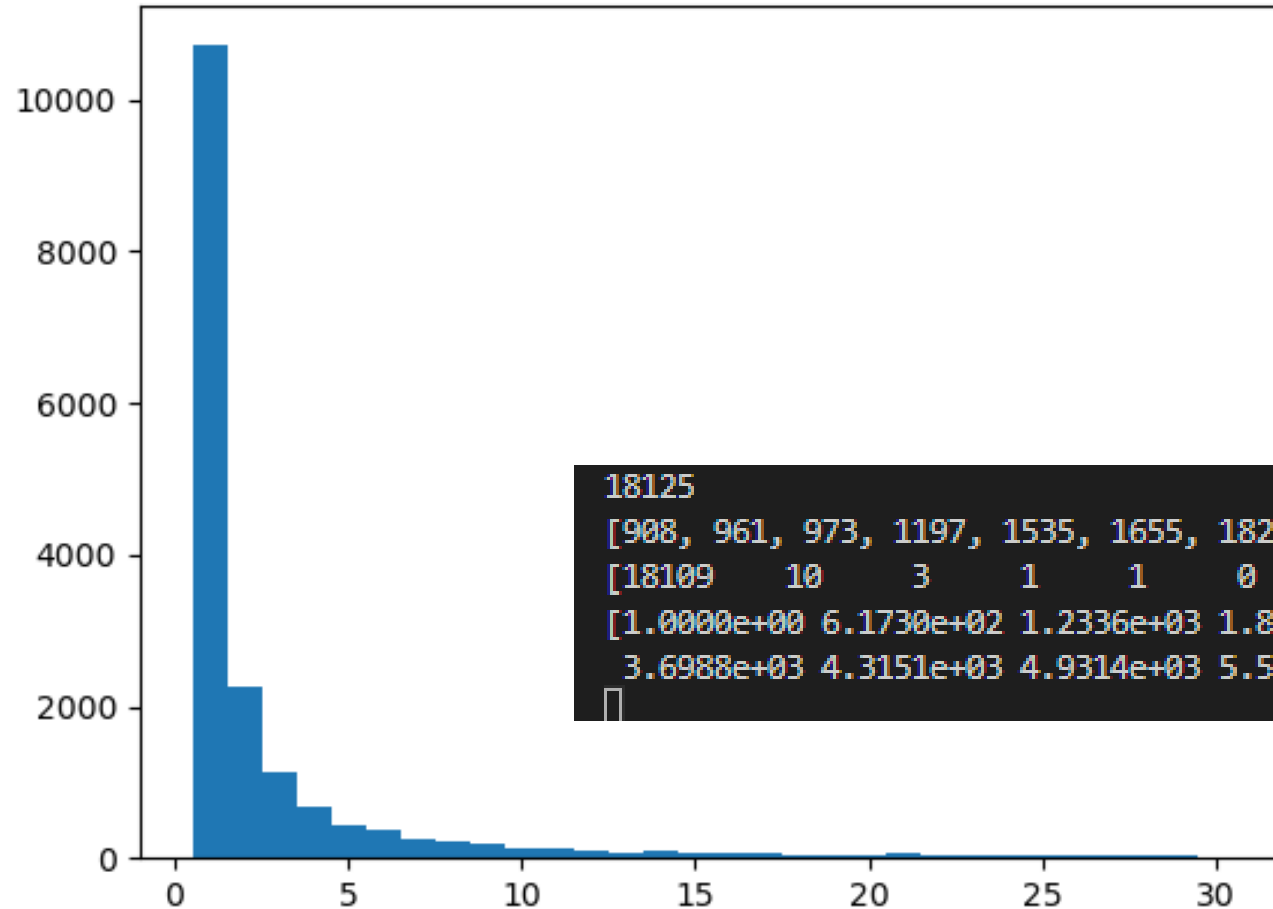
- Grouped based on original tweet ID (each original tweet will be a graph)
  - 1) read fresh tweets from raw .ijson, select relevant data, sort by original tweet ID, write to standard format .json
  - 2) merge that day's tweets into group files
- Stream oriented file processing (not in-memory)



# Preprocessing Output

- original\_tweet\_id 1220585270115876900
- original\_tweet\_text [THREAD] On Friday, January 24, the death toll in China's viral novel coronavirus outbreak has risen to 25, with the number of confirmed cases also leaping to 830, the government said. Here's what we know so far on the novel coronavirus. <https://t.co/ytrVoisbKh>
- retweet\_user\_ids [141914884, 177141319, 1186367835129438200, 58208791, 808923263955071000, 3690705794, 408438147, 140382263, 913692795554033700, 78956685]
- retweet\_dates ['Sat Feb 01 07:37:29 +0000 2020', 'Sat Feb 01 03:28:19 +0000 2020', 'Sat Feb 01 04:34:13 +0000 2020', 'Sat Feb 01 05:31:44 +0000 2020', 'Sat Feb 01 07:35:50 +0000 2020', 'Sat Feb 01 07:03:30 +0000 2020', 'Sat Feb 01 04:31:44 +0000 2020', 'Sat Feb 01 01:53:18 +0000 2020', 'Sat Feb 01 05:28:21 +0000 2020', 'Sat Feb 01 04:39:28 +0000 2020']
- number\_retweets 10

# Feb 1, 2020



18125

[908, 961, 973, 1197, 1535, 1655, 1823, 2407, 2476]

[18109 10 3 1 1 0 0 0 0 1]

[1.0000e+00 6.1730e+02 1.2336e+03 1.8499e+03 2.4662e+03 3.0825e+03

3.6988e+03 4.3151e+03 4.9314e+03 5.5477e+03 6.1640e+03]

# PyG Dataset and DataLoader (resources/references)

- Dataset download and rehydration
  - <https://paperswithcode.com/dataset/geocov19>
  - <https://crisisnlp.qcri.org/covid19>
  - <https://github.com/docnow/hydrator>
- Code for self-supervised graph learning, downstream task
  - <https://medium.com/stanford-cs224w/self-supervised-learning-for-graphs-963e03b9f809>
- Preprocessing into heterogeneous graph dataset (HeteroData), NLP encoding
  - <https://pytorch-geometric.readthedocs.io/en/latest/tutorial/heterogeneous.html>
  - [https://pytorch-geometric.readthedocs.io/en/latest/tutorial/load\\_csv.html](https://pytorch-geometric.readthedocs.io/en/latest/tutorial/load_csv.html)
  - [https://pytorch-geometric.readthedocs.io/en/latest/tutorial/create\\_dataset.html](https://pytorch-geometric.readthedocs.io/en/latest/tutorial/create_dataset.html)
  - <https://www.sbert.net/> (sentence transformer-based encoding)



Paridhi Maheshwari

Jan 18, 2022 · 12 min read · Listen



## Self-Supervised Learning For Graphs

*By Paridhi Maheshwari, Jian Vora, Sharmila Reddy Nangi as part of the Stanford CS 224W course project.*

A large part of deep learning revolves around finding rich representations of unstructured data such as images, text and graphs. Conventional methods try to find these representations using some end goal we want to perform. This is typically done in a supervised setting where we have labelled data. However, in many real-world applications, we do not have labels associated with data, but instead we have an abundance of unlabelled data.

# GeoCoV19 Infrastructure Test

We reproduce work of from Maheshwari et al. This includes:

10 sample DataSets, DataLoader

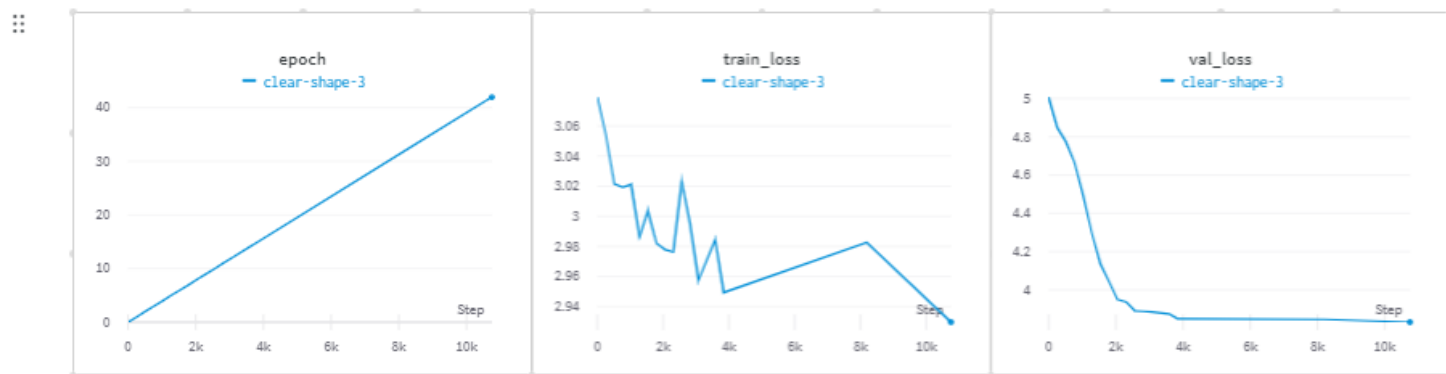
6 augmentation techniques

2 loss functions

PyG training/validation loop

Fine tuning using labeled data

## ▼ Section 1




# Plans for this coming week

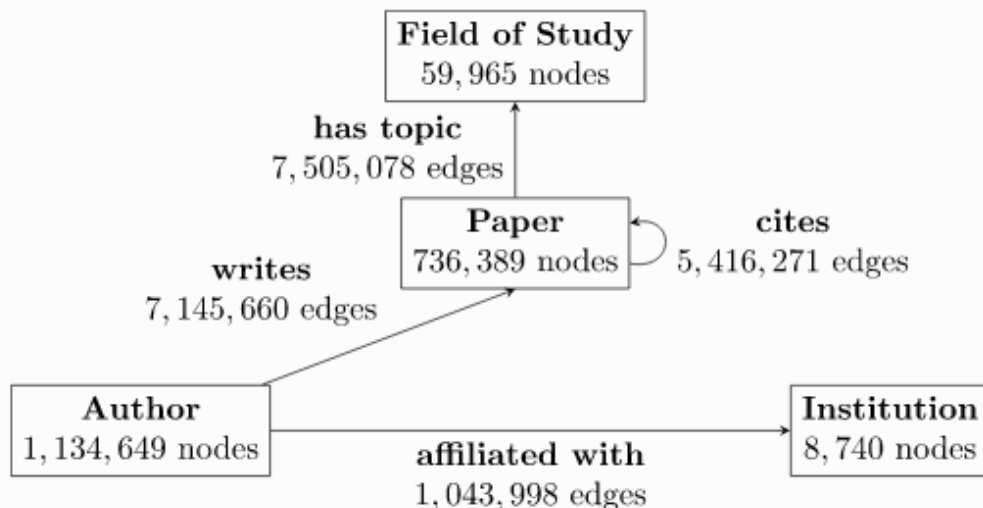
- Complete preprocessing
- Download April 1-5 and merge in
- Attempt first graph using PyG

That's it for today

# Heterogeneous Graph Learning

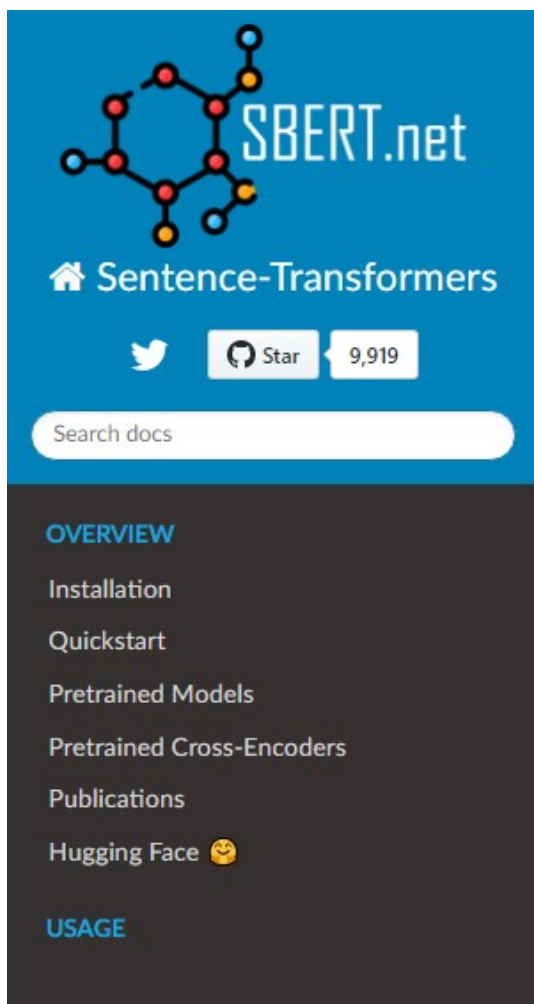
A large set of real-world datasets are stored as heterogeneous graphs, motivating the introduction of specialized functionality for them in 🧠 PyG.

As a guiding example, we take a look at the heterogeneous `ogbn-mag` network from the  dataset suite:



The given heterogeneous graph has 1,939,743 nodes, split between the four node types **author**, **paper**, **institution** and **field of study**. It further has 21,111,007 edges, which also are of one of four types:





## SentenceTransformers Documentation

SentenceTransformers is a Python framework for state-of-the-art sentence, text and image embeddings. The initial work is described in our paper [Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks](#).

You can use this framework to compute sentence / text embeddings for more than 100 languages. These embeddings can then be compared e.g. with cosine-similarity to find sentences with a similar meaning. This can be useful for [semantic textual similar](#), [semantic search](#), or [paraphrase mining](#).

The framework is based on [PyTorch](#) and [Transformers](#) and offers a large collection of [pre-trained models](#) tuned for various tasks. Further, it is easy to [fine-tune your own models](#).

# Interesting Paper

## Graph Isomorphic Network (GIN)

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How Powerful are Graph Neural Networks?

- Given the goal of a GNN is to model a graph, the question is:
  - What are the limits of the expressive power of a GNN architecture?
  - How do we evaluate the expressive power GNN architectures?
- The paper:
- Introduces framework to evaluate expressive power of a given GNN architecture
- Finds certain GNN architectures (GCN, GraphSage) are limited in the types of graph can be represented
- Establishes “a simple architecture that is provably the most expressive among the class of GNNs”

*“Our results characterize the discriminative power of popular GNN variants, such as Graph Convolutional Networks and GraphSAGE, and show that they cannot learn to distinguish certain simple graph structures.”*

*“We then develop a simple architecture that is provably the most expressive among the class of GNNs”*

# Graph Isomorphic Network (GIN)

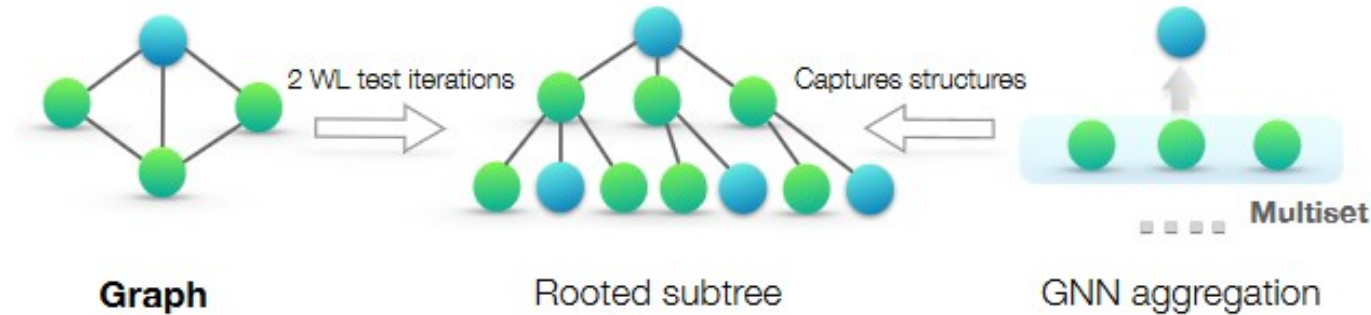


Figure 1: **An overview of our theoretical framework.** Middle panel: rooted subtree structures (at the blue node) that the WL test uses to distinguish different graphs. Right panel: if a GNN's aggregation function captures the *full multiset* of node neighbors, the GNN can capture the rooted subtrees in a recursive manner and be as powerful as the WL test.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How Powerful are Graph Neural Networks? (arXiv:1810.00826). arXiv. <https://doi.org/10.48550/arXiv.1810.00826>

Our framework is inspired by the close connection between GNNs and the Weisfeiler-Lehman (WL) graph isomorphism test (Weisfeiler & Lehman, 1968)

# Graph Isomorphic Network (GIN)

- A node's representation captures the structural information within its k-hop network neighborhood
- For graph classification, the READOUT(\*) function aggregates node features (...) to obtain an entire graph's representation
- “The choice of AGGREGATE(\*) and COMBINE(\*) in GNNs is crucial” (in determining a GNN's expressive power)

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How Powerful are Graph Neural Networks?

[Get Started](#)[Updates](#)[Large-Scale Challenge ▾](#)[Datasets ▾](#)[Leaderboards ▾](#)[Papers ▾](#)[Team](#)[Gi](#)

# Open Graph Benchmark

**Benchmark datasets, data loaders and evaluators for graph machine learning**

[GET STARTED](#)[OGB-LSC @ NEURIPS 2022 \(NEW!\)](#)

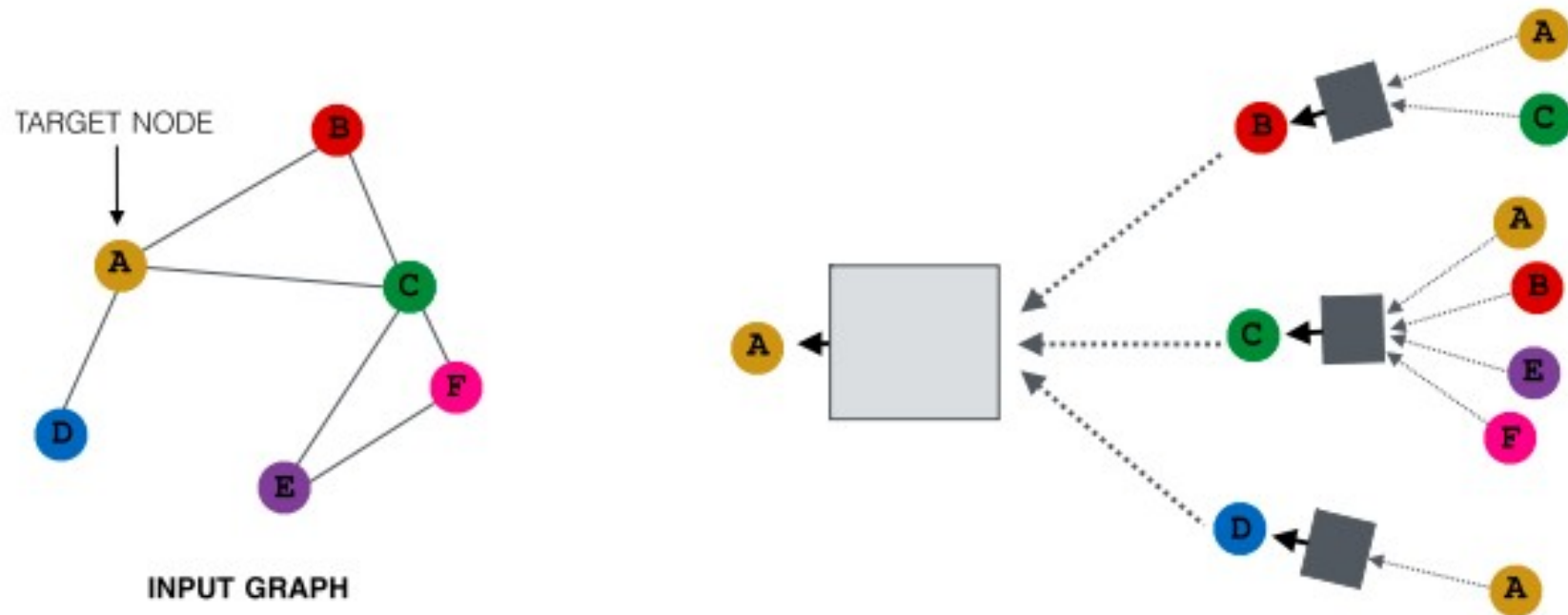
The Open Graph Benchmark (OGB) is a collection of realistic, large-scale, and diverse benchmark datasets for machine learning on graphs. OGB datasets are automatically downloaded, processed, and



# K-hop Neighborhood

- In a K-layer GNN, every node has a “receptive field” of its K-hop neighborhood.
- Two main operations in a GNN layer
  - 1) **Aggregate** messages from all neighboring nodes
  - 2) **Combine** with the previous embedding of the given node.
- Above achieves:
  - GNN can operate on graphs of arbitrary size and shape
  - Independent of ordering of nodes (permutation invariance)

- **Key idea:** Generate node embeddings based on **local network neighborhoods**



# GeoCoV19 Graph Structure

- Set of Graphs (one graph per Original Tweet)
- Nodes:
  - Original Tweet
  - Users
  - Dates
- Vertices
  - Daily Summary to Original Tweet
  - Daily Summary to Daily Summary (prior day)
  - Daily Summary to User IDs (who re-tweeted)



# GeoCoV19 Graph Structure

- Each graph relatively small (order of 100K retweets or less)
- Millions of graphs (one for each original tweet)
- Embeddings:
  - 1) **Aggregate** messages from neighboring nodes
  - 2) **Combine** with the previous embedding of the given node.

# Current Activity: Dataset Creation

- TUDataset (derives from PyG.InMemoryDataset)
- For larger (not in memory) PyG:Dataset additionally requires:
  - Dataset.len(): Returns the number of examples in your dataset.
  - Dataset.get(): Implements the logic to load a single graph.

The data sets have the following format

- `n` = total number of nodes
  - `m` = total number of edges
  - `N` = number of graphs
1. `DS_A.txt` (`m` lines): adjacency matrix for all graphs
  2. `DS_graph_indicator.txt` (`n` lines): column vector all nodes of all graphs, the value in the `graph_id` of the node
  3. `DS_graph_labels.txt` (`N` lines): class labels for all graphs in the data set
  4. `DS_node_labels.txt` (`n` lines): column vector of node labels

There are optional files if the respective information is available:

- `DS_edge_labels.txt` (`m` lines; same size as `DS_A_sparse.txt`): labels for the edges
- `DS_edge_attributes.txt` (`m` lines; same size as `DS_A.txt`): attributes for the edges
- `DS_node_attributes.txt` (`n` lines): matrix of node attributes,
- `DS_graph_attributes.txt` (`N` lines): regression values for all graphs in the data set,

Source: <https://chrsmrrs.github.io/datasets/docs/format/>

# TUDataset Format

That's it for today.

Additional slides on Self-supervised Learning on Graphs...



Paridhi Maheshwari

Jan 18, 2022 · 12 min read · Listen



## Self-Supervised Learning For Graphs

*By Paridhi Maheshwari, Jian Vora, Sharmila Reddy Nangi as part of the Stanford CS 224W course project.*

A large part of deep learning revolves around finding rich representations of unstructured data such as images, text and graphs. Conventional methods try to find these representations using some end goal we want to perform. This is typically done in a supervised setting where we have labelled data. However, in many real-world applications, we do not have labels associated with data, but instead we have an abundance of unlabelled data.

# Self-supervised Learning

- Unlabeled data
- Augmentations create positive and negative pairs (“contrastive learning”)
- Loss function pulls positive pairs together, negative pairs apart
- Fine tune resulting model for downstream tasks

# Augmentations and Graphs

- Node/Embedding prediction – predict if a node should exist and/or it's embeddings
- Edge prediction – predict if an edge should exist
- Graph prediction - is the graph class A or class B
- Augmentation defines the goal of the learning exercise, not necessarily the final task



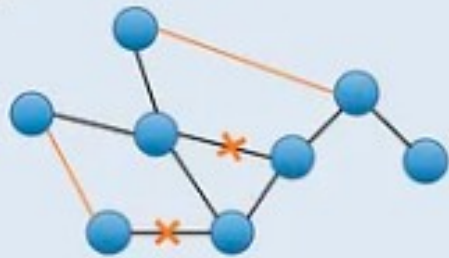
# Augmentations on Graphs

- **Edge Perturbation:** randomly add or remove edges (entries in adjacency matrix) considered "first order"
- **Diffusion:** "a denoising filter (that) allows messages to pass through higher-order neighborhoods" (Hassani et al. [1])
- **Node Dropping:** randomly drop nodes
- **Random Walk based Sampling:** starting from a node randomly walk pre-decided number of nodes creating a sub-graph
- **Node Attribute Masking:** Mask features of some nodes

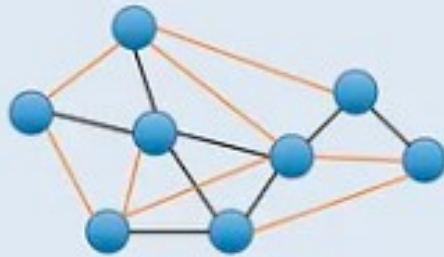
Maheshwari, P. (2022, January 18). Self-Supervised Learning For Graphs. Stanford CS224W GraphML Tutorials.  
<https://medium.com/stanford-cs224w/self-supervised-learning-for-graphs-963e03b9f809>

[1] Hassani, Kaveh, and Amir Hosein Khasahmadi. "Contrastive multi-view representation learning on graphs." International Conference on Machine Learning. PMLR, 2020.

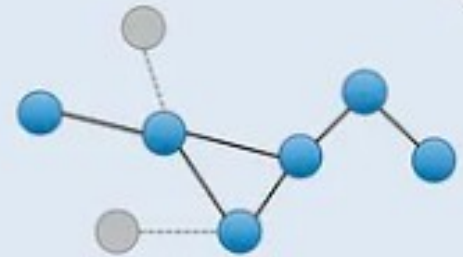
# Augmentations



Edge Perturbation



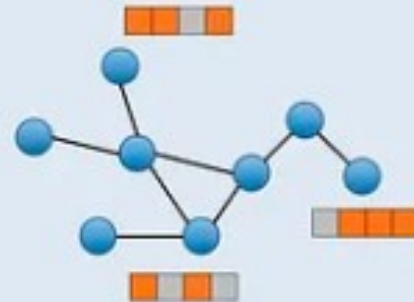
Diffusion



Node Dropping



Random Walk Sampling

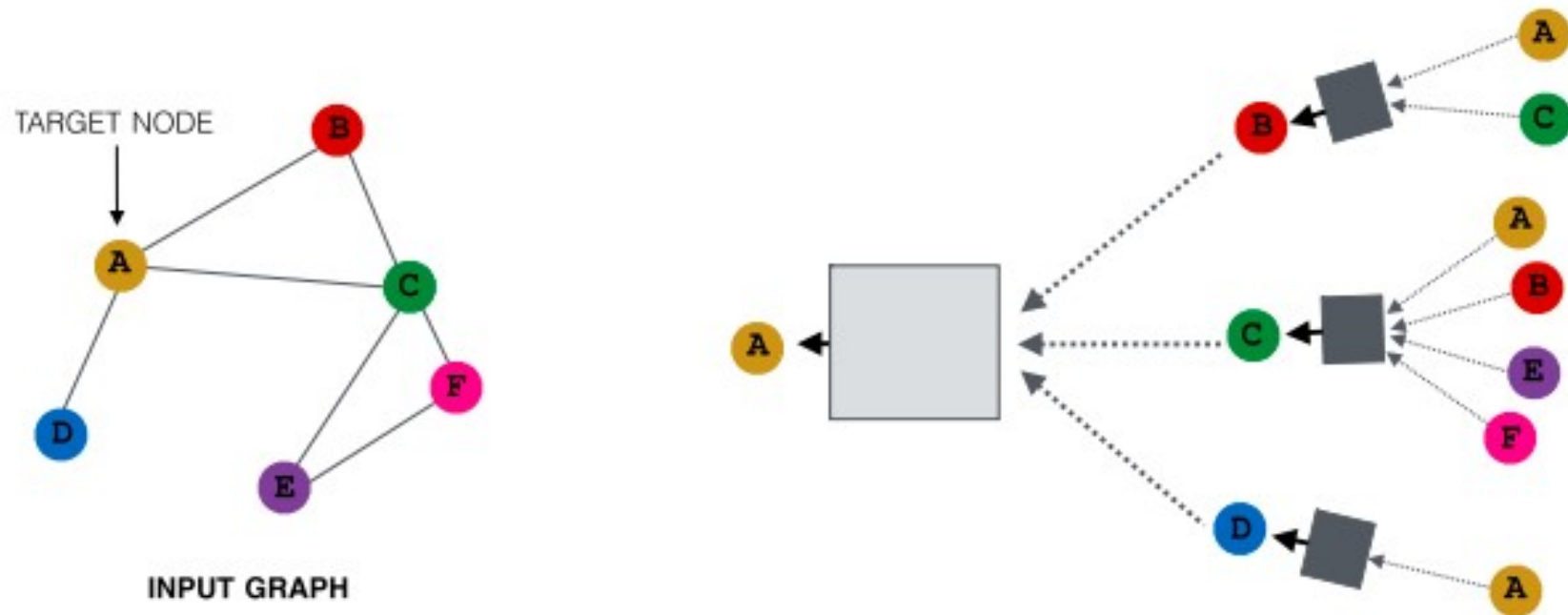


Node Attribute Masking

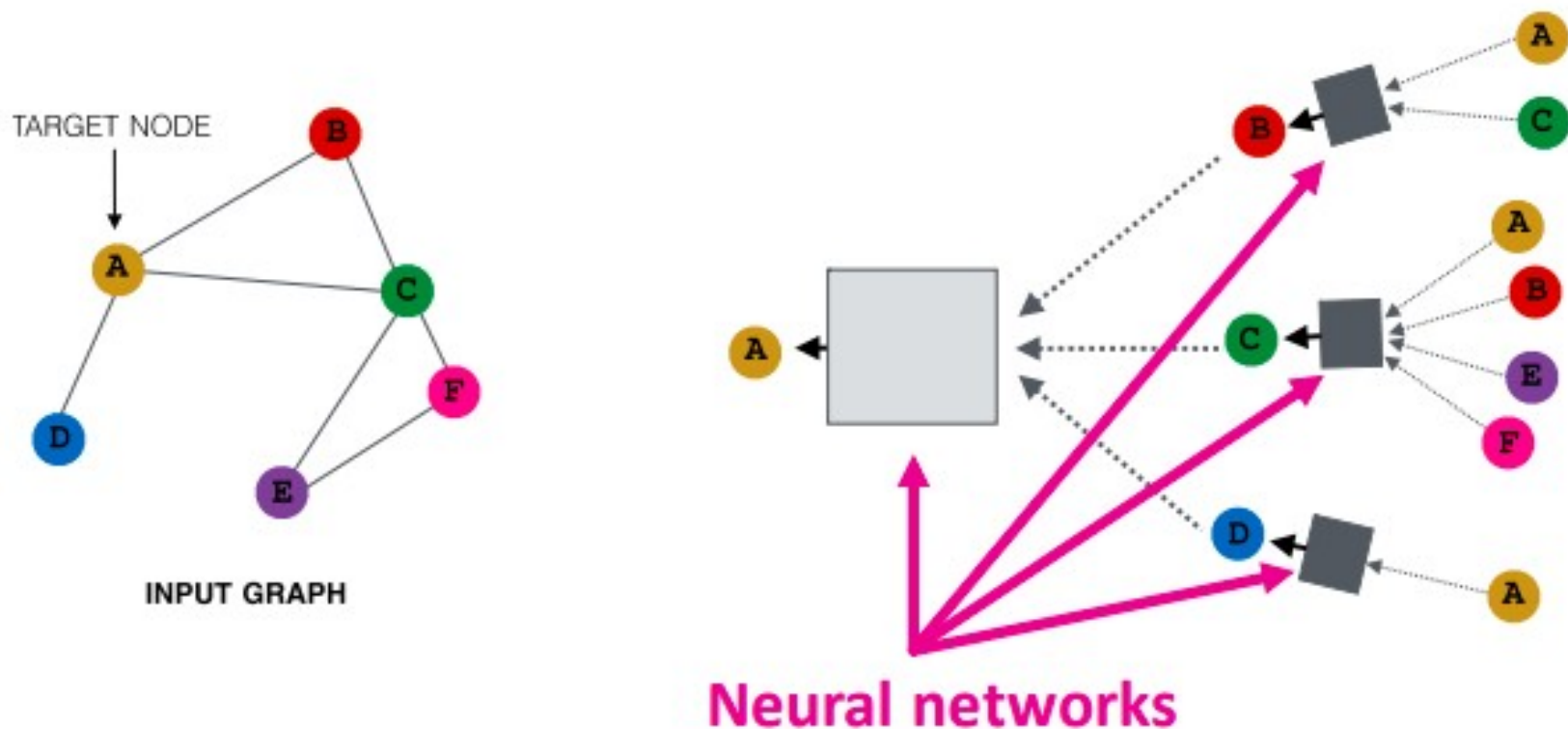
# K-hop Neighborhood

- In a K-layer GNN, every node has a “receptive field” of its K-hop neighborhood.
- Two main operations in a GNN layer
  - 1) **Aggregate** messages from all neighboring nodes
  - 2) **Combine** with the previous embedding of the given node.
- Above achieves:
  - GNN can operate on graphs of arbitrary size and shape
  - Independent of ordering of nodes (permutation invariance)

- **Key idea:** Generate node embeddings based on **local network neighborhoods**



- **Intuition:** Nodes aggregate information from their neighbors using neural networks



# Neural Network vs Graph

- Graph is the Data
- Neural Network is a Computational Mechanism

# Aggregating and Combining

- GraphSAGE [1]
- Graph Convolutional Network [2]
- Graph Attention Network [3]
- Graph Isomorphism Network [4]
- Simple Graph Convolution [5]

1. Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." *Advances in neural information processing systems* 30 (2017).

2. Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907* (2016).

3. Veličković, Petar, et al. "Graph attention networks." *arXiv preprint arXiv:1710.10903* (2017).

4. Xu, Keyulu, et al. "How powerful are graph neural networks?." *arXiv preprint arXiv:1810.00826* (2018).

5. Wu, Felix, et al. "Simplifying graph convolutional networks." *International conference on machine learning*. PMLR, 2019.

# Next Steps

- Preceding gives a Model in PyTorch (layers and "forward" method).
- To train we need Objective Function
- Contrastive loss: positive pairs scored higher than negative pairs



- InfoNCE objective [6]
- Jensen-Shannon Estimator [7]

$$\mathcal{L}_{\text{InfoNCE}} = -\log \left( \frac{e^{d(v, v^+)}}{e^{d(v, v^+)} + \sum_{u \in \{v^-\}} e^{d(v, u)}} \right)$$

where  $d(u, v) = u^\top v / \tau$

and  $\tau$  is the temperature hyperparameter

$$\mathcal{L}_{\text{JS}} = -sp(-d(v, v^+)) - \frac{1}{|\{v^-\}|} \sum_{u \in \{v^-\}} sp(d'(v, u))$$

where  $d'(u, v) = u^\top v$

and  $sp(x) = \log(1 + e^x)$  is the softplus function

6. Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv preprint arXiv:1807.03748* (2018).

7. P. Veličković, W. Fedus, W. L. Hamilton, P. Lio`, Y. Bengio, and D. Hjelm, "Deep graph infomax," in *International Conference on Learning Representations*, 2019.

# Downstream Tasks

- With above we have a model that will produce embeddings that ...
  - Predict nodes and/or Node embedding
  - Edge presence
  - Graph Classification - embed entire graphs s.t. separable given a graph

# MUTAG (<https://chrsmrrs.github.io/datasets/>)

- Each graph is a chemical compound with binary label "mutagenetic effect" (note label)
  - 188 graphs, 18 nodes, 20 edges on average
- 1) Supervised + cross-entropy loss + Adam optimizer
    - 60% accuracy
  - 2) Pretrain w/augmentations (edge perturbation, node dropping) + InfoNCE objective function + fine tune using supervised
    - 75% accuracy.