Graph Neural Network 101

Yihui "Ray" Ren (yren@bnl.gov)
Computational Science Initiative (CSI), BNL

2nd workshop on AI for the Electron Ion Collider
10-14 October 2022, William & Mary / Jefferson Lab

@BrookhavenLab

Brookhaven Supports Data-rich Experimental and Computational Facilities and Programs

Relativistic Heavy Ion Collider (**RHIC**): Supports more than 1000 scientists worldwide

National Synchrotron Light Source II (**NSLS-II**): Newest and brightest synchrotron in the world; supports a multitude of scientific research in academia, industry, and national security

Center for Functional Nanomaterials (**CFN**): Combines theory and experiment to probe materials

Accelerator Test Facility (**ATF**)

Large Hadron Collider (**LHC**) ATLAS: Largest Tier-1 center outside of CERN

Atmospheric Radiation Measurement (**ARM**) program: Partner in multi-site facility, operating its external data center

Belle II: Tier 0 computing for neutrino experiment

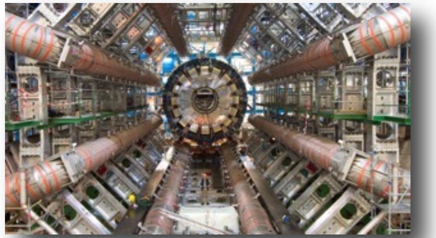
Quantum chromodynamics (**QCD**) computing facilities for Brookhaven Lab, RIKEN, and U.S. QCD communities

RHIC

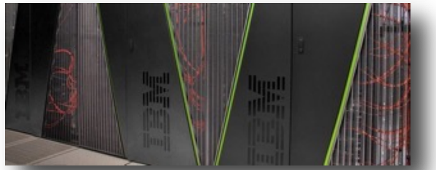
NSLS-II

CFN

ATLAS

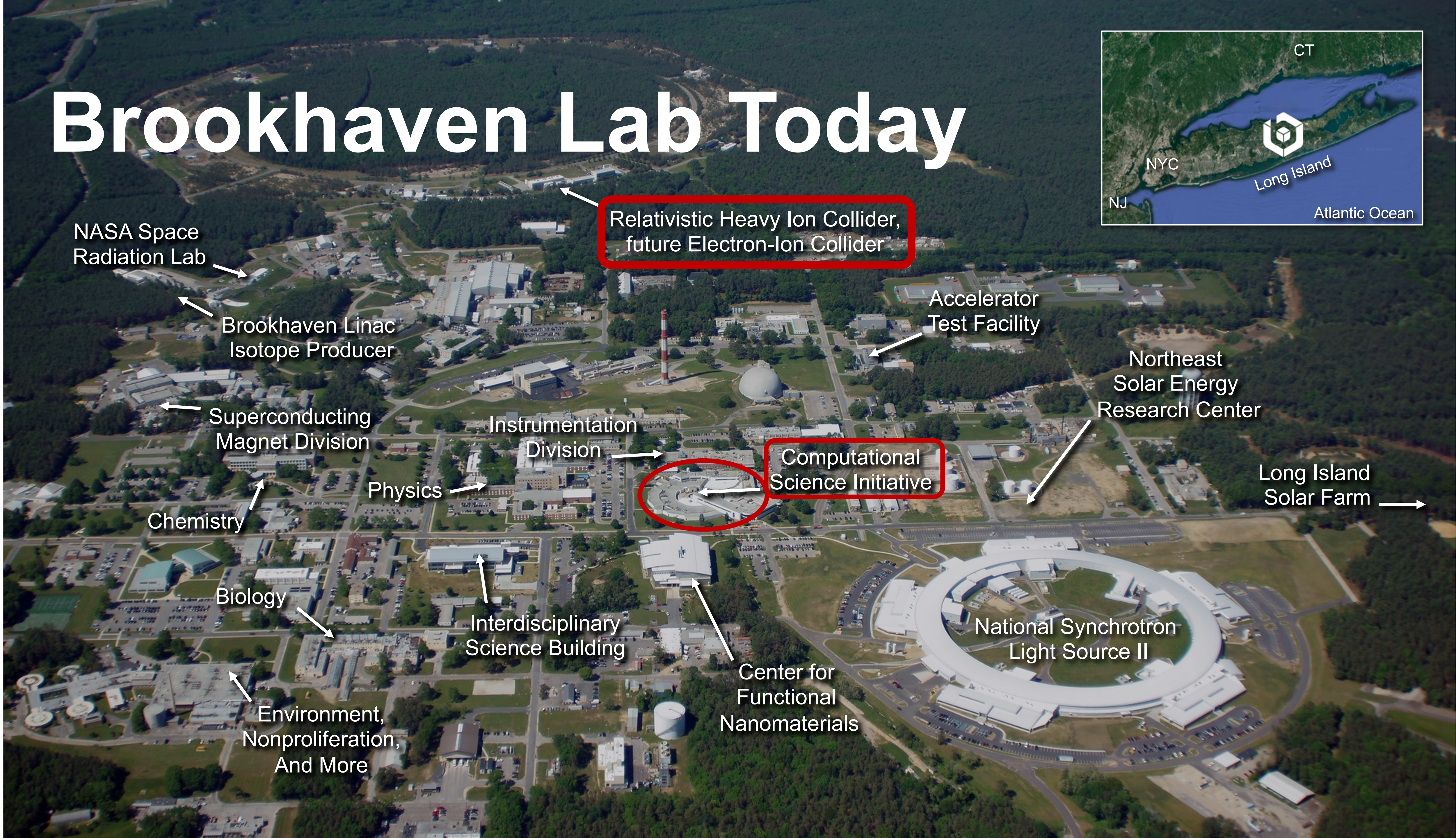


QCD



Brookhaven National Laboratory

Brookhaven Lab Today



Scientific Data and Computing Center

One of the top-10 scientific archives in the world*

- ~215 PB of data archived
- 30 million files injected (26 PB)
- 95 million files restored (196 PB)

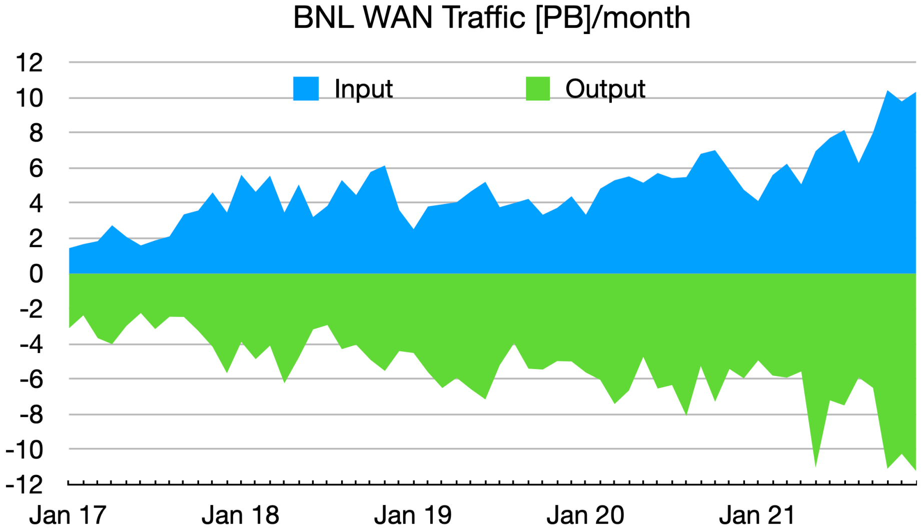
2021 Statistics

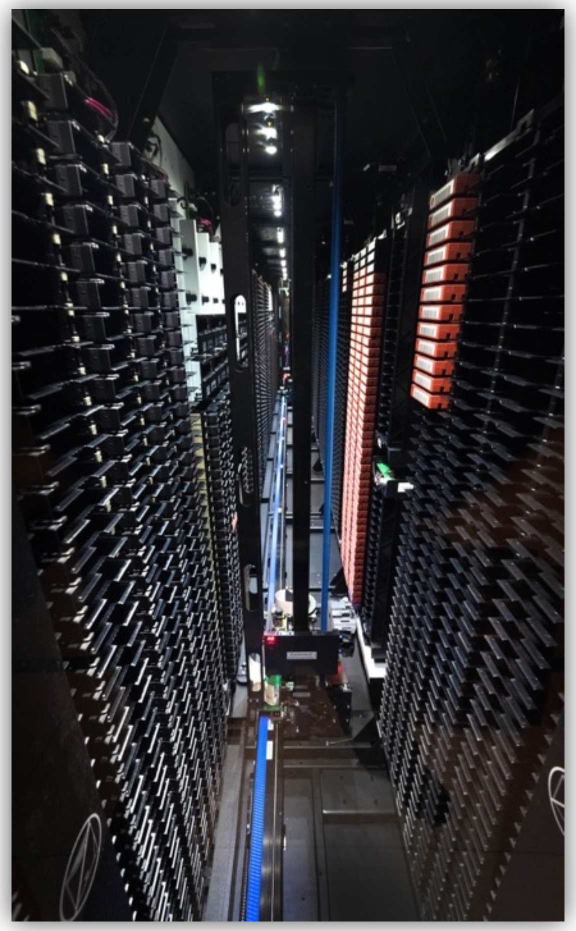
1,750 active accounts

1.1 EB of data analyzed

~180 PB of data transferred

- Data import: 85 PB
- Data export: 95 PB
    - ~30% increase/year





Brookhaven National Laboratory

*Source: http://www.hpss-collaboration.org/customersT.shtml

State-of-the-Art Data Center

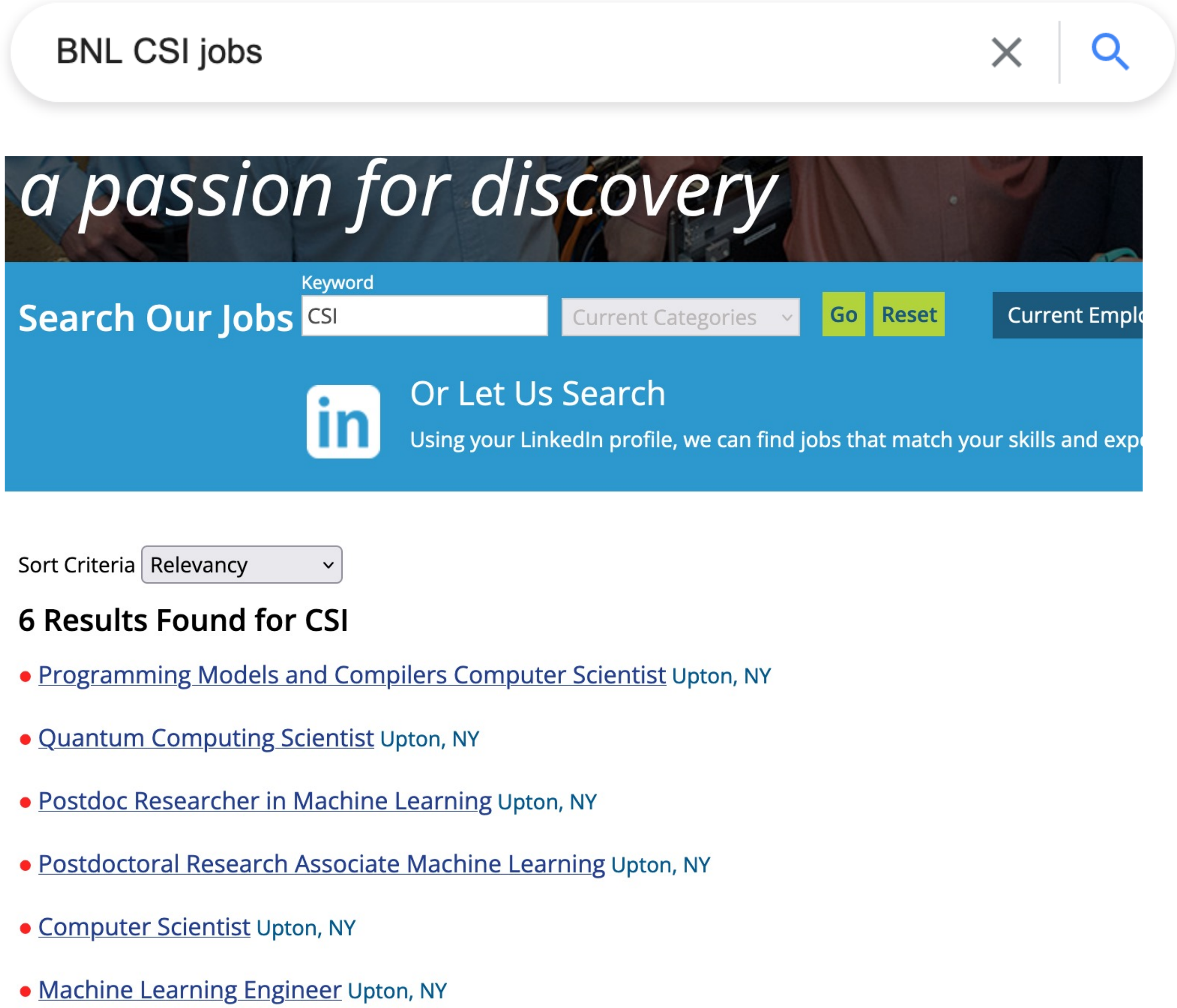
New Infrastructure: **New 60,000 sq-ft^2 Data Center** opened in September 2021

Running community services:

- ATLAS Tier 1 Data Center, Belle II Tier 0 Data Center, RHIC, NSLS-II, CFN, LQCD, IBM-Q Hub

We are hiring

- If you are passionate about computing, programming, or ML.
- Inter-disciplinary research environment.
- We are very diverse.



Outline

- Graph and Its Application (10 min)
- Graph Neural Networks (15 min)
- Code Dive (20 min)

“if I cannot implement it, I cannot say I understand it” – someone, Knuth maybe?

Brookhaven National Laboratory

Graph

A Graph *G* is an ordered pair of disjoint sets *(V, E)* such that *E* is a subset of $V^{(2)}$ of unordered pairs of *V*. *V* is the set of vertices and *E* is the set of edges. -- "Modern Graph Theory, Bela Bollobas"

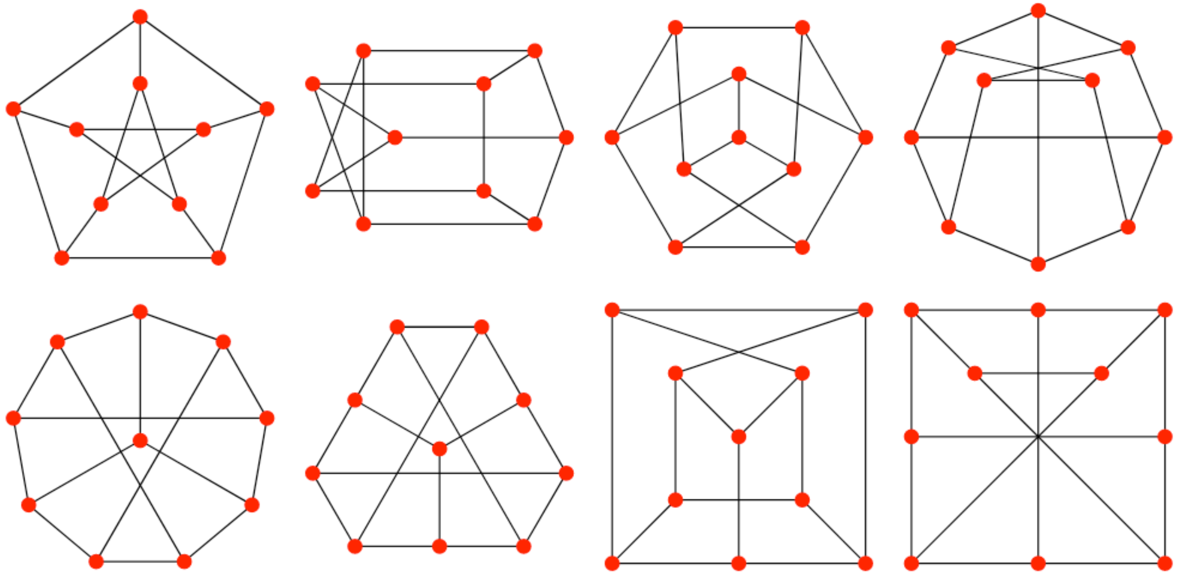


Image credit: https://mathworld.wolfram.com/PetersenGraph.html

Brookhaven National Laboratory

Graph

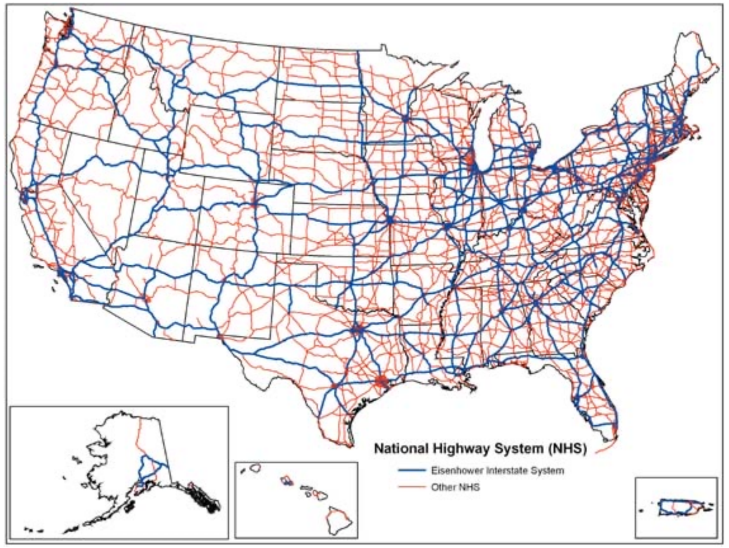
~~A Graph *G* is an ordered pair of disjoint sets *(V, E)* such that *E* is a subset of $V^{(2)}$ of unordered pairs of *V*. *V* is the set of vertices and *E* is the set of edges. -- "Modern Graph Theory, Bela Bollobas"~~

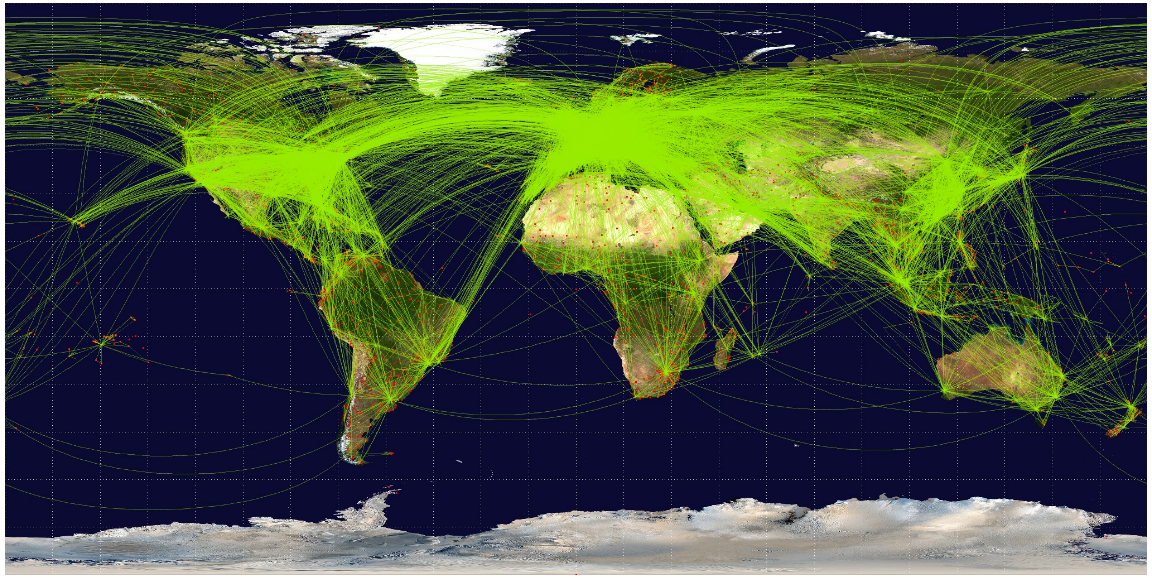
A Graph is just a representation of a connected system.

And here are some examples…

Graph (Network)

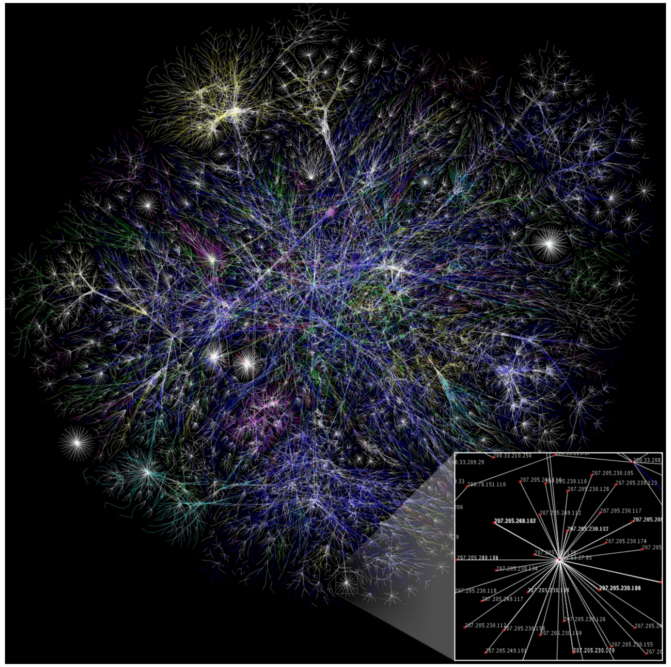
- Transportation Network
  - Roadway:
    - Nodes are intersections
    - Edges are roads
  - Airline:
    - Nodes are airports
    - Edges are routes
  - Global shipping network
  - Subway system
  - …

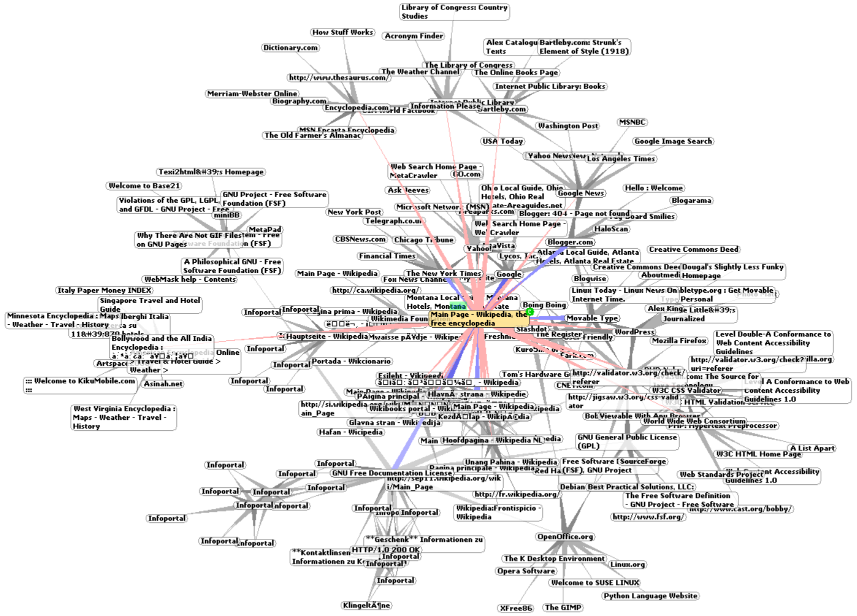




Graph (Network)

- Transportation Network
- Communication Network
  - Internet (TCP/IP):
    - Nodes are terminals and servers
    - Edges are internet connections
  - World Wide Web (WWW)
    - Nodes are web-pages
    - Edges are hyper-links
  - Cellular network
  - Starlink (😎)
  - …





Graph (Network)

- Transportation Network
- Communication Network
- Social Network
  - Collaboration:
    - Nodes are authors
    - Edges are co-authorship
  - Facebook:
    - Nodes are people (and robots)
    - Edges are friendship (perhaps)
  - Contact Network (covid tracing)
  - …

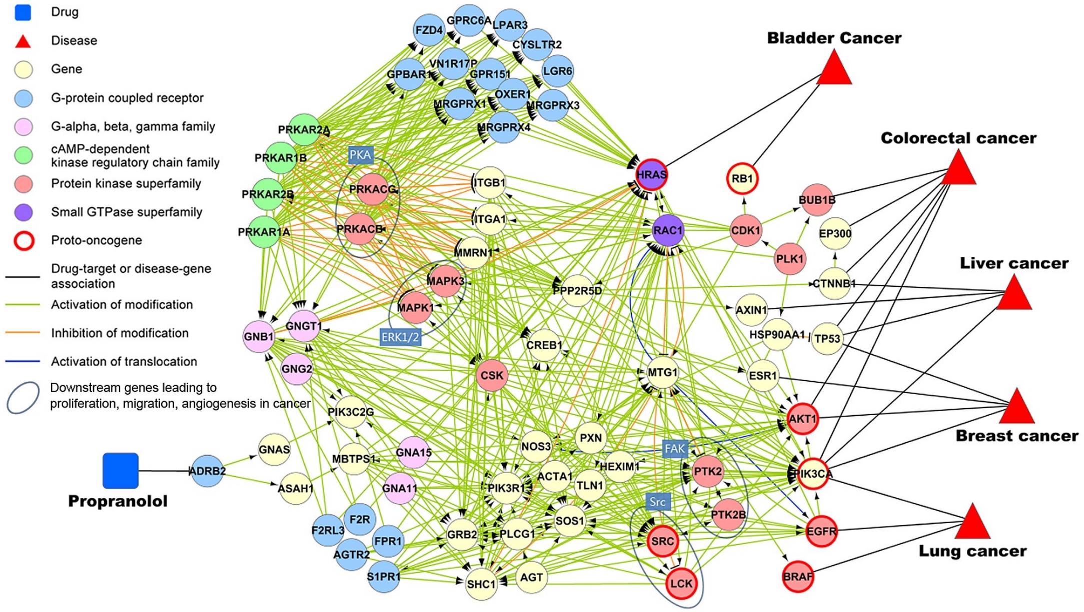
Physicists on ArXiv in 2002 and 2011
Figure credit: https://arxiv.org/abs/1608.03251

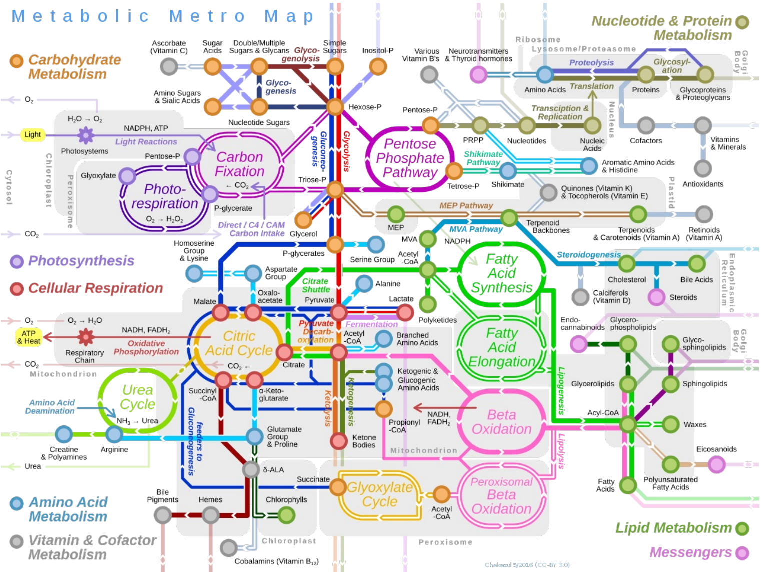
Facebook passed 1bn mark in 2015. Image credit: the Guardian

Brookhaven National Laboratory

Graph (Network)

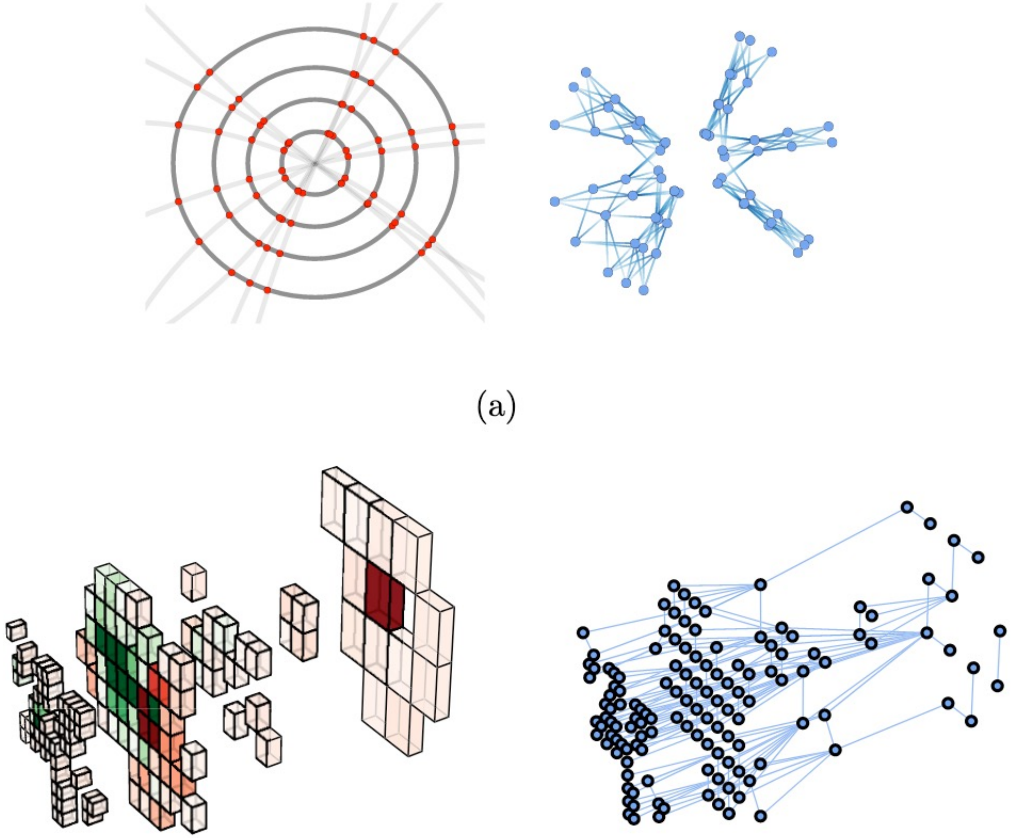
- Transportation Network
- Communication Network
- Social Network
- Biology Network
  - Gene regulatory network
  - Cellular Pathways
  - Metabolic Pathways
  - Molecules (Drugs)





Graph (Network)

- Transportation Network
- Communication Network
- Social Network
- Biology Network
- HEP / NP (Physics)

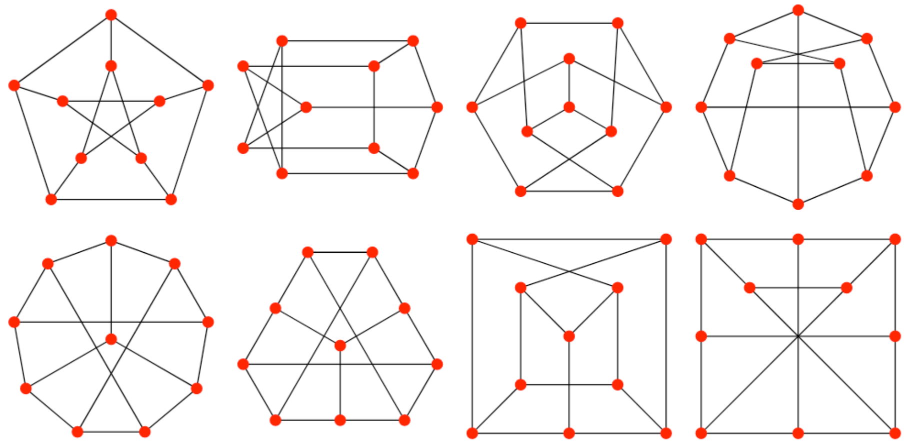


(a)

(b)

Image credit: https://doi.org/10.1088/2632-2153/abbf9a

How to Represent a graph?

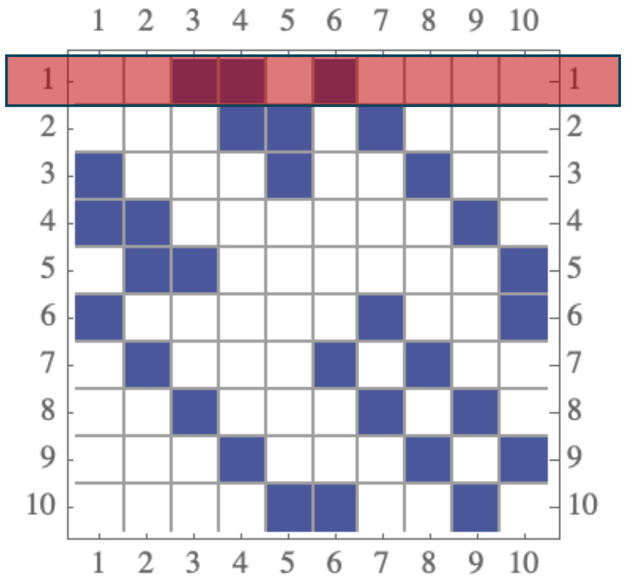


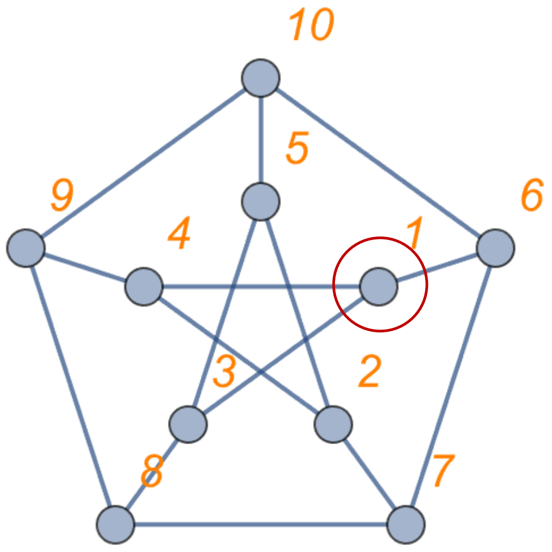
Adjacency Matrix, A .

If there is an edge between i and j, $A_{ij} = 1$. otherwise, $A_{ij} = 0$.

Row- i marks the neighborhood of node i.

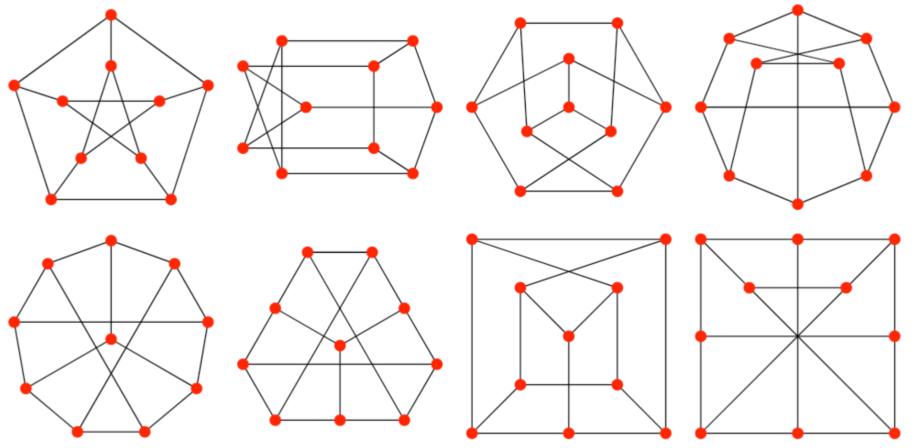
Sum of a row is the number of neighbors, aka, “node degree”.





Brookhaven National Laboratory

How to Represent a graph?

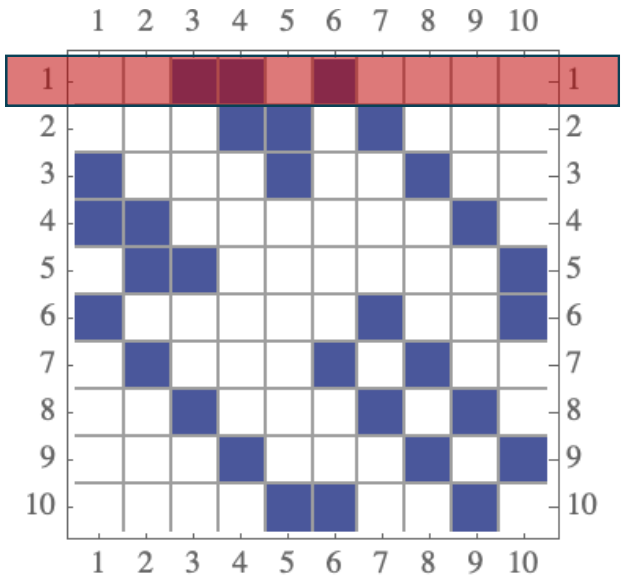


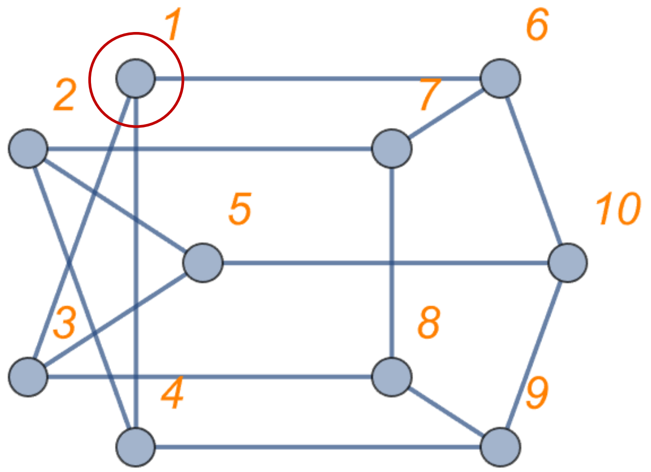
Adjacency Matrix, A .

If there is an edge between i and j, $A_{ij} = 1$. otherwise, $A_{ij} = 0$.

Row- i marks the neighborhood of node i.

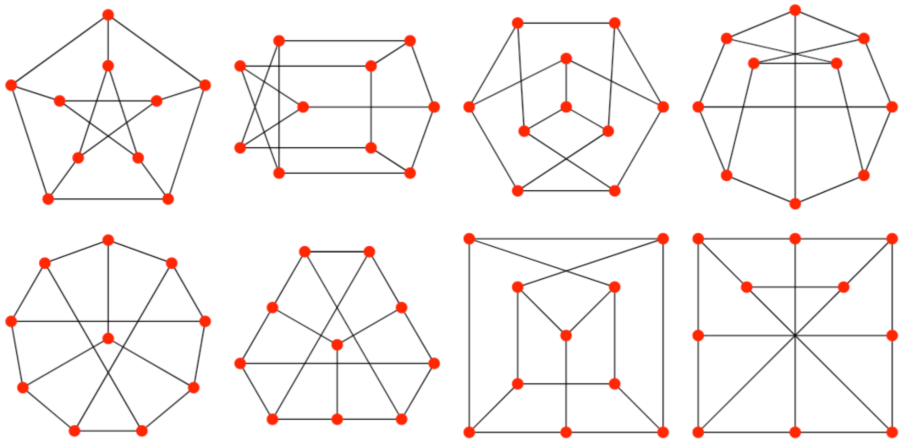
Sum of a row is the number of neighbors, aka, “node degree”.

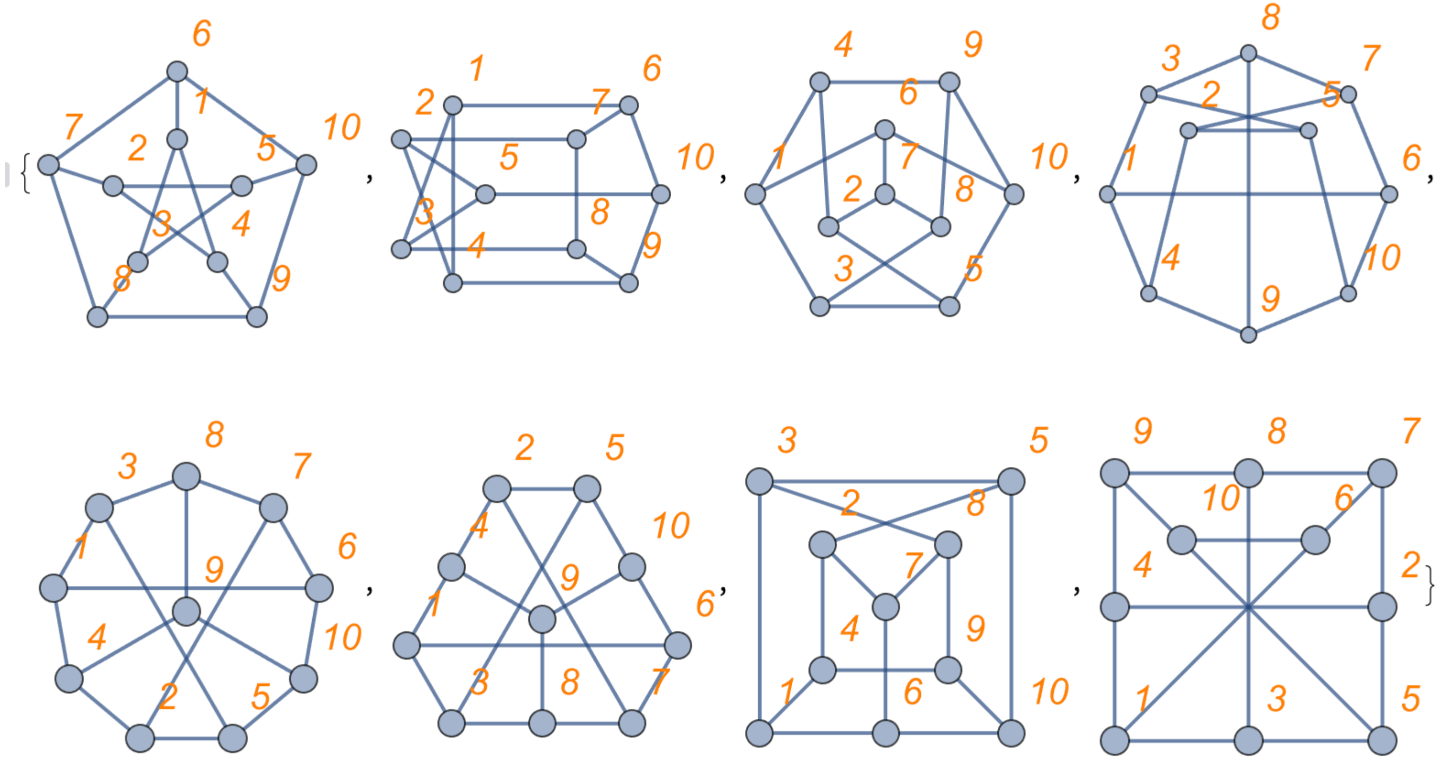


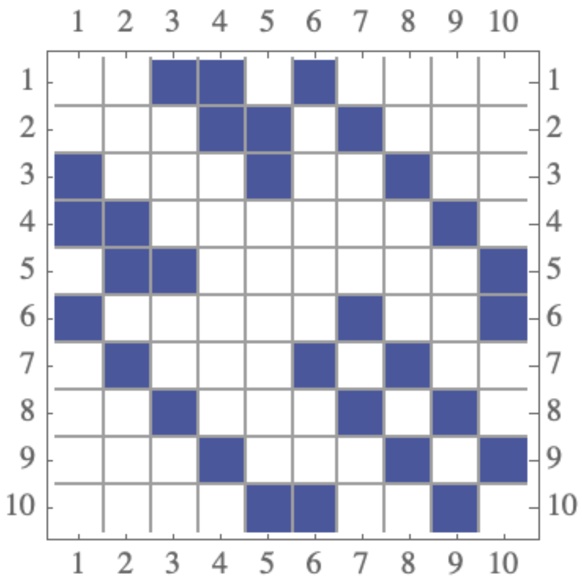


How to Represent a graph?

Adjacent matrix does not depend on how a graph is drawn.







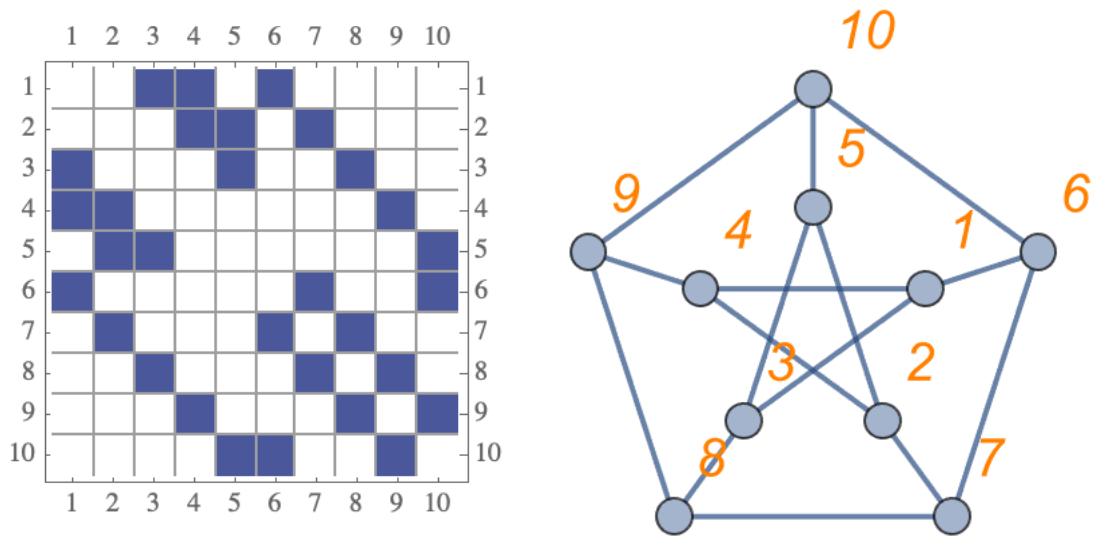
How to Represent a graph?

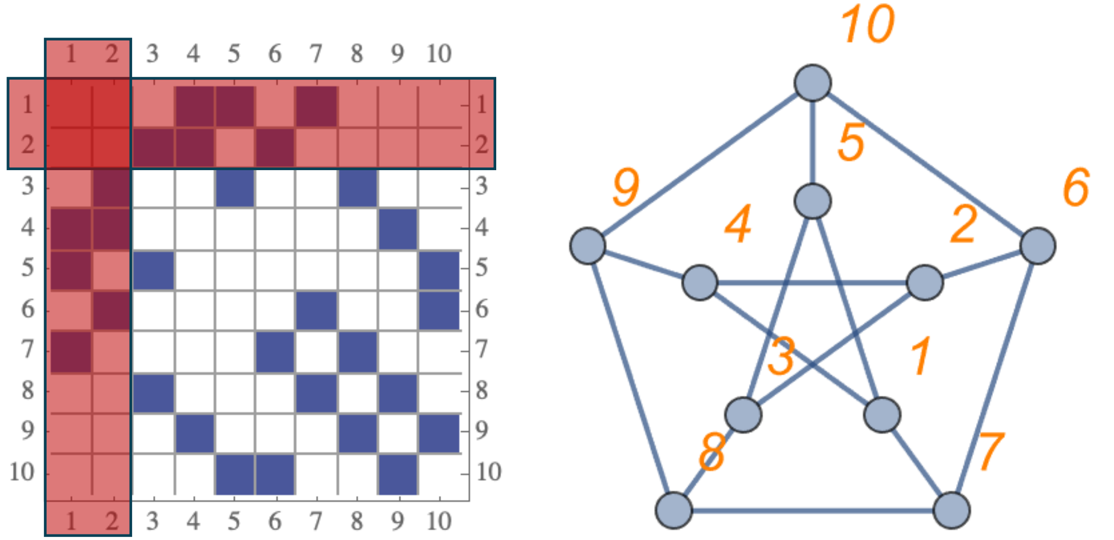
Adjacent matrix does not depend on how a graph is drawn.

But how a graph is labeled.

If we swap the labels of node-1 and node-2, the first and second rows and columns are swapped.

However, the graph is still the same. (Isomorphism)





Brookhaven National Laboratory

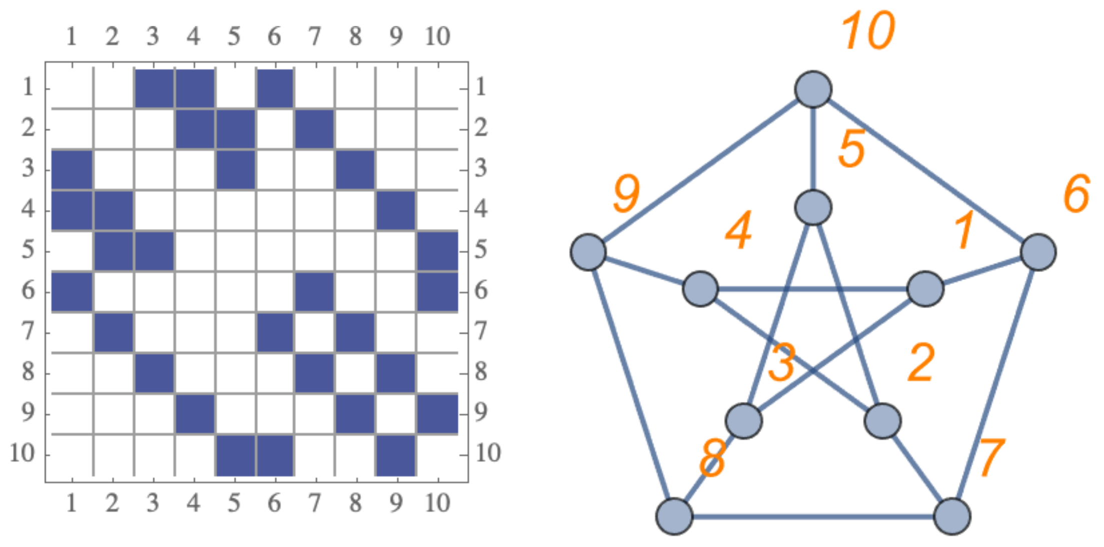
How to Represent a graph?

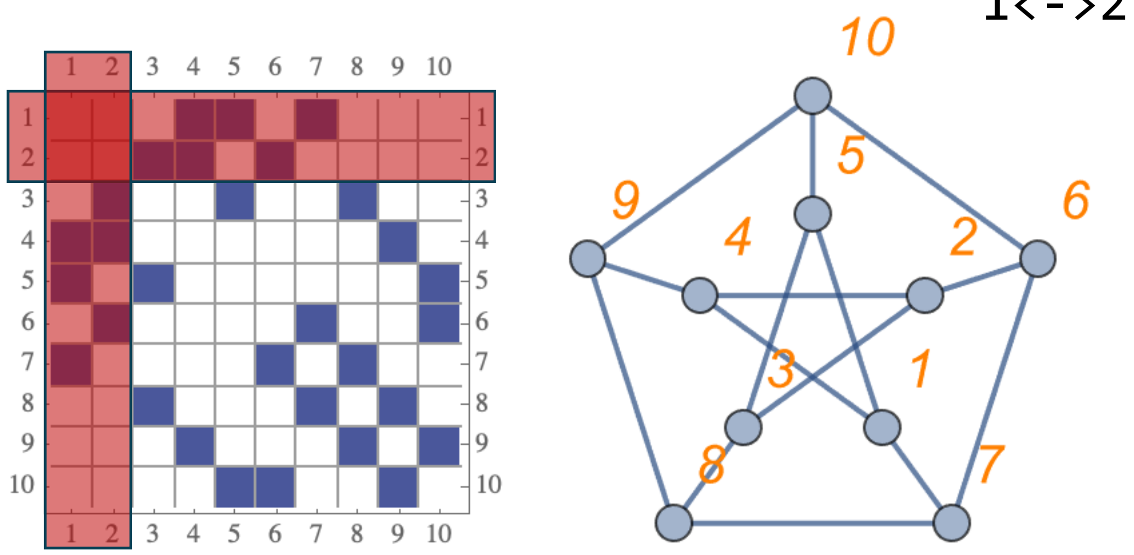
But how a graph is labeled.

If we swap the labels of node-1 and node-2, the first and second rows and columns are swapped.

However, the graph is still the same. (Isomorphism)

=> The ML algorithm should be “permutation invariant” or “equivariant”.





Revisit:
Convolutional Neural Network (CNN)

- CNN applies the same "kernels" on different locations of the input. (an image or activations of previous layer.)
- An image can be viewed as a graph, where near by pixels are connected.

=> Graph Convolution?

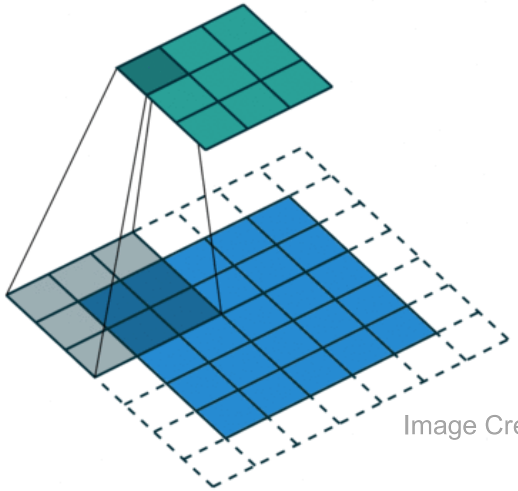


Image Credit: https://github.com/vdumoulin/conv_arithmetic

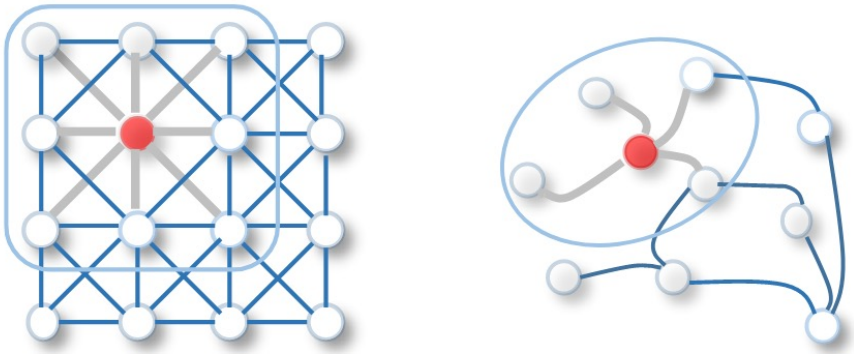


Image Credit: arXiv:1901.00596

GCN in a nutshell

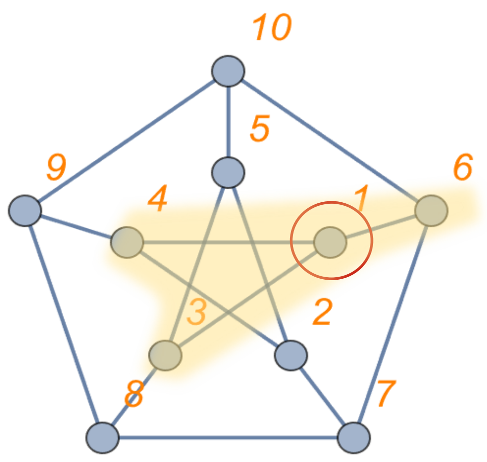
Every node i has a feature vector $\boldsymbol{h}_{\boldsymbol{i}}^{(\boldsymbol{l})}$ of size H_l at layer l.

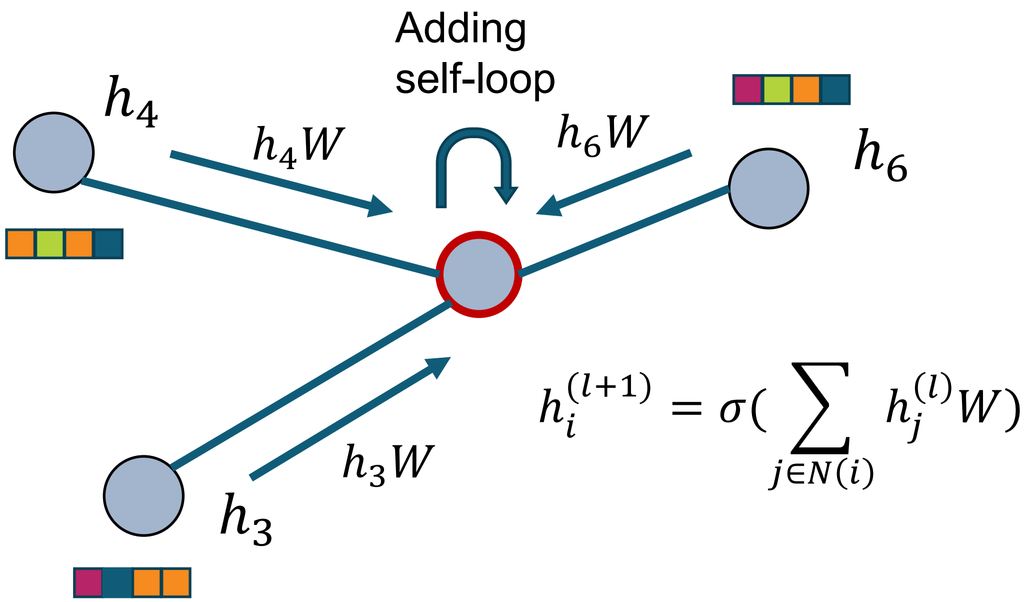
For every node:

- Transform its neighbors' features: $h_j^{(l)} W$
- Aggregate the results and update $h_i^{(l+1)}$ feature.

$W \in R^{H_l \times H_{(l+1)}}$ trainable weights, shared by all the nodes at layer l.

$\sigma(\cdot)$ is some non-linear activation function.





GCN in a nutshell

“Neighborhood” can be obtained by Adjacency Matrix. This node-centric formula can be written in the matrix format:

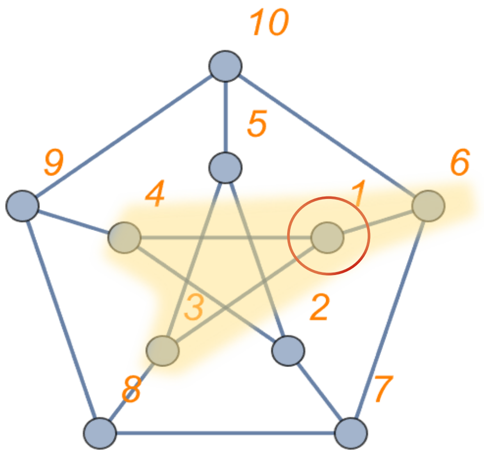
$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W)$$

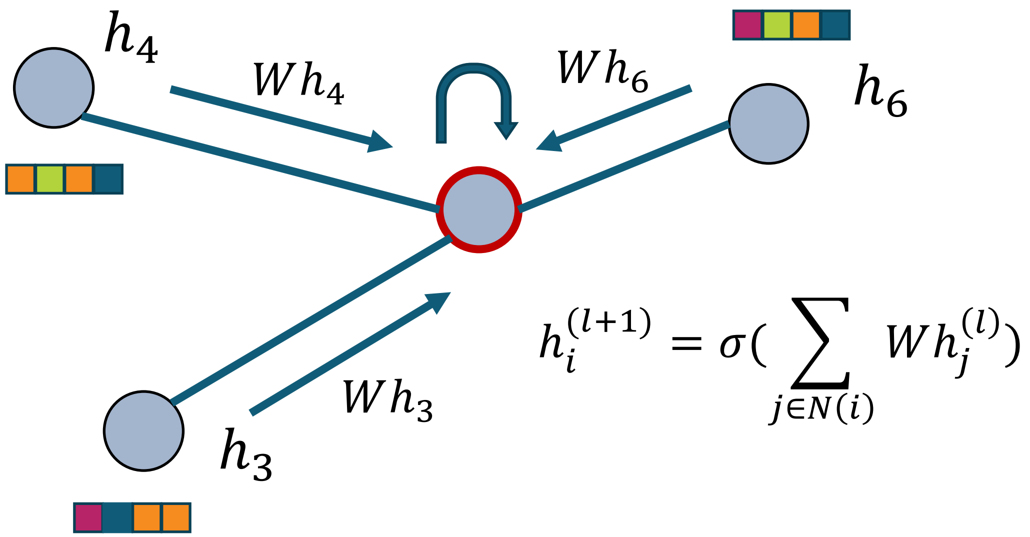
$W \in R^{H_l \times H_{(l+1)}}$

$H^{(l)} \in R^{N \times H_l}$

$\hat{A} \in R^{N \times N}$

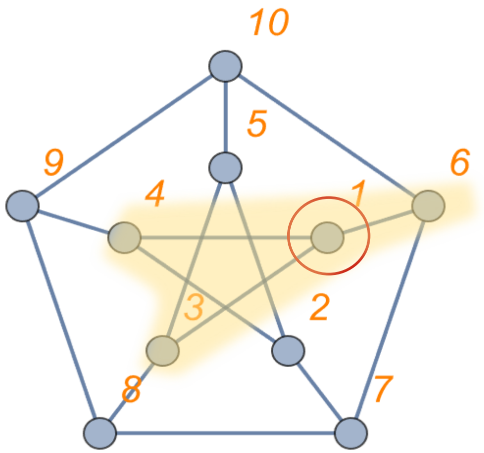
- Added self-loop, $A + I$.
- Normalized by degree*, $\hat{A} = D^{-1}(A + I)$





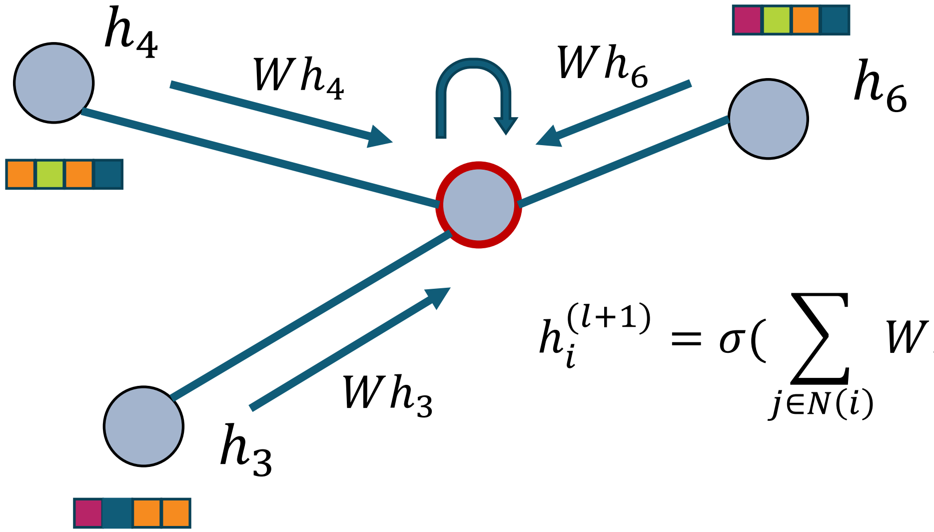
* Simplified normalization.

Efficient Implementation



Usually, $\hat{A}$ can be very sparse, we can use sparse matrix multiplication.

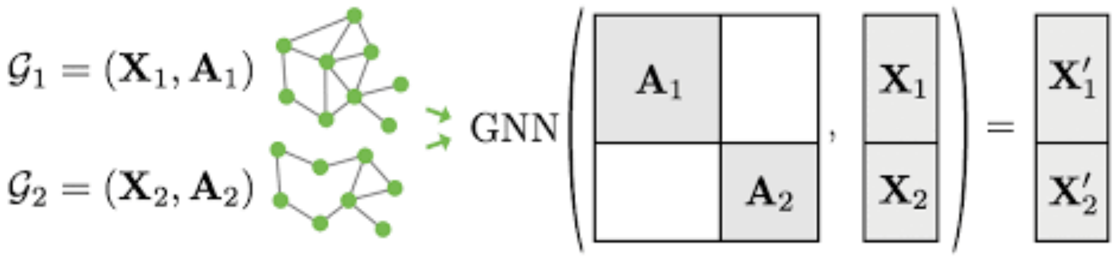
How can one create a mini-batch of graphs with different sizes (orders)?



$$h_i^{(l+1)} = \sigma(\sum_{j \in N(i)} W h_j^{(l)})$$

Efficient Implementation

How can one create a mini-batch of graphs with different sizes (orders)?



"Graph-batching": concatenate graphs into a single adjacency matrix along the diagonal.

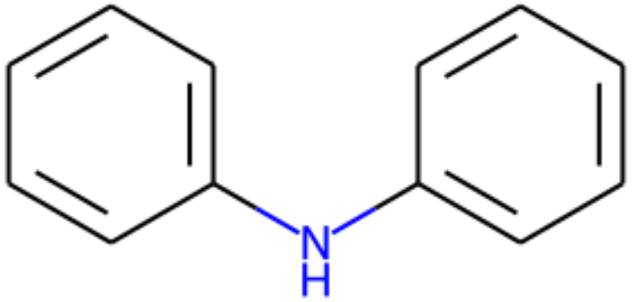
$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_n \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_n \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_n \end{bmatrix}.$$

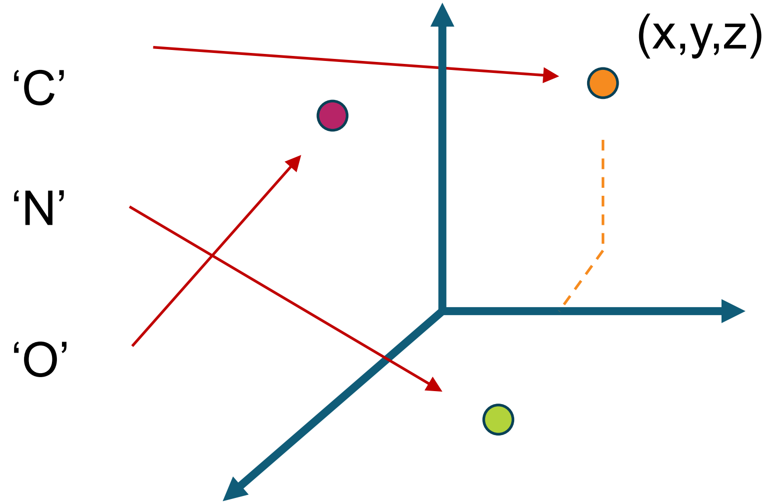
https://pytorch-geometric.readthedocs.io/en/latest/notes/batching.html?highlight=graph%20batching

How to get the node features?

"Embeddings": map intrinsic features into a vector space.

- `torch.nn.Embedding` maps categorical features into learnable representation in a vector space.
- For example, different atom types can be mapped to vectors.





GNN Readout Layer

Multi-layer perceptron (MLP) and Graph Pooling Layer.

- node-level classification / regression.
- graph-level classification / regression.

Code Dive

- Problem setting: predict molecule solubility
- Technology: GCN
- Key points:
    - Construct graphs from molecules
    - Create node features
    - Node embedding
    - Graph batching (`collate_fn`)

https://colab.research.google.com/drive/16fF6q1CSnxnEqRSI7LDAb0evscfqMOrf?usp=sharing

Further Reading

- D. Duvenaud’s early work on GCN.
- T. Kipf’s GCN paper provides a proper degree normalization.
- MPNN generalizes the “message passing” pattern.
- EGNN uses node coordinates and equivariant to rotation, permutation, etc.

Convolutional Networks on Graphs for Learning Molecular Fingerprints

arXiv:1509.09292

David Duvenaud†, Dougal Maclaurin†, Jorge Aguilera-Iparraguirre
Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, Ryan P. Adams
Harvard University

SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

arXiv:1609.02907

Thomas N. Kipf
University of Amsterdam
T.N.Kipf@uva.nl

Max Welling
University of Amsterdam
Canadian Institute for Advanced Research (CIFAR)
M.Welling@uva.nl

Neural Message Passing for Quantum Chemistry

arXiv:1704.01212

Justin Gilmer [1] Samuel S. Schoenholz [1] Patrick F. Riley [2] Oriol Vinyals [3] George E. Dahl [1]

E(n) Equivariant Graph Neural Networks

arXiv:2102.09844

Victor Garcia Satorras [1] Emiel Hoogeboom [1] Max Welling [1]