Stanford CS224W: Machine Learning with Graphs Fall 2023/24

CS224W: Machine Learning with Graphs
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

Stanford CS224W: Course Logistics

CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

http://cs224w.stanford.edu

CS224W Course Outline

We are going to explore Machine Learning and Representation Learning for graph data:

- Methods for node embeddings: DeepWalk, Node2Vec
- Graph Neural Networks: GCN, GraphSAGE, GAT…
- Graph Transformers
- Knowledge graphs and reasoning: TransE, BetaE
- Generative models for graphs: GraphRNN
- Graphs in 3D: Molecules
- Scaling up to large graphs
- Applications to Biomedicine, Science, Technology

CS224W Course Outline

Date	Topic	Date	Topic
Tue, 9/26	1. Introduction to Machine Learning for Graphs	**Tue, 10/31**	11. GNNs for Recommenders
Thu, 9/27	2. Node Embeddings	**Thu, 11/2**	12. Deep Generative Models for Graphs
Tue, 10/3	3. Graph Neural Networks	**Tue, 11/7**	13. Advanced Topics in GNNs
Thu, 10/5	4. Building blocks of GNNs	**Thu, 11/9**	14. Graph Transformers
Tue, 10/10	5. GNN augmentation and training	**Tue, 11/14**	15. Scaling up GNNs
Thu, 10/12	6. Theory of GNNs	**Thu, 11/16**	16. Geometric Deep Learning
Tue, 10/17	7. Heterogenous graphs	**Tue, 11/28**	17. Link Prediction and Causality
Thu, 10/19	8. Knowledge Graph Completion	**Thu, 11/30**	18. Frontiers of GNN Research
Tue, 10/24	9. Complex Reasoning in KGs	**Tue, 12/5**	19. Algorithmic reasoning with GNNs
Thu, 10/26	10. Fast Neural Subgraph Matching	**Thu, 12/7**	20. Conclusion

Prerequisites

- **The course is self-contained.**
- **No single topic is too hard by itself.**
- **But we will cover and touch upon many topics and this is what makes the course hard.**
  - **Some background in:**
    - Machine Learning
    - Algorithms and graph theory
    - Probability and statistics
  - **Programming:**
    - You should be able to write non-trivial programs (in Python)
    - Familiarity with PyTorch is a plus

Graph Machine Learning Tools

- **We use PyG (PyTorch Geometric):**
  - The ultimate library for Graph Neural Networks
- **We further recommend:**
  - GraphGym: Platform for designing Graph Neural Networks.
    - Modularized GNN implementation, simple hyperparameter tuning, flexible user customization
  - Both platforms are very helpful for the course project (save your time & provide advanced GNN functionalities)
- **Other network analytics tools**: SNAP.PY, NetworkX

CS224W Course Logistics

- **The class meets Tue and Thu 3:00-4:20pm Pacific Time *in person***
  - Videos of the lectures will be recorded and posted on Canvas
- **Structure of lectures:**
  - ~80 minutes of a lecture
    - During this time you can ask questions
  - ~10 minutes of a live Q&A/discussion session at the end of the lecture

Logistics: Teaching Staff

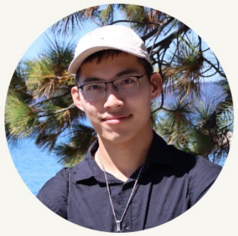
Instructor

Jure Leskovec

Guest Instructor

Joshua Robinson

Course Assistants



Xikun Zhang
Head CA

Hamed Nilforoshan

Aditya Agrawal

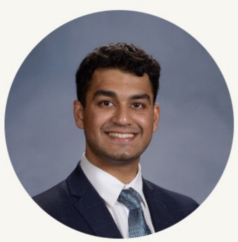
Abhinav Garg

Matthew Jin

Yunqi Li

Tolu Oyeniyi

Chenshu (Jupiter) Zhu



Pratham Soni

Anirudh Sriram

Logistics: Website

- http://cs224w.stanford.edu
  - Slides posted before the class
- **Readings:**
  - Graph Representation Learning Book by Will Hamilton
  - Research papers
- **Optional readings:**
  - Papers and pointers to additional literature
  - **This will be very useful for course projects**

Logistics: Communication

- **Ed Discussion:**
  - Access via link on Canvas
  - **Please participate and help each other!**
    - Don't post code, annotate your questions, search for answers before you ask
  - We will post course announcements to Ed (make sure you check it regularly)
- **Please don't communicate with prof/TAs via personal emails, but always use:**
  - cs224w-aut2324-staff@lists.stanford.edu

Logistics: Office Hours

- **OHs will be both in person and virtual**
  - We will have OHs every day, starting from 2nd week of the course
  - See http://web.stanford.edu/class/cs224w/oh.html for Zoom links and link to QueueStatus
  - Schedule to be announced by end of week

Work for Course: Grading

- **Final grade will be composed of:**
  - **Homework: 20%**
    - 3 written homeworks, each worth 6.67%
  - **Coding assignments: 15%**
    - 5 coding assignments using Google Colab, each worth 3%
  - **Exam: 35%**
  - **Course project: 30%**
    - Proposal, Milestone, and Final report
  - **Extra credit: Ed participation, PyG/GraphGym code contribution**
    - Used if you are on the boundary between grades

Work for Course: Submitting

- **How to submit?**
  - **Upload via Gradescope**
    - You will be automatically registered to Gradescope once you officially enroll in CS224W
  - Homeworks, Colabs (numerical answers), and project deliverables are submitted on Gradescope
- **Total of <u>2 Late Periods (LP)</u> per student**
  - Max 1 LP per assignment (no LP for the final report)
    - LP gives **4 extra days**: assignments usually due on Thursday (11:59pm) → with LP, it is due the following Monday (11:59pm)

Work for Course: HWs, Colabs

- **Homeworks (20%, n=3)**
  - **Written assignments take longer and take time (~10-20h) – start early!**
    - A combination of theory, algorithm design, and math
- **Colabs (15%, n=5)**
  - **We have more Colabs but they are shorter (~3-5h); Colab 0 is not graded.**
    - Get hands-on experience coding and training GNNs; good preparation for final projects and industry

Work for Course: Exam

- **Single exam: Wednesday, Nov 29 (35%)**
  - **Take-home, open-book, timed**
    - Administered via Gradescope
    - Released at 5 PM PT on Wednesday, Nov 29, available until 5 AM PT on Friday, Dec 1.
    - Once you open it, you will have 120 minutes to complete the exam.
  - **Content**
    - Will have written questions (similar to Homeworks), Will possibly have a coding section (similar to Colabs)
    - More details to come!

Work for Course: Project (30%)

- **Details will be posted soon:**
  - Focus is on real-world applications of GNNs
- **Logistics**
  - **Groups of up to 3 students**
    - Groups of 1 or 2 are allowed (but discouraged); the team size will be taken under consideration when evaluating the scope of the project. But 3 person teams can be more efficient.
  - **Google Cloud credits**
    - We will provide $50 in Google Cloud credits to each student
    - You can also get $300 with Google Free Trial (https://cloud.google.com/free/docs/gcp-free-tier)
- **Read: http://cs224w.stanford.edu/info.html**

Course Schedule

Assignment	Due on (11:59pm PT)
Colab 0	Not graded
Colab 1	Thu, 10/12 (week 3)
Project Proposal	Tue, 10/17 (week 4)
Homework 1	Thu, 10/19 (week 4)
Colab 2	Thu, 10/26 (week 5)
Homework 2	Thu, 11/2 (week 6)
Colab 3	Thu, 11/9 (week 7)
Project Milestone	Thu, 11/9 (week 7)
Homework 3	Thu, 11/16 (week 8)
EXAM	Wed, 11/29 5pm – Fri, 12/1 5am (week 9)
Colab 4	Thu, 11/30 (week 9)
Colab 5	Tue, 12/5 (week 10)
Project Report	Thu, 12/14 **(No Late Periods!)**

Honor Code

Make sure you read and understand it!

- **We strictly enforce the Stanford Honor Code**
  - Violations of the Honor Code include:
    - Copying or allowing another to copy from one's own paper
    - Unpermitted collaboration
    - Plagiarism
    - Giving or receiving unpermitted aid on a take-home examination
    - Representing as one's own work the work of another
    - Giving or receiving aid on an assignment under circumstances in which a reasonable person should have known that such aid was not permitted
  - The standard sanction for a first offense includes a one-quarter suspension and 40 hours of community service.

Course Logistics: Q&A

Two ways to ask questions during lecture:

- **In-person (encouraged)**
- **On Ed:**
  - At the beginning of class, we will open a new discussion thread dedicated to this lecture
  - When to ask on Ed?
    - If you have a minor clarifying question
    - If we run out of time to get to your question live
    - **Otherwise, try raising your hand first!**

Course Logistics: Colab 0

- **Colabs 0 and 1 will be released on our course website at 3pm Thursday (9/28)**
- **Colab 0:**
  - Does not need to be handed-in
- **Colab 1:**
  - Due on Thursday 10/12 (2 weeks from today)
  - Submit written answers and code on Gradescope
  - Will cover material from Lectures 1-4, but you can get started right away!

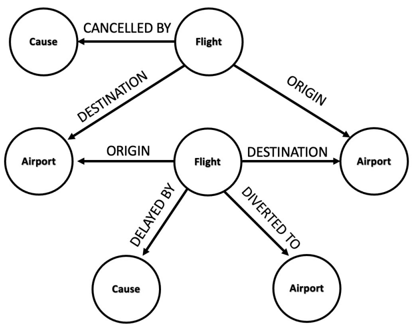
Stanford CS224W: Machine Learning with Graphs

CS224W: Machine Learning with Graphs
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

Why Graphs?

Graphs are a general language for describing and analyzing entities with relations/interactions

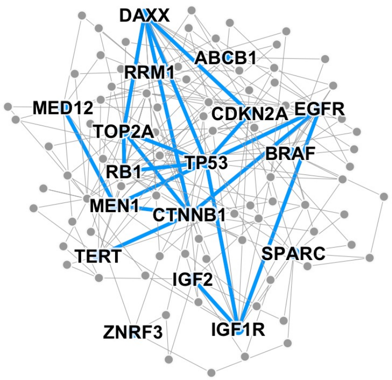
Many Types of Data are Graphs (1)



Event Graphs

Image credit: SalientNetworks

Computer Networks



Disease Pathways

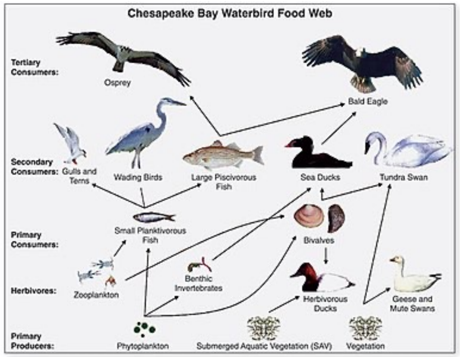


Image credit: Wikipedia

Food Webs

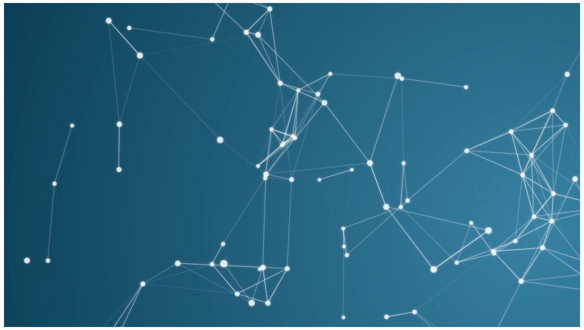


Image credit: Pinterest

Particle Networks

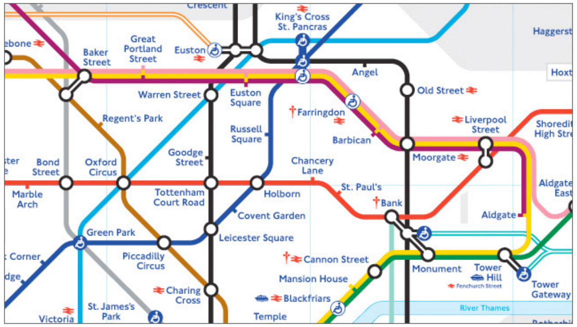


Image credit: visitlondon.com

Underground Networks

Many Types of Data are Graphs (2)

Image credit: Medium

Social Networks

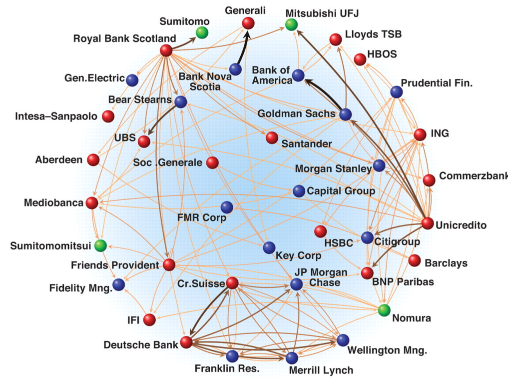
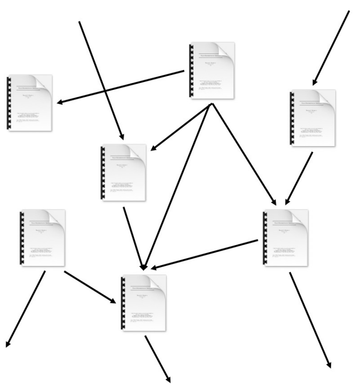


Image credit: Science

Economic Networks

Image credit: Lumen Learning

Communication Networks



Citation Networks

Image credit: Missoula Current News

Internet

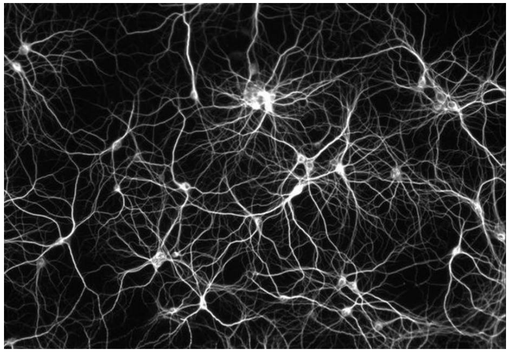


Image credit: The Conversation

Networks of Neurons

Many Types of Data are Graphs (3)

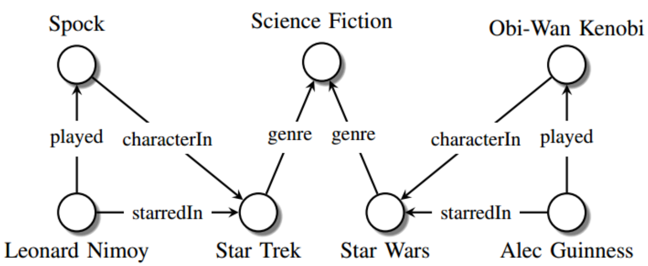


Image credit: Maximilian Nickel et al

Knowledge Graphs

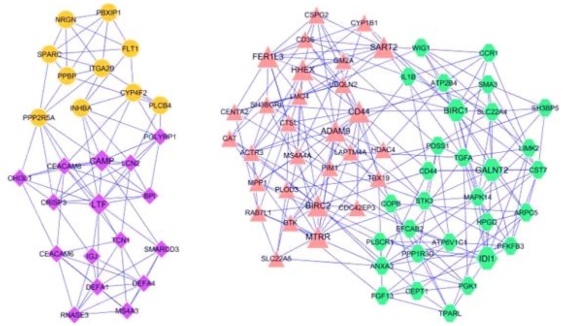


Image credit: ese.wustl.edu

Regulatory Networks

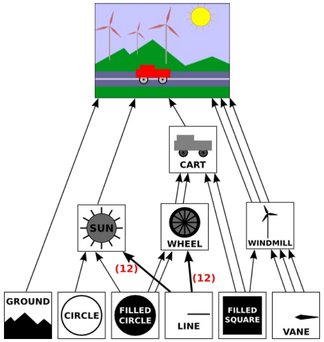


Image credit: math.hws.edu

Scene Graphs

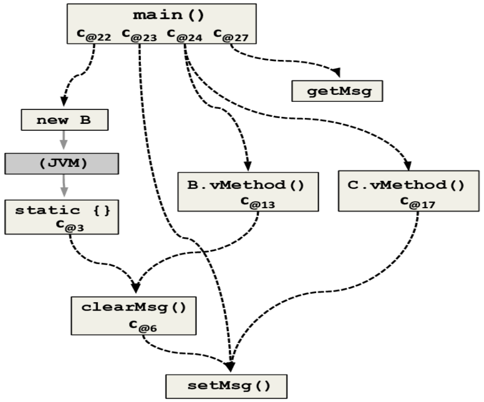


Image credit: ResearchGate

Code Graphs

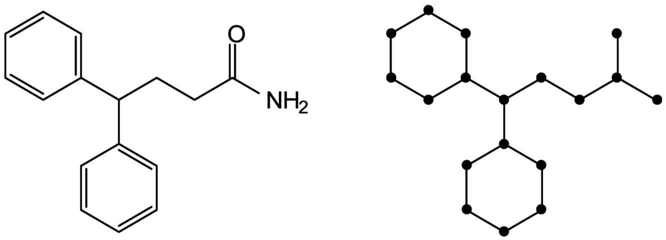


Image credit: MDPI

Molecules

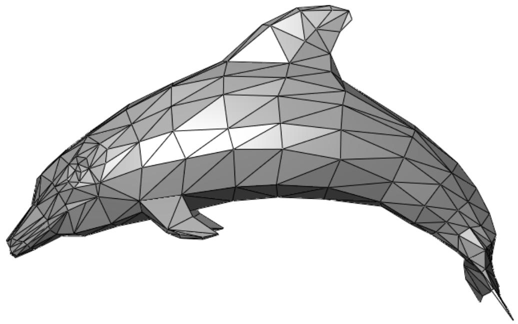


Image credit: Wikipedia

3D Shapes

Graphs: Machine Learning

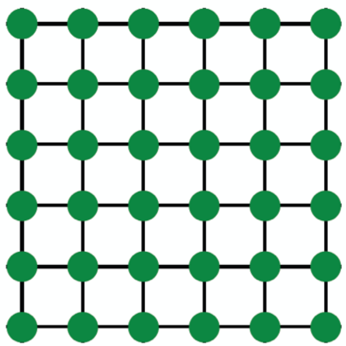
Complex domains have a rich relational structure, which can be represented as a **relational graph**

By explicitly modeling relationships we achieve better performance!

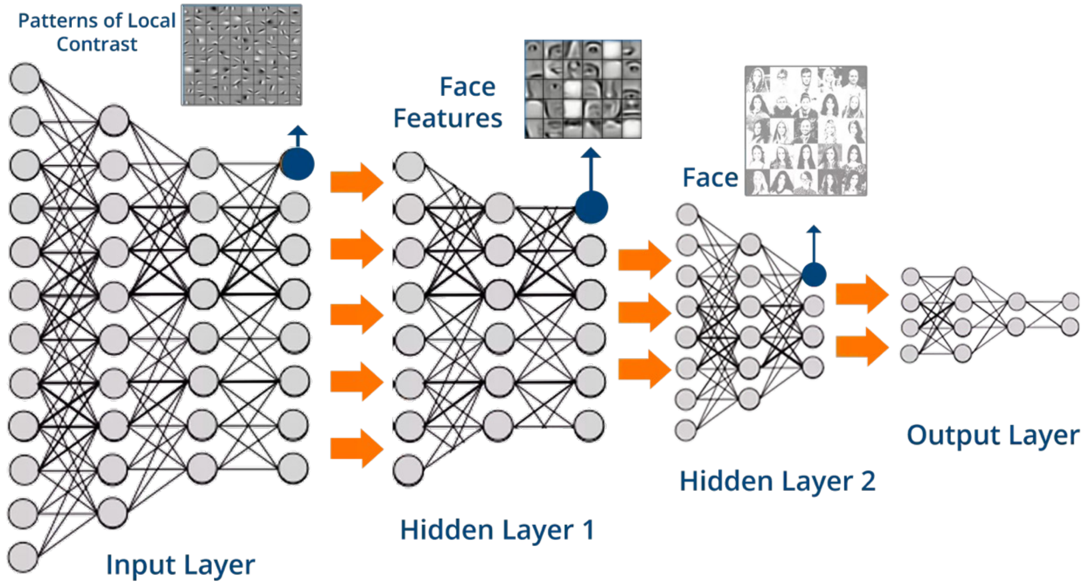
Main question:

How do we take advantage of relational structure for better prediction?

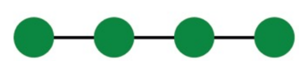
Today: Modern ML Toolbox

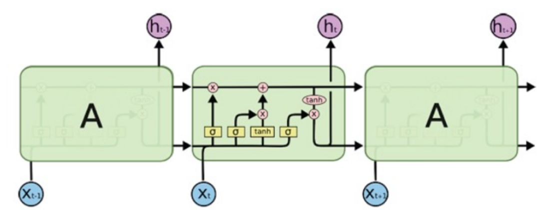


Images



Text/Speech



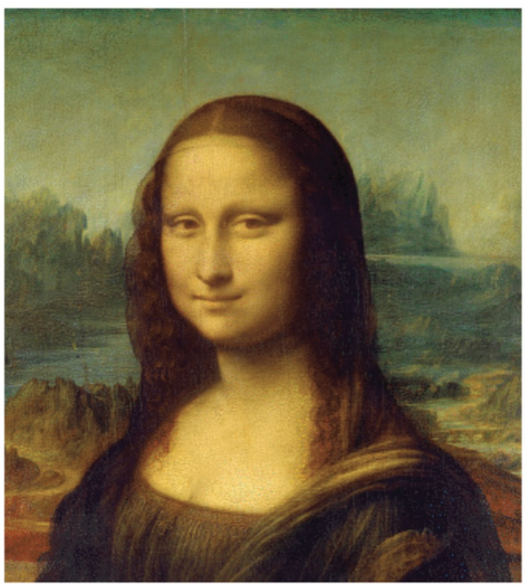


Modern deep learning toolbox is designed for simple sequences & grids

Doubt thou the stars are fire;
Doubt that the sun doth move;
Doubt truth to be a liar;
But never doubt I love...

Text

Audio signals



Images

Modern
deep learning toolbox
is designed for
sequences & grids

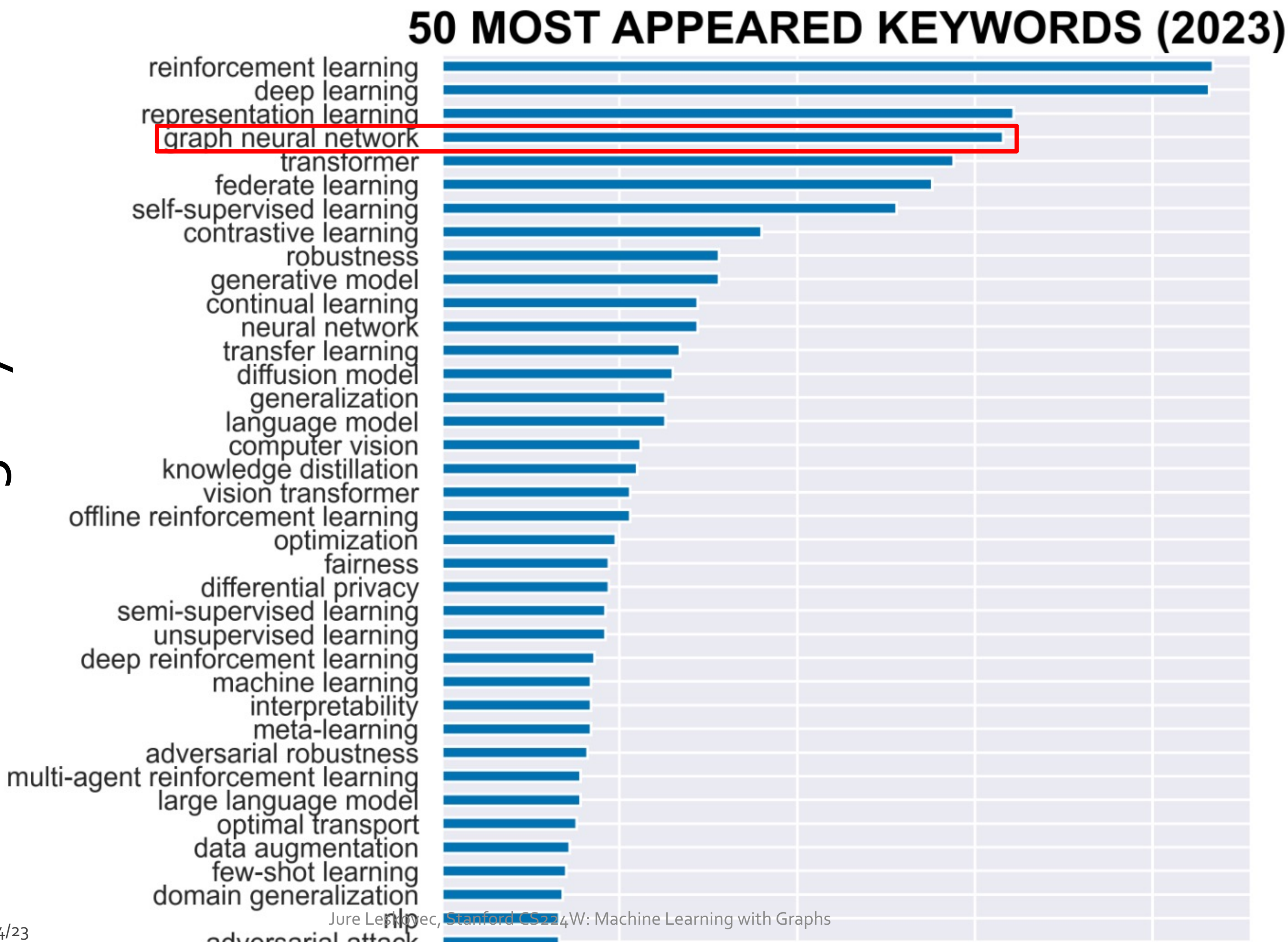
This Course: CS224W

How can we develop neural networks that are much more broadly applicable?

Graphs are the new frontier of deep learning

Hot subfield in ML

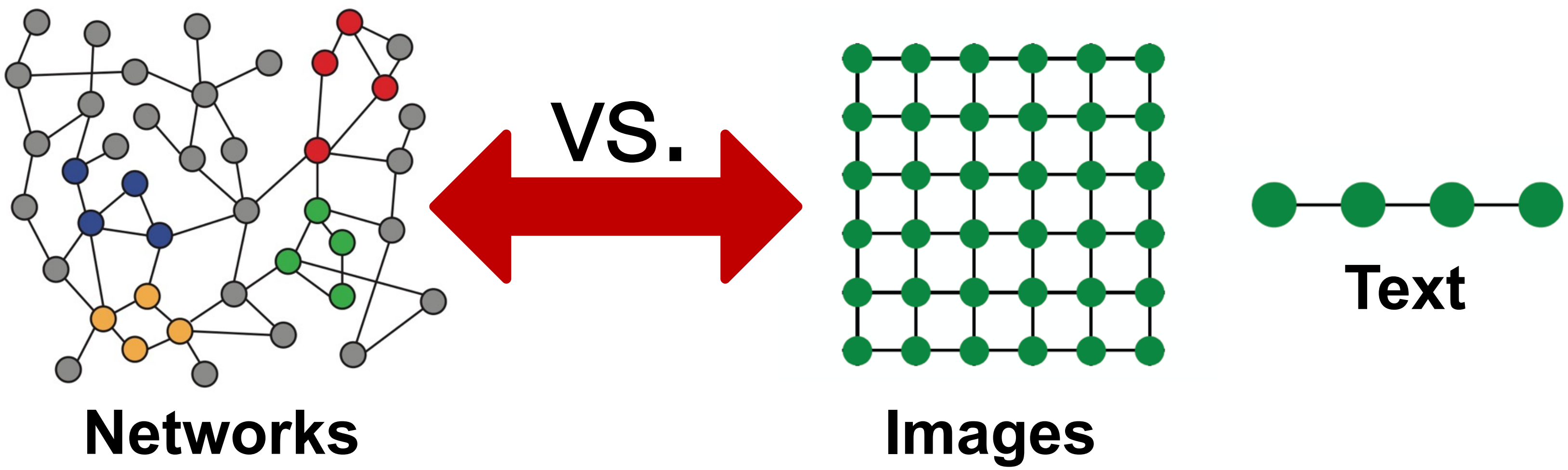
ICLR 2023 keywords



Why is Graph Deep Learning Hard?

Networks are complex.

- Arbitrary size and complex topological structure (*i.e.*, no spatial locality like grids)



- No fixed node ordering or reference point
- Often dynamic and have multimodal features

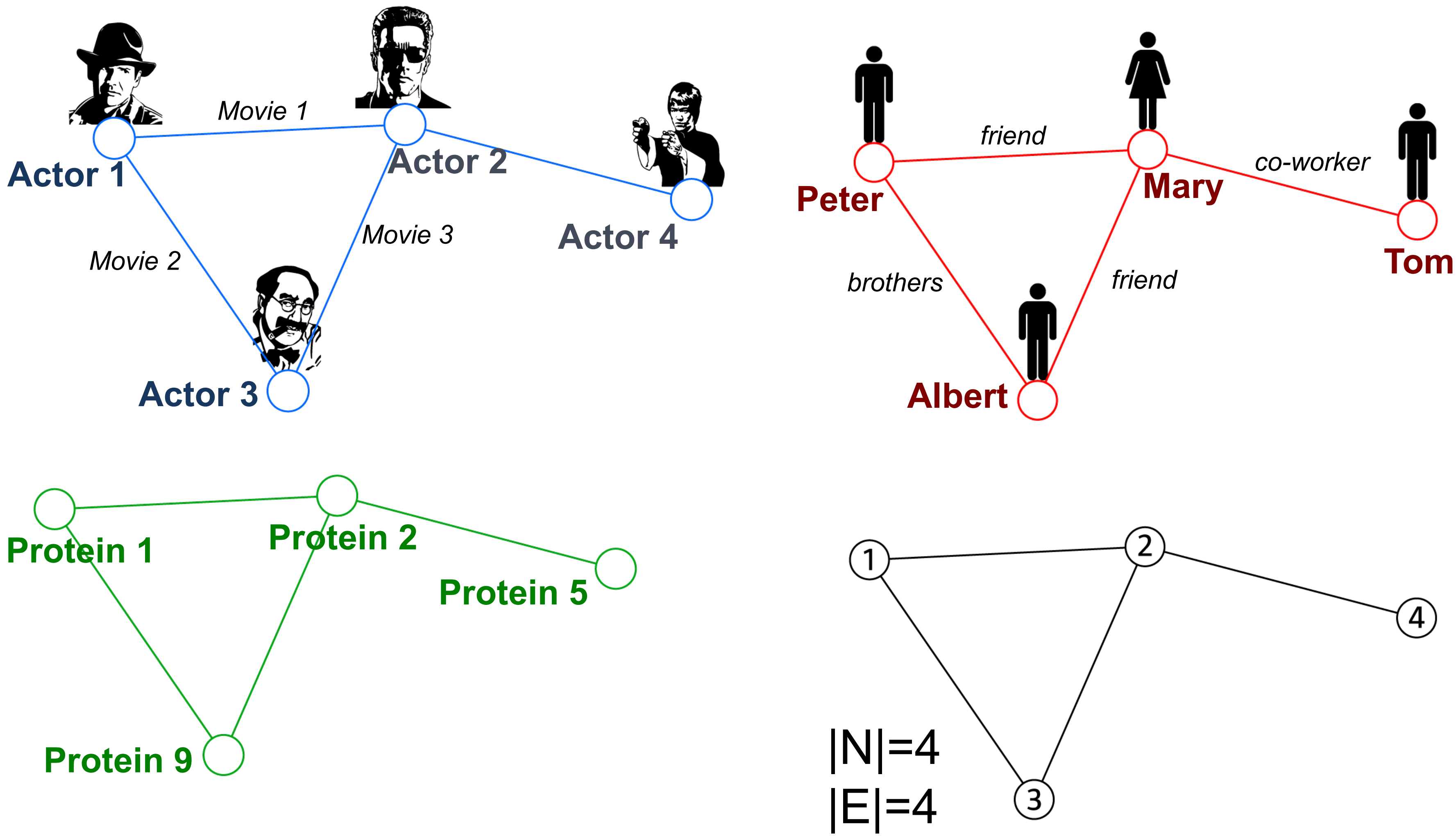
Stanford CS224W: Choice of Graph Representation

CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

http://cs224w.stanford.edu

Graphs: A Common Language



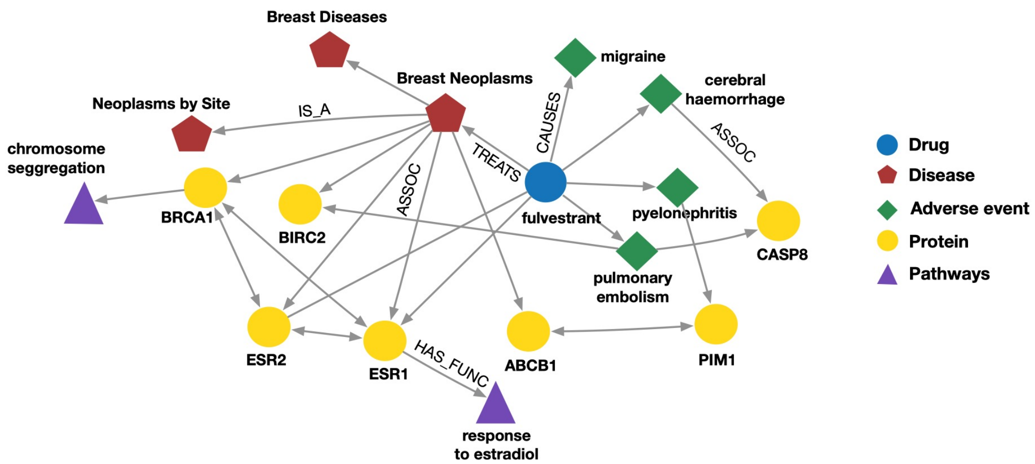
Heterogeneous Graphs

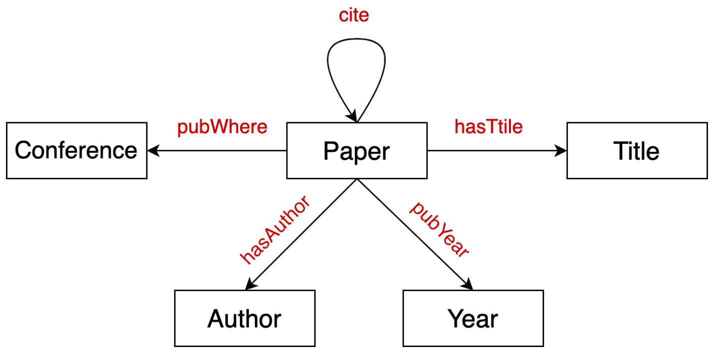
- **A heterogeneous graph is defined as**

$$\boldsymbol{G = (V, E, R, T)}$$

  - Nodes with node types $v_i \in V$
  - Edges with relation types $(v_i, r, v_j) \in E$
  - Node type $T(v_i)$
  - Relation type $r \in R$
  - Nodes and edges have attributes/features

Many Graphs are Heterogeneous





Biomedical Knowledge Graphs

Example node: Migraine
Example edge: (fulvestrant, Treats, Breast Neoplasms)
Example node type: Protein
Example edge type (relation): Causes

Academic Graphs

Example node: ICML
Example edge: (GraphSAGE, NeurIPS)
Example node type: Author
Example edge type (relation): pubYear

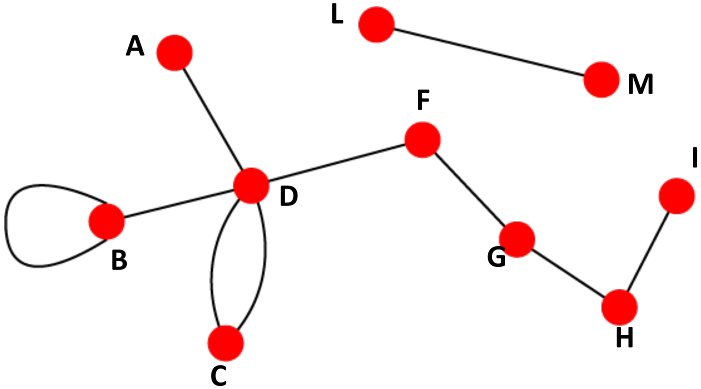
Choosing a Proper Representation

- **How to build a graph:**
  - **What are nodes?**
  - **What are edges?**
- **Choice of the proper network representation of a given domain/problem determines our ability to use networks successfully:**
  - In some cases, there is a unique, unambiguous representation
  - In other cases, the representation is by no means unique
  - The way you assign links will determine the nature of the question you can study

Directed vs. Undirected Graphs

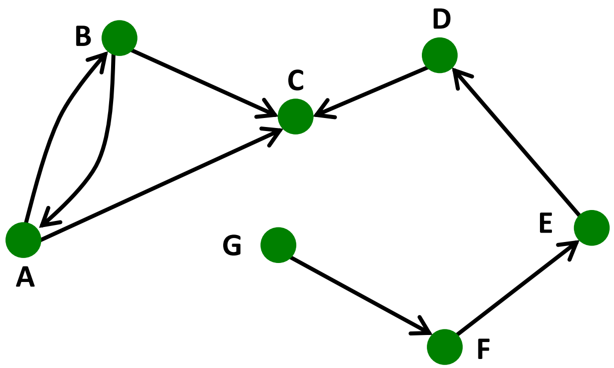
Undirected

- **Links:** undirected (symmetrical, reciprocal)



Directed

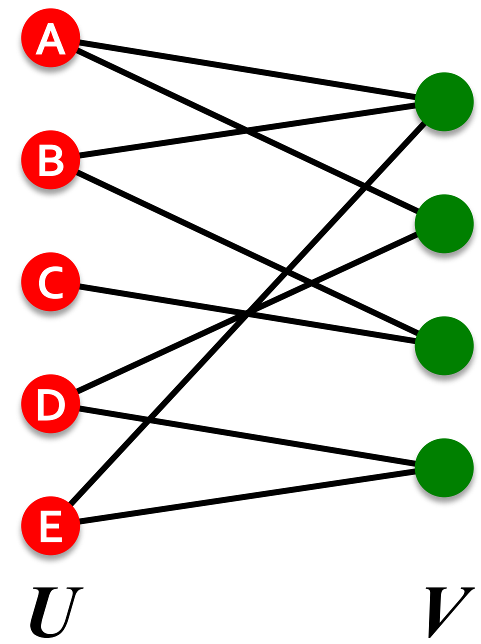
- **Links:** directed



- **Other considerations:**
  - Weights
  - Properties
  - Types
  - Attributes

Bipartite Graph

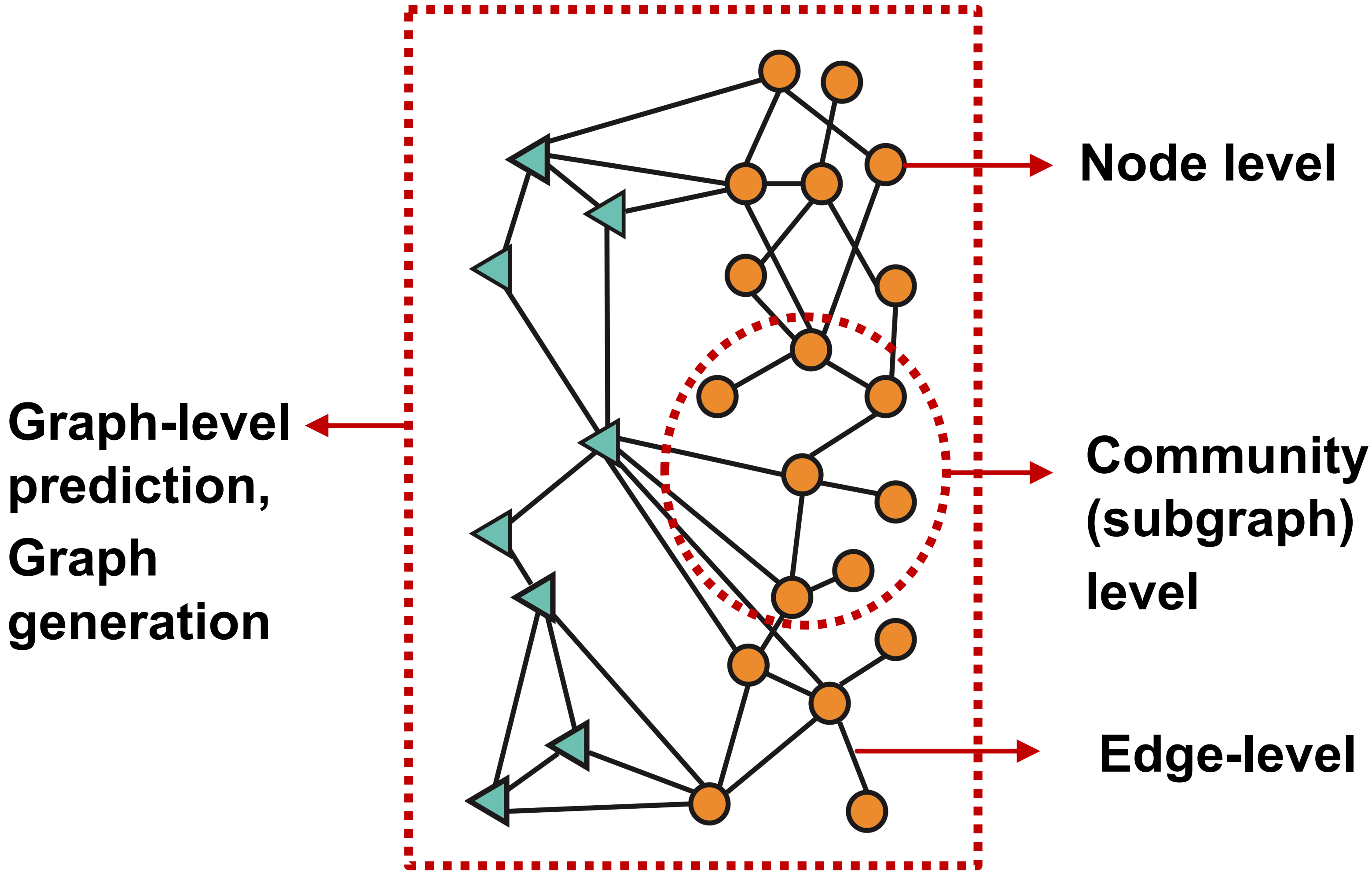
- **Bipartite graph** is a graph whose nodes can be divided into two disjoint sets ***U*** and ***V*** such that every link connects a node in ***U*** to one in ***V***; that is, ***U*** and ***V*** are **independent sets**
- **Examples:**
  - Authors-to-Papers (they authored)
  - Actors-to-Movies (they appeared in)
  - Users-to-Movies (they rated)
  - Recipes-to-Ingredients (they contain)
- **"Folded" networks:**
  - Author collaboration networks
  - Movie co-rating networks



Stanford CS224W: Applications of Graph ML

CS224W: Machine Learning with Graphs
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

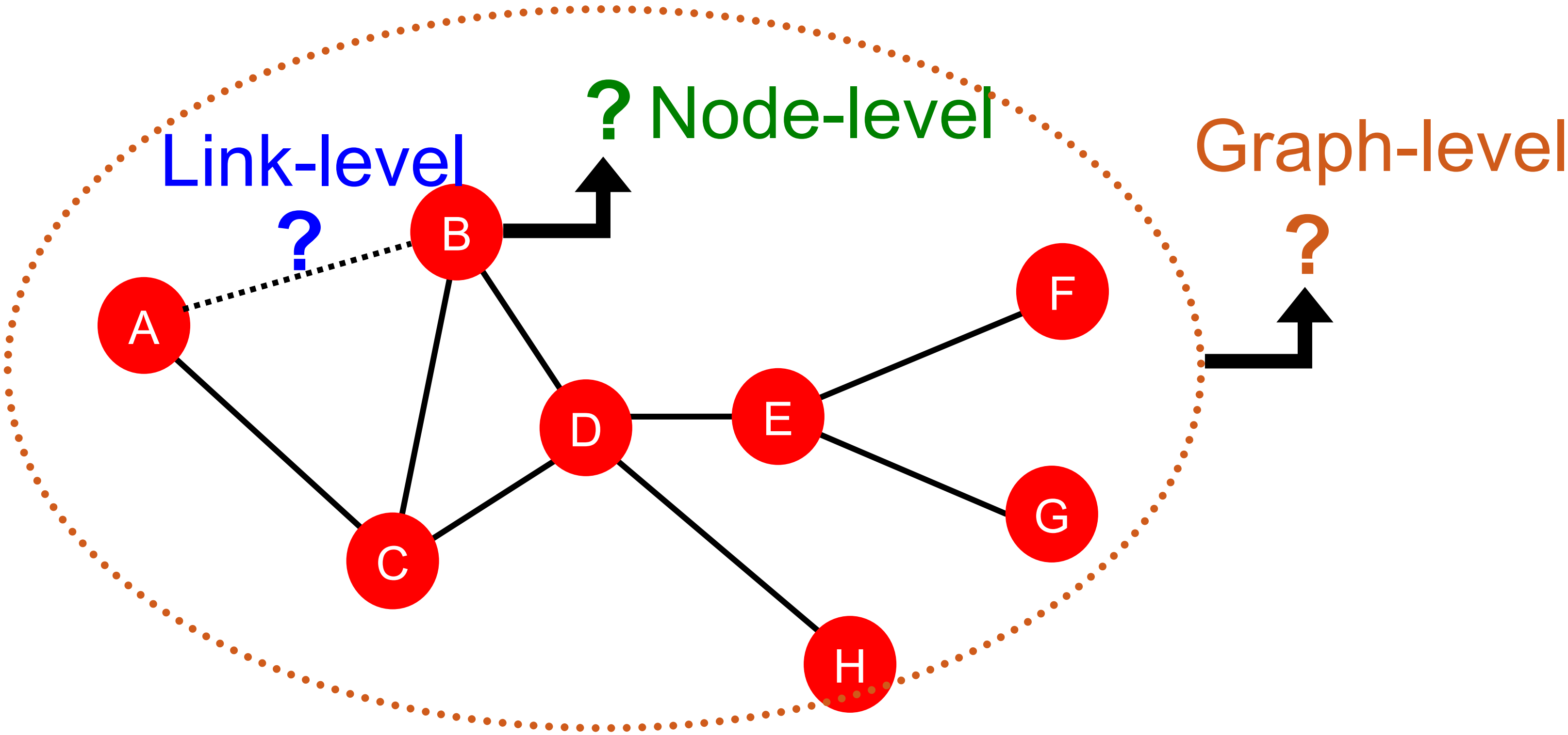
Different Types of Tasks



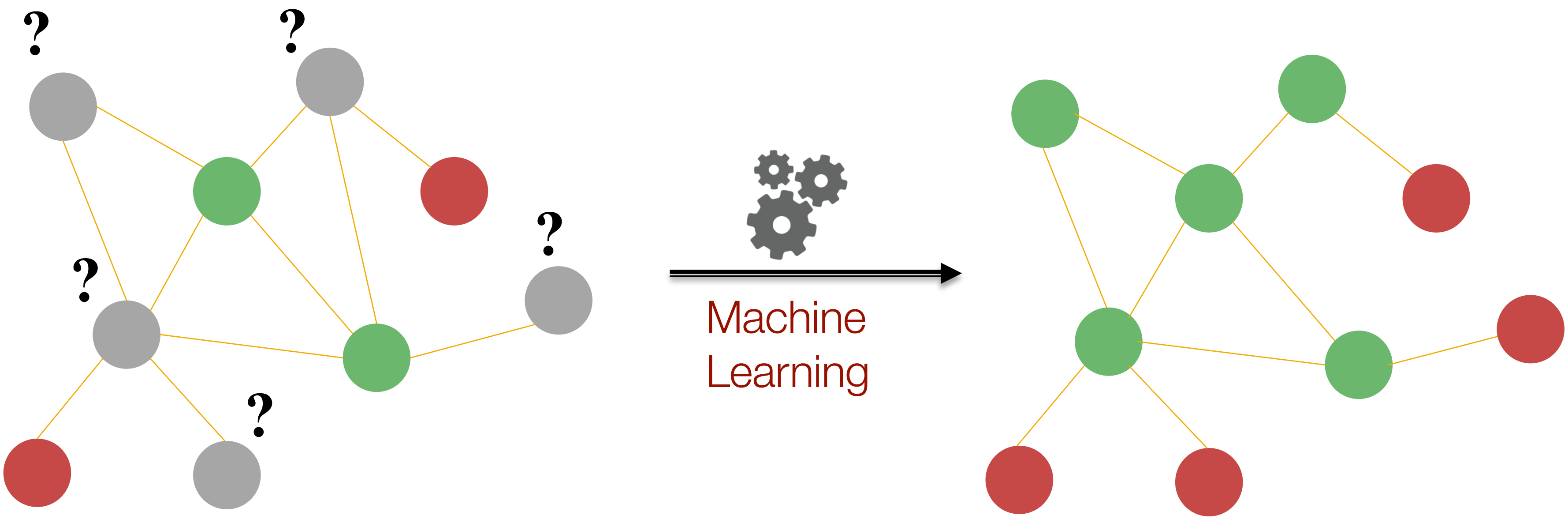
Stanford CS224W:
Node-Level Tasks

Machine Learning Tasks: Review

- Node-level prediction
- Link-level prediction
- Graph-level prediction



Node-Level Tasks

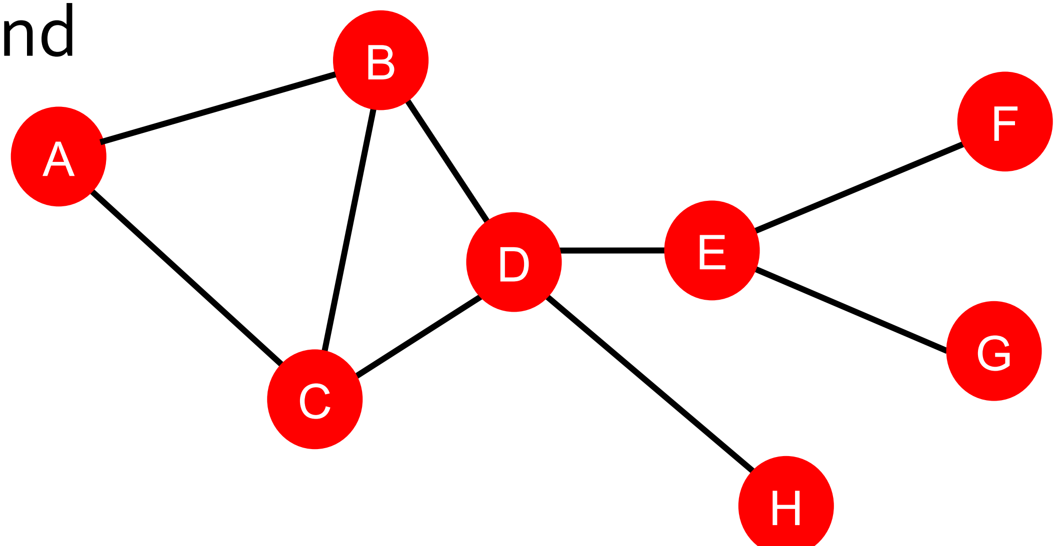


Node classification

Node-Level Network Structure

Goal: Characterize the structure and position of a node in the network:

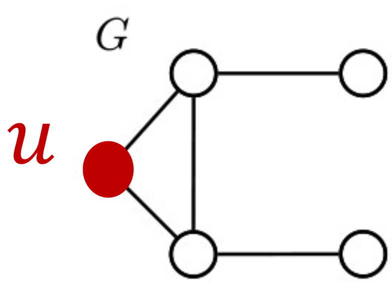
- Node degree
- Node importance & position
  - E.g., Number of shortest paths passing through a node
  - E.g., Avg. shortest path length to other nodes
- Substructures around the node

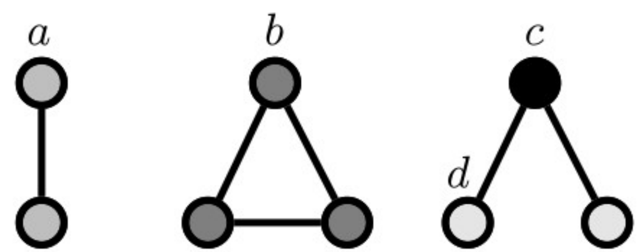


Node's Subgraphs: Graphlets

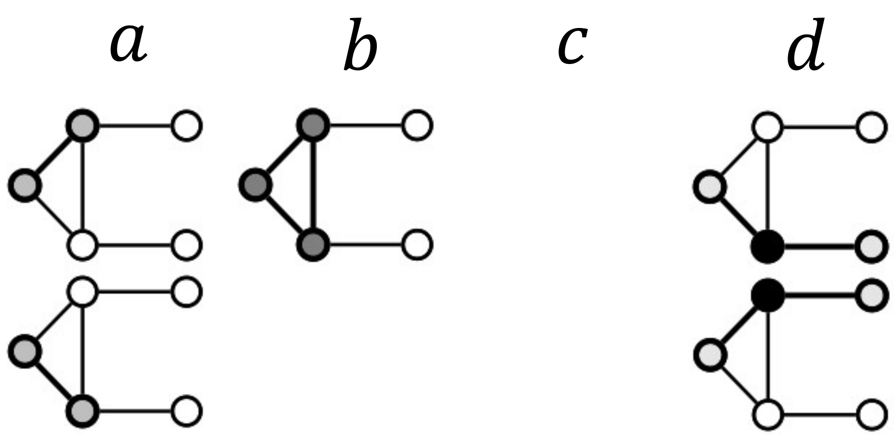
- **Graphlets:** A count vector of rooted subgraphs at a given node.
- **Example:**

All possible graphlets on up to 3 nodes





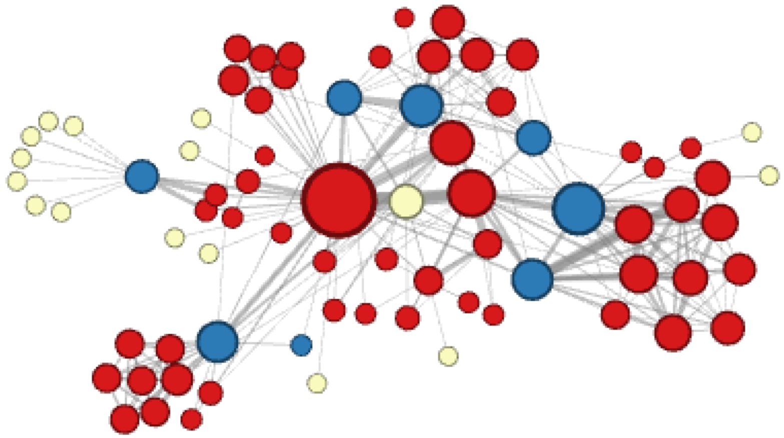
Graphlet instances of node u:



Graphlets of node u:
a, b, c, d
[2,1,0,2]

Discussion

Different ways to label nodes of the network:



Node features defined so far would allow to distinguish nodes in the above example

However, the features defines so far would not allow for distinguishing the above node labelling

Example (1): Protein Folding

Computationally predict a protein's 3D structure based solely on its amino acid sequence:

For each node predict its 3D coordinates

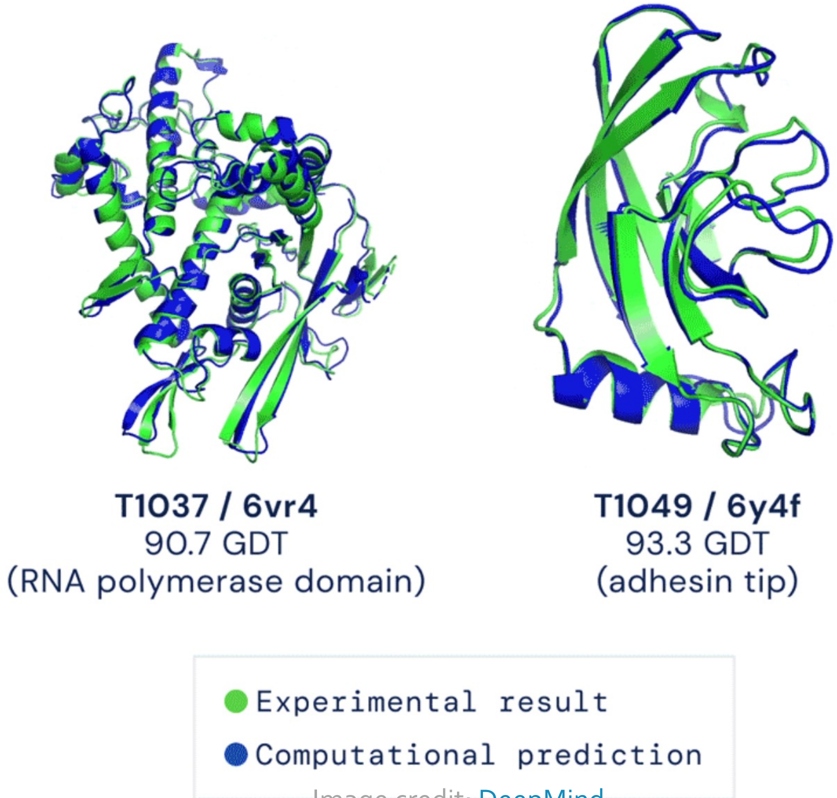


Image credit: DeepMind

AlphaFold: Impact

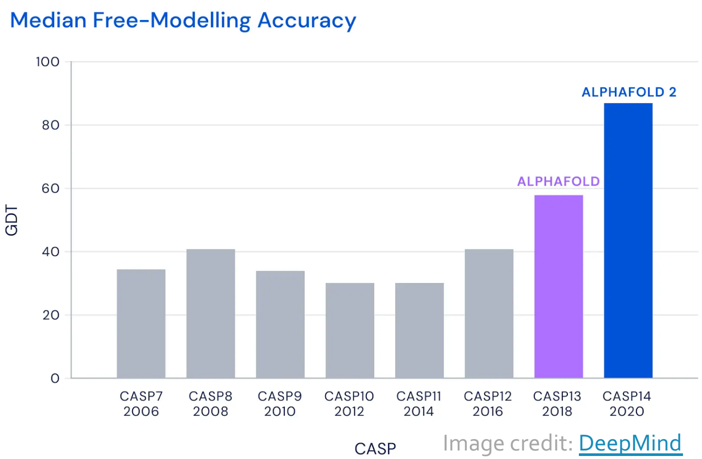


Image credit: DeepMind

Image credit: SingularityHub

AlphaFold's AI could change the world of biological science as we know it

DeepMind's latest AI breakthrough can accurately predict the way proteins fold

Has Artificial Intelligence 'Solved' Biology's Protein-Folding Problem?

12-14-20

DeepMind's latest AI breakthrough could turbocharge drug discovery

AlphaFold: Solving Protein Folding

- **Key idea:** "Spatial graph"
  - **Nodes:** Amino acids in a protein sequence
  - **Edges:** Proximity between amino acids (residues)

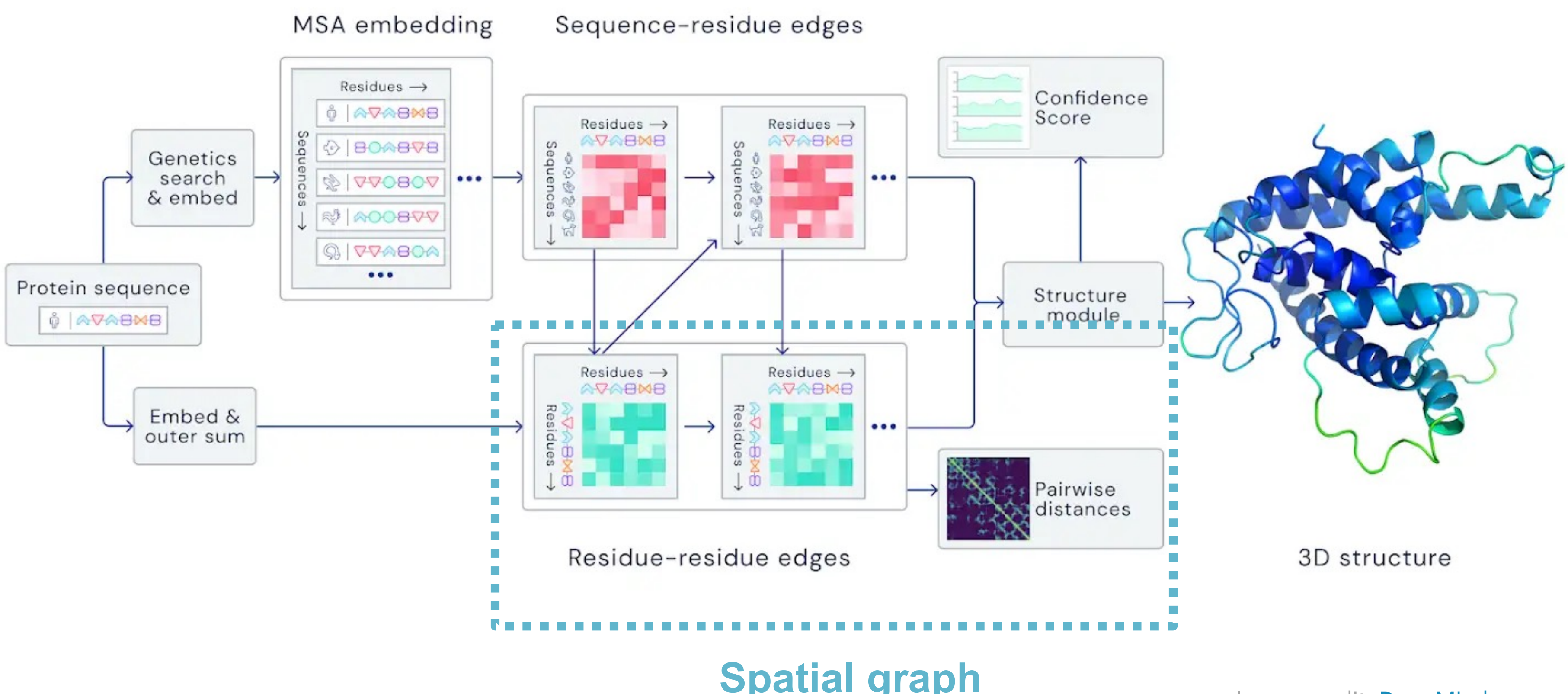


Image credit: DeepMind

Stanford CS224W: Link Prediction

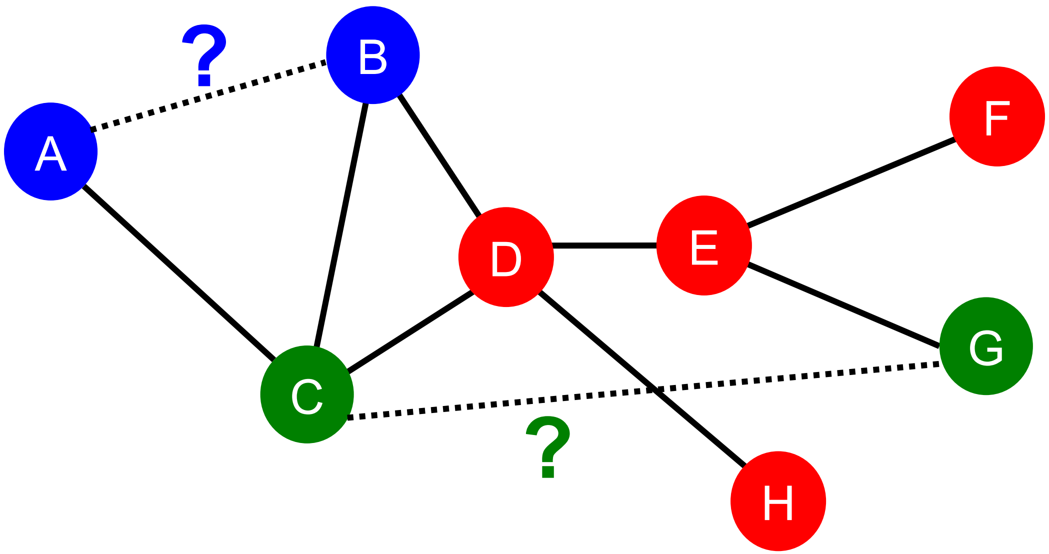
CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

http://cs224w.stanford.edu

Link-Level Prediction Task

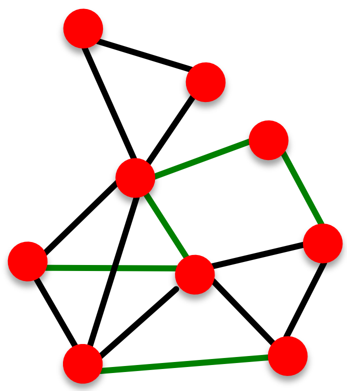
- The task is to predict **new/missing/unknown links** based on the existing links.
- At test time, node pairs (with no existing links) are ranked, and top K node pairs are predicted.
- Task: Make a prediction for **a pair of nodes**.



Link Prediction as a Task

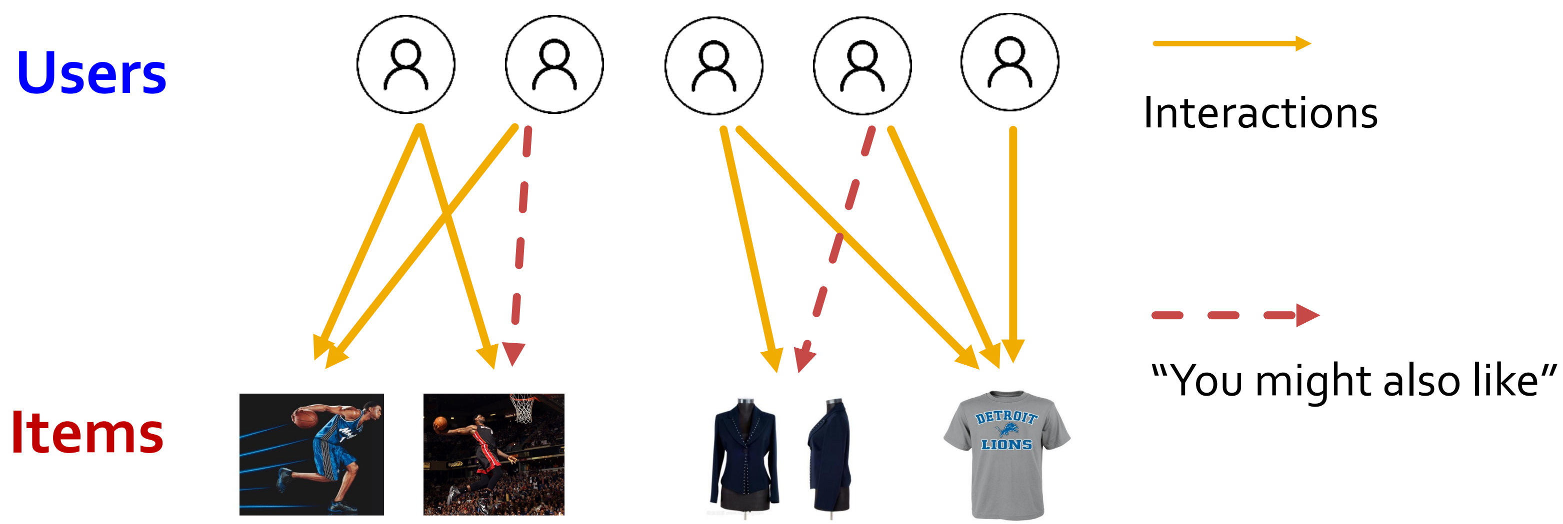
Two formulations of the link prediction task:

- **1) Links missing at random:**
  - Remove a random set of links and then aim to predict them
- **2) Links over time:**
  - Given $G[t_0, t'_0]$ a graph defined by edges up to time t'_0, **output a ranked list L** of edges (not in $G[t_0, t'_0]$) that are predicted to appear in time $G[t_1, t'_1]$
  - **Evaluation:**
    - $n = |E_{new}|$: # new edges that appear during the test period $[t_1, t'_1]$
    - Take top n elements of L and count correct edges



Example (1): Recommender Systems

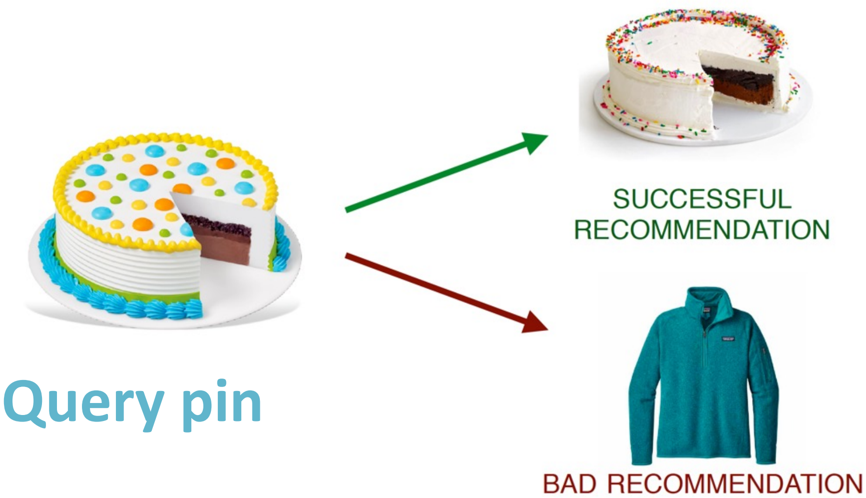
- **Users interacts with items**
  - Watch movies, buy merchandise, listen to music
  - **Nodes:** Users and items
  - **Edges:** User-item interactions
- **Goal: Recommend items users might like**



Ying et al., [Graph Convolutional Neural Networks for Web-Scale Recommender Systems](), KDD 2018

PinSage: Graph-based Recommender

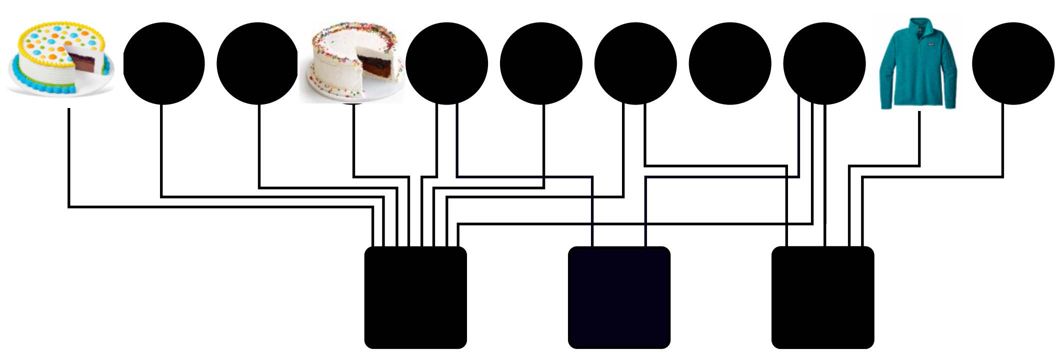
Task: Recommend related pins to users

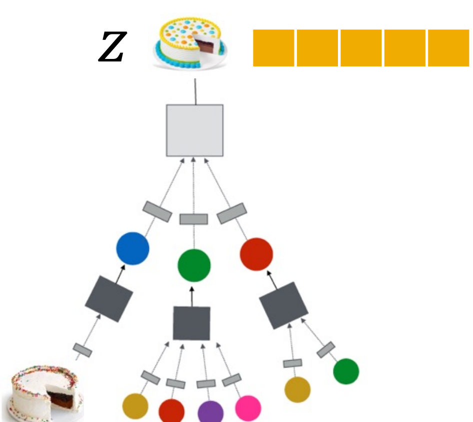


Task: Learn node embeddings z_i such that

$$d(z_{cake1}, z_{cake2}) < d(z_{cake1}, z_{sweater})$$

Predict whether two nodes in a graph are related



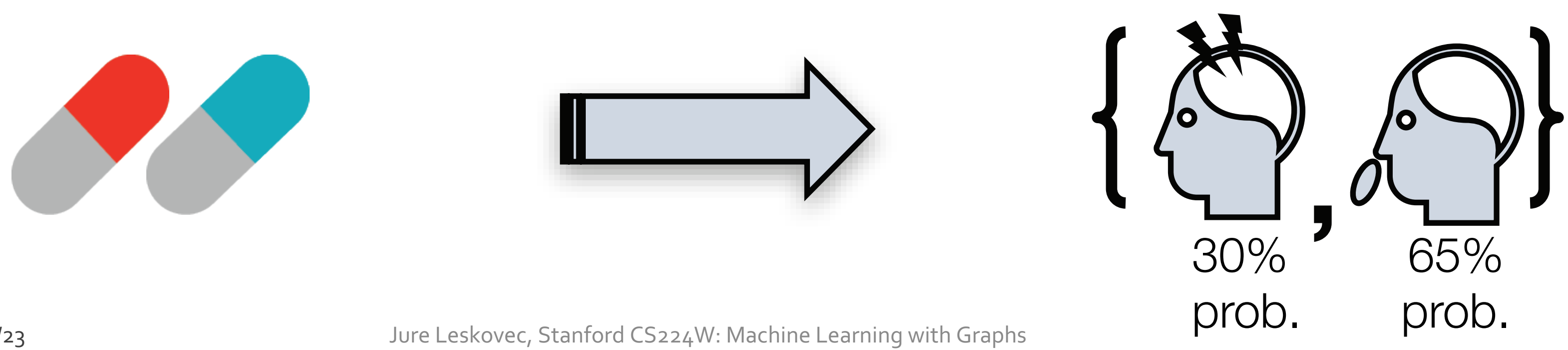


Example (2): Drug Side Effects

Many patients take multiple drugs to treat complex or co-existing diseases:

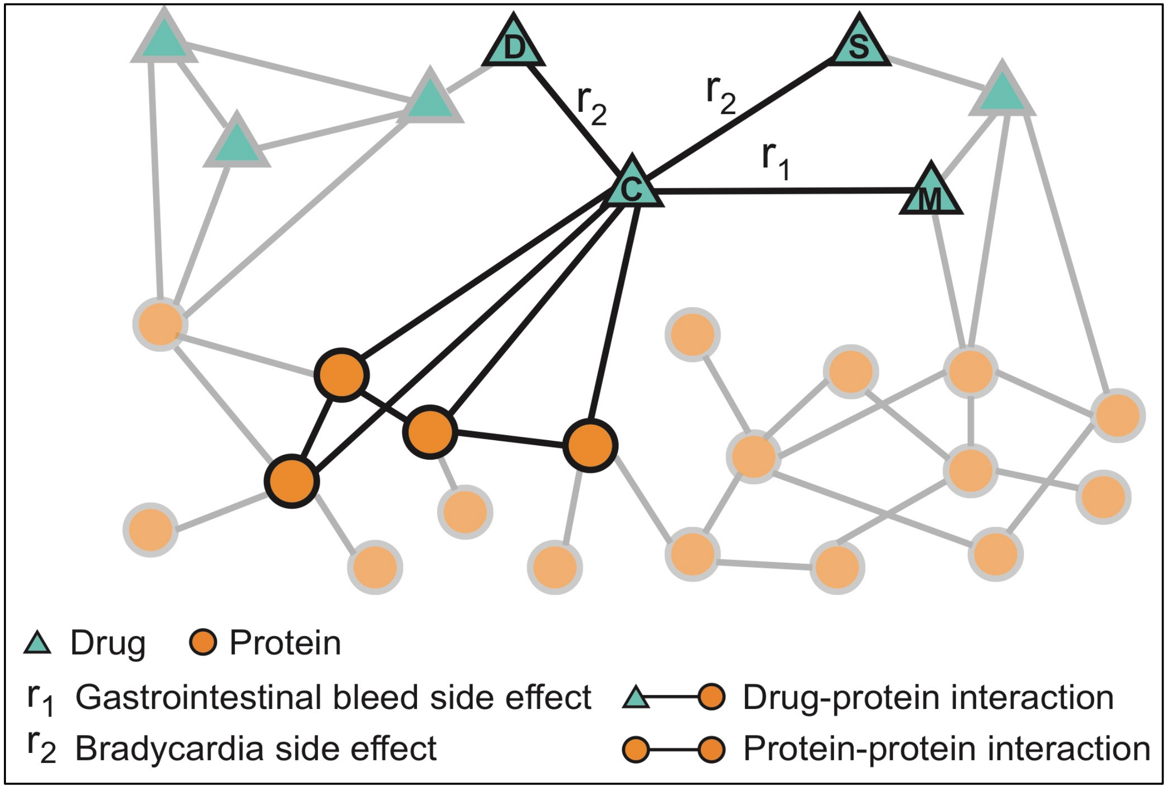
- 46% of people ages 70-79 take more than 5 drugs
- Many patients take more than 20 drugs to treat heart disease, depression, insomnia, etc.

Task: Given a pair of drugs predict adverse side effects

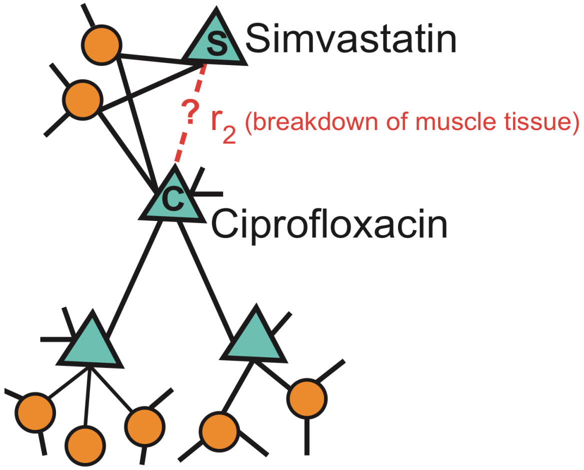


Biomedical Graph Link Prediction

- **Nodes**: Drugs & Proteins
- **Edges**: Interactions



Query: How likely will Simvastatin and Ciprofloxacin, when taken together, break down muscle tissue?



Results: *De novo* Predictions

Rank	Drug c	Drug d	Side effect r	Evidence found
1	Pyrimethamine	Aliskiren	Sarcoma	Stage *et al.* 2015
2	Tigecycline	Bimatoprost	Autonomic neuropathy	
3	Omeprazole	Dacarbazine	Telangiectases	
4	Tolcapone	Pyrimethamine	Breast disorder	Bicker *et al.* 2017
5	Minoxidil	Paricalcitol	Cluster headache	
6	Omeprazole	Amoxicillin	Renal tubular acidosis	Russo *et al.* 2016
7	Anagrelide	Azelaic acid	Cerebral thrombosis	
8	Atorvastatin	Amlodipine	Muscle inflammation	Banakh *et al.* 2017
9	Aliskiren	Tioconazole	Breast inflammation	Parving *et al.* 2012
10	Estradiol	Nadolol	Endometriosis	

Case Report

Severe Rhabdomyolysis due to Presumed Drug Interactions between Atorvastatin with Amlodipine and Ticagrelor

Stanford CS224W: Graph-Level Tasks

CS224W: Machine Learning with Graphs

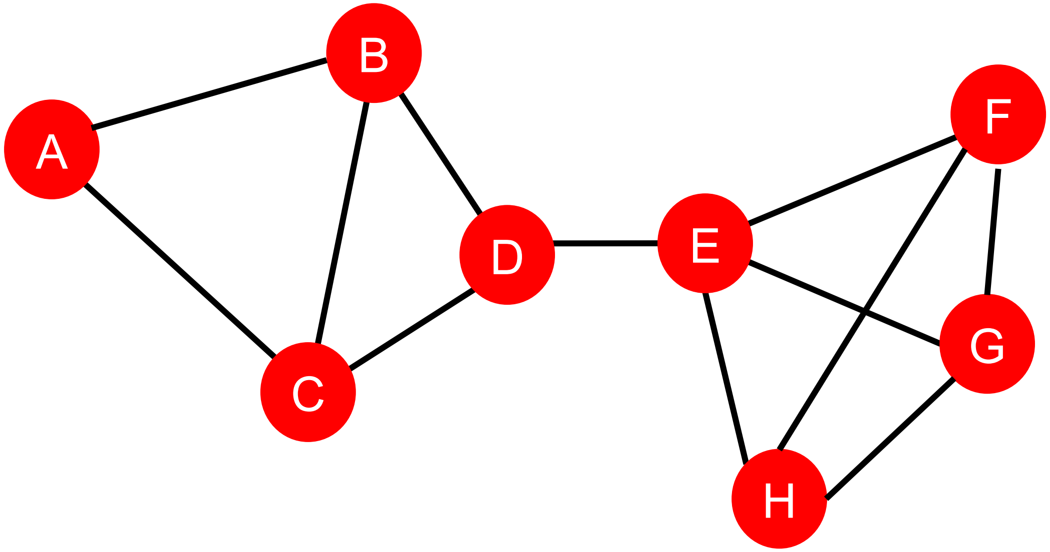
Jure Leskovec, Stanford University

http://cs224w.stanford.edu

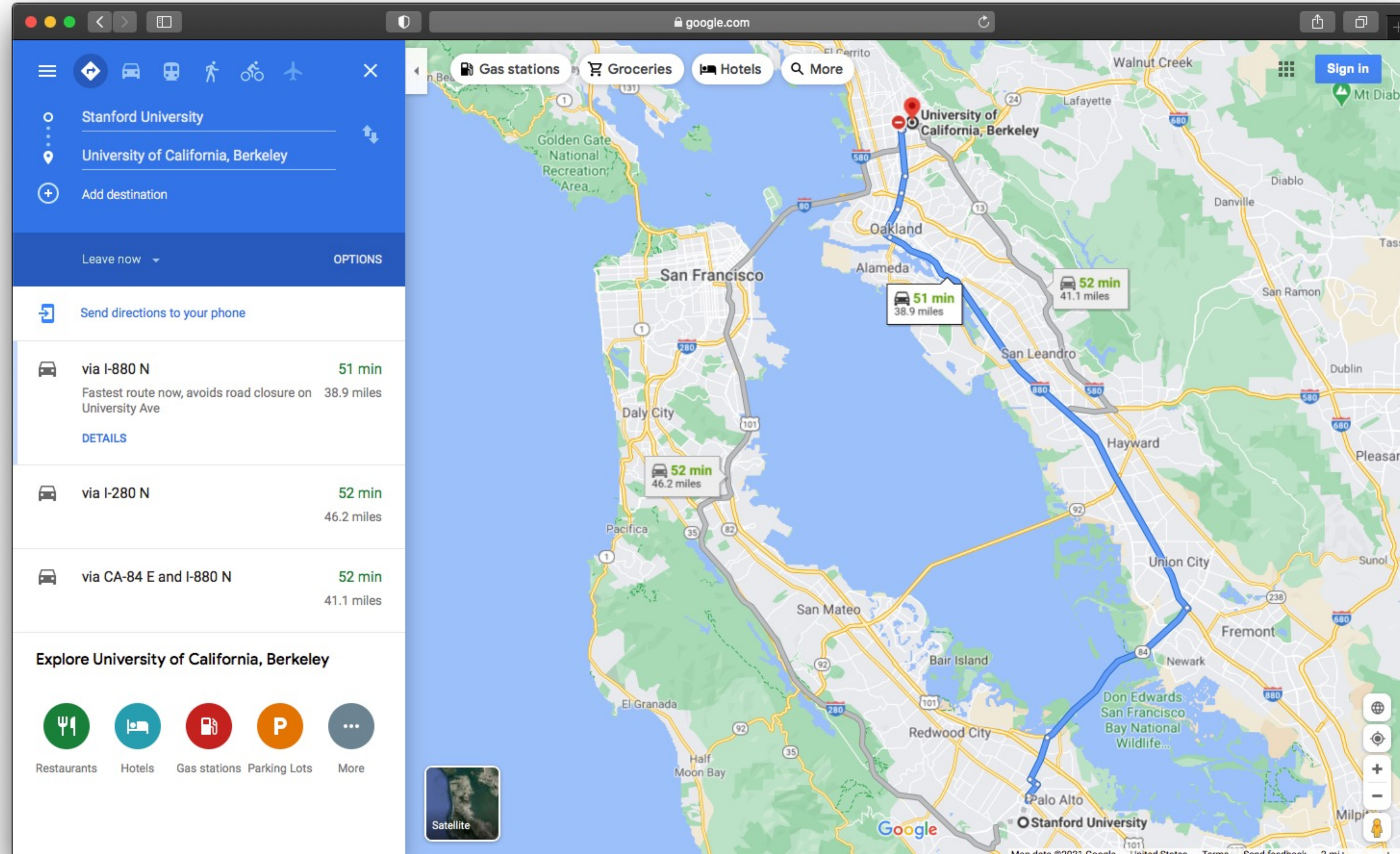
Graph-Level Features

- **Goal:** We want make a prediction for an entire graph or a subgraph of the graph.

- **For example:**



Example (1): Traffic Prediction



Road Network as a Graph

- **Nodes:** Road segments
- **Edges:** Connectivity between road segments
- **Prediction:** Time of Arrival (ETA)

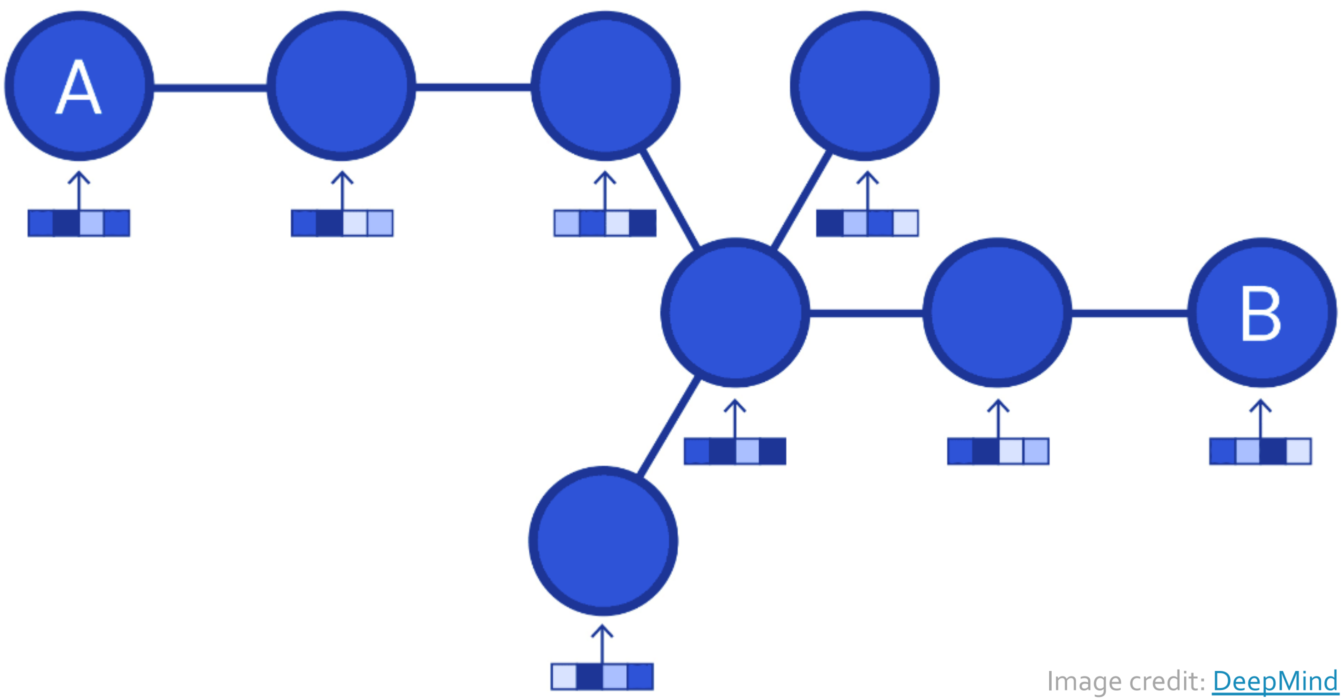
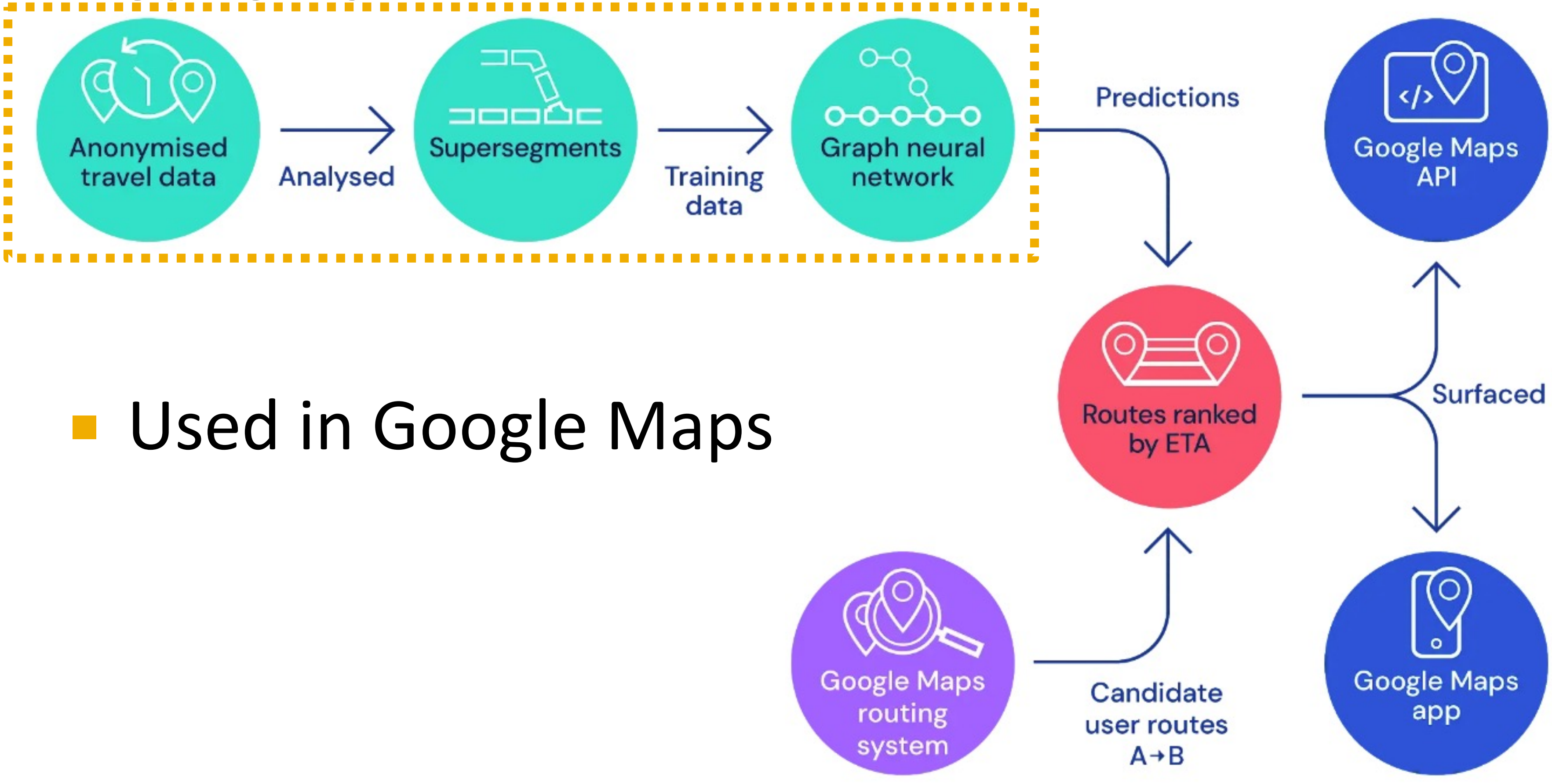


Image credit: DeepMind

Traffic Prediction via GNN

Predicting Time of Arrival with Graph Neural Networks



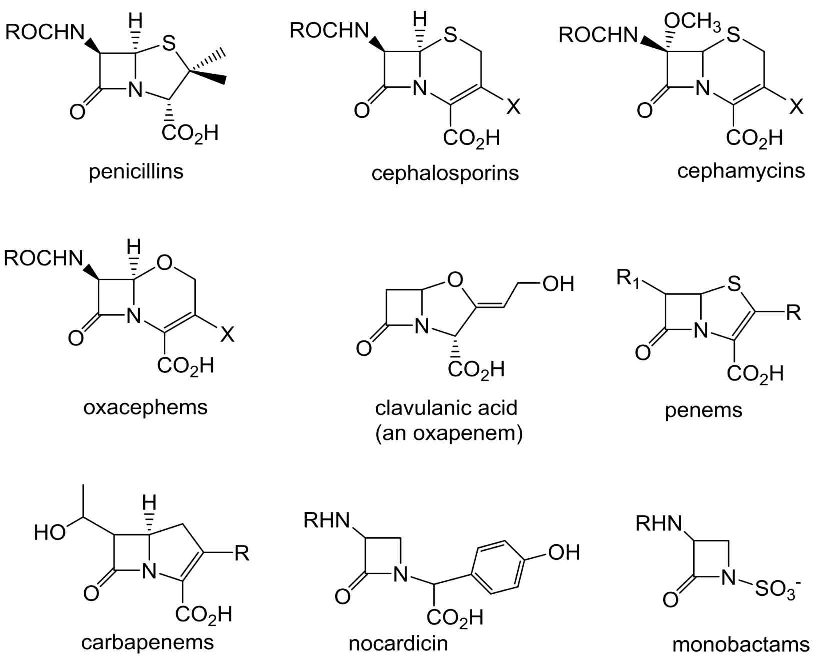
- Used in Google Maps

THE MODEL ARCHITECTURE FOR DETERMINING OPTIMAL ROUTES AND THEIR TRAVEL TIME.

Image credit: DeepMind

Example (2): Drug Discovery

- **Antibiotics are small molecular graphs**
  - **Nodes:** Atoms
  - **Edges:** Chemical bonds



Konaklieva, Monika I. "Molecular targets of β-lactam-based antimicrobials: beyond the usual suspects." Antibiotics 3.2 (2014): 128-142.

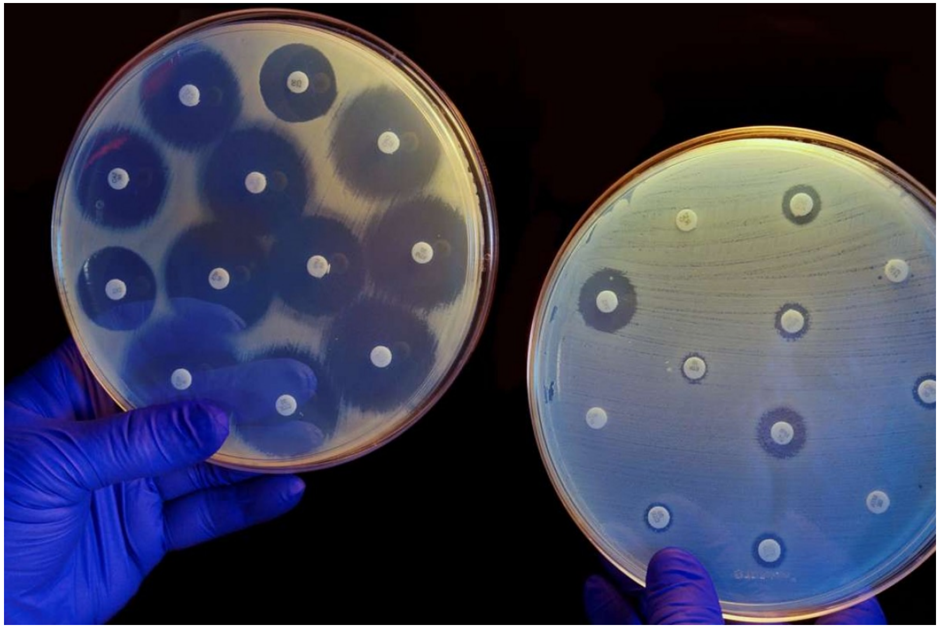
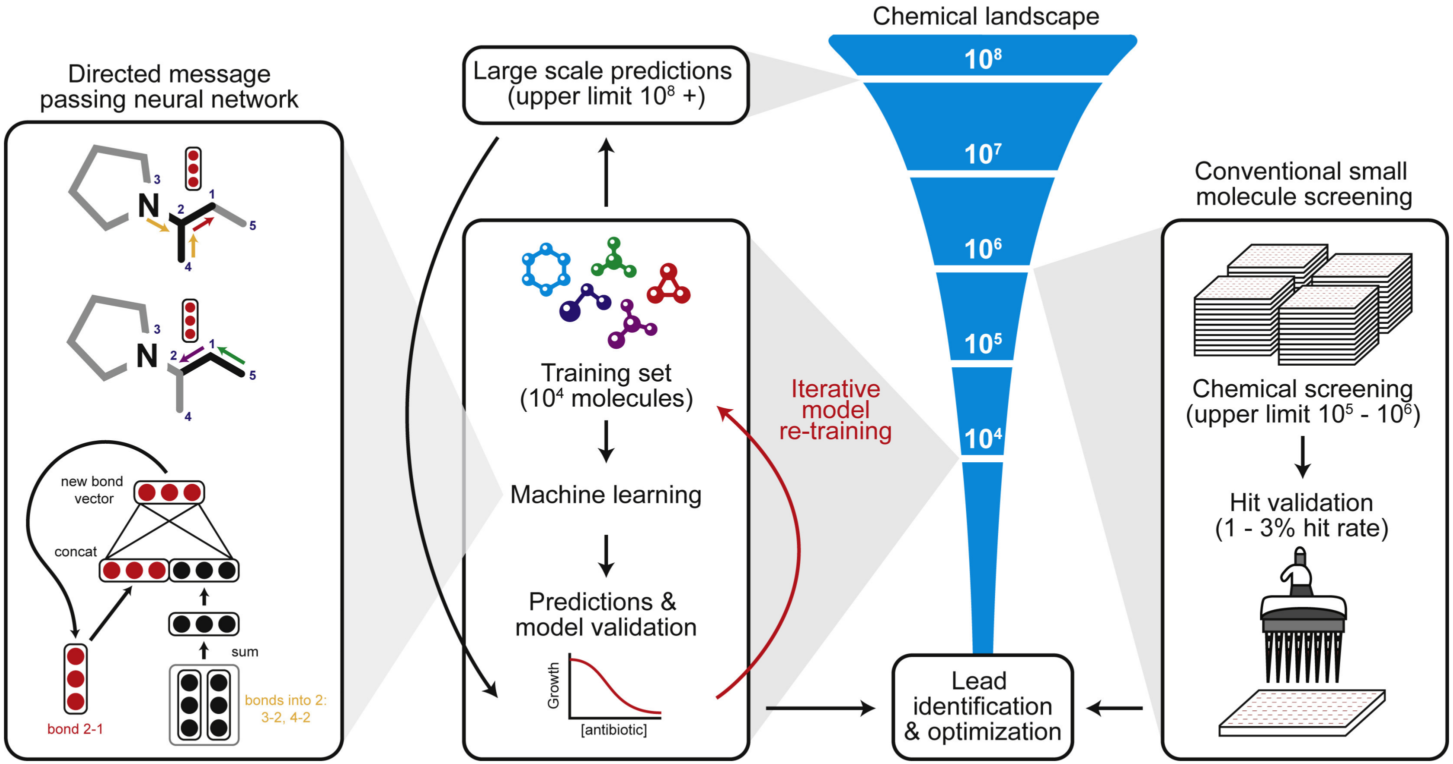


Image credit: [CNN]

Stokes et al., A Deep Learning Approach to Antibiotic Discovery, Cell 2020

Deep Learning for Antibiotic Discovery

- A Graph Neural Network **graph classification model**
- Predict promising molecules from a pool of candidates



Stokes, Jonathan M., et al. "A deep learning approach to antibiotic discovery." Cell 180.4 (2020): 688-702.

Sanchez-Gonzalez et al., [Learning to simulate complex physics with graph networks](), ICML 2020

Example (3): Physics Simulation

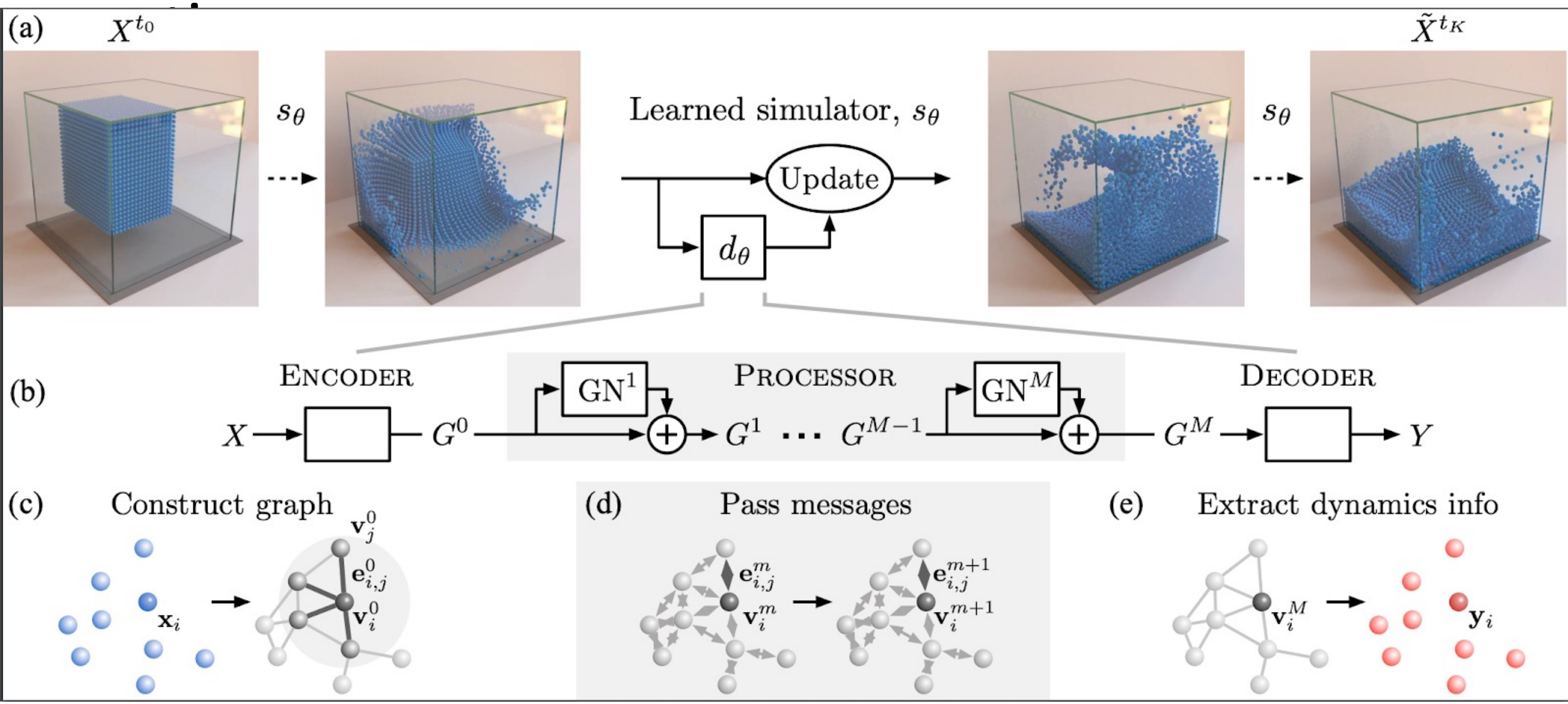
Physical simulation as a graph:

- **Nodes**: Particles
- **Edges**: Interaction between particles

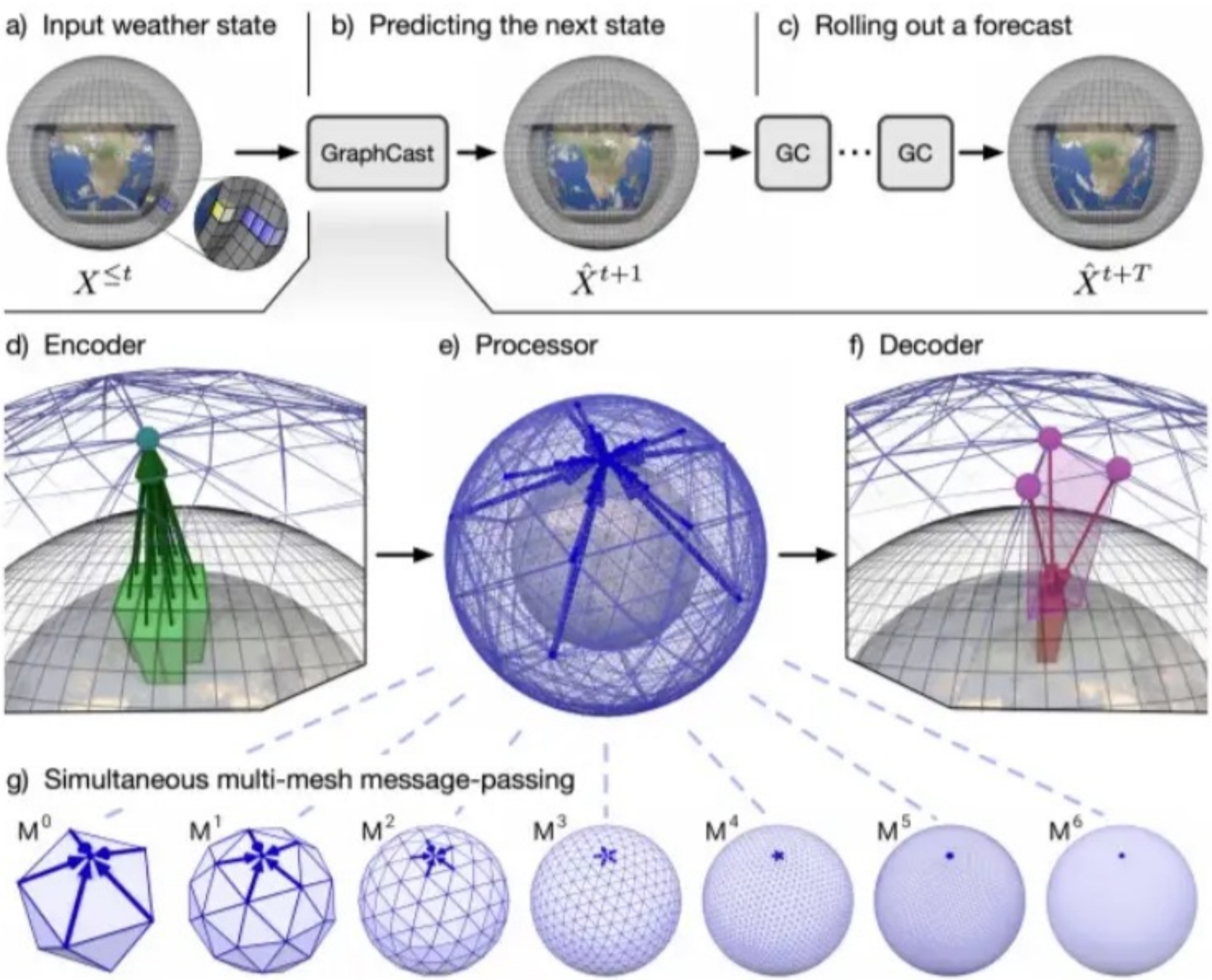
Simulation Learning Framework

A graph evolution task:

- **Goal**: Predict how a graph will evolve over

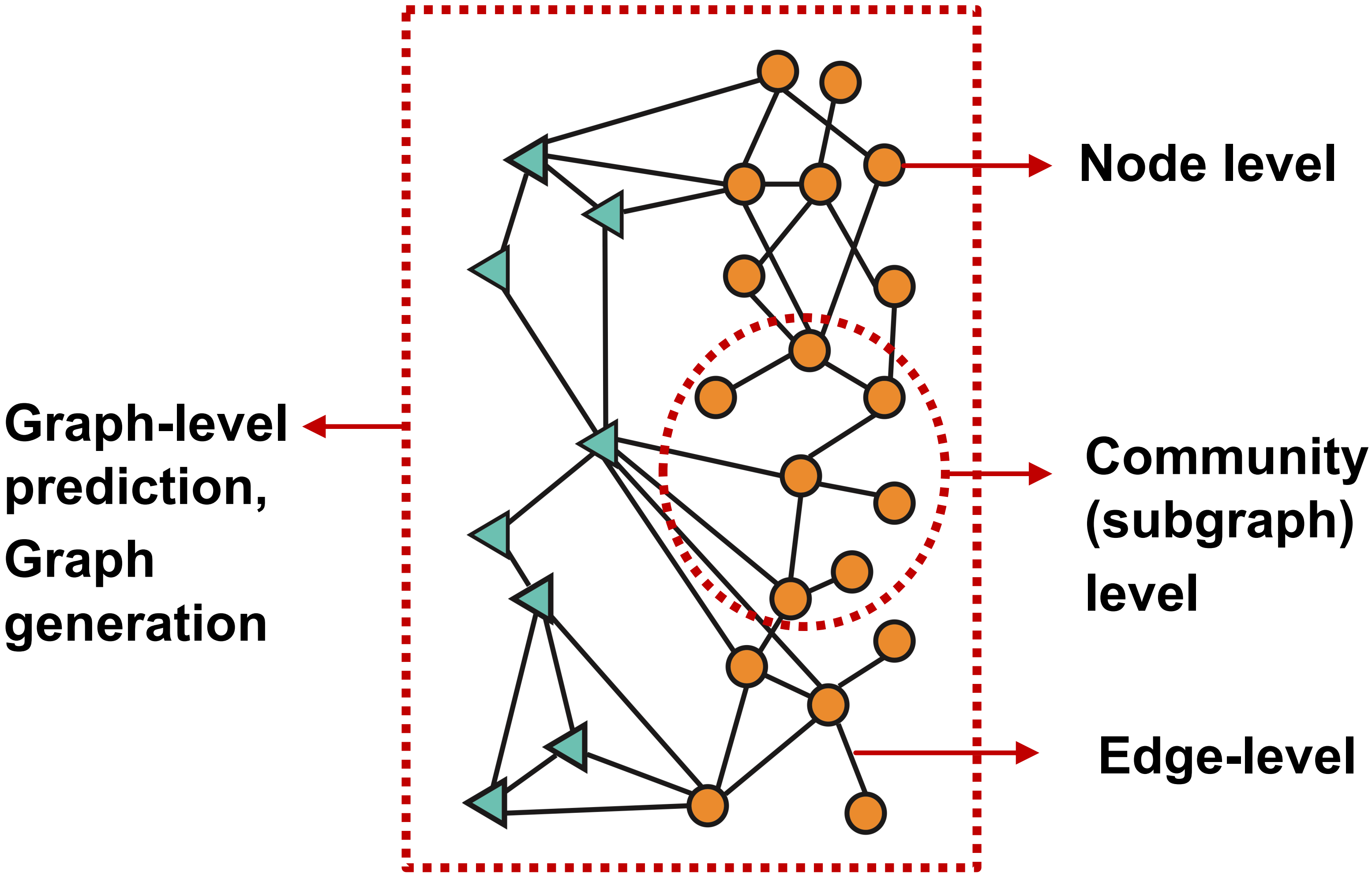


Application: Weather forecasting



https://medium.com/syncedreview/deepmind-googles-ml-based-graphcast-outperforms-the-world-s-best-medium-range-weather-9d114460aa0c

Summary


Node level
Graph-level prediction, Graph generation
Community (subgraph) level
Edge-level