

Introduction

Jiaxuan You
Assistant Professor at UIUC CDS



CS598: Deep Learning with Graphs, 2024 Fall

<https://ulab-uiuc.github.io/CS598/>

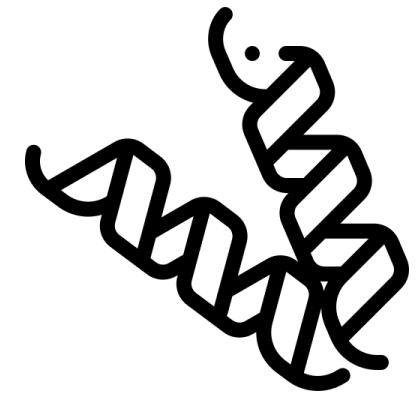
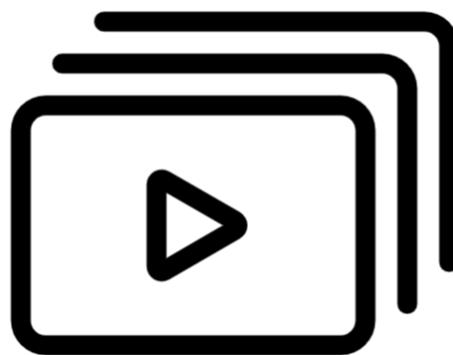
Introduction

Why Graphs?



Interconnected world

Gap

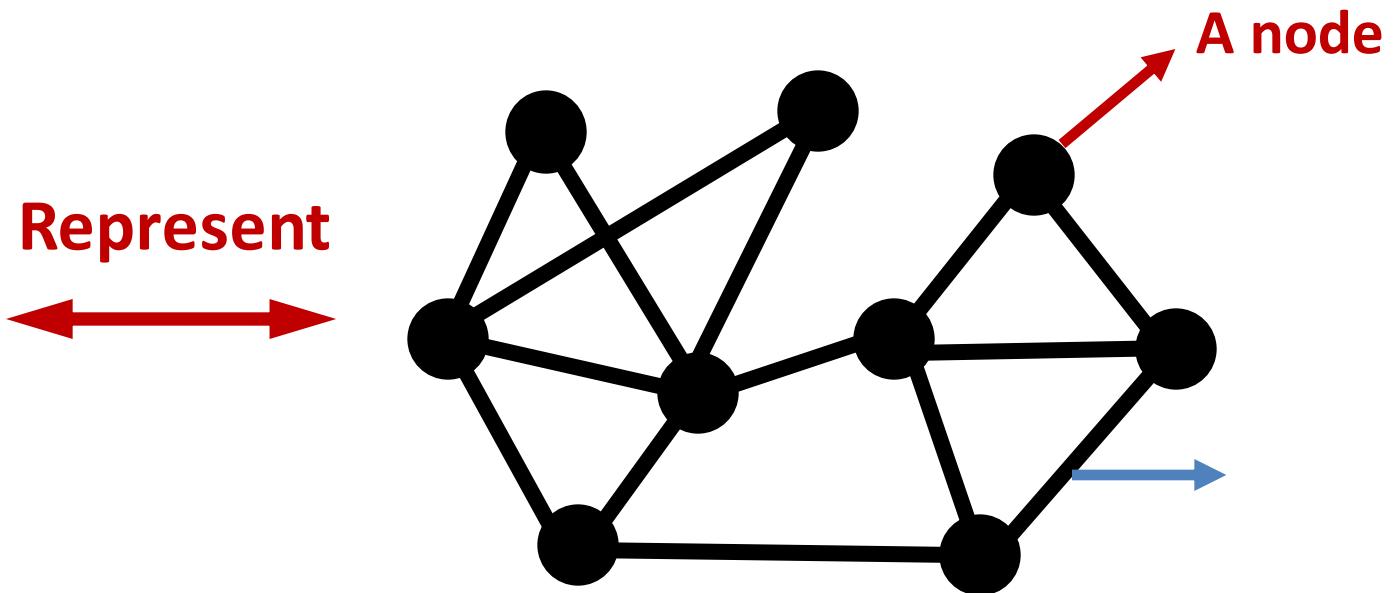


Modern ML

How to Represent Interconnected Data?



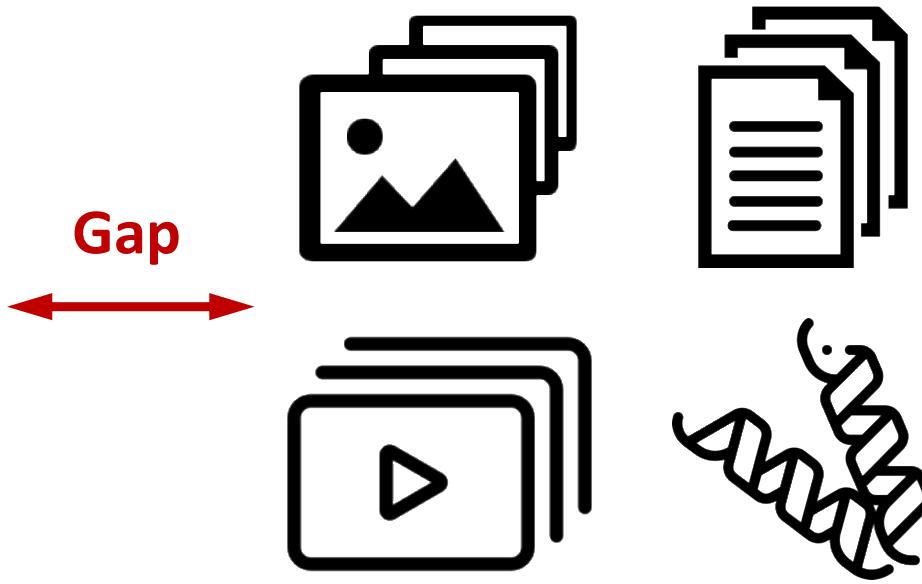
Interconnected world



Graph: The language for **describing entities with relations**



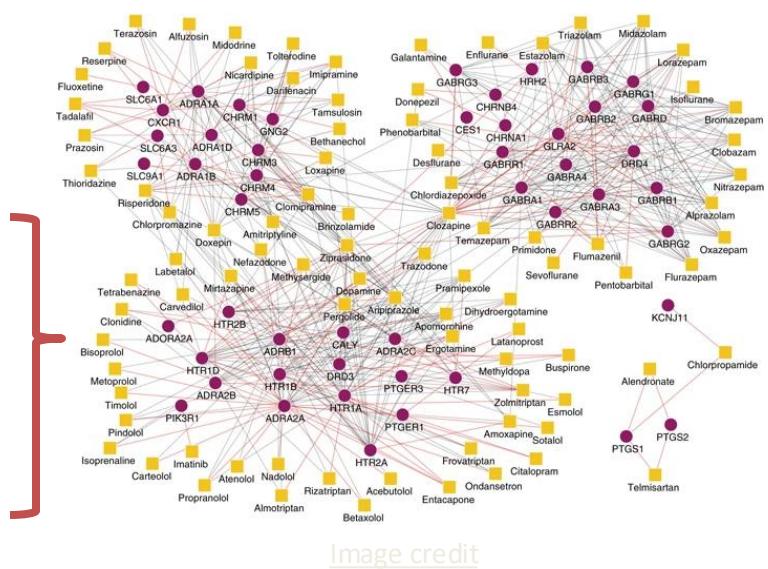
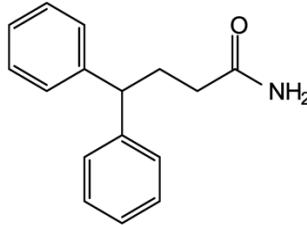
Interconnected world



Modern ML

Goal of Graph Deep Learning
Enable DL research for the
interconnected data

Graph: Ubiquitous across Disciplines



Molecule

Molecule design

Protein interaction

Drug discovery



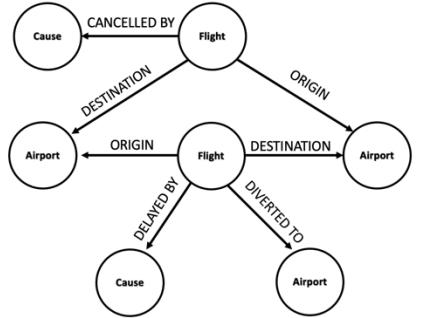
Social network

Recommender systems



- **Graphs: *flexible* and *expressive***
 - **Graphs can *bridge* interdisciplinary data**

Many Types of Data are Graphs (1)

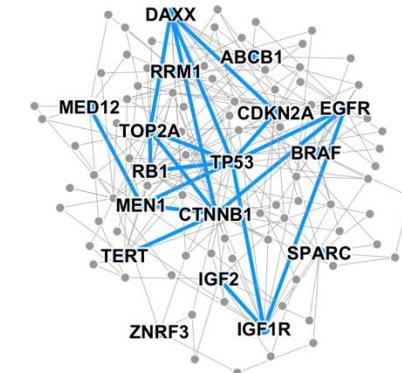


Event Graphs



Image credit: [SalientNetworks](#)

Computer Networks



Disease Pathways

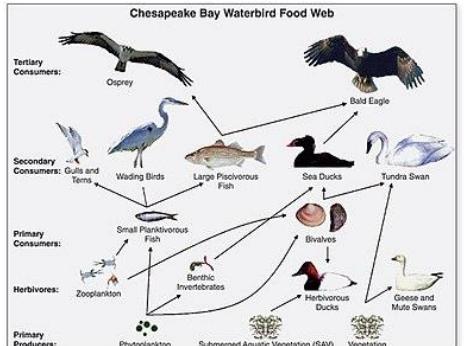


Image credit: [Wikipedia](#)

Food Webs



Image credit: [Pinterest](#)

Particle Networks



Image credit: [visitlondon.com](#)

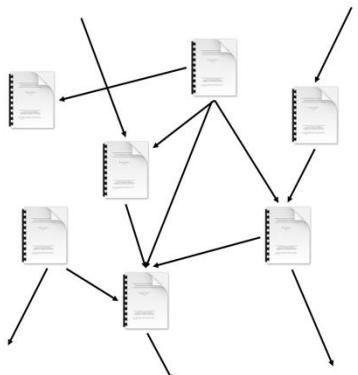
Underground Networks

Many Types of Data are Graphs (2)



Image credit: [Medium](#)

Social Networks



Citation Networks

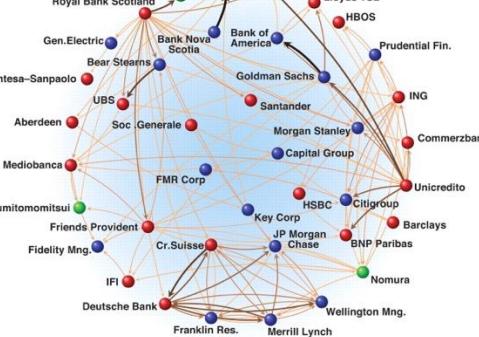


Image credit: [Science](#)

Economic Networks

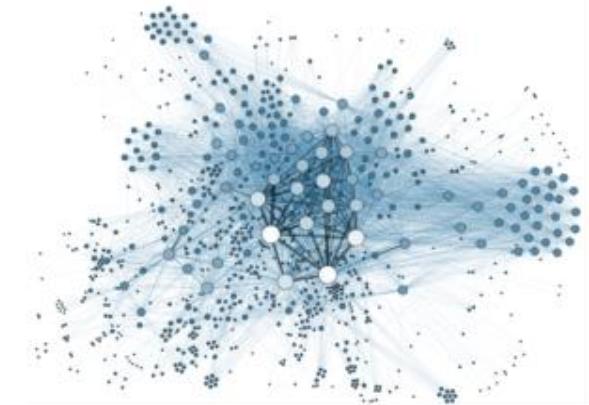


Image credit: [Lumen Learning](#)

Communication Networks



Image credit: [Missoula Current News](#)

Internet

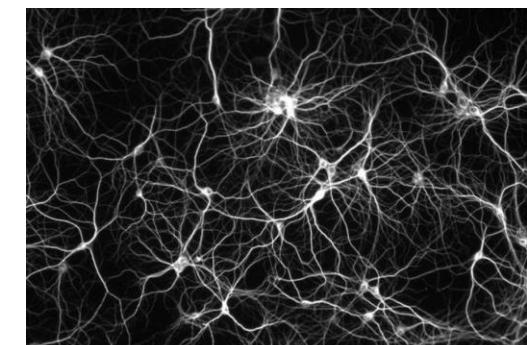


Image credit: [The Conversation](#)

Networks of Neurons

Many Types of Data are Graphs (3)

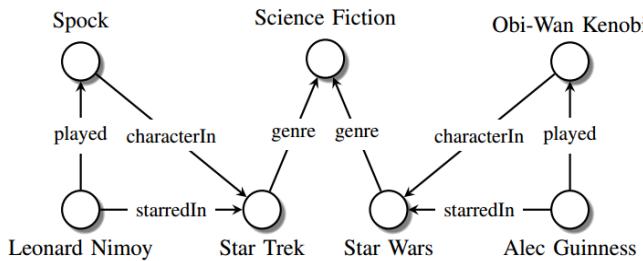


Image credit: [Maximilian Nickel et al](#)

Knowledge Graphs

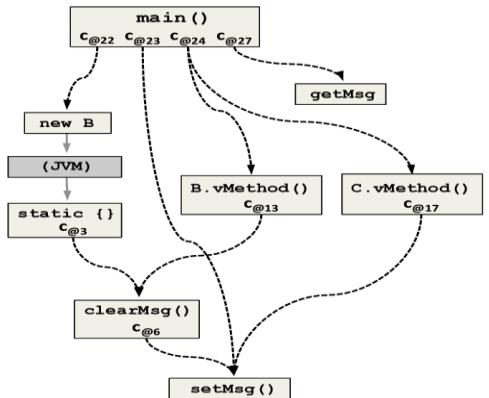


Image credit: [ResearchGate](#)

Code Graphs

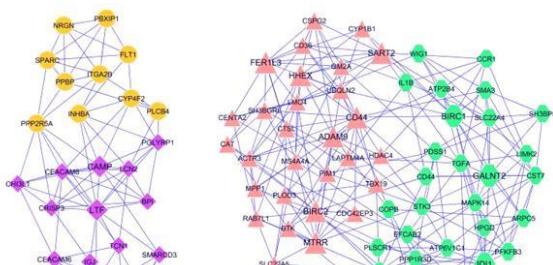


Image credit: [ese.wustl.edu](#)

Regulatory Networks

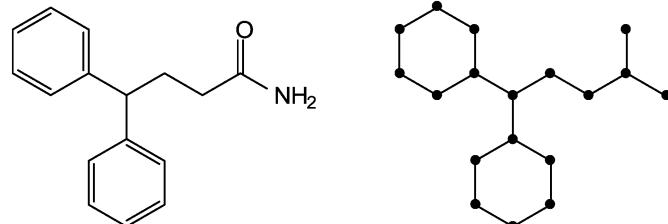


Image credit: [MDPI](#)

Molecules

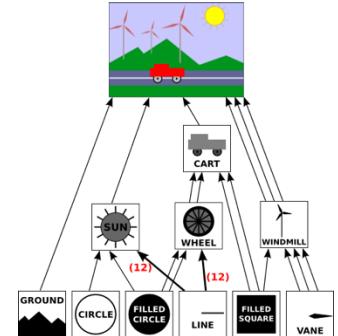


Image credit: [math.hws.edu](#)

Scene Graphs

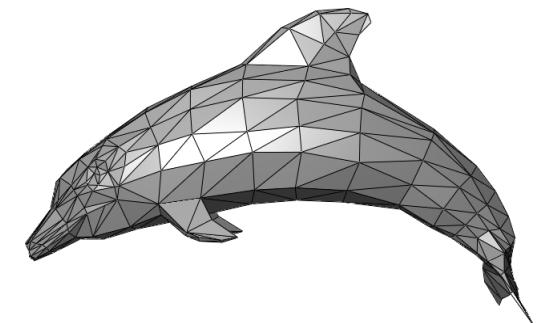


Image credit: [Wikipedia](#)

3D Shapes

Graph Machine Learning

Machine learning: predict from data

Observation:

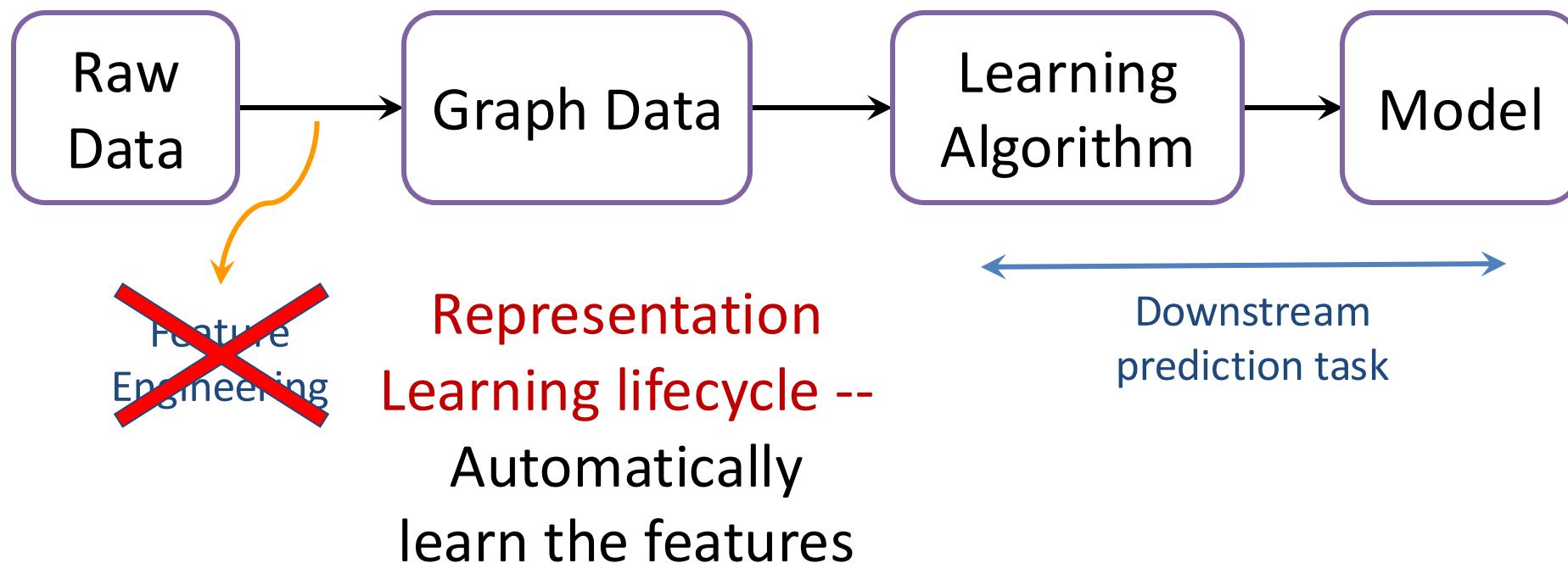
- Relational information – graphs – are ubiquitous
- Standard ML methods do not (explicitly) model relations

Graph machine learning:

- How do we take advantage of **relational information** for better prediction?

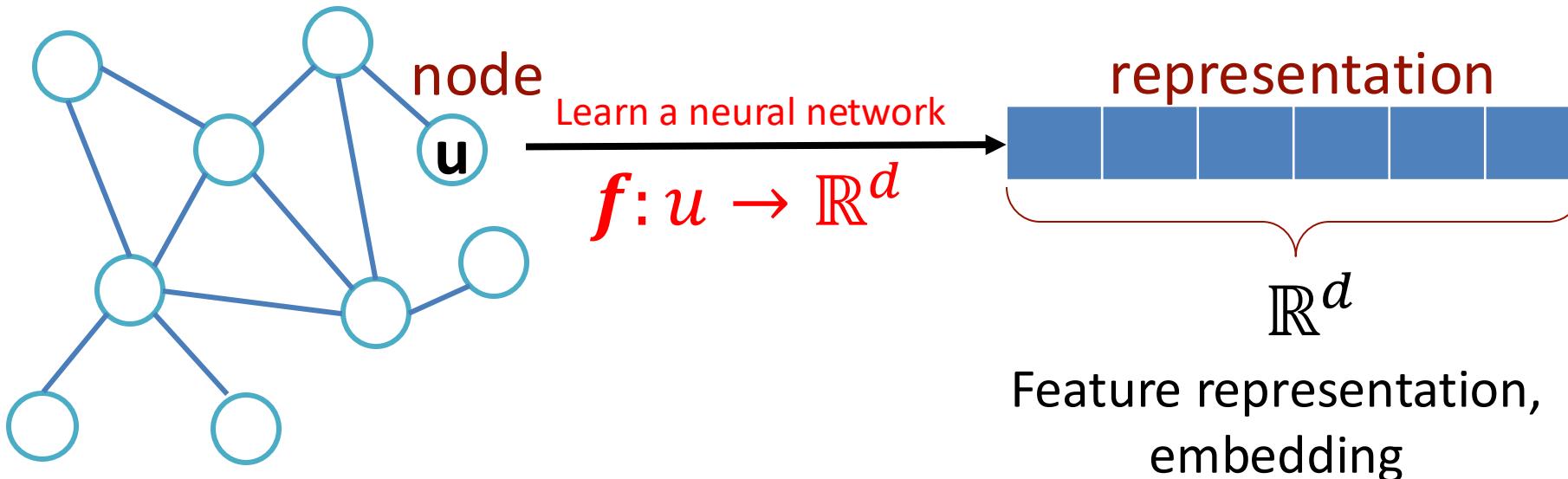
Traditional ML vs Representation Learning

- Traditional Machine Learning Lifecycle: Feature engineering

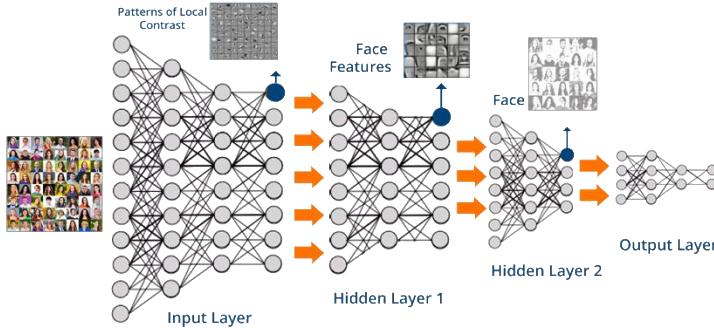
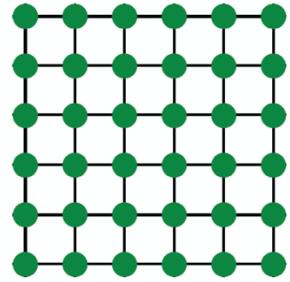


Graph Representation Learning

- Map nodes to d -dimensional embeddings to fit an objective function
 - E.g., Similar nodes in the graph are embedded close together



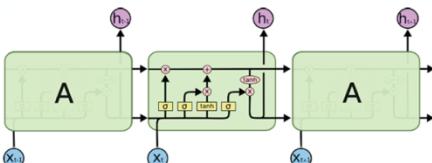
Modern Deep Learning Toolbox



Images



Text/Speech



Text
Audio signals



Images

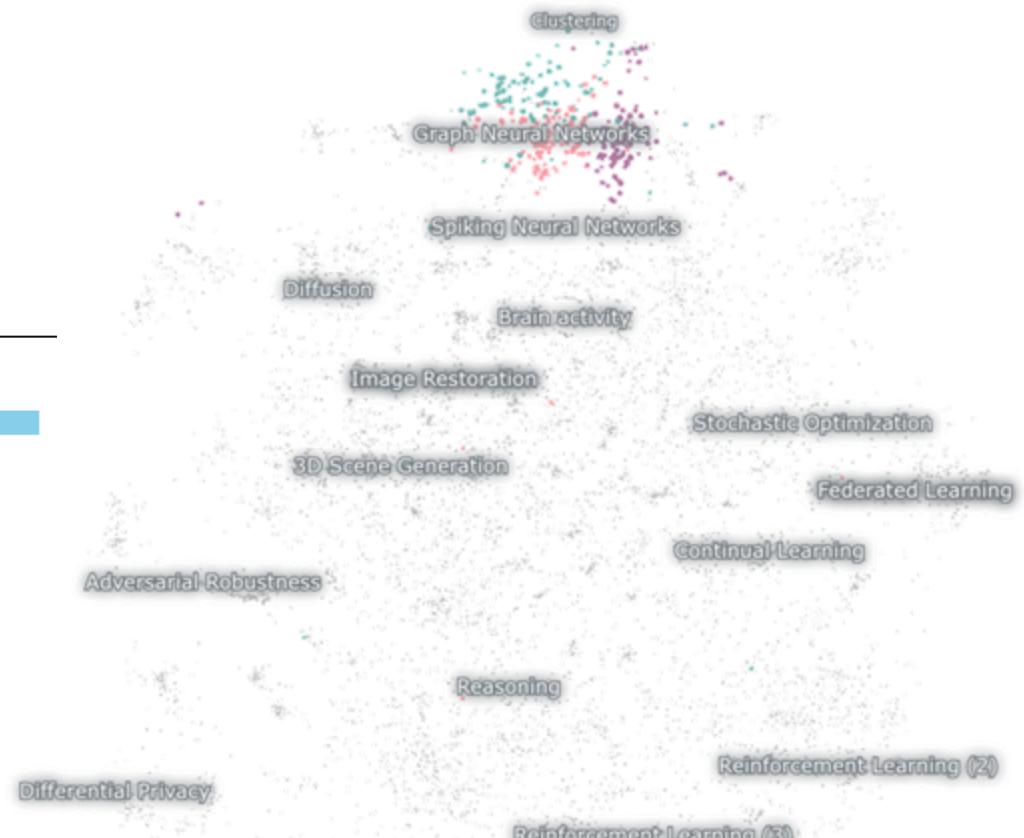
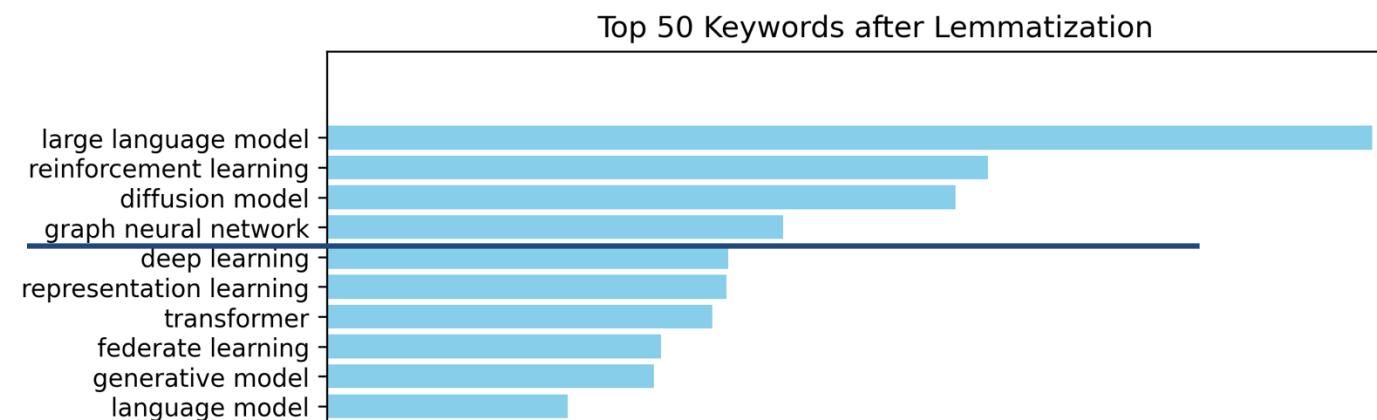
Modern deep learning toolbox is designed for simple sequences & grids.

Modern ML -> Graph ML

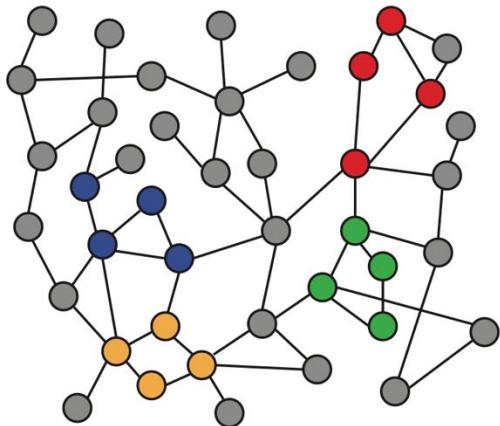
- Not everything can be represented as a sequence or a grid.
- How can we develop neural networks that are much more broadly applicable, beyond sequences and images?
- **Graph neural network** is a new class of neural networks, representing a new frontier of deep learning

Graph - Hot Subfield in AI

- ICLR 2024 keyword visualization



Machine Learning with Graphs is Hard



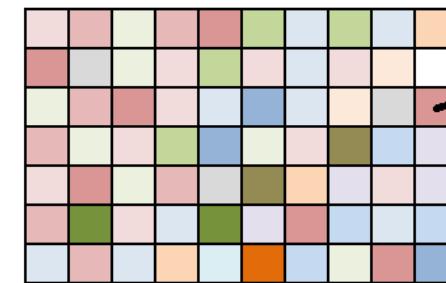
Graphs

VS.

This is a girl



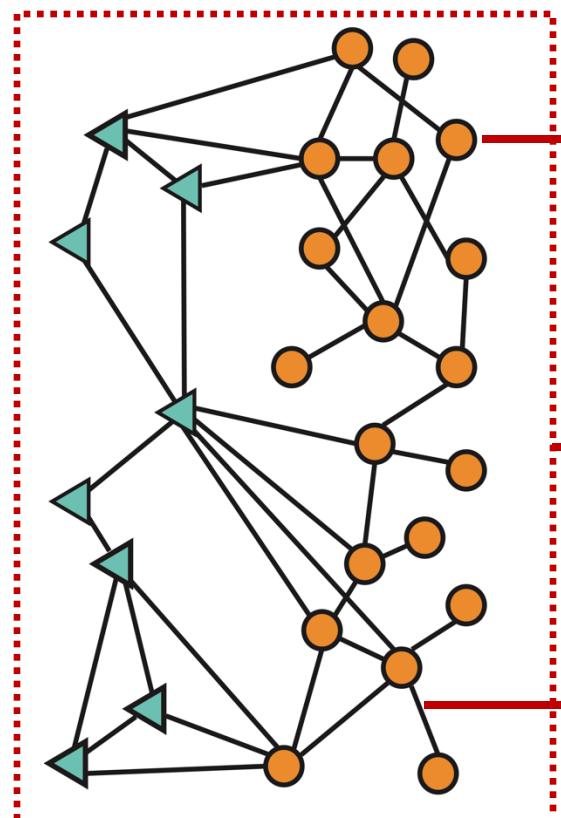
Text



Images

- **Arbitrary size and topological structure**
- **Nodes have no fixed ordering**

Graph Machine Learning Tasks



Node-level prediction

“Classify user by their type in a social network”

Graph-level prediction

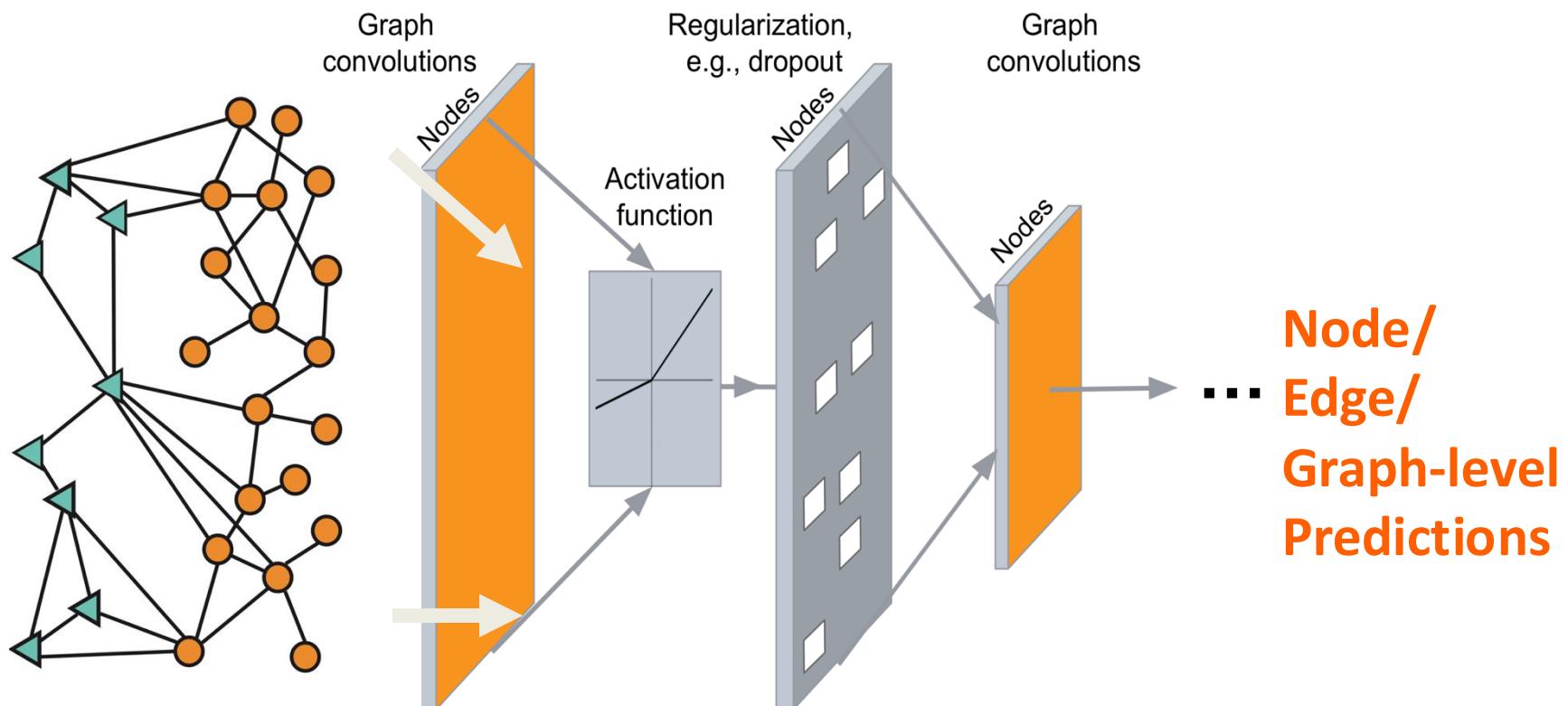
“Predict which molecules are drug-like”

Edge-level prediction

“Recommend item nodes to user nodes”

Deep Learning with Graphs

Input:
Network





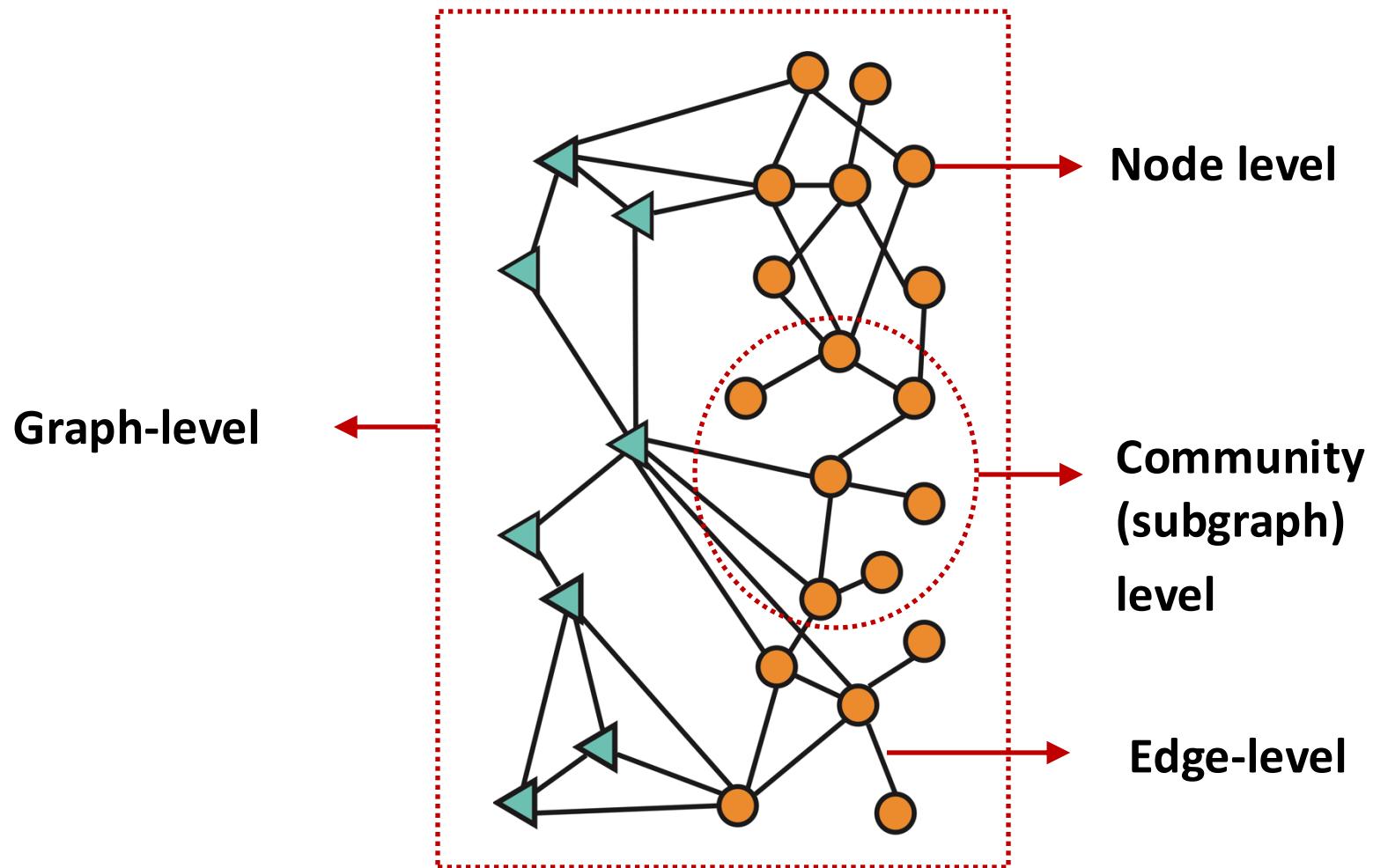
Reminder: Use Graphs Wisely

- Graph is a general language, but don't over-use it!
 - E.g., graphs subsumes lattices/sequences
- Suggested checklist
 - Will representing my data as graphs bring more information?
 - If your graph can be fully induced from existing data, think twice
 - E.g., construct K nearest neighbor graph from your embeddings, worth it?
 - Will representing my data as graphs lose information?
 - Graphs are unordered. Graph nodes/edges need features
 - E.g., molecule as graphs -> lose the distance information -> add as edge feature
 - Are there more efficient alternative representations?
 - Understand the trade-off between expressiveness & efficiency. How important is the relational info?
 - E.g., images as grids, text as sequences

Introduction

Applications of Graph ML

Different Types of Tasks



Classic Graph ML Tasks

- **Node classification:** Predict a property of a node
 - Example: Categorize online users / items
- **Link prediction:** Predict whether there are missing links between two nodes
 - Example: Knowledge graph completion
- **Graph classification:** Categorize different graphs
 - Example: Molecule property prediction
- **Clustering:** Detect if nodes form a community
 - Example: Social circle detection
- Other tasks:
 - **Graph generation:** Drug discovery
 - **Graph evolution:** Physical simulation

Classic Graph ML Tasks

- **Node classification:** Predict a property of a node

- Example: Category of a node

- **Link prediction:** Predict the existence of a link between two nodes

- Example: Known or unknown

- **Graph classification:** Predict a property of a graph

- Example: Model a graph

- **Clustering:** Detect communities in a graph

- Example: Social circle detection

- Other tasks:

- **Graph generation:** Drug discovery

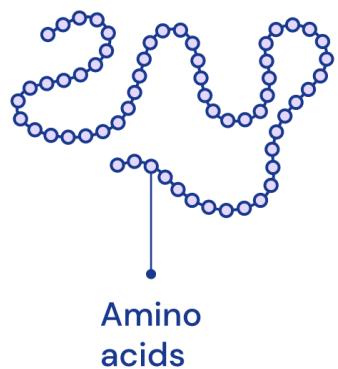
- **Graph evolution:** Physical simulation

These Graph ML tasks lead to
high-impact applications!

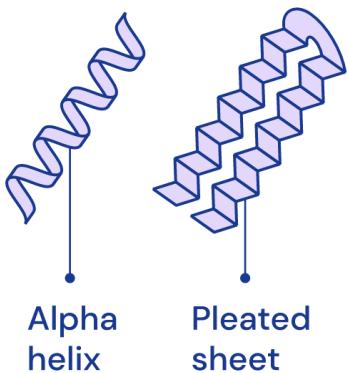
Example (1): Protein Folding

- A protein chain acquires its native 3D structure.

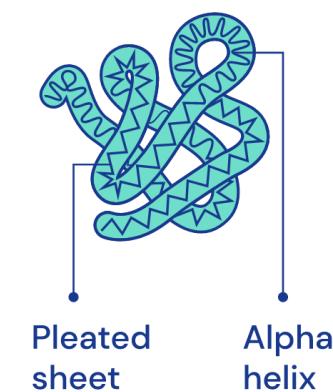
Every protein is made up of a sequence of amino acids bonded together



These amino acids interact locally to form shapes like helices and sheets



These shapes fold up on larger scales to form the full three-dimensional protein structure



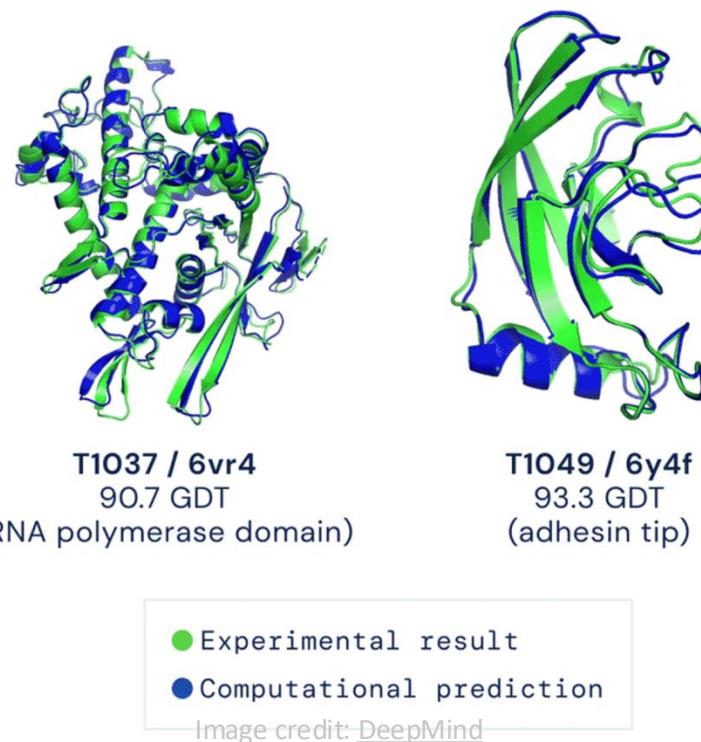
Proteins can interact with other proteins, performing functions such as signalling and transcribing DNA



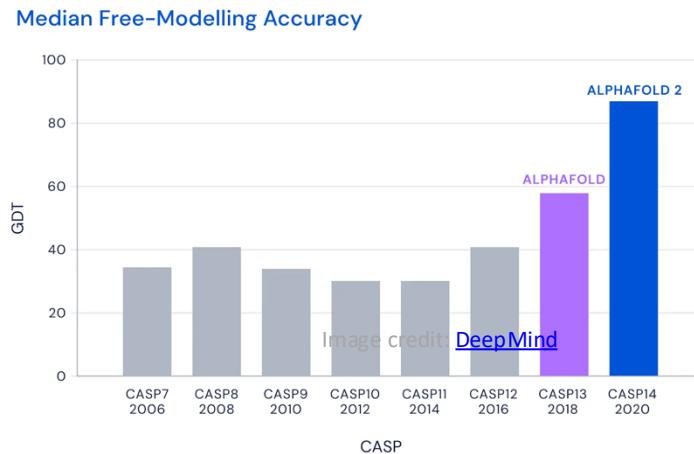
Image credit: [DeepMind](#)

Example (1): Protein Folding

- The Protein Folding Problem: Computationally predict a protein's 3D structure based solely on its amino acid sequence.



AlphaFold: Impact



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

12-14-20

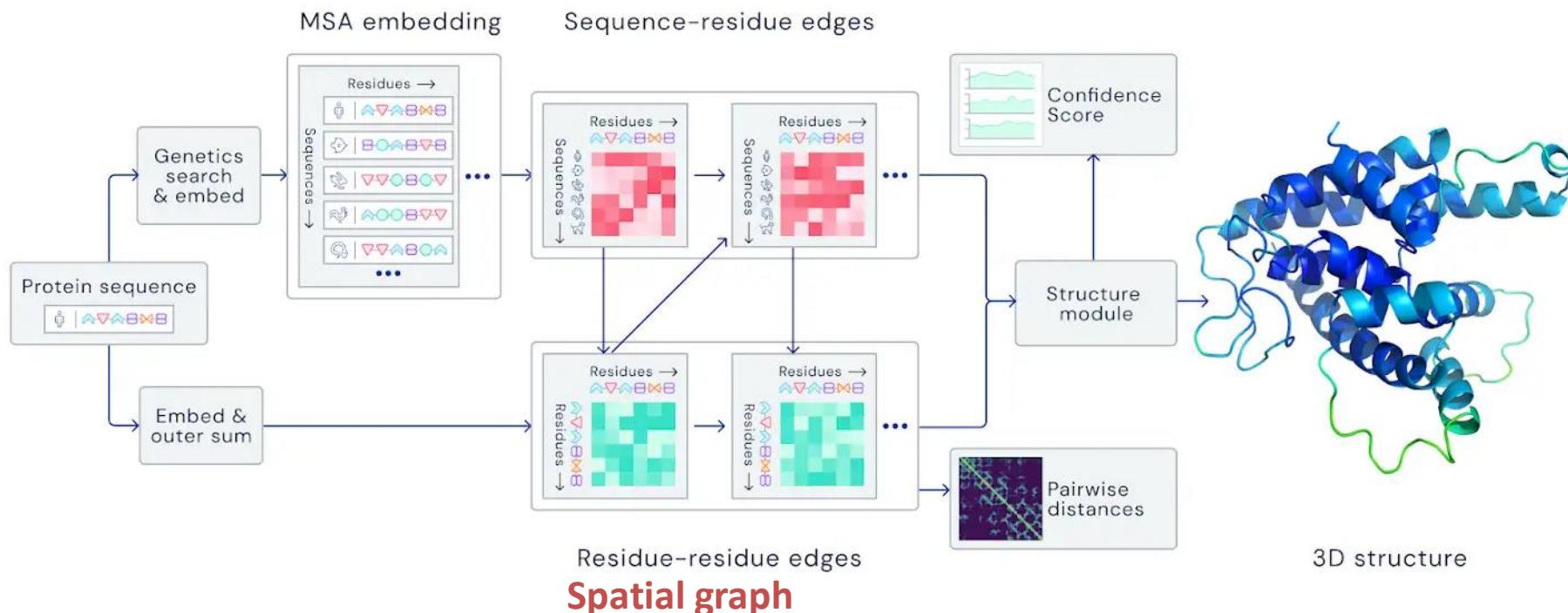
DeepMind's latest AI breakthrough could turbocharge drug discovery

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

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

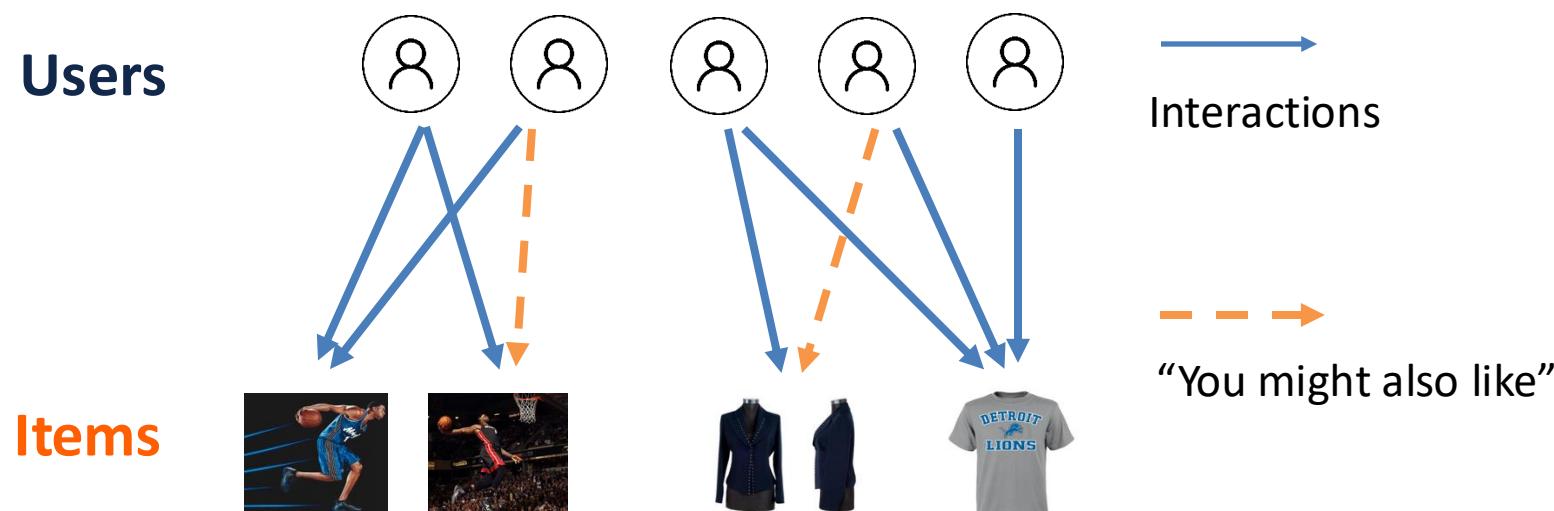
AlphaFold: Solving Protein Folding

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



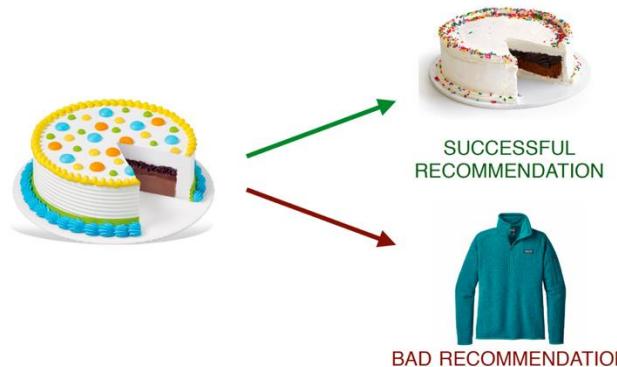
Example (2): Recommender Systems

- **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**



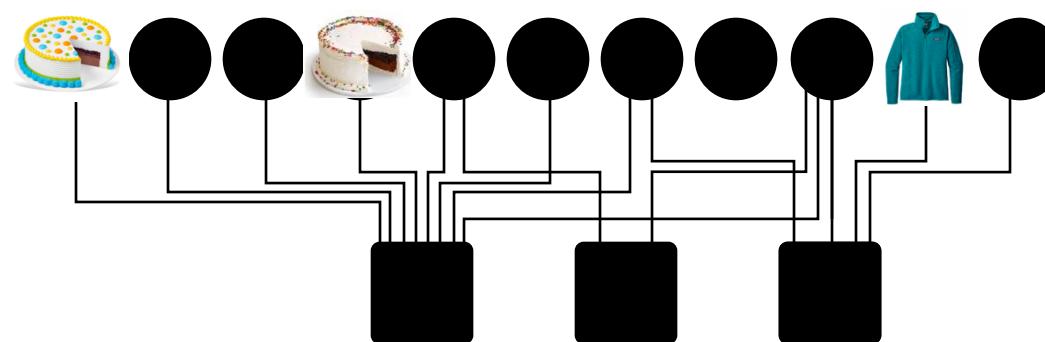
PinSage: Graph-based Recommender

- Task: Recommend related pins to users



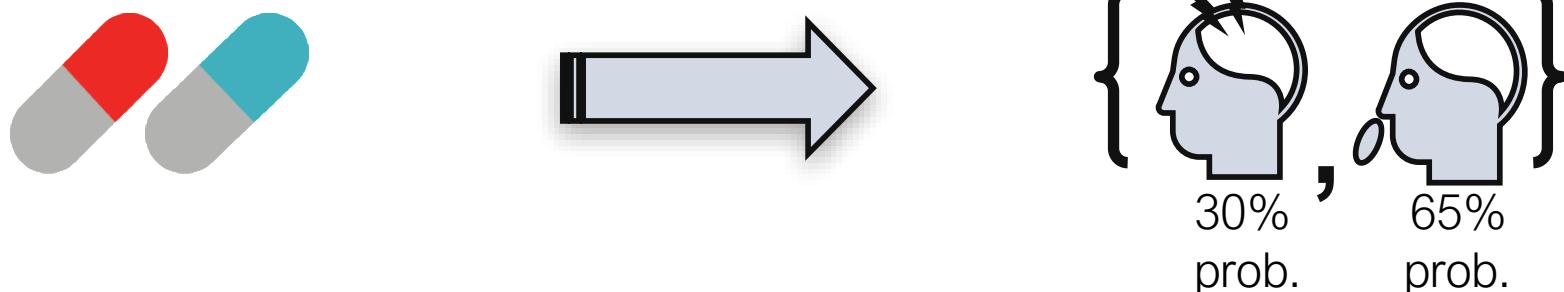
Task: Learn node embeddings z_i such that
 $d(z_{cake1}, z_{cake2}) < d(z_{cake1}, z_{sweater})$

Predict whether two nodes in a graph are related



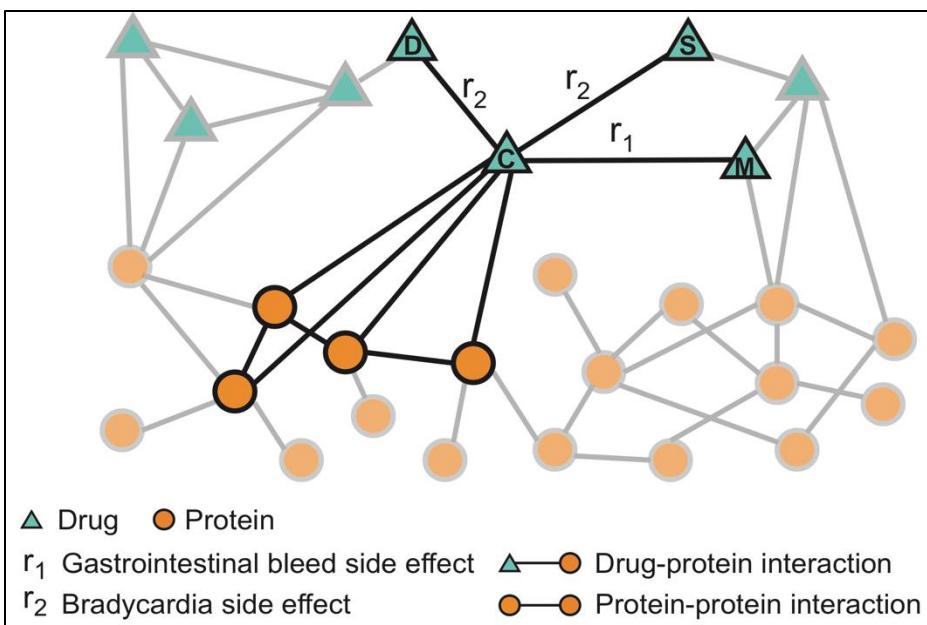
Example (3): Drug Side Effects

- 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.

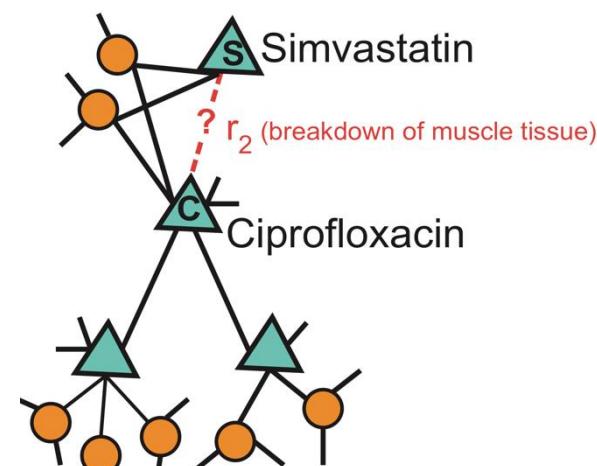


Biomedical Graph Link Prediction

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



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



Zitnik et al., [Modeling Polypharmacy Side Effects with Graph Convolutional Networks](#), Bioinformatics 2018

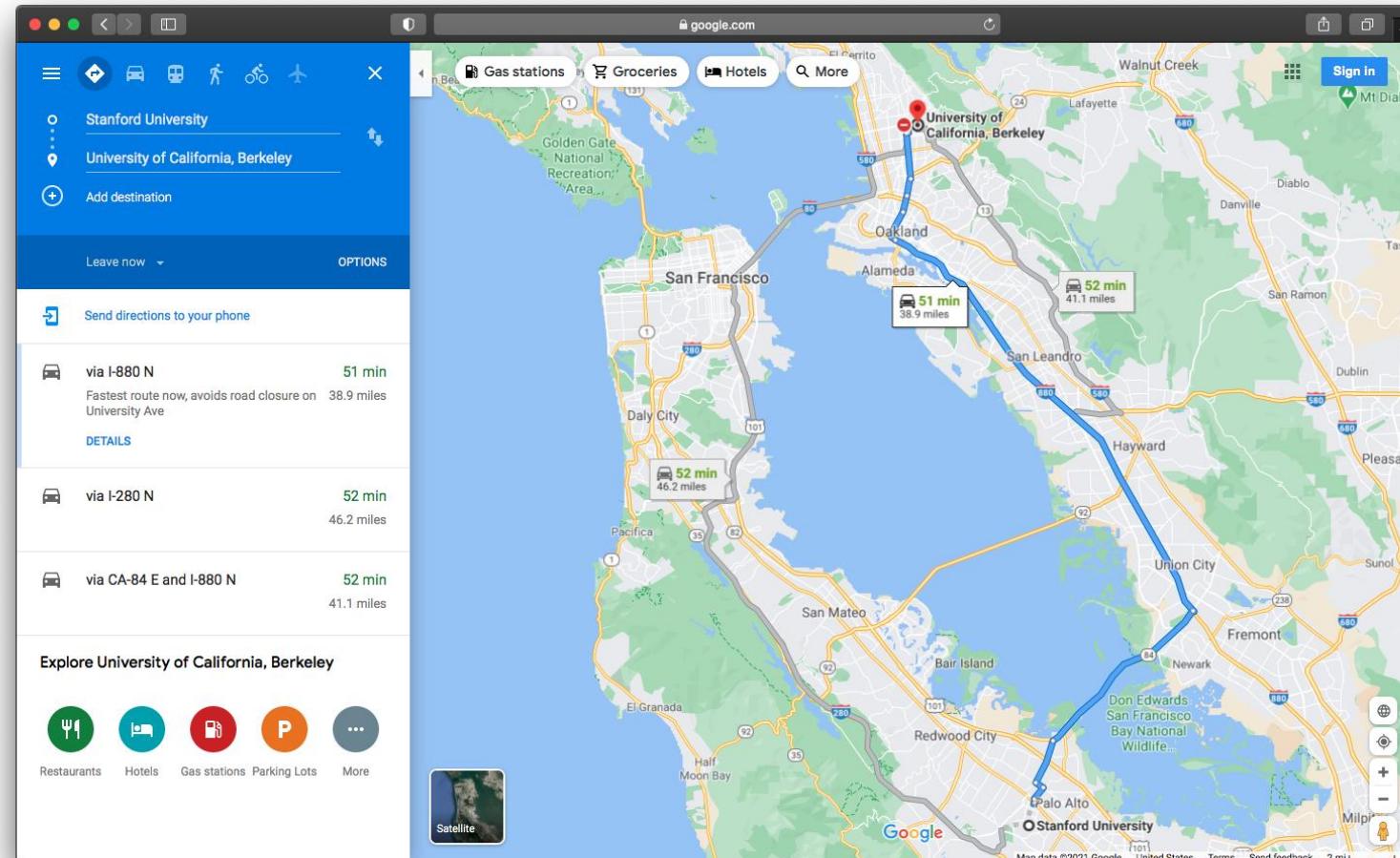
Results: *De novo* Predictions

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

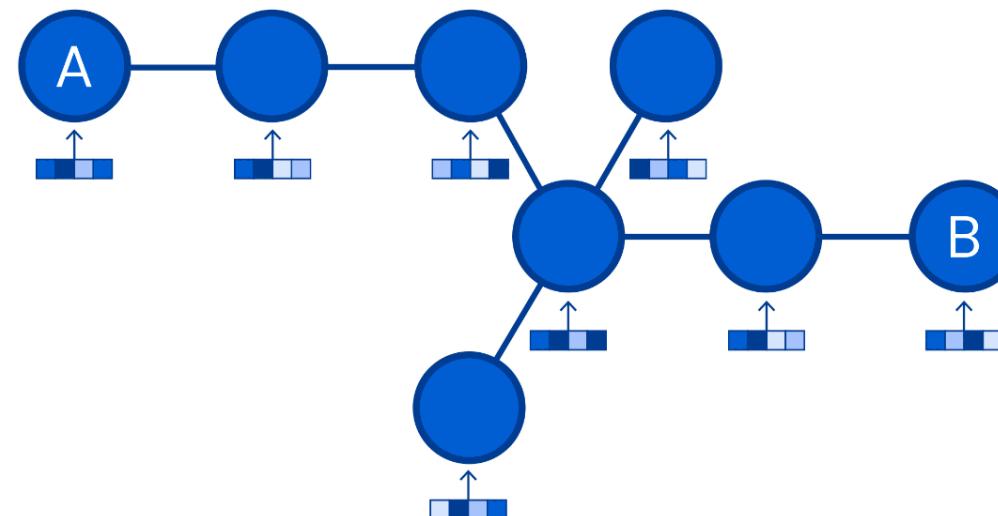
**Severe Rhabdomyolysis due to Presumed Drug Interactions
between Atorvastatin with Amlodipine and Ticagrelor**

Example (4): Traffic Prediction



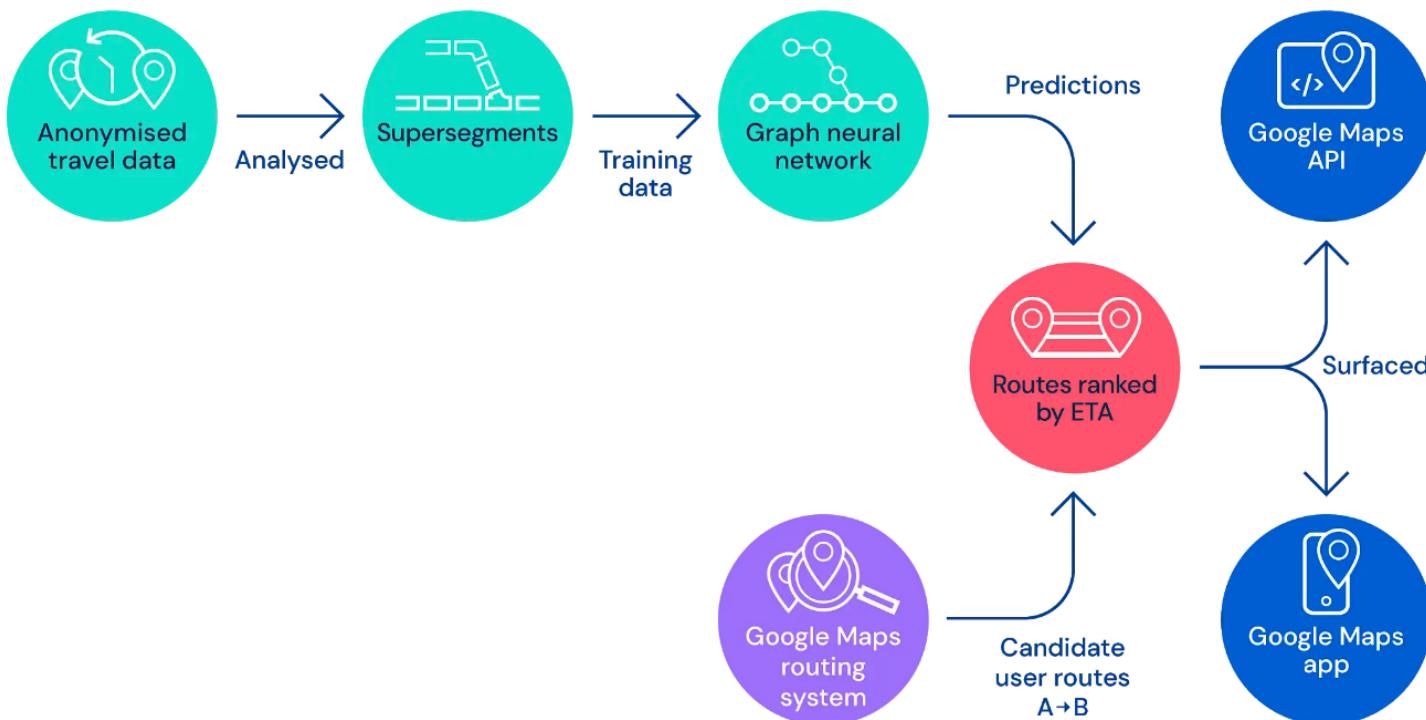
Road Network as a Graph

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



Traffic Prediction via GNN

- Predicting Time of Arrival with Graph Neural Networks

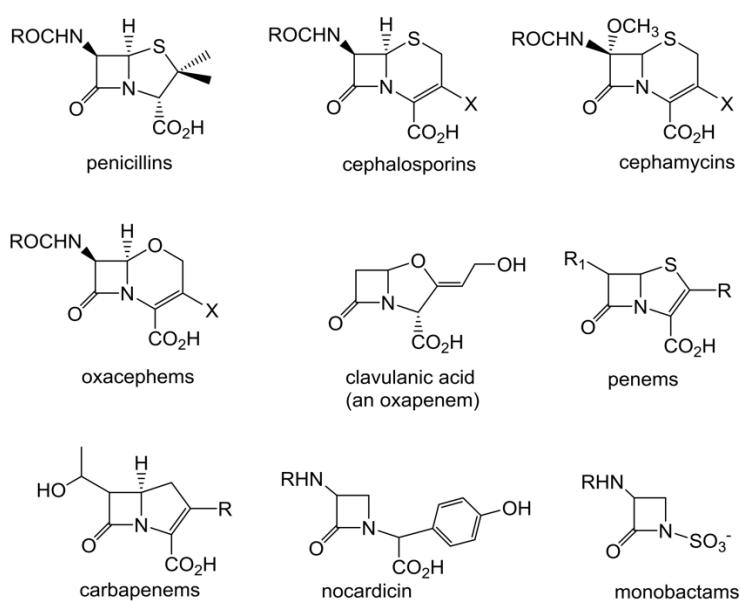


THE MODEL ARCHITECTURE FOR DETERMINING OPTIMAL ROUTES AND THEIR TRAVEL TIME.

Used in Google Maps

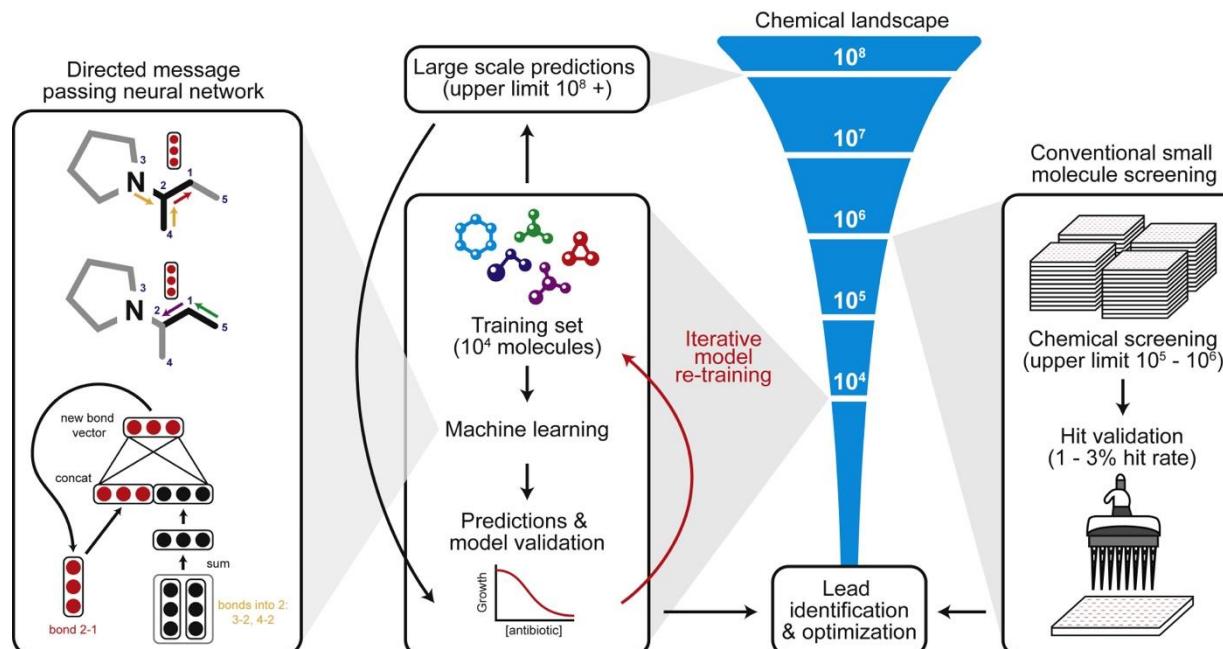
Example (5): Drug Discovery

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



Deep Learning for Antibiotic Discovery

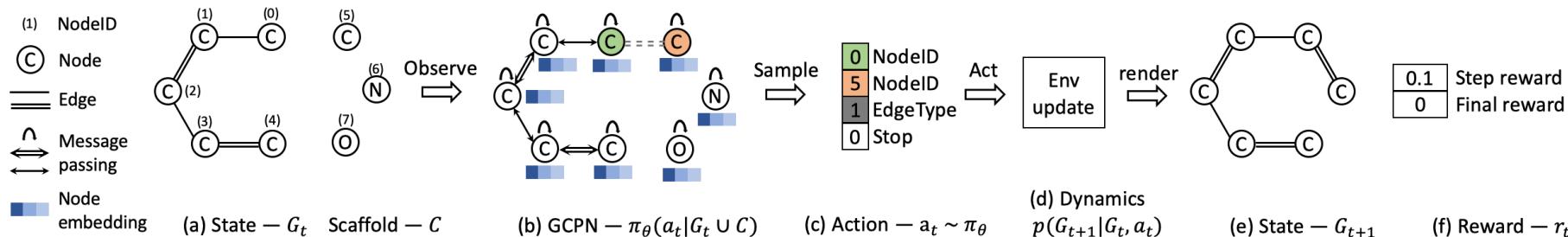
- A Graph Neural Network **graph classification model**
- Predict promising molecules from a pool of candidates



Stokes et al., [A Deep Learning Approach to Antibiotic Discovery](#), Cell 2020

Molecule Generation / Optimization

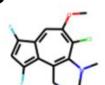
■ Graph generation: Generating novel molecules



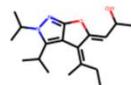
Use case 1: Generate novel molecules with high Drug likeness value



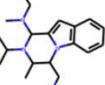
0.948



0.945



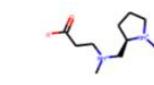
0.944



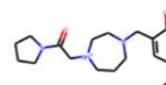
0.941

Drug likeness

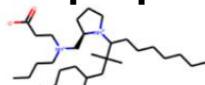
Use case 2: Optimize existing molecules to have desirable properties



-8.32



-5.55



-0.71



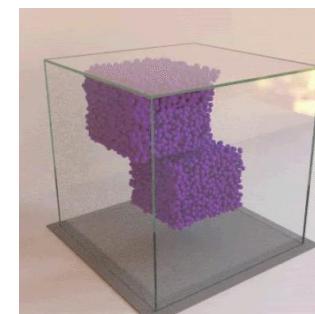
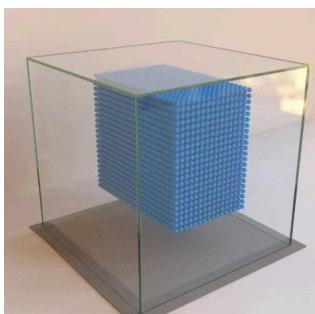
-1.78

You et al., Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation, NeurIPS 2018

Example (6): Physics Simulation

- Physical simulation as a graph:

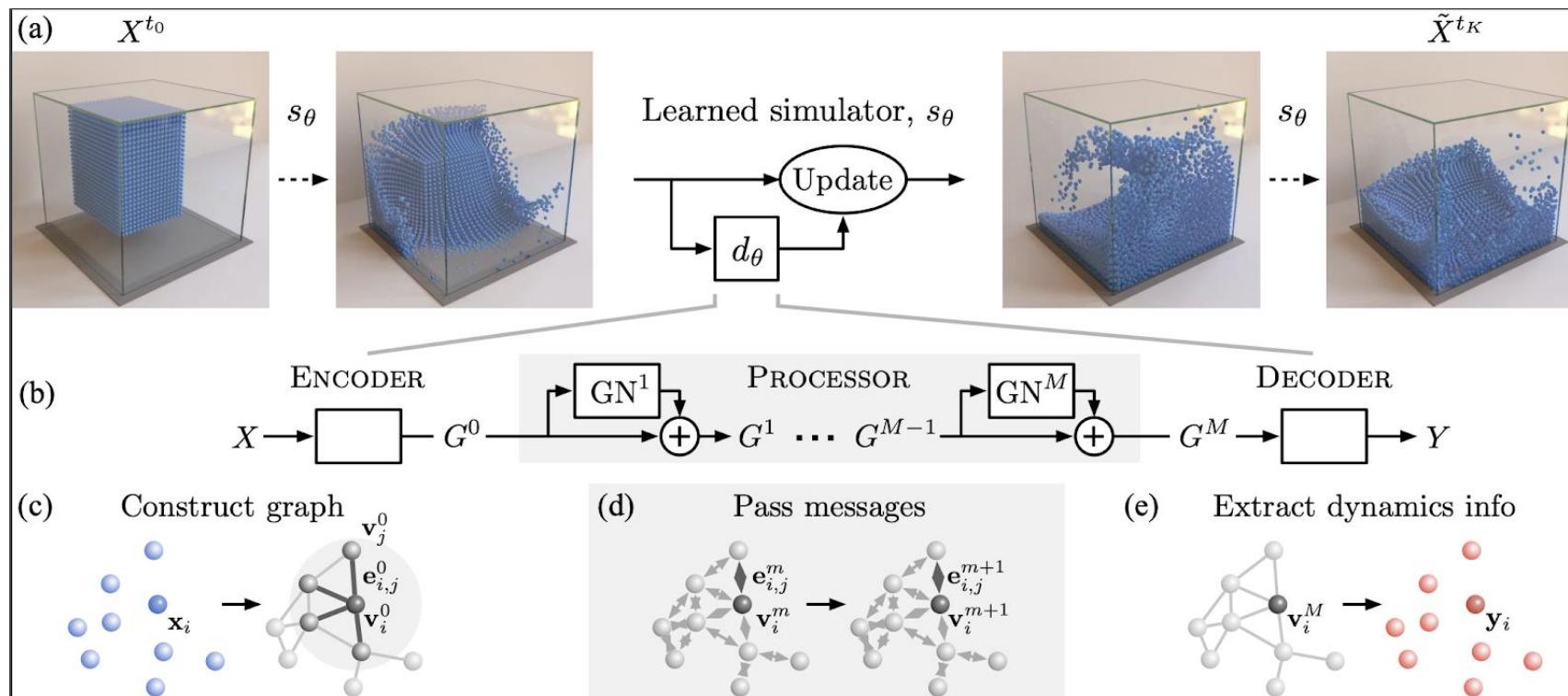
- Nodes:** Particles
- Edges:** Interaction between particles



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

Simulation Learning Framework

- A graph evolution task:
 - Goal: Predict how a graph will evolve over time



Introduction

Course Logistics

Teaching Staff

Instructor



Jiaxuan You

Assistant



Zirui Cheng

Course Logistics

- Meeting Times:
 - 12:30 PM - 01:45 PM, Wednesday & Friday
 - Urbana-Champaign Campus | Siebel Center Room 1304
 - Slides of the lectures will be recorded and posted on Canvas
- Structure of lectures:
 - 60-70 minutes of a lecture. Feel free to ask questions
 - More details on short student presentations will be announced later
- Office hours: 1 hour (Instructor) + 2 hours (TA). Details to be announced.

Course Website



Website: <https://ulab-uiuc.github.io/CS598/>

Announcement: Canvas

Prerequisites

- The course is mostly self-contained
- Minimum prerequisites
 - Deep learning: basic concepts
 - Python programming
- Recommended background
 - Deep learning: PyTorch
 - Graph: data structure
 - Machine learning, Probability and statistics
 - Comfortable of using (free) LLM Tools: ChatGPT/Gemini/Copilot/...

Course Objective

Takeaways from this course:

- **Knowledge** about graph deep learning
 - Core knowledge: Insights, Coding, Math
 - Latest knowledge: Recent research papers
- **Training** for AI research
 - Experience the **full lifecycle as an AI researcher**
 - Read, Ideate, Discuss, Code, Write, Review, Present

Schedule (1)

Week	Date	Knowledge learning	Research training	Events	Deadlines
1	Aug 28 Wed	Introduction	Paper reading & analysis		
	Aug 30 Fri	Graph learning tasks	Paper reading & analysis	Writing task, out	
2	Sept 04 Wed	"Shallow" graph learning	Paper reading & analysis		
	Sept 06 Fri	Graph neural networks: perspective	Paper reading & analysis		
3	Sept 11 Wed	Graph neural networks: model I	Paper reading & analysis		
	Sept 13 Fri	Graph neural networks: model II	Paper reading & analysis		Writing task due
4	Sept 18 Wed	Paper reading discussions	Ideate & discussion		
	Sept 20 Fri	Graph neural networks: objective	Ideate & discussion	Proposal task, out	
5	Sept 25 Wed	Graph neural networks: pipeline	Ideate & discussion		
	Sept 27 Fri	Graph neural networks: theory	Ideate & discussion		
6	Oct 02 Wed	Graph neural networks: add-ons	Ideate & discussion		
	Oct 04 Fri	GNN implementation: PyG & GraphGym	Ideate & discussion		Proposal due
7	Oct 09 Wed	Project idea discussions	Prototype implementation		
	Oct 11 Fri	Beyond simple graphs: heterogeneous graphs	Prototype implementation	Submission task, out	

Schedule (2)

8	Oct 18 Fri	Beyond simple graphs: knowledge graph reasoning	Prototype implementation		
9	Oct 23 Wed	Beyond prediction: graph generative models	Prototype implementation		
10	Oct 25 Fri	Beyond message passing: expressive GNNs	Prototype implementation		
11	Oct 30 Wed	Beyond small graphs: scale GNNs to large graphs	Paper Writing		
12	Nov 01 Fri	Beyond small GNNs: graph foundation models	Paper Writing		
13	Nov 06 Wed	Beyond sparse graphs: graph transformers	Paper Writing		
14	Nov 08 Fri	GNN applications: recommender systems	Paper Writing		Submission due
15	Nov 13 Wed	GNN applications: graph mining	Review & Response	Review & response task, out	
16	Nov 15 Fri	GNN applications: science	Review & Response		
17	Nov 20 Wed	Conclusion	Review & Response		
18	Nov 22 Fri	Course project presentation 1	Review & Response		
19	Dec 04 Wed	Course project presentation 2	Presentation		
20	Dec 06 Fri	Course project presentation 3	Presentation		Review & response due

We Embrace "ChatGPT Moment"



- Fundamentally changed AI & beyond
- In this course, **we embrace LLMs**
 - **Lectures:** feel free to ask LLMs to help understand lecture materials
 - **Assignments:** feel free to use LLMs to help, but you *must include your prompt*
 - **Projects:** focus on ideas that naively applying LLMs would fail
- But we cherish **non-LLM-able** things
 - **Research training:** asking good questions is always more important than answering given questions

Grading (To be finalized)

- Final grade will be composed of:
 - Assignments: 20%
 - 4 coding assignments using Google Colab - each 5%
 - Course project: 80%
 - Paper reading and analysis – 15%
 - Ideate and discussion – 15%
 - Prototype implementation – 15%
 - Paper writing – 15%
 - Review and response – 10%
 - Presentations – 10%
 - Details will be announced in Week 2