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

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

Course Assistants
- Xikun Zhang (Head CA)
- Hamed Nilforoshan
- Aditya Agrawal
- Abhinav Garg

Guest Instructor
- Joshua Robinson
- Matthew Jin
- Yunqi Li
- Tolu Oyeniyi
- Chenshu (Jupiter) Zhu
- Pratham Soni
- Anirudh Sriram
Logistics: Website

- [http://cs224w.stanford.edu](http://cs224w.stanford.edu)
  - Slides posted before the class
- **Readings:**
  - [Graph Representation Learning Book](http://cs224w.stanford.edu) by Will Hamilton
  - Research papers
- **Optional readings:**
  - Papers and pointers to additional literature
  - This will be very useful for course projects
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 2\textsuperscript{nd} week of the course
  - See [http://web.stanford.edu/class/cs224w/oh.html](http://web.stanford.edu/class/cs224w/oh.html) for Zoom links and link to QueueStatus
  - Schedule to be announced by end of week
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 2 Late Periods (LP) 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!
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](https://cloud.google.com/free/docs/gcp-free-tier))

Read: [http://cs224w.stanford.edu/info.html](http://cs224w.stanford.edu/info.html)
## Course Schedule

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

Computer Networks

Disease Pathways

Food Webs

Particle Networks

Underground Networks

Image credit: SalientNetworks


Image credit: Pinterest

Image credit: visitlondon.com
Many Types of Data are Graphs (2)

Social Networks

Economic Networks

Communication Networks

Citation Networks

Internet

Networks of Neurons
Many Types of Data are Graphs (3)

Knowledge Graphs

<table>
<thead>
<tr>
<th>Image credit: Maximilian Nickel et al</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spock</td>
</tr>
<tr>
<td>played</td>
</tr>
<tr>
<td>Leonard Nimoy</td>
</tr>
<tr>
<td>Science Fiction</td>
</tr>
<tr>
<td>genre</td>
</tr>
<tr>
<td>Star Trek</td>
</tr>
<tr>
<td>Obi-Wan Kenobi</td>
</tr>
<tr>
<td>characterIn</td>
</tr>
<tr>
<td>Star Wars</td>
</tr>
<tr>
<td>Alec Guinness</td>
</tr>
<tr>
<td>starredIn</td>
</tr>
</tbody>
</table>

Regulatory Networks

<table>
<thead>
<tr>
<th>Image credit: ese.wustl.edu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regulatory Network Diagram</td>
</tr>
</tbody>
</table>

Scene Graphs

<table>
<thead>
<tr>
<th>Image credit: math.hws.edu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene Graph Diagram</td>
</tr>
</tbody>
</table>

Code Graphs

<table>
<thead>
<tr>
<th>Image credit: ResearchGate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Graph Diagram</td>
</tr>
</tbody>
</table>

Molecules

<table>
<thead>
<tr>
<th>Image credit: MDPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Molecule Diagram</td>
</tr>
</tbody>
</table>

3D Shapes

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Shape Diagram</td>
</tr>
</tbody>
</table>

11/14/23
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?
Modern deep learning toolbox is designed for simple sequences & grids.
Modern deep learning toolbox is designed for sequences & grids
How can we develop neural networks that are much more broadly applicable?

Graphs are the new frontier of deep learning
Hot subfield in ML

50 MOST APPEARED KEYWORDS (2023)

- reinforcement learning
- deep learning
- representation learning
- graph neural network
- transformer
- federate learning
- self-supervised learning
- contrastive learning
- robustness
- generative model
- continual learning
- neural network
- transfer learning
- diffusion model
- generalization
- language model
- computer vision
- knowledge distillation
- vision transformer
- offline reinforcement learning
- optimization
- fairness
- differential privacy
- semi-supervised learning
- unsupervised learning
- deep reinforcement learning
- machine learning
- interpretability
- meta-learning
- adversarial robustness
- multi-agent reinforcement learning
- large language model
- optimal transport
- data augmentation
- few-shot learning
- domain generalization
- NLP
- adversarial attack

ICLR 2023 keywords
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

- **Actors**
  - Peter
  - Mary
  - Tom
  - Albert

- **Proteins**
  - Protein 1
  - Protein 2
  - Protein 5
  - Protein 9

**Example Graphs**

- **Left**: Actors connected by movies.
- **Right**: Connections between people.

**Mathematical Notation**

- $|N|=4$
- $|E|=4$
A heterogeneous graph is defined as

\[ G = (V, E, R, T) \]

- 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
Undirected

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

- **Other considerations:**
  - Weights
  - Properties

Directed

- **Links:** directed

- **Other considerations:**
  - Types
  - Attributes
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

Graph-level prediction, Graph generation

Node level

Community (subgraph) level

Edge-level
Stanford CS224W: Node-Level Tasks
Machine Learning Tasks: Review

- **Node-level prediction**
- **Link-level prediction**
- **Graph-level prediction**
Node-Level Tasks

Node classification
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 $u$:

- $a$, $b$, $c$, $d$

Graphlets of node $u$:

- $a, b, c, d$
- $[2, 1, 0, 2]$
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 defined 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

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?

DeepMind’s latest AI breakthrough could turbocharge drug discovery
**Key idea:** “Spatial graph”

- **Nodes:** Amino acids in a protein sequence
- **Edges:** Proximity between amino acids (residues)
Stanford CS224W: Link Prediction
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.
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

Users interact with items through interactions, which can be visualized as arrows connecting users to items. The goal is to recommend items that users might also like.
Task: Recommend related pins to users

Query pin

Task: Learn node embeddings $z_i$ such that

$$d(z_{\text{cake}_1}, z_{\text{cake}_2}) < d(z_{\text{cake}_1}, z_{\text{sweater}})$$

Predict whether two nodes in a graph are related
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

30% prob., 65% prob.
**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

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

*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
- **Goal**: We want to 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
Predicting Time of Arrival with Graph Neural Networks

- Used in Google Maps

Image credit: DeepMind
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

Example (3): Physics Simulation

Physical simulation as a graph:
- **Nodes**: Particles
- **Edges**: Interaction between particles
A graph evolution task:

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

Sanchez-Gonzalez et al., *Learning to simulate complex physics with graph networks*, ICML 2020

Simulation Learning Framework
Application: Weather forecasting

Summary

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