Graph Learning Basics

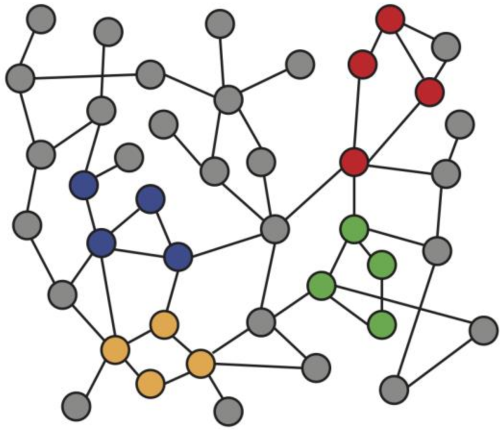
Jiaxuan You

Assistant Professor at UIUC CDS

CS512: Data Mining Principles, 2025 Fall

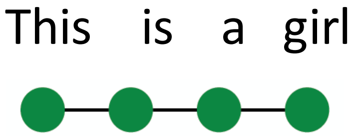
https://ulab-uiuc.github.io/CS512/

Recap: Machine Learning with Graphs is Hard

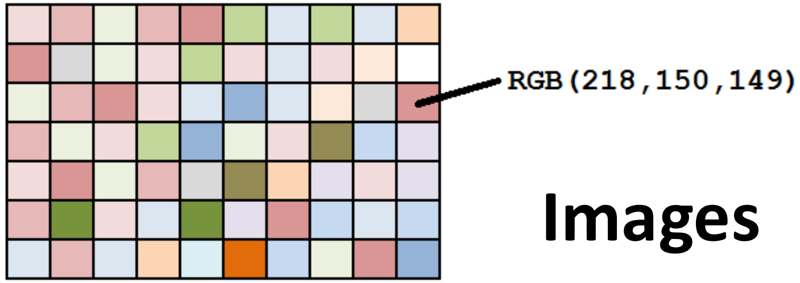


Graphs

vs.



Text



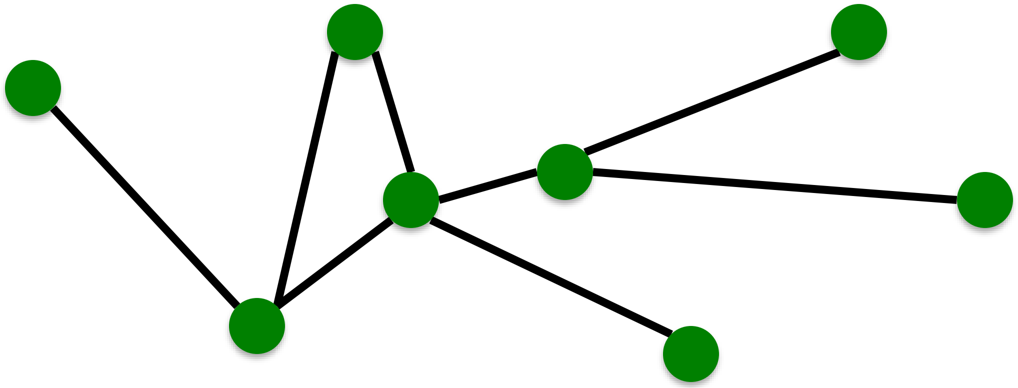
Images

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

Graph Learning Basics

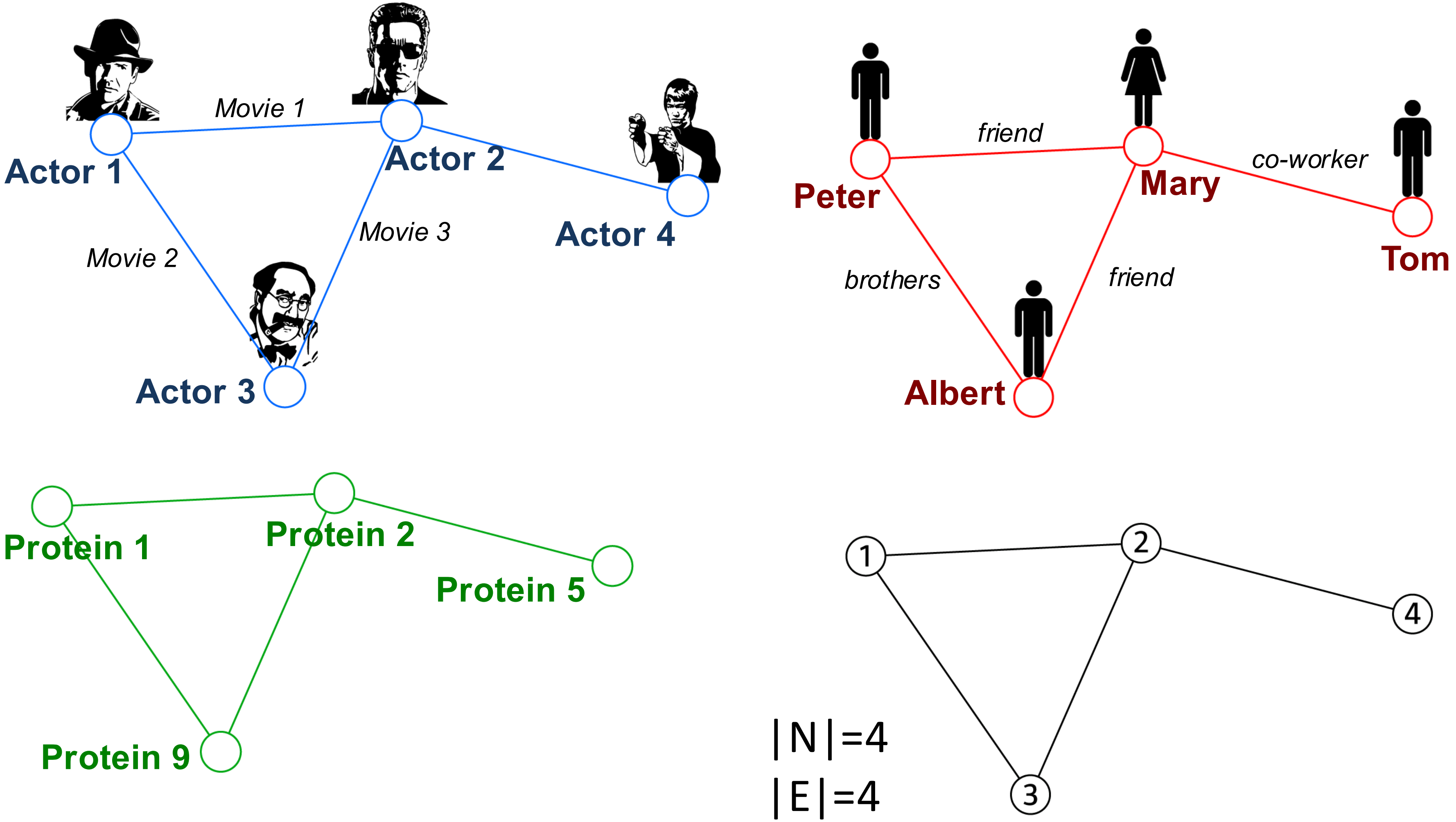
Graph Representation Basics

Components of a Network



▪ **Objects**: nodes, vertices	***N***
▪ **Interactions**: links, edges	***E***
▪ **System**: network, graph	***G(N,E)***

Graphs: A Common Language



$|N|=4$

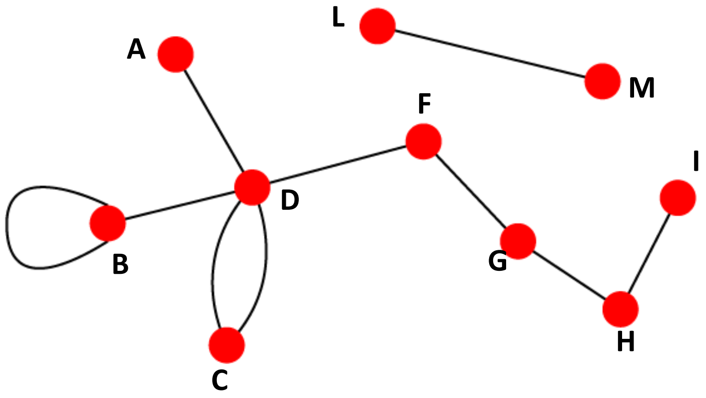
$|E|=4$

How do you define a graph?

- **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

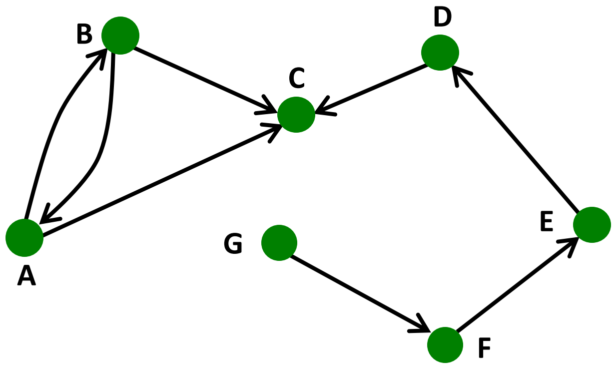
Directed vs. Undirected Graphs

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



  - Examples:
    - Collaborations
    - Friendship on Facebook
    - Pairs of positive/negative samples in contrastive learning

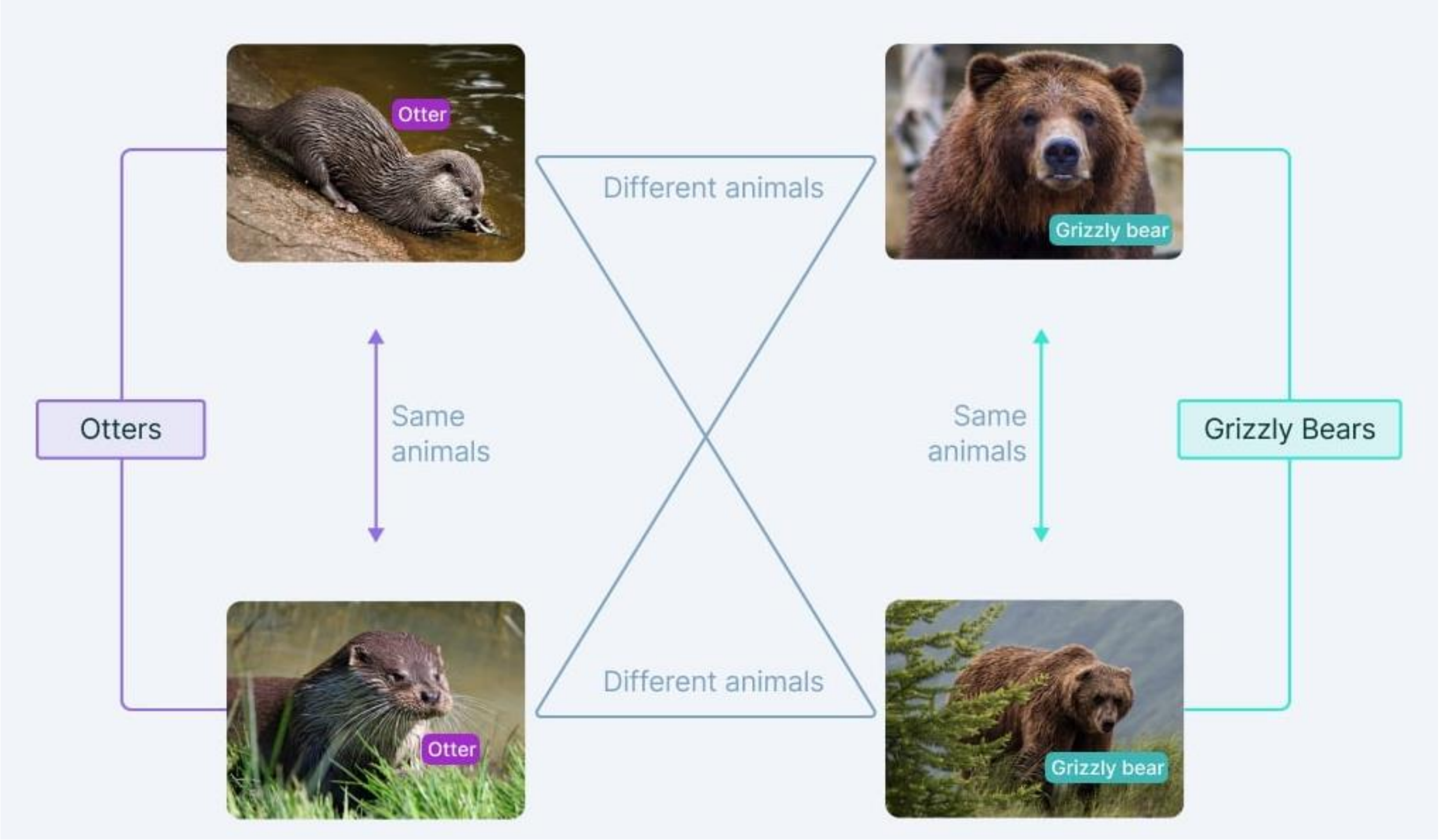
- **Directed**
  - Links: directed (arcs)



  - Examples:
    - Phone calls
    - Following on Twitter
    - Computational graphs in deep learning

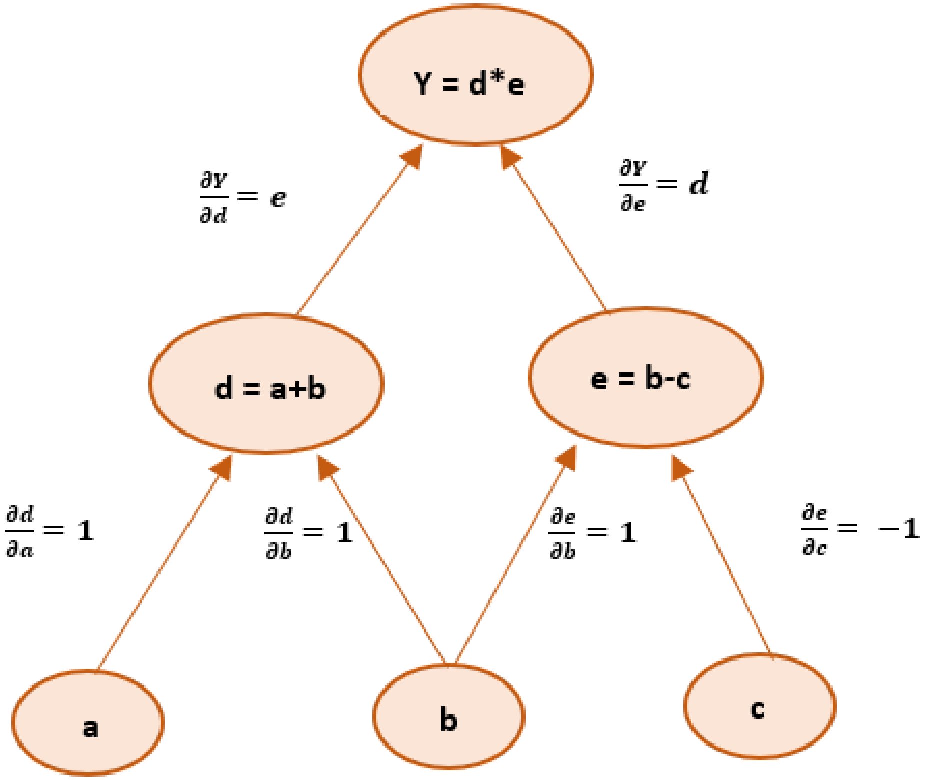
Undirected Graph Example in DL

- Contrastive learning

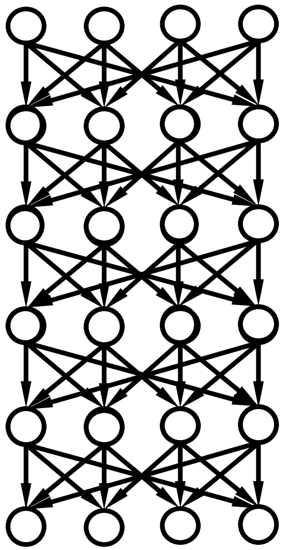


Directed Graph Examples in DL

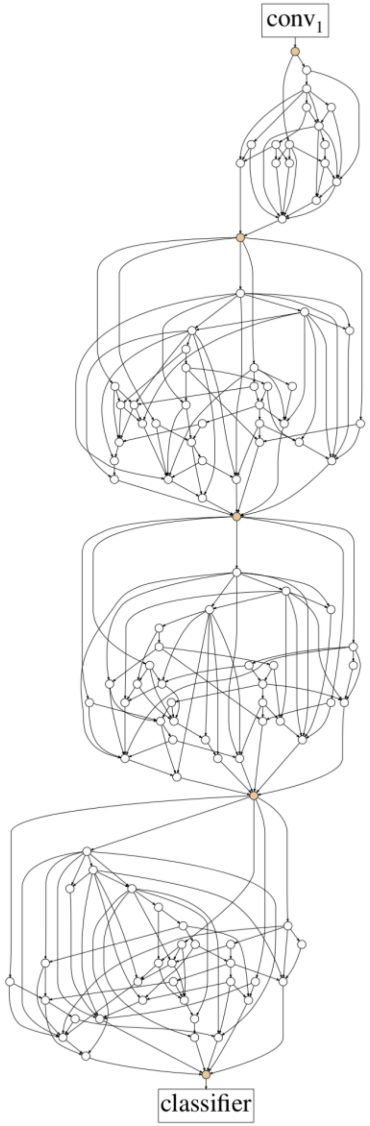
- Computational graphs



A 5-layer
Neural network



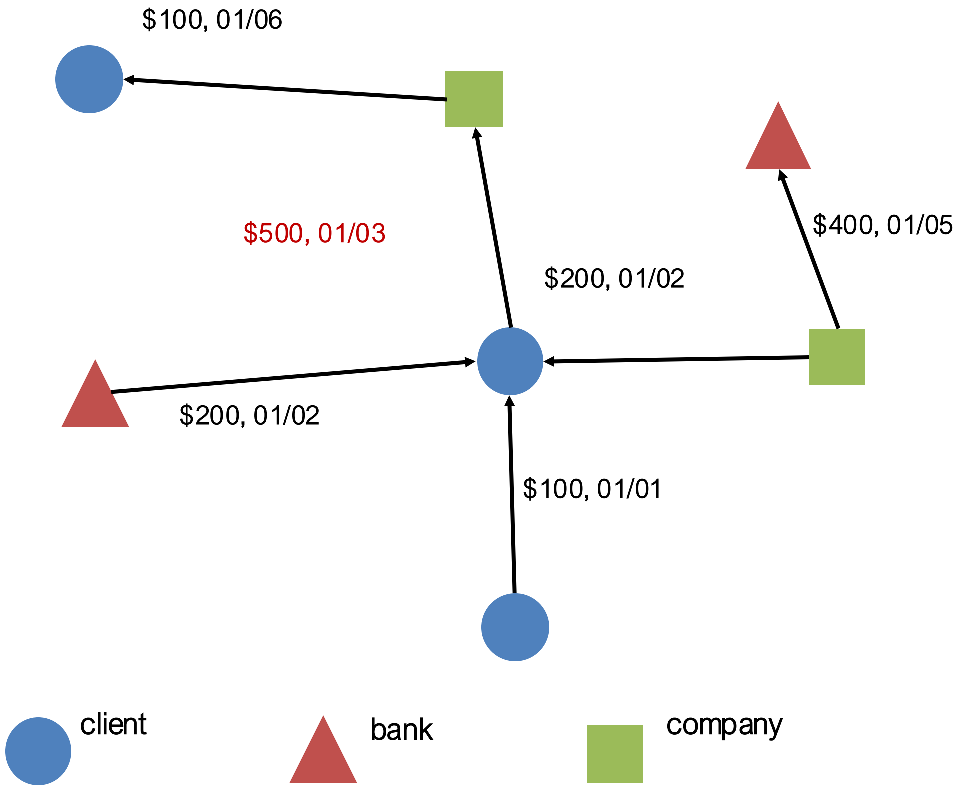
RandWire, Xie et al., 2019



Dynamic Graphs

- **Dynamic graph representation option 1: Graph + timestamp**
  $$\boldsymbol{G = (V, E, T)}$$
  - Nodes $v_i \in V$
  - Edges $(v_i, v_j) \in E$
  - Timestamps $T(v_i), T(v_i, v_j)$
- **Dynamic graph representation option 2: Snapshots of graphs**
  - Each snapshot is a standard graph G_t
  - A dynamic graph is a series of graph snapshots $G = (G_1, \dots, G_T)$

Dynamic Graph Example: Financial Networks



- Transaction-based approach
  - "On 01/03, Client A sends Company B $500"
- **Graph-based approach**
  - Represent a transaction in a much broader context
  - A dynamic network, changing over time

Heterogeneous Graphs

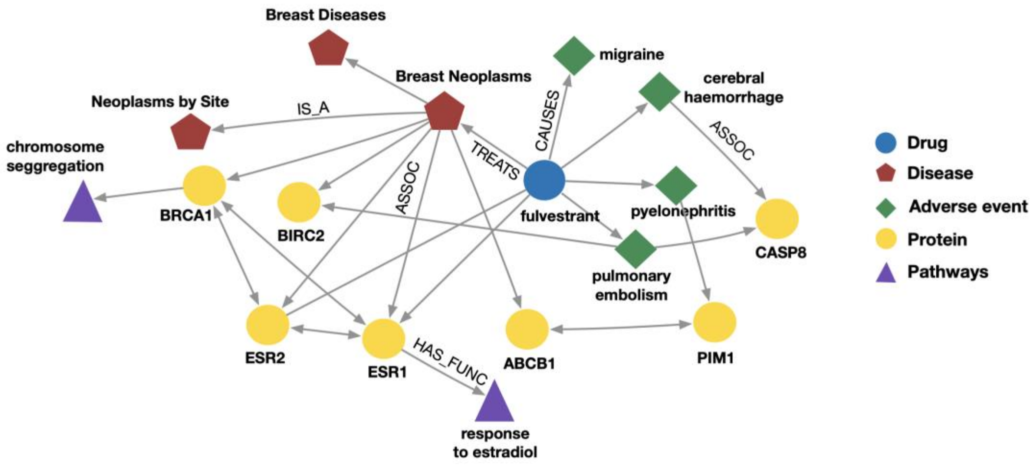
- **A heterogeneous graph is defined as**

$$\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \tau, \phi)$$

  - Nodes **with node types** $v \in V$
    - **Node type** for node v: $\tau(v)$
  - Edges **with edge types** $(u, v) \in E$
    - **Edge type** for edge (u, v): $\phi(u, v)$
    - **Relation type** for edge e is a tuple: $r(u, v) = (\tau(u), \phi(u, v), \tau(v))$

An edge can be described as a pair of nodes

Many Graphs are Heterogeneous Graphs



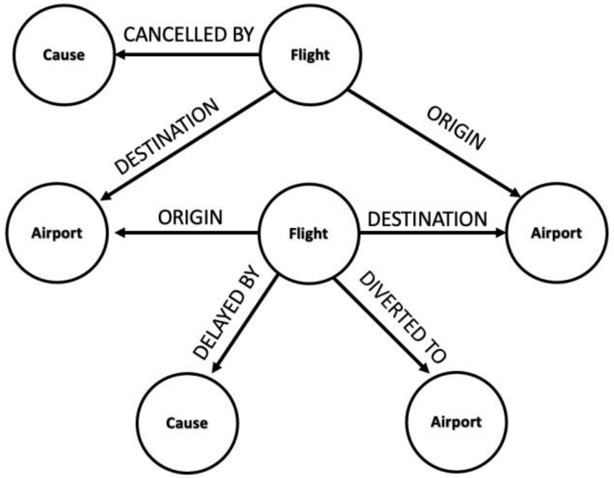
Biomedical Knowledge Graphs

Example node: Migraine

Example relation: (fulvestrant, Treats, Breast Neoplasms)

Example node type: Protein

Example edge type: Causes



Event Graphs

Example node: SFO

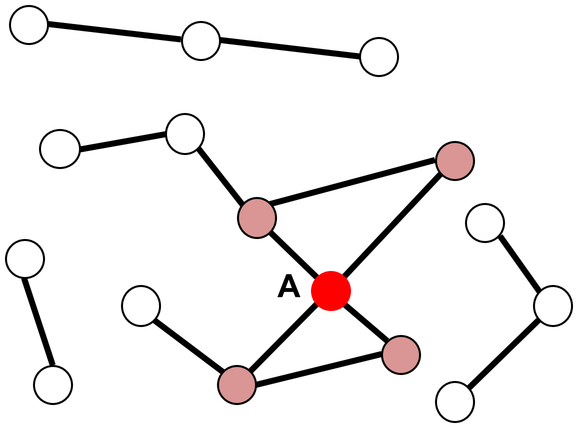
Example relation: (UA689, Origin, LAX)

Example node type: Flight

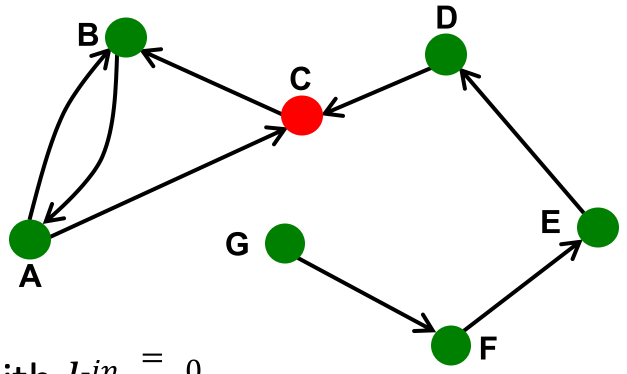
Example edge type: Destination

Node Degrees

- **Undirected**
  - **Node degree, k_i:** the number of edges adjacent to node $\boldsymbol{i}$, $\boldsymbol{k_A = 4}$
  - ***Avg. degree:*** $\boldsymbol{\bar{k} = \langle k \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i = \frac{2E}{N}}$



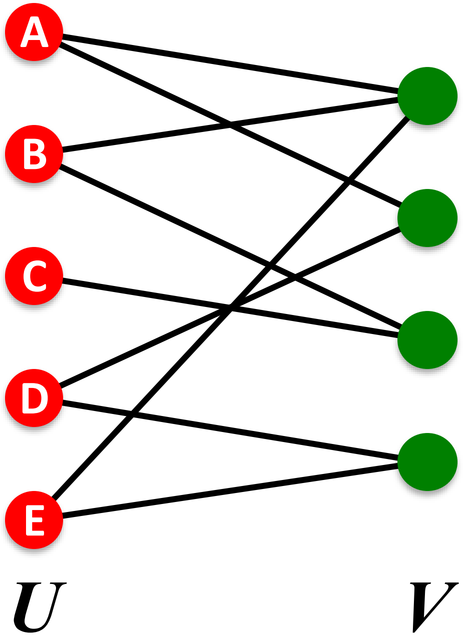
- **Directed**
  - In directed networks we define an **in-degree** and **out-degree**.
    The (total) degree of a node is the sum of in- and out-degrees.
  - $k_C^{in} = 2 ,k_C^{out} = 1 ,k_C = 3$
    $\bar{k} = \frac{E}{N} ,\overline{k^{in}} = \overline{k^{out}}$



Source: Node with $k^{in} = 0$
Sink: Node with $k^{out} = 0$

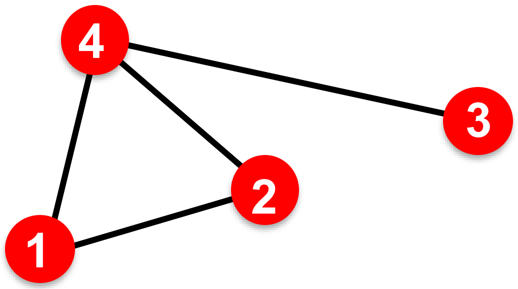
Bipartite Graph

- **Bipartite graph** is a graph whose nodes can be divided into two disjoint sets $\boldsymbol{U}$ and $\boldsymbol{V}$ such that every link connects a node in $\boldsymbol{U}$ to one in $\boldsymbol{V}$; that is, $\boldsymbol{U}$ and $\boldsymbol{V}$ 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

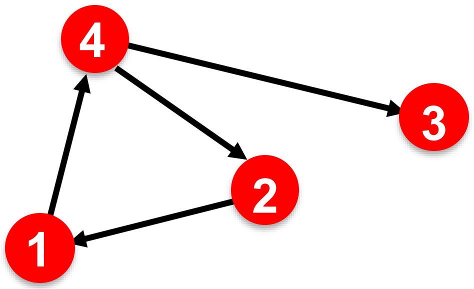


Representing Graphs: Adjacency Matrix

- $\boldsymbol{A_{ij} = 1}$ if there is a link from node $\boldsymbol{i}$ to node $\boldsymbol{j}$
- $\boldsymbol{A_{ij} = 0}$ otherwise



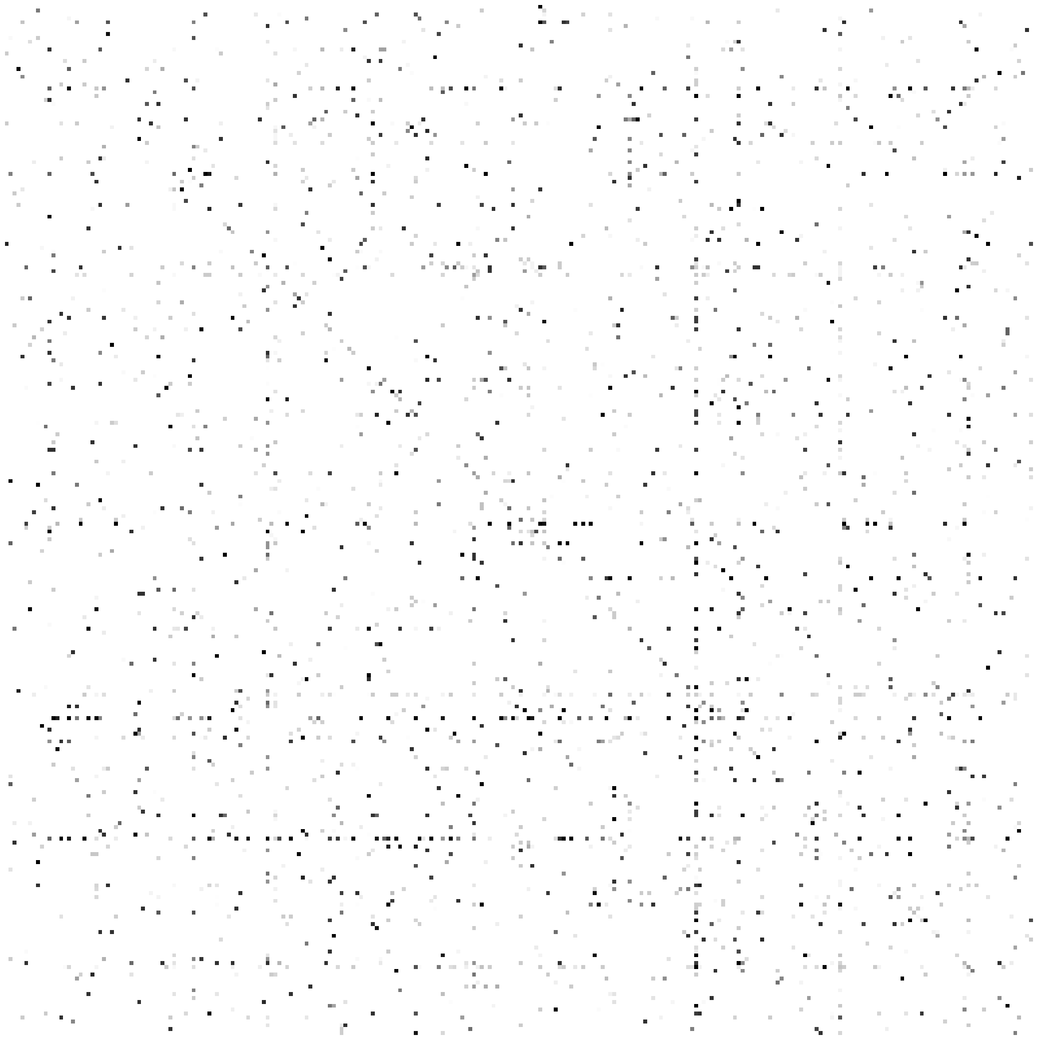
$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



$$A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

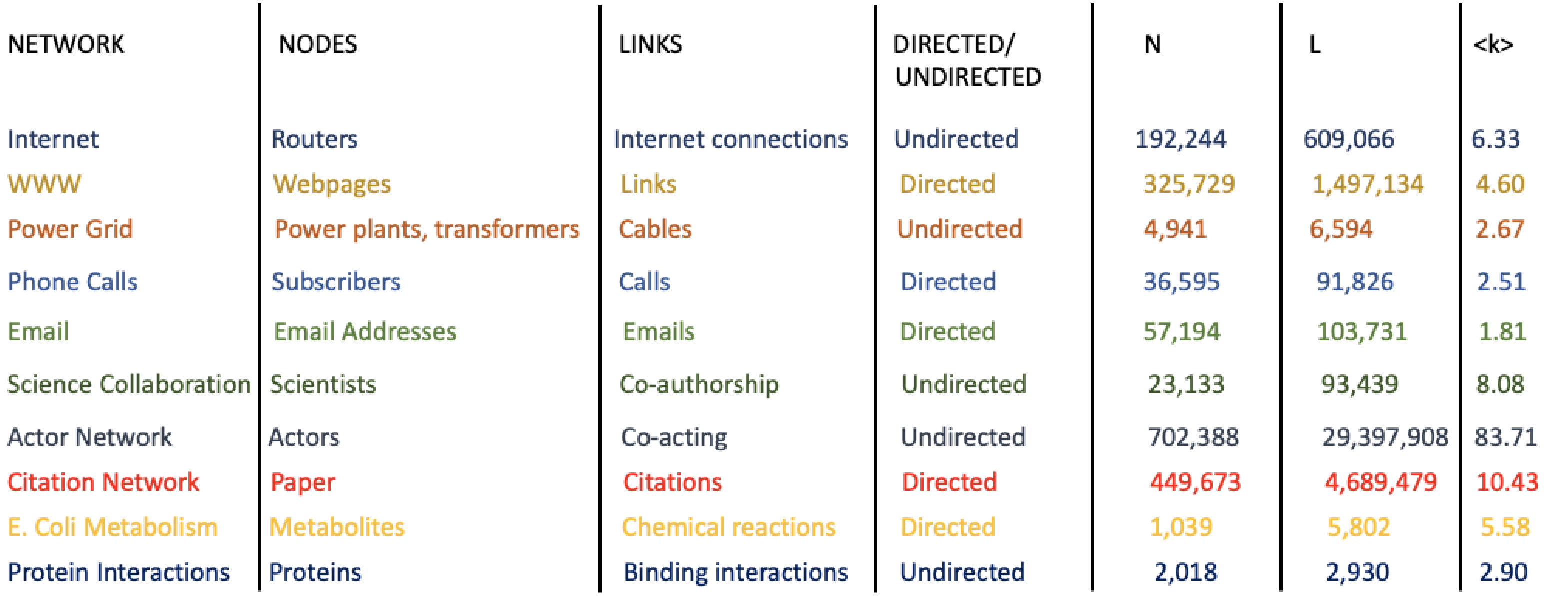
Note that for a directed graph (right) the matrix is not symmetric.

Adjacency Matrices are Sparse



Networks are Sparse Graphs

- **Most real-world networks are sparse**
- **$E \ll E_{max}$ (or $k \ll N-1$)**



NETWORK	NODES	LINKS	DIRECTED/ UNDIRECTED	N	L	<k>
Internet	Routers	Internet connections	Undirected	192,244	609,066	6.33
WWW	Webpages	Links	Directed	325,729	1,497,134	4.60
Power Grid	Power plants, transformers	Cables	Undirected	4,941	6,594	2.67
Phone Calls	Subscribers	Calls	Directed	36,595	91,826	2.51
Email	Email Addresses	Emails	Directed	57,194	103,731	1.81
Science Collaboration	Scientists	Co-authorship	Undirected	23,133	93,439	8.08
Actor Network	Actors	Co-acting	Undirected	702,388	29,397,908	83.71
Citation Network	Paper	Citations	Directed	449,673	4,689,479	10.43
E. Coli Metabolism	Metabolites	Chemical reactions	Directed	1,039	5,802	5.58
Protein Interactions	Proteins	Binding interactions	Undirected	2,018	2,930	2.90

- **Consequence: Adjacency matrix is filled with zeros**!
- (**Density of the matrix (E/N^2):** WWW=1.51×10^{-5}, MSN IM = 2.27×10^{-8})

Node and Edge Attributes

Possible options:

- Weight (*e.g.*, frequency of communication)
- Ranking (best friend, second best friend…)
- Type (friend, relative, co-worker)
- Sign: Friend vs. Foe, Trust vs. Distrust
- Properties depending on the structure of the rest of the graph: Number of common friends

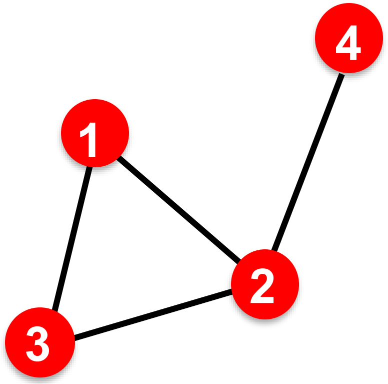
More Types of Graphs

- Unweighted (undirected)

$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2} \sum_{i,j=1}^{N} A_{ij} \qquad \bar{k} = \frac{2E}{N}$$

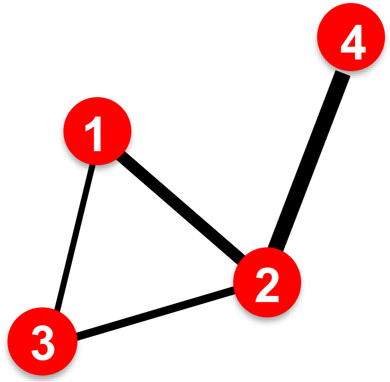


- Weighted (undirected)

$$A_{ij} = \begin{pmatrix} 0 & 2 & 0.5 & 0 \\ 2 & 0 & 1 & 4 \\ 0.5 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2} \sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$



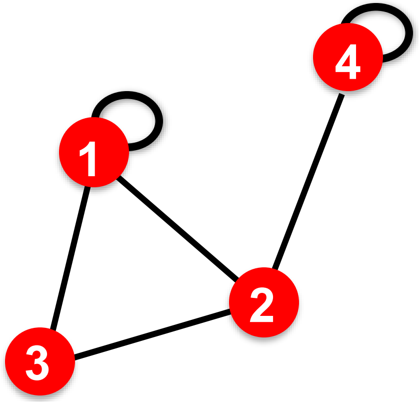
More Types of Graphs

- Self-edges (self-loops) (undirected)
  - Examples: Proteins, Hyperlinks

$$A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$A_{ii} \neq 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2} \sum_{i,j=1, i \neq j}^{N} A_{ij} + \sum_{i=1}^{N} A_{ii}$$

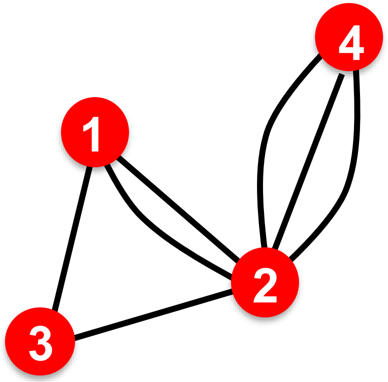


- Multigraph (undirected)
  - Examples: Communication, Collaboration

$$A_{ij} = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix}$$

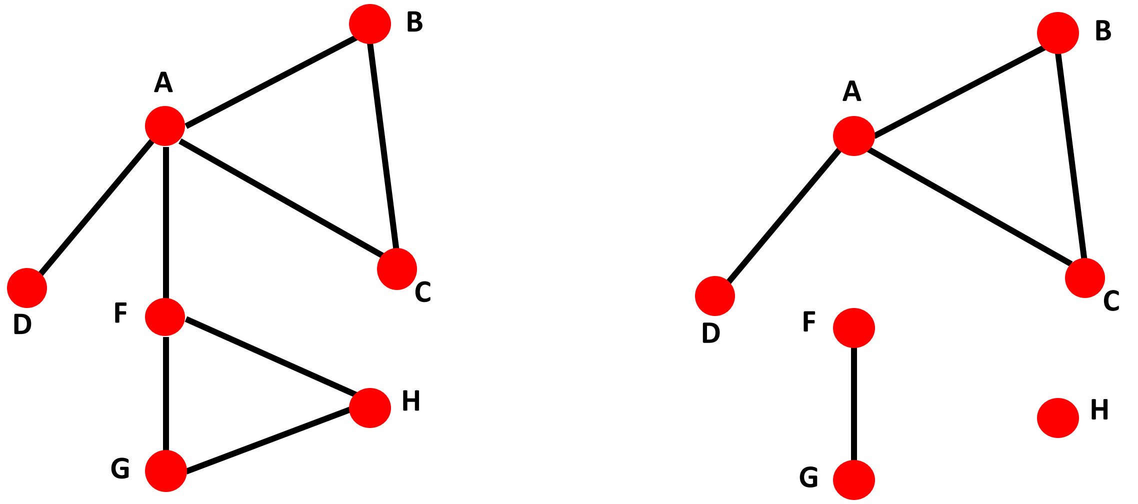
$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2} \sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$



Connectivity of Undirected Graphs

- Connected (undirected) graph:
  - Any two vertices can be joined by a path
- A disconnected graph is made up by two or more connected components

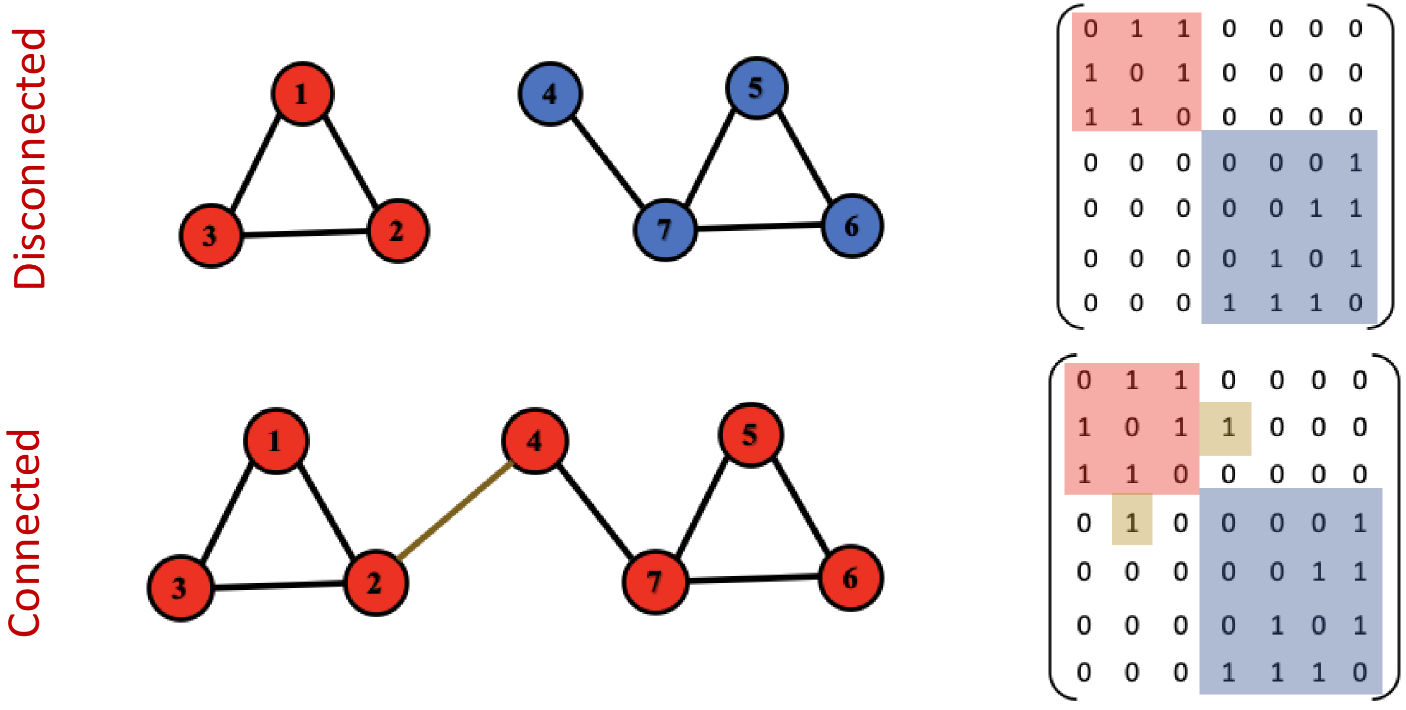


Largest Component:
Giant Component

Isolated node (node H)

Connectivity: Example

- The adjacency matrix of a network with several components can be written in a block- diagonal form, so that nonzero elements are confined to squares, with all other elements being zero:

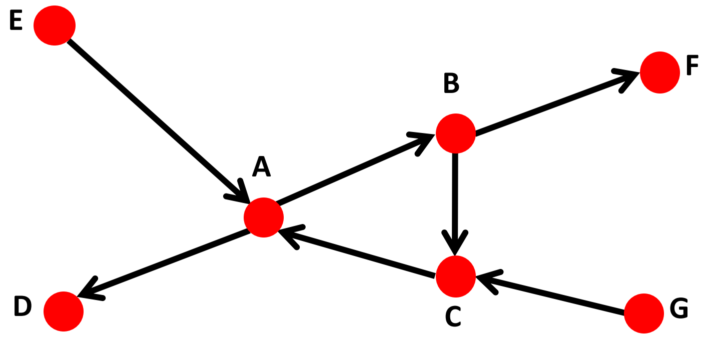


$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Connectivity of Directed Graphs

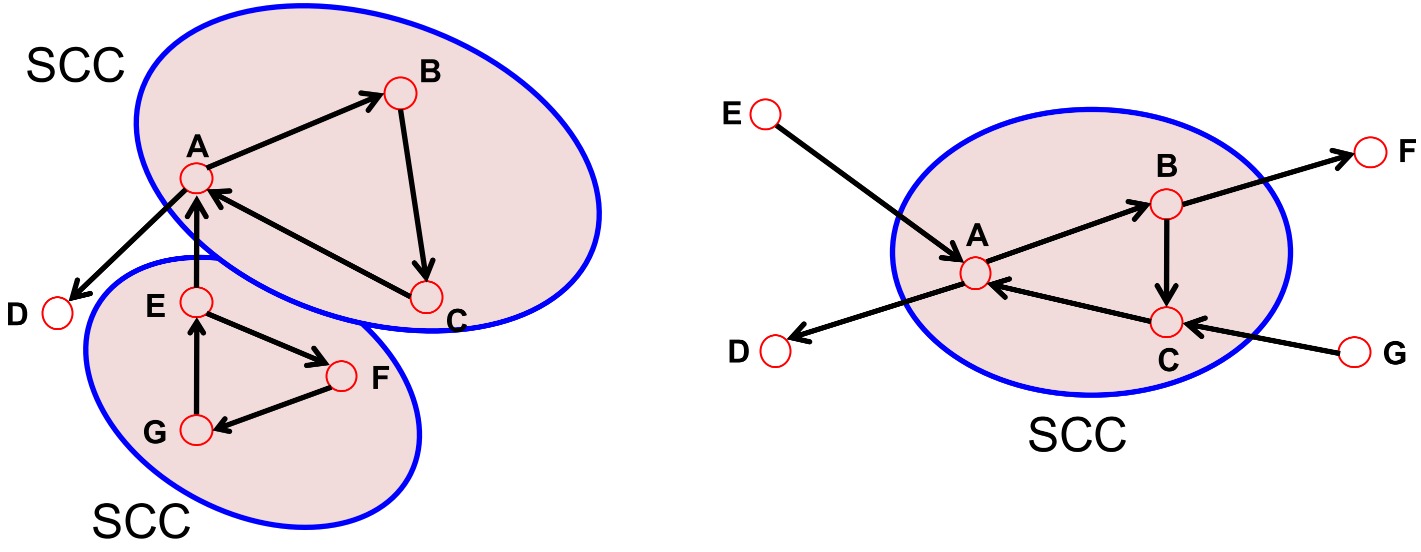
- **Strongly connected directed graph**
  - has a path from each node to every other node and vice versa (e.g., A-B path and B-A path)
- **Weakly connected directed graph**
  - is connected if we disregard the edge directions



Graph on the left is connected but not strongly connected (e.g., there is no way to get from F to G by following the edge directions).

Connectivity of Directed Graphs

- Strongly connected components (SCCs) can be identified, but not every node is part of a nontrivial strongly connected component.



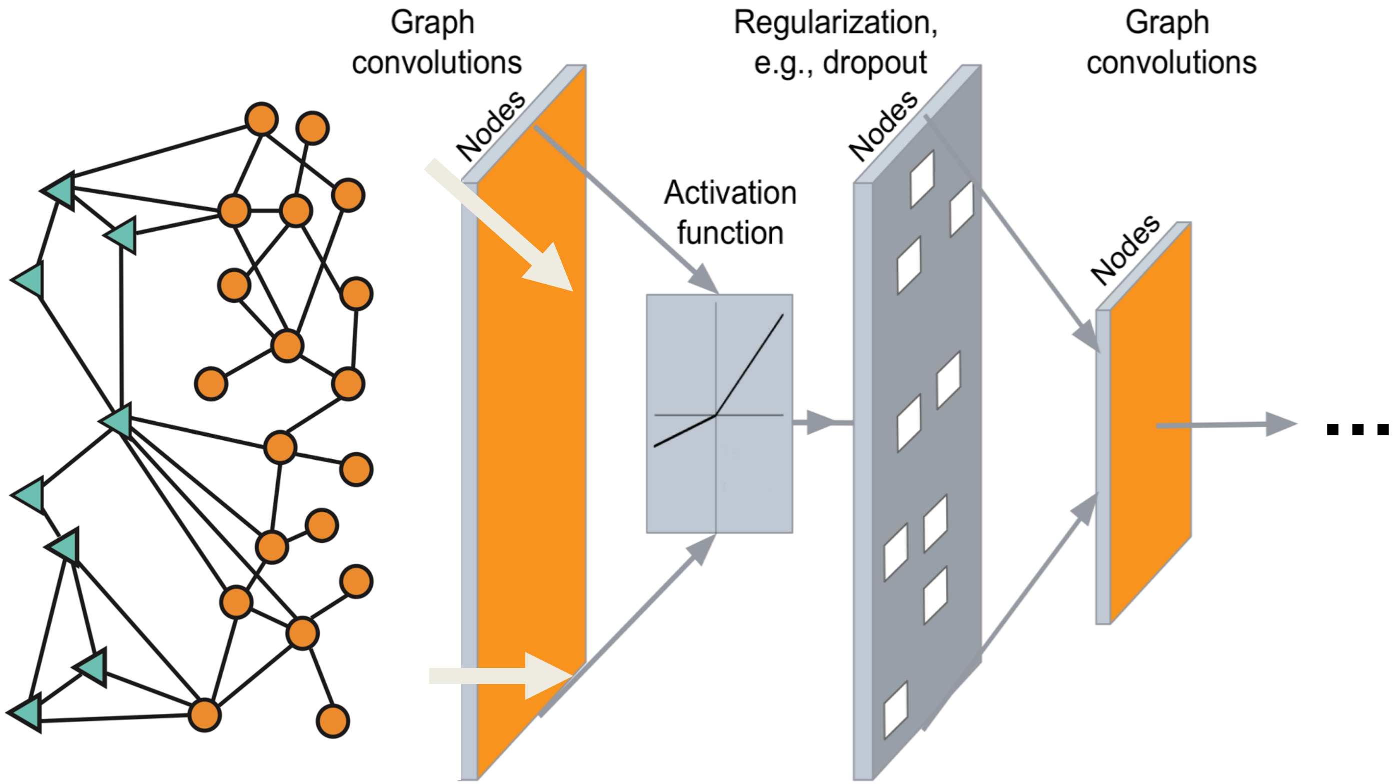
In-component: nodes that can reach the SCC,
Out-component: nodes that can be reached from the SCC.

Graph Learning Basics

Graph Learning Prediction Tasks

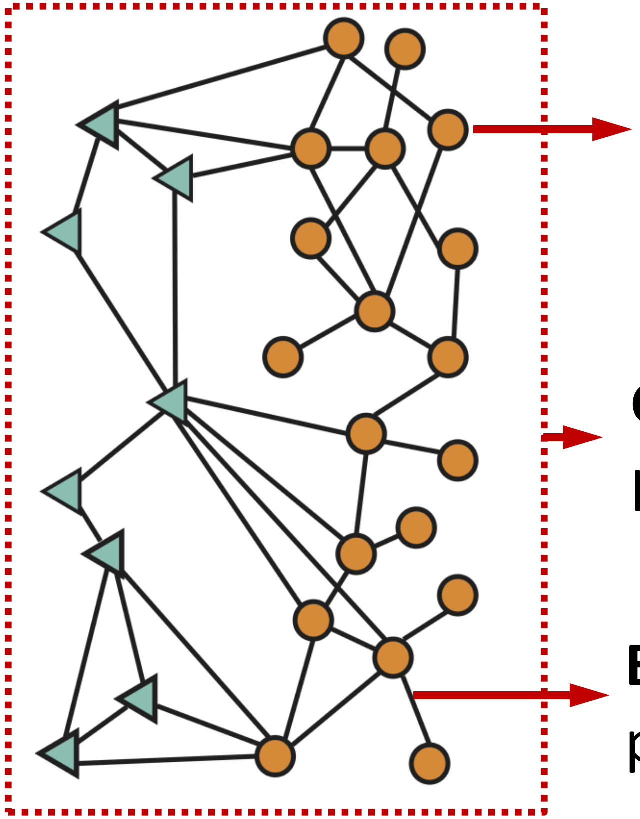
Recap: Deep Learning with Graphs

Input:
Graph



Node/
Edge/
Graph-level
Predictions

Graph ML **Tasks**

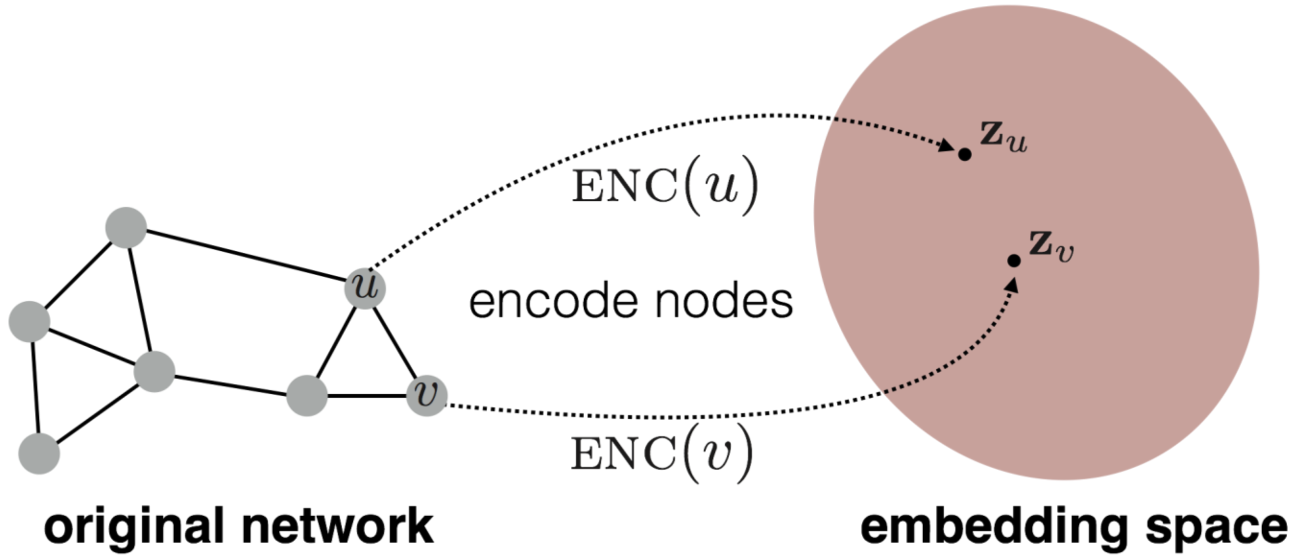


Node-level prediction

Graph-level prediction

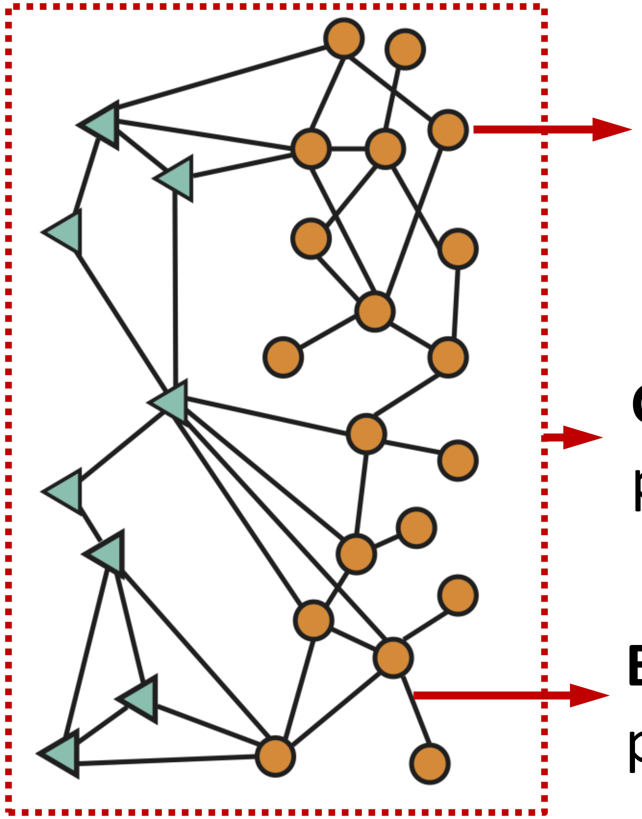
Edge-level prediction

Key Idea: **Node Embeddings**



Intuition: Map nodes to d-dimensional embeddings such that similar nodes in the graph are embedded close together

Graph ML **Tasks**

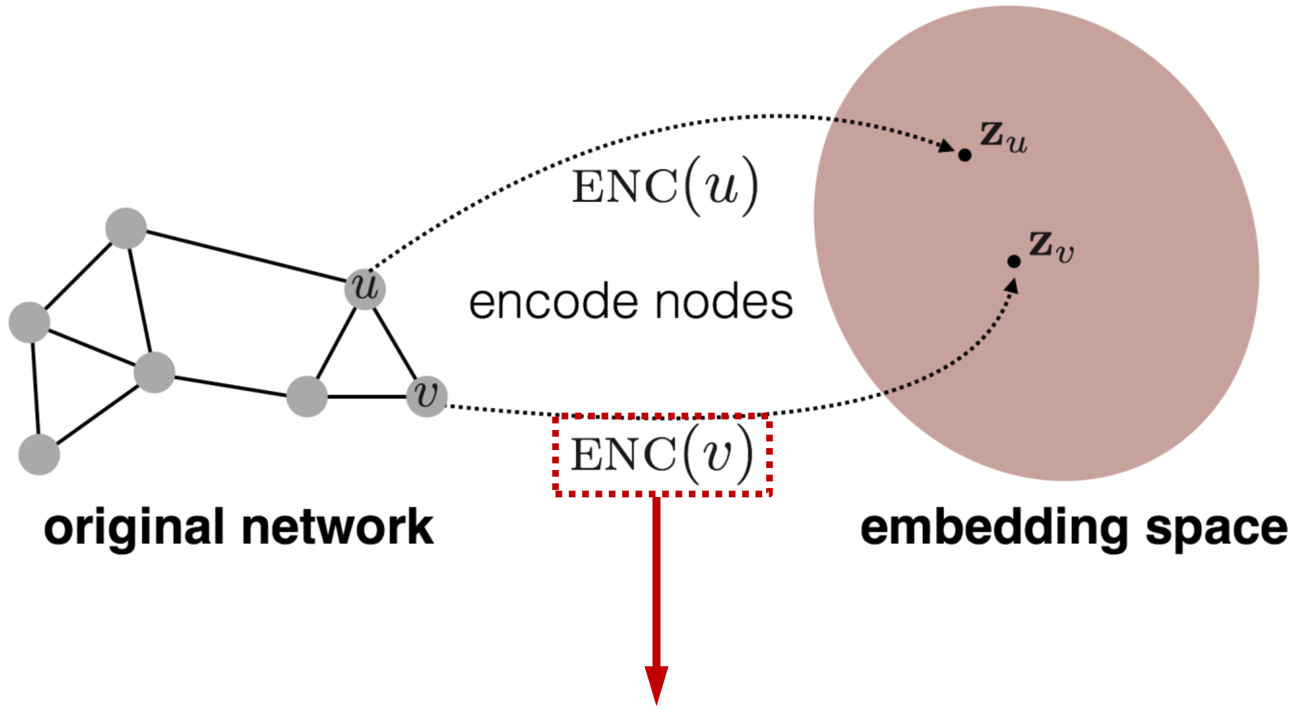


Node-level prediction

Graph-level prediction

Edge-level prediction

Key Idea: **Node Embeddings**

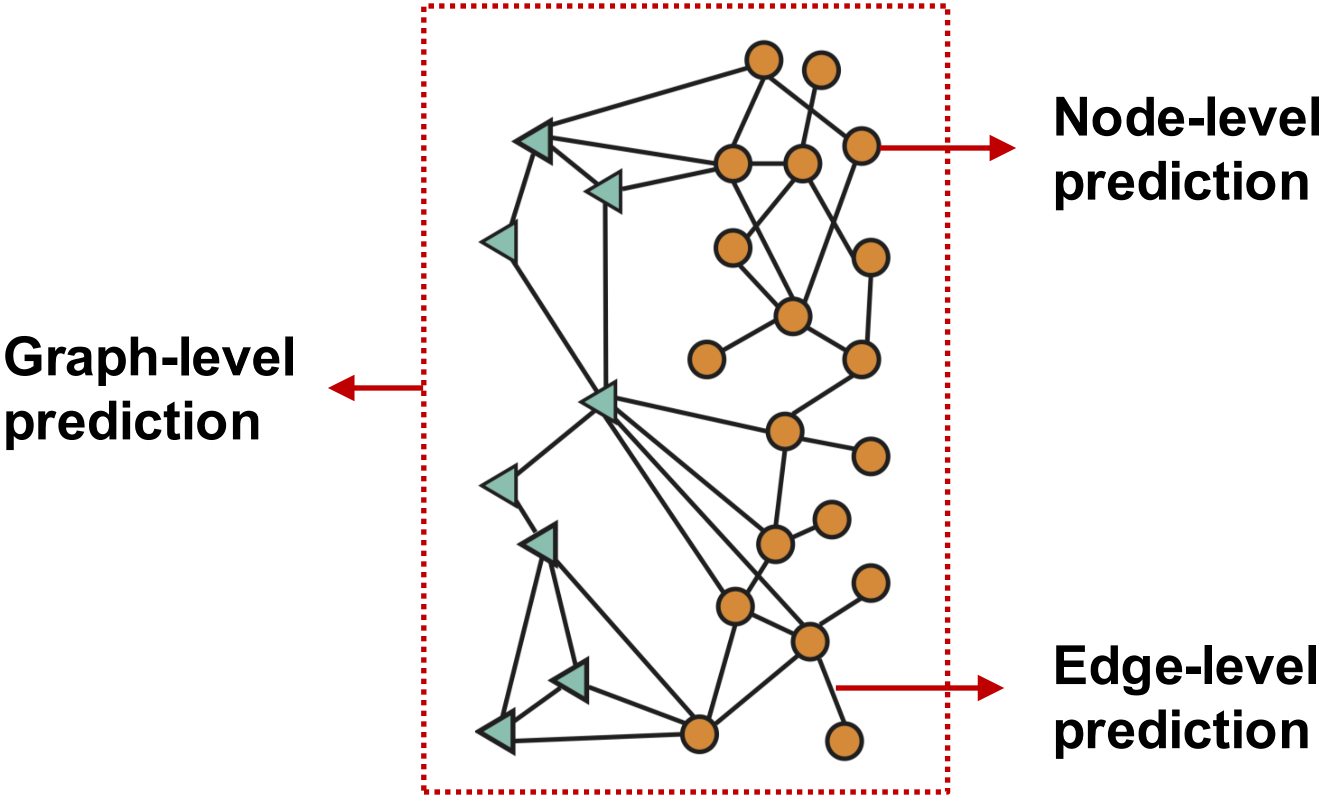


Embedding Matrix,

Graph Neural Networks, … stay tuned

Graph Learning Prediction Heads

- **Idea:** Different task levels require different prediction heads
- **Prediction head:** map node embeddings to the predictions of interest

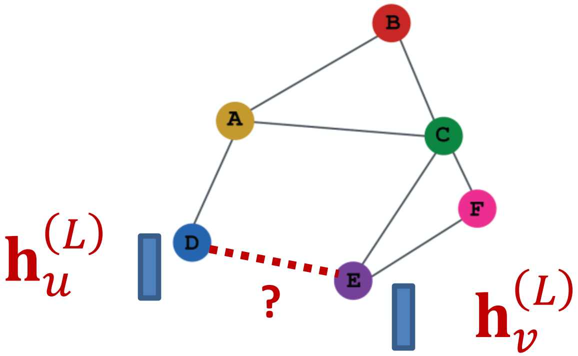


Prediction Heads: Node-level

- **Node-level prediction**: We can directly make prediction using node embeddings!
- Assuming we have d-dim node embeddings: $\{\mathbf{h}_v \in \mathbb{R}^d, \forall v \in G\}$
- Suppose we want to make k-way prediction
  - Classification: classify among k categories
  - Regression: regress on k targets
- $\hat{\boldsymbol{y}}_{\boldsymbol{v}} = \mathrm{Head}_{\mathrm{node}}(\mathbf{h}_v) = \mathbf{W} \quad \mathbf{h}_v$
  - $\mathbf{W} \quad \in \mathbb{R}^{k*d}$: We map node embeddings from $\mathbf{h}_v \in \mathbb{R}^d$ to $\hat{\boldsymbol{y}}_v \in \mathbb{R}^k$ so that we can compute the loss

Prediction Heads: Edge-level

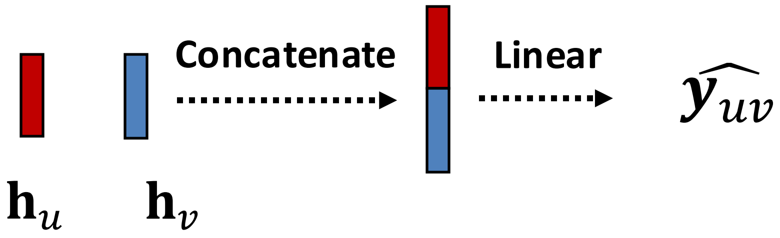
- **Edge-level prediction**: Make prediction using pairs of node embeddings
- Suppose we want to make k-way prediction
- $\hat{\boldsymbol{y}}_{\boldsymbol{uv}} = \mathrm{Head}_{\mathrm{edge}}(\mathbf{h}_u, \mathbf{h}_v)$



- What are the options for $\mathrm{Head}_{\mathrm{edge}}(\mathbf{h}_u, \mathbf{h}_v)$?

Prediction Heads: Edge-level

- **Options for** $\text{Head}_{\text{edge}}(\mathbf{h}_u , \mathbf{h}_v)$:
- **(1) Concatenation + Linear**



  - $\widehat{\boldsymbol{y}}_{\boldsymbol{uv}} = \mathbf{W} \quad \text{Concat}(\mathbf{h}_u , \mathbf{h}_v)$
  - Here $\mathbf{W} \quad \in \ \mathbb{R}^{k*2d}$ will map $2d$-dimensional embeddings (since we concatenated embeddings) to k-dim embeddings (k-way prediction)
  - $\mathbf{W}$ can be replace with deeper neural networks, e.g., MLP

Prediction Heads: Edge-level

- **Options for** $\text{Head}_{\text{edge}}(\mathbf{h}_u, \mathbf{h}_v)$:
- **(2) Dot product**
  - $\widehat{\boldsymbol{y}}_{\boldsymbol{uv}} = (\mathbf{h}_u)^T \mathbf{h}_v$
  - **This approach only applies to 1-way prediction** (e.g., link prediction: predict the existence of an edge)
  - **Applying to $\boldsymbol{k}$-way prediction:**
    - Similar to **multi-head attention**: $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(k)}$ trainable

$$\widehat{\boldsymbol{y}}_{\boldsymbol{uv}}^{(1)} = (\mathbf{h}_u)^T \mathbf{W}^{(1)} \mathbf{h}_v$$

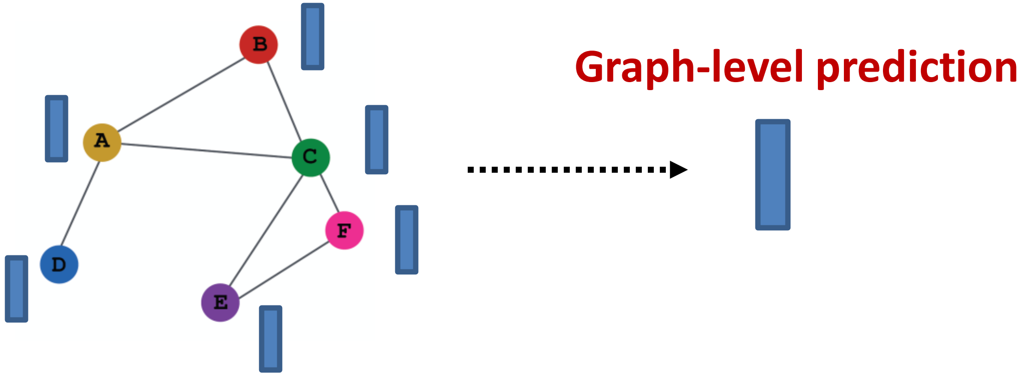
$$\ldots$$

$$\widehat{\boldsymbol{y}}_{\boldsymbol{uv}}^{(k)} = (\mathbf{h}_u)^T \mathbf{W}^{(k)} \mathbf{h}_v$$

$$\widehat{\boldsymbol{y}}_{uv} = \text{Concat}(\widehat{\boldsymbol{y}}_{\boldsymbol{uv}}^{(1)}, \ldots, \widehat{\boldsymbol{y}}_{\boldsymbol{uv}}^{(k)}) \in \mathbb{R}^k$$

Prediction Heads: Graph-level

- **Graph-level prediction**: Make prediction using all the node embeddings in our graph
- Suppose we want to make k-way prediction
- $\widehat{\boldsymbol{y}}_G = \text{Head}_{\text{graph}}(\{\mathbf{h}_v \in \mathbb{R}^d, \forall v \in G\})$



K. Xu*, W. Hu*, J. Leskovec, S. Jegelka. How Powerful Are Graph Neural Networks, ICLR 2019

Prediction Heads: Graph-level

- Options for $\text{Head}_{\text{graph}}(\{\mathbf{h}_v \in \mathbb{R}^d, \forall v \in G\})$
- **(1) Global mean pooling**

$$\hat{\boldsymbol{y}}_G = \text{Mean}(\{\mathbf{h}_v \in \mathbb{R}^d, \forall v \in G\})$$

- **(2) Global max pooling**

$$\hat{\boldsymbol{y}}_G = \text{Max}(\{\mathbf{h}_v \in \mathbb{R}^d, \forall v \in G\})$$

- **(3) Global sum pooling**

$$\hat{\boldsymbol{y}}_G = \text{Sum}(\{\mathbf{h}_v \in \mathbb{R}^d, \forall v \in G\})$$

- **(4) Global attention pooling**

$$\hat{\boldsymbol{y}}_G = \text{Sum}\left(\{\alpha_v \mathbf{h}_v \in \mathbb{R}^d, \forall v \in G\}\right), \alpha_v = \frac{\exp(W h_v + b)}{\sum_{u \in \mathrm{G}} \exp(W h_u + b)}$$

Reading papers

Suggestions for Research

Sources of AI/ML papers

Recent major AI/ML conferences:

- NeurIPS 2024:

https://openreview.net/group?id=NeurIPS.cc/2024/Conference#tab-accept-oral

- ICML 2025:

https://openreview.net/group?id=ICML.cc/2025/Conference#tab-accept-oral

- ICLR 2025:

https://openreview.net/group?id=ICLR.cc/2025/Conference#tab-accept-oral

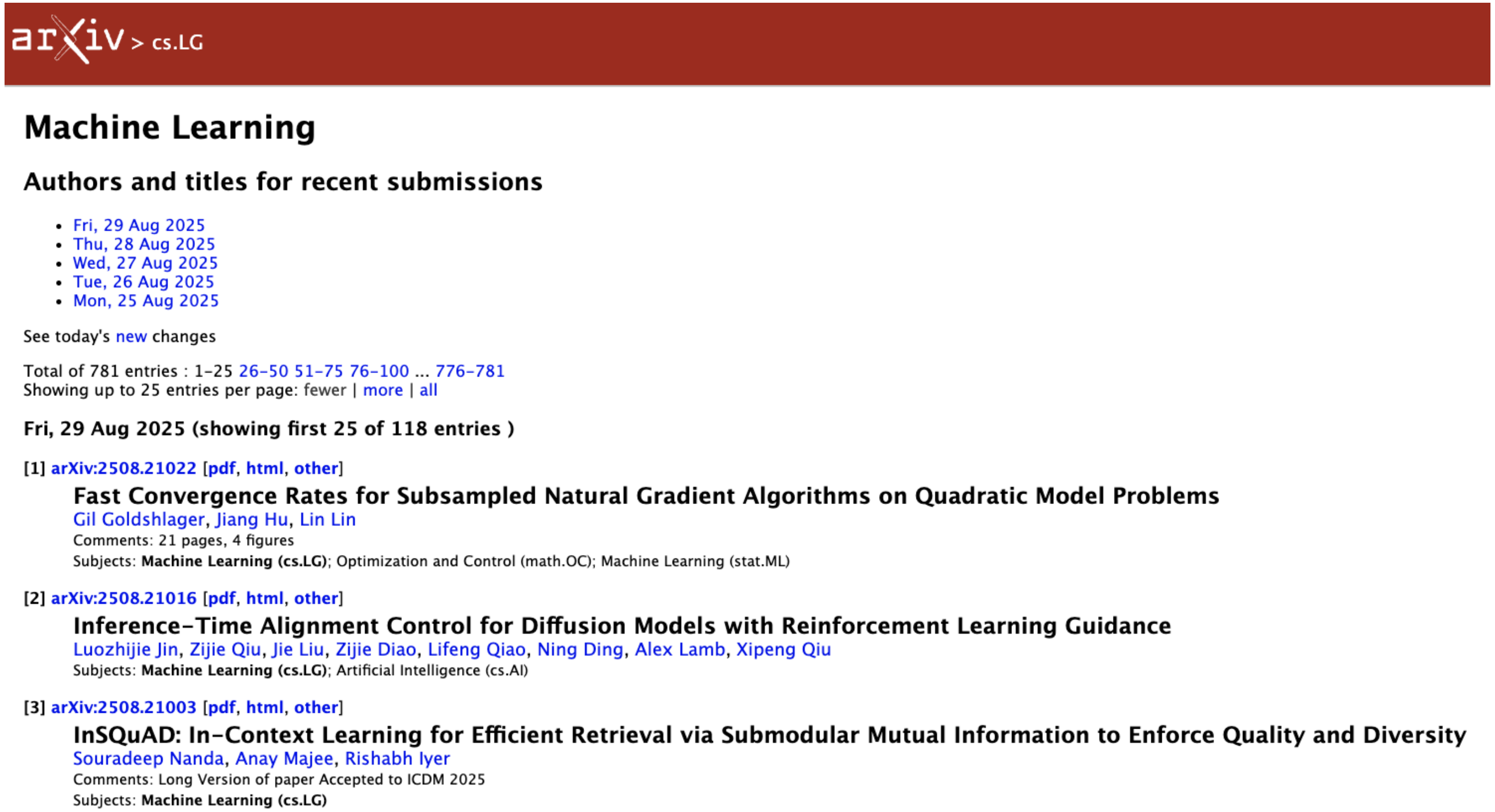
- LOG 2024 (Learning on graphs):

https://openreview.net/group?id=logconference.io/LOG/2024/Conference#tab-accept-oral

Sources of AI/ML papers

Latest Arxiv papers:

https://arxiv.org/list/cs.LG/pastweek?skip=0&show=25



Machine Learning

Authors and titles for recent submissions

- Fri, 29 Aug 2025
- Thu, 28 Aug 2025
- Wed, 27 Aug 2025
- Tue, 26 Aug 2025
- Mon, 25 Aug 2025

See today's new changes

Total of 781 entries : 1–25 26–50 51–75 76–100 ... 776–781
Showing up to 25 entries per page: fewer | more | all

Fri, 29 Aug 2025 (showing first 25 of 118 entries)

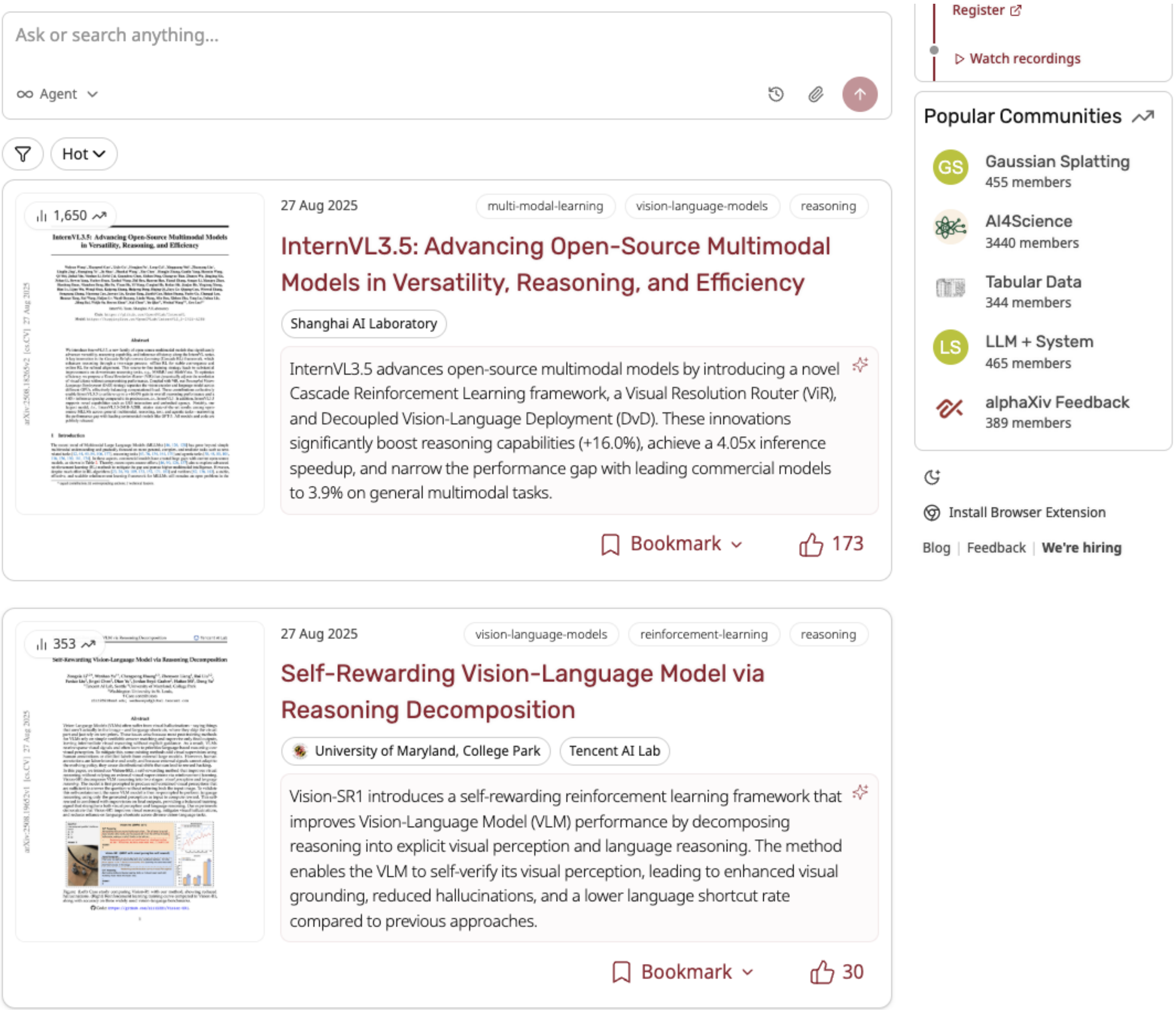
[1] **arXiv:2508.21022** [**pdf**, **html**, **other**]
Fast Convergence Rates for Subsampled Natural Gradient Algorithms on Quadratic Model Problems
Gil Goldshlager, Jiang Hu, Lin Lin
Comments: 21 pages, 4 figures
Subjects: **Machine Learning (cs.LG)**; Optimization and Control (math.OC); Machine Learning (stat.ML)

[2] **arXiv:2508.21016** [**pdf**, **html**, **other**]
Inference–Time Alignment Control for Diffusion Models with Reinforcement Learning Guidance
Luozhijie Jin, Zijie Qiu, Jie Liu, Zijie Diao, Lifeng Qiao, Ning Ding, Alex Lamb, Xipeng Qiu
Subjects: **Machine Learning (cs.LG)**; Artificial Intelligence (cs.AI)

[3] **arXiv:2508.21003** [**pdf**, **html**, **other**]
InSQuAD: In–Context Learning for Efficient Retrieval via Submodular Mutual Information to Enforce Quality and Diversity
Souradeep Nanda, Anay Majee, Rishabh Iyer
Comments: Long Version of paper Accepted to ICDM 2025
Subjects: **Machine Learning (cs.LG)**

LLM powered Paper Reading

- https://www.alphaxiv.org/



27 Aug 2025 · multi-modal-learning · vision-language-models · reasoning

InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency

Shanghai AI Laboratory

InternVL3.5 advances open-source multimodal models by introducing a novel Cascade Reinforcement Learning framework, a Visual Resolution Router (ViR), and Decoupled Vision-Language Deployment (DvD). These innovations significantly boost reasoning capabilities (+16.0%), achieve a 4.05x inference speedup, and narrow the performance gap with leading commercial models to 3.9% on general multimodal tasks.

27 Aug 2025 · vision-language-models · reinforcement-learning · reasoning

Self-Rewarding Vision-Language Model via Reasoning Decomposition

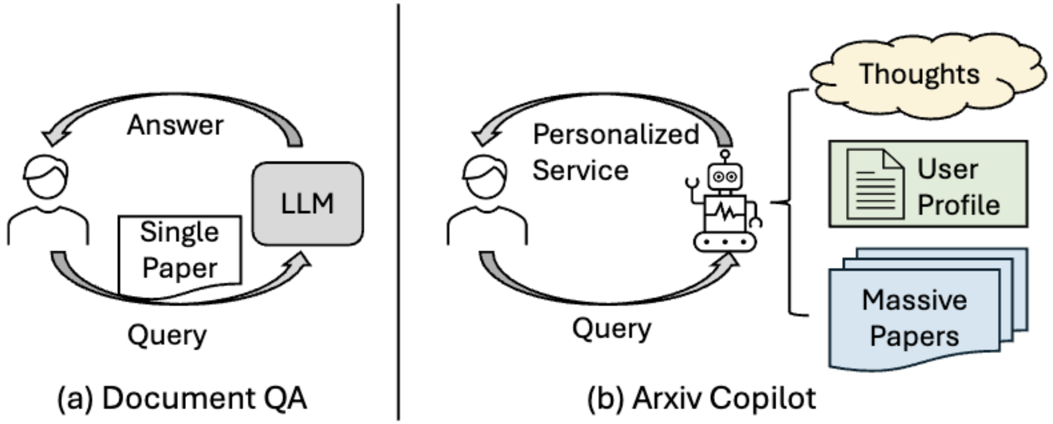
University of Maryland, College Park · Tencent AI Lab

Vision-SR1 introduces a self-rewarding reinforcement learning framework that improves Vision-Language Model (VLM) performance by decomposing reasoning into explicit visual perception and language reasoning. The method enables the VLM to self-verify its visual perception, leading to enhanced visual grounding, reduced hallucinations, and a lower language shortcut rate compared to previous approaches.

Arxiv Copilot

Arxiv Copilot: A Self-Evolving and Efficient LLM System for Personalized Academic Assistance

Guanyu Lin[1 2*], Tao Feng[1*], Pengrui Han[1 3*], Ge Liu[1], Jiaxuan You[1]
[1]University of Illinois at Urbana-Champaign, [2]Carnegie Mellon University, [3]Carleton College
[*]Equal Contribution



(a) Document QA (b) Arxiv Copilot

Figure 1: **Comparison of (a) document Question Answering (QA) with our (b) Arxiv Copilot.** Conven-

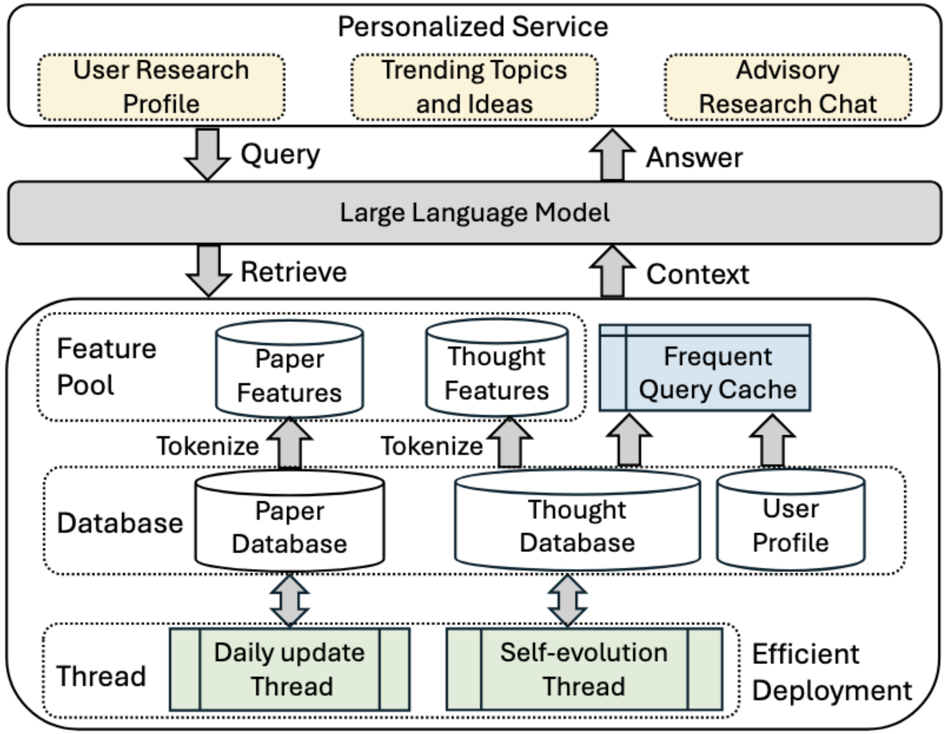
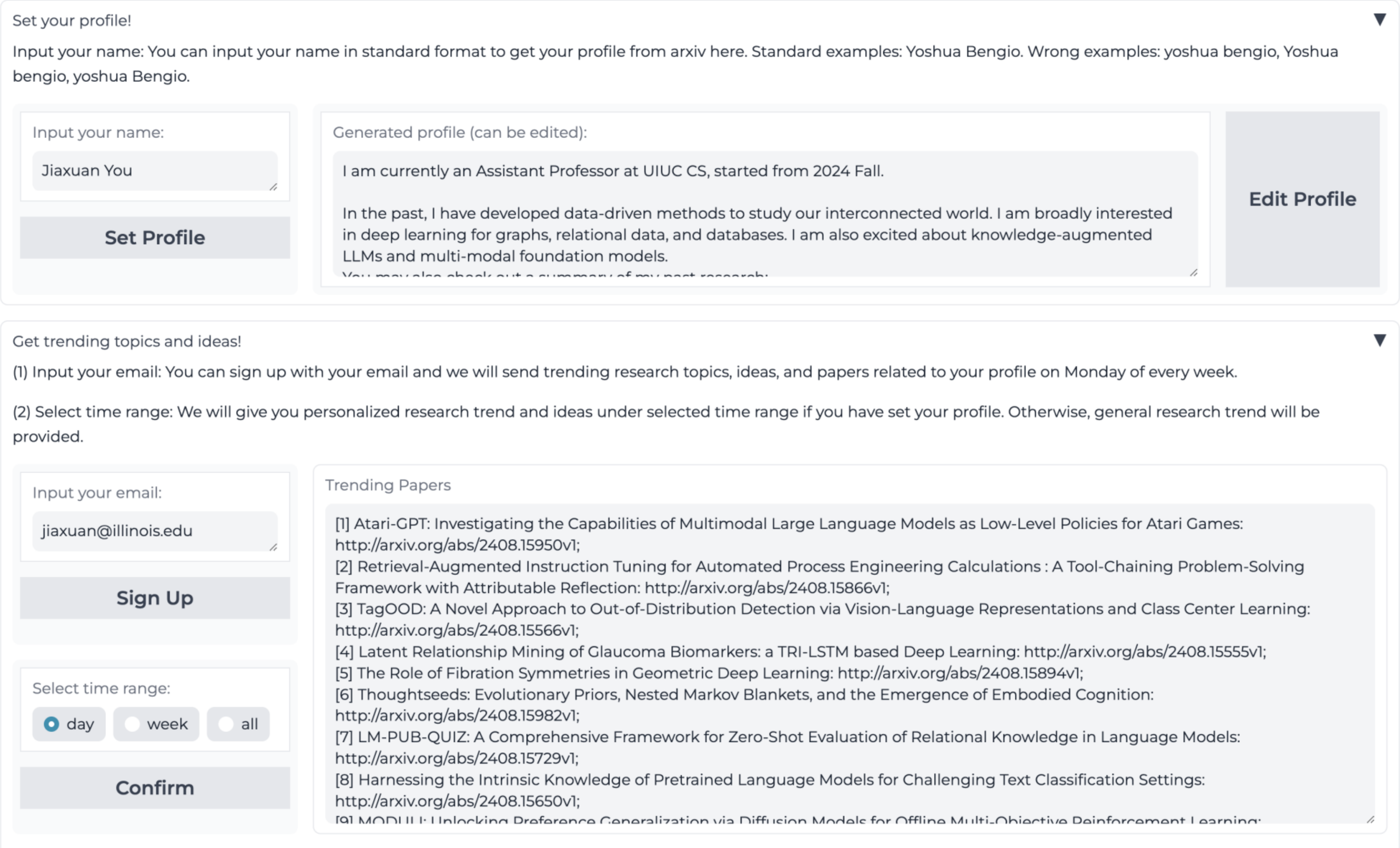


Figure 2: **Architecture of Arxiv Copilot from bottom-to-up perspective.** (a) In personalized service, Arxiv

Arxiv Copilot

Arxiv Copilot summarize latest Arxiv papers based on your profile



Set your profile!

Input your name: You can input your name in standard format to get your profile from arxiv here. Standard examples: Yoshua Bengio. Wrong examples: yoshua bengio, Yoshua bengio, yoshua Bengio.

Input your name: Jiaxuan You

Generated profile (can be edited):

I am currently an Assistant Professor at UIUC CS, started from 2024 Fall.

In the past, I have developed data-driven methods to study our interconnected world. I am broadly interested in deep learning for graphs, relational data, and databases. I am also excited about knowledge-augmented LLMs and multi-modal foundation models.

You may also check out a summary of my past research:

Get trending topics and ideas!

(1) Input your email: You can sign up with your email and we will send trending research topics, ideas, and papers related to your profile on Monday of every week.

(2) Select time range: We will give you personalized research trend and ideas under selected time range if you have set your profile. Otherwise, general research trend will be provided.

Input your email: jiaxuan@illinois.edu

Select time range: ● day ○ week ○ all

Trending Papers

[1] Atari-GPT: Investigating the Capabilities of Multimodal Large Language Models as Low-Level Policies for Atari Games: http://arxiv.org/abs/2408.15950v1;
[2] Retrieval-Augmented Instruction Tuning for Automated Process Engineering Calculations : A Tool-Chaining Problem-Solving Framework with Attributable Reflection: http://arxiv.org/abs/2408.15866v1;
[3] TagOOD: A Novel Approach to Out-of-Distribution Detection via Vision-Language Representations and Class Center Learning: http://arxiv.org/abs/2408.15566v1;
[4] Latent Relationship Mining of Glaucoma Biomarkers: a TRI-LSTM based Deep Learning: http://arxiv.org/abs/2408.15555v1;
[5] The Role of Fibration Symmetries in Geometric Deep Learning: http://arxiv.org/abs/2408.15894v1;
[6] Thoughtseeds: Evolutionary Priors, Nested Markov Blankets, and the Emergence of Embodied Cognition: http://arxiv.org/abs/2408.15982v1;
[7] LM-PUB-QUIZ: A Comprehensive Framework for Zero-Shot Evaluation of Relational Knowledge in Language Models: http://arxiv.org/abs/2408.15729v1;
[8] Harnessing the Intrinsic Knowledge of Pretrained Language Models for Challenging Text Classification Settings: http://arxiv.org/abs/2408.15650v1;
[9] MODULI: Unlocking Preference Generalization via Diffusion Models for Offline Multi-Objective Reinforcement Learning:

Arxiv Copilot is a research prototype demo. Feedback is welcomed!

Tips for Writing Papers

Highly recommend this webpage:

https://cs.stanford.edu/people/widom/paper-writing.html

Jennifer Widom

Frederick Emmons Terman Dean of the School of Engineering
Fletcher Jones Professor in Computer Science and Electrical Engineering

Stanford University

5 Research Questions: Originally as tips for writing the Introduction

- We adapt to a **checklist/summary** of a research paper

The 5 Questions for A Good Paper

Research question:

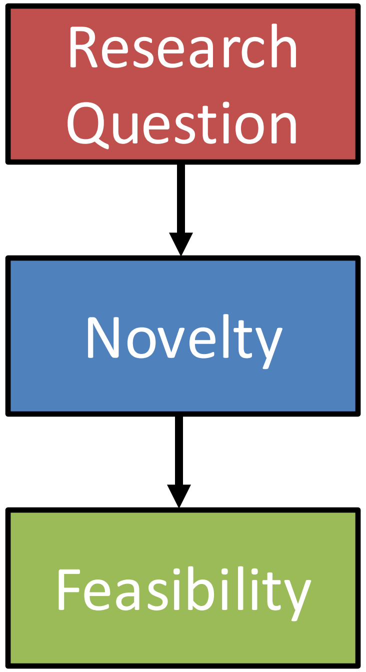
- Q1: What is the problem?
- Q2: Why is it interesting and important?

Novelty:

- Why is it hard?
- Why hasn't it been solved before?

Feasibility:

- What are the key components of my approach and results?



An Example of Research 5Q - GraphRNN

☐ GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models
J You, R Ying, X Ren, WL Hamilton, J Leskovec
Proceedings of the International Conference on Machine Learning (ICML)

1263 2018

My first paper during my PhD :p

Abstract: Modeling and generating graphs is fundamental for studying networks in biology, engineering, and social sciences. However, modeling complex distributions over graphs and then efficiently sampling from these distributions is challenging due to the non-unique, high-dimensional nature of graphs and the complex, non-local dependencies that exist between edges in a given graph. Here we propose GraphRNN, a deep autoregressive model that addresses the above challenges and approximates any distribution of graphs with minimal assumptions about their structure. GraphRNN learns to generate graphs by training on a representative set of graphs and decomposes the graph generation process into a sequence of node and edge formations, conditioned on the graph structure generated so far. In order to quantitatively evaluate the performance of GraphRNN, we introduce a benchmark suite of datasets, baselines and novel evaluation metrics based on Maximum Mean Discrepancy, which measure distances between sets of graphs. Our experiments show that GraphRNN significantly outperforms all baselines, learning to generate diverse graphs that match the structural characteristics of a target set, while also scaling to graphs 50 times larger than previous deep models.

An Example of Research 5Q - GraphRNN

What is the research problem?

- By Jiaxuan in Aug 2025:

Modeling arbitrary distributions of graphs and sampling from it

Why is it interesting and important?

- By Jiaxuan in Aug 2025:

Fundamental in studying graphs across science disciplines

An Example of Research 5Q - GraphRNN

Why is it hard? (E.g., why do naive approaches fail?)

- By Jiaxuan in Aug 2025:

Graphs are complex and diverse (challenge 1), and present non-local dependencies between edges and large scale (challenge 2)

An Example - GraphRNN

Why hasn't it been solved before? (Or, what's wrong with previous proposed solutions? How does mine differ?)

- By Jiaxuan in Aug 2025:

Classic graph generative models are hand-engineered to model a particular family of graphs (didn't stress challenge 1), while deep models are either limited to learning from a single graph or generating tiny graphs (didn't stress challenge 2).

An Example - GraphRNN

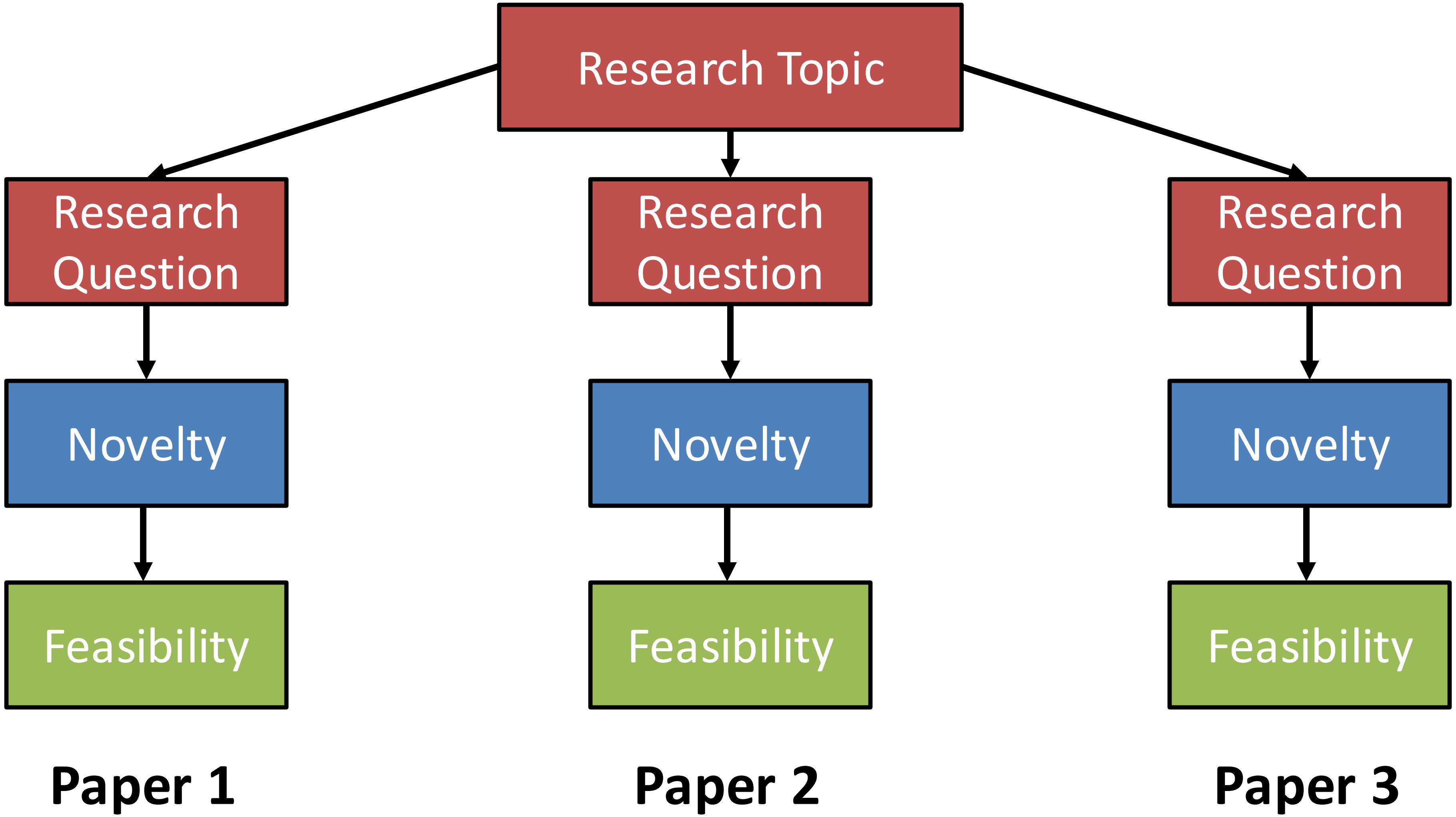
- **What are the key components of my approach and results? Also include any specific limitations.**
- By Jiaxuan in Aug 2025:

The first autoregressive graph generator, modeling graph generation as a sequence of additions of new nodes and edges (stressed both challenge 1 and 2), and a comprehensive suite of benchmark tasks and baselines for the graph generation problem (and more!).

Project Task – Paper Reading and Analysis

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	

Paper Reading and Analysis: Writing Task 1



Research Topic

Research Question

Novelty

Feasibility

Research Question

Novelty

Feasibility

Research Question

Novelty

Feasibility

Paper 1

Paper 2

Paper 3

Paper Reading and Analysis: Writing Task 1

Total 7.5% of final grade:

- **Identify a key research topic**
  - Make sure the topic is related to your interest
- **Pick 3 recent research papers related to graphs**, under the research topic
  - Suggested paper sources: NeurIPS 2024, ICML 2025, ICLR 2025, or Arxiv 2025
  - **Read 3+ papers** and pick 3. The papers will be used to inspire your projects
  - **Read the paper.** Search / Ask LLM tool for relevant concepts to help you understand the paper, if needed.
- **Summarize the 5 questions for each paper**
- **Submission DDL: Sept 17 (Wed)**
- We will announce the exact submission instructions early next week.

Summary

- Choice of a **graph representation**:
  - Directed, undirected, bipartite, dynamic, heterogeneous weighted, adjacency matrix, …
- Different types of tasks require different **prediction heads:**
  - Node level
  - Edge level
  - Graph level
- Tips for reading research papers
  - Paper sources & Arxiv Copilot
  - **The 5 research questions**