Google Research

Stanford University

Machine Learning on Graphs: A Model and Comprehensive Taxonomy

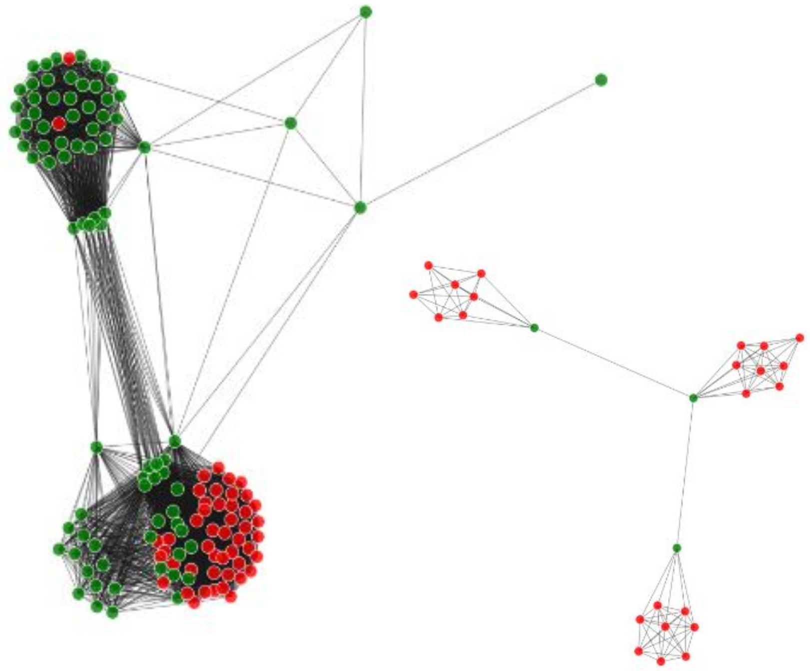
Bryan Perozzi w/ Ines Chami, Sami Abu-El-Haija, Christopher Ré and Kevin Murphy

Talk Outline

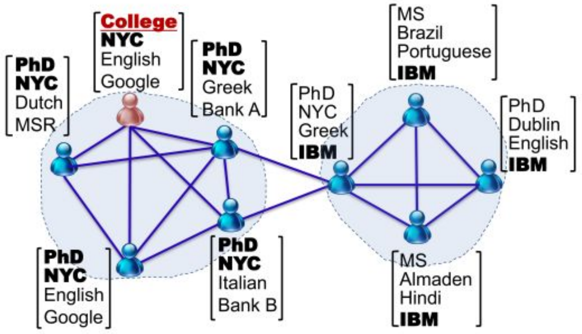
- Graph Representation Learning
- Our Model & Taxonomy
- Unsupervised Graph Embedding
- Supervised Graph Embeddings

Graph Representation Learning

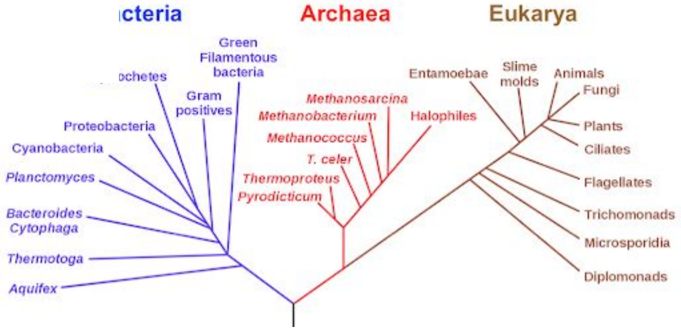
Graph-structured data



- **Graph**: $G = (V, E)$ = Nodes + Edges
- **Universal data structure** to represent complex relational data
- **Ubiquitous** in ML

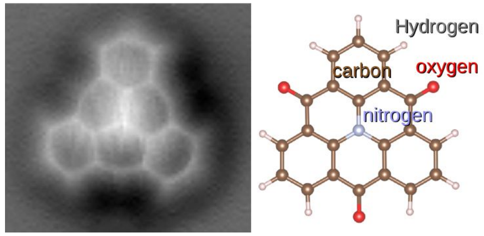


Social Networks



Phylogenetics

Figure: NASA



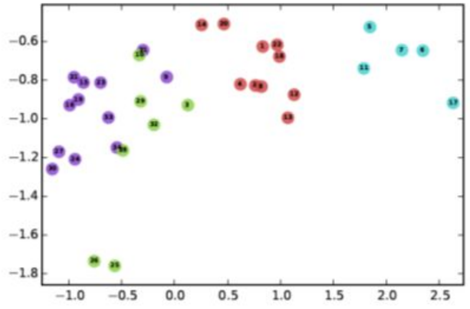
Molecules

Figure: Hapala et al - Nature Comms

Graph Representation Learning (GRL)

Goal: Map graphs to **continuous**, **low-dimensional**, **dense** vector representations that preserve the graph information

Input



Output

Why useful: These representations can be used to solve any ML task!

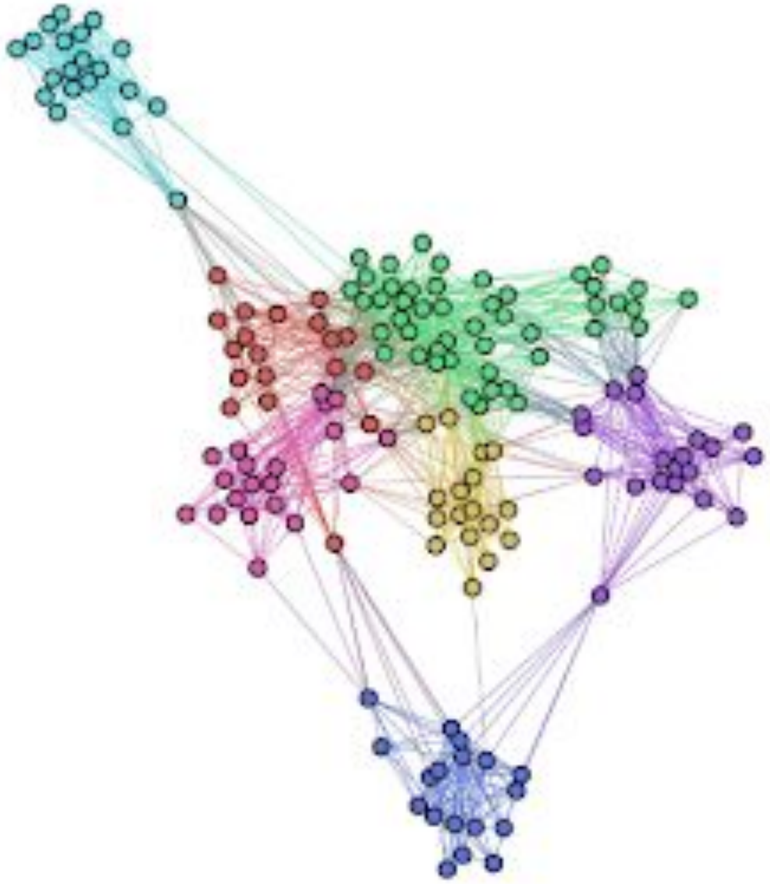
Challenge: Graphs = **Discrete**, **high-dimensional** and **sparse** representations, how do we preserve the graph structure/similarities between nodes?

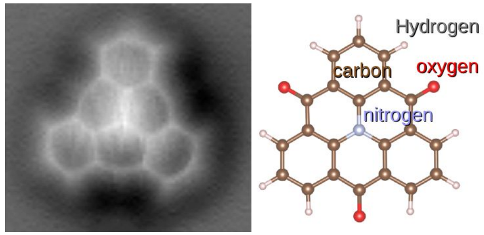
ML Applications

- **Unsupervised**
  - Link Prediction $f : V \times V \rightarrow \{0, 1\}$
  - Visualization/Graph compression
  - Clustering/Community detection

$$f : V \rightarrow \{1, \ldots, k\}$$

- **Supervised**
  - Node Classification $f : V \rightarrow \{0, 1\}$
  - Graph Classification $f : G \rightarrow \{0, 1\}$



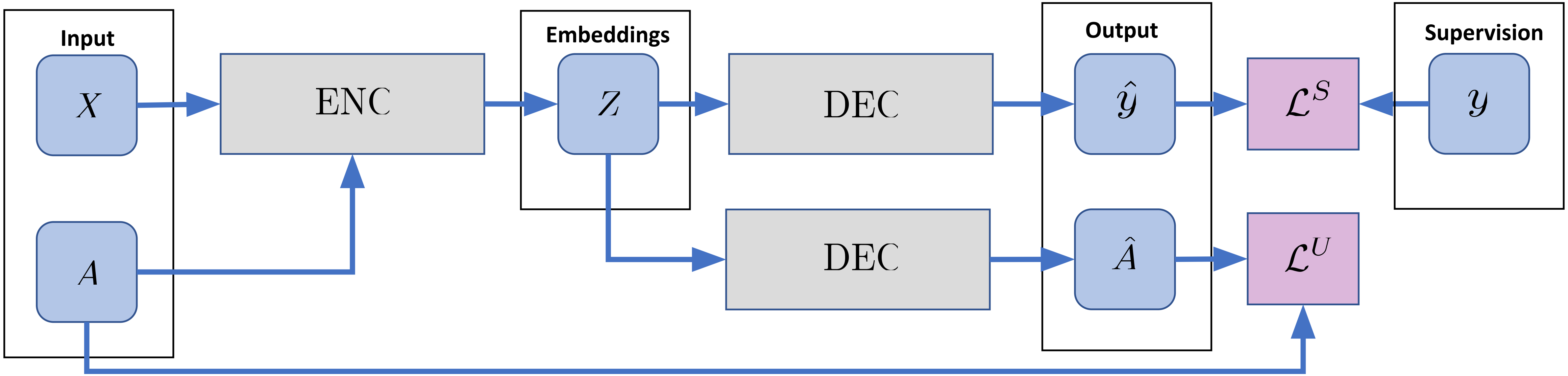


$\rightarrow \left\{ \begin{array}{l} \text{Toxic} \\ \text{Non-toxic} \end{array} \right.$

Figure: Hapala et al - Nature Comms

Our Model & Taxonomy

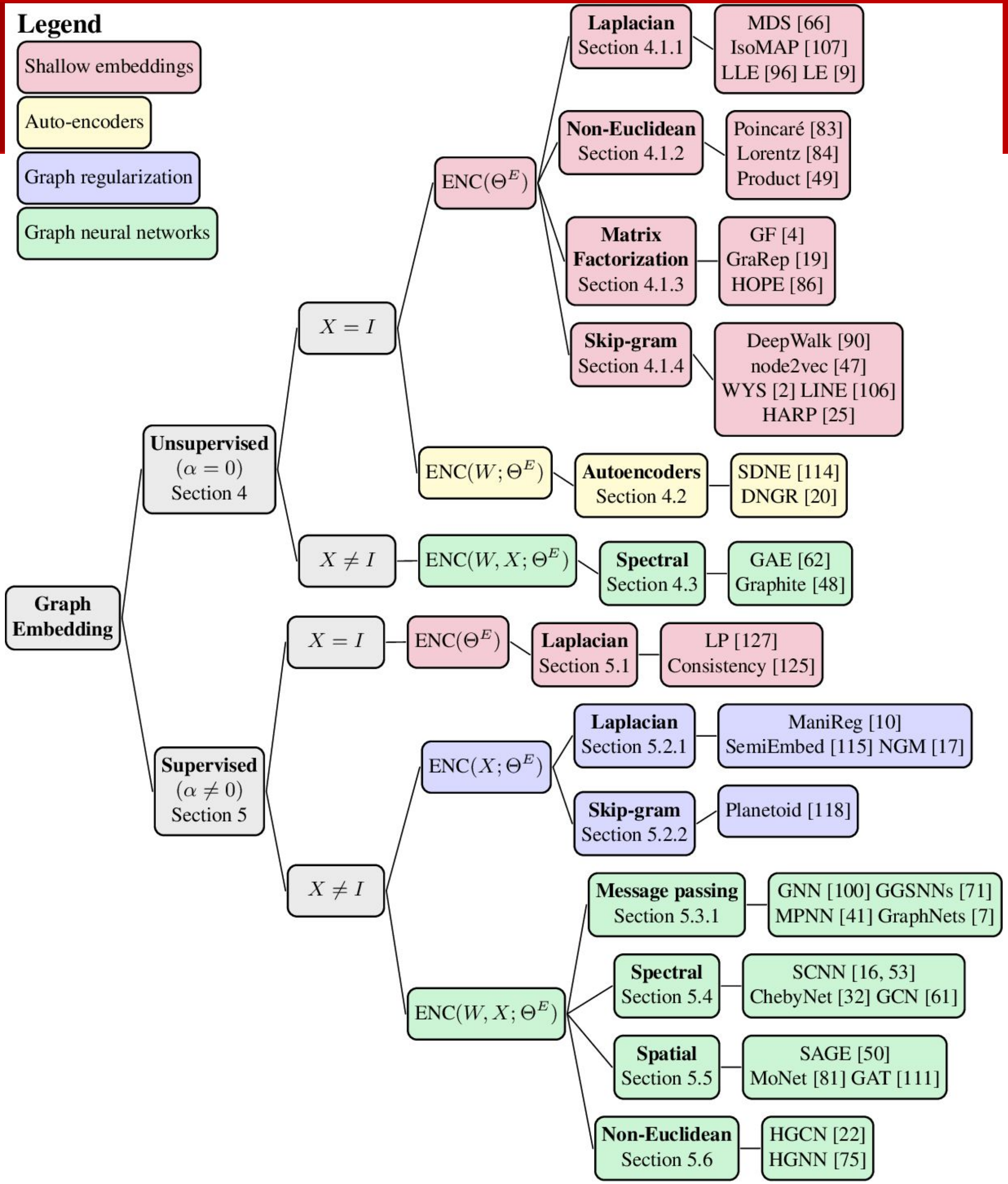
The GraphEDM Model



GraphEDM is a general model describing how two inputs (Graph A and Feature Matrix X) are transformed into latent space (Z) and then utilized for tasks.

Taxonomy of GNNs

We can then map GNN models into the parameter space of GraphEDM to get a Taxonomy of methods.



Unsupervised Methods

Problem Setup

Input: Graph $G = (V, E)$

- Vertices: $V = \{v_1, \dots, v_n\}$
- Adjacency matrix: $A \in \mathbb{R}^{n \times n}$
- (Optional) Node features: $X \in \mathbb{R}^{n \times d_0}$

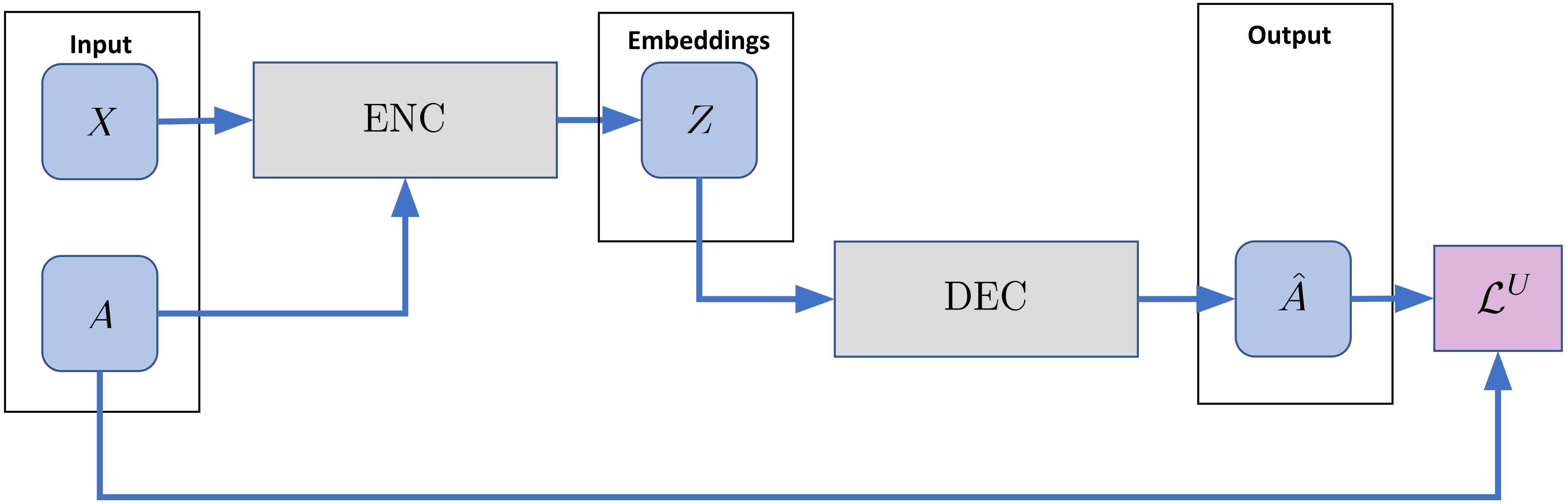
Goal: Learn a mapping

$$f : V \to \mathbb{R}^d$$
$$v_i \to z_i$$

Such that graph information is preserved

$$\mathrm{sim}(v_i, v_j) \approx \mathrm{sim}(z_i, z_j)$$

GraphEDM for Unsupervised GRL



Shallow Encoders

Shallow encoder = simple embedding look-up:

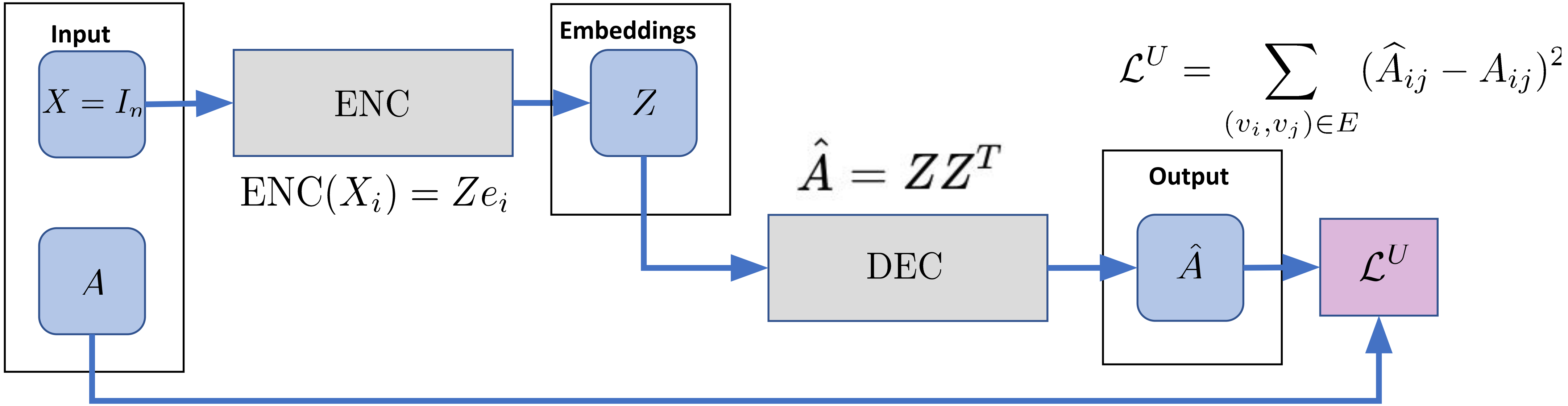
$$X = I_n$$

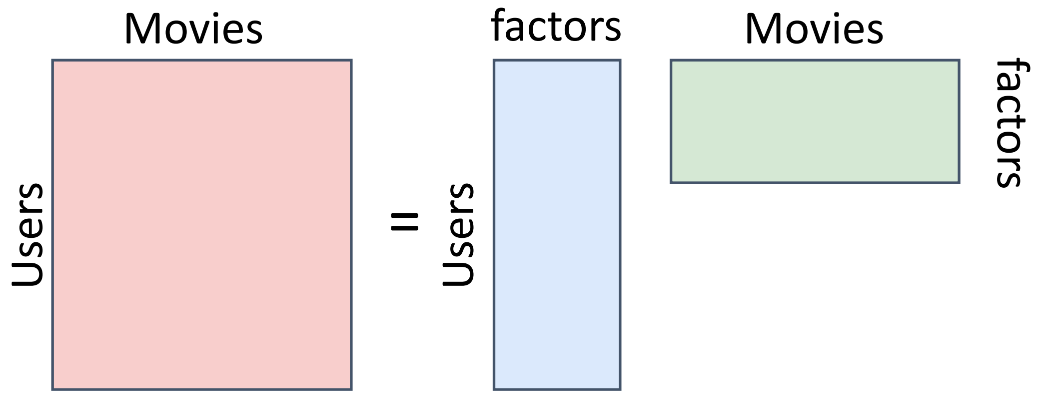
$$\mathrm{ENC}(X_i) = Z e_i = \underbrace{\begin{bmatrix} -z_1^\top- \\ \vdots \\ -z_i^\top- \\ \vdots \\ -z_{|V|}^\top- \end{bmatrix}}_{|V| \times d} \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}}_{d \times 1}$$

Embedding matrix

Example: Graph Factorization (Ahmed et al. 2013)

Idea: Learn a low-rank decomposition of the similarity matrix (1st order)





Possible extensions:

- directed graphs using asymmetric embeddings (source + target): GraRep (Cao et al. 2015), HOPE (Ou et al. 2016)
- Higher-order representations using random walks: DeepWalk (Perozzi et al. 2014), node2vec (Grover et al. 2016)

Supervised Methods

Problem Setup

Input: Graph $G = (V, E)$

- Vertices: $V = \{v_1, \ldots, v_n\}$
- Adjacency matrix: $A \in \mathbb{R}^{n \times n}$
- (Optional) Node features: $X \in \mathbb{R}^{n \times d_0}$

Goal: Learn a mapping $f : V \rightarrow y$

That predicts graph properties (node/graph labels, …)

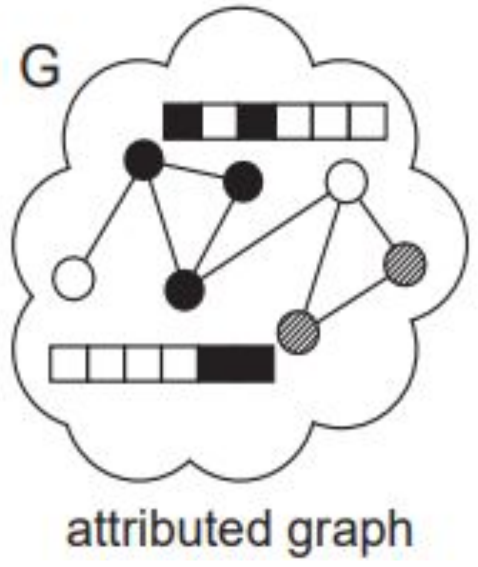
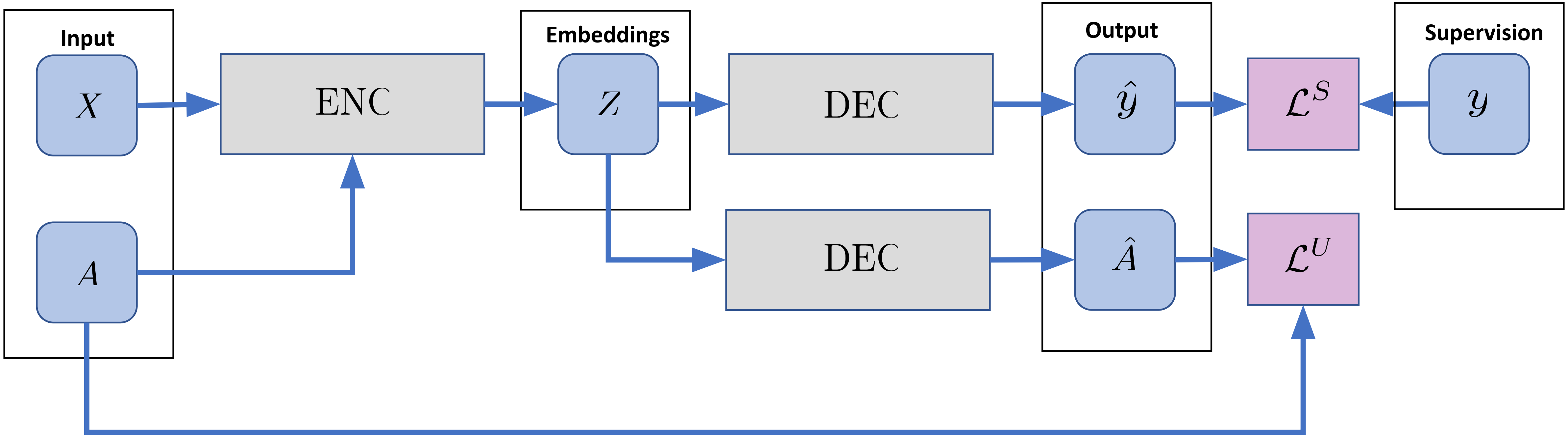
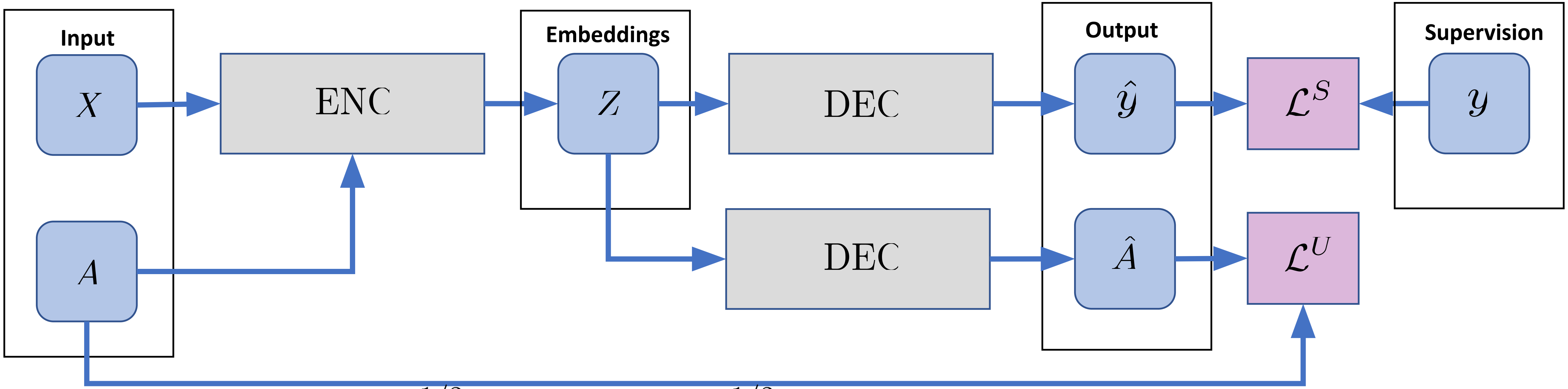


Figure: [Rezaei et al, WWW'17]

GraphEDM for Supervised GRL



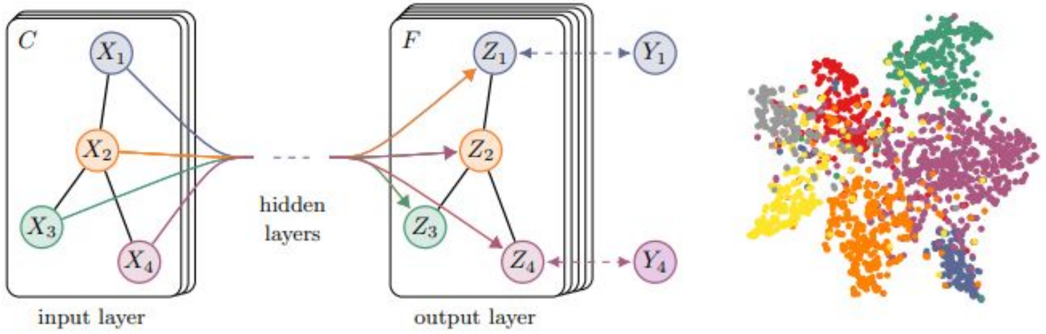
Example: Graph Convolutions (Kipf et al. 2017)



$$\mathrm{ENC}(A, X) = \sigma((D+I)^{-1/2}(A+I)(D+I)^{-1/2}XW)$$

Neighborhood aggregation

Feature transform



(a) Graph Convolutional Network

(b) Hidden layer activations

Figure: Kipf and Welling

Thank you!

Many other methods!

Paper:

Machine Learning on Graphs: A Model and Comprehensive Taxonomy (**JMLR**, 2022)

Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, Kevin Murphy

Paper: https://arxiv.org/abs/2005.03675

LinkedIn:

Twitter: @phanein