# Generating Large Synthetic and Real Graph Datasets

**Arpandeep Khatua,** Mentors: Professor Wen-Mei Hwu, Vikram Sharma Mailthody
Department of Electrical and Computer Engineering, Grainger College of Engineering, University of Illinois at Urbana-Champaign

C3SR center for cognitive computing systems research

## INTRODUCTION

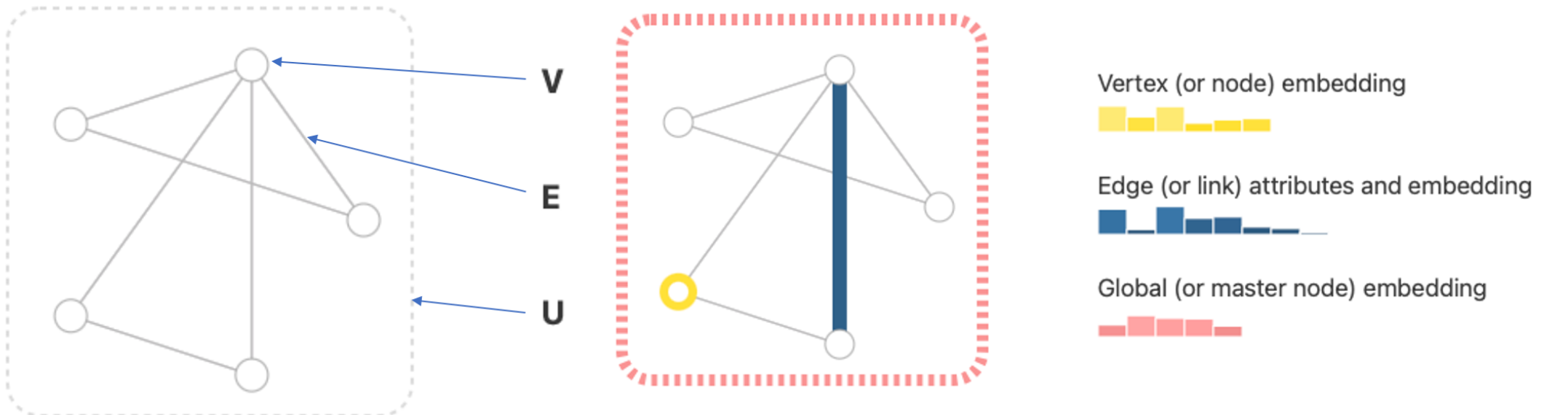
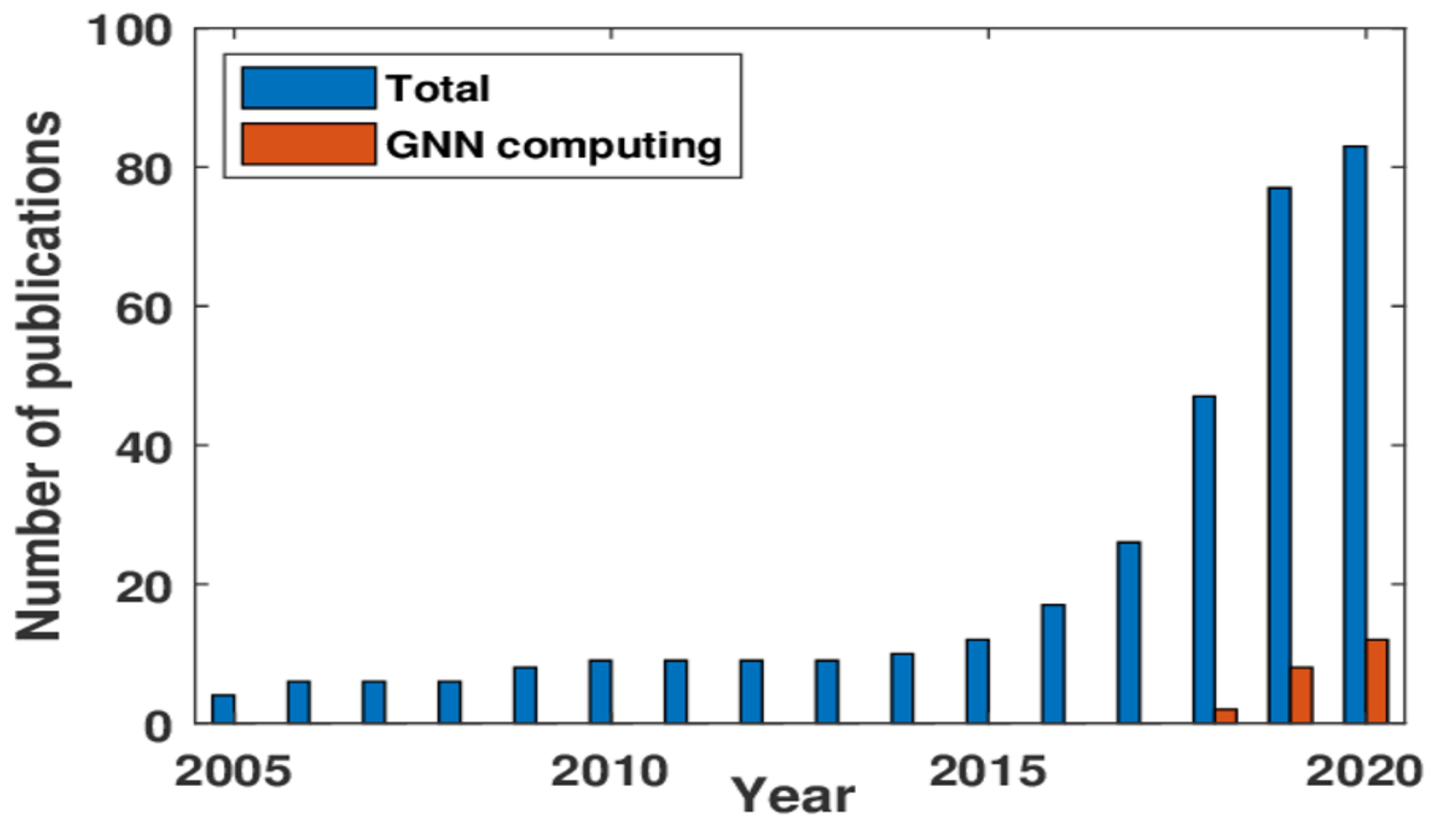

V
E
U
Vertex (or node) embedding
Edge (or link) attributes and embedding
Global (or master node) embedding

Fig 1: Graph representation. Sanchez-Lengeling, et al., "A Gentle Introduction to Graph Neural Networks", Distill, 2021.

Graphs are powerful data structures used to solve complex problems like recommender systems, path optimization and influence prediction.

Broadly there are 5 main groups we can split these tasks:

- *Node Classification*
- *Graph Classification*
- *Node Clustering*
- *Link Prediction*
- *Influence Maximization*



Plot 1: Bar chart comparing the research publication of all machine learning papers vs GNN papers.

## WHY

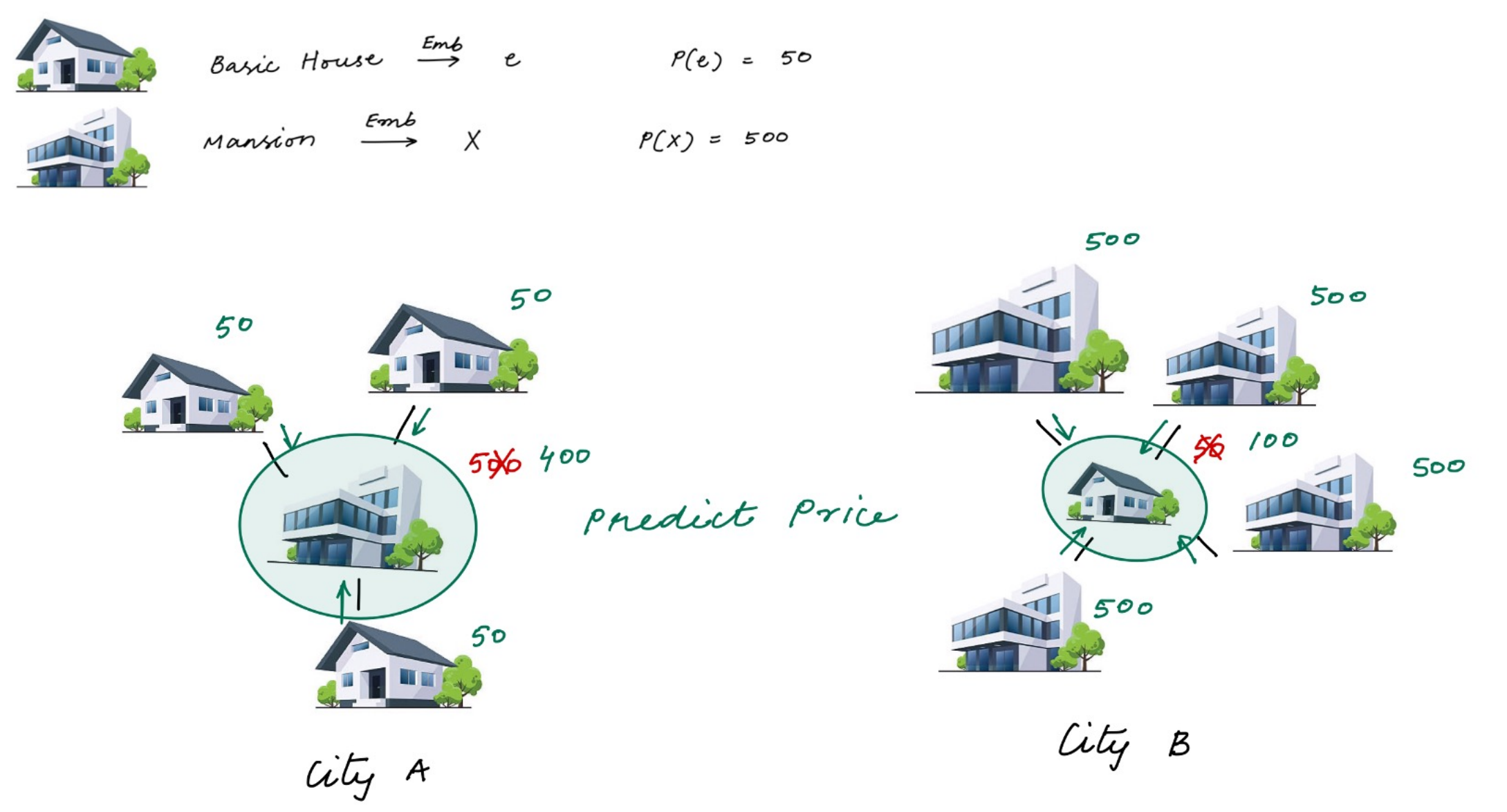


Fig 2: Why GNNs are an area of interest.

- *Using embedding generation methods, we can assign a certain price to different types of house.*
- *When used to predict house price this method is not optimal since the neighboring houses will affect the price of the house.*
- *This is where GNNs can be used.*

Classical algorithms cannot provide additional insight about new data. This can be addressed using deep learning methods. Graph Neural Nets (GNNs), a class of deep neural net, has become a popular method to process graph data to solve downstream tasks like clustering, classification, edge prediction and influence maximization problems pertaining to above emerging applications due to their unique capabilities like incorporation of node, edge and graph level information into the output prediction.

## GRAPH NEURAL NET MODELS

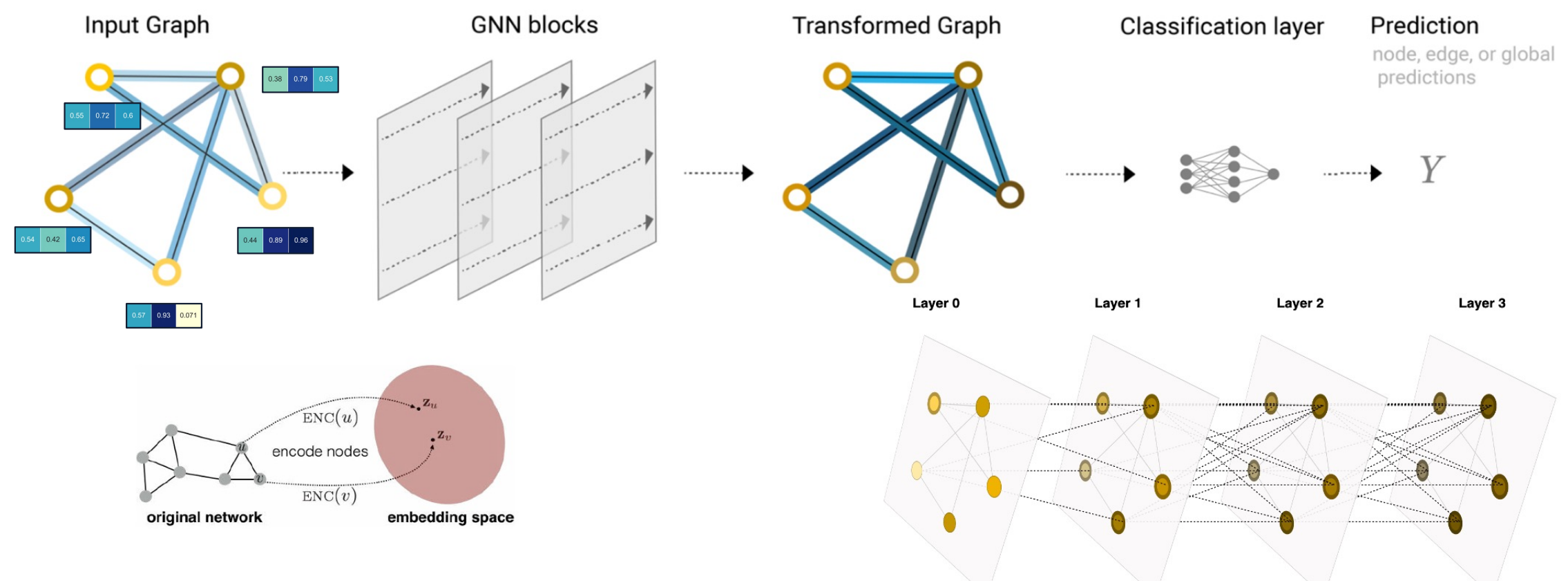


Fig 3: GNN pipeline and embedding generation illustration.

Given an input graph, we can create node embedding using NLP processes like word2vec. This becomes the input for the N-layer Graph Neural Net Model which gives us the transformed graph. These node embeddings can now be used as an input for a neural net for the downstream tasks.

$$h_i^{l+1} = \sigma\Big(\sum_{j\in\mathcal{N}_i} W^l \cdot h_j^l\Big) \Rightarrow \mathbf{H}^{l+1} = \sigma\big(\mathbf{W}^l \cdot \mathbf{H}^l\big)$$

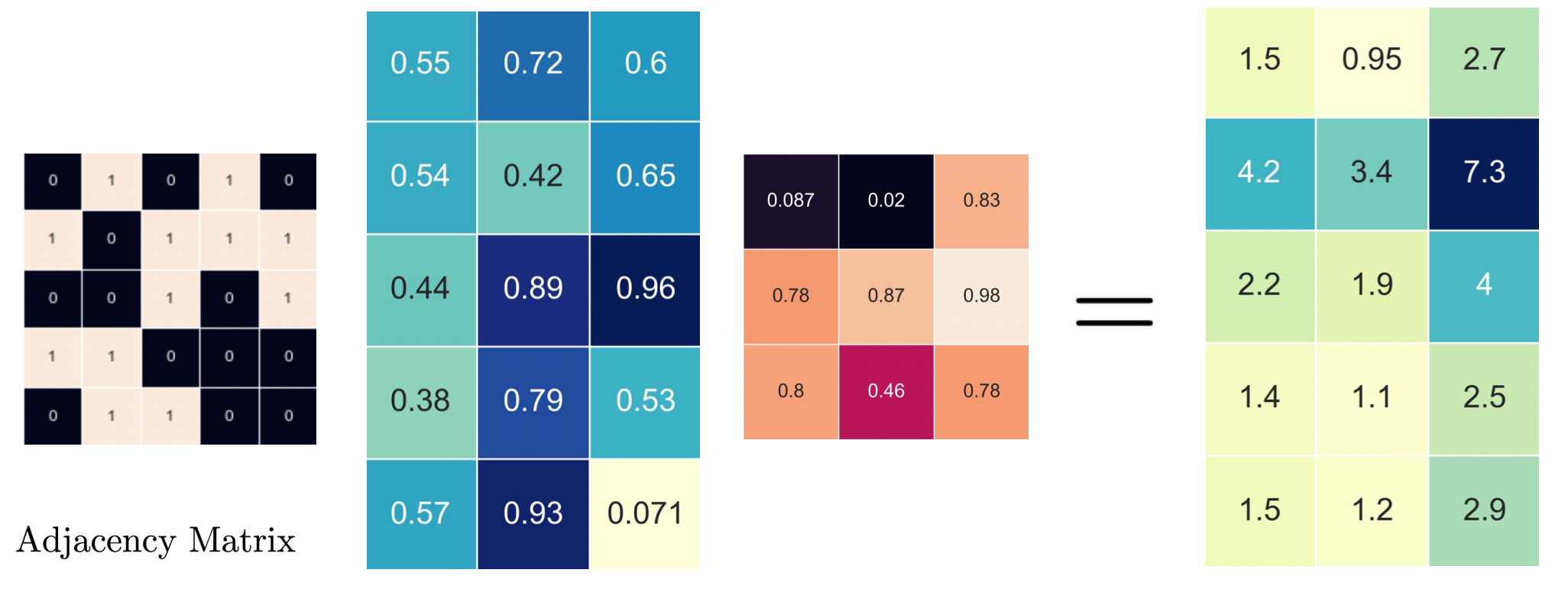


Fig 4: Matrix multiplication showing one layer of a naïve GNN.

### Homogeneous Graph Neural Net Models

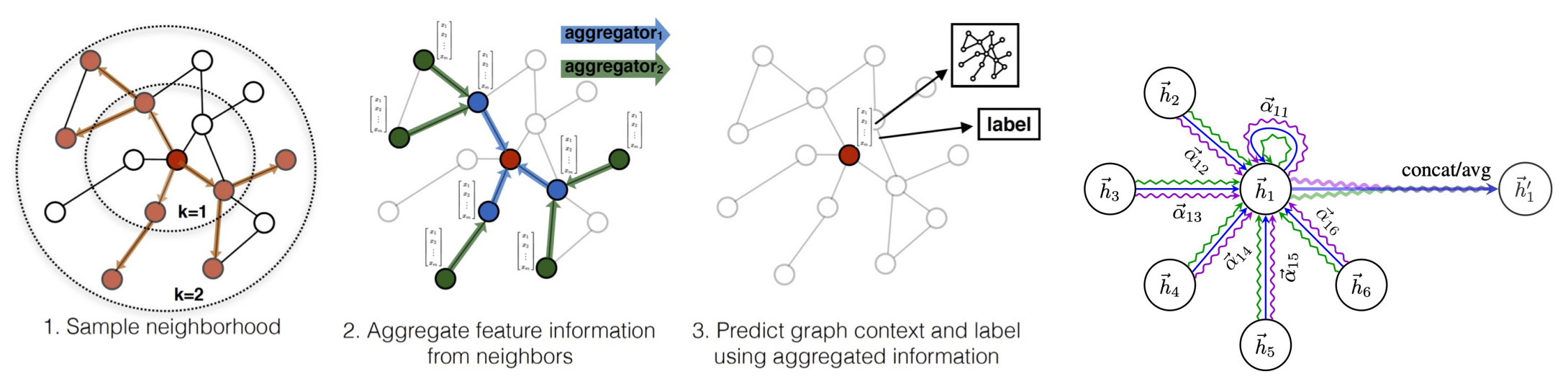

k=1
k=2
aggregator1
aggregator2
label
concat/avg
1. Sample neighborhood
2. Aggregate feature information from neighbors
3. Predict graph context and label using aggregated information

Fig 5: Different types of homogeneous graph neural network models

$$\textbf{Naïve Graph Neural Net: } h_i^{l+1} = \sigma\Big(\sum_{j\in\mathcal{N}_i} \mathbf{W}^l \cdot h_j^l\Big)$$

$$\textbf{Graph Convolutional Neural Net: } h_i^{l+1} = \sigma\Big(\sum_{j\in\mathcal{N}_i} \frac{1}{c_{ij}} \mathbf{W}^l \cdot h_j^l\Big)$$

$$\textbf{GraphSAGE: } h_i^{l+1} = \sigma\Big(\mathbf{W}^l \cdot [\mathbf{AGGEGATOR}_{j\in\mathcal{N}_i}(h_j^l), h_i^l]\Big)$$

$$\textbf{Graph Attention Network: } h_i^{l+1} = \Big\Vert_{k=1}^{K} \sigma\Big(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^{k,l} \cdot h_j^l\Big)$$

### Heterogeneous Graph Neural Net Models

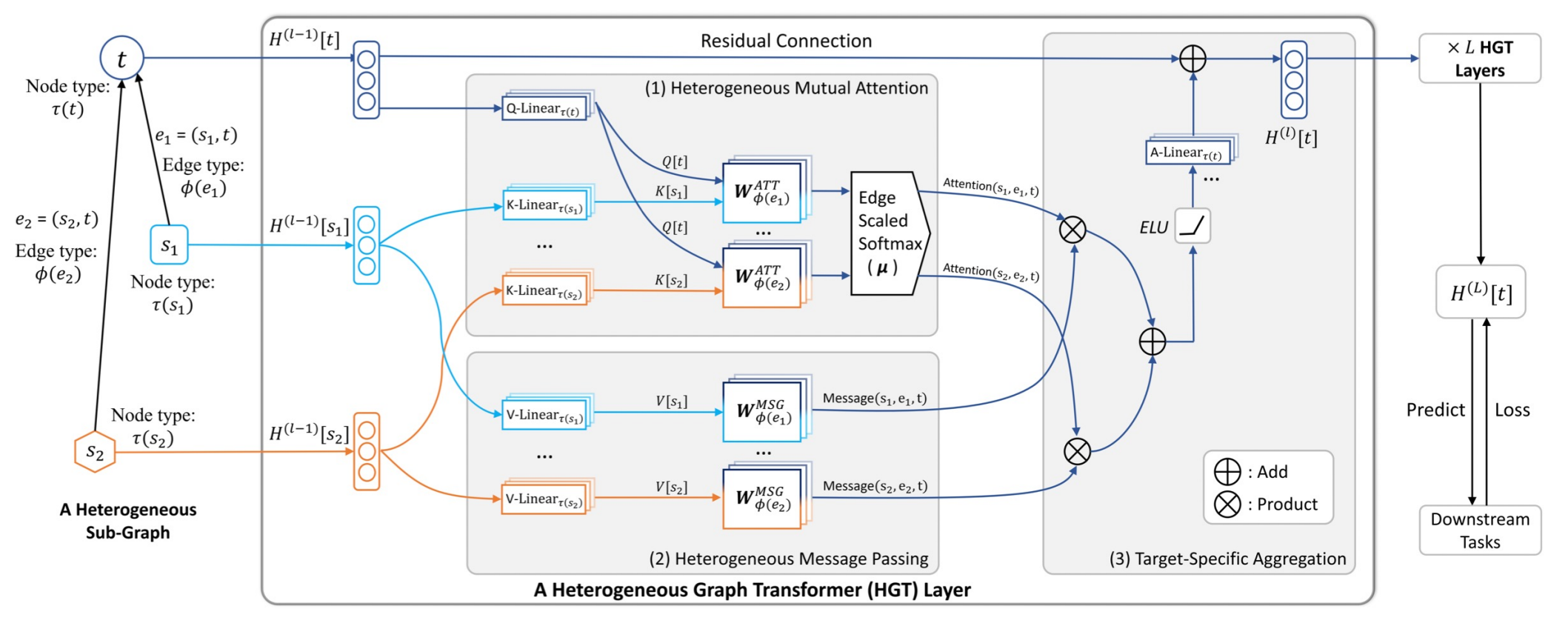

Residual Connection
(1) Heterogeneous Mutual Attention
(2) Heterogeneous Message Passing
(3) Target-Specific Aggregation
A Heterogeneous Graph Transformer (HGT) Layer
A Heterogeneous Sub-Graph
Predict
Loss
Downstream Tasks
Add
Product

Fig 6: One layer of a Heterogeneous Graph Transformer Model.

## CREATION OF IGB

### Challenges to GNN community

| Dataset | Date | Type | Node emb size | #nodes (millions) | #edges (millions) |
|---|---|---|---|---|---|
| OBGN-papers | 2020 | Real | 128* | 111 | 1,615 |
| MAG240M | 2021 | Real | 768 | 260 | 1,300 |
| **OUR DATASET v1** | **2022** | **Real/Synthetic** | **128 - 4kB** | **267 (only paper)** | **1,900 (only paper)** |
| **OUR DATASET v2** | **2022** | **Real/Synthetic** | **variable** | **~600+ (expected)** | **~ 3,000+ (expected)** |
| PinSAGE dataset | 2018 | Real | 128-2K (avg. 1K) | 3000 | 18,000 |

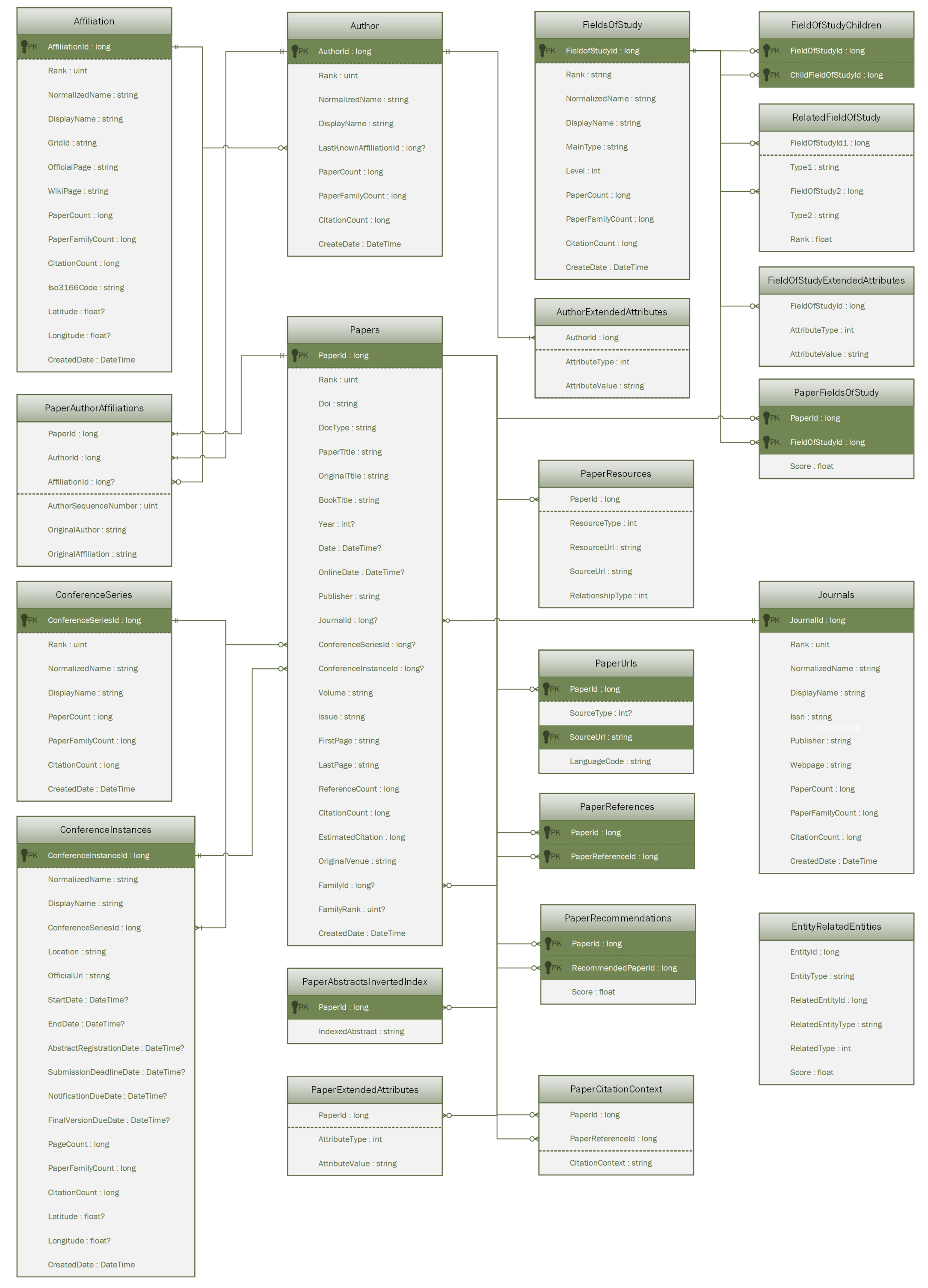
Table 1: Comparison between existing publicly available graph datasets, our dataset and an industry proprietary dataset.

Our goal is to propose a new dataset, Illinois Graph Benchmark (IGB) that will help both system designers and GNN researchers in ***two*** ways:

- Given a dataset schema, propose a methodology to generate ***arbitrary sized graphs (homogenous or heterogenous) and node embeddings with prescribed number of nodes, edges and relations.***
- Provide a ***dataset with synthetic node embeddings*** for system developers and another ***dataset with node embeddings generated using NLP methods*** for GNN researchers and system developers.

### Synthetic embedding vs real embedding

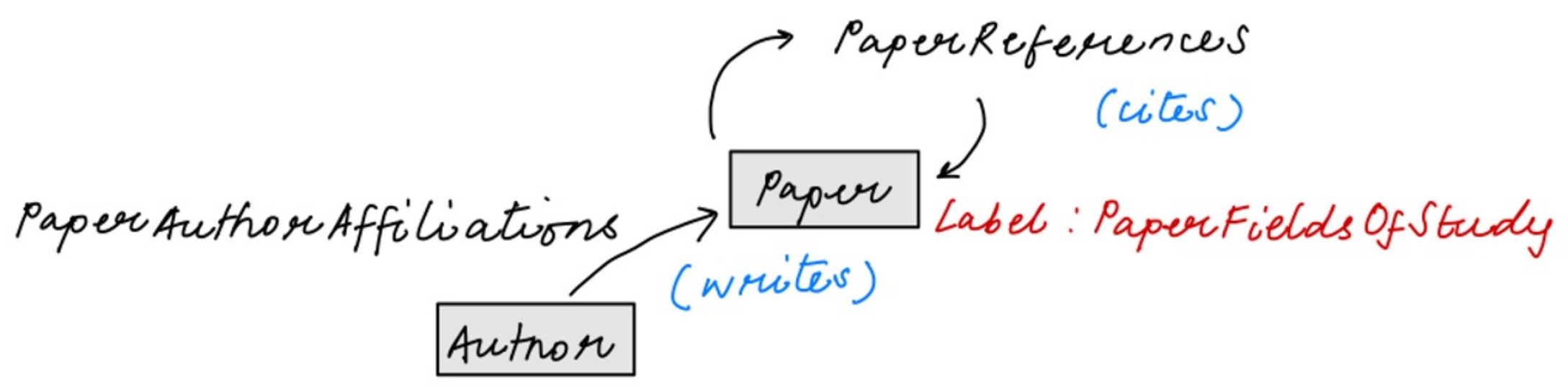
- Fundamentally GNN's find structural information of the graph to improve the embeddings and have ***no idea about the node's information***.
- Synthetic graph dataset would be useful to test computation and optimization the results
- Synthetic graph datasets ***do not have any real-world significance for downstream tasks.*** For example, GAT would be completely useless for a synthetic dataset



Graph 1: Microsoft Academic Graph (MAG) database schema

## PROPERTIES

### Illinois Graph Benchmark (IGB) dataset schema



Graph 2: IGB heterogeneous graph dataset

### Generating Real Word Node Embeddings

In the MAG dataset every paper entry has a linked abstract and title. We use this text as an input to the Sentence-BERT model to generate an embedding of an arbitrary length.

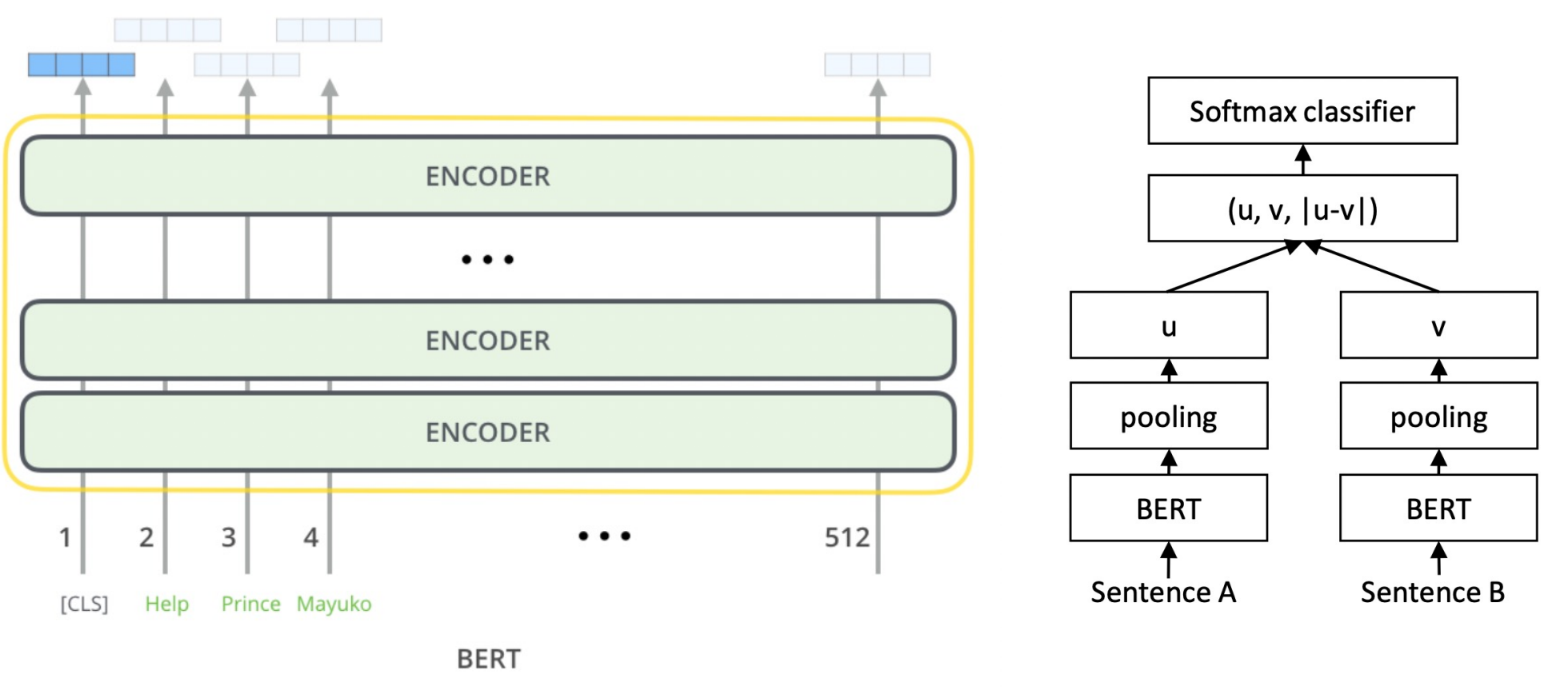


Fig 7: Each BERT module and a Siamese BERT model architecture of Sentence-BERT

### Generating Labels

We use the Semantic Scholars dataset which maps paper_ids to the fieldOfStudy.

```
{
    "id": "4cd223df721b722b1c40689caa52932a41fcc223",
    "fieldsOfStudy": ["Computer Science"],
}
```

Fig 8: Each entry of the Semantic Scholar dataset

This dataset has 11 different classes for all the English paper nodes. However, to increase flexibility of number of classification classes, we use the the K-Means algorithm to generate unsupervised labelling data for even more graph nodes using the conference and Journal information. Using this method, we can create an arbitrary number of classes for GNN developers to test their models.

## ACKNOWLEDGEMENTS

Special thanks to all the IMPACT Research Group members, C3SR Center, Bhagyashree Taleka and Tengfei Ma.

ILLINOIS