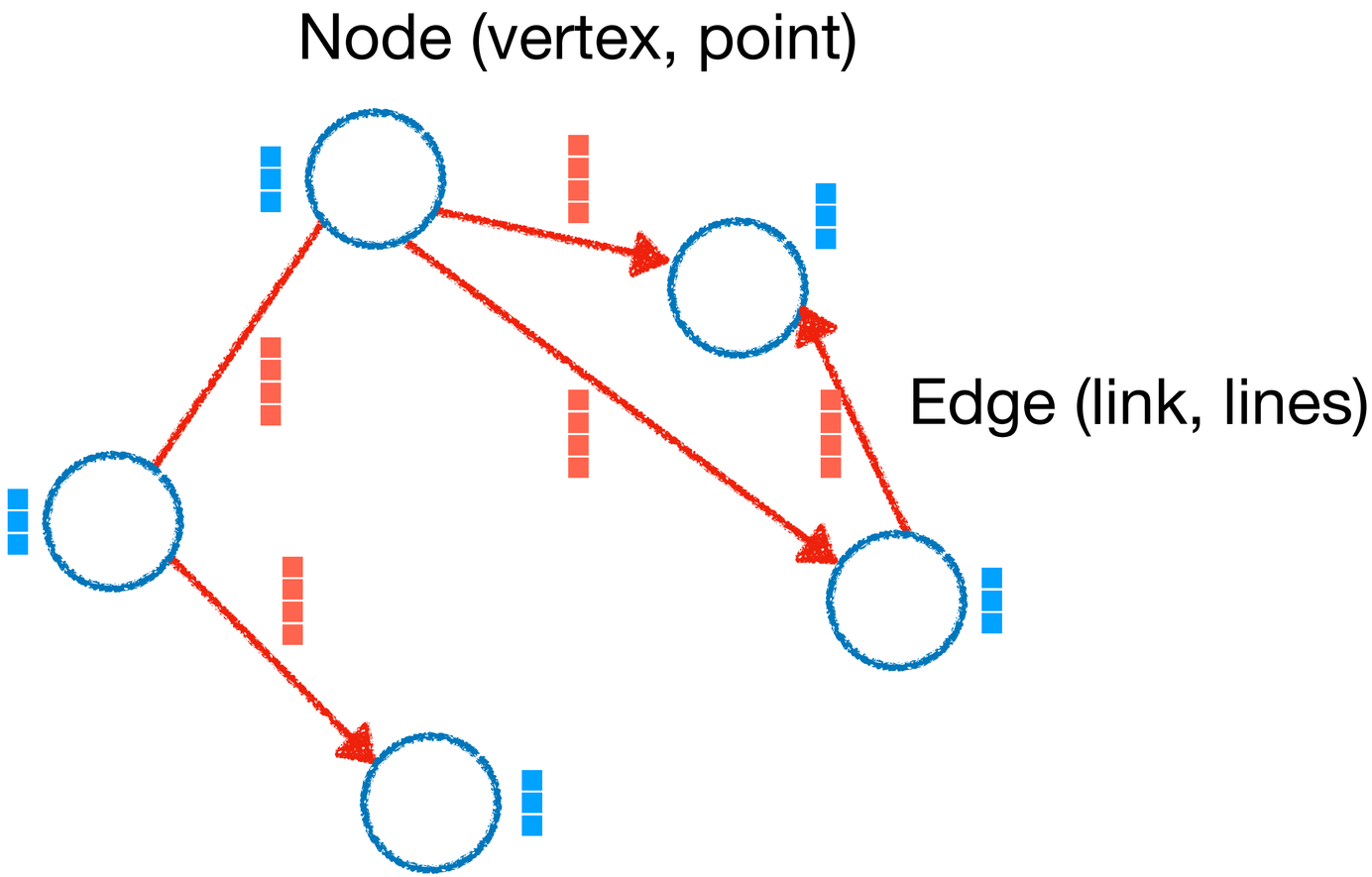
# Review of A Review of GNN

**Reference**

**Jinheung Kim**

# Graph

$$G = \{N, E\}$$



# GNN Application

**Table 1** GNN papers with their performance

| Refs. | Application area | Dataset used | Model applied | Summary | Performance evaluation |
|---|---|---|---|---|---|
| [13] 2018 | Link prediction | Usair<br>NS<br>PB<br>Yeast<br>C.ele<br>Power<br>Router<br>And E.coli | GNN | They defend the use of prediction heuristics to learn from local enclosing subgraphs | Area Under Curve:<br>Usair—97.09 ± 0.70<br>NS—97.71 ± 0.93<br>PB—95.01 ± 0.34<br>Yeast—97.20 ± 0.64<br>C.ele—89.54 ± 2.04<br>Power—84.18 ± 1.82<br>Router—95.68 ± 1.22<br>E.coli—97.22 ± 0.28 |
| [14] 2018 | Solving matrix equations | Own examples | Hybrid GNN<br>GNN + ZNN<br>(Zhang Neural Network) | They solve the matrix equations BX = D and XC = D in time-invariant cases | The global convergence rate has improved by taking their example |
| [15] 2018 | Solving matrix equations | Explains theorems with their examples | Gradient-based neural dynamics (GND) | Solve matrix equation AXB = D | The global convergence rate has improved by taking their example |
| [16] 2019 | Link forecast<br>Recommendation<br>Node Clustering and Node Classification | Academic I (A-I)<br>Academic II (A-II)<br>Movies Review (R-I)<br>Cds Review | Hetgnn | Hetgnn considered combining heterogeneous types, type-based neighbors, and heterogeneous node contents | AUC:<br>Multi-label classification—0.978<br>Node clustering –<br>0.901 |
| [17] 2019 | Social Recommendation | Ciao and Epinions | Graphrec | They predict ratings and provide interactions and opinions on the user-item graph | RMSE:<br>Ciao—0.9794<br>Epinions—0.8168 |
| [18] 2019 | Chinese-named entity recognition | Ontonotes<br>MSRA<br>Weibo<br>Resume | Lexicon-based GNN | Chinese NER is achieved as a graph node classification using a vocabulary to build a graph neural network | 74.89<br>93.46<br>60.21<br>95.37 |
| [19] 2019 | Taxable detection & structure recognition | UW3, UNLV,<br>ICDAR 2013 | CNN + GNN | The best networks for detecting representative visual features are convolutional neural networks, whereas the best networks for quick message transfer between vertices are graph networks. With the help of the gather operation, we have demonstrated how to integrate these two skills | 68.5 |

# GNN Application

**Table 1** (continued)

| Refs. | Application area | Dataset used | Model applied | Summary | Performance evaluation |
|---|---|---|---|---|---|
| [20] 2019 | Link prediction<br>Pair-wise node classification | Grid Communities PPI | P-GNN<br>Point of view GNN | To compute node embeddings that contain node positional information while maintaining inductive capability and leveraging node attributes, they introduce a new class of GNN | AUC:<br>0.940 ± 0.027<br>0.985 ± 0.008<br>0.808 ± 0.003 |
| [21] 2020 | Time-series Forecasting | METR-LA<br>PEMS-BAY<br>PEMS07<br>PEMS03<br>PEMS04<br>PEMS08<br>Solar<br>Electricity<br>ECG5000<br>COVID-19 | Spectral Temporal GNN<br>Stemgnn) | | RMSE:<br>5.06<br>2.48<br>4.01<br>21.64<br>32.15<br>24.93<br>0.07<br>0.06<br>0.07<br>19.3 |
| [22] [2020] | Citation Network | Cora, Citeseer, Pubmed, and NELL | Continuous GNN(CGNN) | Enable continuous instances to be handled by existing discrete graph neural networks by describing the evolution of node representations with ODE | 82.1 ± 1.3<br>72.9 ± 0.9<br>82.7 ± 1.4<br>73.1 ± 0.9 |
| [23] 2020 | Node representation visualization | Cora,<br>Citeseer,<br>Pubmed<br>Coauthors | Differentiable group normalization (DGN), simple graph convolution networks (SGC) | They propose group distance ratio and instance information gain as two over-smoothing metrics based on graph architectures | 80.2%<br>58.2%<br>76.2%<br>85.8% |
| [24] 2020 | Medical | MUTAG<br>PTC<br>COX2<br>PROTEINS<br>NCI1 | Implicit graph neural network (IGCN) | They outline a Perron-Frobenius hypothesis necessary condition for very well and a projected gradient descent training approach | 89.3 ± 6.7<br>70.1 ± 5.6<br>86.9 ± 4.0<br>77.7 ± 3.4<br>80.5 ± 1.9 |
| [25] 2021 | Text classification | IMDB webkb<br>R52<br>R8<br>AG_news | Deep Attention Diffusion Graph Neural Network (DADGNN) | Proposes an attention diffusion technique that captures non-direct-neighbor context information in a single layer and decouples the required GNN training processes (representation transformation and propagation) | 88.49 ± 0.59 90.92 ± 0.42 95.16 ± 0.22<br>98.15 ± 0.16 92.24 ± 0.36 |

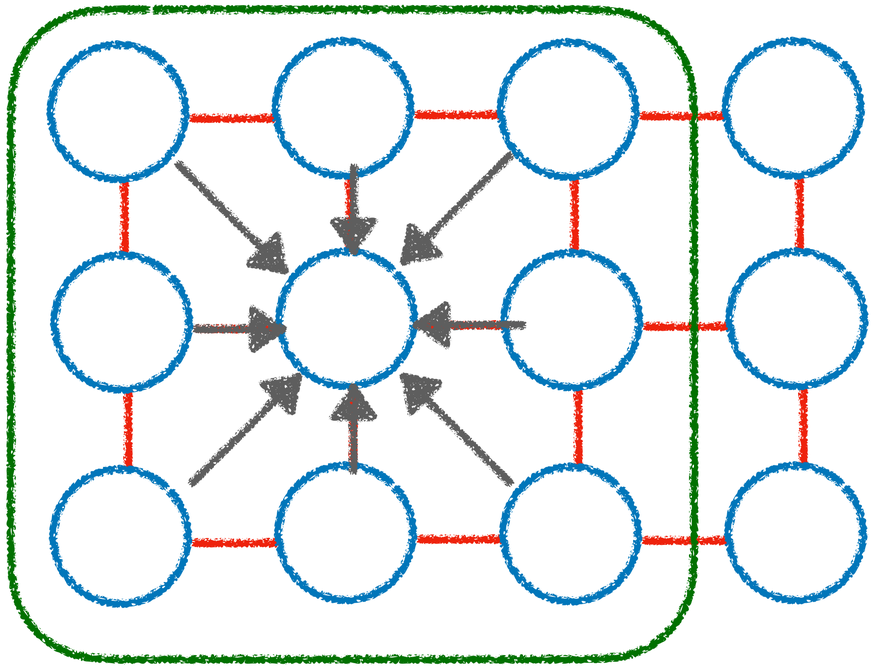
# GNN Application

**Table 1** (continued)

| Refs. | Application area | Dataset used | Model applied | Summary | Performance evaluation |
|---|---|---|---|---|---|
| [26] 2021 | Medicines | COLL, MD17, and OC20 | Neural Network For Geometric Messages Passing | Gemnet uses effective bilinear layers and symmetric message passing | 34%, 41%, and 20% |
| [27] 2022 | Feature extraction | Pavia University<br>Salinas<br>Houston 2013 | Deep Hybrid Multi-Graph Neural Network (DHMG) | To reduce the noise in the graph, they created a unique ARMA filter and implemented it recursively | $97.81 \pm 0.82$<br>$98.33 \pm 0.28$<br>$93.31 \pm 0.65$ |
| [28] (2023) | Traffic Prediction | AIS data and global port geospatial data | GAT | Research on Multi-Port Ship Traffic Prediction Method Based on Spatiotemporal Graph Neural Networks | Around 90% |
| [29] (2023) | Traffic Forecasting | PEMS03,<br>PEMS04,<br>PEMS07,<br>PEMS08 | GNN | Hybrid GCN and branch-and-bound optimization for traffic flow forecasting | 0.58<br>0.63<br>0.63<br>0.73 |

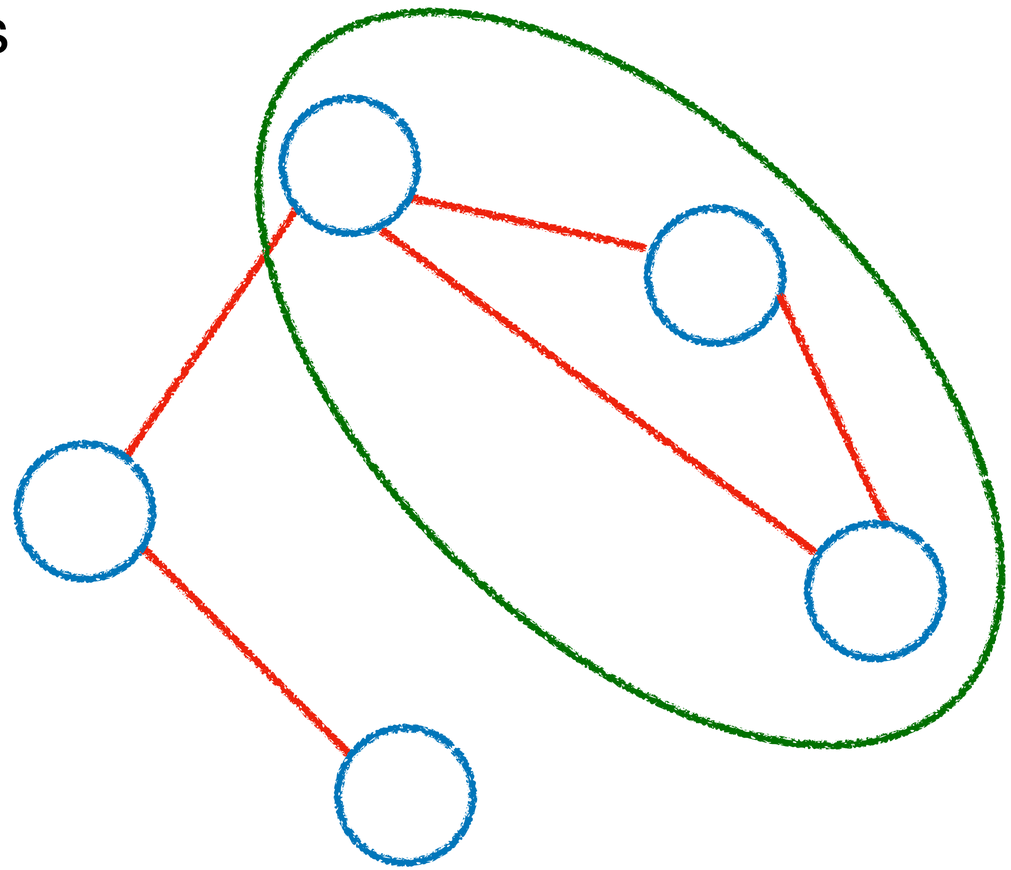
# Convolution Neural Network

- Grid Structures



# Graph Convolution Network

- Non-Grid Structures



# Message Passing Mechanism

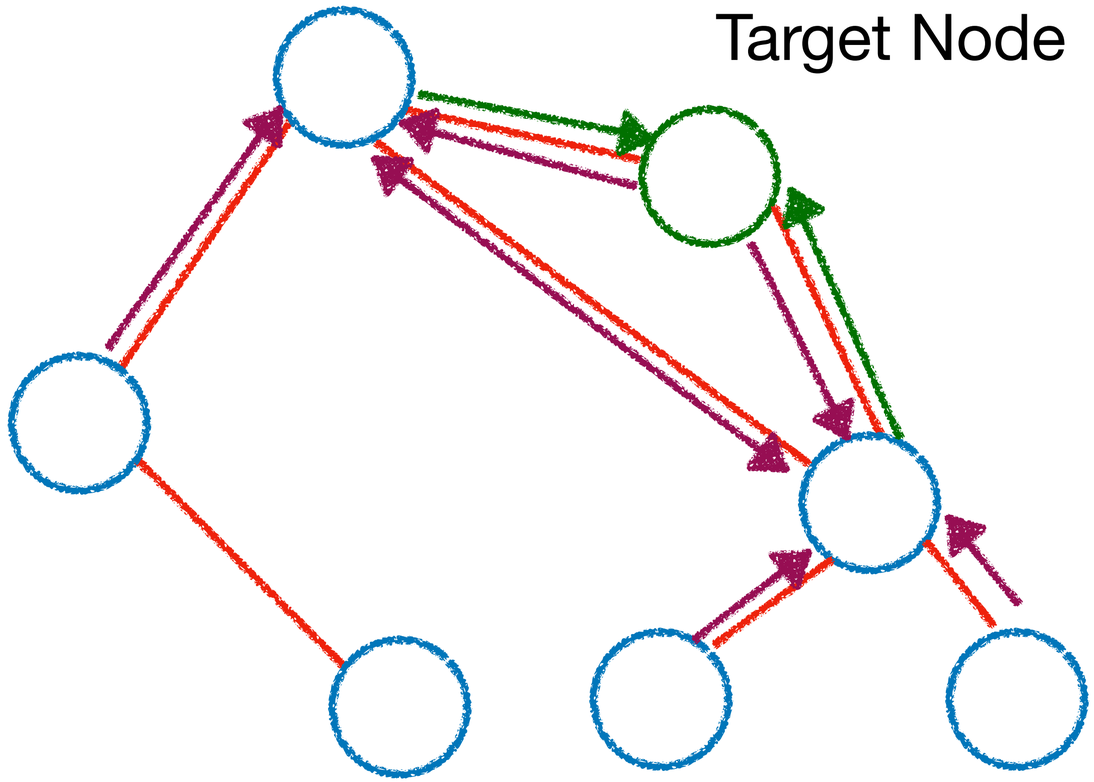
Node collects information form the neighborhood

iteration

$k = 1$, 1-hop neighborhood

$k = 2$, 2-hop neighborhood

$k = n$, n-hop neighborhood



# Message Passing Mechanism

Aggregates the information

Features of node $u$

$$h_u^{(k+1)} = UPDATE^{(k)}\left(h_u^{(k)}, AGG^{(k)}\left(\{h_v^{(k)}, \forall v \in N(u)\}\right)\right)$$

Neighborhood of node $u$

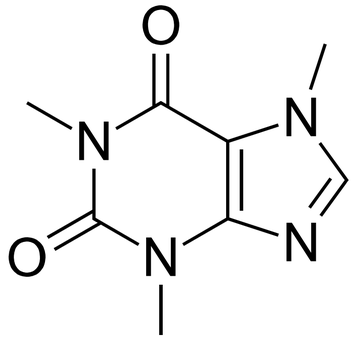
# Graph Taxonomy

## Structure and Non-structure

### Structure

Explicit Graph structure

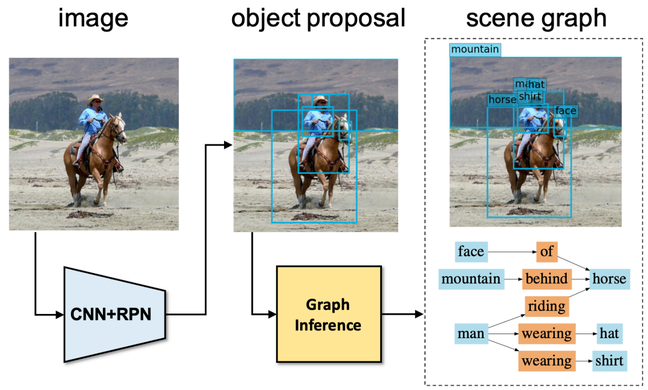
- Molecules
- Physical systems
- Knowledge graphs




### Non-structure

Implicit Graph structure

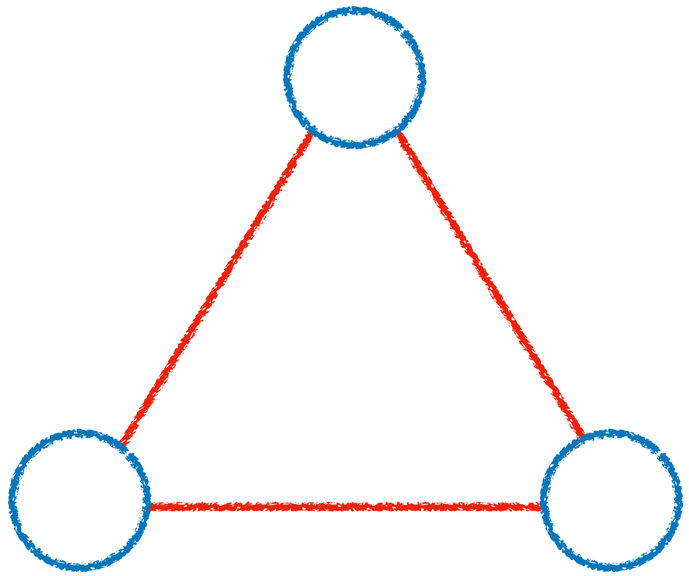
- “Word” graph for text
- “Scene” graph for an image



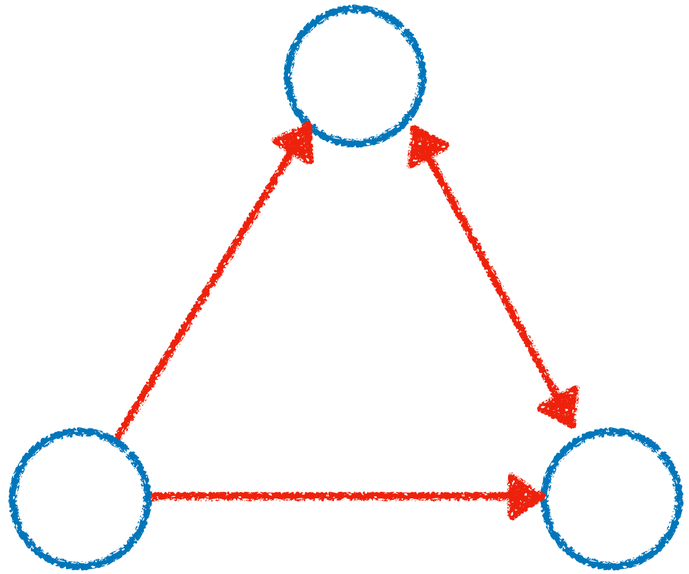

1701.02426

# Graph Taxonomy

## Directed and Undirected Graph



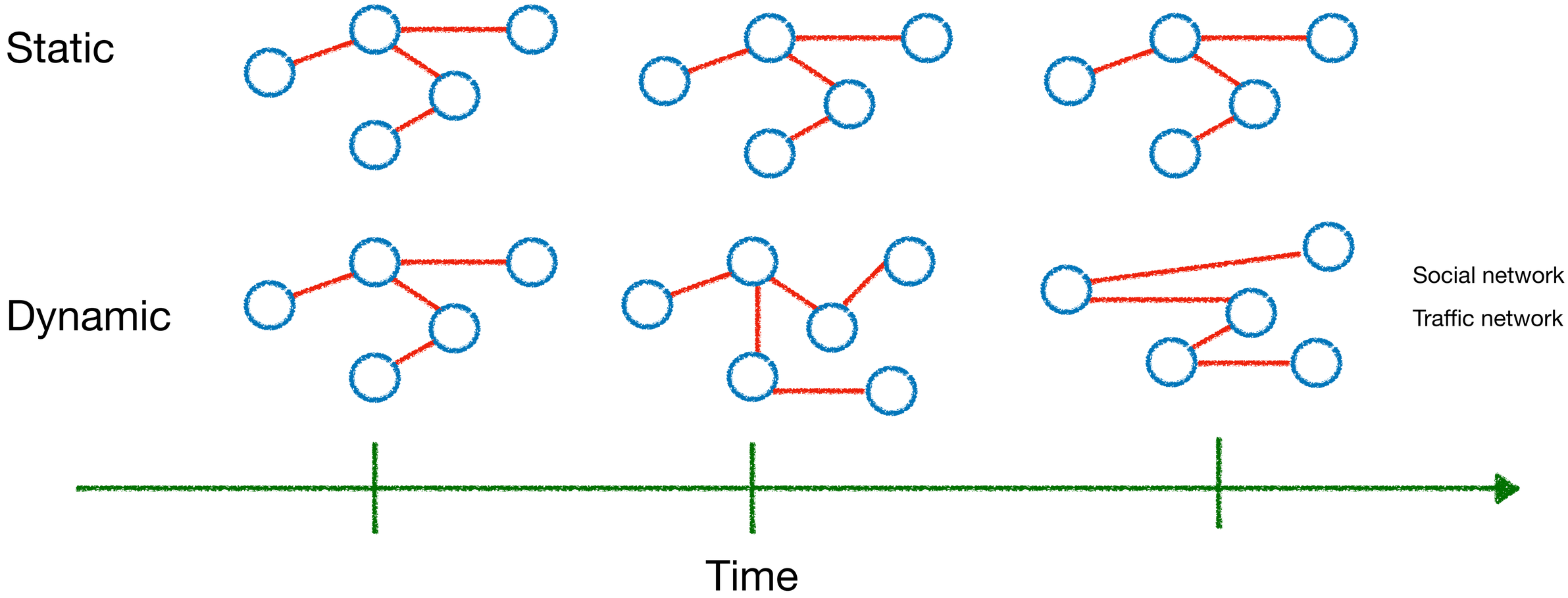
Undirected



Directed

# Graph Taxonomy

## Static and Dynamic Graph



# Graph Taxonomy

## Homogeneous and Heterogeneous

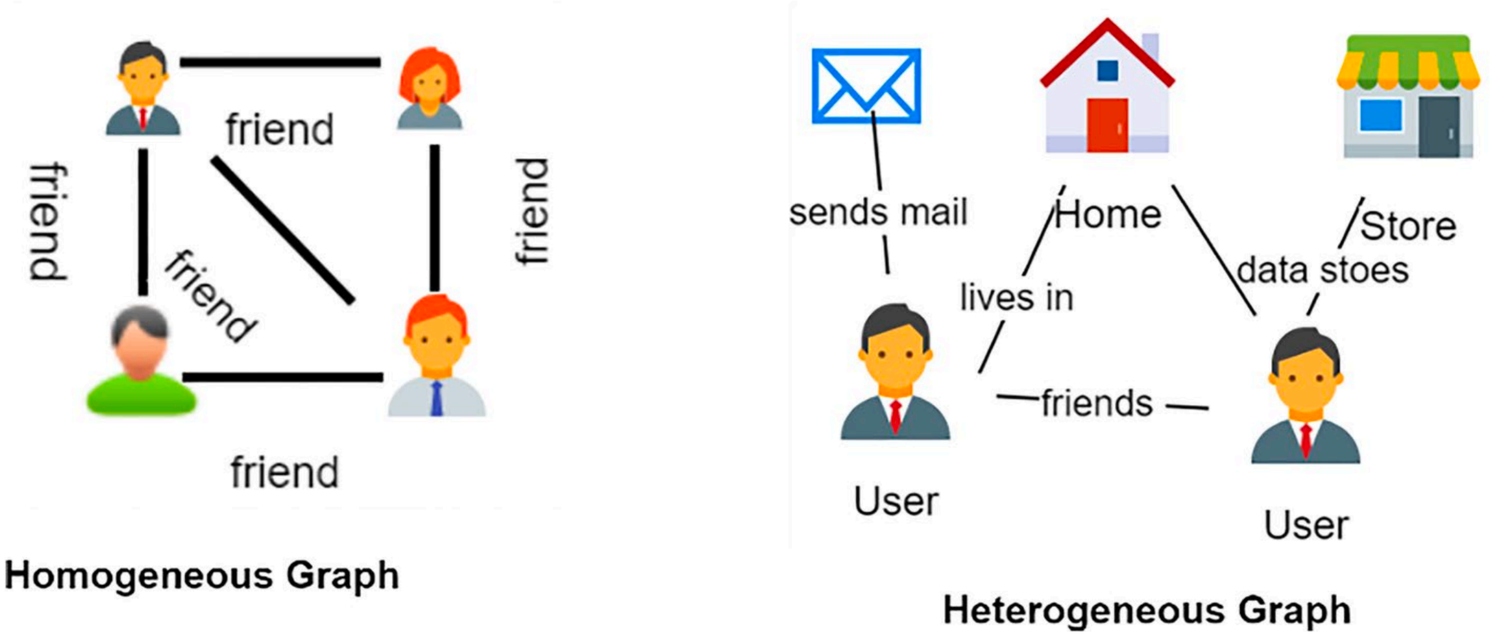
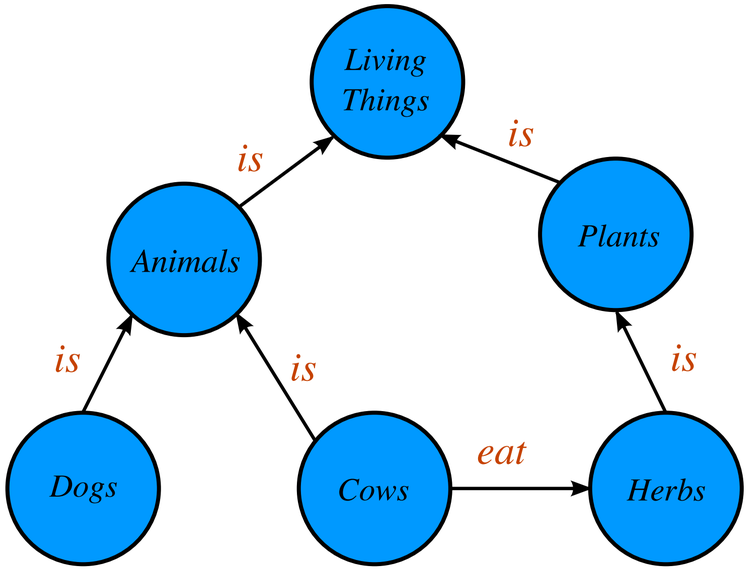



**Fig. 13** Homogeneous (Left), Heterogeneous (Right) Graph

# Graph Taxonomy

## Knowledge Graph

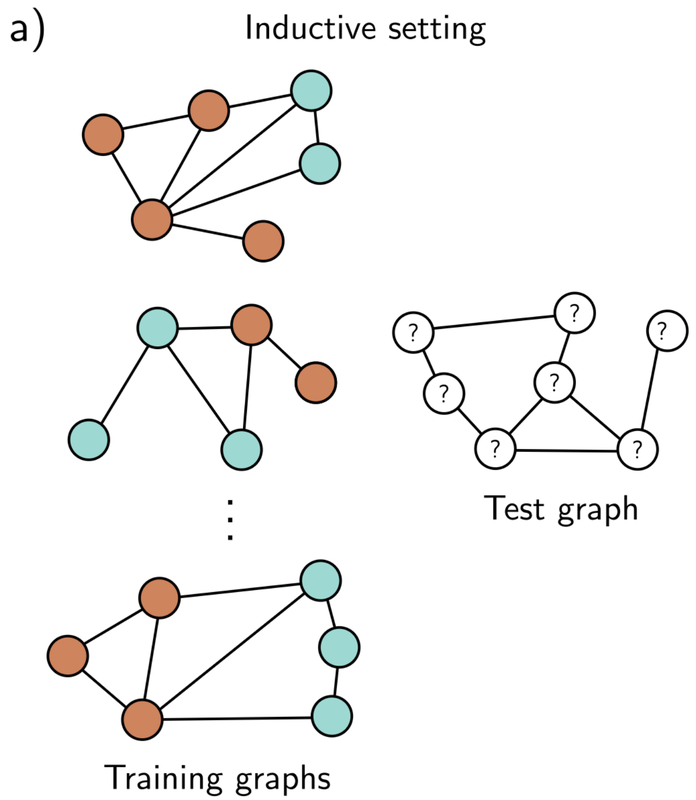
- Array of triples in form of $(h, r, t)$
- $h$: Head
- $r$: Relationship
- $t$: Tail
-

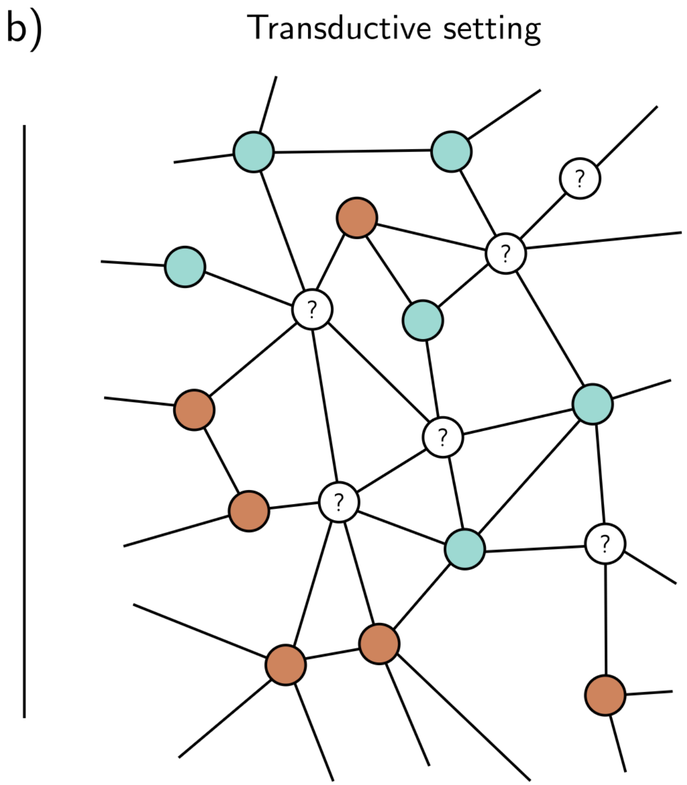


# Graph Taxonomy

## Transductive and Inductive Graph

Representation-> Predict




$$G = \{N, E\}$$

$$h_u^k = UPDATE^k(h_u, N(h_v))$$

$$z_u$$

$$P(C \mid z_u)$$

# Graph Taxonomy

## Graph Learning Task

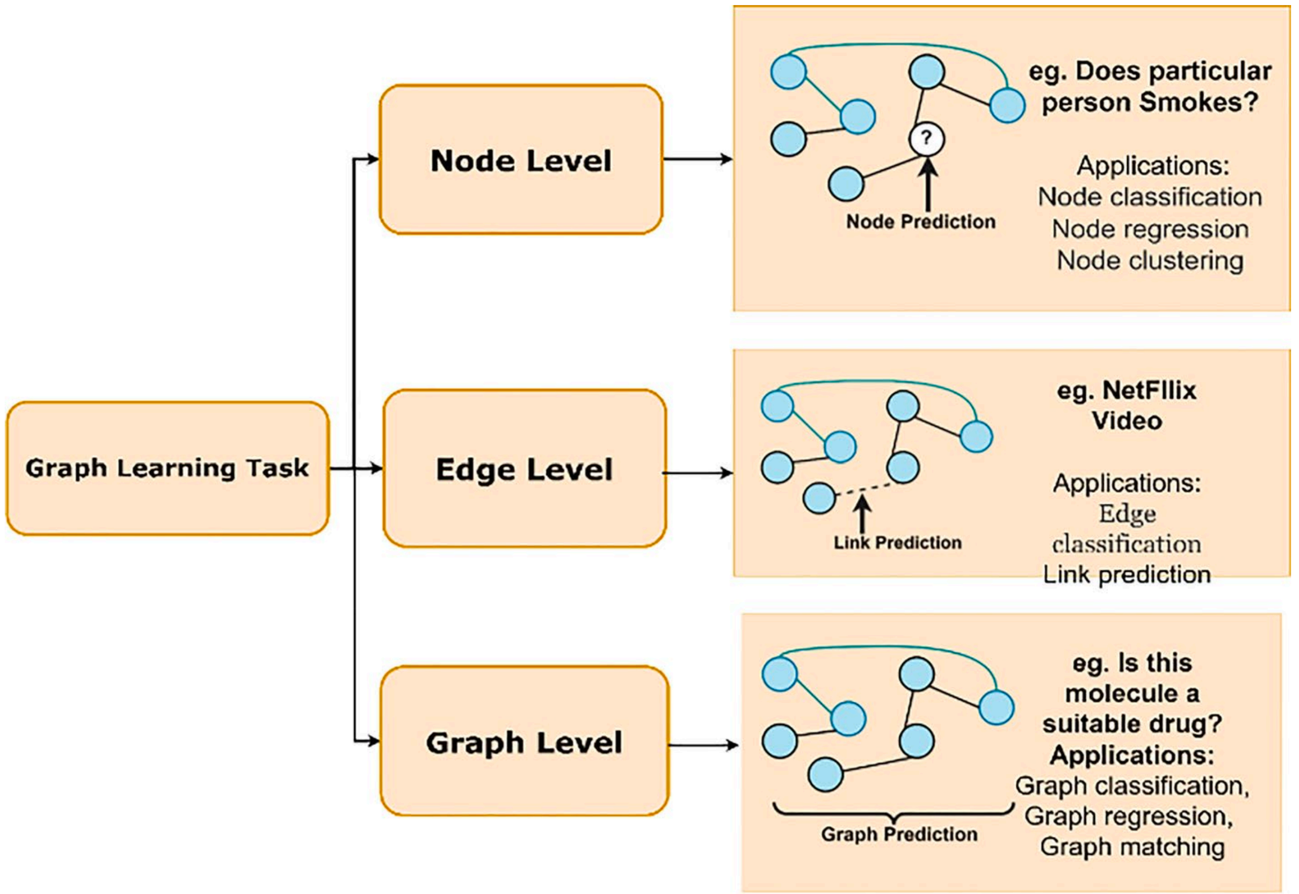



**Fig. 17** Graph Learning Tasks Summary

# GNN Model

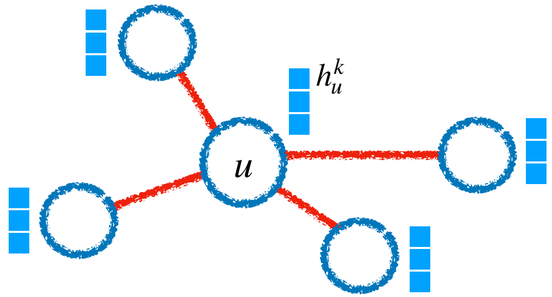
## Graph Convolution Neural Network (GCN)

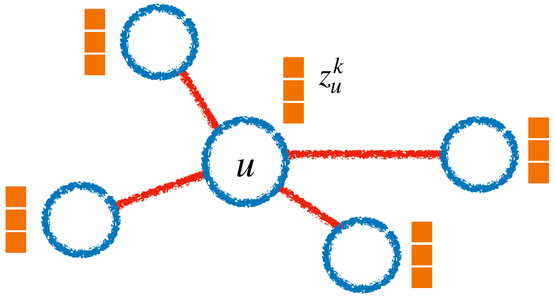
- 1706.02263
  - $$h_u^{(k+1)} = \sigma\left[\beta^k + W^k \cdot h_u^k + W^k \cdot AGG^{(k)}\left(\{h_v^{(k)}, \forall v \in N(u)\}\right)\right]$$

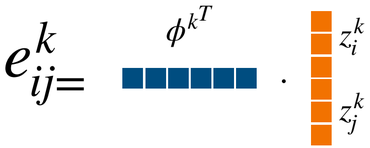
# GNN Model

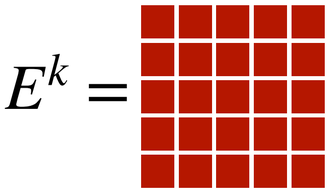
## Graph Attention Network (GAT/GAN)

- Attention mechanism
  - Assigns different attention coefficients to the neighboring nodes.
- Linear transformation: $Z_i^k = W^k \cdot h_i^k$
- Attention Coefficients: $e_{ij}^k = a\left(\phi^{k^T}\begin{pmatrix} z_i^k \\ z_j^k \end{pmatrix}\right)$, $\phi^k$: learnable weight vector
- Softmax: $\alpha_{ui}^l = \dfrac{\exp(e_{ui}^l)}{\sum_{j \in N(u)} \exp(e_{uj}^l)}$
- Aggregation: $h_u^{k+1} = a\left(\sum_{i \in N(u)} \alpha_{ui}^k z_i\right)$



# GNN Model

## GraphSAGE

- GraphSAGE(SAmple and aggreGatE) 1706.02216
- Inductive learning framework
- Learning topological structure of node and distribution of node features
- $h_u^k = \sigma\left[W^k \begin{pmatrix} h_u^{k-1} \\ h_{N(u)}^k \end{pmatrix}\right],\ h_{N(u)}^k = AGG^{(k)}\left(\{h_v^{(k)}, \forall v \in N(u)\}\right)$
- Mean aggregator: Symmetric, Not trainable, $h_u^k = \sigma\left(W \cdot MEAN(\{h_u^{k-1}\} \cup \{k_v^{k-1}, \forall v \in N(u)\})\right)$
- LSTM aggregator: Not Symmetric, Trainable, random shuffle
- Pooling aggregator: Symmetric, Trainable, $AGG_k^{pool} = \max\left(\sigma(W_{pool} h_v^k + b), \forall v \in N(u)\right)$

# GNN Model
## Performance

**Table 6** Performance metrics of different models with different datasets [33, 47, 48]

| Dataset | Model | | | | | |
|---|---|---|---|---|---|---|
| | GCN | GAT | GraphSAGE | | | |
| | | | GraphSAGE-Simple | GraphSAGE—Mean | GraphSAGE—LSTM | GraphSAGE—Pooling |
| CORA | 81.5 (4 s) | 83 | 76.8 | 78.7 | 79.7 | 80.7 |
| Citeseer | 70.3 (7 s) | 72.5 | 74.2 | 77.8 | 78.8 | 79.8 |
| Pubmed | 79.0 (38 s) | 79 | | | | |

# GNN Applications

Molecular graphs
Social networks
Citation networks
Computer vision

Document Classification
Text Generation
Question Answering
Sentiment Analysis

Image Segmentation
Object Detection
Scene Understanding

Protein-Protein Interaction
Genomic Sequence
Drug Discovery

# Dataset

- Citation Networks
  - Pubmed
  - Cora
  - Citeseer
  - NELL
- Social Networks
  - Reddit
  - Ciao
  - Epinions
  - Microblogs

# Python Libraries

- PyTorch Geometric
- Deep Graph Library (DGL)
- GraphVite
- Plato
- Paddle graph learning