图神经网络

Graph Neural Networks

张 瑞

rui.zhang@cern.ch

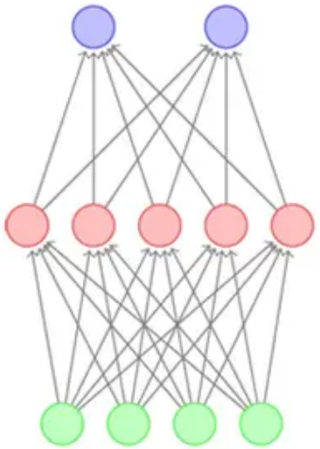
1st Machine Learning and Quantum Computing Winter School

2025.1.14

目录

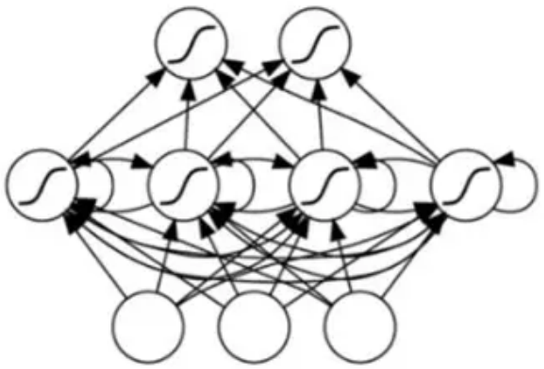
GNN之前的经典模型

- 早年深度学习一直被几大经典模型统治，如RNN、CNN等
- RNN擅长处理不固定长度的输入

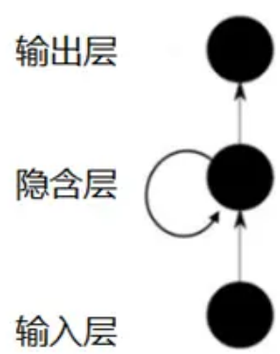


经典神经网络

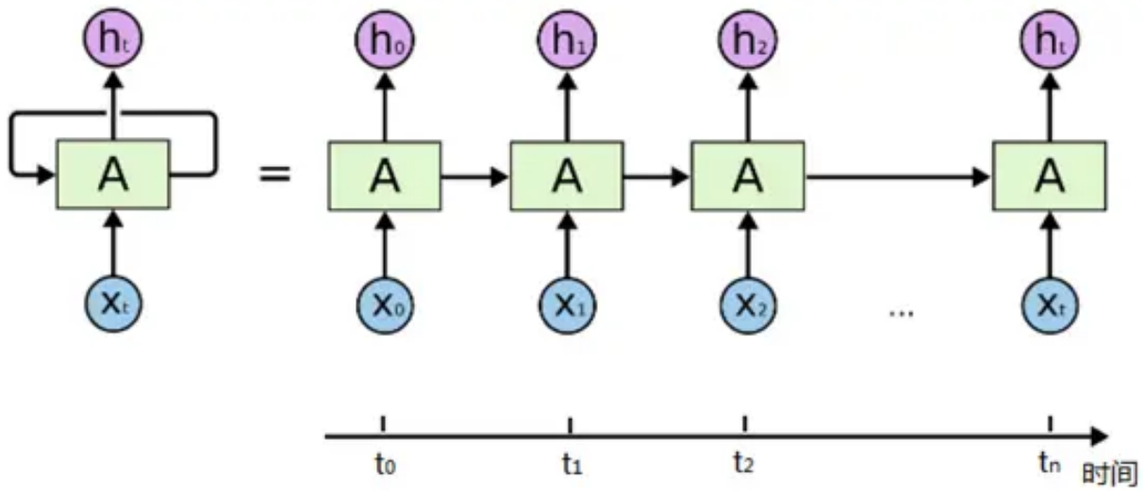
简化图



RNN 网络结构



RNN 简化图

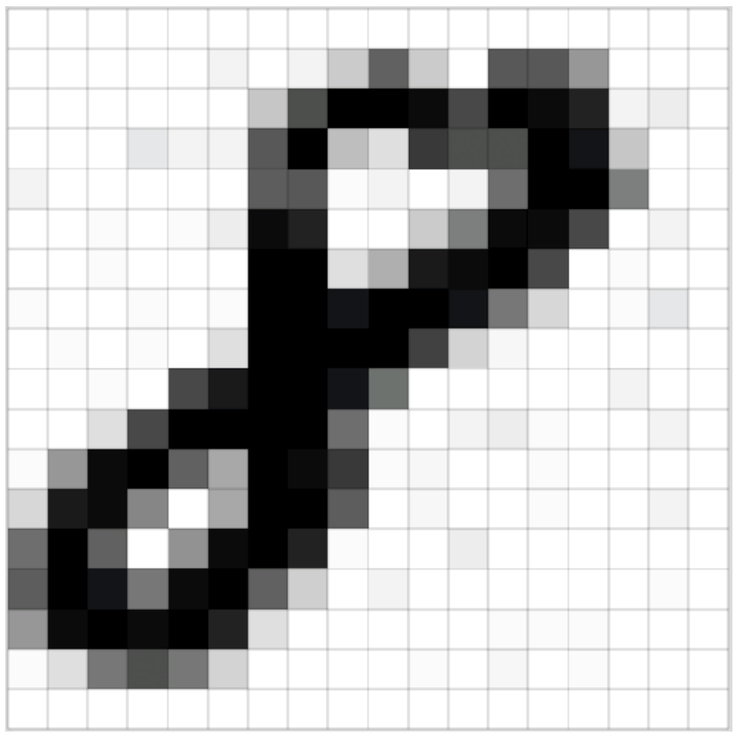


参考：https://www.cnblogs.com/hd-zg/p/8724661.html

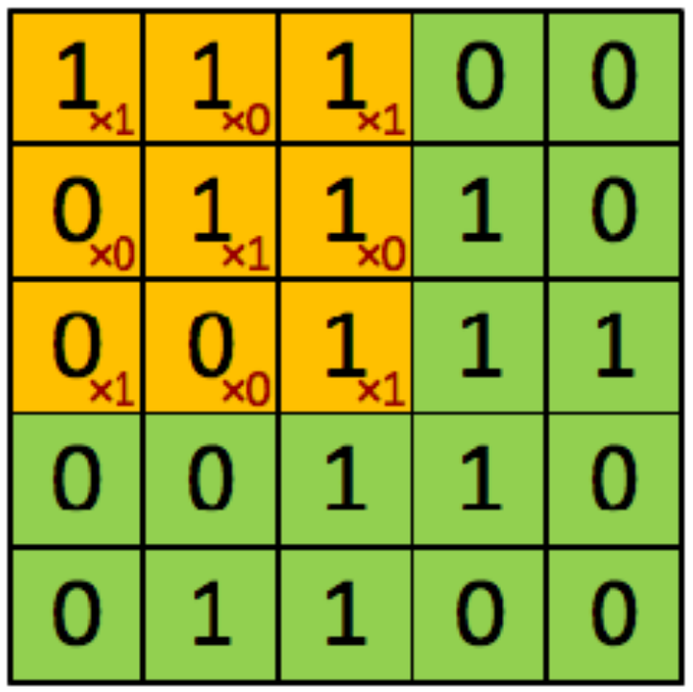
GNN之前的经典模型

- CNN擅长分析图片

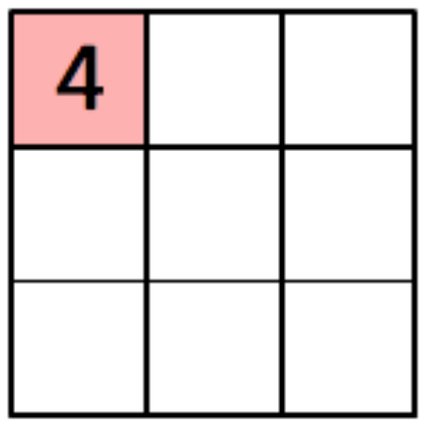
卷积核kernel = $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$



卷积运



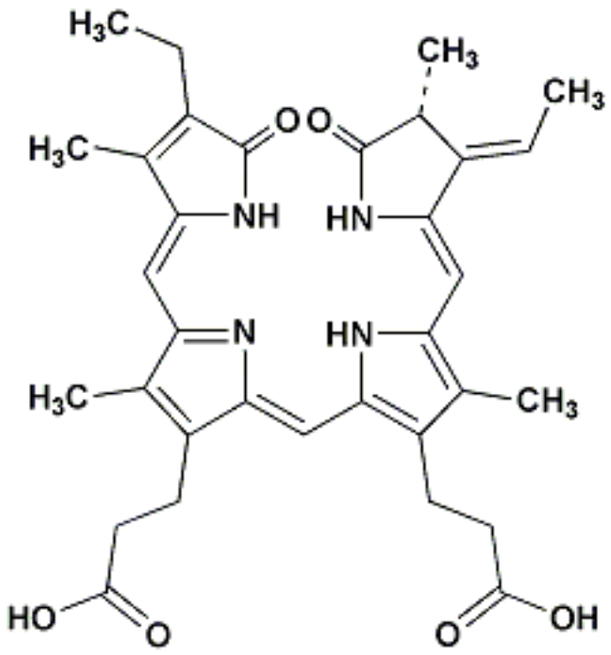
Image

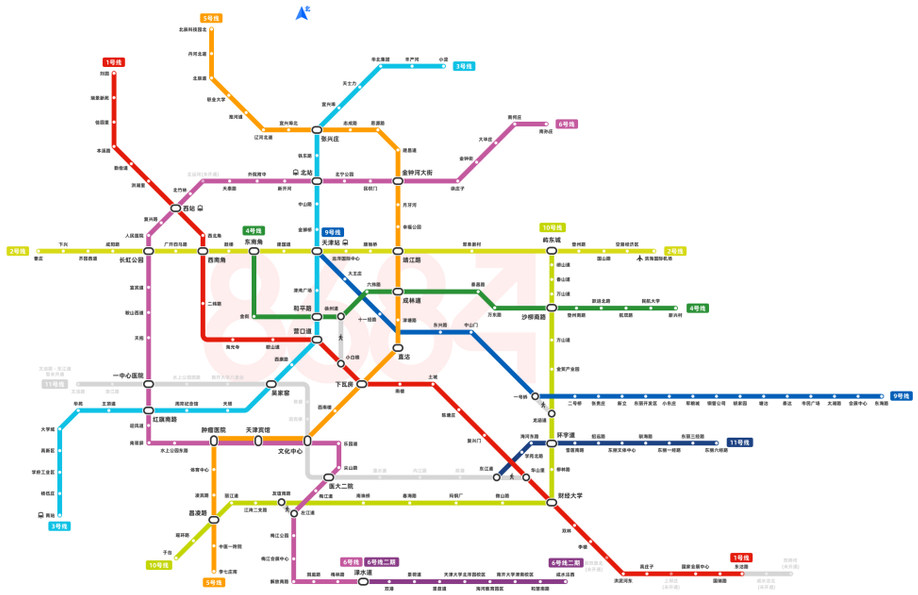


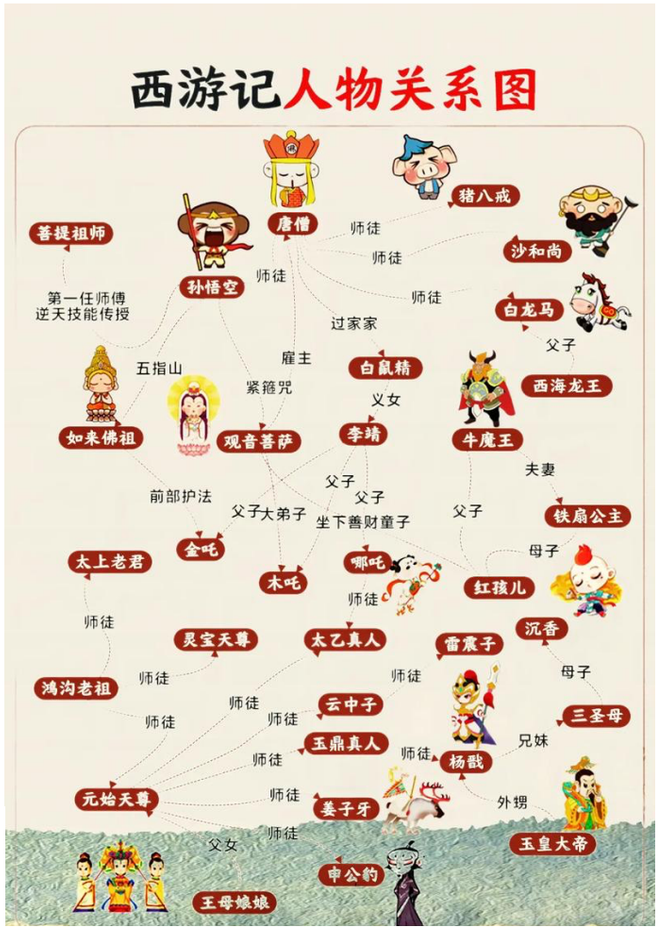
Convolved Feature

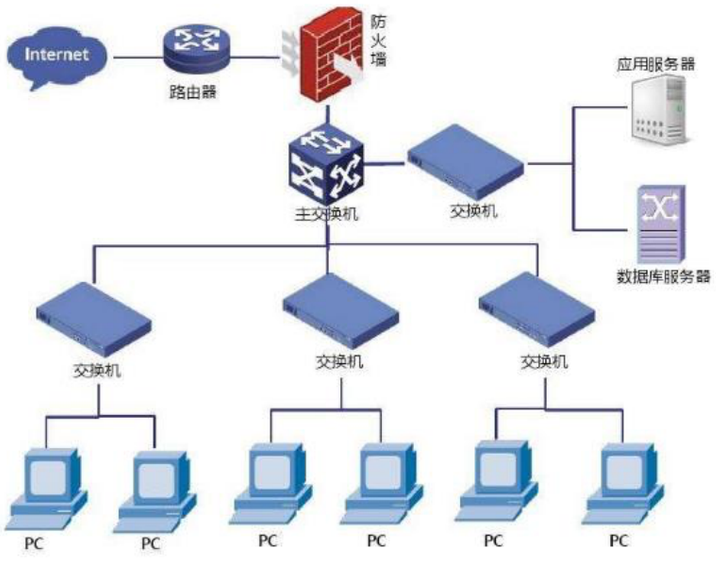
图数据结构

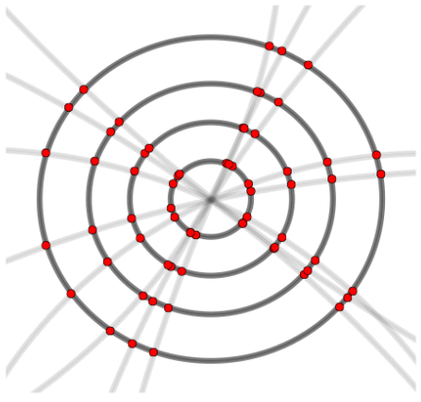
- 图数据结构在现实世界存在更广泛

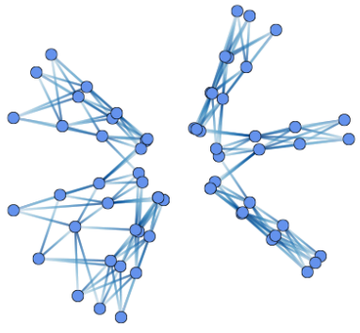




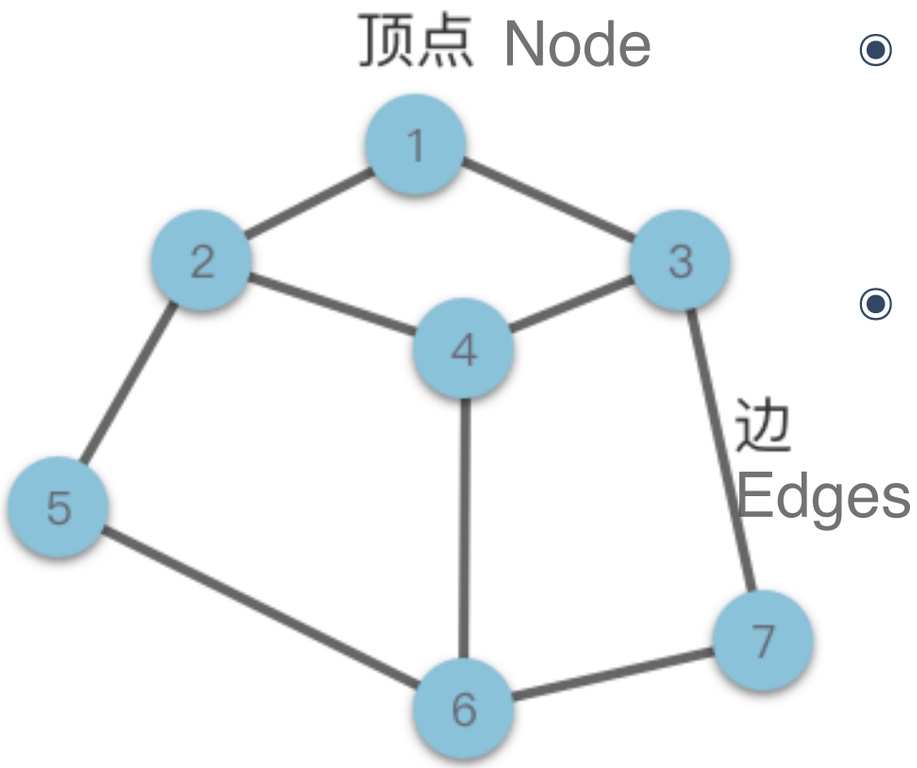








如何处理图数据结构

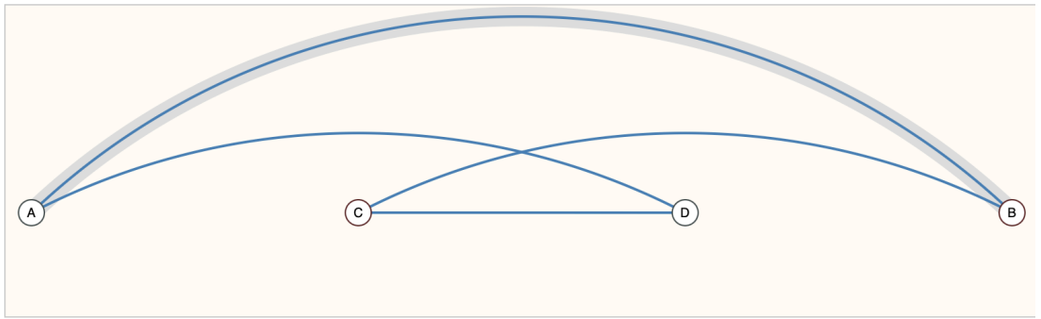


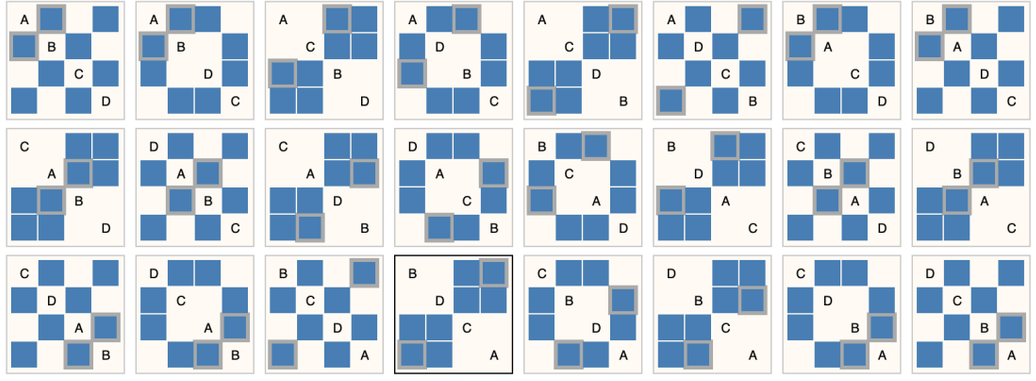
- 相对于images，sentences，graphs很不规则
  - 总顶点数、每个顶点上的边数不确定
- 如何编码一个图?

一篇可以互动的GNN简介：https://distill.pub/2021/gnn-intro/

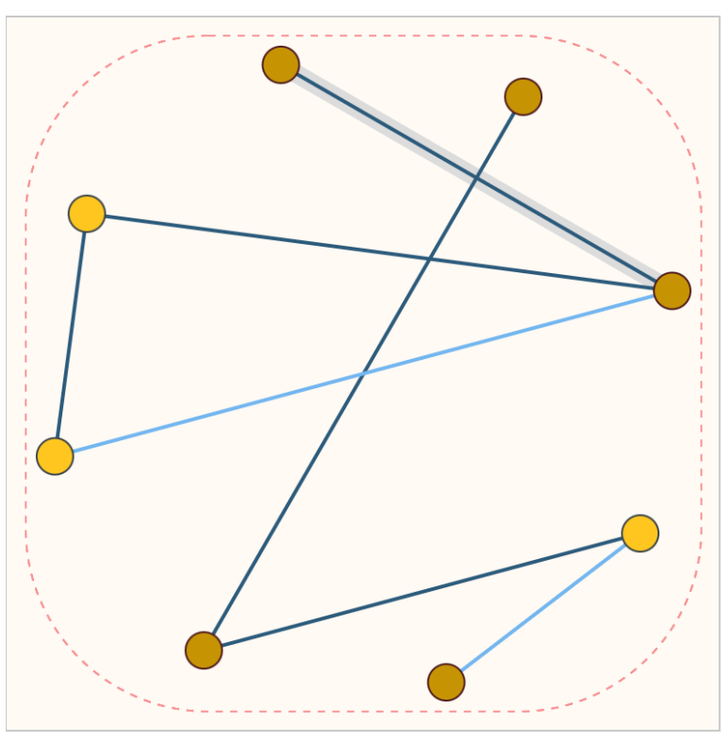
图的编码

- 邻接矩阵(adjacency matrices)法
  - 但邻接矩阵不唯一，如下图示例





- 目前比较经济/主流的方式是
  - 现将所有notes编号
  - 按照编号顺序保存每个node上的值
  - 保存每个edge上的值以及定义该edge的两端的note编号



Nodes

```
[0 , 1 , 1 , 0 , 0 , 1 , 1 , 1 ]
```

Edges

```
[2 , 1 , 1 , 1 , 2 , 1 , 1 ]
```

Adjacency List

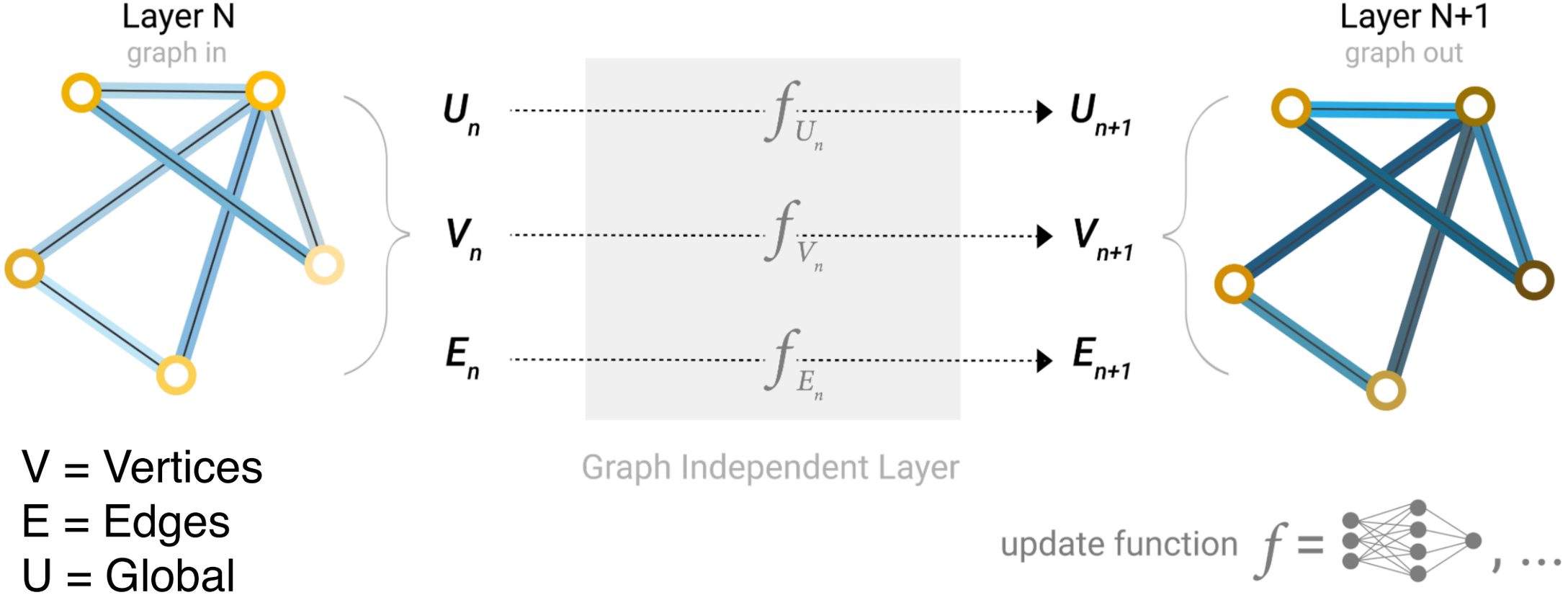
```
[[1, 0] , [2, 0] , [4, 3] , [6, 2] ,
 [7, 3] , [7, 4] , [7, 5] ]
```

Global

```
0
```

图信息的处理

- 如何处理图的信息?
  - 参考卷积操作
  - 信息既可以从边流向顶点，又可以从顶点流向边
  - 最终的输出可以是指定的边的特征，或者指定的顶点的特征，或者全图的特征
- 最简单的GNN



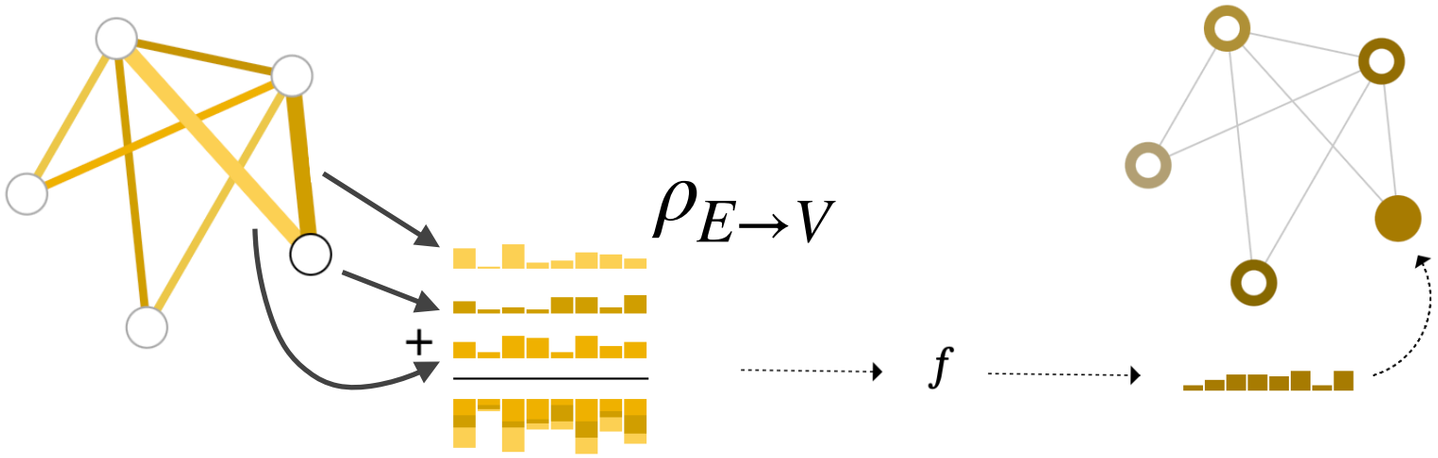
V = Vertices
E = Edges
U = Global

GNN做prediction

- 当叠加了很多graph layer之后，终于得到了最后一步的图
- 如何从图中获取最终信息呢?

Message passing

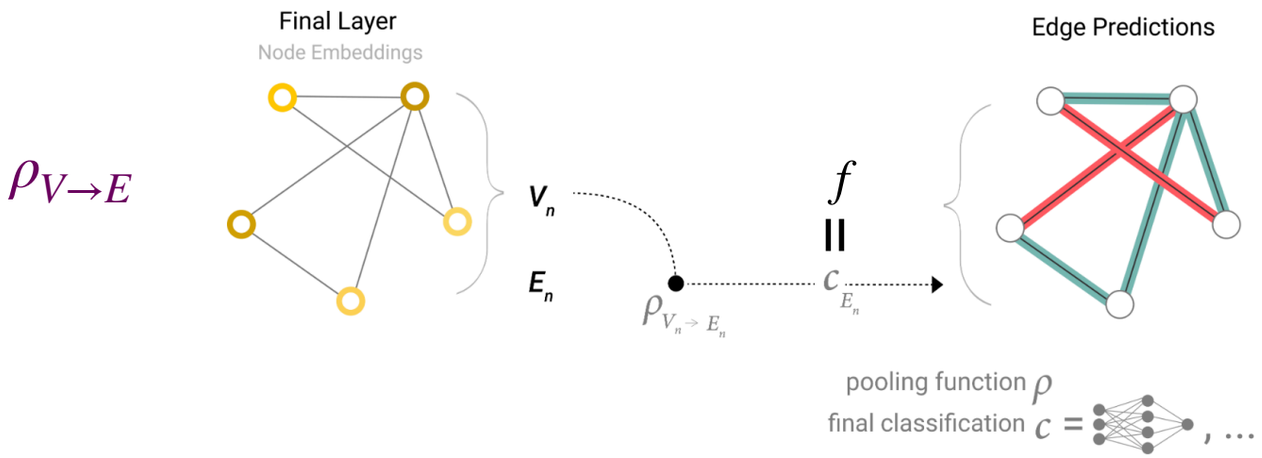
1. For each node or edge, gather all the neighbouring embeddings and aggregate using ρ, e.g. sum
2. All pooled messages are passed through an update function f, usually a learned neural network.

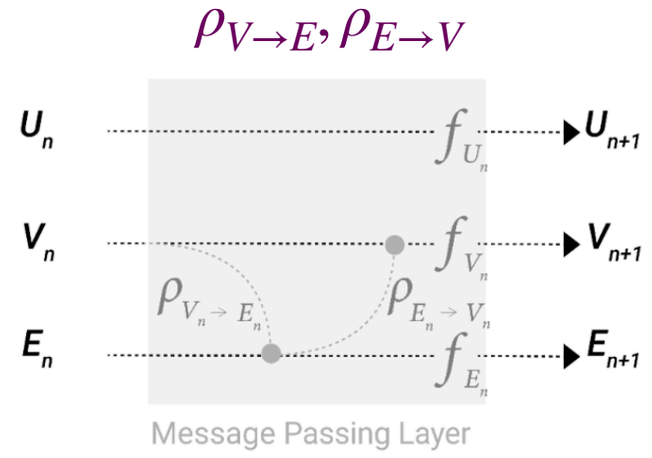


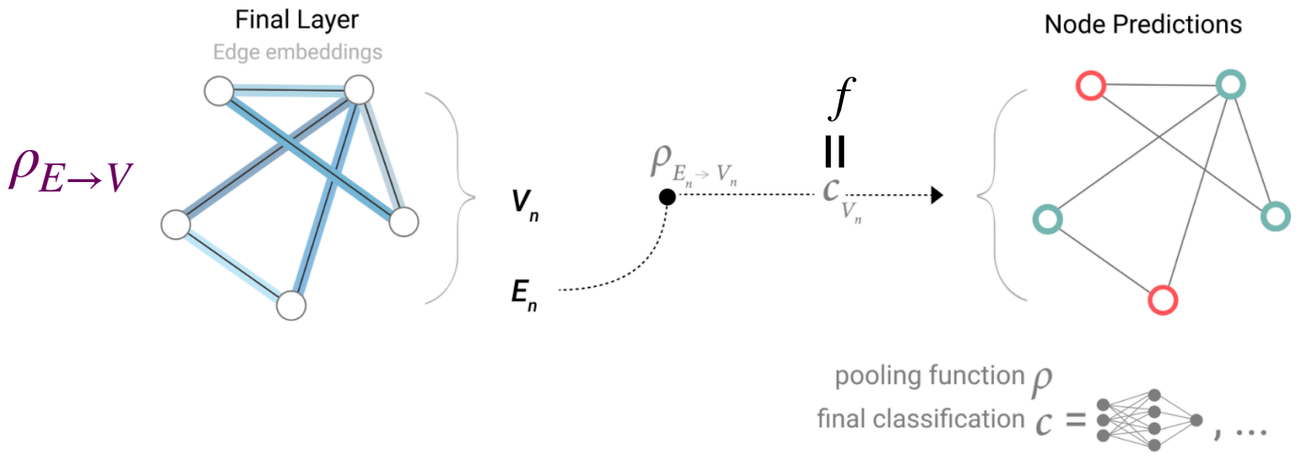
Aggregate information
from adjacent edges

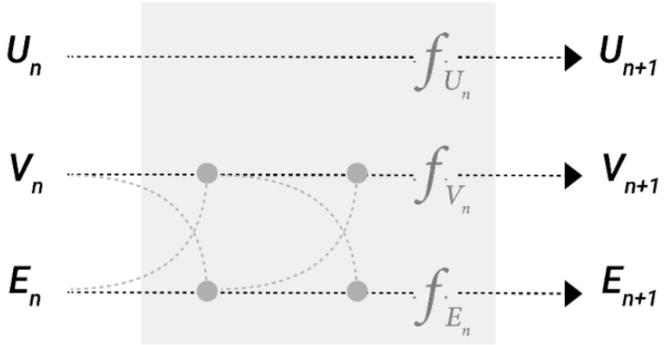
V = Vertices
E = Edges
U = Global

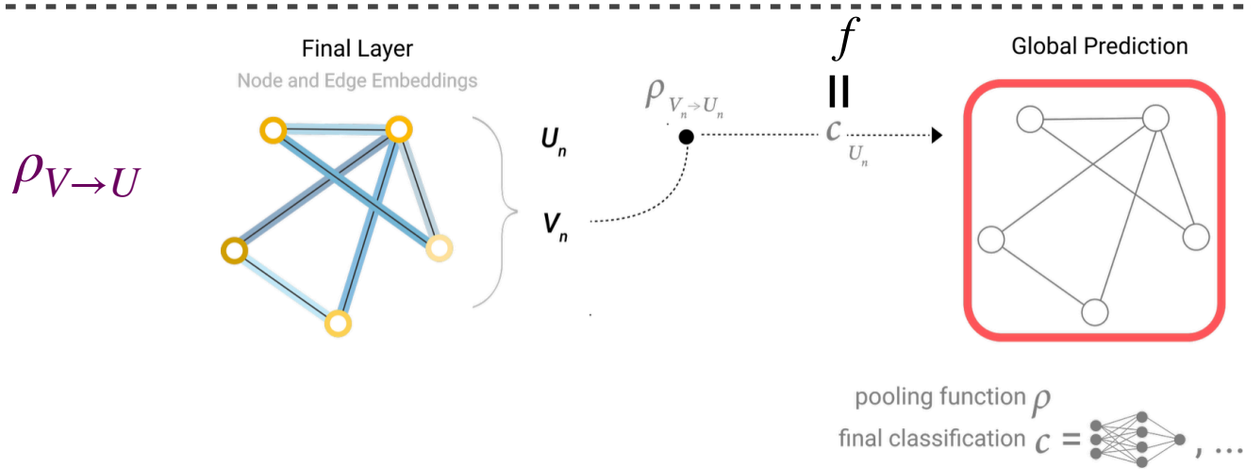
Message passing

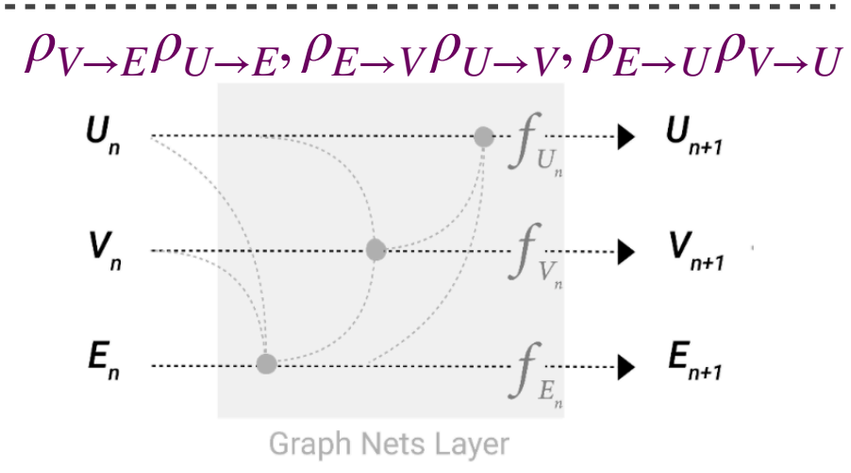




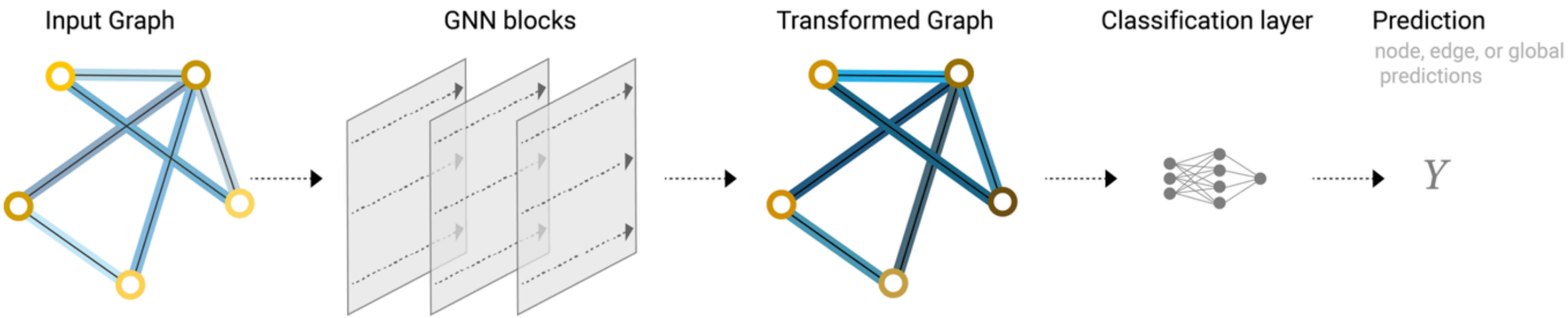


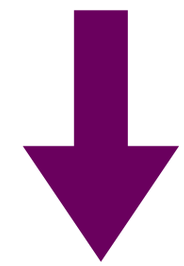






处理图数据的过程





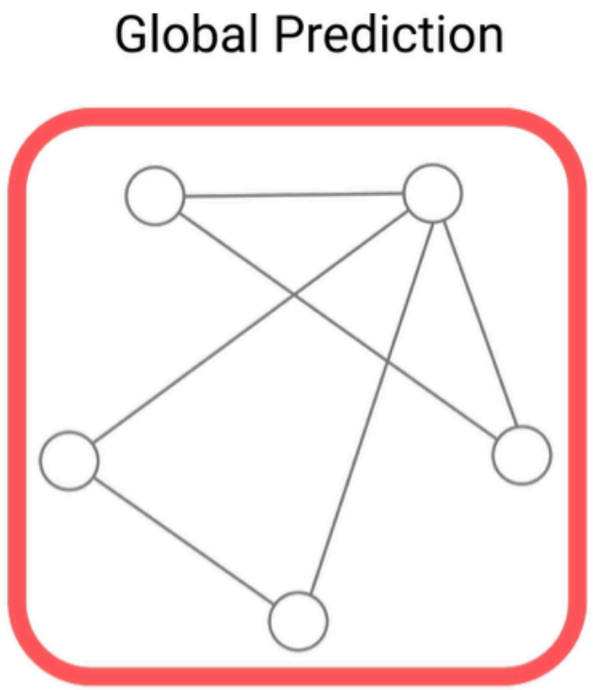
GNN

流行的GNN架构

- 不同GNN架构对message passing处理不同
- GCN (Graph Convolutional Network)：
  - 通过图卷积操作聚合邻居节点的信息。使用“归一化的邻接矩阵”来传播节点特征
  - 简单高效，但对所有邻居节点赋予相同的权重，忽略了节点间的重要性差异
- GAT (Graph Attention Network)
  - 引入注意力机制，计算节点间的注意力系数
  - 能动态调整邻居节点的重要性，适用于异质图，但计算复杂度较高
- GraphSAGE (Graph Sample and Aggregate)
  - 通过采样固定数量的邻居节点进行聚合
  - 适用于大规模图，通过采样减少计算量，但可能丢失部分信息
- ...

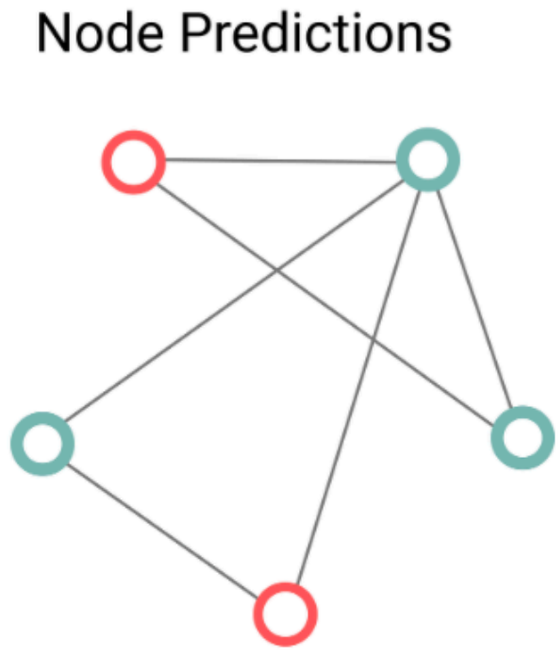
目录

Graph classification



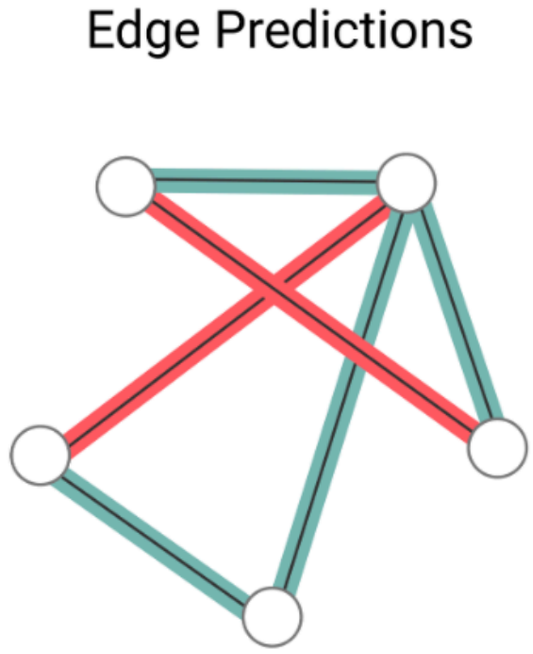
- Jet Classification (jet constitutes are nodes)
  - Classify jets are quark or gluon jets
  - Classify jets are b-jets or not
- Event Classification (particles seen in detector are nodes)
  - Classify events are signal or background

Node classification



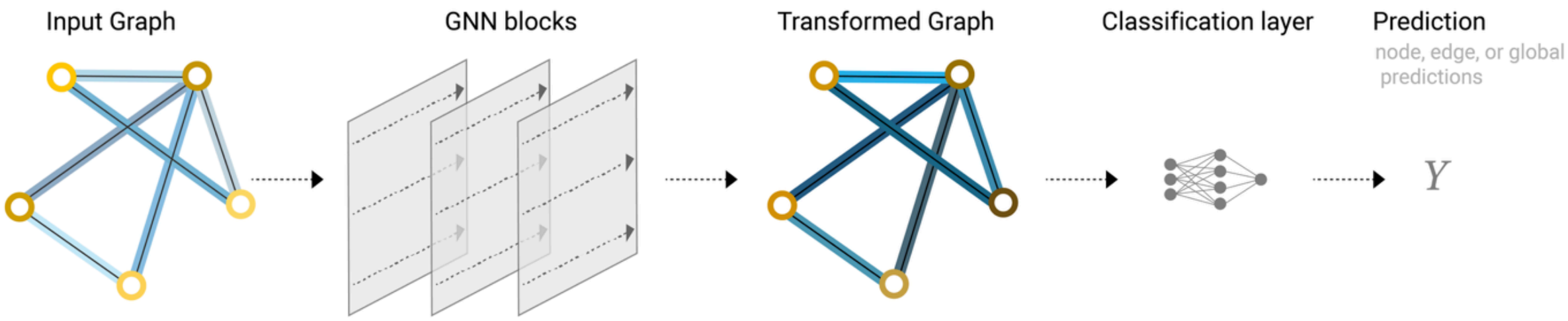
- Jet vertex tagger (tagger to classify jets from pile-up or hard vertex)
  - To each track (node) assign a probability that it is from pile-up

Edge classification



- Charged Particle Tracking
  - Classify connections between two hits (edge) are from a track or not
- Secondary Vertex Reconstruction
  - Classify particles within a jet are coming from a secondary vertex

总结



- GNN 的核心目标是通过**Message Passing**，将图中每个节点的信息与其邻居节点的信息进行聚合，从而更新节点的表示。
- 能够灵活地捕获局部和全局的图结构
- 计算复杂度高，尤其是对于大规模图