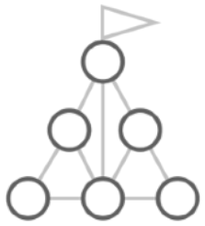
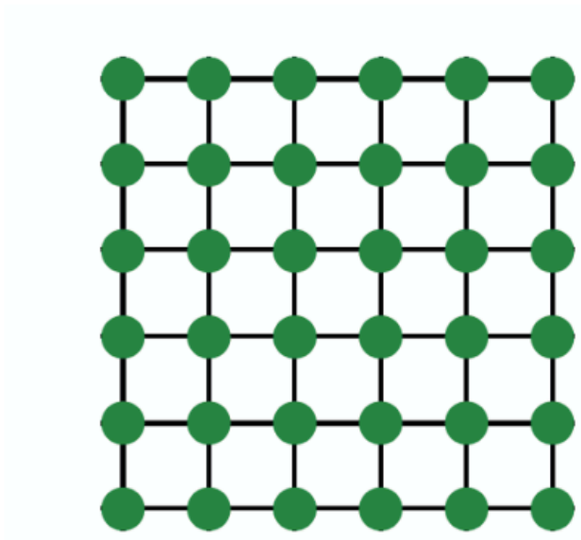


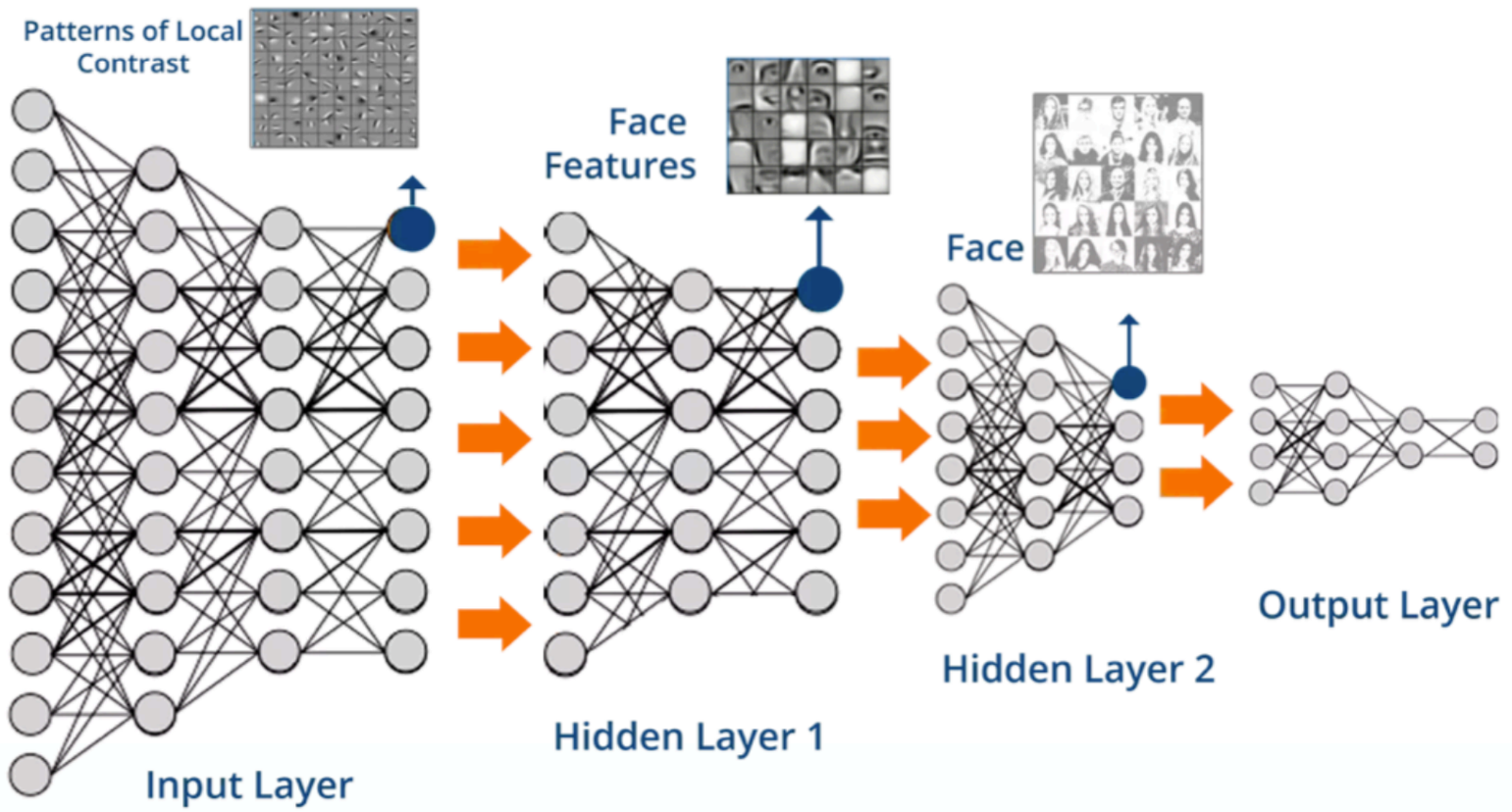
Graph Neural Network

Itthi Chatnuntawech

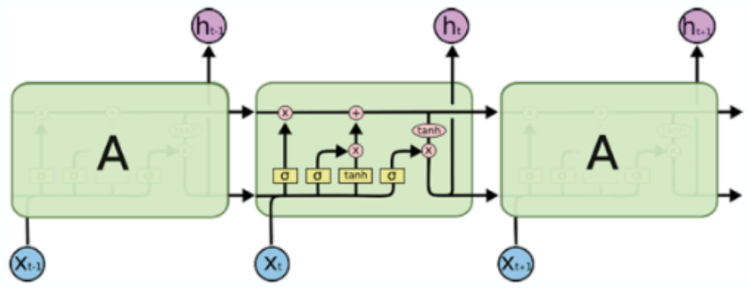
Graphs



Image



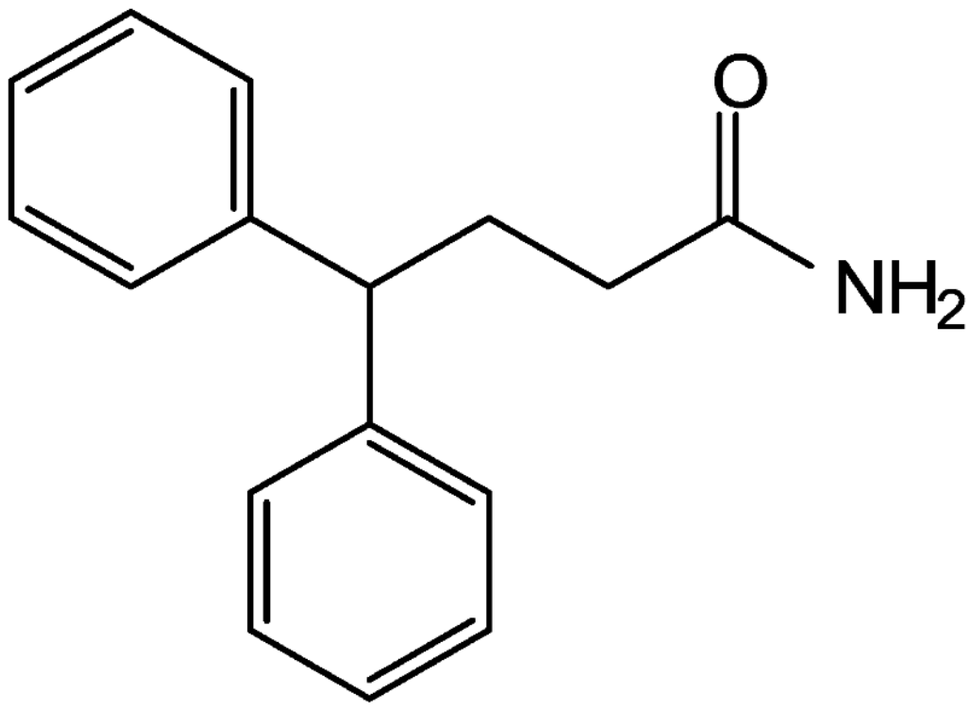
Sequence



Graphs

3,3-Diphenylpropylamine

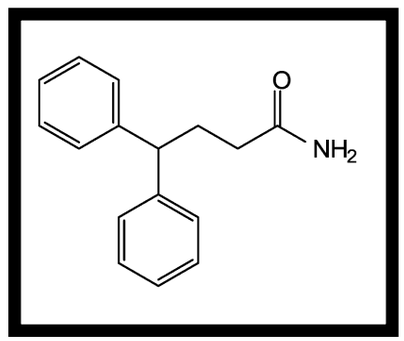
$C_{15}H_{17}N$

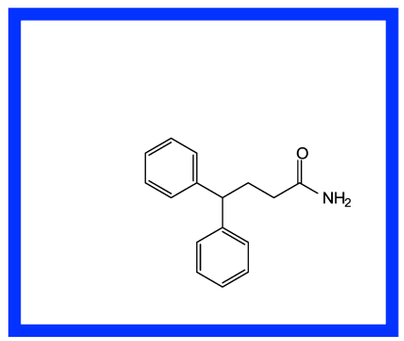


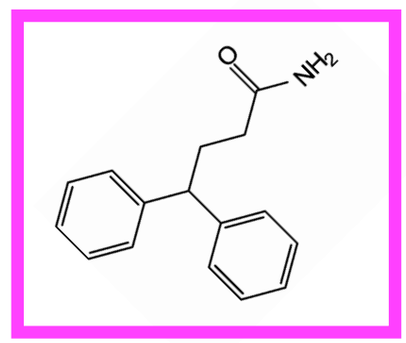
How should we process this type of data?

Can we treat it as image and use a CNN to analyze it?

If we zoom / translate / shift the molecule, will our CNN understand that? More importantly, rotate?







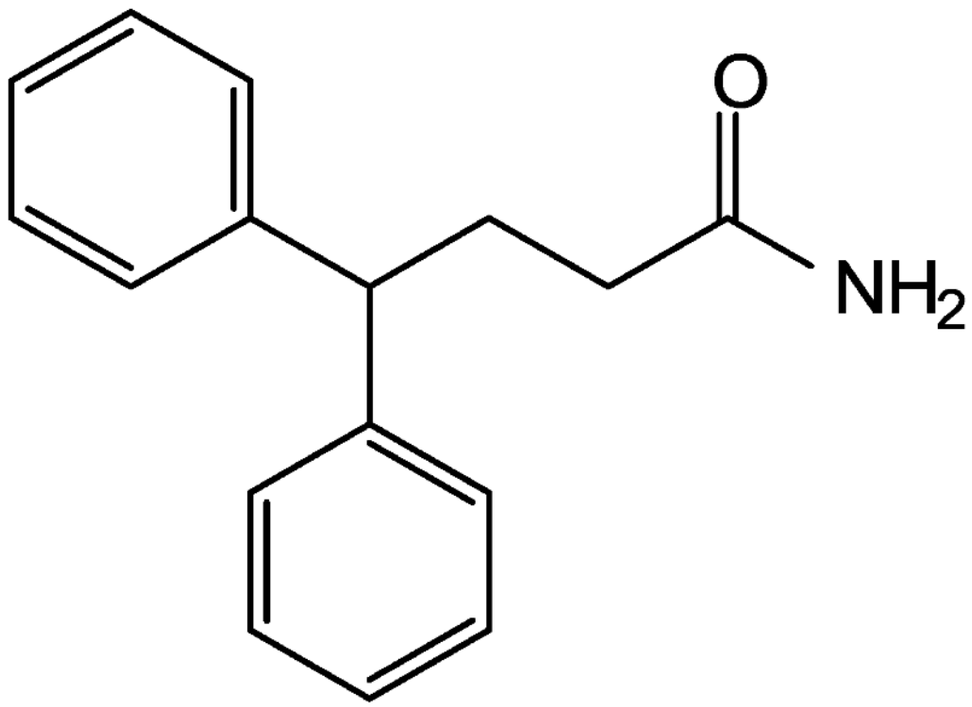
Vogt, Martin, Anne Mai Wassermann, and Jürgen Bajorath. "Application of information—Theoretic concepts in chemoinformatics." Information 1.2 (2010): 60-73.

keywords: Simplified molecular-input line-entry system (SMILES)

Graphs

3,3-Diphenylpropylamine

$C_{15}H_{17}N$



How should we process this type of data?

Can we treat it as a string of characters and use a RNN to analyze it?

If we swap the order of the characters, will our model understand it?

CCC…CHH…HN

NCCC…CHH…HN

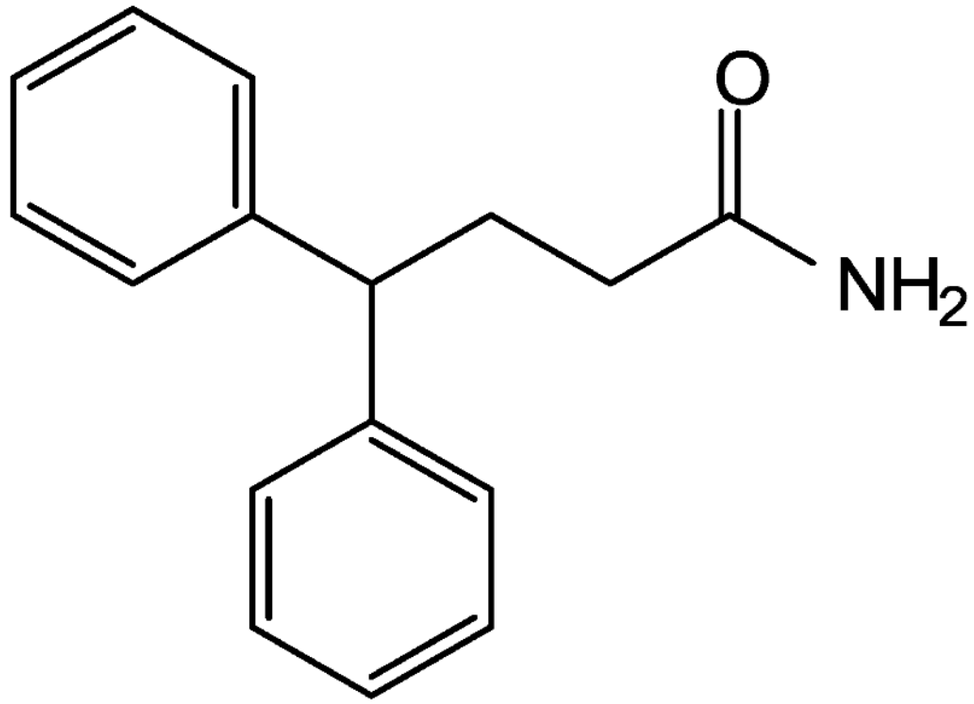
NCHCH…CHHH

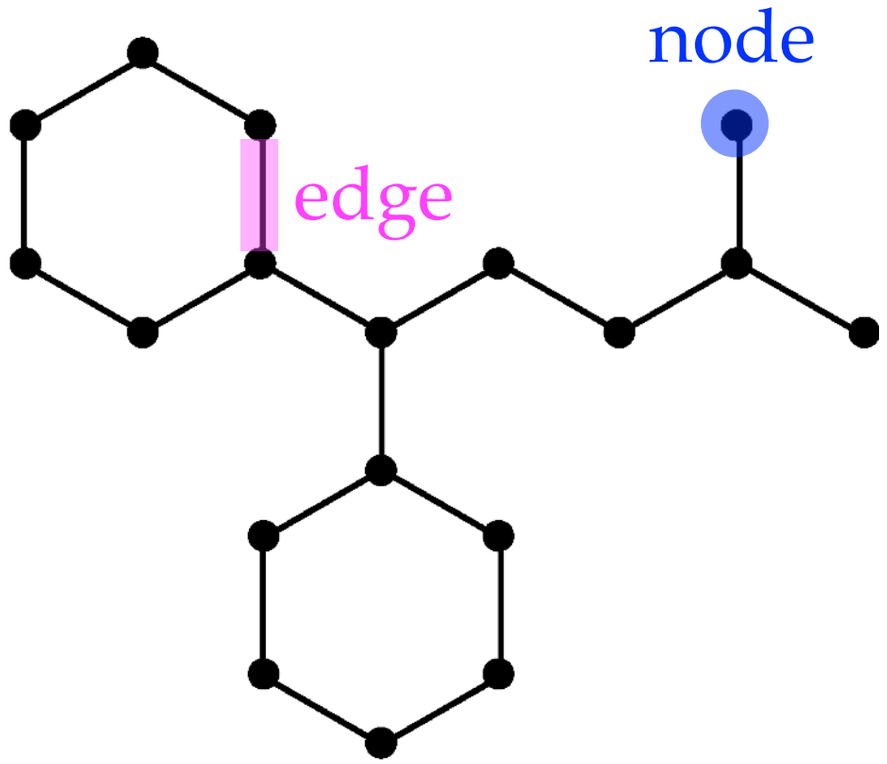
Vogt, Martin, Anne Mai Wassermann, and Jürgen Bajorath. "Application of information —Theoretic concepts in chemoinformatics." Information 1.2 (2010): 60-73.

Graphs

3,3-Diphenylpropylamine

$C_{15}H_{17}N$





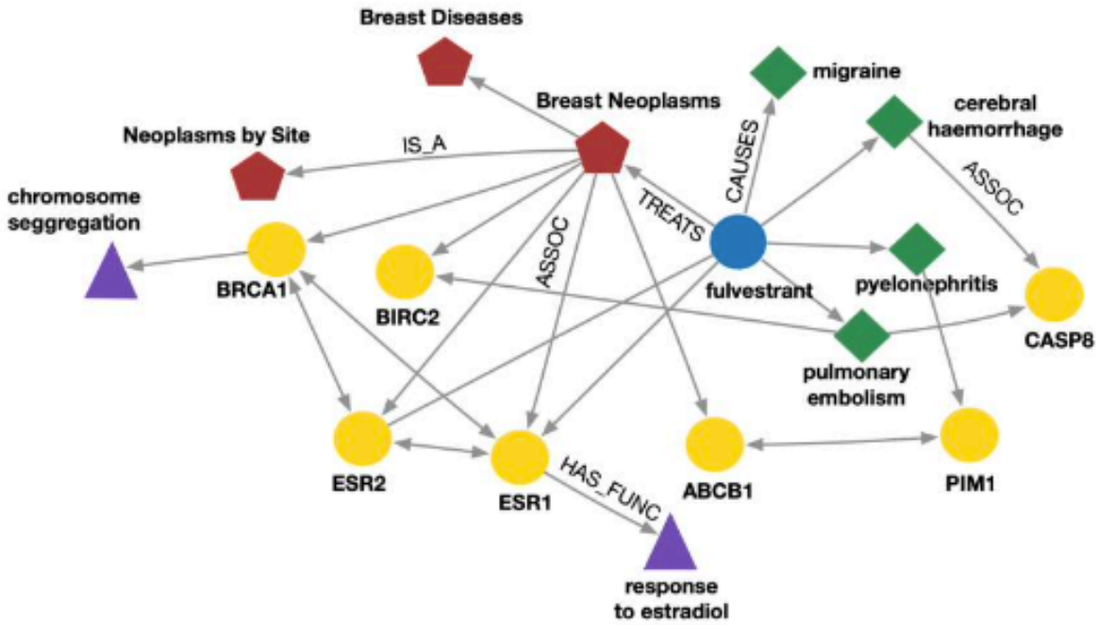
Graph

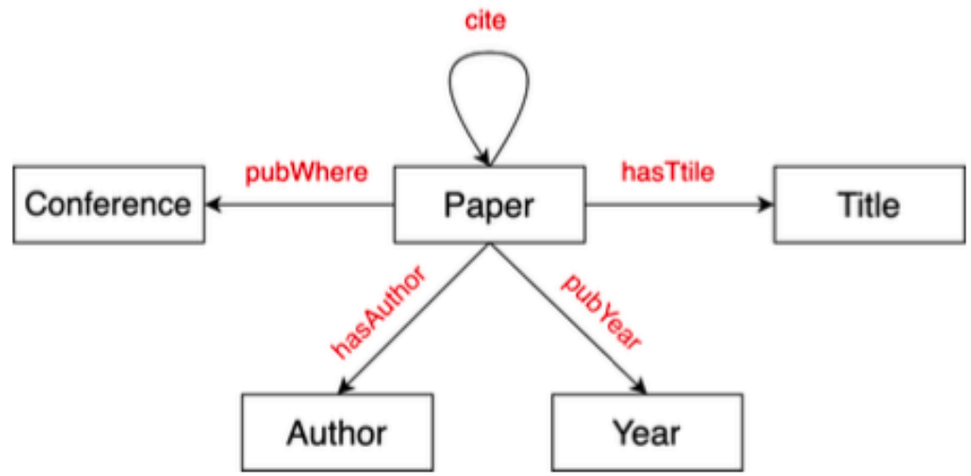
nodes: atoms / molecules

edges: chemical bonds

Vogt, Martin, Anne Mai Wassermann, and Jürgen Bajorath. "Application of information —Theoretic concepts in chemoinformatics." Information 1.2 (2010): 60-73.

Examples: Knowledge/Academic Graphs





Biomedical Knowledge Graphs

Example node: Migraine
Example edge: (fulvestrant, Treats, Breast Neoplasms)
Example node type: Protein
Example edge type (relation): Causes

Academic Graphs

Example node: ICML
Example edge: (GraphSAGE, NeurIPS)
Example node type: Author
Example edge type (relation): pubYear

Example: Physical Simulation

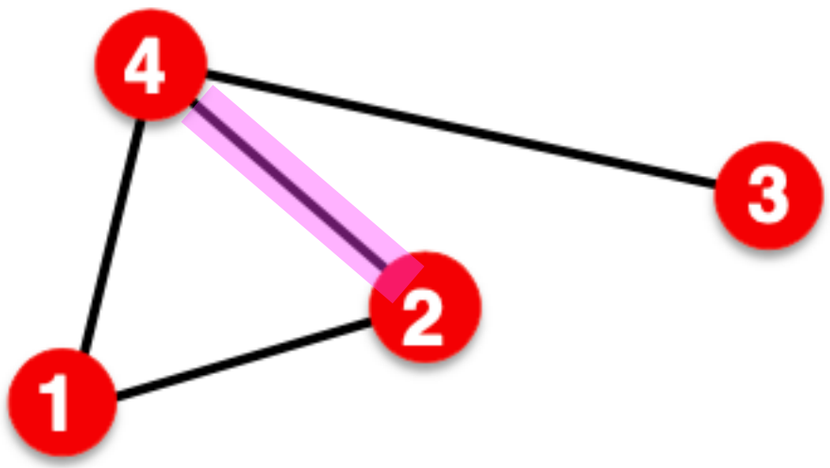
nodes: particles

edges: interaction between particles

Sanchez-Gonzalez, Alvaro, et al. "Learning to simulate complex physics with graph networks." International Conference on Machine Learning. PMLR, 2020.

Graphs: Representation

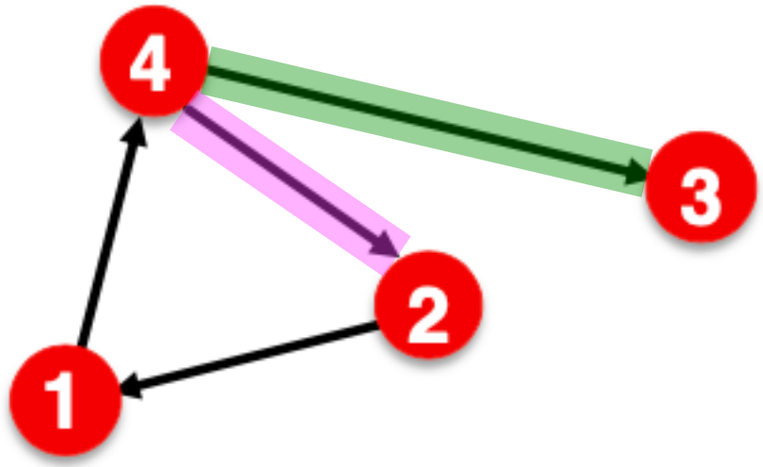
Undirected graph



Adjacency matrix, A

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array}$$

Directed graph



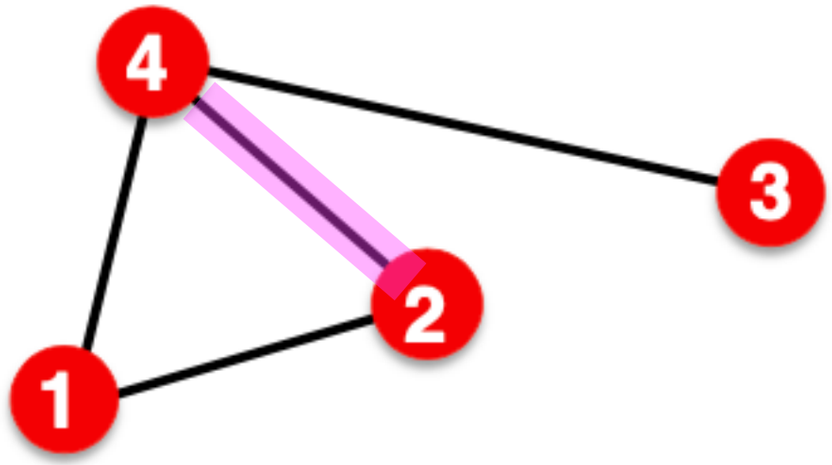
Adjacency matrix, A

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array}$$

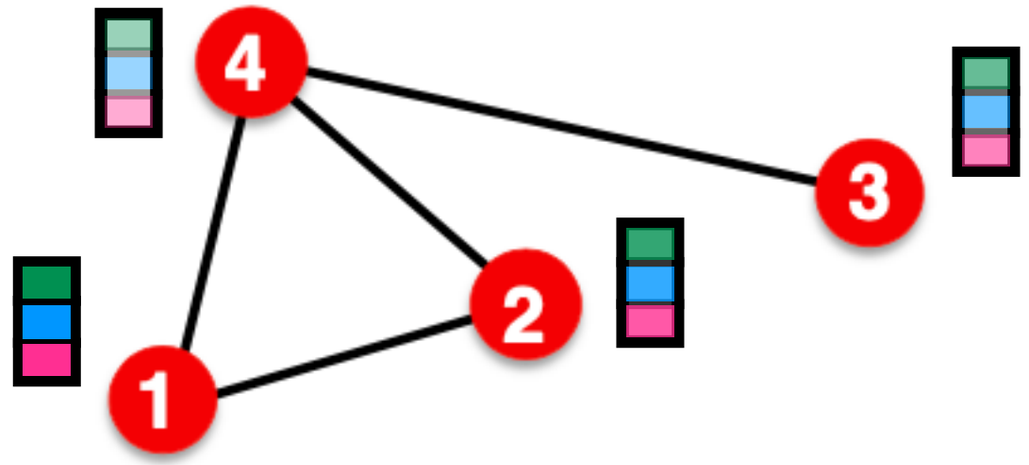
source

destination

Graphs: Representation



Each node has a feature vector



Adjacency matrix, A

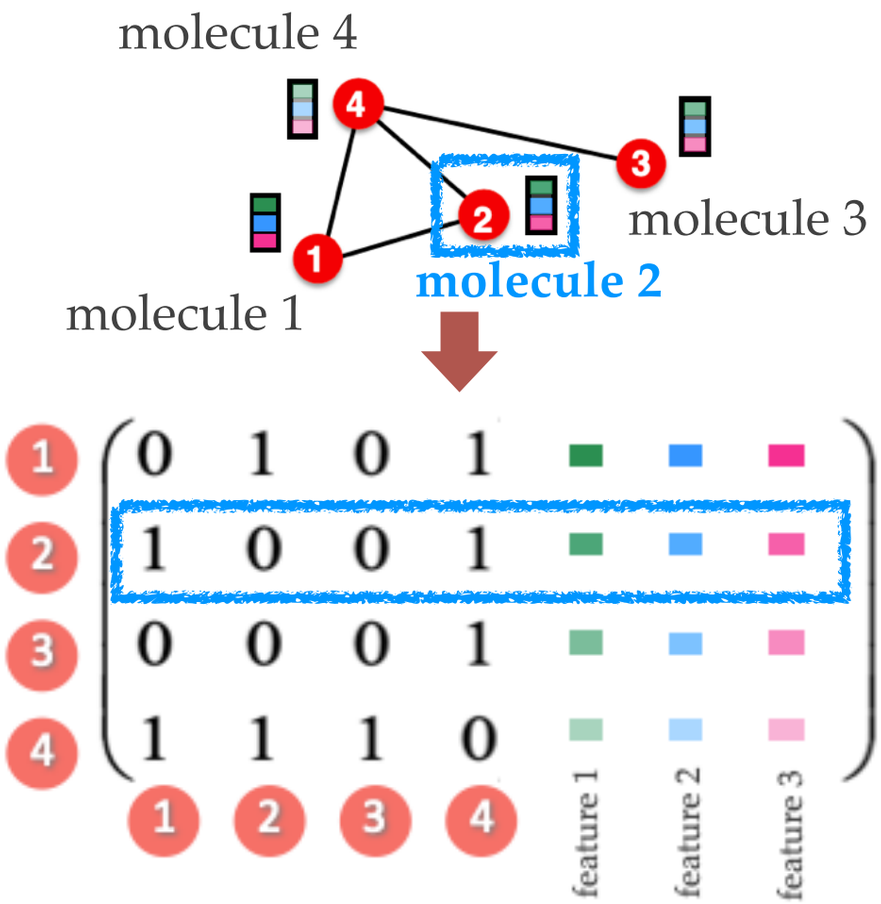
$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Extended adjacency matrix, A

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

feature 1 feature 2 feature 3

Node-Level Classification

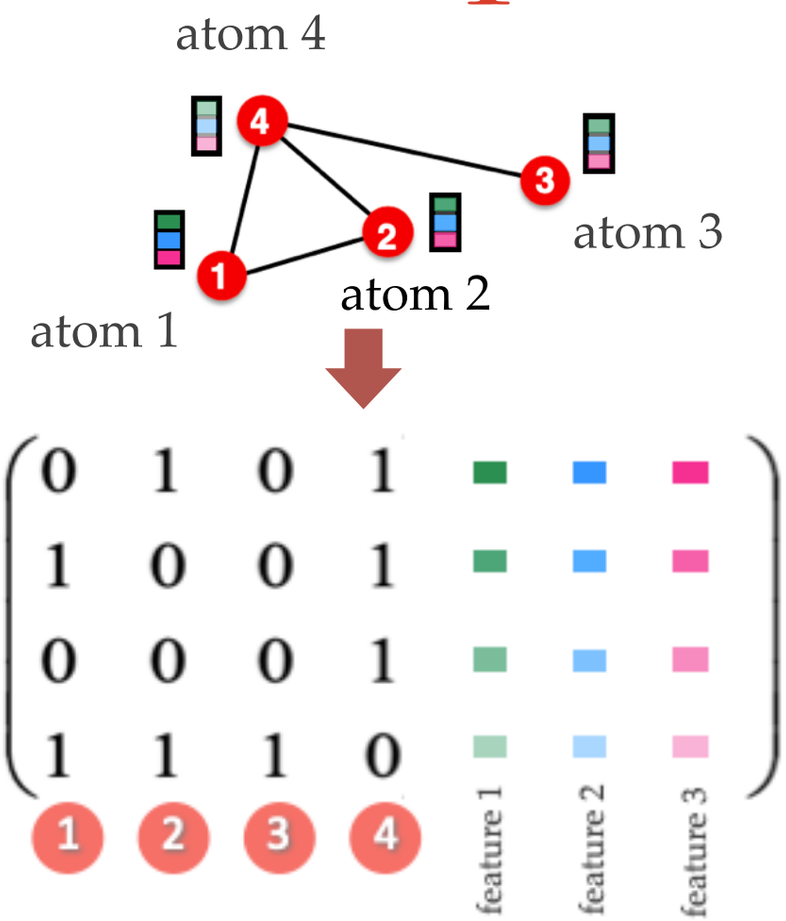


Example: Determine the class of molecule 2 (e.g., toxic vs non-toxic)

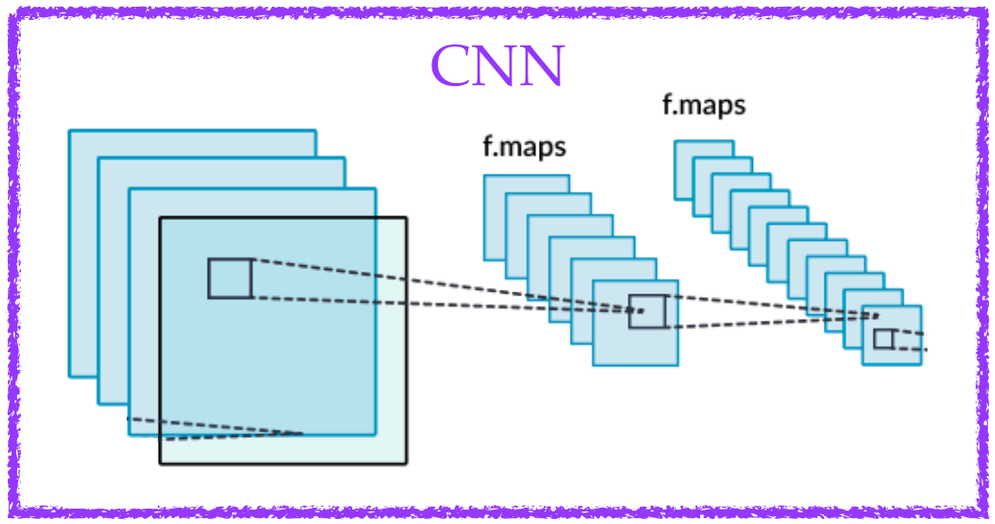
Potential Issues

- Once trained, it is not applicable to graphs of different sizes
- It is sensitive to node ordering
- It becomes computationally intensive when we have lots of nodes

Graph-Level Classification



Example: Determine if the molecule is a potential drug



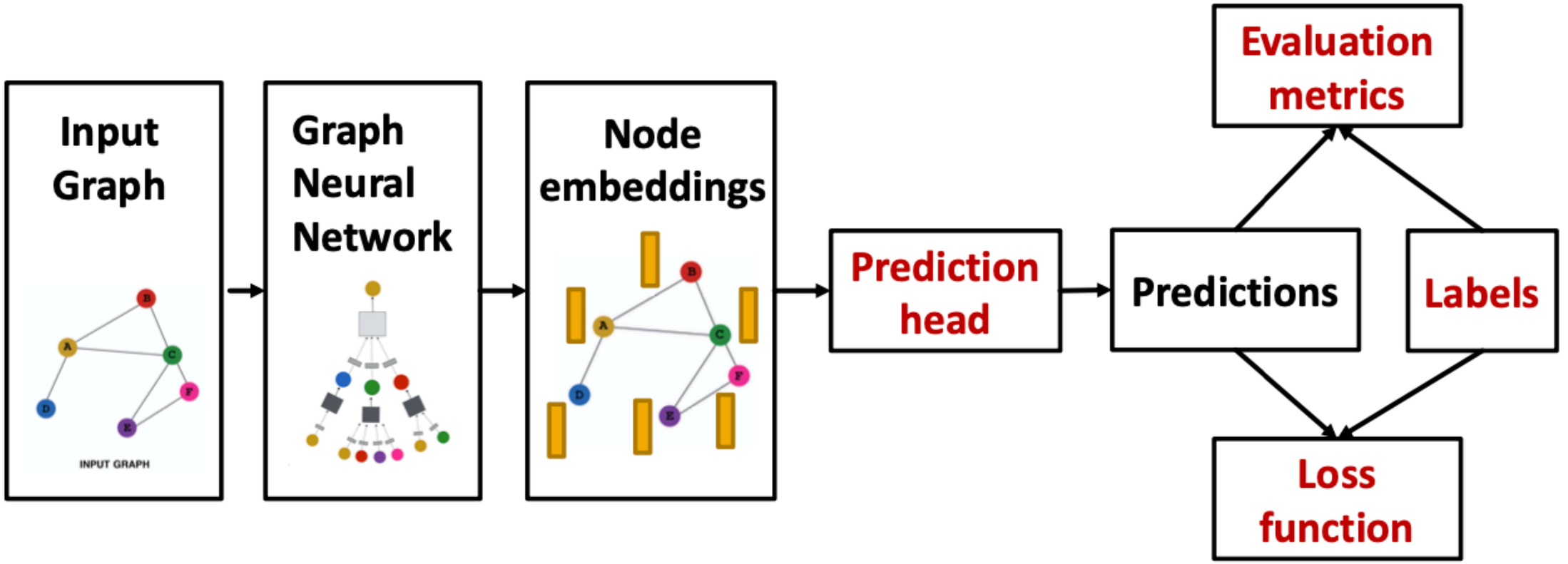
drug?
non-drug?

Potential Issues

- It is sensitive to node ordering
- It becomes computationally intensive when we have lots of nodes
- Does it really make sense to apply 2D convolution to the extended adjacency matrix?

Machine Learning on Graphs

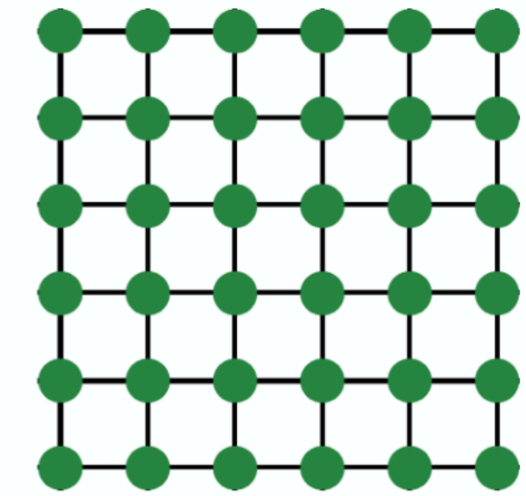
- Consider the node-level classification task as an example



- Obtaining a good node embedding ("features") for each node that takes into account not only its own information, but also other nodes' information will be beneficial for this task

  Idea: In most graphs, similar nodes are likely to connect to each other, so we can come up with operations that exploit local information (small neighborhood in a graph)

Machine Learning on Graphs



Image

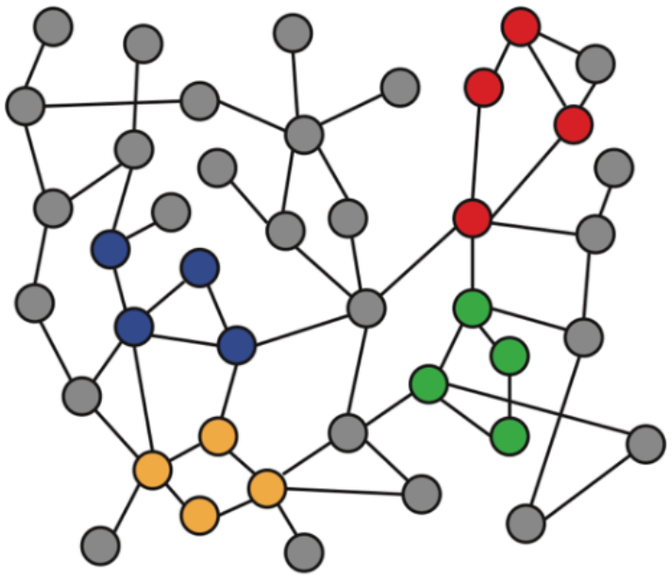
CNN

Sequence

1D CNN

MLP RNN

LSTM GRU

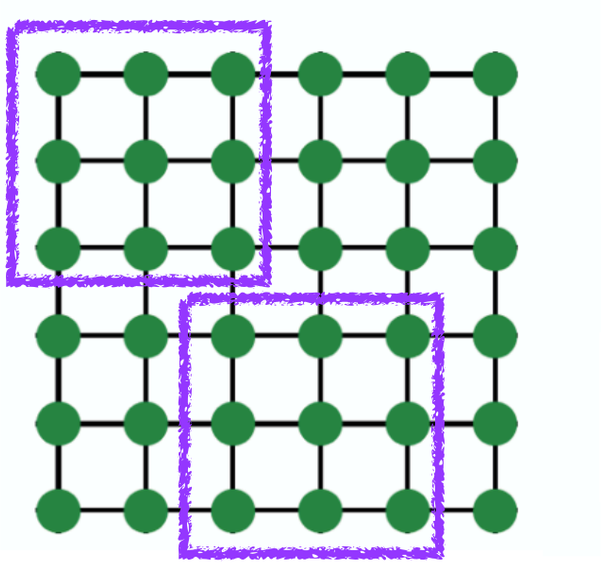


Graph

Transformer

Machine Learning on Graphs

Will convolution work?



Image

CNN

Sequence

1D CNN

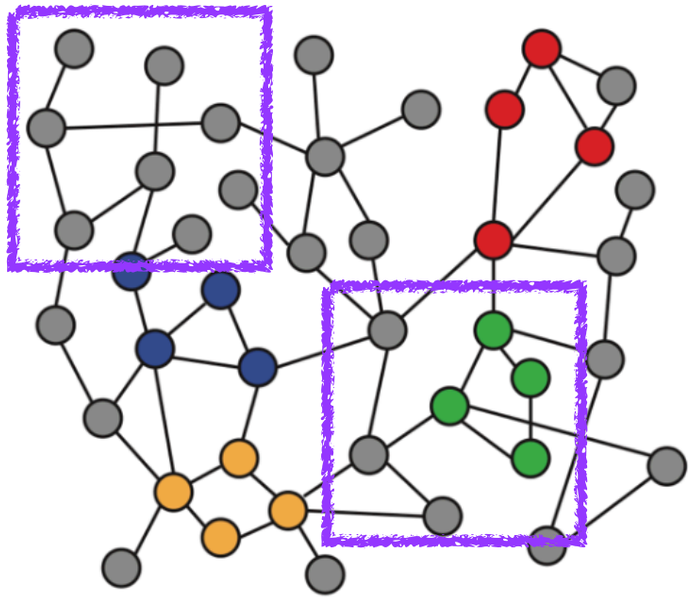
MLP RNN

LSTM GRU

Transformer

No spatial locality like grids

No fixed node ordering



Graph

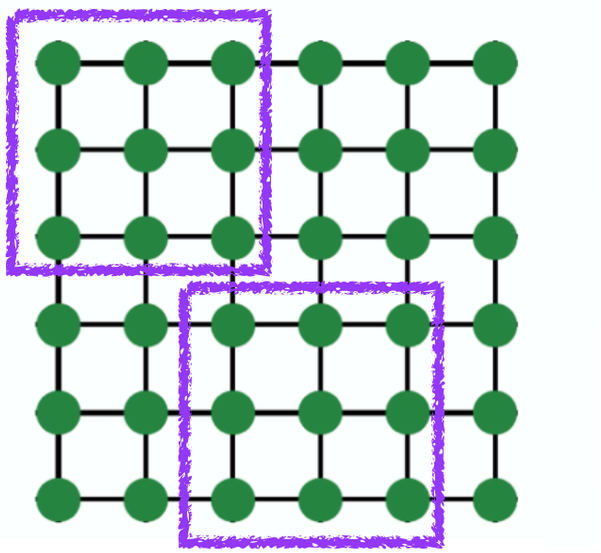
Graph Neural Network

Machine Learning on Graphs

Will convolution work?

No spatial locality like grids

No fixed node ordering



Image

CNN

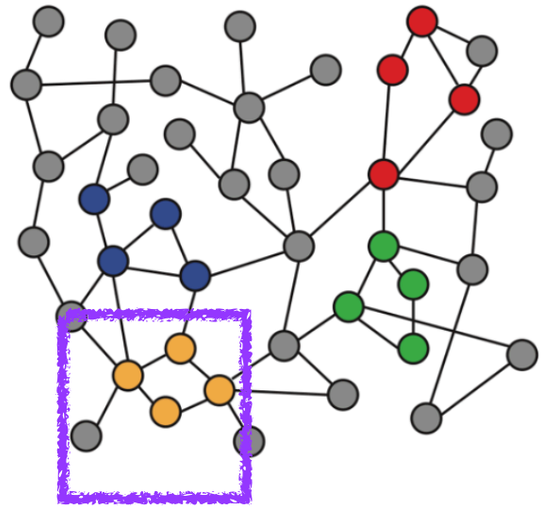
Sequence

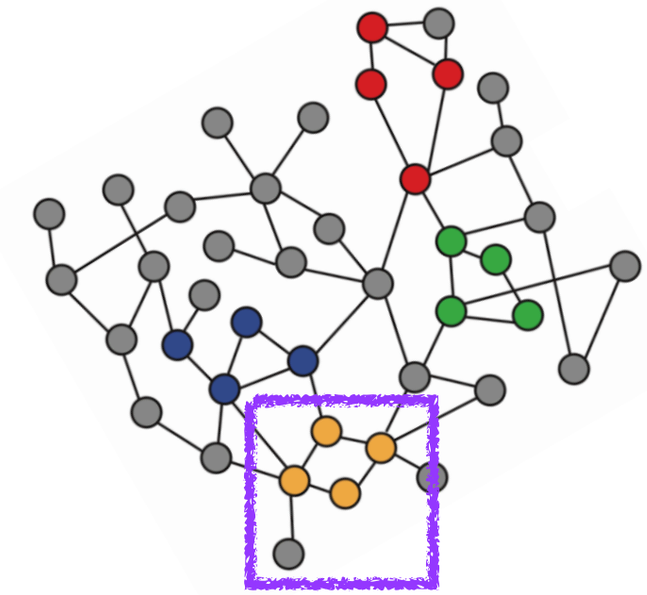
1D CNN

MLP RNN

LSTM GRU

Transformer



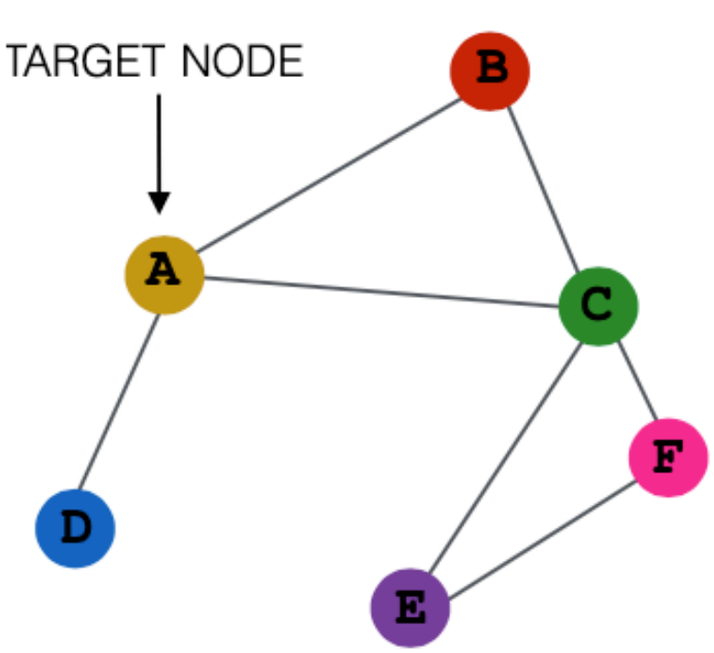


Graph

Graph Neural Network

Graph Neural Network (GNN)

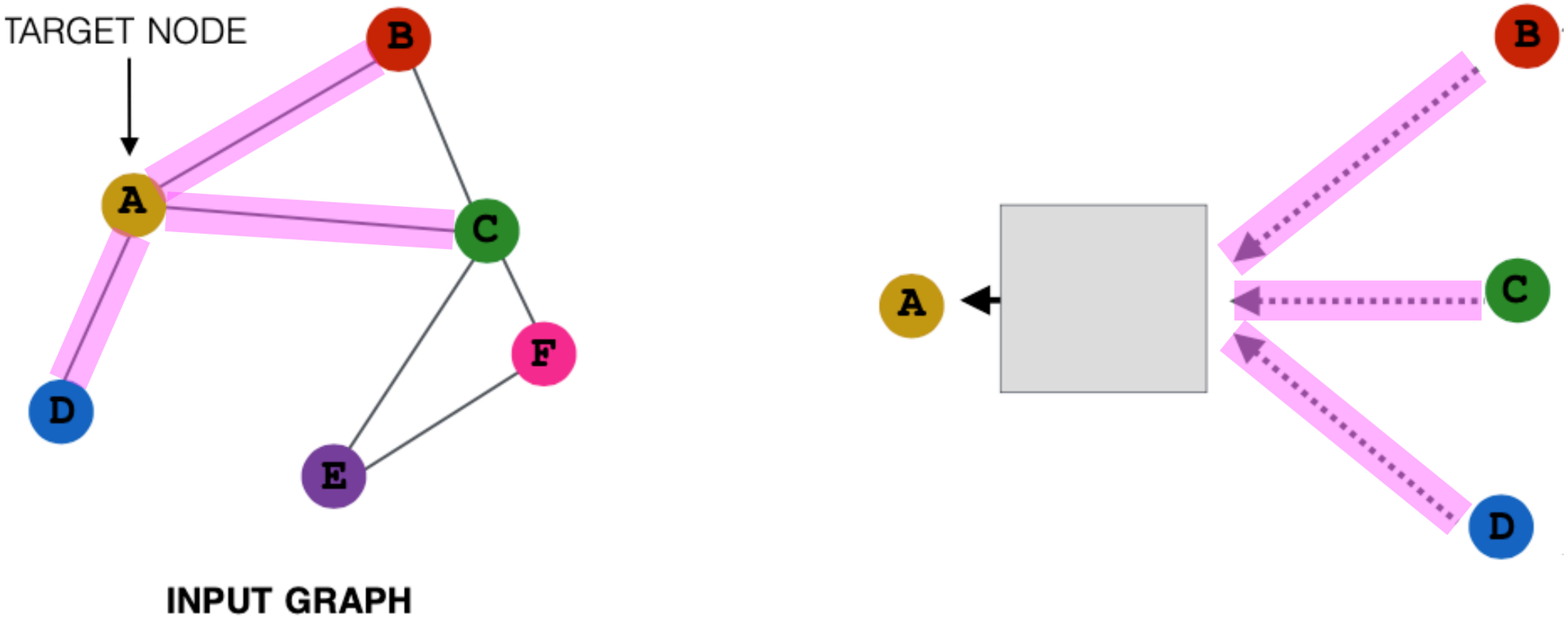
Idea: For each node, generate its node embedding ("features") based on its local neighborhood



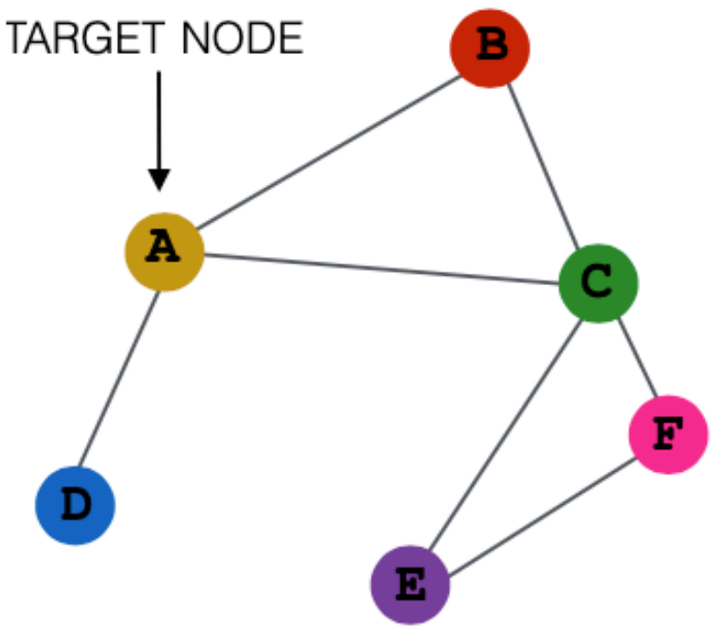
INPUT GRAPH

Graph Neural Network (GNN)

Idea: For each node, generate its node embedding ("features") based on its local neighborhood



Graph Neural Network (GNN)



INPUT GRAPH

Layer 2

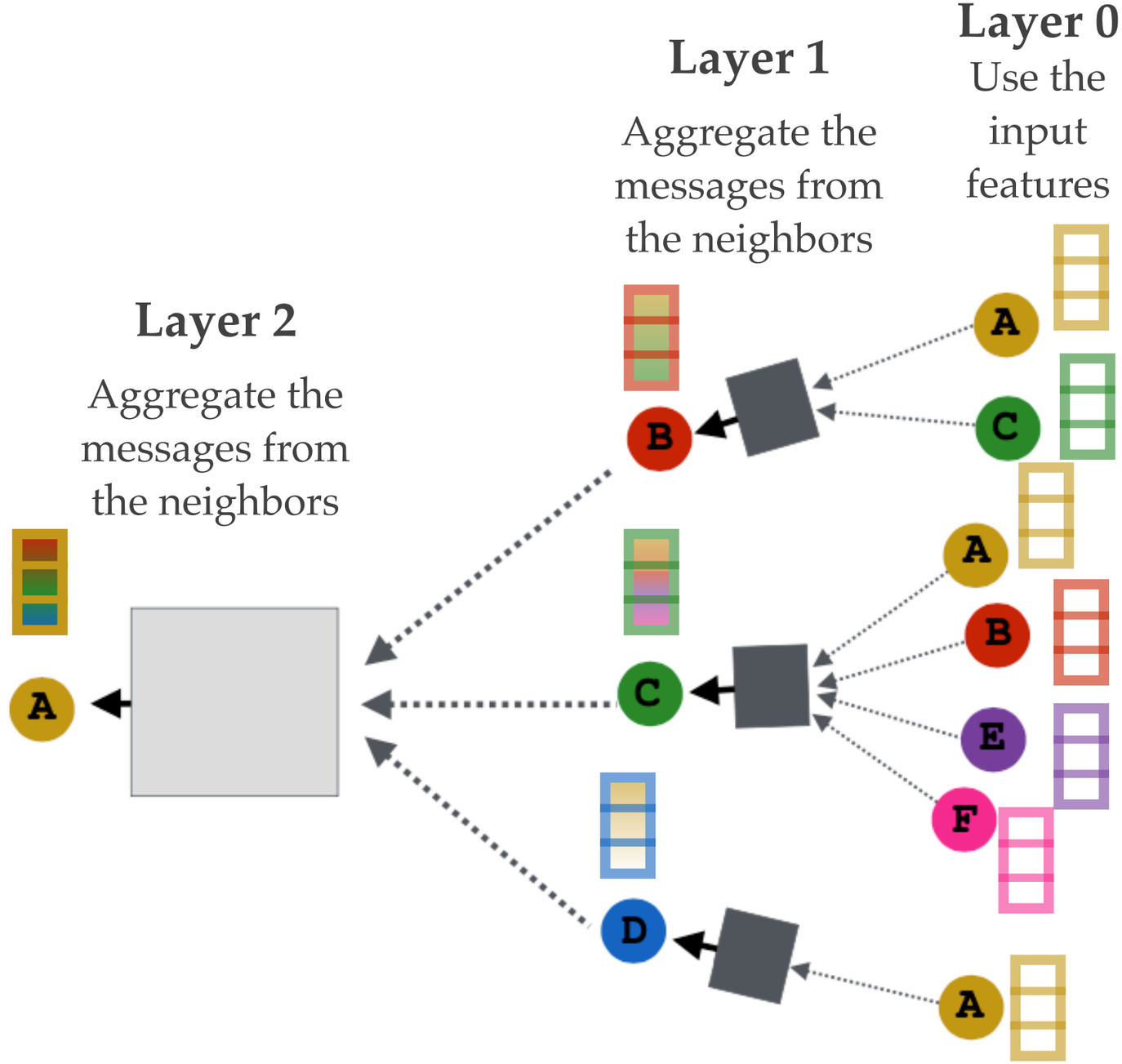
Aggregate the messages from the neighbors

Layer 1

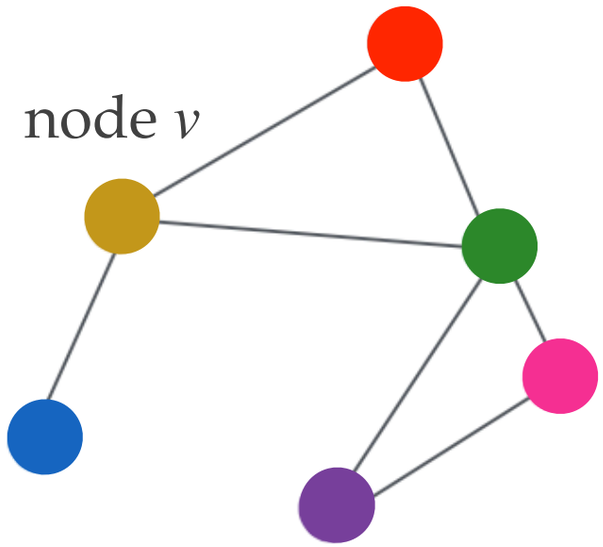
Aggregate the messages from the neighbors

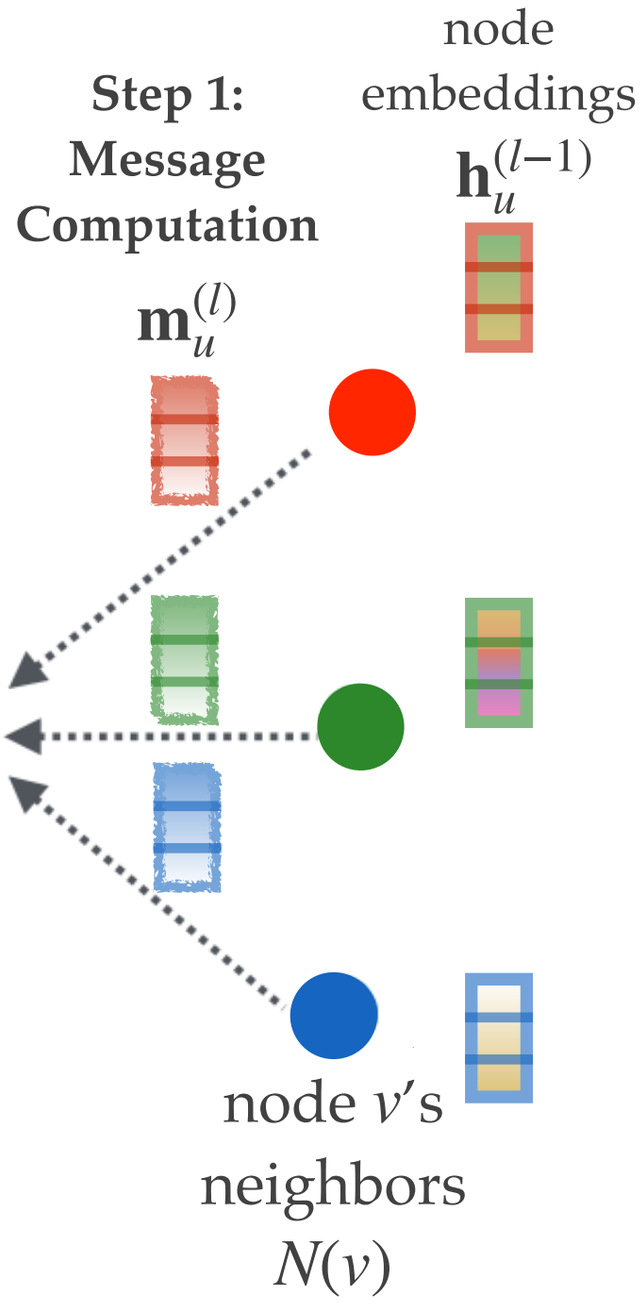
Layer 0

Use the input features



A Single GNN Layer



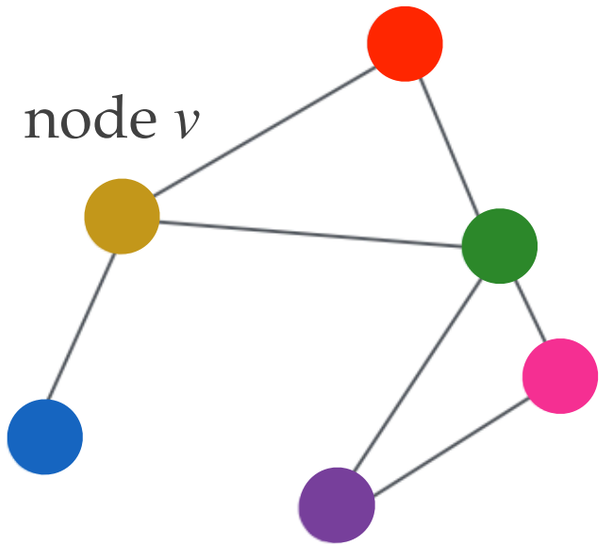


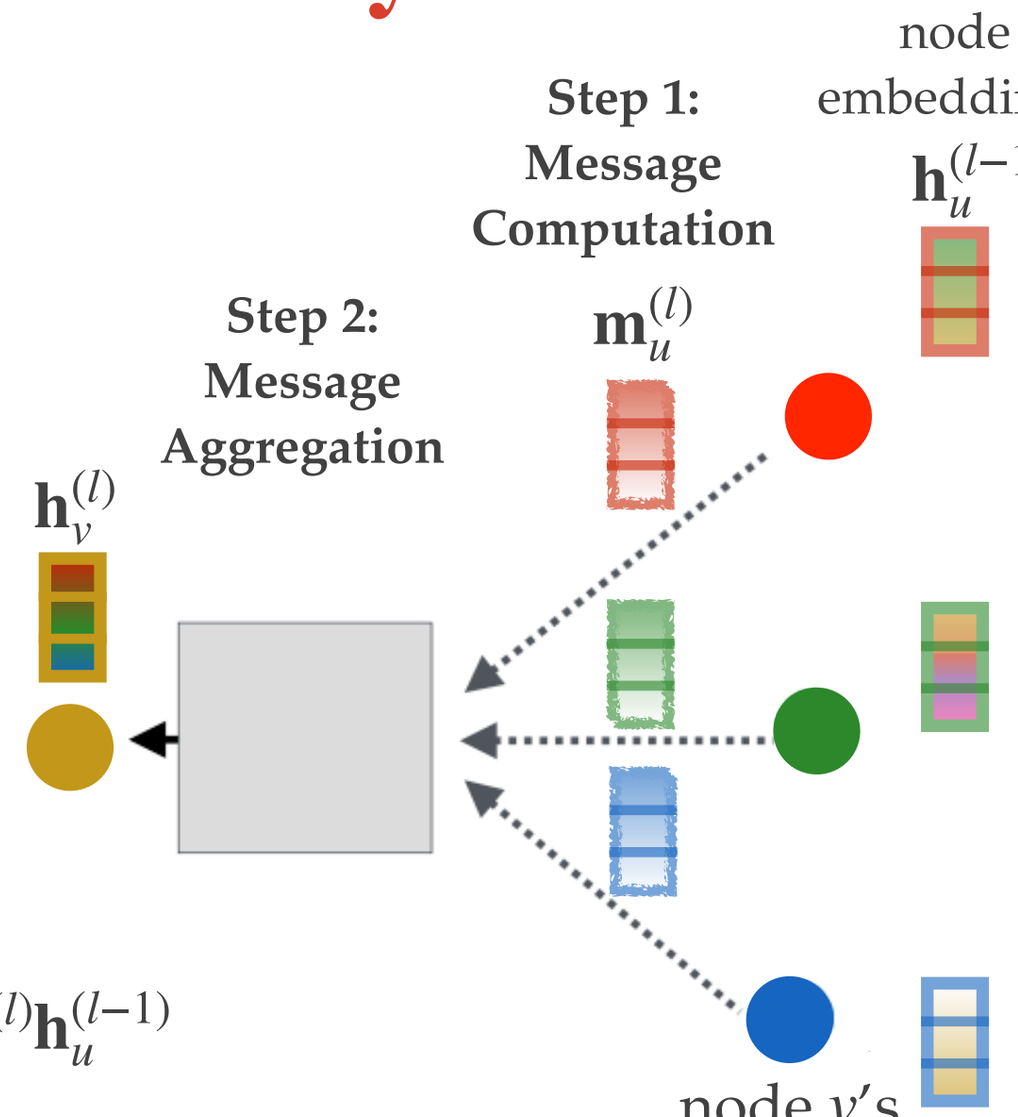
1. Message computation

$$\mathbf{m}_u^{(l)} = MSG^{(l)}(\mathbf{h}_u^{(l-1)}), u \in \{N(v) \cup v\}$$

Ex. Applying a linear layer $\mathbf{m}_u^{(l)} = W^{(l)}\mathbf{h}_u^{(l-1)}$

A Single GNN Layer





1. Message computation

$$\mathbf{m}_u^{(l)} = MSG^{(l)}(\mathbf{h}_u^{(l-1)}), u \in \{N(v) \cup v\}$$

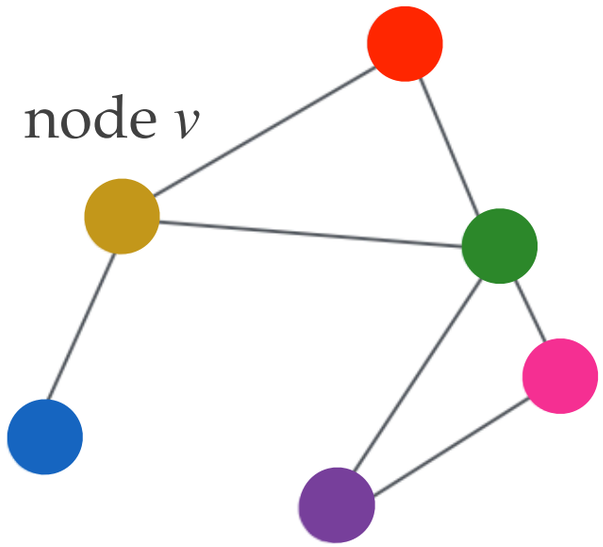
Ex. Applying a linear layer $\mathbf{m}_u^{(l)} = W^{(l)}\mathbf{h}_u^{(l-1)}$

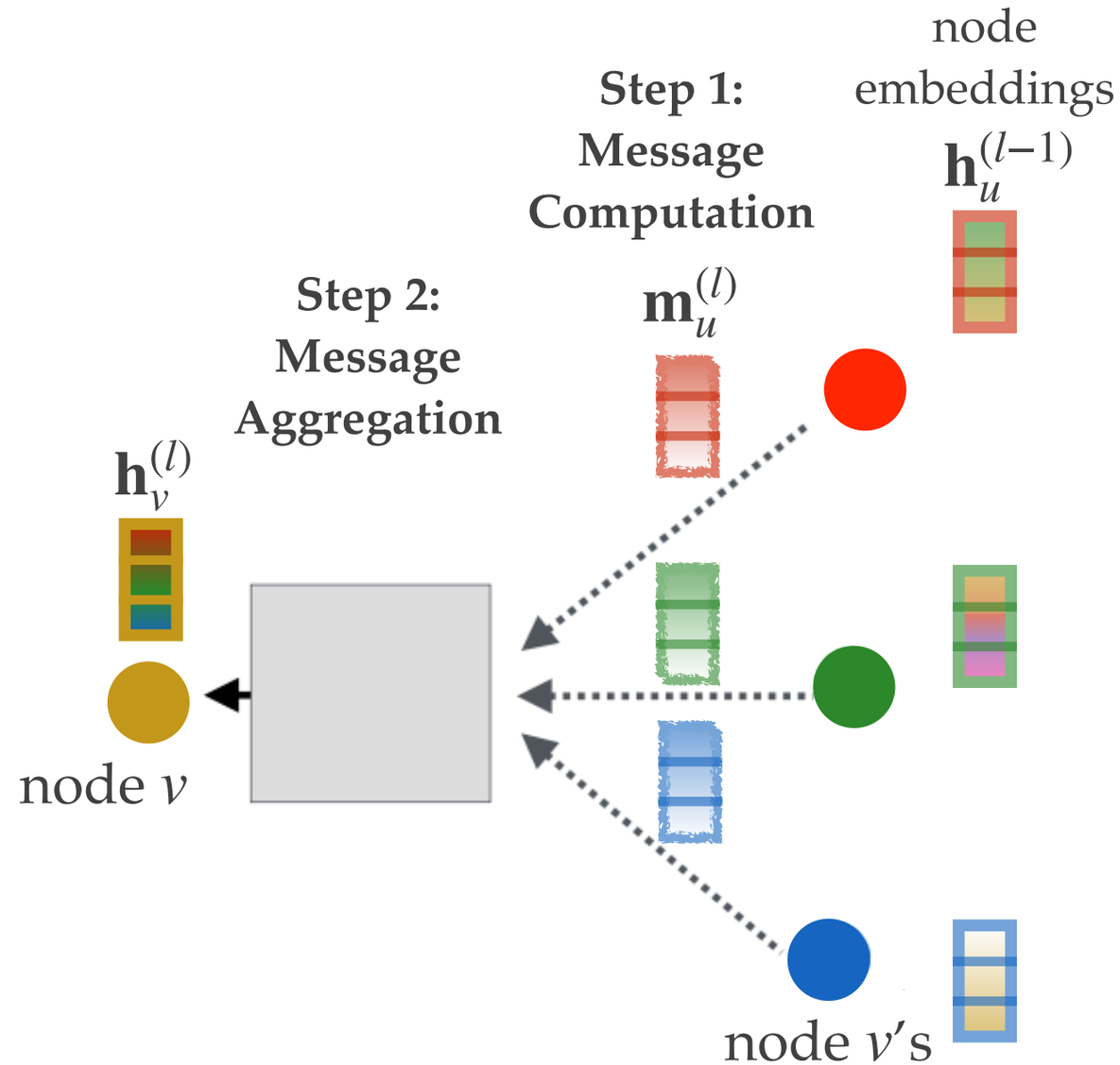
2. Message aggregation possibly with its own message

$$\mathbf{h}_v^{(l)} = AGG^{(l)}(\{\mathbf{m}_u^{(l)}, u \in N(v)\}), \mathbf{m}_v^{(l)})$$

Ex. Summing over the messages $\mathbf{h}_v^{(l)} = \sum_{u \in \{N(v) \cup v\}} \mathbf{m}_u^{(l)}$

Graph Convolutional Network (GCN)





1. Message computation

$$\mathbf{m}_u^{(l)} = \frac{W^{(l)}\mathbf{h}_u^{(l-1)}}{|N(v)|}$$

2. Message aggregation

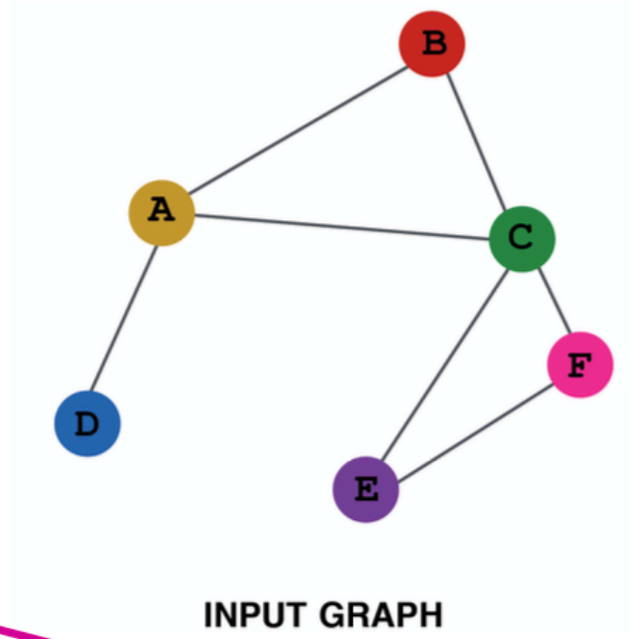
$$\mathbf{h}_v^{(l)} = ReLU\left(\sum_{u \in N(v)} \mathbf{m}_u^{(l)}\right)$$

matrix form

$$\mathbf{H}^{(l)} = ReLU(\tilde{\mathbf{A}}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)})$$

Graph Neural Network (GNN)

Every node defines a computation graph based on its neighborhood!

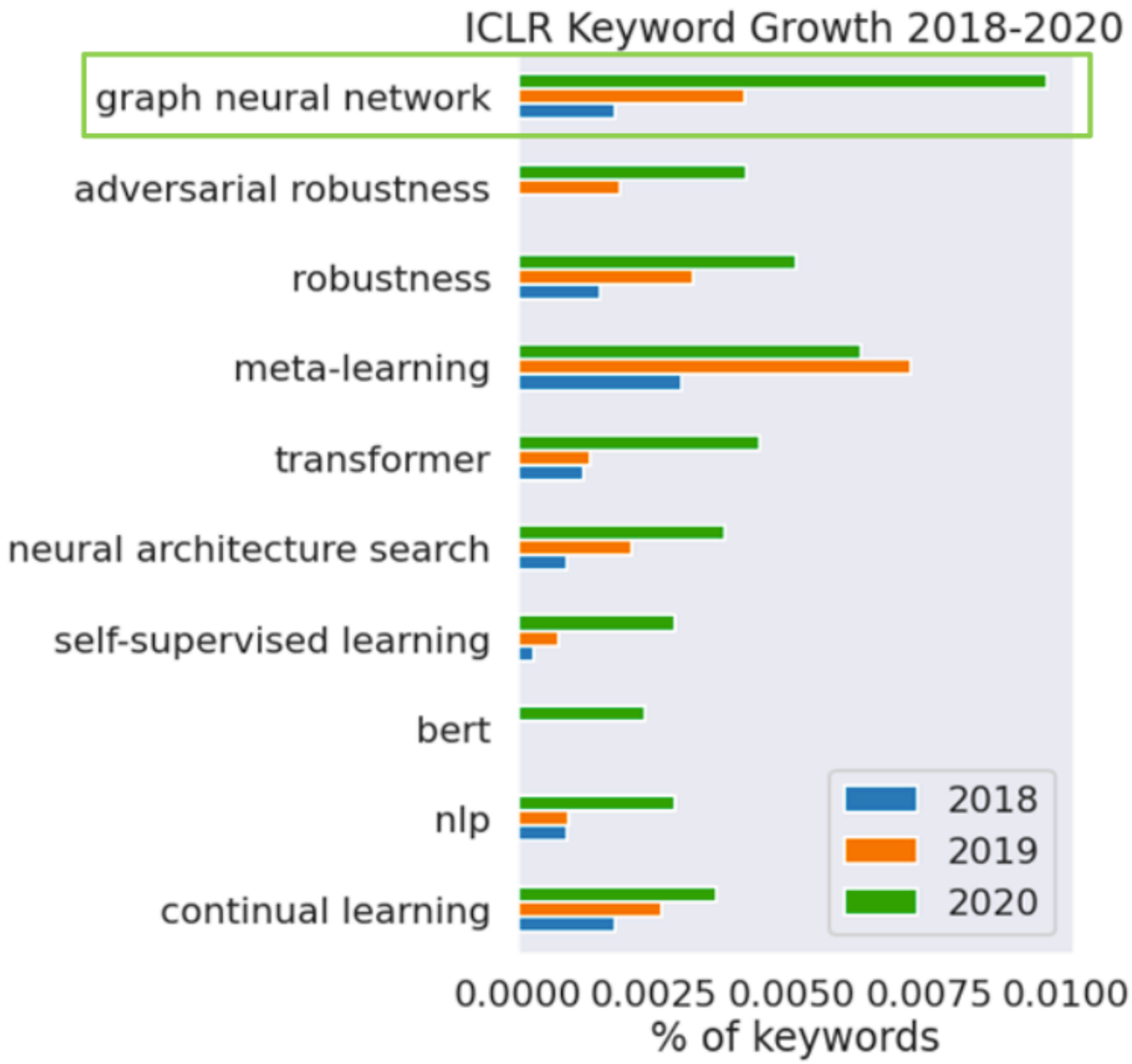


keywords: graph isomorphism network (GIN), graphSAGE, graph attention network (GAT), Cluster-GCN

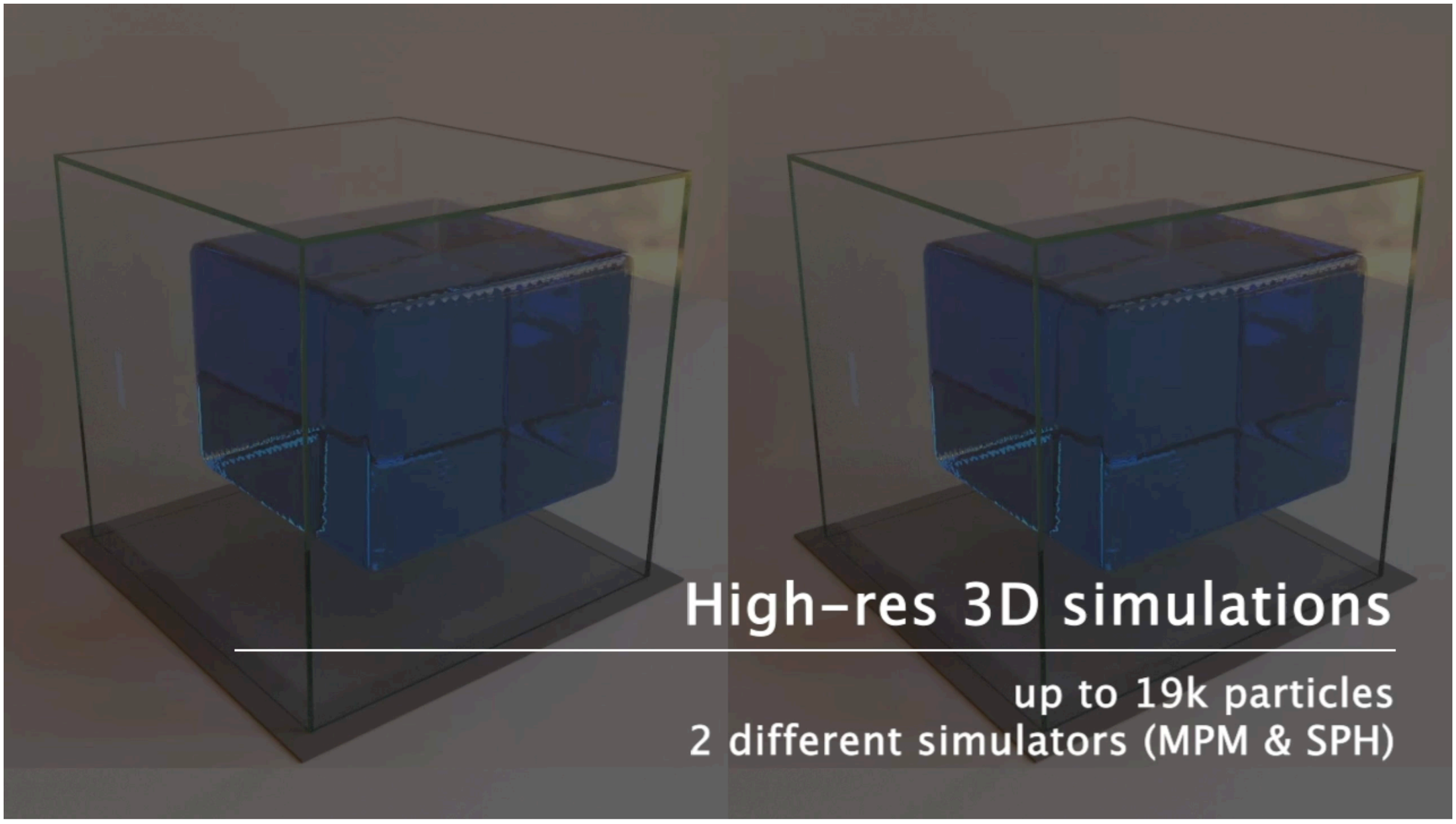
Graph Neural Network (GNN)

- There are many more techniques that are related to graph neural networks
  - Different ways to perform message computation and aggregation
  - Feature augmentation
  - Neighbor sampling
  - Position- and identity-aware GNNs
- Many deep learning components can also be incorporated such as
  - Attention mechanism
  - Gating
  - Skip/residual connections
  - Batch normalization
  - Dropout
  - Data augmentation

Graph Neural Network (GNN)



Applications: Physical Simulation



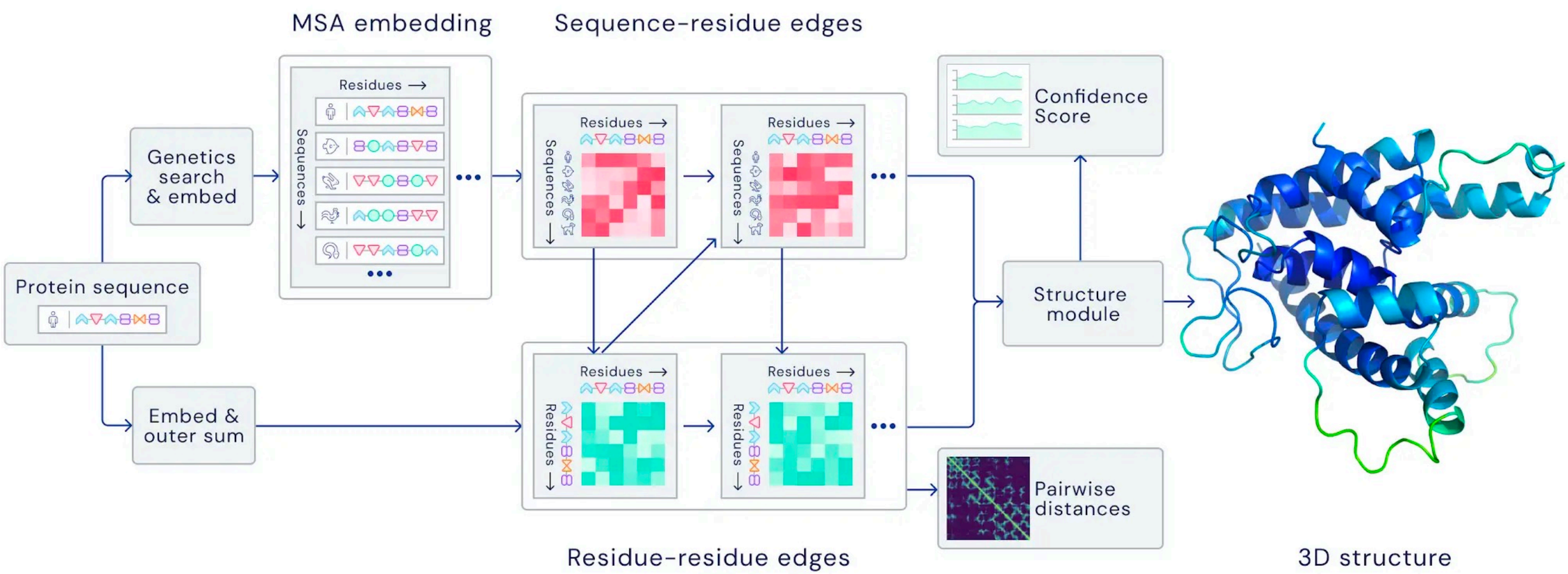
nodes: particles

edges: interaction between particles

Sanchez-Gonzalez, Alvaro, et al. "Learning to simulate complex physics with graph networks." International Conference on Machine Learning. PMLR, 2020.

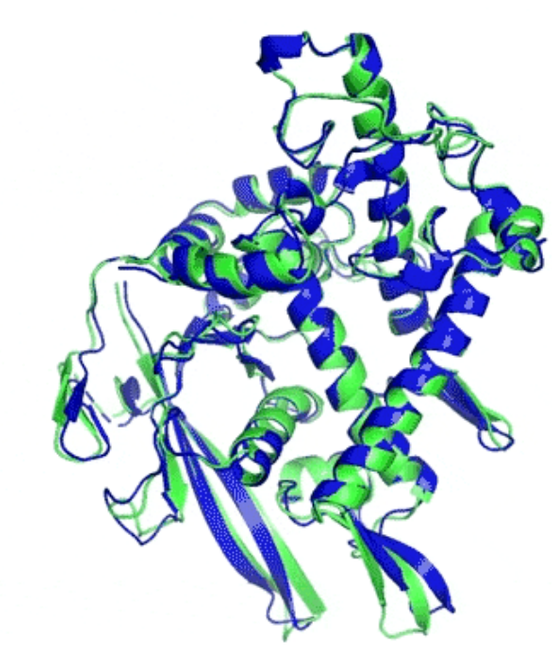
Applications: Protein Folding

an amino acid sequence → the corresponding protein's 3D structure

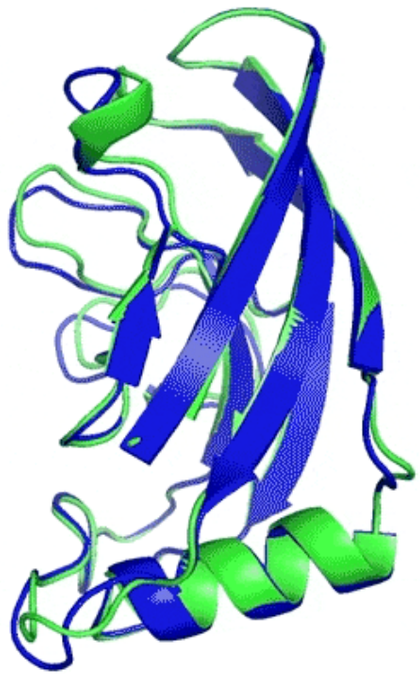


AlphaFold: a solution to a 50-year-old grand challenge in biology

Applications: Protein Folding



T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)



T1049 / 6y4f
93.3 GDT
(adhesin tip)

Experimental result

Computational prediction

AlphaFold: a solution to a 50-year-old grand challenge in biology