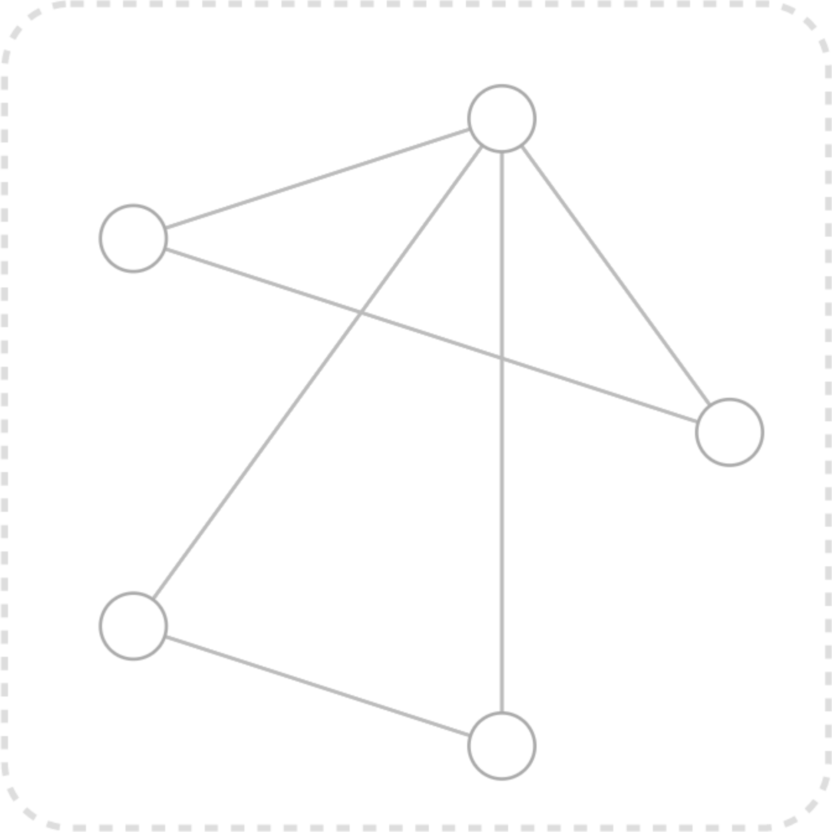
Graph Neural Networks

최민동

A Table of Contents

Part1 Introduction

What are Graphs?



V Vertex (or node) attributes

e.g., node identity, number of neighbors

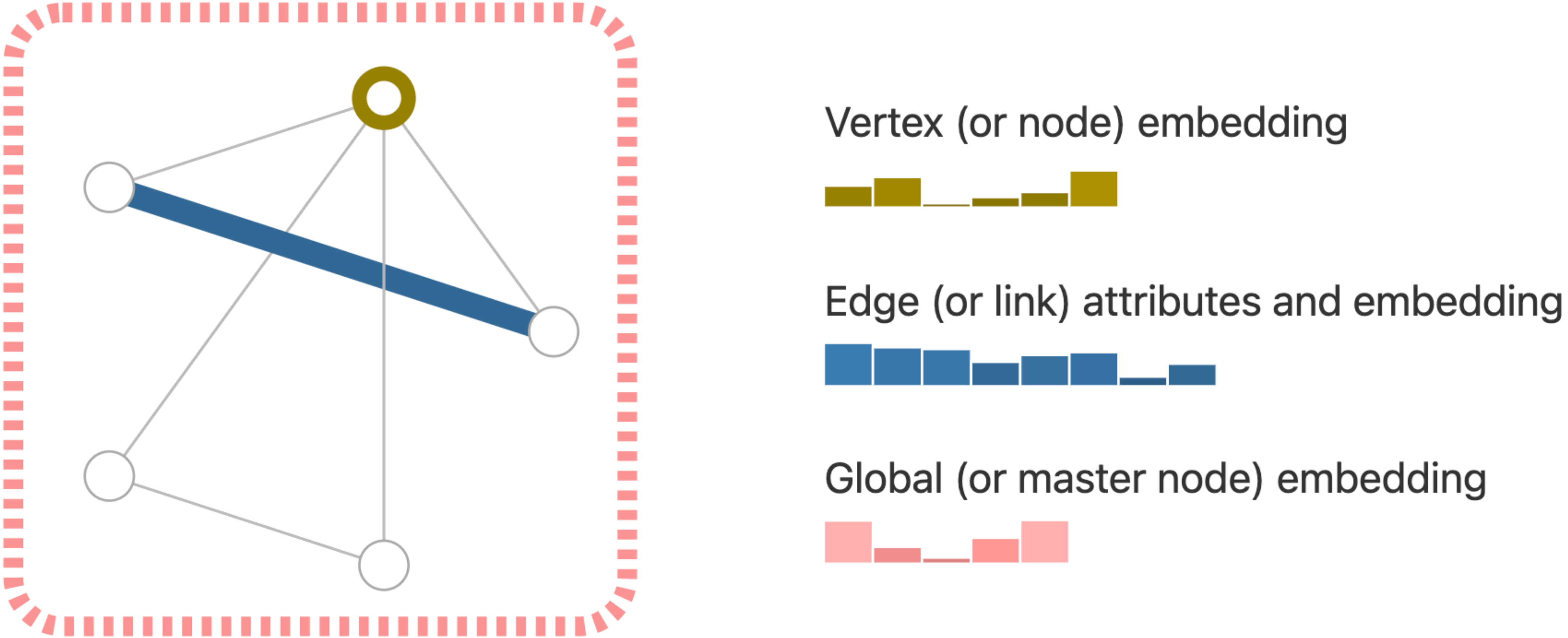
E Edge (or link) attributes and directions

e.g., edge identity, edge weight

U Global (or master node) attributes

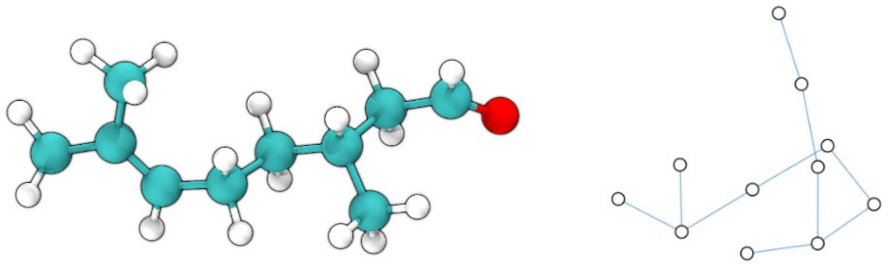
e.g., number of nodes, longest path

Embedding of Graphs

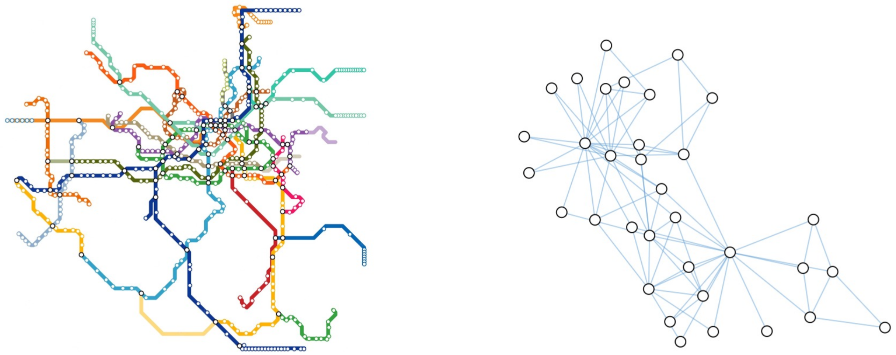


Information in the form of scalars or embeddings can be stored at each graph node (left) or edge (right).

Example of Graphs



Molecular Representation



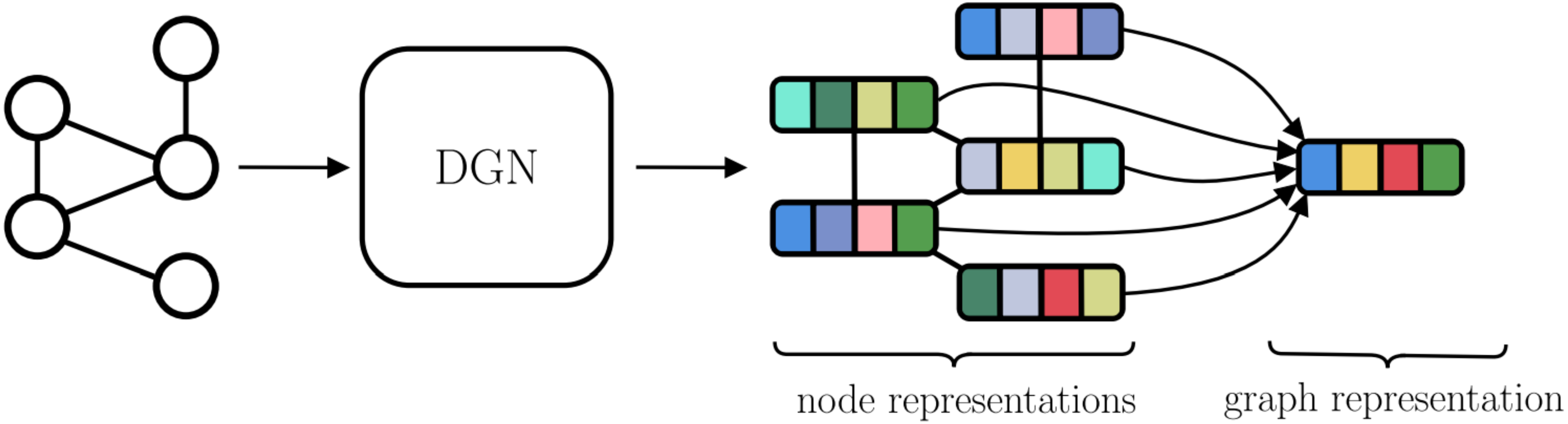
Transport Network Representation

Social Network Representaiton

Visual Scene Graph

Part2 High-Level Overview

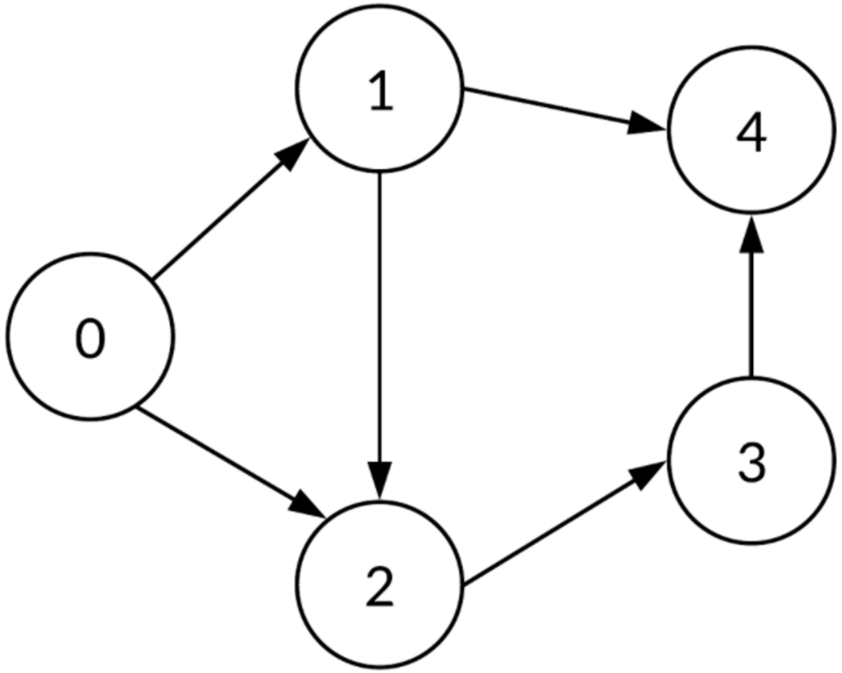
State



The bigger picture that all graph learning methods share. A “Deep Graph Network” takes an input graph and produces node representations. Such representations can be aggregated to form a single graph representatio
n.

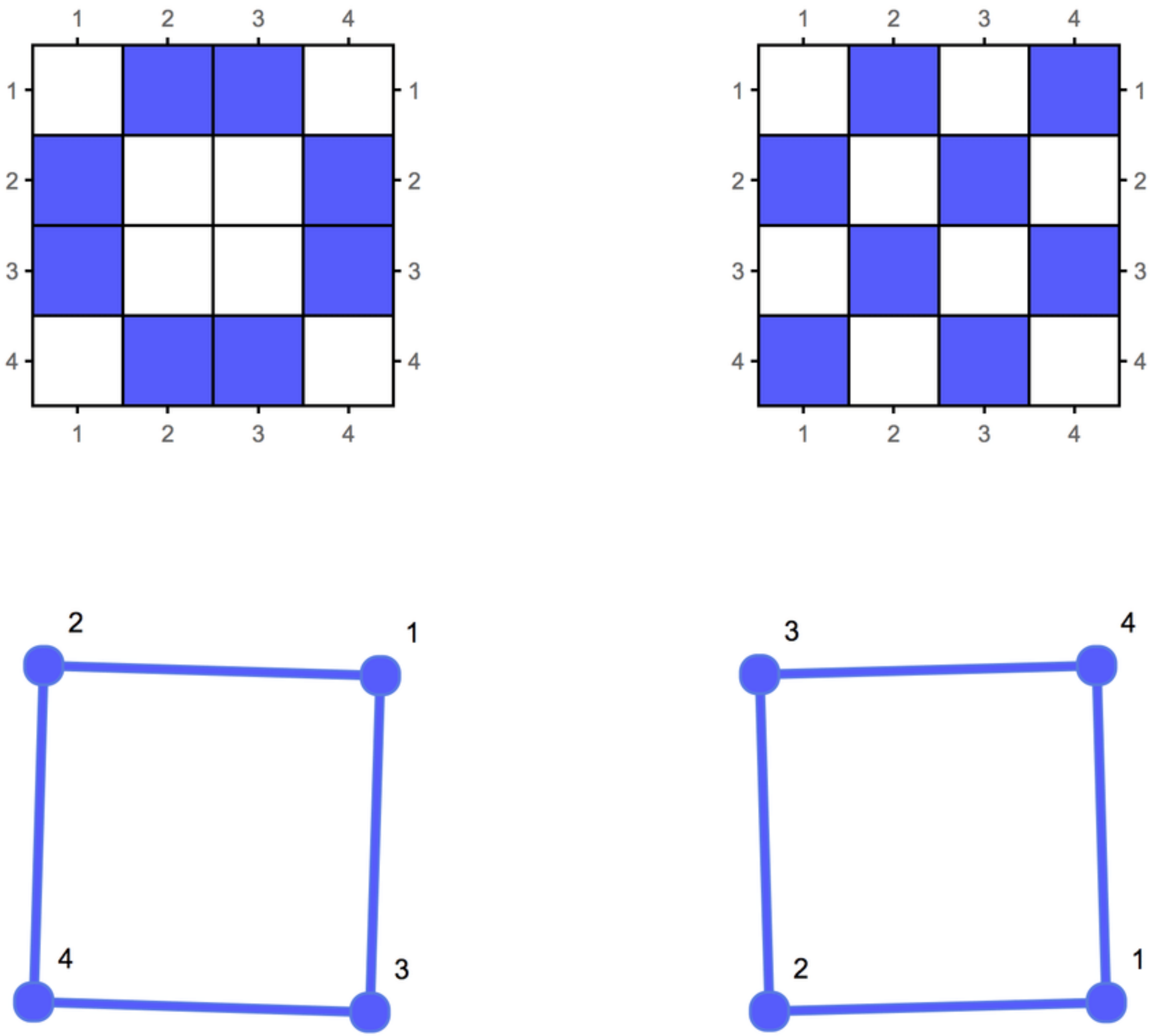
States: Represent Each Node as a Vector !

Adjacency Matrix

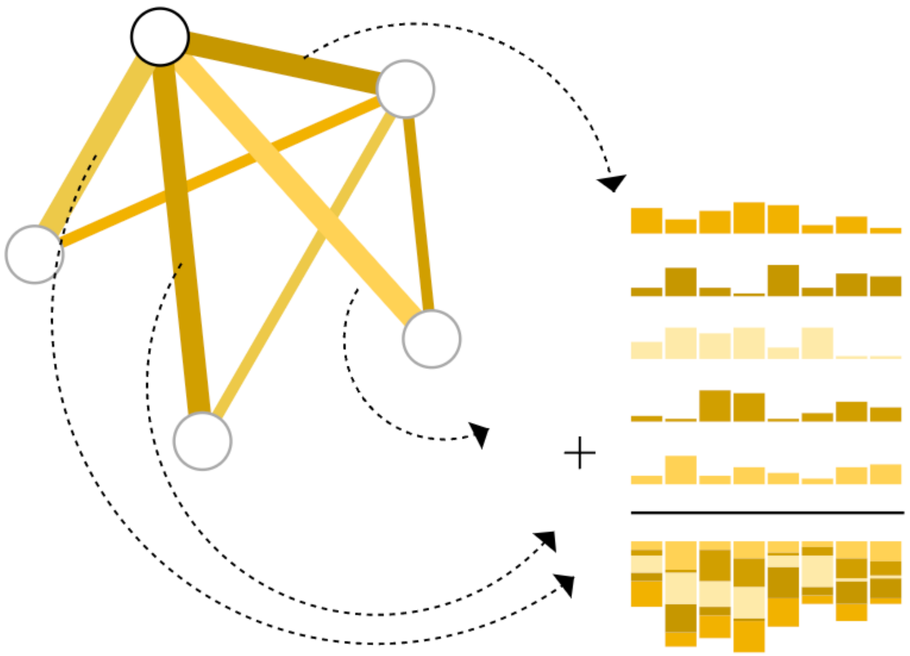


	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	1
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0

Isomorphic Graphs



Permutation Invariant Fuction



Aggregate information
from adjacent edges

Aggregation using PIF

Sum

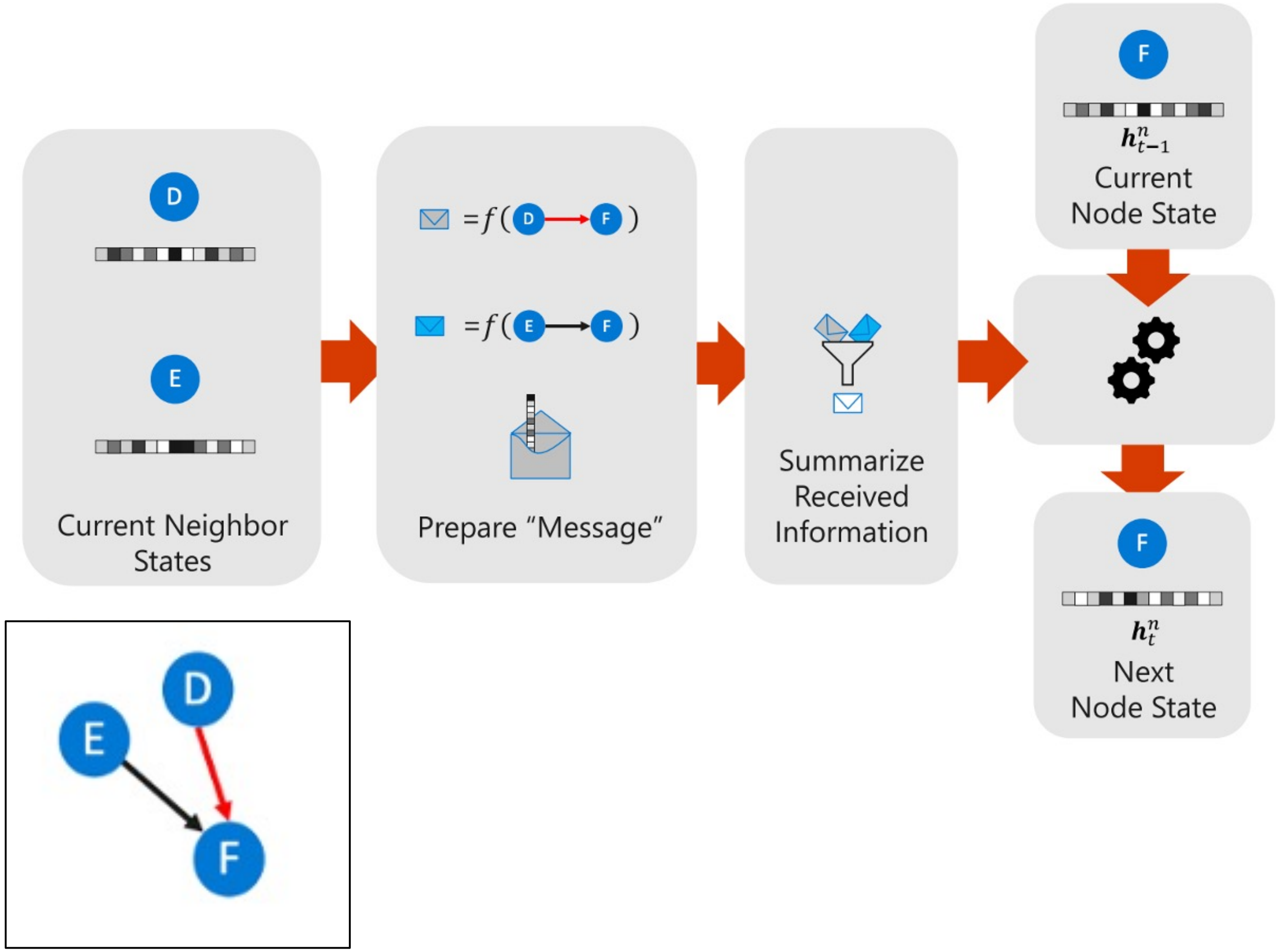
Mean

Max

MLP

Self-Attention

Message Passing



Message Dispatching

A message is computed for each node, using its current state and (possibly) edge information. Then, the message is sent to neighboring nodes according to the graph structure

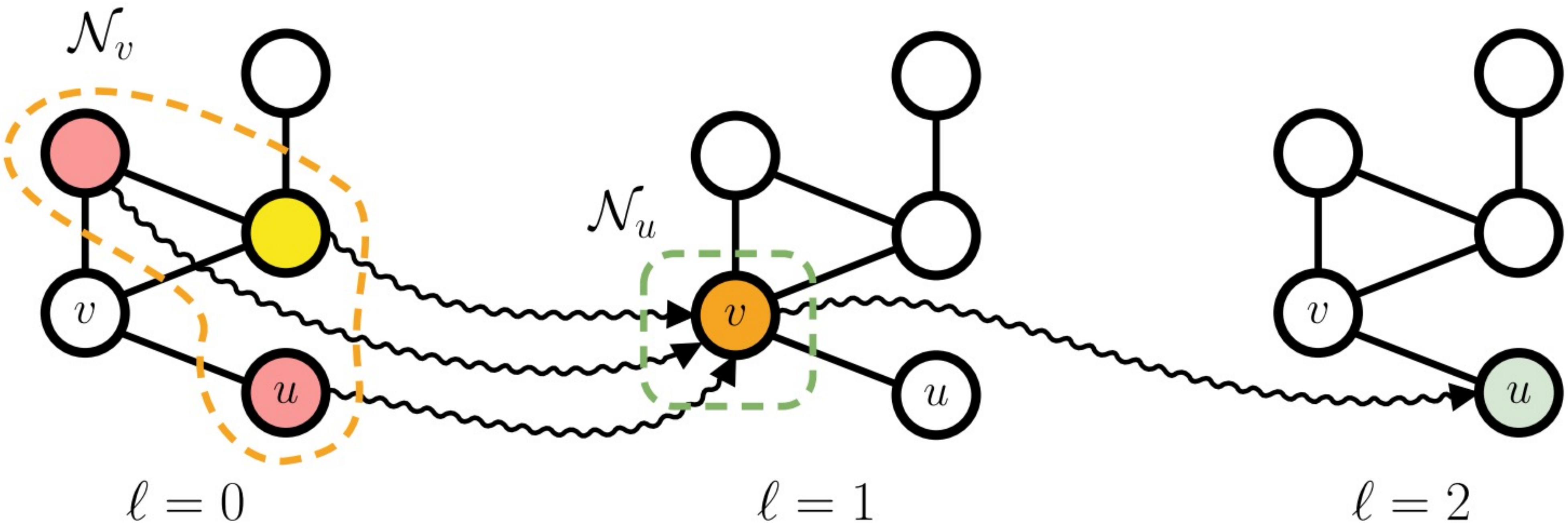
State Update

The incoming node messages, and possibly its state, are collected and used to update the node state.

Processing

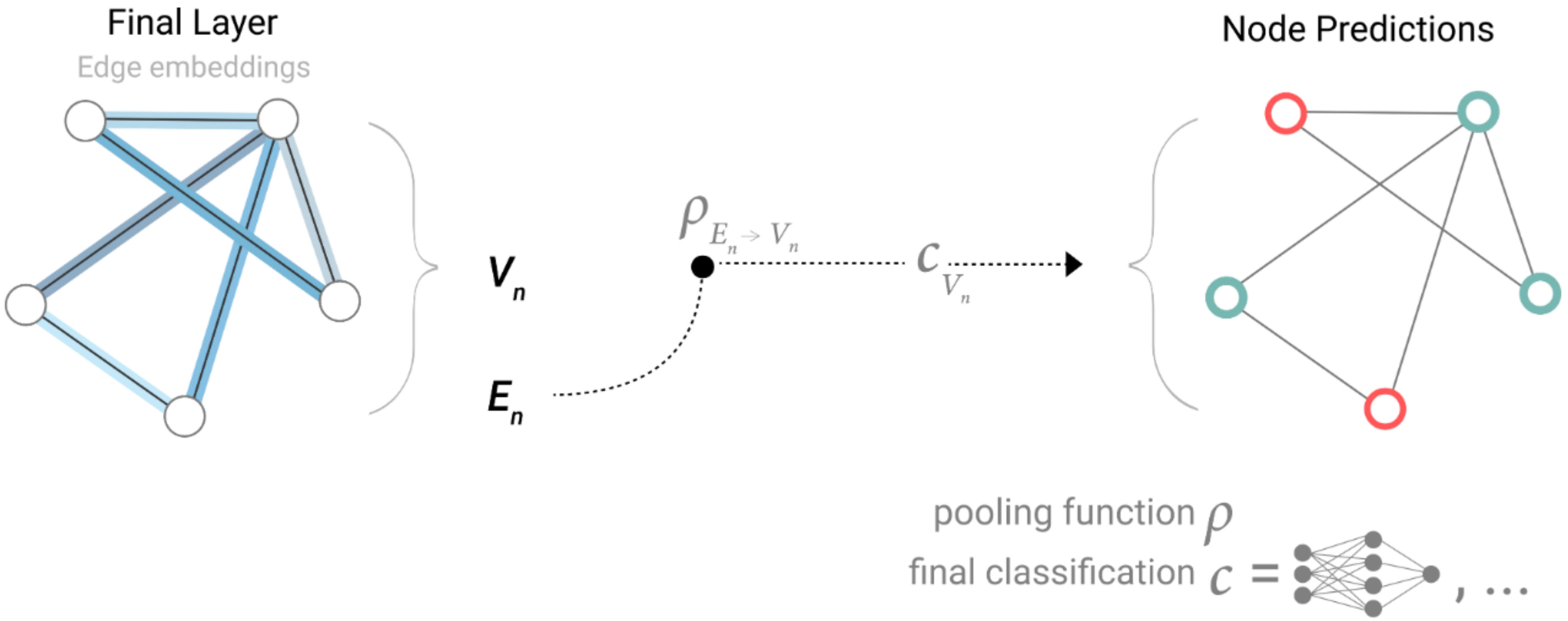
Convolutionally or Recurrently

Context Diffusion

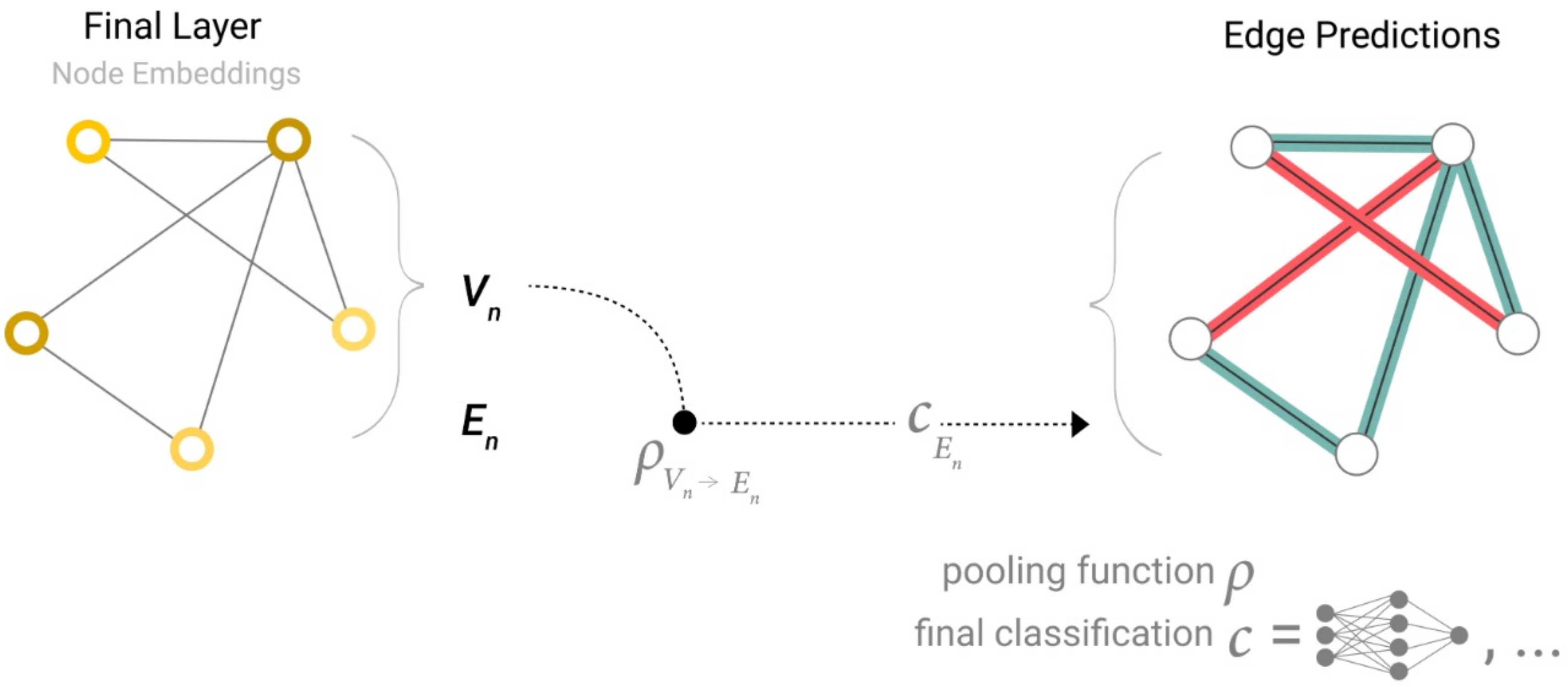


Part3 Building Blocks

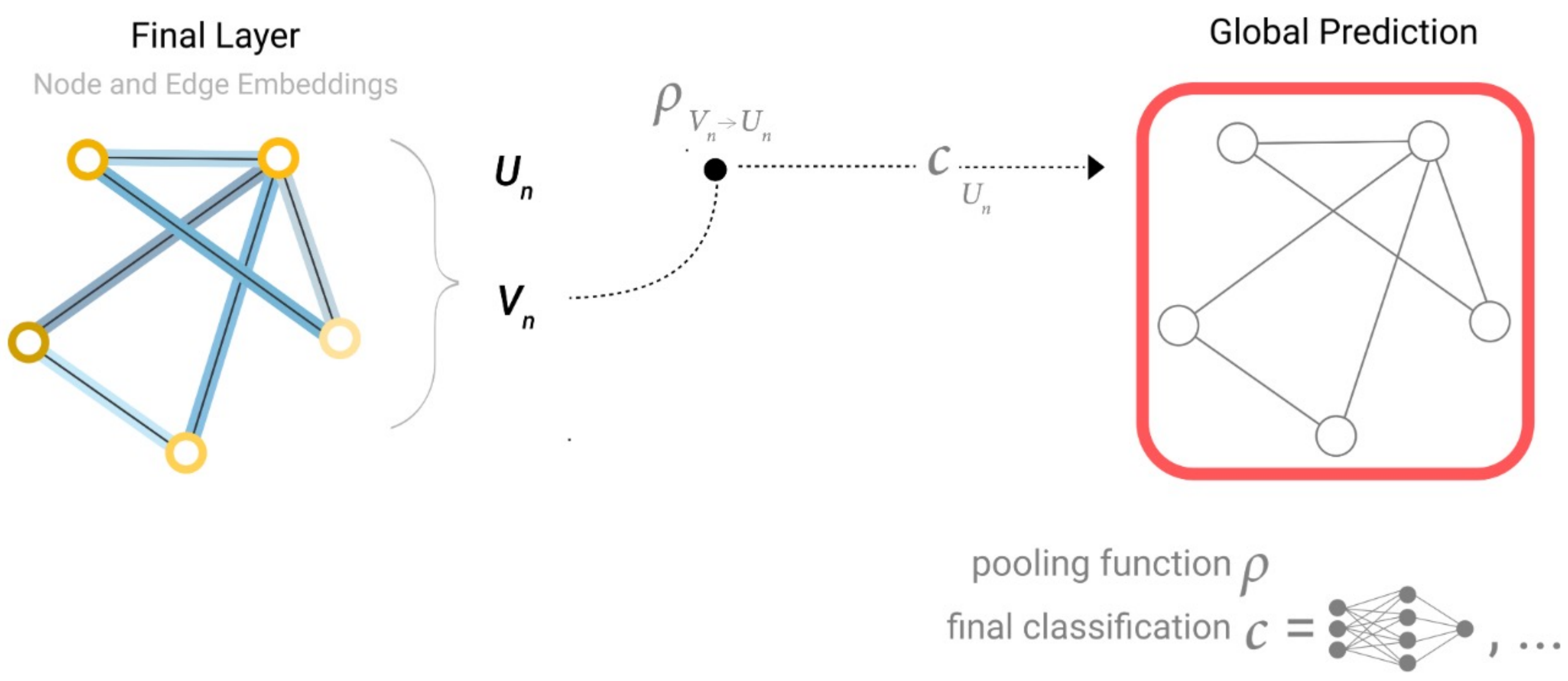
Pooling (Edges → Nodes)



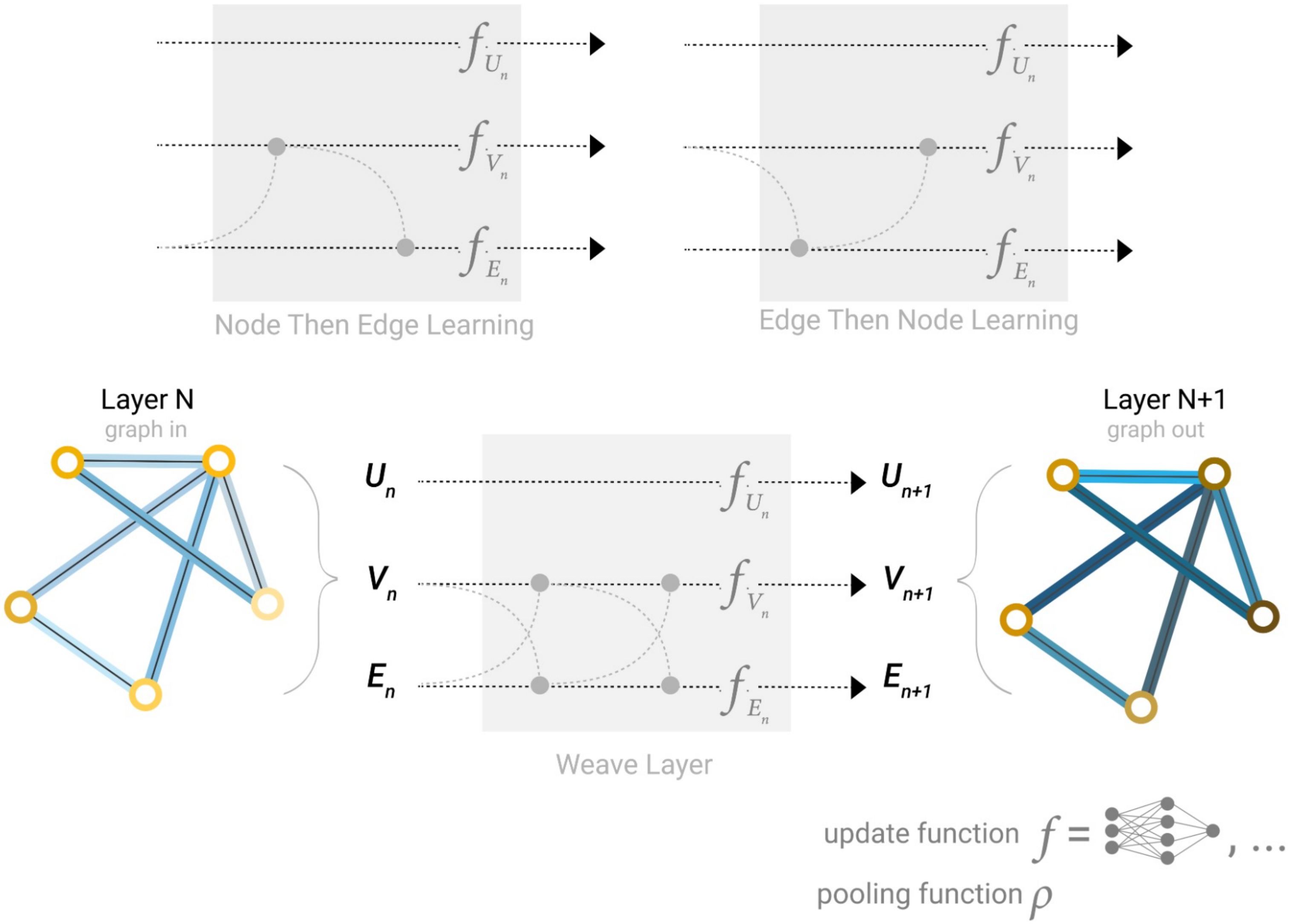
Pooling (Nodes → Edges)



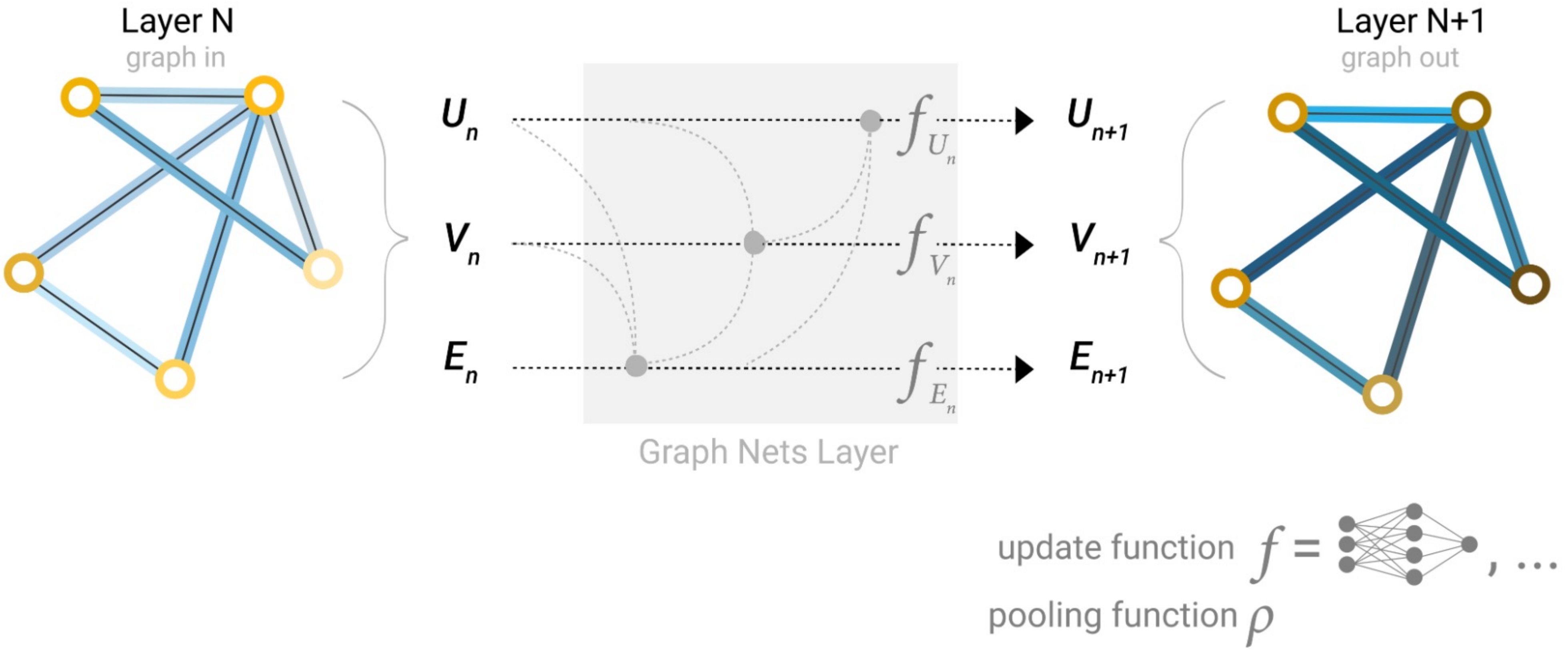
Pooling (Nodes → Global)



Combining (Weaving)

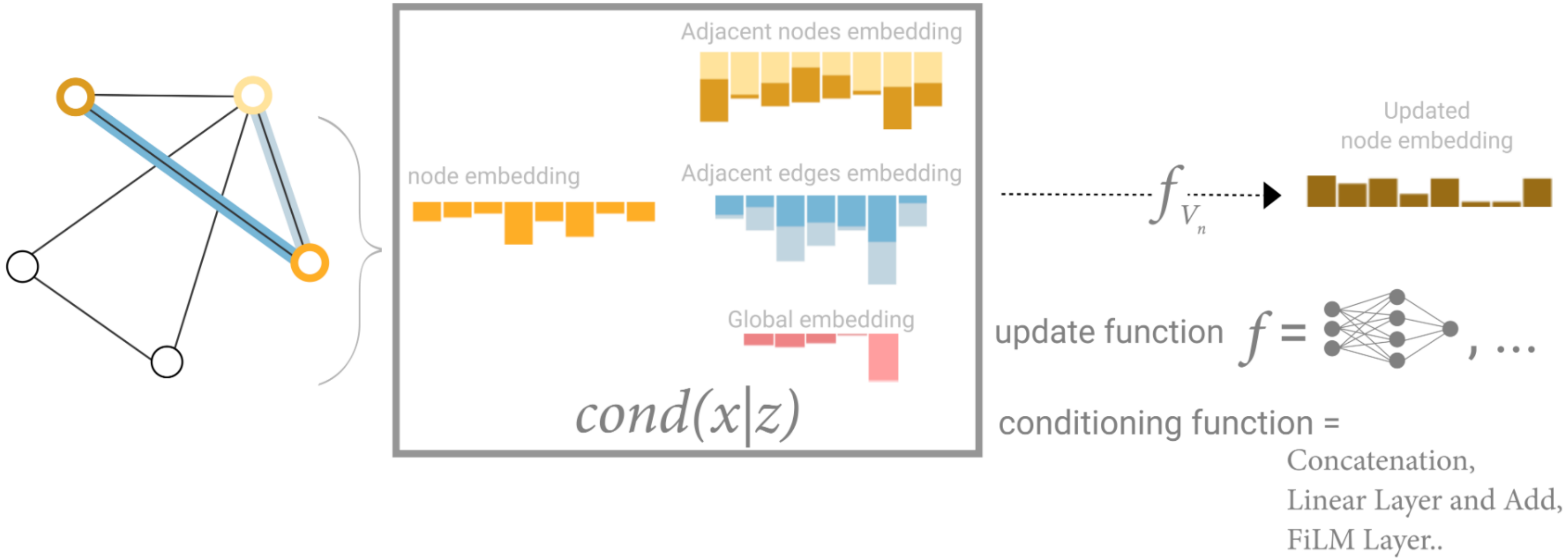


Combining



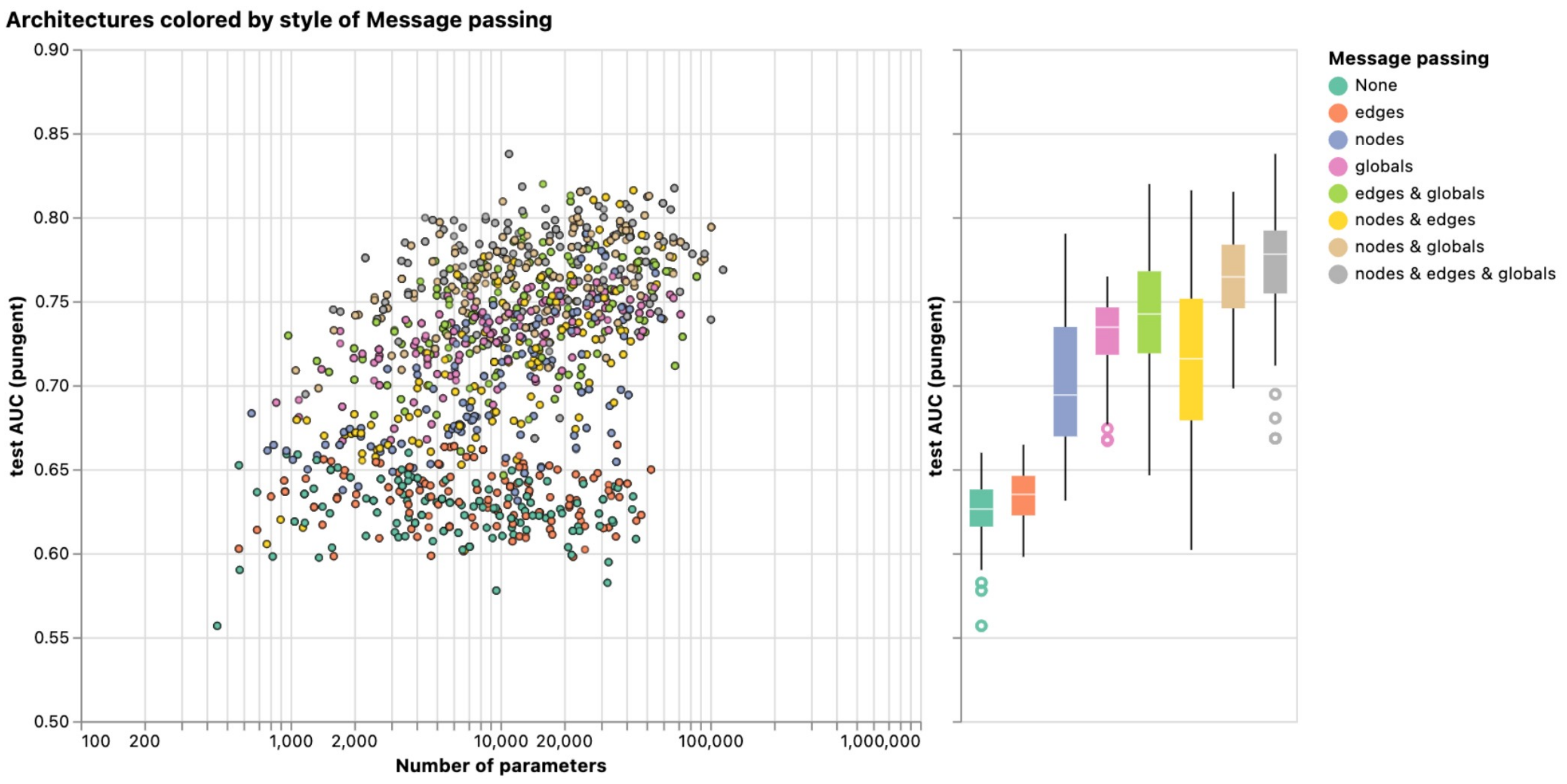
Schematic of a Graph Nets architecture leveraging global representations.

In General



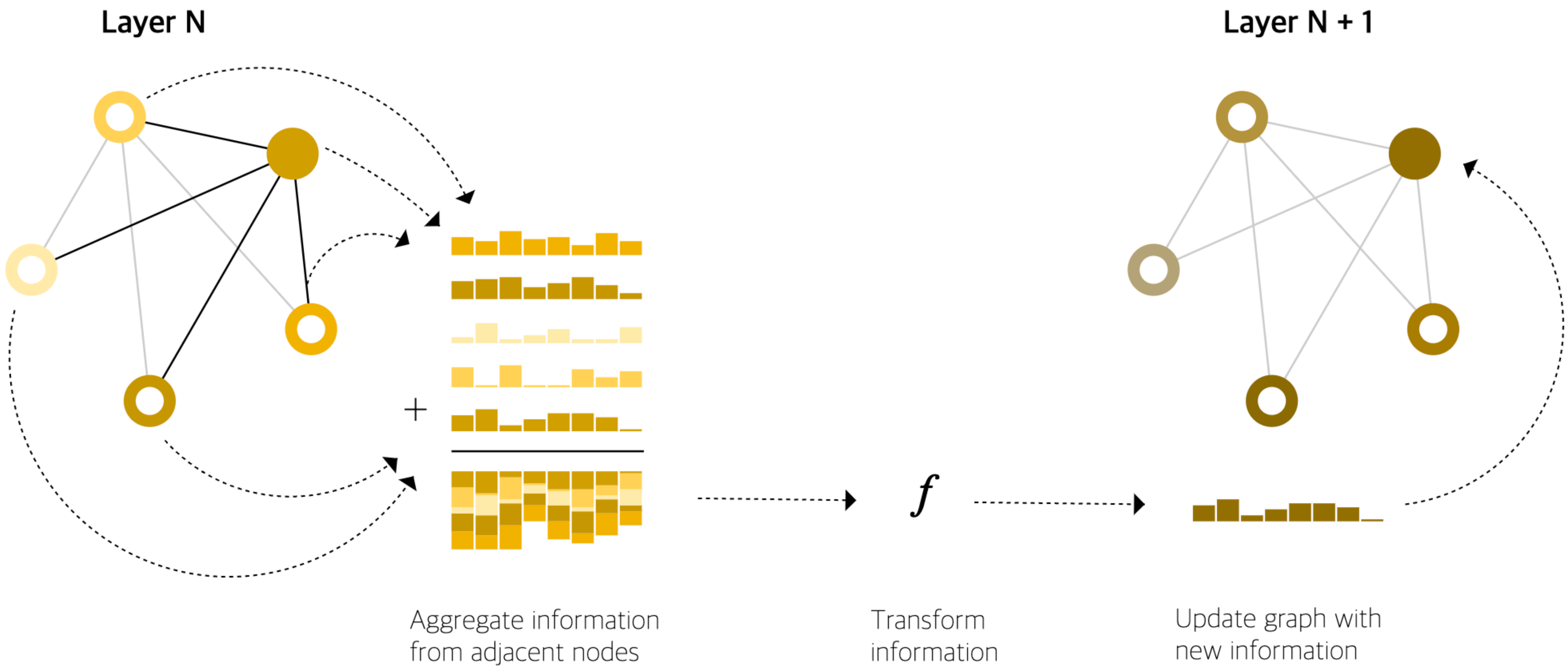
Adjacent nodes embedding
node embedding
Adjacent edges embedding
Global embedding
cond(x|z)
Updated
node embedding
f_{V_n}
update function f = , ...
conditioning function =
Concatenation,
Linear Layer and Add,
FiLM Layer..

Result

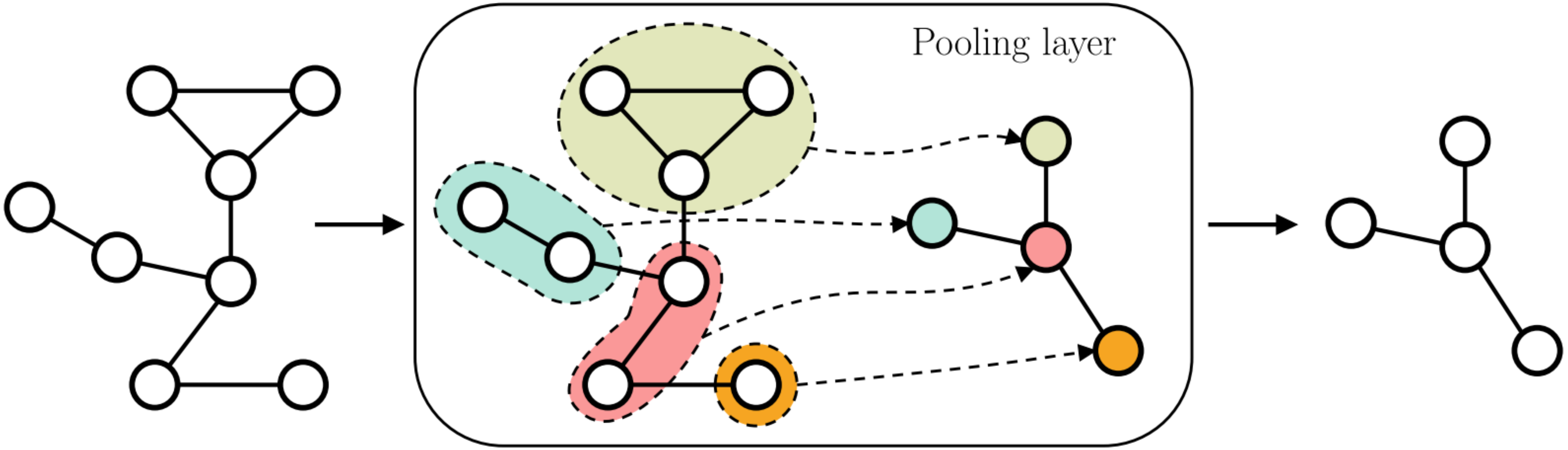


Architectures colored by style of Message passing
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
test AUC (pungent)
100
200
1,000
2,000
10,000
20,000
100,000
1,000,000
Number of parameters
test AUC (pungent)
Message passing
None
edges
nodes
globals
edges & globals
nodes & edges
nodes & globals
nodes & edges & globals

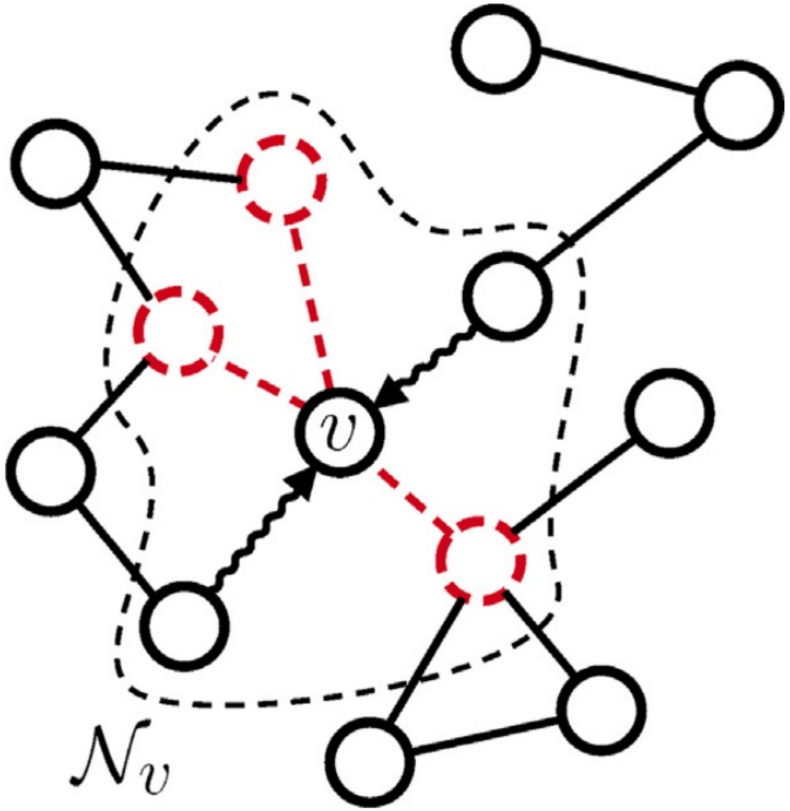
Update


Layer N
+
f
Layer N + 1
Aggregate information
from adjacent nodes
Transform
information
Update graph with
new information

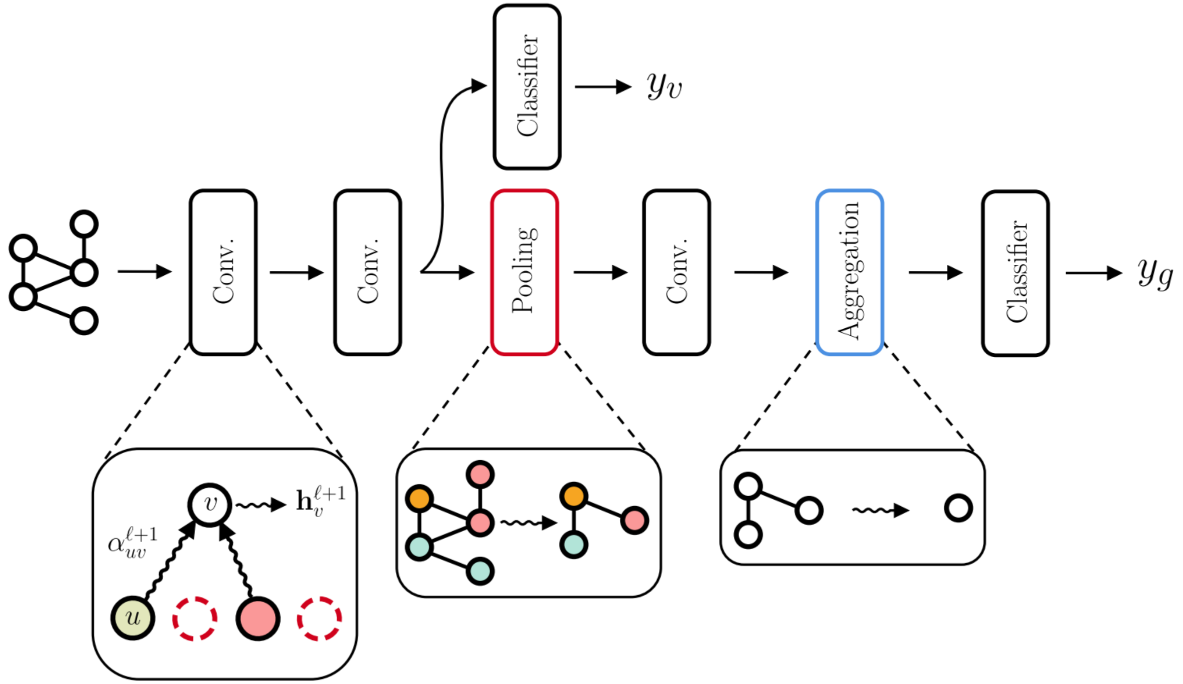
Some Techniques : Pooling

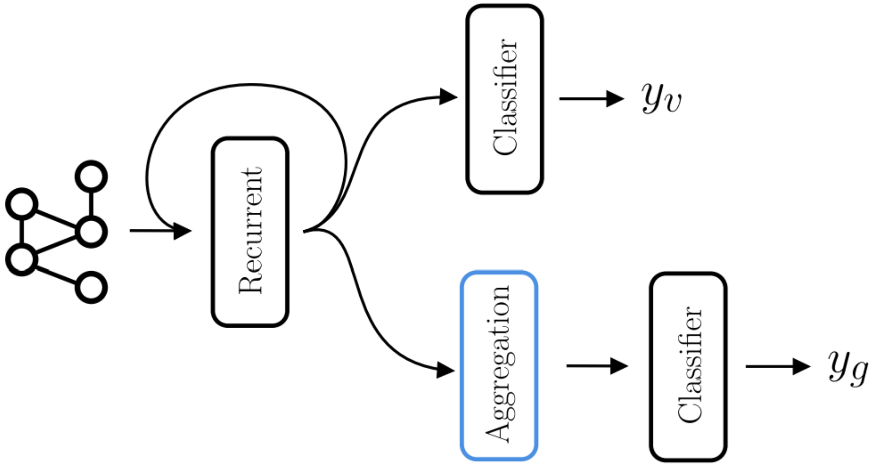


Some Techniques : Sampling

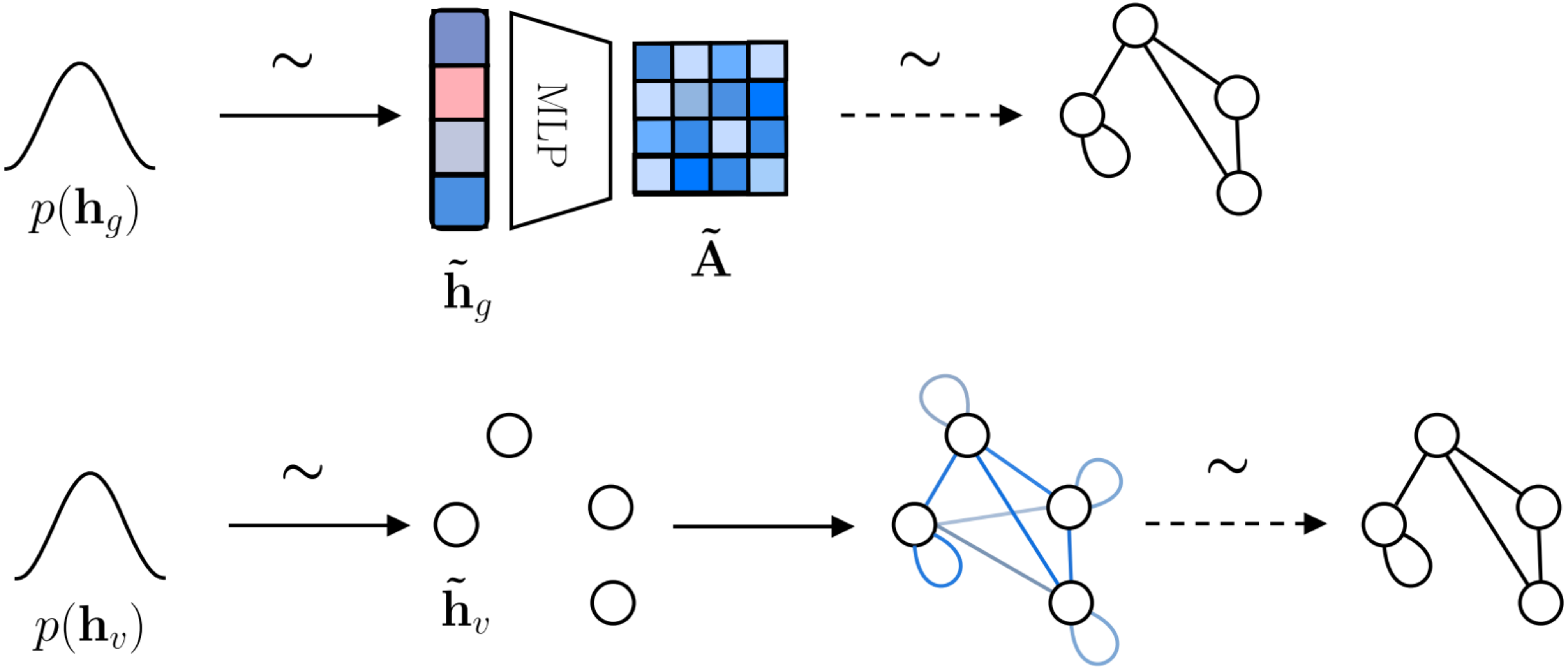


Recurrence VS Convolution

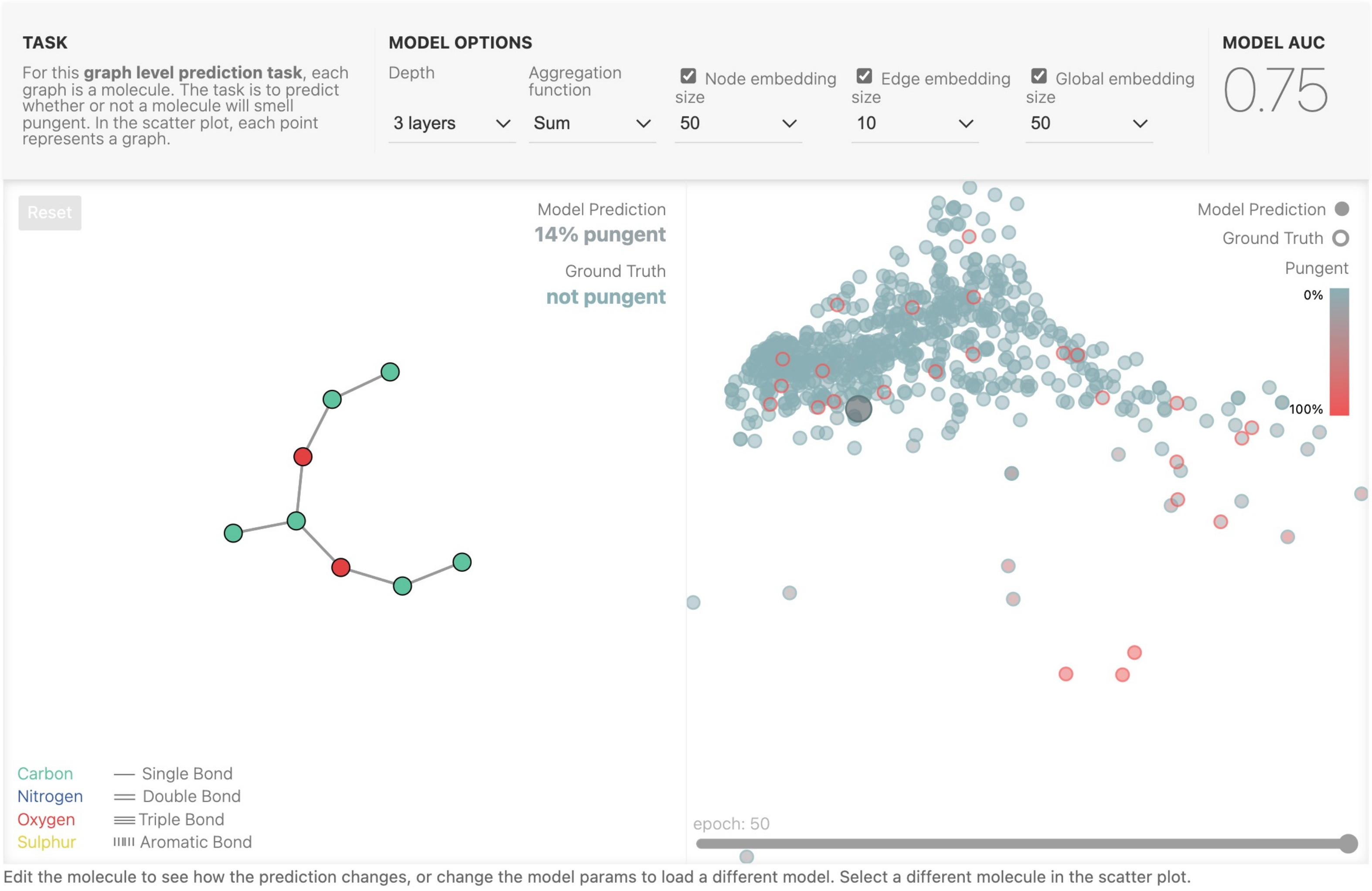




Generative Purpose



Playground!



https://distill.pub/2021/gnn-intro/#graph-to-tensor

Part4 Reference

Reference

1. Bacciu, Davide, et al. "A gentle introduction to deep learning for graphs." *Neural Networks* 129 (2020): 203-2

2. Sanchez-Lengeling, Benjamin, et al. "A gentle introduction to graph neural networks." *Distill* 6.9 (2021): e33.

3. Allamanis , Miltos. "An Introduction to Graph Neural Networks: Models and Applications." *YouTube*, uploaded by Microsoft Research, 9 May 2020, https://www.youtube.com/watch?v=zCEYiCxrL_0