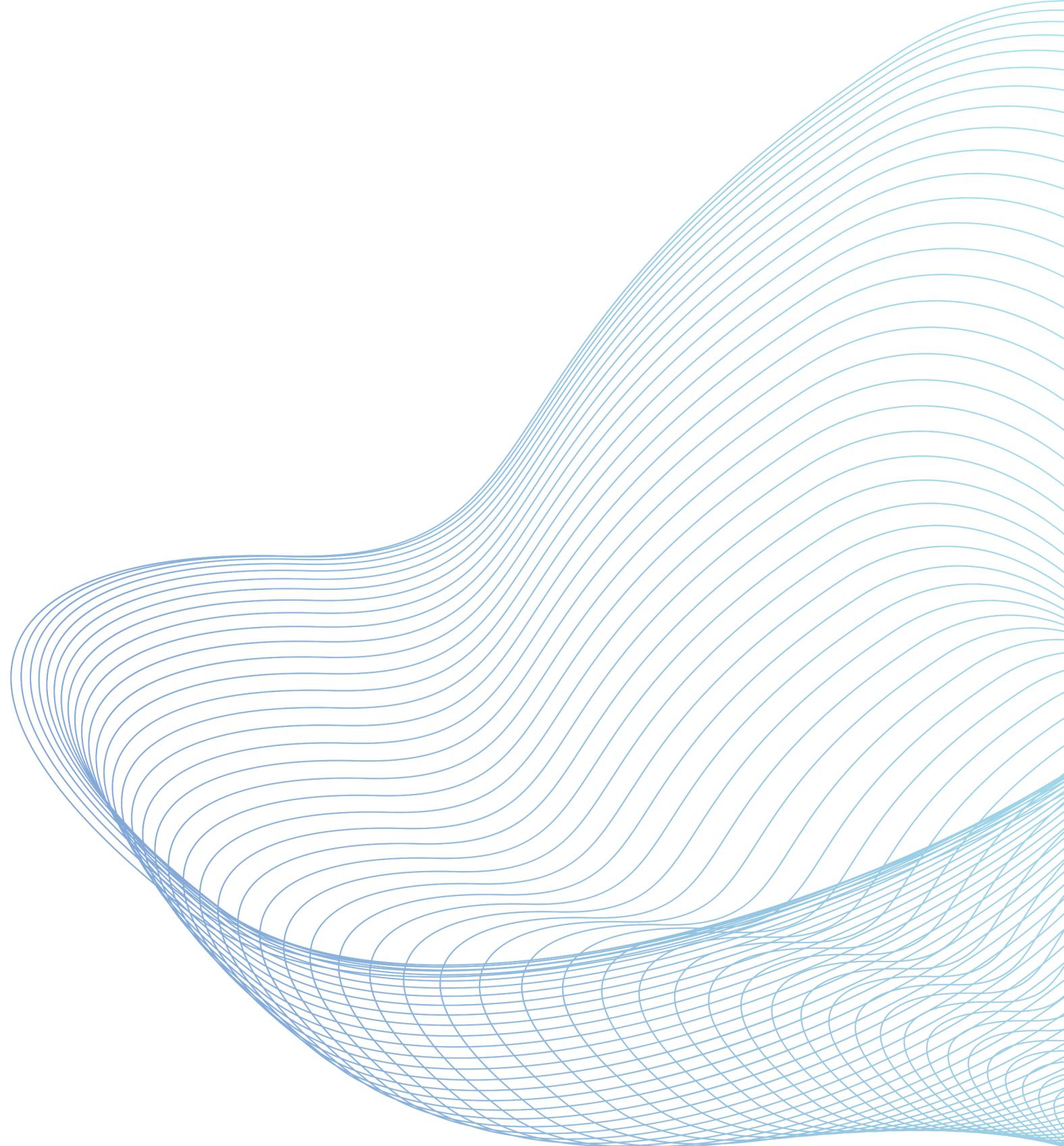


**VR487795**

Chiara Solito

# Beyond low-frequency Information in **GRAPH CONVOLUTIONAL NETWORKS**



[Bo et al., 2021] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 3950–3957, 2021.

# Table of contents

## Basics

- What are GNNs?
- Message Passing

## Problem

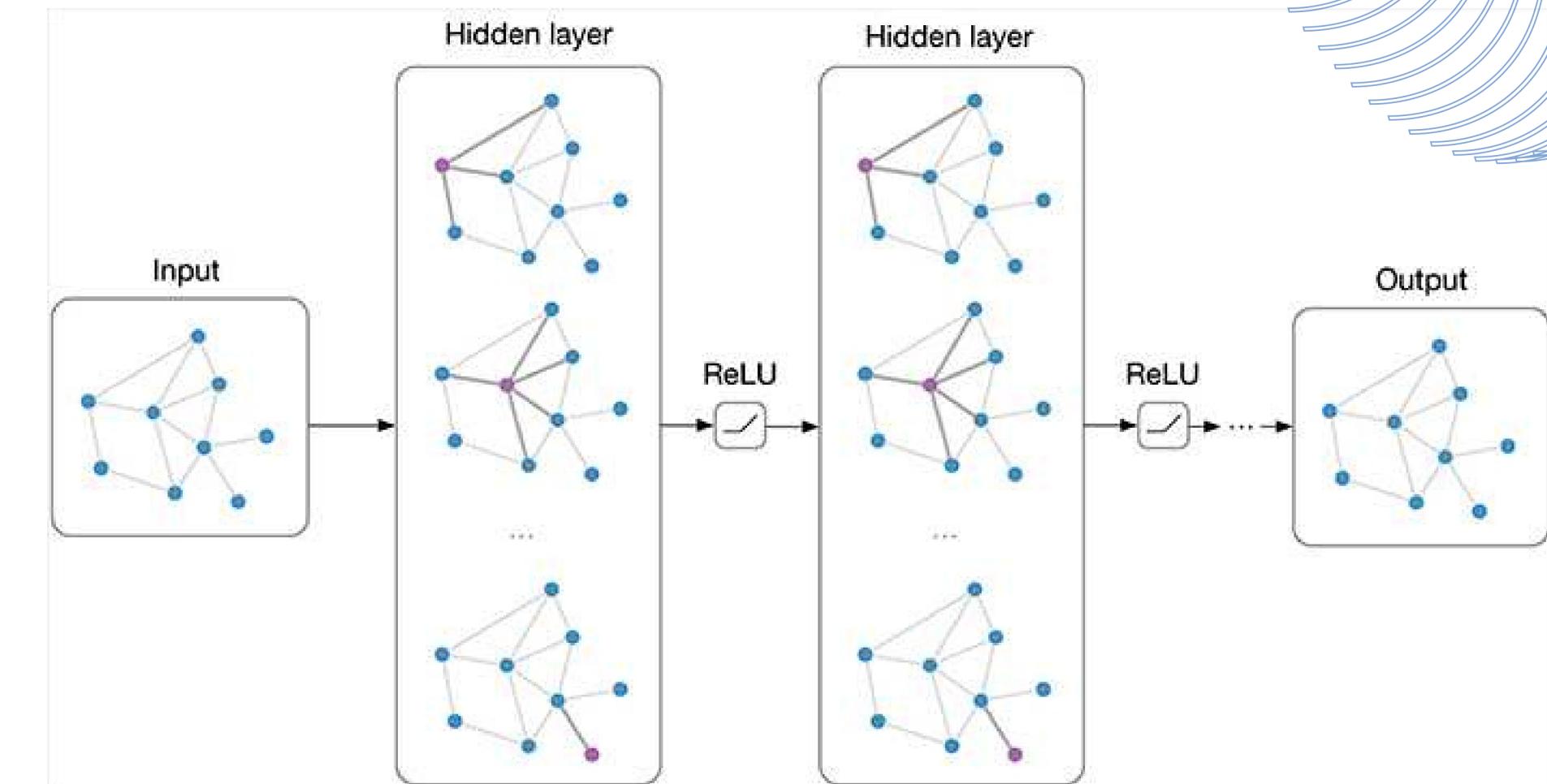
- Signal frequency in Graphs
- High-frequency information

## FAGCN

- Separation&Aggregation
- Architecture
- Experiments
- Conclusions

# BASICS

A Graph Neural Network is an optimizable transformation on all attributes of the graph (nodes, edges, global context) that preserves graph symmetries (permutation invariance).\*



\*Sanchez-Lengeling, et al., "A Gentle Introduction to Graph Neural Networks", Distill, 2021.



GNNs adopt a “**graph-in, graph-out**” architecture meaning that these model types accept a graph as input, with information loaded into its nodes, edges, and global context, and **progressively transform these embeddings, without changing the connectivity of the input graph.**\*

### **Graph-level task**

We predict a single property for a whole graph.

### **Node-level task**

We predict some properties for each node in a graph.

### **Edge-level task**

We want to predict the property or presence of edges in a graph.

\*Sanchez-Lengeling, et al., "A Gentle Introduction to Graph Neural Networks", Distill, 2021.

# MESSAGE PASSING

Message-passing type GNNs, also called Message Passing Neural Networks (MPNN)\*\*, propagate node features by exchanging information between adjacent nodes.\*

## Step 1

For each node in the graph, gather all the neighboring node embeddings (or messages), which is the **g** function.

## Step 2

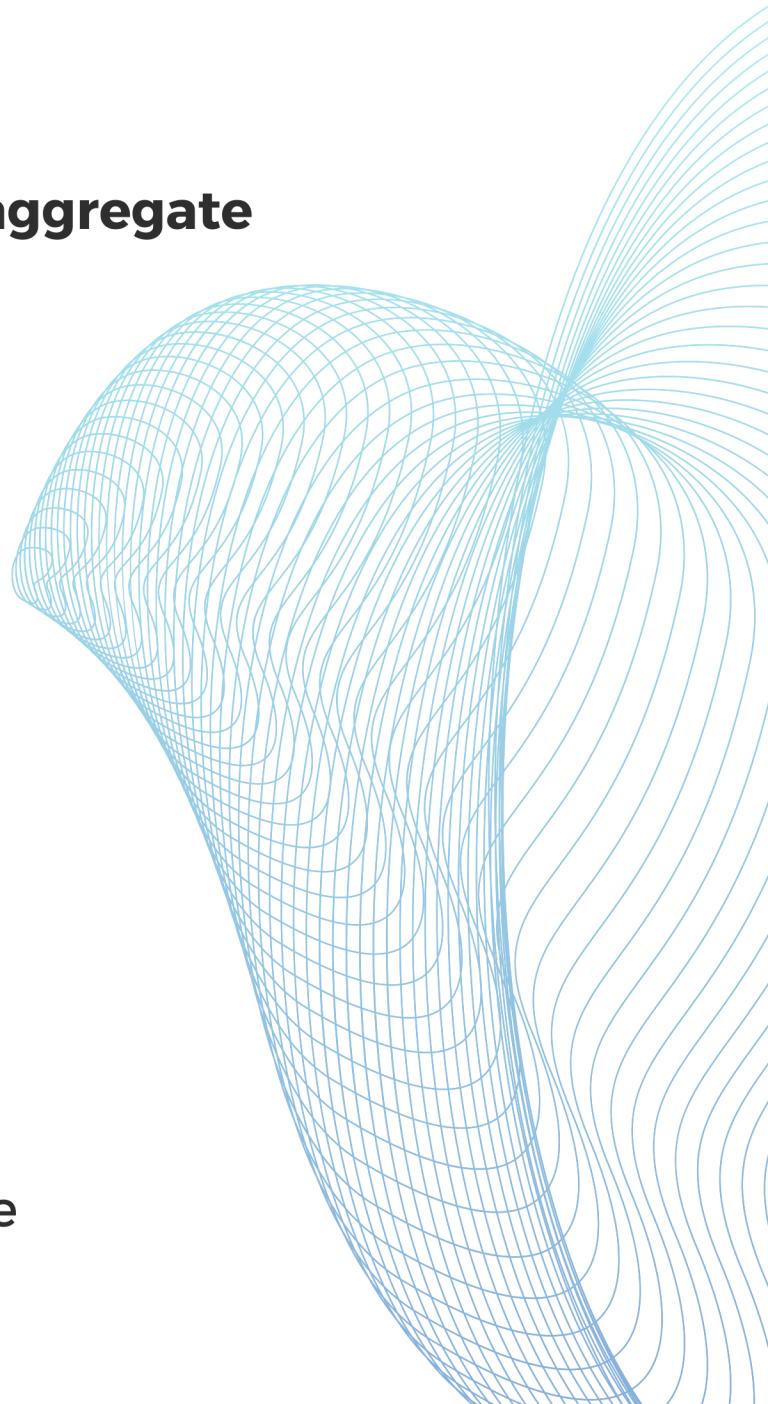
Aggregate all messages via an **aggregate function** (like the sum).

## Step 3

All pooled messages are passed through an **update function**, usually a learned neural network.\*\*

\*Michael Bronstein, "Beyond Message Passing, a Physics-Inspired Paradigm for Graph Neural Networks", The Gradient, 2022.

\*\*Gilmer, et al. "Neural message passing for Quantum chemistry", Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17). JMLR.org, 1263-1272. 2017.



# MESSAGE PASSING

**Classic update:**

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

where

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

During the message passing phase, hidden states  $h_t$ , at each node in the graph, are updated by the vertex update functions. This is based on message  $m_{t+1}$ , the sum of the message function  $M_t$  for each neighbor of node  $v^*$

# Graph signal processing

Classical signal processing tools developed in the Euclidean domain have been generalized to irregular domains such as graphs.

Most existing GNNs usually exploit the **low-frequency** signals of node features. In general, GNNs update node representations by aggregating information from neighbors, which can be seen as a **special form of low-pass filters**

Is the low-frequency information all we need? And **what roles does other information play in GNNs?**

# High-frequency Information

The high-frequency information, capturing the difference between nodes, may be more suitable for certain tasks.

Furthermore, node representation becomes indistinguishable when we always utilize low-pass filtering, causing over-smoothing. **We find that when a network exhibits dissorativity, high-frequency signals perform much better!**

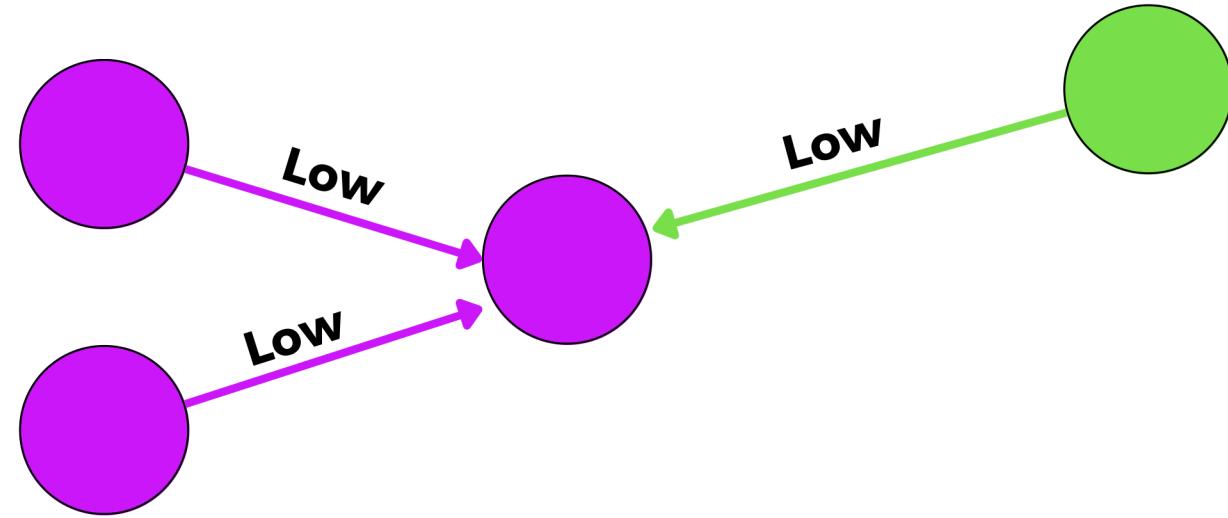
## Low-frequency Information Assortativity

The tendency of nodes to connect with other "similar" nodes over dissimilar ones.

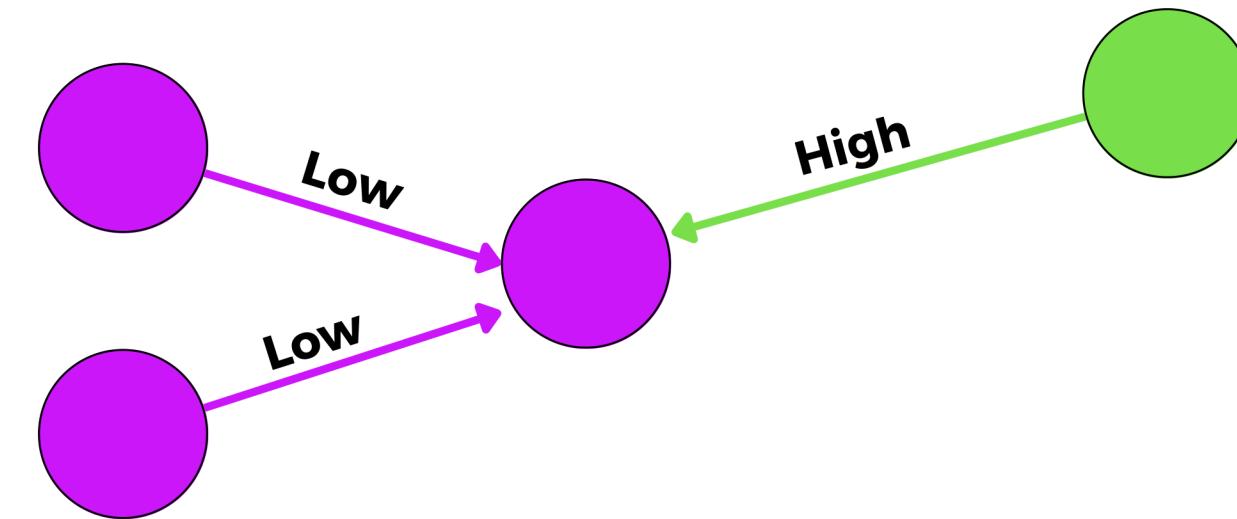
**Low-pass filters**  
Mainly retains the commonality of node features, ignoring the differences: the learned representation becomes similar.

# FAGCN

The proposed model, FAGCN, combines the advantages of both low-pass and high-pass filters. It can **aggregate the low-frequency signals of neighbors within the same class and the high-frequency signals of neighbors from different classes.**



Existing GNNs aggregate the low-frequency signals of neighbors.



FAGCN aggregates the low-frequency signals of neighbors within the same class and high-frequency signals of neighbors from different classes, where the color indicates the node label.

# THE MODEL

To understand how to make full use of both frequencies, we need to start from the **Graph Fourier Transform**. We can treat the eigenvectors of the Normalized Laplacian matrix as the bases in graph FT.

The convolutional  $*G$  between the signal  $x$  and convolution kernel  $f$  is:

$$f *_G x = U \left( (U^\top f) \odot (U^\top x) \right) = U g_\theta U^\top x$$

The convolutional kernel  $g_\theta$  is a diagonal matrix, which represents the convolutional kernel in the spectral domain, replacing  $U^\top f$ . GCN defines it as:

$$g_\theta = I - \Lambda$$

where  $I$  is the identity matrix and  $\Lambda$  is the diagonal of eigenvalues.

# SEPARATION

We present a low-pass filter  $\mathcal{F}_L$  and a high-pass filter  $\mathcal{F}_H$  to **separate signals from the node features**:

$$\mathcal{F}_L = \varepsilon I + D^{-1/2} A D^{-1/2} = (\varepsilon + 1)I - L$$

$$\mathcal{F}_H = \varepsilon I - D^{-1/2} A D^{-1/2} = (\varepsilon - 1)I + L$$

If we use these filters to replace the convolutional kernel  $f$  in the graph Fourier transform, calculating the convolutional  $*G$  between the signal  $x$  and convolution kernel  $f$ , the signal  $x$  is filtered as:

$$\mathcal{F}_L *_G x = U[(\varepsilon + 1)I - \Lambda]U^\top x = \mathcal{F}_L \cdot x,$$

$$\mathcal{F}_H *_G x = U[(\varepsilon - 1)I + \Lambda]U^\top x = \mathcal{F}_H \cdot x.$$

To avoid a negative amplitude we consider the second-order convolution.

# PROBLEMS

The concrete meaning of  $\mathcal{F}_L \cdot x$  is the sum of node features and neighborhood features in the spatial domain, while  $\mathcal{F}_H \cdot x$  is the difference.

Separating the frequencies provides a way to deal with different networks, but has **two main disadvantages**:

1. We don't actually know whether a network is assortative or disassortative beforehand
2. It requires matrix multiplication, undesirable for large graphs

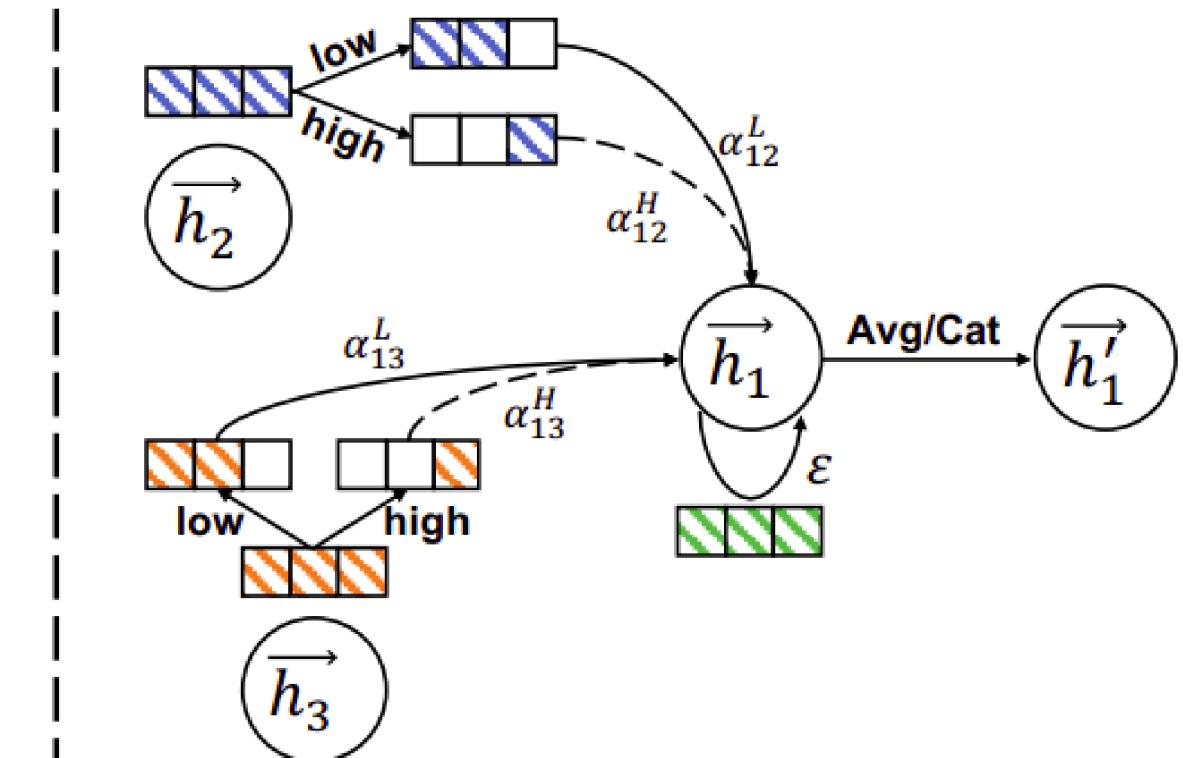
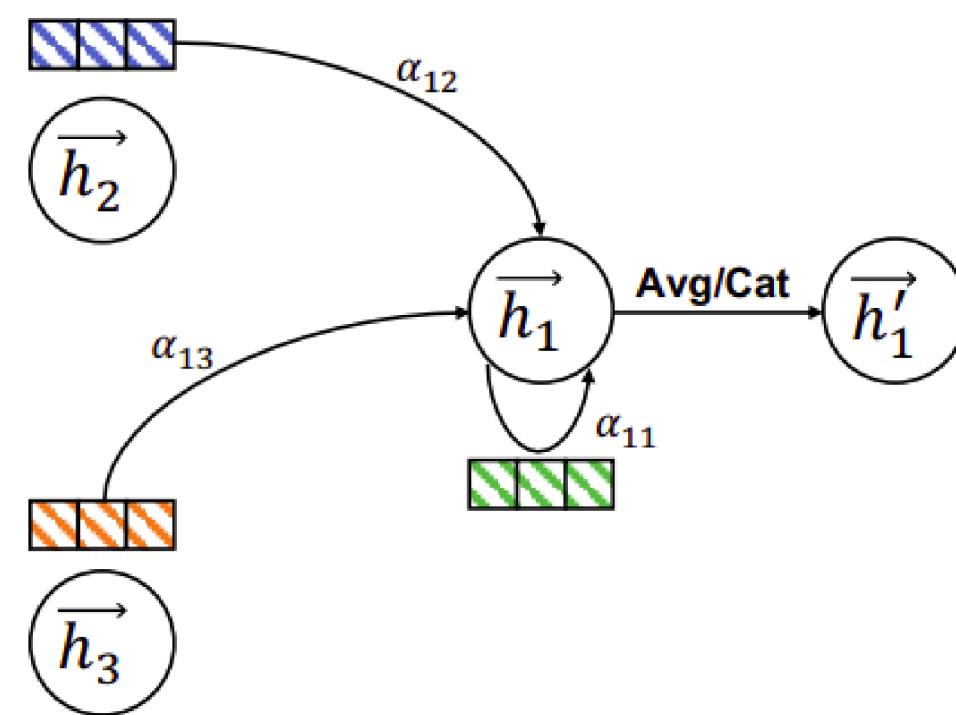
An efficient method that can adaptively aggregate signals is desired!



# AGGREGATION

GNNs consider learning the importance ( $\alpha_{ij}$ ) of each node in aggregation.

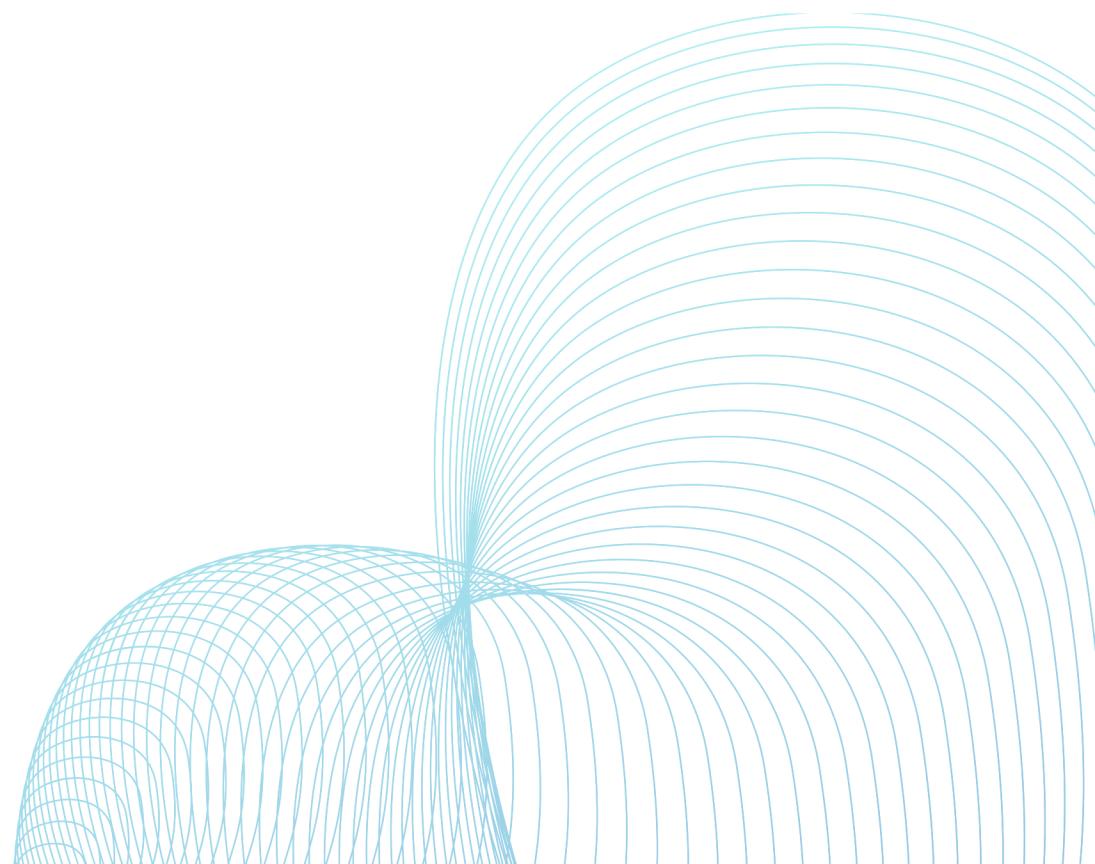
FAGCN uses two coefficients:  $\alpha_{ij}^L$  and  $\alpha_{ij}^H$  to aggregate low-frequency and high-frequency signals from the neighbors respectively.



## Aggregated representation of node i

Using the attention mechanism to learn the proportion of signals.

$$\tilde{\mathbf{h}}_i = \alpha_{ij}^L (\mathcal{F}_L \cdot \mathbf{H})_i + \alpha_{ij}^H (\mathcal{F}_H \cdot \mathbf{H})_i = \varepsilon \mathbf{h}_i + \sum_{j \in \mathcal{N}_i} \frac{\alpha_{ij}^L - \alpha_{ij}^H}{\sqrt{d_i d_j}} \mathbf{h}_j,$$



# AGGREGATION

Based on G we can calculate L and H and represent the coefficients of neighbors in aggregation:

- if G is greater than 0 it represents the sum of node features and neighborhood
- if it's smaller than 0 represents the difference between them
- when it's around 0 the contributions of neighbors will be limited, so the raw features will dominate the node representations.

We set:

$$\alpha_{ij}^L + \alpha_{ij}^H = 1$$

$$\alpha_{ij}^G = \alpha_{ij}^L - \alpha_{ij}^H$$

# SELF GATING MECHANISM

In order to learn the coefficients we need to consider the features of both the node itself and its neighbors, it is therefore proposed a shared self-gating mechanism:

**Learn the coefficient**

$$\alpha_{ij}^G = \tanh(\mathbf{g}^\top [\mathbf{h}_i \parallel \mathbf{h}_j])$$

We only calculate the coefficient among the node and it's first order neighbors  $\mathcal{N}_i$

**Aggregate the representation  
of neighbors**

$$\mathbf{h}'_i = \varepsilon \mathbf{h}_i + \sum_{j \in \mathcal{N}_i} \frac{\alpha_{ij}^G}{\sqrt{d_i d_j}} \mathbf{h}_j$$

# FAGCN ARCHITECTURE

Now that we have defined the message-passing process, we can introduce the architecture of the network.

## MLP

Since the entanglement of filters and weight matrices may be harmful to the performance and robustness of the model, to apply a non-linear transformation to the raw feature a MLP is used.

## Mathematical Expression

$$\mathbf{h}_i^{(0)} = \phi(\mathbf{W}_1 \mathbf{h}_i) \in \mathbb{R}^{F' \times 1}$$

$$\mathbf{h}_i^{(l)} = \varepsilon \mathbf{h}_i^{(0)} + \sum_{j \in \mathcal{N}_i} \frac{\alpha_{ij}^G}{\sqrt{d_i d_j}} \mathbf{h}_j^{(l-1)} \in \mathbb{R}^{F' \times 1}$$

$$\mathbf{h}_{out} = \mathbf{W}_2 \mathbf{h}_i^{(L)} \in \mathbb{R}^{K \times 1},$$

## Complexity

The complexity of a single layer is  $O((N + |E|) \times F')$ , approximately linear with the number of edges and nodes.

# Accuracies

	Assortative net			disassortative net		
	Cora	Citeseer	Pubmed	Chameleon	Squirrel	Actor
FAGCN original experiment	84.1+-0.5%	72.7+-0.8%	79.4+-0.3%	76.1%	66.7%	36%
FAGCN repeated experiment*	83.3%	72.12%	78.9%	73.46%	64.45%	
GCN original experiment	81.5%	70.3%	79.1%	72.3%	61.9%	30%
	83.7%	72.5%	80.5%	75%	65%	35%
	GraphHeat			Chebnet		

\*mean over 5 repetition

# AIDS DATASET

The AIDS data set consists of graphs representing molecular compounds. Graphs are constructed from the Antiviral Screen Database of Active Compounds. It consists of 2000 fully connected graphs that **we treat as a single, disconnected, graph**. The assortativity is, understandably, **-0.395**

## GRAPHS

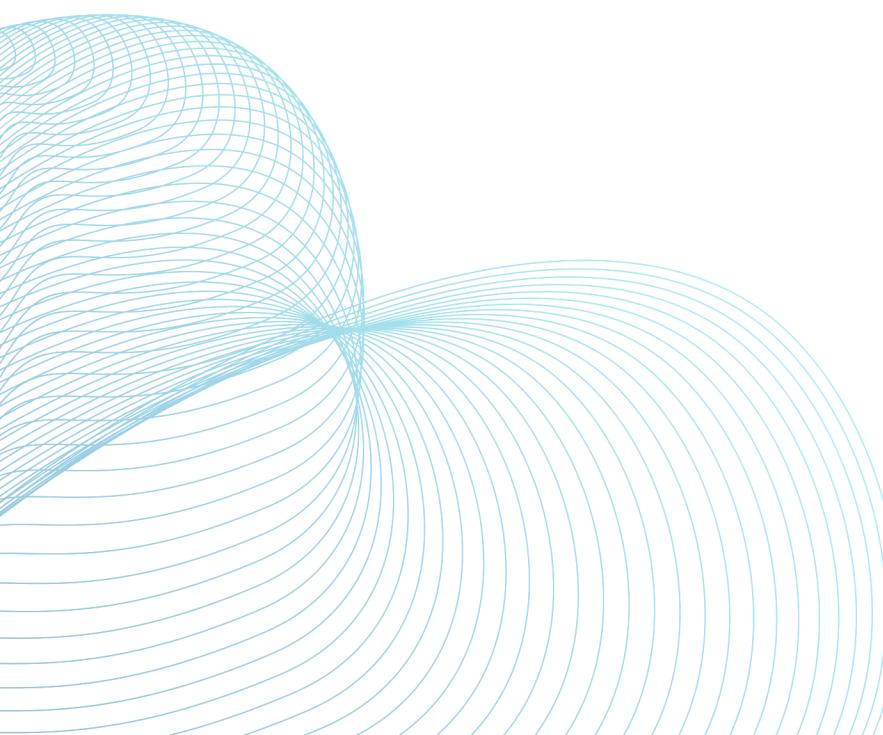
This data set consists of two classes (active and inactive), which represent molecules with activity against HIV or not. Since FAGCN performs node classification we do not care about this.

## NODES

Nodes represent atoms and are labeled with the number of the corresponding chemical symbol

## EDGES

Edges represent covalent bonds and are labeled by the valence of the linkage.



# ACCURACY

The accuracy for the aids dataset was calculated in the same way as the experiments repeated for others.

We run the training 5 times and report the mean accuracy: in this case with the hyperparameters used for the disassortative networks, with epsilon search space in {0.3, 0.4, 0.5}

Epsilon	Accuracy
0.3	62.83%
0.4	62.68%
0.5	62.68%

