

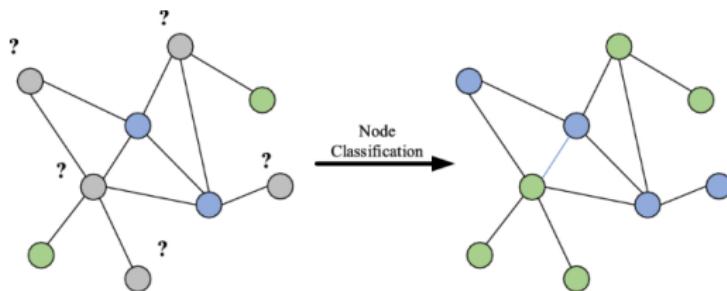
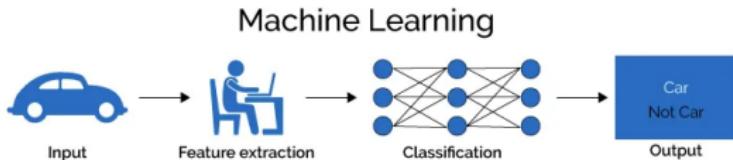
# Exploring Community-Based Graph Variables in NOGraph Neural Networks

Sebastian Zajac

20th Workshop on Modelling and Mining Networks 2025  
02.07.2023



# Motivation



**Node classification** is an important problem in which data is represented as a network, and the goal is to predict labels associated with its nodes.

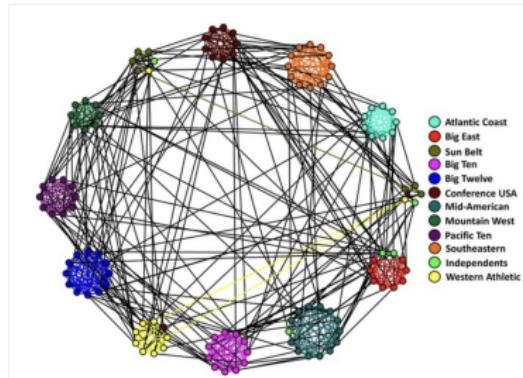
Practical applications: recommended systems, social network analysis, or applied chemistry.

# Classical Node features - Graph measures



abbreviation	name
lcc	local clustering coefficient
bc	betweenness centrality
cc	closeness centrality
dc	degree centrality
ndc	average degree centrality of neighbours
ec	eigenvector centrality
eccen	node eccentricity
core	node coreness
n2v	16-dimensional node2vec embedding
s2v	16-dimensional struc2vec embedding

# Community-aware features



Identifying communities in a network can be done in an unsupervised way and is often the analysts' first step.

We investigate a family of features that pay attention to **community structure** in complex networks.

## Community features already introduced in the literature:

A partition  $A = \{A_1, A_2, \dots, A_\ell\}$  of  $V$  into  $\ell$  communities.

Each part  $A_i$  ( $i \in [\ell]$ ) is denser comparing to the global density of the graph.  $\deg_{A_i}(v)$ : the number of neighbors of  $v$  in  $A_i$

### Participation coefficient of a node $v$

$$p(v) = 1 - \sum_{i=1}^{\ell} \left( \frac{\deg_{A_i}(v)}{\deg(v)} \right)^2$$

### Anomaly score CADA

$$\text{cd}(v) = \frac{\deg(v)}{d_A(v)}$$

where  $d_A(v)$  is a max number of neighboring nodes that belong to the same community.

T.J. Helling et al, A community-aware approach for identifying node anomalies in complex networks. 2019

## Community-aware features

Normalized anomaly score for a node  $v$

$$\overline{cd}(v) = \frac{\deg_{A_i}(v)}{\deg(v)}$$

Normalized within-module degree of a node  $v$

$$z(v) = \frac{\deg_{A_i}(v) - \mu(v)}{\sigma(v)}$$

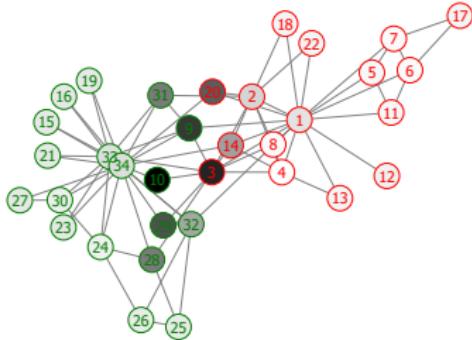
where  $\mu(v)$  and  $\sigma(v)$  are, respectively, the mean and the standard deviation of  $\deg_{A_i}(u)$  over all nodes  $u$  in the part  $v$  belongs to.

# Our Community-aware features

## Community Association Strength

For any  $v \in A_i$ , we define the *community association strength* as follows:

$$\beta^*(v) = 2 \left( \frac{\deg_{A_i}(v)}{\deg(v)} - \lambda \frac{\text{vol}(A_i) - \deg(v)}{\text{vol}(V)} \right).$$



The lower the value of  $\beta^*(v)$ , the less associated node  $v$  with its community is.

## Community-aware node features used in our experiments.

abbreviation	symbol	name
CADA	$cd(v)$	anomaly score CADA
CADA*	$\overline{cd}(v)$	normalized anomaly score
WMD	$z(v)$	normalized within-module degree
CPC	$p(v)$	participation coefficient
CAS	$\beta^*(v)$	community association strength
CD_L11	$L_1^1(v)$	$L^1$ norm for the 1st neighbourhood
CD_L21	$L_1^2(v)$	$L^2$ norm for the 1st neighbourhood
CD_KL1	$kl_1(v)$	Kullback–Leibler div. 1st neigh.
CD_HD1	$h_1(v)$	Hellinger distance for the 1st neighbourhood
CD_L12	$L_2^1(v)$	$L^1$ norm for the 2nd neighbourhood
CD_L22	$L_2^2(v)$	$L^2$ norm for the 2nd neighbourhood
CD_KL2	$kl_2(v)$	Kullback–Leibler div. for the 2nd neihb.
CD_HD2	$h_2(v)$	Hellinger distance for the 2nd neighbourhood

# Julia's Library

## GraphFeatures

```
using Pkg  
Pkg.add(PackageSpec(url="https://github.com/sebkaz/GraphFeatures.jl"))
```

### Karate Example

```
using Graphs  
using GraphFeatures  
g = Graphs.SimpleGraphs.smallgraph(:karate)  
comm = [1,1,1,1,2,2,2,1,3,3,2,1,1,1,3,3,2,1,3,1,3,1,3,4,4,4,3,4,4,3,3,4,3,3]  
  
df = compute_features(g, comm)
```

Row	node_id	participation	beta_star	edge_contr	cada	kl
	Int64	Float64	Float64	Float64	Float64	Float64
1	1	0.460938	0.342949	0.625	1.6	0.371125
2	2	0.802469	0.561966	0.888889	1.125	0.769965
3	3	0.38	0.179487	0.5	2.0	0.193936
4	4	1.0	0.653846	1.0	1.0	1.12847
5	5	0.555556	0.583333	0.666667	1.5	1.01759
6	6	0.625	0.673077	0.75	1.33333	1.15747
7	7	0.625	0.673077	0.75	1.33333	1.15747
8	8	1.0	0.641026	1.0	1.0	1.12847
9	9	0.52	0.273077	0.6	1.66667	0.403247
10	10	0.5	0.153846	0.5	2.0	0.391812
11	11	0.555556	0.583333	0.666667	1.5	1.01759

# Experimental data

## Syntetic networks ABCD + o

Artificial Benchmark for Community Detection with Outliers

4 networks with different level of noise:  $\chi \in \{0.3, 0.4, 0.5, 0.6\}$

$N = 10,000$  nodes with 1,000 of outliers. The node's degree distr with a power-law with exponent  $\gamma = 2.5$  and degrees between 5 and 500. The community's distr with a power-law with exponent  $\beta = 1.5$  and size range from 50 to 2,000.

## Empirical Graphs

dataset	# of nodes	avg deg	# of clust	target
Reddit	10,980	14.30	12	3.661%
Grid	13,478	2.51	78	0.861%
LastFM	7,624	7.29	28	20.619%
Facebook	22,470	15.20	58	25.670%
Amazon	9,314	37.49	39	8.601%

# Experiment 1: Information overlap

The source code allowing for reproduction of all results is available at

<https://github.com/sebkaz/BetaStar>

This experiment aims to show that classical features cannot entirely explain community-aware features.

Models: Regression kind

Linear regression, Ridge regression, Random forest, XGBoost, Lightgbm.

Measures:

Kendall correlation, spearman correlation, and  $R^2$  score.

The conclusion is that it is worth including such features in predictive models as they could improve their predictive power. However, this additional information could be noise and not valuable for practice.

## Results 1 - ABCD +o

target	$\xi = 0.3$	$\xi = 0.4$	$\xi = 0.5$	$\xi = 0.6$
CADA	0.3305	0.2541	0.2292	0.1766
CADA*	0.3613	0.2877	0.2772	0.1713
CPC	0.3540	0.3568	0.3231	0.3106
CAS	0.4205	0.3584	0.3138	0.2167
CD_L21	0.4539	0.4043	0.3823	0.3313
CD_L22	0.6265	0.5589	0.5009	0.4492
CD_L11	0.5935	0.5571	0.5834	0.5648
CD_L12	0.6503	0.5799	0.5464	0.5188
CD_KL1	0.6991	0.6411	0.5918	0.4929
CD_HD1	0.6809	0.6334	0.6170	0.5584
CD_KL2	0.7453	0.6602	0.6090	0.5471
CD_HD2	0.7546	0.7119	0.6815	0.6352
WMD	0.7670	0.7288	0.6915	0.6387

## Results 1 - Empirical graph

target	Amazon	Facebook	Grid	LastFM	Reddit
CADA	0.5830	0.5666	0.2156	0.4815	0.6826
CADA*	0.6058	0.5828	0.2174	0.5058	0.6867
CPC	0.6338	0.5992	0.2193	0.5175	0.7193
CAS	0.6538	0.6257	0.2999	0.5594	0.7306
CD_L21	0.7052	0.6464	0.3496	0.5698	0.7574
CD_L22	0.7554	0.7355	0.3557	0.6295	0.7941
CD_L11	0.7251	0.7041	0.6978	0.6220	0.7735
CD_L12	0.7794	0.7785	0.6447	0.6884	0.7810
CD_KL1	0.7176	0.7516	0.7394	0.6289	0.7755
CD_HD1	0.7383	0.7482	0.7168	0.6459	0.7853
CD_KL2	0.7706	0.7826	0.7292	0.6853	0.8097
CD_HD2	0.8212	0.8173	0.6930	0.7369	0.8221
WMD	0.8447	0.8456	0.8488	0.8531	0.7638

## Experiment 2: one-way predictive power

Target: Verify the usefulness of the community-aware features for the node classification task. For each graph, we build a single model predicting the target variable.

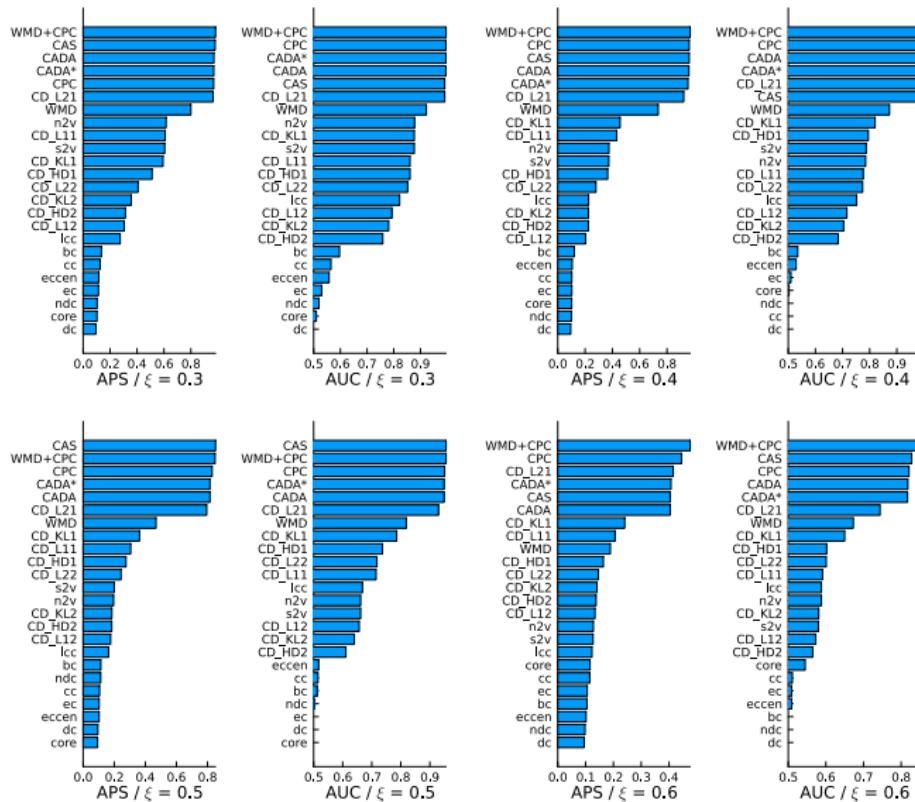
Models: Classification kind

Logistic regression, Random forest, XGBoost, Lightgbm.

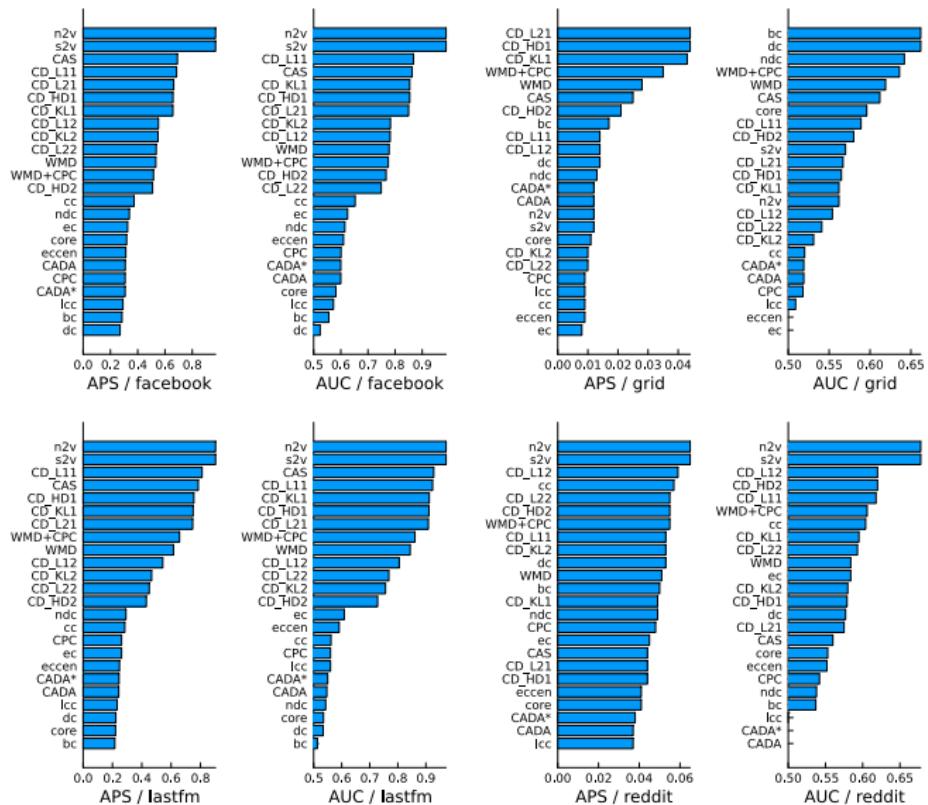
Measures:

ROC AUC score, Average Precision Score.

# ABCD + o Results

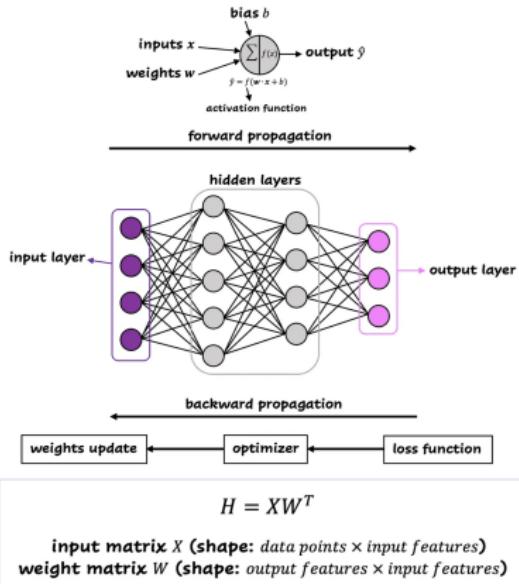


# Empirical Graphs Results



# Neural networks

## MLP



## GCN

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

Adjacency matrix + identity matrix = adapted adjacency matrix.

$$H = \hat{A}^T X W^T$$

$$\text{normalized } \hat{A} = \hat{D}^{-\frac{1}{2}} \hat{A}^T \hat{D}^{-\frac{1}{2}}$$

$$H = \hat{D}^{-\frac{1}{2}} \hat{A}^T \hat{D}^{-\frac{1}{2}} X W^T$$

## Experiment - Cora Dataset

The Cora dataset contains scientific publications classified into topics, along with citation links between them. Each node represents a publication, and edges represent citation relationships.

Property	Value
# of nodes	2,708
# of edges	5,429
# of features	1433 , +20
# of classes	7
# of communities	Louvain: 101 (modularity = 0.810) Leiden: 108 (modularity = 0.823)

All small communities are connected together to form 10 main communities.

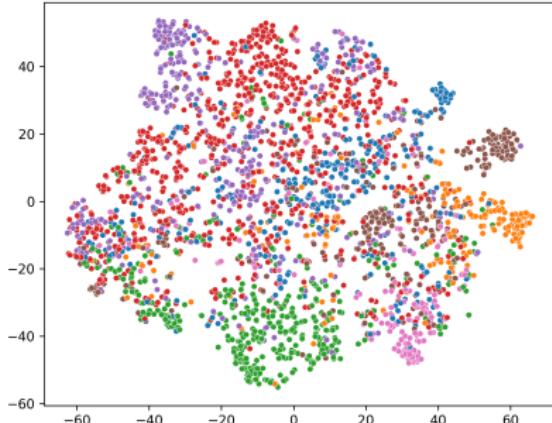
# Results

For each model i create 10 iteration with 100 epochs.

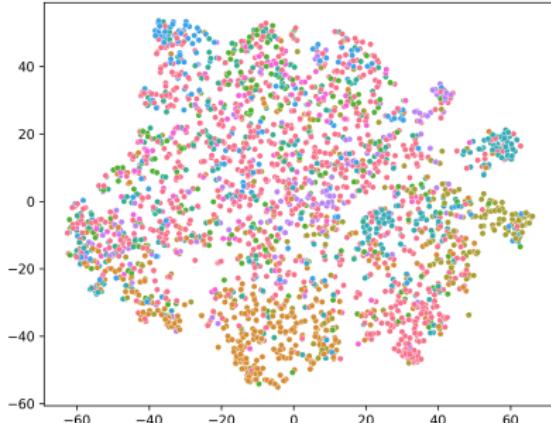
## Node features

Model	Accuracy
MLP	$0.53 \pm 0.01$
GCN	$0.79 \pm 0.01$
GAT	$0.79 \pm 0.01$
GraphSAGE	$0.78 \pm 0.01$

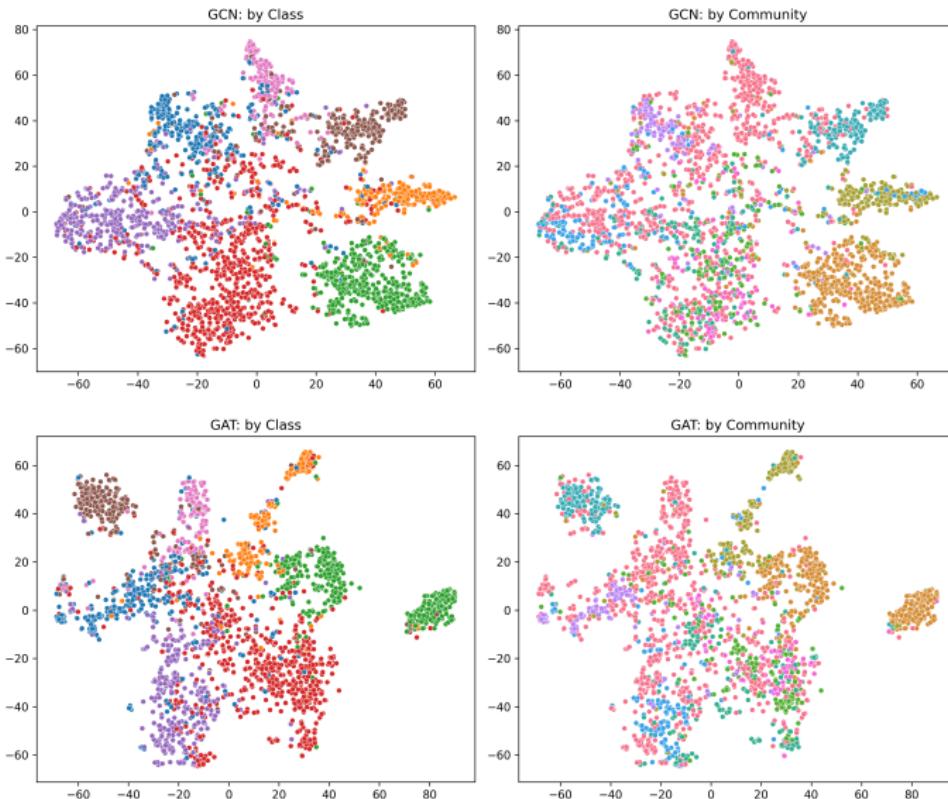
MLP: by Class



MLP: by Community



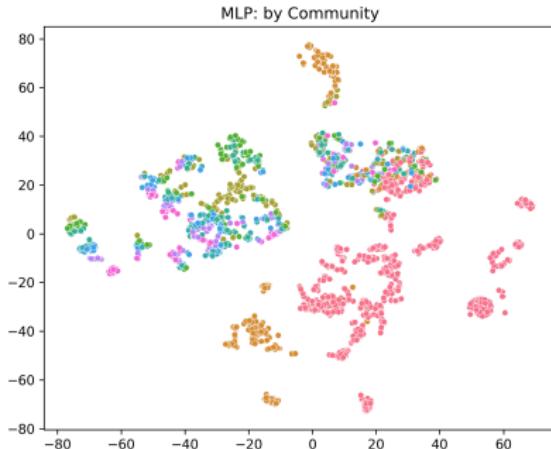
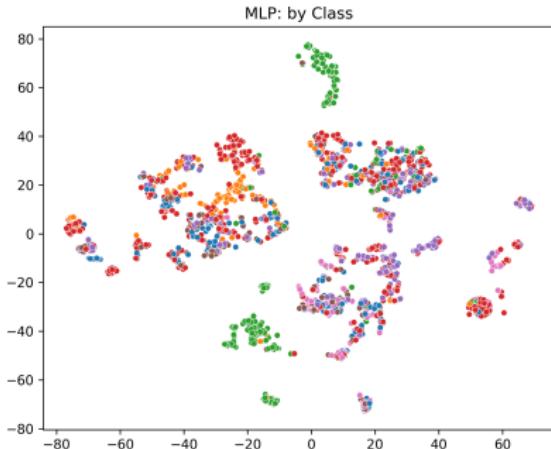
# Node Features



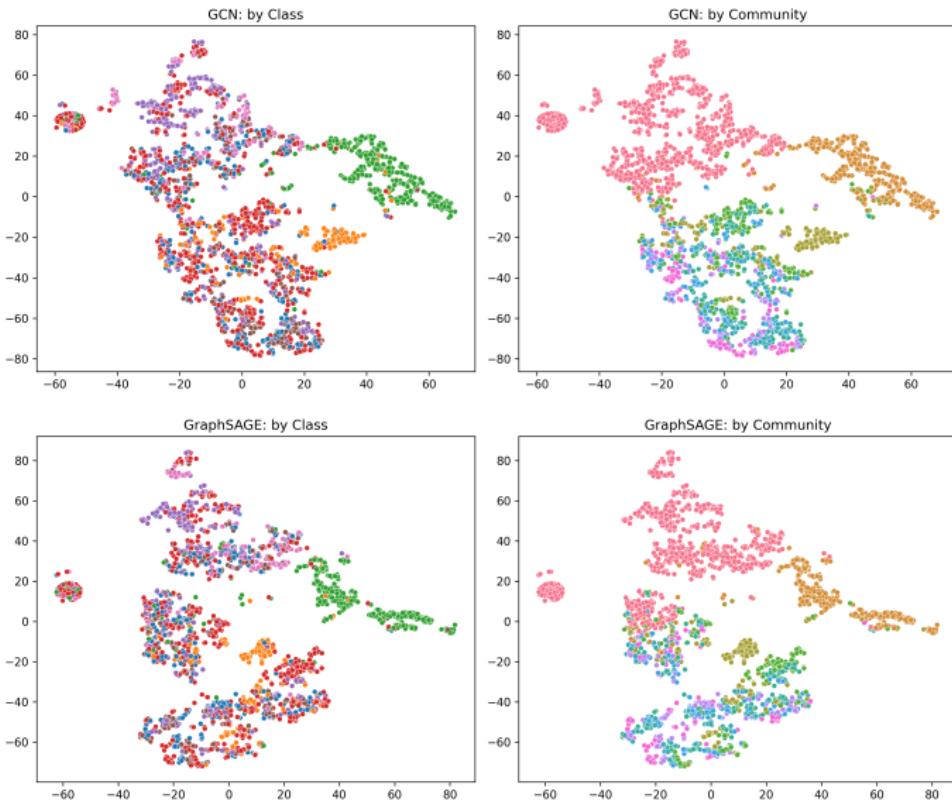
# Results

## Community-Aware Features

Model	Accuracy
MLP	$0.36 \pm 0.01$
GCN	$0.41 \pm 0.01$
GAT	$0.41 \pm 0.01$
GraphSAGE	$0.42 \pm 0.02$



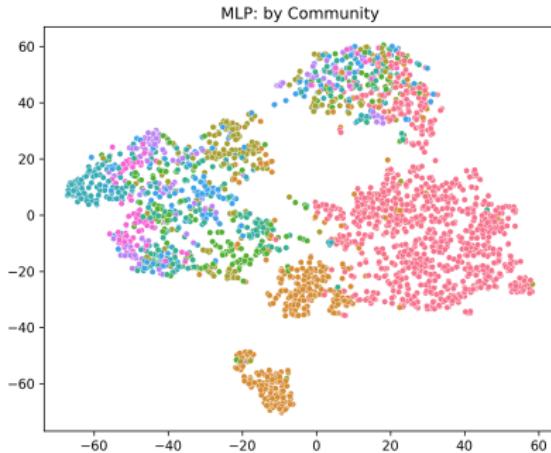
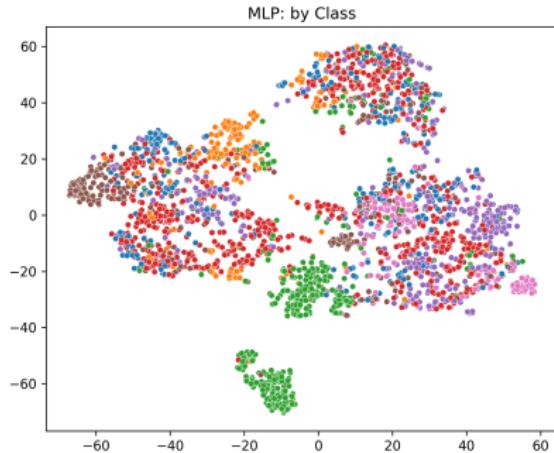
# Community Aware Features



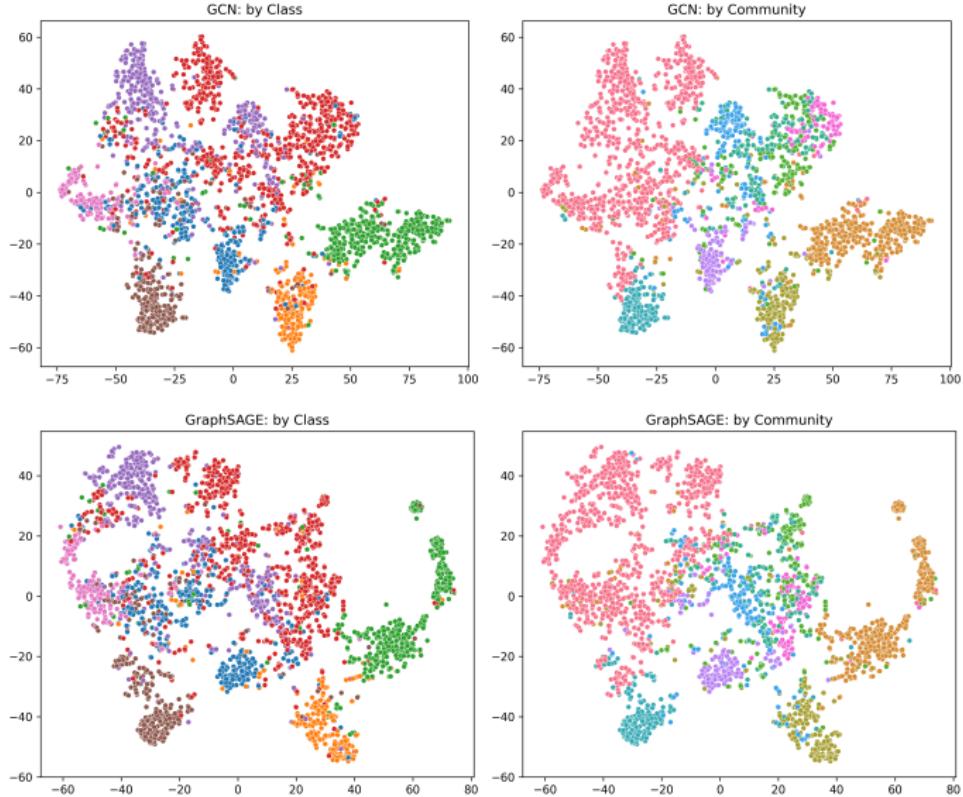
# Results

## Node + Community Aware features

Model	Accuracy
MLP	$0.52 \pm 0.01$
GCN	$0.78 \pm 0.01$
GAT	$0.75 \pm 0.02$
GraphSAGE	$0.76 \pm 0.01$



# Node + Community Aware Features

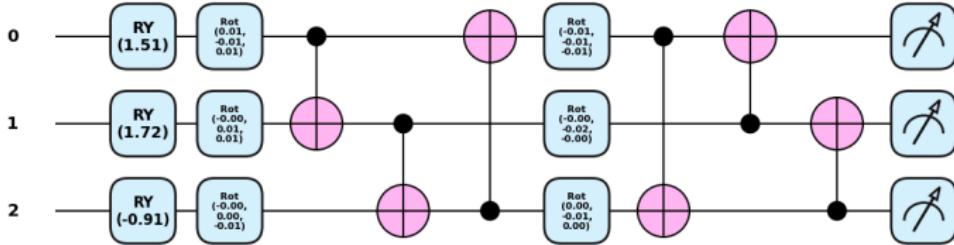


## Main results

We verify that:

- ① the information in the proposed features is non-redundant against classical node features and against node embeddings
- ② the proposed features have predictive power in node classification tasks.
- ③ We propose a new set of community-aware node features

# Hybrid QNN



HybridGCNQuantumNN Test Acc: 0.3660

# Summary

Thanks for Your Attention!  
sebastian.zajac@sgh.waw.pl