CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

ICML International Conference On Machine Learning

Fixed Aggregation Features Can Rival GNNs

Celia Rubio-Madrigal, Rebekka Burkholz

July 2026 — ICML

Questions for today

Efficiency

Are we adding complexity unnecessarily?

Benchmarking

Are we polishing simple baselines enough?

Simplicity

When do simple models win in practice?

Graph data: table + connections

Node features F

	GDP	#
v_1	x_{11}	x_{12}
v_2	x_{21}	x_{22}
v_3	x_{31}	x_{32}
v_4	x_{41}	x_{42}
v_5	x_{51}	x_{52}

one row per node

Adjacency A

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

who is connected to whom

↑ How to use this data? ↑

Model: GNN

Graph Neural Network (GNN)

Learns representations by *iteratively* aggregating information from neighbors $N(\cdot)$

$$h_i^{(\ell+1)} = \text{UPDATE}^{(\ell)}\left(h_i^{(\ell)},\ \text{AGGREGATE}^{(\ell)}\left(\left\{h_j^{(\ell)} : j \in N(i)\right\}\right)\right)$$

4-layer GCN: AC-AC-AC-AC

L-hop neighborhood

$h_i^{(L)}$ contains information from the L-hop neighborhood of node i.

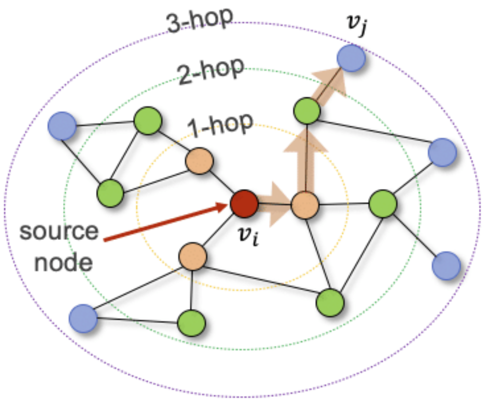


Figure from *"Understanding graph embedding methods and their applications"*

Graph tabularization

Can we **not learn** the graph part of the process?
Can we *tabularize* the graph without learning?

$\Longrightarrow$ It is theoretically viable, and
$\Longrightarrow$ It is well-performing for **the vast majority** of datasets

How much may we lose (expressiveness)
vs. gain (efficiency, interpretability)?

Related work

Classic GNNs can empirically compete with Graph Transformers

- With proper **hyperparameter tuning** & optimization techniques.

"Classic GNNs are strong baselines: Reassessing GNNs for node classification." (Luo et al., 2024)

Related work

Classic GNNs can empirically compete with Graph Transformers

- With proper **hyperparameter tuning** & optimization techniques.

"Classic GNNs are strong baselines: Reassessing GNNs for node classification." (Luo et al., 2024)

Efficiency has inspired pre-computing the aggregation

- Performance loss is often assumed in exchange of **scalability**.

"Simplifying Graph Convolutional Networks" (Wu et al., 2019)

"SIGN: Scalable Inception Graph Neural Networks" (Frasca et al., 2020)

Related work

Classic GNNs can empirically compete with Graph Transformers

- With proper **hyperparameter tuning** & optimization techniques.

"Classic GNNs are strong baselines: Reassessing GNNs for node classification." (Luo et al., 2024)

Efficiency has inspired pre-computing the aggregation

- Performance loss is often assumed in exchange of **scalability**.

"Simplifying Graph Convolutional Networks" (Wu et al., 2019)

"SIGN: Scalable Inception Graph Neural Networks" (Frasca et al., 2020)

GNN expressiveness is over neighborhood multiset functions

- Derives GIN, a classic architecture that is maximally expressive over them.

"How Powerful are Graph Neural Networks?" (Xu et al., 2018)

What does it take to preserve information?

Color code: Aggregation Combination Neighborhood function

Kolmogorov-Arnold theorem (Schmidt-Hieber et al., 2021)

For dimension $d \geq 2$, there exists a monotone function $f : [0, 1] \to C$ (the Cantor set) such that

What does it take to preserve information?

Color code: Aggregation Combination Neighborhood function

Kolmogorov-Arnold theorem (Schmidt-Hieber et al., 2021)

For dimension $d \geq 2$*, there exists a monotone function* $f : [0, 1] \to \mathcal{C}$ *(the Cantor set) such that any function* $g : [0, 1]^d \to \mathbb{R}$ *can be decomposed as*

What does it take to preserve information?

Color code: Aggregation Combination Neighborhood function

Kolmogorov-Arnold theorem (Schmidt-Hieber et al., 2021)

For dimension $d \geq 2$, there exists a monotone function $f : [0,1] \to C$ (the Cantor set) such that any function $g : [0,1]^d \to \mathbb{R}$ can be decomposed as

$$g(x_1, \ldots, x_d) = \varphi\left(3 \sum_{p=1}^{d} 3^{-p} f(x_p)\right)$$

for some univariate function $\varphi : C \to \mathbb{R}$.

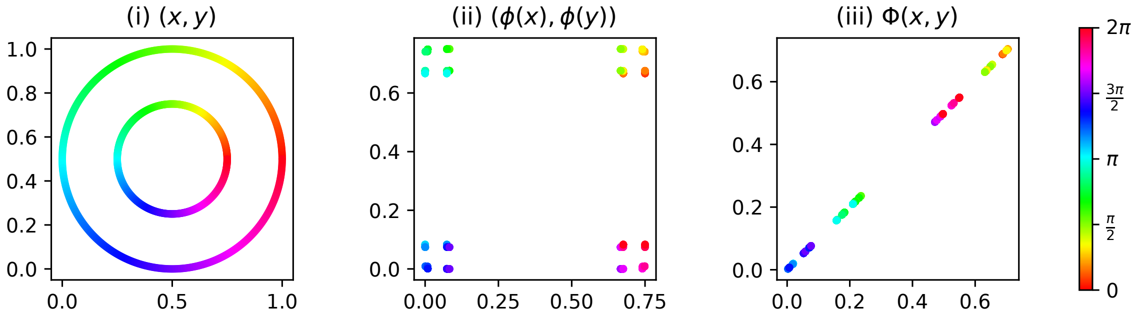
Moreover, if g is continuous, then also φ is continuous.

Aggregation is fixed for every neighborhood function!

- We can aggregate neighborhood information as a preprocessing step.

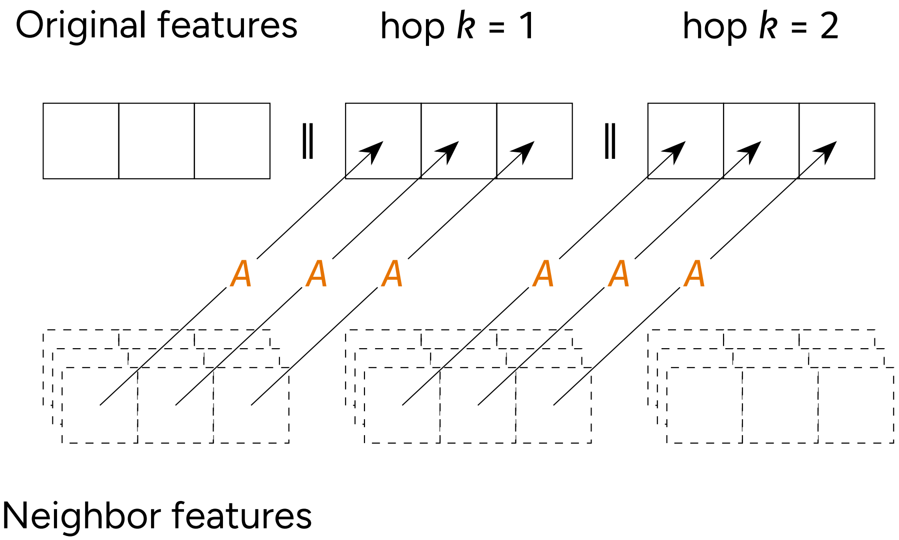
e.g., 4-hop: AAAA-CCCC, as long as A is an **injective** aggregation.

- In practice, the K-A aggregation is not well-behaved.
- We often rely on theorems that do not translate to practice!



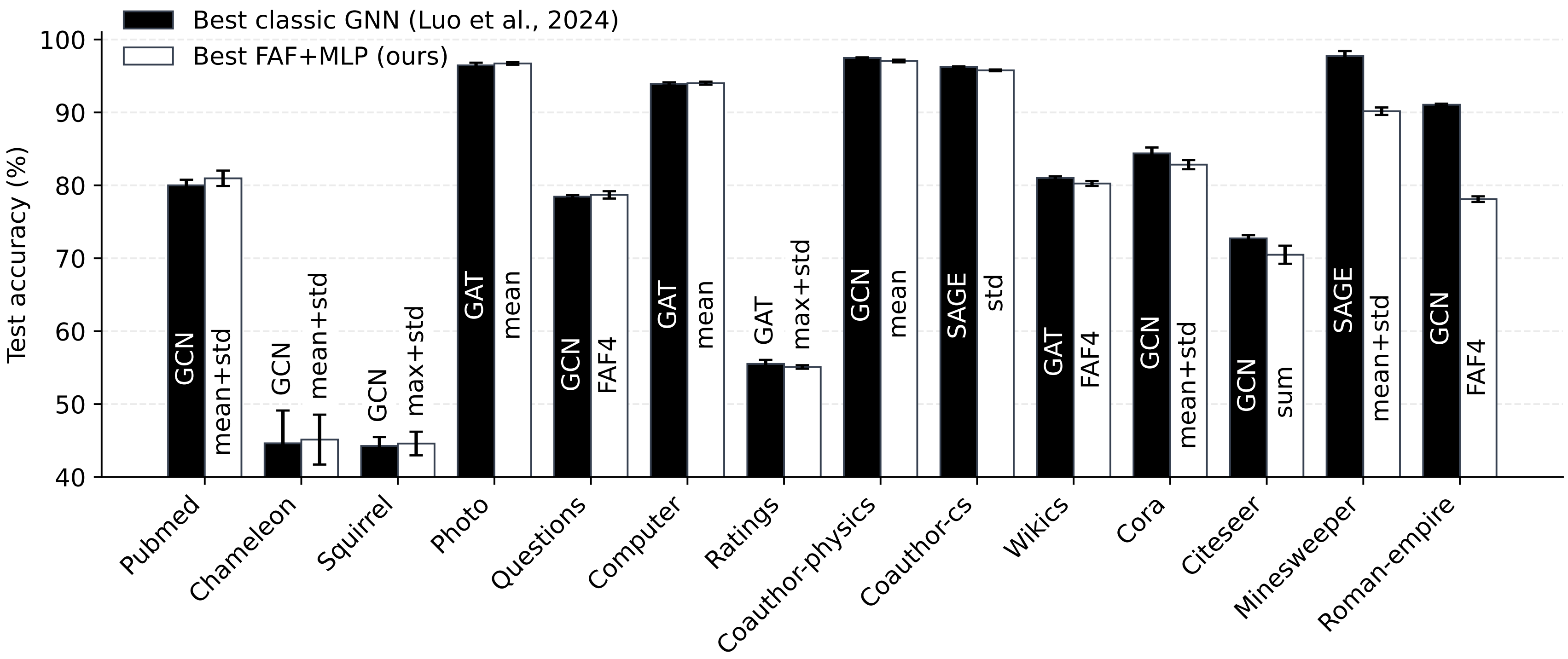
Fixed Aggregation Features (FAF)

- Choice of aggregation(s): $A \in$ [mean, sum, max, std] $\rightarrow$ **concatenate all**
- Choice of tabular model: well-tuned MLP with input size $F \cdot (1 + 4K)$



We do this for 4 different A and concatenate together.

Results on FAF vs. classic GNNs



Observations

1. If GNNs are necessary, they should clearly beat FAFs.
   $\implies$ We are also competitive with complex baselines, including GTs.

Observations

1. If GNNs are necessary, they should clearly beat FAFs.
   $\Longrightarrow$ We are also competitive with complex baselines, including GTs.

2. Two failure points of FAF: Minesweeper and Roman-Empire.
   $\Longrightarrow$ Their GNNs require learnable residual blocks and **long ranges** ($>$10 hops).

Observations

1. If GNNs are necessary, they should clearly beat FAFs.
   $\Longrightarrow$ We are also competitive with complex baselines, including GTs.

2. Two failure points of FAF: Minesweeper and Roman-Empire.
   $\Longrightarrow$ Their GNNs require learnable residual blocks and **long ranges** ($>$10 hops).

3. Extra benefits of using tabular models.
   $\Longrightarrow$ Interpretability, efficiency, augmenting capabilities, and TFMs.

Conclusions from FAFs

Most graph-learning **benchmarks** may not require learned message passing.

FAFs can turn graph ML into **tabular** ML—they should be used as **baselines**.

Sometimes a theoretical explanation does not lead to a performance boost.

Thank you!

Celia Rubio-Madrigal

Rebekka Burkholz

Fixed Aggregation Features Can Rival GNNs, ICML 2026