Fixed Aggregation Features Can Rival GNNs

Celia Rubio-Madrigal Rebekka Burkholz

CISPA Helmholtz Center for Information Security

Background and motivation

- GNNs learn neighborhood aggregations (message passing).
- Luo et al. (2024): Classic GNNs can rival Graph Transformers and heterophily-aware models if well-tuned.

Reducing complexity in graph learning is possible! But what's left?
$\Longrightarrow$ How relevant is **learning the aggregation** itself?

Idea: Replace trainable with fixed, concatenated aggregations, transforming graph data into tabular data to be learned by MLPs.
$\Longrightarrow$ Can it be maximally expressive? Can it be performant?

Method: Fixed Aggregation Features (FAFs)

FAFs are a preprocessing step (Fig. 1). For depth $k \in \{1, \dots, K\}$ and reducer $r \in \mathcal{R} \subset \{\Phi, \text{mean}, \text{sum}, \text{max}, \text{min}, \text{std}, \dots\}$:

$$h_v^{(0,r)} = x_v, \qquad h_v^{(k,r)} = r\Big(\{\, h_u^{(k-1,r)} : u \in N(v) \,\}\Big). \tag{1}$$

We then train an MLP on the concatenated representation:

$$z_v = x_v \oplus \left(\bigoplus_{r \in \mathcal{R}} \bigoplus_{k \in \{1,\dots,K\}} h_v^{(k,r)} \right) \tag{2}$$

If r is **injective**, the neighborhood information is preserved.

Theory: Existence of lossless aggregation

Kolmogorov–Arnold representation (KA): There exists a fixed, injective Φ (and ϕ) such that any multivariate f can be written as

$$f\left(\{\, h_u : u \in N(v) \,\}\right) = g(\Phi(\{\, h_u : u \in N(v) \,\})) = g(\Sigma_{u \in N(v)} \phi(h_u))$$

for a univariate g (modeled by an MLP). While Φ is discontinuous, the learnable part g inherits continuity from f.

Benefits of the tabular form

We turn the graph into a tabular matrix, unlocking many ML tools:
- Interpretability: feature importance per hop k and reducer r.
- Efficiency: no backpropagation through convolutions.
- Augmentations: e.g., structural features or rewiring schemes.

In theory, we can always transform *graph* data into *tabular* data without losing information

- Despite being injective and thus information preserving, the KA transformation Φ **performs worse** than mean/sum/max/etc.
- There is potential for designing fixed aggregations that preserve more information and **provide good representations** for MLPs.
- Expressivity and information preservation is not everything. We also need **trainability** to **learn** better features.

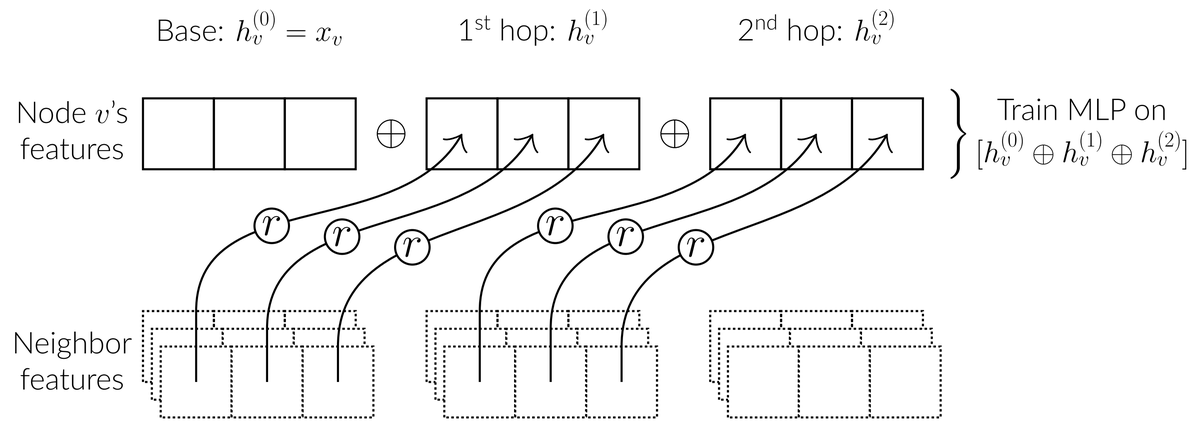
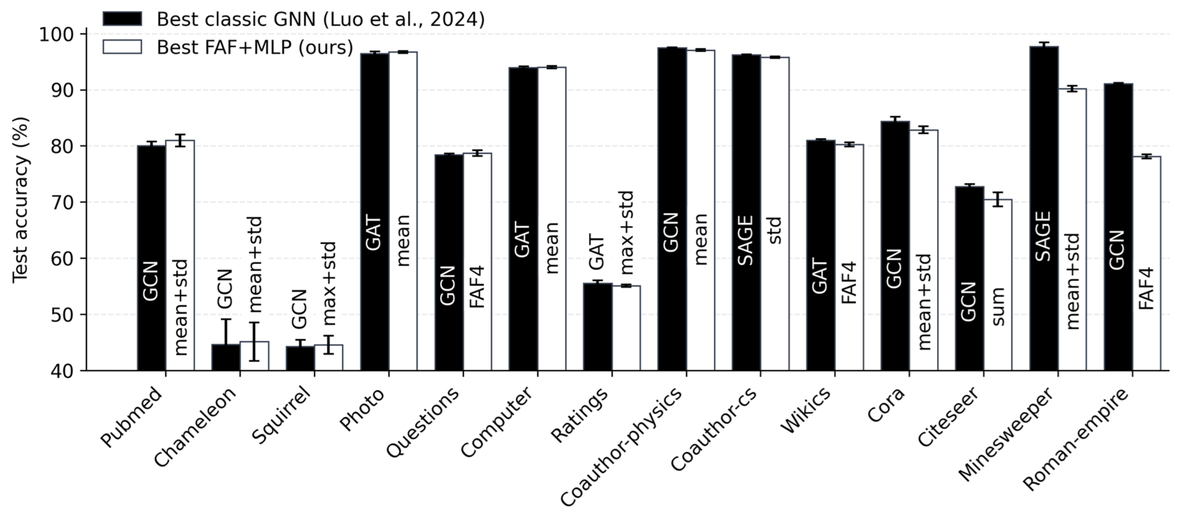


Figure 1. Construction of Fixed Aggregation Features (FAFs) for one reducer r.

Experiments on node classification

On 12 out of 14 node classification benchmarks, we rival well-tuned GCNs, GATs, and GraphSAGEs with FAFs trained on MLPs.



Ablations and derived insights

Hops: Accuracy peaks at $K = 2$-4, and 2 is often enough.
$\Longrightarrow$ Current benchmarks **don't require depth**.
$\Longrightarrow$ **Higher depths overfit**. (Is this also the case for GNNs?)

(Non)-Injectivity: Φ behaves badly. Mean/max/etc. are not lossless, but yield easier features to optimize and are enough here.
$\Longrightarrow$ Benchmarks **don't require injectivity**, but **distributional** info.

Reducers: Mean alone is often the best, and std complements it. Max/sum are more helpful on specific datasets.
$\Longrightarrow$ Concatenating **all reducers** is a robust default (FAF4).

Classifier: A well-tuned MLP outperforms a single linear layer. Only using the last hop is worse than concatenating **all hops**.
$\Longrightarrow$ Different insights from, e.g., freezing weights or reservoirs.

Many benchmarks do not empirically need either *learned* or *injective* aggregation

- How much is there to gain from learning neighborhoods?
- Either the signals live in **0–2 hops**, or GNNs overfit.
- We need **richer benchmarks** with long-range dependencies.
- FAFs + well-tuned MLPs should always be added as **baselines**.