# Are Graph Transformers Necessary? Efficient Long-Range Message Passing with Fractal Nodes in MPNNs

Jeongwhan Choi[1], Seungjun Park[1], Sumin Park[1], Sung-Bae Cho[2], Noseong Park[1]

[1]Korea Advanced Institute of Science & Technology [2]Yonsei University

## **Fractal Nodes**: Long-Range Message Passing **Without** Graph Transformers!

Mitigating **over-squashing** and enabling long-range interactions at MPNN **efficiency**.

## Motivation: Limitations of MPNNs and Graph Transformers

Table: Comparison of GNN Approaches

| Method | Long-Range | Efficiency | Locality |
|---|---|---|---|
| MPNN | ✗ | $\mathcal{O}(\lvert\mathcal{E}\rvert)$ | ✓ |
| Graph Transformer | ✓ | $\mathcal{O}(\lvert\mathcal{V}\rvert^2)$ | ✗ |
| **MPNN + Fractal Nodes** | ✓ | $\mathcal{O}(\lvert\mathcal{E}\rvert)$ | ✓ |

**Key Question:** *Can we enhance long-range message passing while preserving MPNN efficiency?*

**Our Answer:** Yes! Graph partitioning induces **fractal structure** that enables long-range interactions.

## Main Idea: Fractal Nodes Architecture

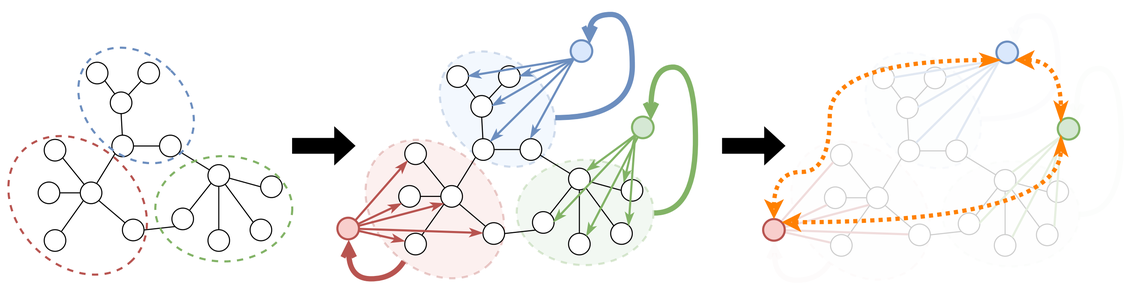


Figure: *Fractal nodes* (●, ●, ●) coexist with original nodes, aggregate subgraph-level features, and enable long-range interactions via MLP-Mixer (orange dashed lines).

## Method: Message Passing with Fractal Nodes

**Three-step update process:**

$$\widetilde{h}_{v,c}^{(\ell+1)} = \varphi^{(\ell)}(h_{v,c}^{(\ell)}, \psi^{(\ell)}(\{h_{u,c}^{(\ell)} : u \in \mathcal{N}_v\})) \quad \text{(Node update)} \quad (1)$$

$$f_c^{(\ell+1)} = \underbrace{\text{LPF}(\{h_{v,c}\})}_{\text{Sub-global}} + \omega_c^{(\ell)} \cdot \underbrace{\text{HPF}(\{h_{v,c}\})}_{\text{Local}} \quad \text{(Fractal node)} \quad (2)$$

$$h_{v,c}^{(\ell+1)} = \widetilde{\varphi}^{(\ell)}(\widetilde{h}_{v,c}^{(\ell+1)}, f_c^{(\ell+1)}) \quad \text{(Final update)} \quad (3)$$

**Key Insight:** Mean pooling captures only the **DC component** (0-th Fourier basis, i.e., constant/average). Our fractal nodes capture richer **subgraph-level** frequency information:

- LPF $= \frac{1}{|\mathcal{V}_c|}\sum_{v\in\mathcal{V}_c} h_v$: DC component (sub-global mean)
- HPF $= h_v -$ LPF: Higher-freq components (deviation from mean)
- $\omega_c^{(\ell)}$: Learnable weight balancing low + high freq

$\Rightarrow$ FN: fractal nodes only.

$\Rightarrow$ FN$_M$: + MLP-Mixer for inter-subgraph interaction.

## Theoretical Analysis: Over-squashing Mitigation

**Over-squashing**: information from distant nodes compressed into fixed-size vectors. Fractal nodes mitigate this by reducing **effective resistance**.

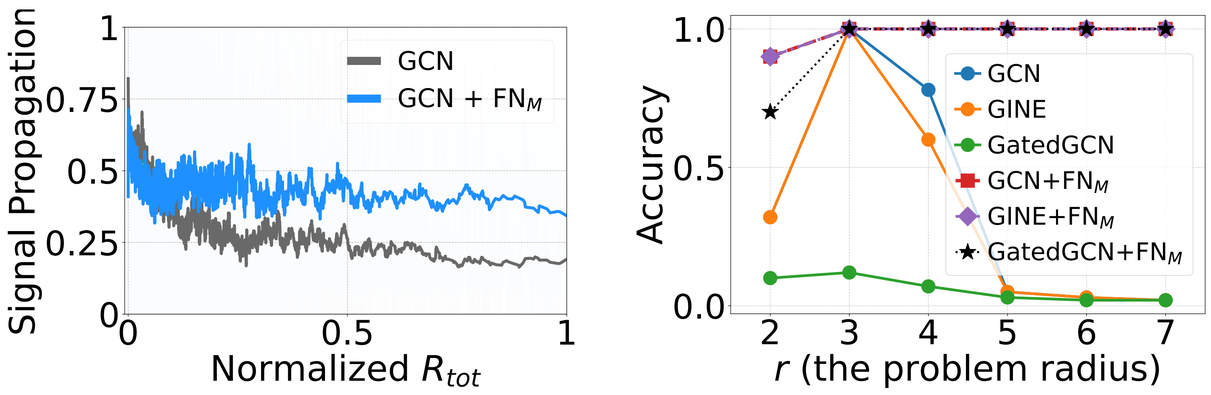
**Theorem 1:** For augmented graph $\mathcal{G}_f$ with fractal nodes: $R_f(u,v) \leq R(u,v)$

$\rightarrow$ Adding fractal nodes creates shortcuts that **lower effective resistance** between nodes.

**Theorem 2:** After $\ell$ layers: $\|h_u^{(\ell)} - h_v^{(\ell)}\| \leq \exp(-\ell/R_f(u,v))\|h_u^{(0)} - h_v^{(0)}\|$

$\rightarrow$ Lower resistance enables **faster signal propagation** between distant node pairs.

### Empirical Analysis:

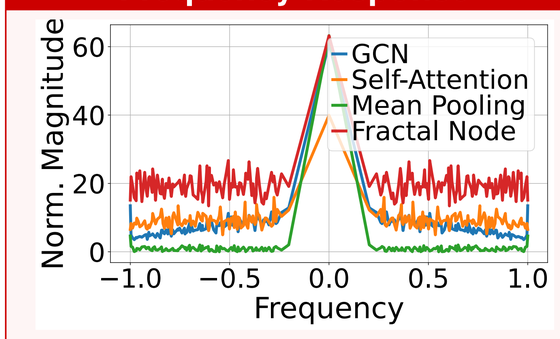



(a) Signal propagation on PEPTIDES-FUNC (b) TreeNeighbourMatch accuracy

Figure: (a) GCN+FN$_M$ maintains higher signal flow. (b) MPNNs fail for $r > 3$.

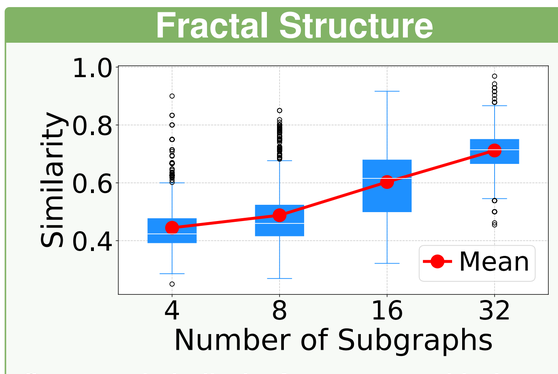
## Properties of Fractal Nodes

### Frequency Response




Fractal nodes preserve **high-frequency** information that mean pooling discards.

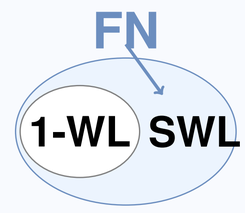
### Fractal Structure




Structural similarity **increases** with the number of subgraphs $C$.

## Expressive Power: Beyond 1-WL Test

| Method | CSL | SR25 | EXP |
|---|---|---|---|
| GCN | 10.00 | 6.67 | 52.17 |
| GINE | 10.00 | 6.67 | 51.35 |
| GatedGCN | 10.00 | 6.67 | 51.25 |
| GCN + FN$_M$ | 39.67 | **100.0** | 86.40 |
| GINE + FN$_M$ | 47.33 | **100.0** | 95.58 |
| GatedGCN + FN$_M$ | **49.67** | **100.0** | **96.50** |




**100%** on SR25! (MPNNs: 6.67%)

$\Rightarrow$ Standard MPNNs **fail** on graphs indistinguishable by 1-3 WL tests!

## Experimental Results: Graph-Level Benchmarks

Table: Performance comparison on 6 benchmark datasets

| Method | PEP-FUNC AP ↑ | PEP-STRUCT MAE ↓ | MNIST Acc ↑ | CIFAR10 Acc ↑ | MOLHIV ROC ↑ | MOLTOX21 ROC ↑ |
|---|---|---|---|---|---|---|
| GCN | 0.6328 | 0.2758 | 0.9269 | 0.5423 | 0.7529 | 0.7525 |
| GINE | 0.6405 | 0.2780 | 0.9705 | 0.6131 | 0.7885 | 0.7730 |
| GatedGCN | 0.6300 | 0.2778 | 0.9776 | 0.6628 | 0.7874 | 0.7641 |
| GraphGPS | 0.6534 | 0.2509 | 0.9805 | 0.7230 | 0.7880 | 0.7570 |
| GRIT | 0.6988 | 0.2460 | 0.9811 | **0.7647** | - | - |
| Graph-ViT | 0.6970 | 0.2449 | 0.9846 | 0.7158 | 0.7997 | 0.7910 |
| Exphormer | 0.6527 | 0.2481 | 0.9841 | 0.7469 | - | - |
| GECO | 0.6975 | 0.2464 | - | - | 0.7980 | - |
| GNN-AK+ | 0.6480 | 0.2736 | - | 0.7219 | 0.7961 | - |
| CRaWl | 0.6963 | 0.2506 | 0.9794 | 0.6901 | 0.7707 | - |
| GCN + FN$_M$ | 0.6787 | 0.2464 | 0.9455 | 0.6413 | 0.7866 | 0.7882 |
| GINE+FN$_M$ | **0.7018** | **0.2446** | 0.9786 | 0.6672 | **0.8127** | **0.7926** |
| GatedGCN+FN$_M$ | 0.6950 | 0.2453 | **0.9848** | 0.7526 | 0.8097 | 0.7922 |

$\Rightarrow$ Fractal nodes outperform Graph Transformers and Subgraph-based methods!

## Large-Scale Node Classification & Efficiency

Table: Results on large-scale graphs (%)

| Method | ogbn-arxiv | ogbn-products |
|---|---|---|
| GraphGPS | 70.97 | OOM |
| Exphormer | 72.44 | OOM |
| GCN | 71.74 | 75.64 |
| GCN + FN | **73.03** | 81.29 |
| GraphSAGE + FN$_M$ | 72.54 | **83.11** |

Table: Runtime & Memory (ogbn-arxiv)

| Method | Time (s) | Mem (GB) |
|---|---|---|
| GCN | 1.27 | 16.49 |
| GraphGPS | 1.32 | 38.91 |
| Exphormer | 0.74 | 34.04 |
| GCN + FN | 1.27 | 16.49 |
| GCN + FN$_M$ | 1.27 | 16.49 |

$\Rightarrow$ Graph Transformers **fail to scale** (OOM), while Fractal Nodes maintain **zero overhead**!

## Comparison with Graph Rewiring Methods & Virtual Nodes

| Method | PEP-FUNC AP ↑ | PEP-STRUCT MAE ↓ |
|---|---|---|
| GCN | 0.5930$\pm$0.0023 | 0.3496$\pm$0.0013 |
| + FoSR | 0.5947$\pm$0.0035 | 0.3473$\pm$0.0007 |
| + SDRF | 0.5947$\pm$0.0126 | 0.3478$\pm$0.0013 |
| + BORF | 0.5994$\pm$0.0037 | 0.3514$\pm$0.0009 |
| + PANDA | 0.6028$\pm$0.0031 | 0.3272$\pm$0.0001 |
| + LASER | 0.6440$\pm$0.0010 | 0.3043$\pm$0.0019 |
| + FN | **0.6445**$\pm$0.0057 | **0.2535**$\pm$0.0012 |

Table: GCN + Rewiring methods

| Method | PEP-FUNC AP ↑ | PEP-STRUCT MAE ↓ |
|---|---|---|
| GCN + VN | 0.6455$\pm$0.0060 | 0.2745$\pm$0.0015 |
| GINE + VN | 0.6575$\pm$0.0080 | 0.2683$\pm$0.0020 |
| GatedGCN + VN | 0.6823$\pm$0.0075 | 0.2475$\pm$0.0012 |
| GatedGCN + VN$_G$ | 0.6822$\pm$0.0077 | 0.2458$\pm$0.0017 |
| GCN + FN$_M$ | 0.6787$\pm$0.0048 | 0.2464$\pm$0.0014 |
| GINE + FN$_M$ | **0.7018**$\pm$0.0074 | **0.2446**$\pm$0.0018 |

Table: Virtual Node methods

$\Rightarrow$ Fractal nodes outperform both graph rewiring and virtual node approaches!