EXPHORMER
SPARSE TRANSFORMERS FOR GRPAHS

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam,
Danica J. Sutherland, Ali Kemal Sinop

Seminar in Deep Neural Networks
Presenter: Johannes Herter
May 21, 2023
GRAPH LEARNING: GRAPHS ARE WIDELY APPLICABLE ACROSS DOMAINS

Social networks

Chemoinformatics

Product recommendations

Traffic prediction

Weather forecasting

Protein-protein associations
CURRENT PARADIGM USES MESSAGE PASSING
CURRENT PARADIGM USES MESSAGE PASSING
CURRENT PARADIGM USES MESSAGE PASSING
CURRENT PARADIGM USES MESSAGE PASSING
MPNNS INCORPORATE A LOCAL VIEW OF THE GRAPH, WHICH CAN CAUSE ISSUES

a node cannot see some nodes
(Barceló et al., 2020)
GNNS INCORPORATE A LOCAL VIEW OF THE GRAPH, WHICH CAN CAUSE ISSUES

Under-reaching

Over-smoothing

all nodes start to look the same
(Oono and Suzuki, 2020)
GNNS INCORPORATE A LOCAL VIEW OF THE GRAPH, WHICH CAN CAUSE ISSUES

Under-reaching

Over-smoothing

Over-squashing

information gets “lost” in transit
(Topping et al., 2022)
GNNS INCORPORATE A **LOCAL VIEW OF THE GRAPH, WHICH CAN CAUSE ISSUES**

- **Under-reaching**
- **Over-smoothing**
- **Over-squashing**
- **Limited Expressivity**

At most as expressive as 1-WL
(Morris et al., 2019)
GNNS INCORPORATE A LOCAL VIEW OF THE GRAPH, WHICH CAN CAUSE ISSUES

Under-reaching

Over-smoothing

Over-squashing

Limited Expressivity

at most as expressive as 1-WL (Morris et al., 2019)
GNNS INCORPORATE A **LOCAL** VIEW OF THE GRAPH, WHICH CAN CAUSE ISSUES

- **Under-reaching**
- **Over-smoothing**
- **Over-squashing**
- **Limited Expressivity**

at most as expressive as 1-WL
(Morris et al., 2019)
COULD WE OVERCOME THESE ISSUES BY INCORPORATING GLOBAL INFORMATION?
INTEGRATING GLOBAL INFORMATION

Ways of incorporating global information into graph learning?

- Virtual nodes
- Unique node identifiers
- Adding global information e.g. as node features
- Transformers
Why use transformers?

- Transformers have had success across a variety of domains
  - Natural language processing (OpenAI, 2023)
  - Computer vision (Dosovitskiy et al., 2020)
  - Speech (Gulati et al., 2020)
  - Biological sequence modelling (Rives et al., 2021)
INTEGRATING GLOBAL INFORMATION VIA TRANSFORMERS

Why use transformers?

- Transformers have had success across a variety of domains
- A single self-attention layer addresses the issues of message passing GNNs
- End-to-end trainable
- Doesn’t require hand-crafted features
TYPES OF TRANSFORMER ARCHITECTURES

Encoder-Decoder

Encoder-only

Decoder-only
TYPES OF TRANSFORMER ARCHITECTURES

Encoder-Decoder

Encoder-only

Decoder-only
HOW CAN WE ADAPT THE TRANSFORMER TO WORK WELL ON GRAPH DATA?
HOW CAN WE ADAPT THE TRANSFORMER TO WORK WELL ON GRAPH DATA?

- **NLP**
  - Input tokenization

- **Graph Learning**
  - Input treats the graph as bag of nodes
HOW CAN WE ADAPT THE TRANSFORMER TO WORK WELL ON GRAPH DATA?

- Input tokenization

- Input treats the graph as bag of nodes
HOW CAN WE ADAPT THE TRANSFORMER TO WORK WELL ON GRAPH DATA?

- **Input tokenization**
  - NLP
  - Graph Learning
    - Input treats the graph as bag of nodes
HOW CAN WE ADAPT THE TRANSFORMER TO WORK WELL ON GRAPH DATA?

NLP

- Input tokenization

Graph Learning

- Input treats the graph as bag of nodes
HOW CAN WE ADAPT THE TRANSFORMER TO WORK WELL ON GRAPH DATA?

- **Input tokenization**
  - Input treats the graph as bag of nodes

- **Positional encoding**
  - Captures a word's position in a text
  - Is added/concatenated to the input
HOW CAN WE ADAPT THE TRANSFORMER TO WORK WELL ON GRAPH DATA?

NLP
- Input tokenization
- Positional encoding

Graph Learning
- Input treats the graph as bag of nodes
- Encode structure via positional encoding
HOW CAN WE ADAPT THE TRANSFORMER TO WORK WELL ON GRAPH DATA?

NLP
- Input tokenization
- Positional encoding
- Transformer layer with self-attention

Graph Learning
- Input treats the graph as bag of nodes
- Encode structure via positional encoding
HOW CAN WE ADAPT THE TRANSFORMER TO WORK WELL ON GRAPH DATA?

- Input tokenization
- Positional encoding
- Transformer layer with self-attention

Graph Learning
- Input treats the graph as a bag of nodes
- Encode structure via positional encoding
- Transformer layer with self-attention
ABSOLUTE POSITIONAL ENCODING (APE)

PE ON GRAPHS?
PE ON GRAPHS: LAPLACIAN POSITIONAL ENCODING

- Graph Laplacian: How does a node relate to its neighbours
- $U$ is sorted by increasing order of the corresponding eigenvalue.
- Take the first $1, \ldots, k$ eigenvectors of $\Delta$

\[
\Delta = I_n - D^{-\frac{1}{2}}AD^\frac{1}{2}
\]

\[
PE_{LPE} = [U_1, \ldots, U_k]
\]
PE ON GRAPHS: LAPLACIAN POSITIONAL ENCODING

- Graph Laplacian: How does a node relate to its neighbours

DEMO
ABSOLUTE POSITIONAL ENCODING VS LAPLACIAN POSITIONAL ENCODING

APE of a sequence

LPE of a graph
HOW CAN WE ADAPT THE TRANSFORMER TO WORK WELL ON GRAPH DATA?

- Input treats the graph as a bag of nodes
- Encode structure via positional encoding
- Transformer layer with self-attention

Graph Learning
LIMITATIONS OF (GRAPH) TRANSFORMERS

- Loss of inductive bias (locality)
- Self-Attention is $\mathcal{O}(N^2)$!
- A lot of research in developing sparse attention
- Tradeoff between performance and speed

(Lin et al., 2022)
WHAT WE HAVE SEEN SO FAR

- Graph Representation Learning
- Adapting Transformers for Graph Data
- Limitations of Graph Transformers
- Now: Graph-GPS
GRAPH-GPS: GENERAL POWERFUL SCALABLE GRAPH TRANSFORMER

Positional & Structural Features

Combine MPNN and Transformer

\[ f(\mathbf{x}, \mathbf{y}) \]

(Rampášek et al., 2023)
Why not using *sparse attention* mechanisms more tailored *for graphs*?
EXPHORMER: SPARSE ATTENTION FOR GRAPHS

Full Attention

- Too memory intensive

Original Graph

(Schirzad et al., 2023)
EXPORMER: SPARSE ATTENTION FOR GRAPHS

- Graphs carry much more topological structure than sequences

Original Graph

Local Neighbourhood Attention
EXPHORMER: SPARSE ATTENTION FOR GRAPHS

- Graphs carry much more topological structure than sequences
- Weak long-range information flow
EXPHORMER: SPARSE ATTENTION FOR GRAPHS

Global Attention

- Diameter 2
- Information bottleneck

Original Graph

Local Neighbourhood Attention
EXPHORMER: SPARSE ATTENTION FOR GRAPHS

Global Attention

Expander Graph Attention

- Approximate complete graphs
  - Spectral properties
  - Mixing properties
  - Diameter is $O(\log N)$

Local Neighbourhood Attention
EXPHORMER: SPARSE ATTENTION FOR GRAPHS – WHY EXPANDER GRAPHS?

- Spectral properties:
  - A $d$-regular expander graph on $n$ vertices approximates the complete graph $K_n$
EXPHORMER: SPARSE ATTENTION FOR GRAPHS – WHY EXPANDER GRAPHS?

- Spectral properties:
  - A $d$-regular expander graph on $n$ vertices approximates the complete graph $K_n$
**EXPHORMER: SPARSE ATTENTION FOR GRAPHS – WHY EXPANDER GRAPHS?**

- Spectral properties:
  - A $d$-regular expander graph on $n$ vertices approximates the complete graph $K_n$
  - There exist simple algorithms to create $d$-regular expander graphs

- Compared to $K_n$ it has only $O(n)$ edges
EXPHORMER: SPARSE ATTENTION FOR GRAPHS – WHY EXPANDER GRAPHS?

- Mixing properties:
  - For a $d$-regular expander graph a random walk mixes well

![Expander Graph Attention Diagram]
EXPHORMER: SPARSE ATTENTION FOR GRAPHS – WHY EXPANDER GRAPHS?

Mixing properties:
- For a $d$-regular expander graph a random walk mixes well
- The diameter is $O_{d,\epsilon}(\log N)$

![Expander Graph Attention](image)
**EXPHORMER: SPARSE ATTENTION FOR GRAPHS**

- **Global Attention**
- **Expander Graph Attention**
- **Local Neighbourhood Attentio**

\[ \Theta(|V| + |E|) \] edges in the attention graph
REMEMBER GRAPH-GPS?

Positional & Structural Features

Combine MPNN and Transformer

\[ f(x, \kappa) \]
EXPHORMER: GRAPH-GPS WITH EXPANDER ATTENTION MODULE

Positional & Structural Features

Combine MPNN and Transformer

\[ f(\cdot, \cdot) \]

- EXPHORMER models have universal approximability!
## Exphormer: Experiments & Results - Baselines

<table>
<thead>
<tr>
<th>Model</th>
<th>CIFAR10 Accuracy↑</th>
<th>MalNet-Tiny Accuracy↑</th>
<th>MNIST Accuracy↑</th>
<th>Cluster Accuracy↑</th>
<th>Pattern Accuracy↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN (Kipf &amp; Welling, 2017)</td>
<td>55.71±0.381</td>
<td>81.0</td>
<td>90.71±0.218</td>
<td>68.50±0.976</td>
<td>71.89±0.334</td>
</tr>
<tr>
<td>GIN (Xu et al., 2018)</td>
<td>55.26±1.527</td>
<td>88.98±0.557</td>
<td>96.49±0.252</td>
<td>64.72±1.553</td>
<td>85.39±0.136</td>
</tr>
<tr>
<td>GAT (Veličković et al., 2018)</td>
<td>64.22±0.455</td>
<td>92.1±0.242</td>
<td>95.54±0.205</td>
<td>70.59±0.447</td>
<td>78.27±0.186</td>
</tr>
<tr>
<td>GatedGCN (Bresson &amp; Laurent, 2017; Divedi et al., 2020)</td>
<td>67.31±0.311</td>
<td>92.23±0.65</td>
<td>97.34±0.143</td>
<td>73.84±0.326</td>
<td>85.57±0.088</td>
</tr>
<tr>
<td>PNA (Corso et al., 2020)</td>
<td>70.35±0.63</td>
<td>–</td>
<td>97.94±0.12</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>DGN (Beaini et al., 2021)</td>
<td>72.84±0.417</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>86.68±0.034</td>
</tr>
<tr>
<td>CRAWI (Toenshoff et al., 2021)</td>
<td>69.01±0.259</td>
<td>–</td>
<td>97.94±0.050</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>GIN-AK+ (Zhao et al., 2022b)</td>
<td>72.19±0.13</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>86.85±0.057</td>
</tr>
<tr>
<td>SAN (Kreuzer et al., 2021)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>76.69±0.65</td>
<td>86.58±0.037</td>
</tr>
<tr>
<td>K-Subgraph SAT (Chen et al., 2022a)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>77.86±0.104</td>
<td>86.85±0.037</td>
</tr>
<tr>
<td>EGT (Hussain et al., 2021)</td>
<td>68.70±0.409</td>
<td>–</td>
<td>98.17±0.087</td>
<td>79.23±0.348</td>
<td>86.82±0.020</td>
</tr>
<tr>
<td>GraphGPS (Rampásek et al., 2022)</td>
<td>72.30±0.356</td>
<td>93.50±0.41</td>
<td>98.05±0.126</td>
<td>78.02±0.180</td>
<td>86.69±0.059</td>
</tr>
<tr>
<td><strong>Exphormer (ours)</strong></td>
<td><strong>74.69±0.125</strong></td>
<td><strong>94.02±0.209</strong></td>
<td><strong>98.55±0.039</strong></td>
<td><strong>78.07±0.037</strong></td>
<td><strong>86.74±0.015</strong></td>
</tr>
</tbody>
</table>
**EXPHORMER: PROMISING RESULTS ON LONG-RANGE GRAPH BENCHMARKS**

<table>
<thead>
<tr>
<th>Model</th>
<th>PascalVOC-SP F1 score ↑</th>
<th>COCO-SP F1 score ↑</th>
<th>Peptides-Func AP ↑</th>
<th>Peptides-Struct MAE ↓</th>
<th>PCQM-Contact MRR ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN</td>
<td>0.1268 ± 0.0060</td>
<td>0.0841 ± 0.0010</td>
<td>0.5930 ± 0.0023</td>
<td>0.3496 ± 0.0013</td>
<td>0.3234 ± 0.0006</td>
</tr>
<tr>
<td>GINE</td>
<td>0.1265 ± 0.0076</td>
<td>0.1339 ± 0.0044</td>
<td>0.5498 ± 0.0079</td>
<td>0.3547 ± 0.0045</td>
<td>0.3180 ± 0.0027</td>
</tr>
<tr>
<td>GatedGCN</td>
<td>0.2873 ± 0.0219</td>
<td><strong>0.2641 ± 0.0045</strong></td>
<td>0.5864 ± 0.0077</td>
<td>0.3420 ± 0.0013</td>
<td>0.3218 ± 0.0011</td>
</tr>
<tr>
<td>GatedGCN+RWSE</td>
<td>0.2860 ± 0.0085</td>
<td>0.2574 ± 0.0034</td>
<td>0.6069 ± 0.0035</td>
<td>0.3357 ± 0.0006</td>
<td>0.3242 ± 0.0008</td>
</tr>
<tr>
<td>Transformer+LapPE</td>
<td>0.2694 ± 0.0098</td>
<td>0.2618 ± 0.0031</td>
<td>0.6326 ± 0.0126</td>
<td><strong>0.2529 ± 0.0016</strong></td>
<td>0.3174 ± 0.0020</td>
</tr>
<tr>
<td>SAN+LapPE</td>
<td><strong>0.3230 ± 0.0039</strong></td>
<td>0.2592 ± 0.0158*</td>
<td>0.6384 ± 0.0121</td>
<td>0.2638 ± 0.0043</td>
<td><strong>0.3350 ± 0.0003</strong></td>
</tr>
<tr>
<td>SAN+RWSE</td>
<td>0.3216 ± 0.0027</td>
<td>0.2434 ± 0.0156*</td>
<td><strong>0.6439 ± 0.0075</strong></td>
<td>0.2545 ± 0.0012</td>
<td><strong>0.3341 ± 0.0006</strong></td>
</tr>
<tr>
<td>GraphGPS</td>
<td><strong>0.3748 ± 0.0109</strong></td>
<td><strong>0.3412 ± 0.0044</strong></td>
<td><strong>0.6535 ± 0.0041</strong></td>
<td><strong>0.2500 ± 0.0005</strong></td>
<td>0.3337 ± 0.0006</td>
</tr>
<tr>
<td>Exphormer (ours)</td>
<td><strong>0.3975 ± 0.0037</strong></td>
<td><strong>0.3455 ± 0.0009</strong></td>
<td><strong>0.6527 ± 0.0043</strong></td>
<td><strong>0.2481 ± 0.0007</strong></td>
<td><strong>0.3637 ± 0.0020</strong></td>
</tr>
</tbody>
</table>
EXPHORMER: NO ONE-FITS-ALL SOLUTION!

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No Local Edges</th>
<th>No Expander Edges</th>
<th>No Global Nodes</th>
<th>All Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cifar10</td>
<td>74.62 ± 0.12</td>
<td>74.53 ± 0.19</td>
<td>74.68 ± 0.19</td>
<td>74.69 ± 0.13</td>
</tr>
<tr>
<td>Malnet-Tiny</td>
<td>92.64 ± 0.55</td>
<td><strong>94.02 ± 0.21</strong></td>
<td>92.48 ± 0.33</td>
<td>92.06 ± 0.18</td>
</tr>
<tr>
<td>Pattern</td>
<td>86.59 ± 0.03</td>
<td>86.70 ± 0.02</td>
<td>86.14 ± 0.08</td>
<td><strong>86.74 ± 0.02</strong></td>
</tr>
<tr>
<td>PascalVOC-SP</td>
<td>0.3708 ± 0.0039</td>
<td>0.3588 ± 0.0013</td>
<td><strong>0.3975 ± 0.0037</strong></td>
<td>0.3682 ± 0.0042</td>
</tr>
<tr>
<td>Peptides-Struct</td>
<td>0.2631 ± 0.0007</td>
<td><strong>0.2481 ± 0.0007</strong></td>
<td>0.2655 ± 0.0003</td>
<td>0.2643 ± 0.0008</td>
</tr>
<tr>
<td>Computer</td>
<td>90.34 ± 0.45</td>
<td>91.48 ± 0.41</td>
<td><strong>91.59 ± 0.31</strong></td>
<td>91.43 ± 0.53</td>
</tr>
</tbody>
</table>
EXPHORMER: SOME IMPROVEMENTS BUT NO ONE-FITS-ALL SOLUTION!

Under-reaching

- a node cannot see some nodes
  (Barceló et al., 2020)

Over-smoothing

- all nodes start to look the same
  (Oono and Suzuki, 2020)

Over-squashing

- information gets “lost” in transit
  (Topping et al., 2022)

Limited Expressivity

- at most as expressive as 1-WL
  (Morris et al., 2019)
EXPHORMER: SOME IMPROVEMENTS BUT NO ONE-FITS-ALL SOLUTION!

- Idea to use expanders as attention graphs is intuitive
- Exphormer promotes its ideas well, but in the end is just a small extension to Graph-GPS
- Need for careful tuning of the components!
- Benchmarking is hard to compare
THANK YOU FOR YOUR ATTENTION!

Johannes Herter

johannes.herter@inf.ethz.ch
BACKUP: EXPANDER GRAPH CONSTRUCTION IS SIMPLE AND FAST!

Generating a Random Regular Expander  We now describe how we generate a random regular expander. Let $G = (V, E)$ be the original graph, where $V = \{1, 2, \ldots, n\}$. For the purposes of experimentation (in Tables 1 to 5), we use the random graph process analyzed in Friedman (2003) (see Theorem C.2 in Appendix C) to generate a random $d$-regular graph $G' = (V, E')$ on the same node set $V$:

- Pick $d/2$ permutations $\pi_1, \pi_2, \ldots, \pi_{d/2}$ on $V$, each $\pi_i$ chosen independently and uniformly among all possible permutations of $n$ elements.
- Then, letting $[k]$ denote $\{1, 2, \ldots, k\}$, choose

  $$E' = \{(i, \pi_j(i)), (i, \pi_j^{-1}(i)) : j \in [d/2], i \in [n]\}.$$

**Theorem C.2.** (Friedman, 2003, Theorem 1.1) Fix $\epsilon > 0$ and an even integer $d > 2$. Then, suppose $G = (V, E)$ is a random graph generated by taking $d/2$ independent uniformly random permutations $\pi_1, \pi_2, \ldots, \pi_{d/2}$ on $V = \{1, 2, \ldots, n\}$ and then choosing the edge set as

  $$E = \{(i, \pi_j(i)), (i, \pi_j^{-1}(i)) : 1 \leq j \leq d, 1 \leq i \leq n\}.$$

Then, with probability $1 - O(n^{-\Omega(\sqrt{d})})$, $G$ satisfies $\lambda_j(G) \leq 2\sqrt{d} - 1 + \epsilon$ for $j = 2, \ldots, n$, where $d = \lambda_1(G) \geq \lambda_2(G) \geq \cdots \geq \lambda_n(G) \geq -d$ are the eigenvalues of the adjacency matrix of $G$. 
Theorem 4.1. (Spielman, 2019, Section 27.2) A $d$-regular $\epsilon$-expander $G$ on $n$ vertices spectrally approximates the complete graph $K_n$ on $n$ vertices:\footnote{1}$^2$

$$(1 - \epsilon) \frac{1}{n} L_K \preceq \frac{1}{d} L_G \preceq (1 + \epsilon) \frac{1}{n} L_K.$$ 

Lemma 4.2. (Hoory et al., 2006, Theorem 3.2) Let $G = (V, E)$ be a $d$-regular $\epsilon$-expander graph on $n = |V|$ nodes. For any initial distribution $\pi^{(0)} : V \to \mathbb{R}^+$ and any $\delta > 0$, $\pi^{(t)}$ satisfies

$$\|\pi^{(t)} - \frac{1}{n}\|_1 \leq \delta$$

as long as $t \geq \frac{1}{2(1-\epsilon)} \log(n/\delta^2)$. 
<table>
<thead>
<tr>
<th>Model/Dataset</th>
<th>Cifar10 Accuracy↑</th>
<th>MalNet-Tiny Accuracy↑</th>
<th>PascalVOC-SP F1 score↑</th>
<th>Peptides-Func AP↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS (MPNN-only)</td>
<td>69.948 ± 0.499</td>
<td>92.23 ± 0.65</td>
<td>0.3016 ± 0.0031</td>
<td>0.6159 ± 0.0048</td>
</tr>
<tr>
<td>GPS-BigBird</td>
<td>70.480 ± 0.106</td>
<td>92.34 ± 0.34</td>
<td>0.2762 ± 0.0069</td>
<td>0.5854 ± 0.0079</td>
</tr>
<tr>
<td>GPS-Performer</td>
<td>70.670 ± 0.338</td>
<td>92.64 ± 0.78</td>
<td>0.3724 ± 0.0131</td>
<td>0.6475 ± 0.0056</td>
</tr>
<tr>
<td>GPS-Transformer</td>
<td>72.305 ± 0.344</td>
<td>93.50 ± 0.41</td>
<td>0.3736 ± 0.0158</td>
<td>0.6535 ± 0.0041</td>
</tr>
<tr>
<td>EXPHORMER</td>
<td>74.69 ± 0.125</td>
<td>94.02 ± 0.21</td>
<td>0.3975 ± 0.0037</td>
<td>0.6527 ± 0.0043</td>
</tr>
</tbody>
</table>
## BACKUP: NO COMPARISON TO GRIT ??

<table>
<thead>
<tr>
<th>Model</th>
<th>ZINC</th>
<th>MNIST</th>
<th>CIFAR10</th>
<th>PATTERN</th>
<th>CLUSTER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAR↓</td>
<td>Accuracy↑</td>
<td>MAR↓</td>
<td>Accuracy↑</td>
<td>MAR↓</td>
</tr>
<tr>
<td>GCN</td>
<td>0.367 ± 0.011</td>
<td>90.705 ± 0.218</td>
<td>55.710 ± 0.381</td>
<td>71.892 ± 0.334</td>
<td>68.498 ± 0.976</td>
</tr>
<tr>
<td>GIN</td>
<td>0.326 ± 0.051</td>
<td>96.485 ± 0.252</td>
<td>55.255 ± 1.527</td>
<td>85.387 ± 0.136</td>
<td>64.716 ± 1.553</td>
</tr>
<tr>
<td>GAT</td>
<td>0.384 ± 0.007</td>
<td>95.335 ± 0.305</td>
<td>62.295 ± 0.455</td>
<td>78.271 ± 0.186</td>
<td>70.587 ± 0.447</td>
</tr>
<tr>
<td>GatedGCN</td>
<td>0.282 ± 0.015</td>
<td>97.534 ± 0.143</td>
<td>67.312 ± 0.311</td>
<td>85.568 ± 0.888</td>
<td>73.280 ± 0.326</td>
</tr>
<tr>
<td>GatedGCN-LSPN</td>
<td>0.090 ± 0.001</td>
<td>97.940 ± 0.12</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PNA</td>
<td>0.188 ± 0.004</td>
<td>97.940 ± 0.12</td>
<td>70.45 ± 0.63</td>
<td>72.838 ± 0.417</td>
<td>86.680 ± 0.344</td>
</tr>
<tr>
<td>DGN</td>
<td>0.186 ± 0.003</td>
<td>-</td>
<td>72.19 ± 0.13</td>
<td>86.521 ± 0.057</td>
<td></td>
</tr>
<tr>
<td>GSN</td>
<td>0.101 ± 0.010</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>CNN</td>
<td>0.079 ± 0.006</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>GIN-AK+</td>
<td>0.085 ± 0.004</td>
<td>97.944 ± 0.050</td>
<td>69.013 ± 0.259</td>
<td>86.505 ± 0.057</td>
<td></td>
</tr>
<tr>
<td>GRIN</td>
<td>0.080 ± 0.001</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SANE</td>
<td>0.130 ± 0.006</td>
<td>-</td>
<td>-</td>
<td>86.581 ± 0.037</td>
<td>76.691 ± 0.65</td>
</tr>
<tr>
<td>Graphormer</td>
<td>0.122 ± 0.006</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>K-Subgraph SAT</td>
<td>0.094 ± 0.008</td>
<td>-</td>
<td>-</td>
<td>86.848 ± 0.327</td>
<td>77.856 ± 0.104</td>
</tr>
<tr>
<td>EGT</td>
<td>0.108 ± 0.009</td>
<td>98.173 ± 0.087</td>
<td>68.702 ± 0.499</td>
<td>86.821 ± 0.020</td>
<td>79.232 ± 0.348</td>
</tr>
<tr>
<td>Graphormer-URPE</td>
<td>0.086 ± 0.007</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Graphormer-GD</td>
<td>0.081 ± 0.009</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>GPS</td>
<td>0.070 ± 0.004</td>
<td>98.051 ± 0.126</td>
<td>72.298 ± 0.356</td>
<td>86.685 ± 0.059</td>
<td>78.016 ± 0.180</td>
</tr>
<tr>
<td>GRIT (ours)</td>
<td>0.059 ± 0.002</td>
<td>98.108 ± 0.111</td>
<td>70.468 ± 0.881</td>
<td>87.196 ± 0.076</td>
<td>80.020 ± 0.277</td>
</tr>
</tbody>
</table>

### Model Performance Metrics

<table>
<thead>
<tr>
<th>Model</th>
<th>CIFAR10 Accuracy↑</th>
<th>MalNet-Tiny Accuracy↑</th>
<th>MNIST Accuracy↑</th>
<th>CLUSTER Accuracy↑</th>
<th>PATTERN Accuracy↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN (Kipf &amp; Welling, 2017)</td>
<td>55.71 ± 0.381</td>
<td>81.0</td>
<td>90.71 ± 0.218</td>
<td>68.50 ± 0.976</td>
<td>71.89 ± 0.334</td>
</tr>
<tr>
<td>GIN (Xu et al., 2018)</td>
<td>55.26 ± 1.527</td>
<td>88.98 ± 0.557</td>
<td>96.49 ± 0.252</td>
<td>64.72 ± 1.553</td>
<td>85.39 ± 0.136</td>
</tr>
<tr>
<td>GAT (velivckovic et al., 2018)</td>
<td>64.22 ± 0.455</td>
<td>92.1 ± 0.242</td>
<td>95.54 ± 0.205</td>
<td>70.59 ± 0.447</td>
<td>78.27 ± 0.186</td>
</tr>
<tr>
<td>GatedGCN (Bresson &amp; Laurent, 2017; Devi et al., 2020)</td>
<td>67.31 ± 0.311</td>
<td>92.23 ± 0.65</td>
<td>97.34 ± 0.143</td>
<td>73.84 ± 0.326</td>
<td>85.57 ± 0.088</td>
</tr>
<tr>
<td>PNA (Corso et al., 2020)</td>
<td>70.35 ± 0.63</td>
<td>-</td>
<td>97.94 ± 0.12</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DGN (Heiassni et al., 2021)</td>
<td>72.84 ± 0.417</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>86.68 ± 0.034</td>
</tr>
<tr>
<td>CRAW1 (Toenshoff et al., 2021)</td>
<td>69.01 ± 0.259</td>
<td>-</td>
<td>97.94 ± 0.050</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GIN-AK+ (Zhao et al., 2022b)</td>
<td>72.19 ± 0.13</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>86.85 ± 0.057</td>
</tr>
<tr>
<td>SAN (Kreuer et al., 2021)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>76.69 ± 0.65</td>
<td>86.58 ± 0.037</td>
</tr>
<tr>
<td>K-Subgraph SAT (Chen et al., 2022a)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>77.86 ± 0.104</td>
<td>86.85 ± 0.037</td>
</tr>
<tr>
<td>EGT (Hussain et al., 2021)</td>
<td>68.70 ± 0.409</td>
<td>-</td>
<td>98.17 ± 0.087</td>
<td>79.23 ± 0.348</td>
<td>86.82 ± 0.020</td>
</tr>
<tr>
<td>GraphGPS (Rampakos et al., 2022)</td>
<td>72.30 ± 0.356</td>
<td>93.50 ± 0.41</td>
<td>98.05 ± 0.126</td>
<td>78.02 ± 0.180</td>
<td>86.69 ± 0.059</td>
</tr>
<tr>
<td><strong>GRIT (ours)</strong></td>
<td><strong>74.69 ± 0.125</strong></td>
<td><strong>94.02 ± 0.209</strong></td>
<td><strong>98.55 ± 0.039</strong></td>
<td><strong>78.07 ± 0.037</strong></td>
<td><strong>86.74 ± 0.015</strong></td>
</tr>
</tbody>
</table>