Mixture of Weak and Strong Experts on Graphs

Hanqing Zeng\textsuperscript{1,*}, Hanjia Lyu\textsuperscript{2,*}, Diyi Hu\textsuperscript{3}, Yinglong Xia\textsuperscript{1}, and Jiebo Luo\textsuperscript{2}
\textsuperscript{1}Meta AI \hspace{0.5cm} \textsuperscript{2}University of Rochester \hspace{0.5cm} \textsuperscript{3}University of Southern California
\*Equal contribution

Motivation

- Realistic Graphs often display non-uniform patterns such as local homophily or heterophily.
- Most GNNs overlook these variations since they focus on global properties of the graph.
- Node-specific adaptations could boost performance.

Mowst

\[ L_{\text{Mowst}} = \frac{1}{|V|} \sum_{v \in V} (C(p_v) \cdot L(p_v, y_v) + (1 - C(p_v)) \cdot L(p_v', y_v)) \]

Target node: \( v \)

MLP’s prediction: \( p_v \)

GNN’s prediction: \( p'_v \)

How confident is MLP: \( C(p_v) \)

High Dispersion

Low Dispersion

Logit Value

OR

Confidence

Nodes are split based on the confidence of the weak expert

Algorithm 1 Mowst inference

Input: \( \mathcal{G}(V, E, X) \); target node \( v \)

Output: prediction of \( v \)

Run the trained MLP expert on \( v \)

Get prediction \( p_v \) and confidence \( C(p_v) \) \( \in [0, 1] \)

if random number \( q \in [0, 1] \) has \( q < C(p_v) \) then

Predict \( v \) by MLP’s prediction \( p_v \)

else

Run the trained GNN expert on \( v \)

Predict \( v \) by GNN’s prediction \( p'_v \)

end if

Algorithm 2 Mowst training

Input: \( \mathcal{G}(V, E, X) \); training labels \( \{y_v\} \)

Initialize MLP & GNN weights as \( \theta_M \) & \( \theta_G \)

for round \( r = 1 \) until convergence do

Fix GNN weights \( \theta_G \) & \( \theta'_G \)

Update MLP weights to \( \theta_M \) by gradient descent on \( L_{\text{Mowst}} \) until convergence

Fix MLP weights \( \theta_M \) & \( \theta'_M \)

Update GNN weights to \( \theta'_G \) by gradient descent on \( L_{\text{Mowst}} \) until convergence

end for

Mowst*

\[ L_{\text{Mowst}^*} = \frac{1}{|V|} \sum_{v \in V} L(C(p_v) \cdot p_v + (1 - C(p_v)) \cdot p'_v, y_v) \]

- Mowst may be easier to optimize, while Mowst* has a theoretically lower loss.

Main Results

- Mowst(*) outperforms all other baselines under the same number of layers and hidden dimensions.
- The decoupling of the self-features and neighbor structures, along with the denoising effect of the weak expert are generally beneficial.

<table>
<thead>
<tr>
<th></th>
<th>Flickr</th>
<th>ogbn-products</th>
<th>ogbn-arxiv</th>
<th>Penn94</th>
<th>pokec</th>
<th>twitch-gamer</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>46.93\pm0.00</td>
<td>61.06\pm0.08</td>
<td>55.30\pm0.23</td>
<td>73.61\pm0.40</td>
<td>62.37\pm0.02</td>
<td>60.92\pm0.07</td>
</tr>
<tr>
<td>GAT</td>
<td>52.47\pm0.14</td>
<td>81.53\pm0.05</td>
<td>71.77\pm0.18</td>
<td>59.89\pm0.02</td>
<td>64.08\pm0.09</td>
<td>62.76\pm0.02</td>
</tr>
<tr>
<td>GPR-GNN</td>
<td>53.23\pm0.14</td>
<td>72.41\pm0.04</td>
<td>71.10\pm0.22</td>
<td>81.38\pm0.16</td>
<td>76.83\pm0.05</td>
<td>68.93\pm0.29</td>
</tr>
<tr>
<td>AdaGCN</td>
<td>48.96\pm0.06</td>
<td>69.06\pm0.04</td>
<td>58.45\pm0.50</td>
<td>74.42\pm0.58</td>
<td>55.92\pm0.35</td>
<td>61.02\pm0.14</td>
</tr>
<tr>
<td>GCN</td>
<td>53.86\pm0.37</td>
<td>75.64\pm0.21</td>
<td>71.74\pm0.29</td>
<td>82.17\pm0.04</td>
<td>76.01\pm0.49</td>
<td>62.42\pm0.53</td>
</tr>
<tr>
<td>GCN-skip</td>
<td>52.98\pm0.00</td>
<td>-</td>
<td>69.56\pm0.00</td>
<td>76.58\pm0.23</td>
<td>73.46\pm0.04</td>
<td>61.05\pm0.23</td>
</tr>
<tr>
<td>GraphMoE-GCN</td>
<td>53.03\pm0.14</td>
<td>73.90\pm0.00</td>
<td>71.88\pm0.32</td>
<td>81.61\pm0.27</td>
<td>76.99\pm0.10</td>
<td>62.76\pm0.22</td>
</tr>
<tr>
<td>Mowst (*)-GCN</td>
<td>54.62\pm0.23</td>
<td>76.49\pm0.22</td>
<td>72.82\pm0.07</td>
<td>83.19\pm0.49</td>
<td>77.28\pm0.08</td>
<td>63.74\pm0.23</td>
</tr>
</tbody>
</table>

Expressive Power & Computation Complexity

- Mowst and Mowst* are at least as expressive as the MLP or GNN expert alone.
- Mowst-GCN and Mowst*-GCN are more expressive than the GCN expert alone.
- The worst-case cost of Mowst-GNN or Mowst*-GNN is similar to that of a vanilla GNN.

Future Work

- Multi-expert (e.g., Mixture of progressively stronger experts, hierarchical mixture)
- Weak and strong experts in non-graph domains (e.g., NLP, computer vision)

Acknowledgments

We are grateful to Jingyang Lin, Dr. Wei Zhu, and Dr. Wei Xiong for their constructive suggestions. Luo was supported in part by NSF Award #2238208.

Code Availability: https://github.com/facebookresearch/mowst-gnn