Mixture of Weak and Strong Experts on Graphs

Hanqing Zeng[1,*], Hanjia Lyu[2,*], Diyi Hu[3], Yinglong Xia[1], and Jiebo Luo[2]

[1]Meta AI [2]University of Rochester [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.

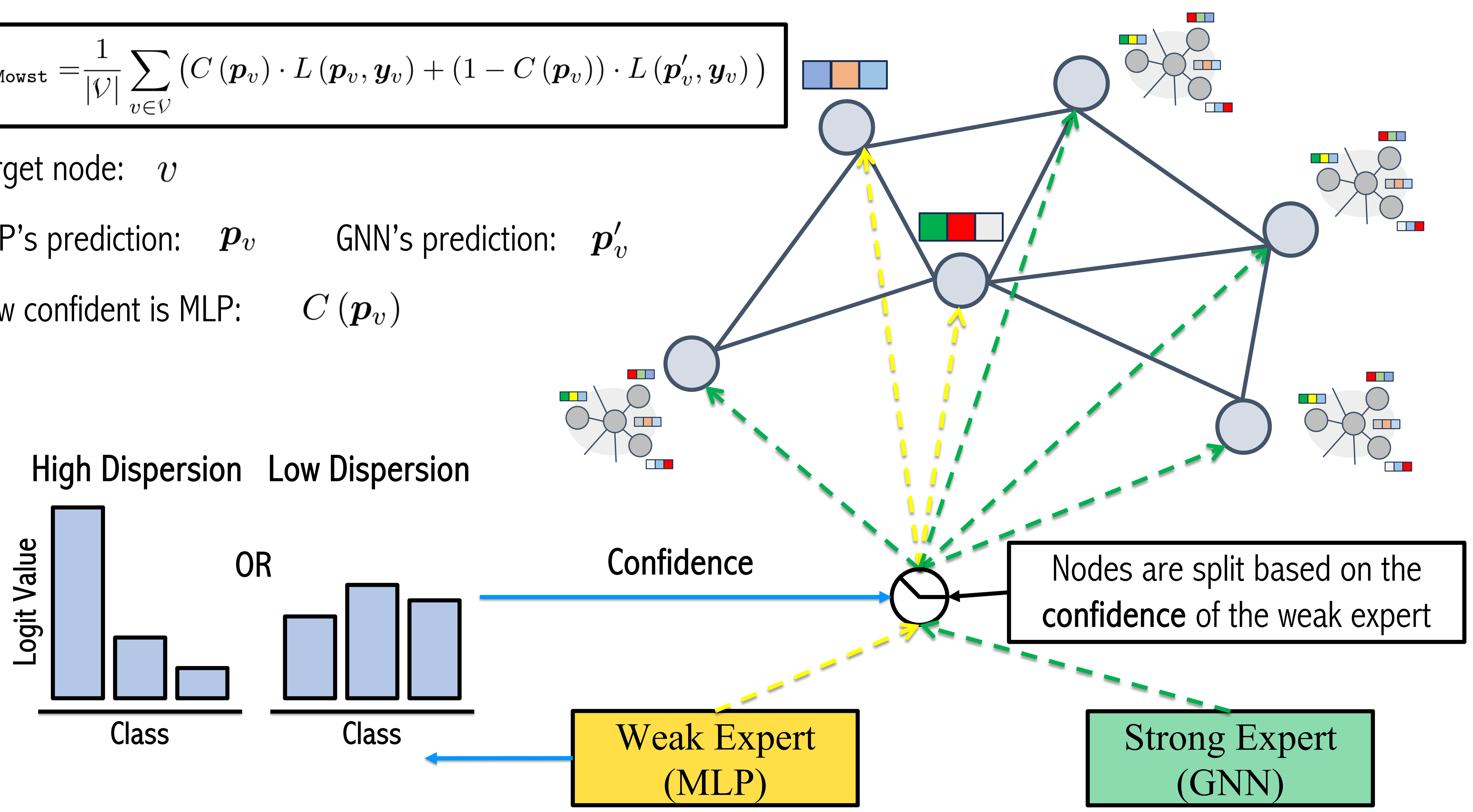
Mowst

$$L_{\texttt{Mowst}} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\left(C\left(\boldsymbol{p}_v\right)\cdot L\left(\boldsymbol{p}_v,\boldsymbol{y}_v\right) + \left(1-C\left(\boldsymbol{p}_v\right)\right)\cdot L\left(\boldsymbol{p}'_v,\boldsymbol{y}_v\right)\right)$$

Target node: v

MLP's prediction: $\boldsymbol{p}_v$ GNN's prediction: $\boldsymbol{p}'_v$

How confident is MLP: $C\left(\boldsymbol{p}_v\right)$



Algorithm 1 `Mowst` inference

Input: $\mathcal{G}\left(\mathcal{V},\mathcal{E},\boldsymbol{X}\right)$; target node v
Output: prediction of v
Run the trained MLP expert on v
Get prediction $\boldsymbol{p}_v$ & confidence $C\left(\boldsymbol{p}_v\right)\in[0,1]$
if random number $q\in[0,1]$ has $q < C\left(\boldsymbol{p}_v\right)$ **then**
  Predict v by MLP's prediction $\boldsymbol{p}_v$
else
  Run the trained GNN expert on v
  Predict v by GNN's prediction $\boldsymbol{p}'_v$
end if

Algorithm 2 `Mowst` training

Input: $\mathcal{G}\left(\mathcal{V},\mathcal{E},\boldsymbol{X}\right)$; training labels $\{\boldsymbol{y}_v\}$
Initialize MLP & GNN weights as $\boldsymbol{\theta}_0$ & $\boldsymbol{\theta}'_0$
for round $r=1$ until convergence **do**
  Fix GNN weights $\boldsymbol{\theta}'_{r-1}$
  Update MLP weights to $\boldsymbol{\theta}_r$ by gradient descent on $L_{\texttt{Mowst}}$
until convergence
  Fix MLP weights $\boldsymbol{\theta}_r$
  Update GNN weights to $\boldsymbol{\theta}'_r$ by gradient descent on $L_{\texttt{Mowst}}$
until convergence
end for

Mowst*

$$L^{\star}_{\texttt{Mowst}} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} L\left(C\left(\boldsymbol{p}_v\right)\cdot\boldsymbol{p}_v + \left(1-C\left(\boldsymbol{p}_v\right)\right)\cdot\boldsymbol{p}'_v,\ \boldsymbol{y}_v\right)$$

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

	Flickr	pokec	twitch-gamer
Mowst-GCN (joint)	53.47 ±0.36	76.62 ±0.11	63.44 ±0.22
Mowst-GCN	**54.62** ±0.23	77.12 ±0.09	**63.74** ±0.23
Mowst*-GCN	53.94 ±0.37	**77.28** ±0.08	63.59 ±0.11

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.

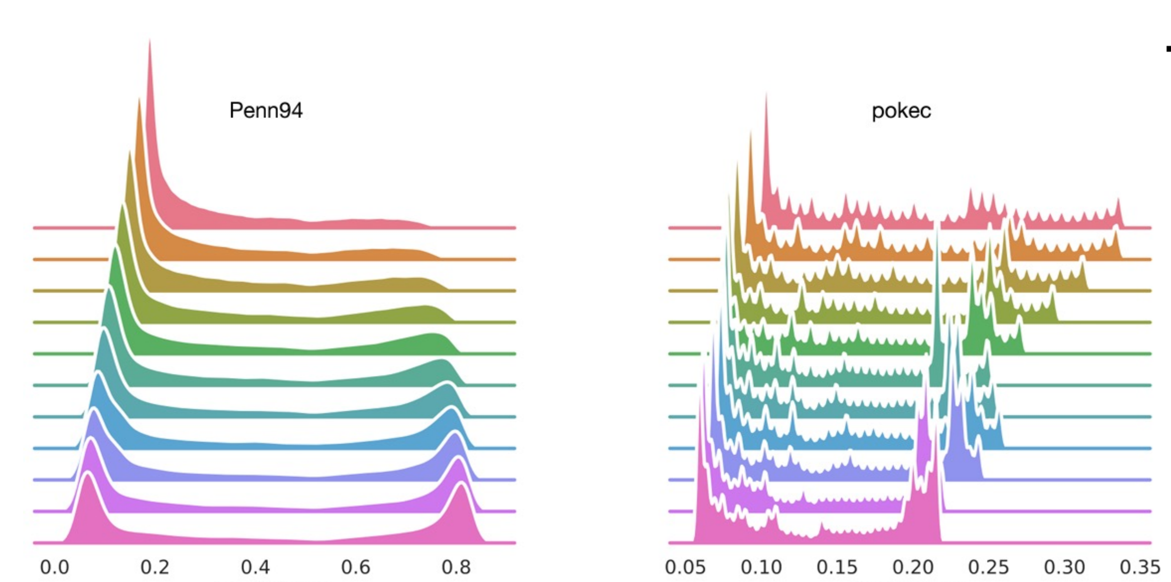
Main Results

- 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.

	Flickr	ogbn-products	ogbn-arxiv	Penn94	pokec	twitch-gamer
MLP	46.93 ±0.00	61.06† ±0.08	55.50† ±0.23	73.61‡ ±0.40	62.37‡ ±0.02	60.92‡ ±0.07
GAT	52.47 ±0.14	OOM	71.58 ±0.17	81.53‡ ±0.55	71.77‡ ±6.18	59.89‡ ±4.12
GPR-GNN	53.23 ±0.14	72.41 ±0.04	71.10 ±0.22	81.38‡ ±0.16	<u>78.83</u>‡ ±0.05	61.89‡ ±0.29
AdaGCN	48.96 ±0.06	69.06 ±0.04	58.45 ±0.50	74.42 ±0.58	55.92 ±0.35	61.02 ±0.14
GCN	53.86 ±0.37	75.64† ±0.21	71.74† ±0.29	82.17 ±0.04	76.01 ±0.49	62.42 ±0.53
GCN-skip	52.98 ±0.00	-	69.56 ±0.00	76.58 ±0.53	73.46 ±0.04	61.05 ±0.23
GraphMoE-GCN	53.03 ±0.14	73.90 ±0.00	71.88†† ±0.32	81.61 ±0.27	76.99 ±0.10	62.76 ±0.22
Mowst(*)-GCN	<u>54.62</u> ±0.23 (**+0.76**)	76.49 ±0.22 (**+0.85**)	**72.52** ±0.07 (**+0.64**)	<u>83.19</u> ±0.43 (**+1.02**)	77.28 ±0.08 (**+0.29**)	63.74 ±0.23 (**+0.83**)
GIN	53.71 ±0.35	-	69.39 ±0.56	82.68 ±0.32	53.37 ±2.15	61.76 ±0.60
Mowst(*)-GIN	**55.48** ±0.32 (**+1.77**)	-	71.43 ±0.26 (**+2.04**)	**84.56** ±0.31 (**+1.88**)	76.11 ±0.39 (**+22.74**)	64.32 ±0.34 (**+2.56**)
GIN-skip	52.70 ±0.00	-	71.28 ±0.00	80.32 ±0.43	76.29 ±0.51	64.27 ±0.25
Mowst(*)-GIN-skip	53.19 ±0.31 (**+0.49**)	-	71.79 ±0.23 (**+0.51**)	81.20 ±0.55 (**+0.88**)	**79.70** ±0.23 (**+3.41**)	**64.91** ±0.22 (**+0.64**)
GraphSAGE	53.51 ±0.05	<u>78.50</u>† ±0.14	71.49† ±0.27	76.75 ±0.52	75.76 ±0.04	61.99 ±0.30
GraphMoE-SAGE	52.16 ±0.13	77.79 ±0.00	71.19 ±0.15	77.04 ±0.55	76.67 ±0.08	63.42 ±0.23
Mowst(*)-SAGE	53.90 ±0.18 (**+0.39**)	**79.38** ±0.44 (**+0.88**)	<u>72.04</u> ±0.24 (**+0.55**)	79.07 ±0.43 (**+2.03**)	77.84 ±0.04 (**+1.33**)	<u>64.38</u> ±0.14 (**+1.05**)

- **Mowst can substantially enhance the performance of state-of-the-art heterophilous GNNs** like H2GCN, with the help of a relatively simple expert such as a standard MLP.

	Penn94	pokec	twitch-gamer
GCN	82.17 ±0.04	76.01 ±0.49	62.42 ±0.53
Mowst(*)-GCN	83.19 ±0.43 (**+1.02**)	77.28 ±0.08 (**+0.29**)	63.74 ±0.23 (**+0.83**)
H_2GCN	82.71 ±0.67	80.89 ±0.16	65.70 ±0.20
Mowst(*)-H_2GCN	**83.39** ±0.43 (**+0.68**)	**83.02** ±0.30 (**+2.13**)	**66.03** ±0.16 (**+0.33**)



- **Specialization via Data Splitting.** Both Mowst and Mowst* adapt their expert collaboration based on the confidence-weighted loss across various graphs.
- See Appendix for the empirical findings on denoised fine-tuning.

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)

Acknowledgements

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