1 VVKE

Thank you for your insightful comment. We believe there may be a misunderstanding regarding the concept of attention and the issue of **unintentional emphasis that occurs in GCN, GraphSAGE, and even in GAT before its attention parameters are learned**. Below, we clarify these points and explain the necessity of addressing the K-skewed-traversal problem.

1. In models like GCN and GraphSAGE, where traditional attention mechanisms are not used, the K-skewed-traversal problem arise as an unintended consequence of the aggregation process. Even in GAT, before the attention parameters are learned, the skewness in emphasis persists. Here attention is applied to skewed representation of nodes. These biases are not dataset-defined rules or optimizations; instead, they are inherent to the computational process of traditional GNNs and vary across nodes and sub-graphs, which we demonstrated in Figures 1, 2. 3 and also stated mathematically in Propositions 1, 2, 3.

2. HGAT first removes the unintended, random, sub-graph dependent emphasis by maintaining an Average Integration Count (AIC) of 1. After that, HGAT applies hop-wise attention on balanced node representation.

3. HGAT is fundamentally different from GAT and GCN. Unlike GAT, which focuses on individual one-hop neighbors, HGAT attends to entire hops (0 to K) and applies attention at the hop level. GCN lacks any attention mechanism, while HGAT dynamically learns hop-level importance. Additionally, HGAT mitigates unintentional aggregation biases (K-skewed-traversal) that occur in GCN, GraphSAGE, and GAT(before its attention mechanism is applied), ensuring uniform contributions from all hops before applying attention.

4. Theorem 4.1 establishes that, prior to applying hop-wise attention, each hop contributes uniformly to the node representation, maintaining an Average Integration Count (AIC) of 1. Attention weights are intentionally ignored at this stage to isolate and demonstrate the uniformity property, which serves as the foundation for applying meaningful hop-wise attention later.

5. Real-world scenarios often require a deep GNN model. For example, in molecular graphs, a chemical property can depend on the interaction of atoms located on opposite sides of the molecule (Ramakrishnan et al., Scientific data 2014; Gilmer et al., ICML 2017), necessitating multi-hop aggregation across the graph. By addressing the K-skewed-traversal problem, HGAT ensures balanced and effective learning in such deep GNN architectures, making it highly applicable to practical, complex tasks.

2 Q5zy

1. Skewness significantly impacts model performance, particularly in deep GNNs. By addressing skewness through uniform integration as a preparation for applying hop-wise attention, HGAT achieves better performance, as demonstrated in Figures 6, 7, and 8, where it outperforms baseline methods with less performance degradation in deeper architectures. While smaller hops may need to contribute more in some datasets, others require distant hops—e.g., a molecule's chemical property might depend on atoms on opposite sides of the molecule (Ramakrishnan et al., Scientific data 2014; Gilmer et al., ICML 2017). HGAT learns the importance of each hop through attention weights, addressing diverse dataset needs. In contrast, traditional models like GCN, and GraphSAGE do not consistently prioritize local nodes but instead exhibit random, sub-graph dependent emphasis as shown in Figures 1, 2, and 3. In the case of GAT, the attention weights are applied to the random, sub-graph-dependent emphasis. This issue is rectified by HGAT leading to superior performance in deep neural network.

2. In HGAT, hop-wise summaries are first computed with uniform contributions from each hop, maintaining an Average Integration Count (AIC) of 1 across all hops. These summaries are then combined using attention weights learned during training. This learning mechanism enables the network to dynamically prioritize specific hop distances based on the dataset's properties. Consequently, the priority of a particular hop (e.g.,$k = 3$ vs. $k = 1$) depends entirely on the dataset and task, and cannot be predetermined.

3. While GCN performs well with fewer layers, there are datasets where distant nodes play a critical role. For example, in molecular graphs, a chemical property can depend on the interaction of atoms located on opposite sides of the molecule (Ramakrishnan et al., Scientific data 2014; Gilmer et al., ICML 2017). In such cases, deeper models are essential for capturing multi-hop dependencies. The weakness of GCN, GraphSAGE, and GAT is their performance degradation in the deep networks as stated in the Introduction section which we illustrated in Figures 6, 7, and 8, while HGAT maintains steady performance with less drop. This robustness highlights HGAT's ability to address the K-skewed-traversal problem, which becomes prominent in deep GNNs.

4. We have considered conducting an ablation study but chose to focus on demonstrating the overall effectiveness of HGAT in addressing the K-skewed-traversal problem and improving performance in deep GNNs. Given this focus, an ablation study was deemed infeasible for the current work.

5. Thank you for the suggestion to include heterophily datasets in our experiments. Due to the page limit for the conference submission, we were

unable to explore them. However, we will consider including heterophily datasets in future works.