Relieving the Over-Aggregating Effect in Graph Transformers

Junshu Sun[1,2] **Wanxing Chang**[3]
Chenxue Yang[4*] **Qingming Huang**[1,2] **Shuhui Wang**[1*]
[1]State Key Lab. of AI Safety, ICT, CAS [2]University of Chinese Academy of Sciences
[3]DAMO Academy, Alibaba Group [4]Agriculture Information Institute, CAAS
{sunjunshu21s,wangshuhui}@ict.ac.cn changwanxing.cwx@alibaba-inc.com
yangchenxue@caas.cn qmhuang@ucas.ac.cn

Abstract

Graph attention has demonstrated superior performance in graph learning tasks. However, learning from global interactions can be challenging due to the large number of nodes. In this paper, we discover a new phenomenon termed over-aggregating. Over-aggregating arises when a large volume of messages is aggregated into a single node with less discrimination, leading to the dilution of the key messages and potential information loss. To address this, we propose Wideformer, a plug-and-play method for graph attention. Wideformer divides the aggregation of all nodes into parallel processes and guides the model to focus on specific subsets of these processes. The division can limit the input volume per aggregation, avoiding message dilution and reducing information loss. The guiding step sorts and weights the aggregation outputs, prioritizing the informative messages. Evaluations show that Wideformer can effectively mitigate over-aggregating. As a result, the backbone methods can focus on the informative messages, achieving superior performance compared to baseline methods.

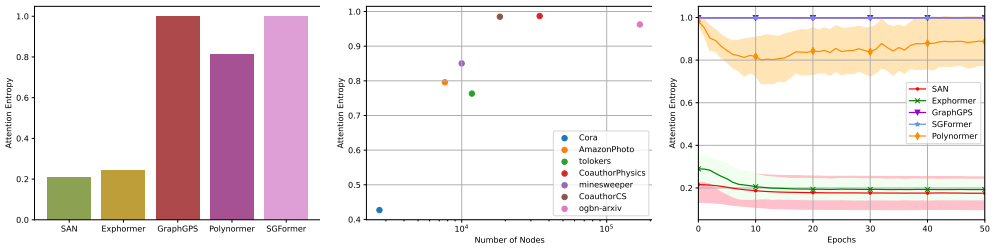
1 Introduction

Transformers have achieved remarkable success in modeling Euclidean-structured data, including natural language processing [47, 26, 25], image understanding [27, 40, 38, 39], and video processing [2, 5, 52]. Built on this success, significant efforts have been devoted to adapting transformers to graph-structured data, giving rise to various graph transformer models [23, 58, 55]. Leveraging the attention mechanism, graph transformers can learn from long-range dependencies and capture global features. These strengths provide effective solutions to address the challenges of over-smoothing [30] and over-squashing [1, 46], commonly encountered in traditional graph neural networks (GNNs) [17, 14, 48, 43, 13, 44, 45]. Evaluations across diverse downstream tasks highlight the promising performance of graph transformers [55, 42, 9].

Nevertheless, the dense global attention in graph transformers incurs a space complexity of $O(n^2)$, where the memory consumption increases quadratically with the number of input nodes. This high complexity hinders graph transformers from scaling to large datasets. To enhance the scalability of graph transformers, two primary approaches have been proposed, including sparse attention [19, 42] and linear global attention [50, 49, 51, 9]. Sparse methods reduce the density of dense global attention, which, in its original form, corresponds to a fully connected graph. The sparsified attention graph contains fewer edges, resulting in reduced memory overhead but also smaller receptive fields for each node. In contrast, linear methods preserve pairwise interactions between nodes and reduce complexity by eliminating the need for explicitly computing the dense attention.

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Average Attention Entropy (b) Entropy across Node Scales (c) Entropy during Training

Figure 1: **Over-Aggregating in Graph Transformers.** The entropy values are normalized to $[0, 1]$, with higher entropy indicating a more uniform distribution of attention scores. In (b), we show the attention entropy with different numbers of nodes for Polynormer [9], a typical linear attention method.

However, learning from pairwise interactions is challenging due to the large number of graph nodes. Fig. 1 illustrates the average attention entropy on different models. Attention entropy denotes the confidence of the nodes in discriminating informative messages during aggregation. Higher entropy indicates a more uniform distribution of attention scores and less confidence. From Fig. 1(a), we can see that linear methods tend to aggregate messages from nodes with higher attention entropy and similar attention scores. When the number of nodes increases in Fig. 1(b), the attention scores become more and more similar. As a result, a large volume of messages is aggregated into a single node with less discrimination, leading to potential information loss. We name this uniform aggregation phenomenon as over-aggregating.

To tackle over-aggregating in graph transformers, we propose Wideformer, a plug-and-play method for graph attention. Wideformer divides the aggregation of all nodes into parallel processes and guides the model to focus on specific subsets of these processes. Specifically, the aggregation inputs are divided into distinct clusters, with each cluster aggregated separately. Instead of compressing messages from all nodes into a single target node, Wideformer limits the input volumes for each aggregation, thereby avoiding message dilution and reducing information loss. To ensure target nodes focus on the informative messages, the aggregation results are sorted and weighed based on the attention scores between the clusters and the target nodes. In contrast to the sparse attention methods that reduce the number of inputs for each target node, Wideformer maintains the global receptive field and distributes the inputs into multiple aggregation processes. By integrating with three popular linear methods GraphGPS [34], SGFormer [51], and Polynormer [9], Wideformer demonstrates effectiveness in tackling over-aggregating. It can consistently benefit the backbones to achieve superior performance over baselines. Codes are available at `https://github.com/sunjss/over-aggregating`.

The contribution of this paper can be summarized as follows:

- We discover a new phenomenon in graph attention, termed over-aggregating, which aggregates a large volume of messages with less discrimination.
- We propose a novel method to relieve over-aggregating, named Wideformer, which divides the aggregation process and guides the model to focus on the informative messages.
- We demonstrate the effectiveness of Wideformer on thirteen real-world datasets, which consistently relieves over-aggregating and benefits backbones to achieve superior performance.

2 Related Work

GNNs perform message passing on input graphs, enabling nodes to learn from their connected neighbors and capture topological features [61]. Among the different message passing methods, GAT [48] first adopts the attention mechanism to aggregate messages. It utilizes attention scores between connected nodes to help central nodes filter out noisy neighbors and focus on the most informative ones. Following GAT, subsequent methods further adopt dense global attention to graphs [23]. A primary challenge for these methods lies in how to effectively encode the graph structure, as global attention performs message passing between all pairs of nodes while neglecting

the underlying graph structure. Several solutions have been proposed, including combining global attention with GNNs to simultaneously encode global features and graph structures [62, 53, 29], learning positional or structural encodings as node features [62, 11, 10, 19], and incorporating graph structures as an attention bias [55].

Another challenge arising from the dense global attention is a space complexity of $O(n^2)$. Efforts to address this challenge can be broadly categorized into two approaches: sparse attention and linear global attention. Sparse attention stems from the Big Bird model [59], which utilizes random masks, sliding windows, and partial global masks to achieve sparsification in traditional transformer architectures. Built upon Big Bird, GraphGPS incorporates these sparse mechanisms into a comprehensive framework for constructing graph transformers [34]. In addition to Big Bird, GAT, which employs input graphs as attention graphs, can also be considered a form of sparse attention. Exphormer [42] and Spexphormer [41] further transform the input graphs into expander graphs, thus better approximating the fully connected graph. However, all of these sparsification methods shrink the receptive fields of each node, leading to a trade-off between model complexity and the ability to capture global information. Different from sparse attentions, Wideformer maintains the global receptive field while requiring linear computing complexity.

In contrast to sparse attention, linear global attention focuses on directly simplifying the computation of dense attention. For example, GraphGPS [34] and Nodeformer [50] extend the linear attention method, Performer [8], from traditional transformer architectures to graph transformers. Performer enables nearly unbiased estimations of dense attention without explicitly computing the attention matrix. In addition to approximation methods, several models have been proposed to model pairwise relations in linear time [51, 9]. However, as shown in Fig. 1(a), the linear methods that maintain the global receptive field during aggregation encounter over-aggregating, leading to dilution of important messages and potential loss of information.

3 Over-Aggregating in Graph Transformers

Graph transformers facilitate global or approximated global message passing between nodes. However, extracting informative features from these global interactions becomes challenging as the number of graph nodes increases. To investigate the behavior of global interactions in graph transformers, we utilize entropy as a measure of attention scores and examine how graph nodes distribute their attention in the receptive field.

3.1 Attention Entropy

In graph attention, input features are projected into query, key, and value features. The value features are aggregated based on the similarity between the query and key features. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ denote the input node features, where n denotes the number of nodes and d denotes the number of features. Node features are mapped to the query, key, and value features through $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$, $\mathbf{K} = \mathbf{X}\mathbf{W}_K$, and $\mathbf{V} = \mathbf{X}\mathbf{W}_V$, respectively. Let $\mathbf{R} = \mathbf{Q}\mathbf{K}^\top$, we now have graph attention as

$$\mathbf{H} = \boldsymbol{\alpha}\mathbf{V}, \quad \boldsymbol{\alpha}_{i,j} = \texttt{softmax}(\mathbf{R})_{i,j} = \frac{\texttt{exp}(\mathbf{R}_{i,j})}{\sum_k \texttt{exp}(\mathbf{R}_{i,k})}. \tag{1}$$

Given the attention scores $\{\boldsymbol{\alpha}_{i,1}, \cdots, \boldsymbol{\alpha}_{i,n}\}$ between the target node v_i and all graph nodes, the attention entropy can be formulated as

$$\mathcal{H}(v_i) = -\sum_{j=1}^{n} \boldsymbol{\alpha}_{i,j} \log \boldsymbol{\alpha}_{i,j}. \tag{2}$$

To ensure comparable results across different datasets, the entropy values are normalized by $\log\ n$ to range $[0, 1]$ for visualizations. Attention entropy measures the distribution of attention scores, indicating the confidence of the target nodes in discriminating informative messages. Lower entropy indicates that each target node focuses on a subset of graph nodes with higher attention scores (confidence), while higher entropy reflects more uniformly distributed attention and lower confidence.

Fig. 1(a) compares the attention entropy in sparse attention and linear global attention methods, including linear methods GraphGPS with Performer [34], Polynormer [9], and SGFormer [51], as

well as sparse methods SAN [19] and Exphormer [42]. The entropy values are averaged over all graph nodes throughout the training process. As shown in Fig. 1(a), linear and sparse methods exhibit distinct attention distribution patterns. Specifically, sparse methods demonstrate low attention entropy, which can be attributed to their sparsification strategies that enable nodes to filter out noisy messages and focus on the most informative ones. Conversely, linear methods involve all the graph nodes for aggregation and tend to assign similar attention scores in the receptive field, resulting in higher entropy. The distinct distribution patterns of the sparse and linear methods indicate the relation between the number of nodes in aggregation and the attention entropy.

Theorem 3.1 (Monotonic Lower Bond of Attention Entropy)**.** *Let $\boldsymbol{\alpha}_{i,\cdot} \in \mathbb{R}^n$ be the attention distribution of node v_i over n nodes. The attention entropy $\mathcal{H}(v_i)$ admits a lower bound that increases monotonically with n.*

The proof is provided in Appendix A. Theorem 3.1 indicates that graph attention will be more likely to have higher attention entropy when aggregating more graph nodes, which is further demonstrated in Fig. 1(b).

3.2 Over-Aggregating

We refer to the aggregation process with high attention entropy as over-aggregating, typically demonstrated as uniform and small attention scores, offering limited discrimination among nodes. As a result, over-aggregating prevents graph nodes from prioritizing the informative messages, leading to message dilution and potential information loss.

To empirically evaluate the effect of over-aggregating with high attention entropy, we add an entropy regularization term to the optimization loss for linear methods GraphGPS with Performer, SGFormer, and Polynormer. The best results among the three backbones are reported. Tab. 1 shows that directly reducing the attention entropy during training gives rise to better model performance. This indicates that over-aggregating with high attention entropy does harm the model performance. However, although directly regularizing attention entropy during training can benefit model performance, this process requires the explicit computation of the whole attention matrix, making it inapplicable to large graphs. As a result, it remains an open problem to alleviate over-aggregating for learning on graphs.

Table 1: **Effect of Over-Aggregating by Attention Entropy Regularization (Measured by accuracy except for ROC-AUC for minesweeper: %).**

	ORI	+REG
CORA	86.03$_{\pm 0.54}$	**86.23**$_{\pm 0.32}$
CITESEER	77.96$_{\pm 0.37}$	**78.19**$_{\pm 0.39}$
AMZPHOTO	95.47$_{\pm 0.53}$	**95.52**$_{\pm 0.44}$
MSWEEPER	97.13$_{\pm 0.17}$	**97.20**$_{\pm 0.32}$

Over-aggregating stems from the optimization of attention scores. As shown in Fig. 1(c), models are typically initialized with high attention entropy, which gradually decreases during training. Throughout this process, models update the parameters $\mathbf{W}_Q$ and $\mathbf{W}_K$ with back-propagation, and thus optimize the attention scores. Take $\mathbf{W}_Q$ as an example, the gradient of $\boldsymbol{\alpha}_{i,j}$ with respect to $\mathbf{W}_Q$ can be derived as

$$\frac{\partial \boldsymbol{\alpha}_{i,j}}{\partial \mathbf{W}_Q}\downarrow = \boldsymbol{\alpha}_{i,j}\downarrow \left(\mathbf{X}_{i,\cdot}^\top \mathbf{X}_{j,\cdot} \mathbf{W}_K^\top - \sum_{k=1}^{n} \boldsymbol{\alpha}_{i,k} \mathbf{X}_{i,\cdot}^\top \mathbf{X}_{k,\cdot} \mathbf{W}_K^\top \right). \tag{3}$$

For more details, please refer to Appendix B. A higher initial attention entropy implies smaller attention scores, which in turn attenuate the signals to update these scores. As a result, models converge with persistently high attention entropy, reflecting suboptimal attention distributions. Eq. 3 can also provide insights for the aggravated over-aggregating problem on large-scale graphs (indicated by Theorem 3.1 and Fig. 1(b))

$$\frac{\partial \boldsymbol{\alpha}_{i,j}}{\partial \mathbf{W}_Q} = \boldsymbol{\alpha}_{i,j}(g_{v_j} - g_{\texttt{others}}\uparrow) = \boldsymbol{\alpha}_{i,j} \left[(1-\boldsymbol{\alpha}_{i,j}) \mathbf{X}_{i,\cdot}^\top \mathbf{X}_{j,\cdot} \mathbf{W}_K^\top - \sum_{\substack{k=1 \\ k\neq j}}^{n\uparrow} \boldsymbol{\alpha}_{i,k} \mathbf{X}_{i,\cdot}^\top \mathbf{X}_{k,\cdot} \mathbf{W}_K^\top \right], \tag{4}$$

where g_{v_j} corresponds to the update signals based on the interaction between node v_i and v_j, and $g_{\texttt{others}}$ corresponds to the cumulative impact of all other nodes. As the number of graph nodes grows, $g_{\texttt{others}}$ increasingly incorporates more irrelevant signals, diluting the relative strength of the meaningful update signals in g_{v_j}. This dilution hinders the optimization of $\boldsymbol{\alpha}_{i,j}$, preventing the model from learning discriminative attention distributions and amplifying the over-aggregating issue on large-scale graphs.

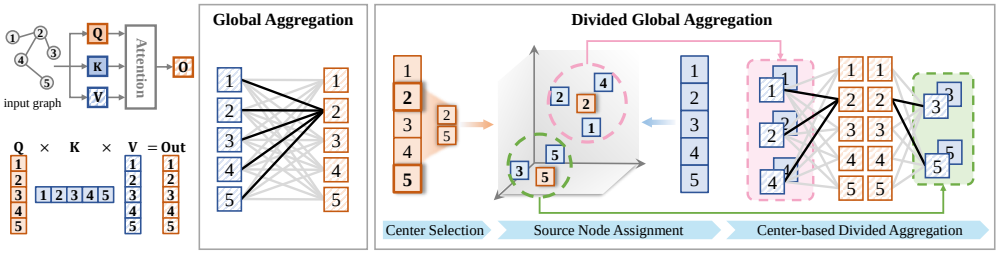


Figure 2: **Attention Comparison between Global Aggregation and Our Divided Global Aggregation.** Global attention employs global aggregation to aggregate source messages into a one-dimensional representation. To relieve the over-aggregating, we assign source nodes into distinct clusters and aggregate each cluster separately, giving rise to multiple aggregation results.

3.3 Discussion

Similar to the occurrence of over-aggregating on the attention graphs, a recent study highlights a phenomenon known as over-globalizing in graph transformers, where models struggle to learn from neighboring nodes in homophilic datasets [54]. While over-globalizing specifically concerns nodes at varying distances, over-aggregating focuses on the discrimination of all nodes during aggregation.

In the context of message passing on input graphs, over-dilution studies aggregation for both feature dimensions and neighboring nodes [21]. Specifically, intra-node dilution focuses on the volumes of the node features, while inter-node dilution focuses on whether a node's own information is preserved during aggregation. Different from over-dilution, over-aggregating focuses on the volumes of the input nodes and emphasizes the preservation of messages from the informative nodes, regardless of whether that includes the node itself.

Besides over-dilution, over-smoothing occurs when node features become increasingly similar as the number of layers grows [30]. This results from the low-pass aggregation among connected nodes in the input graph. In contrast, over-aggregating occurs in the attention graph due to the large number of nodes in the global receptive field. Designing high-pass aggregating functions can tackle over-smoothing. Conversely, for over-aggregating on graph transformers, even when a high-pass behavior is desired, the optimization of attention weights may still be hindered by the overwhelming number of nodes involved in aggregation.

Another notable issue on input graphs is over-squashing, which occurs when message pathways include edges with negative curvatures. It leads to an exponential expansion of the receptive field as the message-passing distance increases [1]. In consequence, a large volume of messages is compressed into fixed-dimensional representations, impeding the effective message exchange between long-range nodes. Unlike over-squashing that occurs during multi-step message passing, over-aggregating arises in one step. It aggregates a large volume of messages into a single node with limited discrimination, leading to the loss of informative messages from one-hop neighbors.

4 Wideformer: Relieving the Over-Aggregating

Based on the analysis in Sec. 3, global attention, including both dense and linear methods, faces a critical trade-off between mitigating over-aggregating and maintaining a global receptive field. To address this challenge, we propose Wideformer, a plug-and-play method designed to enhance graph transformer backbones. Unlike sparse attention methods that reduce the input volume, Wideformer maintains the total inputs and increases the output volume. Specifically, it partitions the inputs into distinct clusters, giving rise to multiple aggregation results for each node and guiding the nodes to focus on the informative clusters. The following subsections detail the two key components of Wideformer: dividing the aggregation and guiding the attention.

4.1 Dividing the Aggregation

Message passing involves aggregating messages from source nodes to target nodes. For each target node v_i in global attention, it treats all the input nodes $\{v_j\}, j \in \{1, \cdots, n\}$ as the source nodes, and aggregate messages $\mathbf{V}_{j,\cdot}$ based on the attention scores between its query features $\mathbf{Q}_{i,\cdot}$ and the

source key features $\mathbf{K}_{j,\cdot}$. As a result, messages from all source nodes at each feature dimension are compressed into a single-dimensional representation, leading to the potential dilution of the key messages.

To relieve the dilution, we propose to divide the source nodes into different clusters based on their attention scores and aggregate each cluster separately. Therefore, source nodes with the highest level of attention can be aggregated together, excluding the other nodes and relieving the message dilution. However, explicitly computing the attention matrix requires a space complexity of $O(n^2)$. To address this issue, Wideformer approximates the division with attention scores by first selecting cluster centers among the query features, and then assigning the source nodes to the clusters based on the similarity between their key features and the centers.

Algorithm 1 Center-selection Function `Cluster`

Input: query $\mathbf{Q} \in \mathbb{R}^{n\times d}$, number of clusters m
Initialize center $\mathbf{C} = \{0\}_{m\times d}$
$\mathbf{C}_{1,\cdot} = \mathbf{Q}_{t,\cdot},\ t = \arg\max_i \sum_j \mathbf{Q}_{i,j}$
for $t = 1$ **to** m **do**
  Compute distances: $\mathbf{q} = \max_j \left[(\mathbf{Q}\mathbf{C}^\top)_{\cdot,j}\right]$
  Select next center indices: $q = \arg\min_i(\mathbf{q}_i)$
  Update centers: $\mathbf{C}_{t,\cdot} = \mathbf{Q}_{q,\cdot}$
end for
Output: centers $\mathbf{C} \in \mathbb{R}^{m\times d}$

Let m be the number of clusters, $\mathbf{C} = \texttt{Cluster}(\mathbf{Q}, m) \in \mathbb{R}^{m\times d}$ be the cluster centers. The center-selection function `Cluter` in Algorithm 1 is implemented based on K-Means++ [3]. The query of the source node with the largest summation across feature dimensions is selected as the initial cluster center. Notably, this initial center is solely used to initiate the selection process and does not participate in the aggregation. `Cluter` then select m centers sequentially, where the node that minimizes the maximum similarity to all previously selected centers is designated as a new cluster center.

The source nodes are assigned to the identified query clusters based on the similarity between $\mathbf{K}$ and $\mathbf{C}$. The assignment result $\mathbf{k} \in \mathbb{R}^n$ can be formulated as

$$\mathbf{k}_i = \arg\max_j \left[\left(\mathbf{K}\mathbf{C}^\top\right)_{i,j}\right]. \tag{5}$$

By performing aggregation within distinct clusters, Wideformer produces multiple outputs for each target node. The aggregation result of the t-th cluster $\mathbf{H}^{(t)} \in \mathbb{R}^{n\times d}$ can be formulated as

$$\mathbf{H}^{(t)} = \sum_{i\in\{j|\mathbf{k}_j=t\}} \texttt{softmax}(\mathbf{Q}\mathbf{K}_{i,\cdot}^\top)\mathbf{V}_{i,\cdot}\,. \tag{6}$$

To illustrate the effect of aggregation, for each target node v_i and the j-th feature dimension of the source message $\mathbf{V}$, the original aggregation method in Eq. 1 compresses messages from all nodes into a one-dimensional representation $\mathbf{H}_{i,j}$, which leads to the dilution of key messages and over-aggregating. In contrast, Wideformer with Eq. 6 increases the output volume, yielding a m-dimensional representation $\{\mathbf{H}_{i,j}^{(1)}, \cdots, \mathbf{H}_{i,j}^{(m)}\}$. Only a limited volume of messages is aggregated for each cluster, preventing the dilution of the key messages. Even when the target nodes apply uniform attention scores to the source nodes, the messages in different clusters remain separate from being over-aggregated.

4.2 Guiding the Attention

The aggregation results $\{\mathbf{H}^{(1)}, \cdots, \mathbf{H}^{(m)}\}$ are concatenated for the subsequent modules in backbones. During the concatenation, the current ordering of the aggregated results depends on the cluster-selection order. However, since each target node assigns different attention scores to the clusters, this order does not reflect the importance of each cluster for the target node. To ensure consistent ordering and enable the target nodes to focus on the informative clusters, Wideformer sorts and weights the clusters for each target node based on the cluster attention score $\bar{\boldsymbol{\alpha}}_{i,j} \in \mathbb{R}^{n\times m}$. These scores can be formulated as

$$\bar{\boldsymbol{\alpha}}_{i,j} = \texttt{softmax}(\mathbf{Q}\bar{\mathbf{K}}^\top)_{i,j}, \qquad \bar{\mathbf{K}}_{t,\cdot} = \frac{1}{|\{v_i|\mathbf{k}_i = t\}|} \sum_{i\in\{j|\mathbf{k}_j=t\}} \mathbf{K}_{i,\cdot}, \tag{7}$$

where $\bar{\mathbf{K}} \in \mathbb{R}^{m\times d}$ denotes the average key features of the clusters. The sorted and weighted aggregation result $\hat{\mathbf{H}}^{(t)} \in \mathbb{R}^{n\times d}$ is then obtained as

$$\hat{\mathbf{H}}_{i,\cdot}^{(t)} = \bar{\boldsymbol{\alpha}}_{i,\mathbf{S}_{i,t}} \mathbf{H}_{i,\cdot}^{(\mathbf{S}_{i,t})}, \qquad \mathbf{S}_{i,\cdot} = \arg\texttt{sort}_j\left[\bar{\boldsymbol{\alpha}}_{i,j}\right], \tag{8}$$

Table 2: **Comparison Results on Homophilic Datasets (Measured by accuracy: %).** OOM indicates out-of-memory under the experimental setups.

	AMAZONCOMPUTERS	COAUTHORCS	AMAZONPHOTO	COAUTHORPHYSICS
#NODES	13,381	18,333	7,487	34,493
#FEATURES	767	6,805	745	8,415
#CLASSES	10	15	8	5
GCN	89.65±0.52	92.92±0.12	92.70±0.20	96.18±0.07
GRAPHSAGE	91.20±0.29	93.91±0.13	94.59±0.14	96.49±0.06
GAT	90.78±0.13	93.61±0.14	93.87±0.11	96.17±0.08
GPRGNN	89.32±0.29	95.13±0.09	94.49±0.14	96.85±0.08
NAGPHORMER	91.19±0.14	95.75±0.09	95.49±0.11	97.34±0.03
NODEFORMER	86.98±0.62	95.64±0.22	93.46±0.35	96.45±0.28
DIFFORMER	91.99±0.76	94.78±0.20	95.10±0.47	96.60±0.18
GOAT	90.96±0.90	94.21±0.38	92.96±1.48	96.24±0.24
EXPHORMER	91.47±0.17	94.93±0.01	95.35±0.22	96.89±0.09
GRAPHGPS	90.62±0.36	95.44±0.03	94.98±0.16	96.75±0.07
+ENTROPY REG	90.68±0.37	95.50±0.19	95.28±0.28	OOM
+WIDEFORMER	90.84±0.57	**95.85**±0.07	95.55±0.74	96.95±0.03
SGFORMER	91.66±0.23	93.50±0.32	95.47±0.66	96.87±0.08
+ENTROPY REG	91.78±0.37	93.59±0.41	95.51±0.49	96.88±0.09
+WIDEFORMER	92.19±0.17	93.86±0.41	95.64±0.57	**97.39**±0.19
POLYNORMER	91.98±0.09	94.53±0.61	95.47±0.53	96.67±0.29
+ENTROPY REG	92.08±0.07	94.57±0.43	95.52±0.44	96.69±0.31
+WIDEFORMER	**92.39**±0.07	94.60±0.27	**95.64**±0.36	96.79±0.20

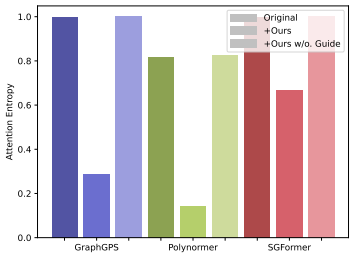
where arg `sort` returns the indices of the sorted elements in ascending order. Eq. 7 and 8 inject the importance of each cluster into the aggregation, guiding the focus of the target nodes. The weighted results are concatenated for the following computation.

5 Experiment

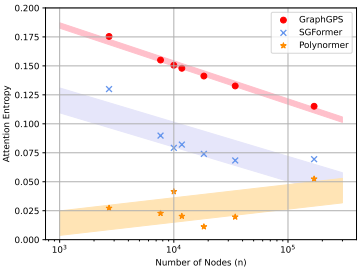
To evaluate the effectiveness of Wideformer, we conduct empirical evaluations on real-world datasets. The detailed setups are presented in Appendix D.

5.1 Effectiveness in tackling Over-Aggregating

Wideformer is evaluated on three linear methods, including GraphGPS with Performer [34], SGFormer [51], and Polynormer [9]. To ensure consistent comparison, these methods are implemented under the framework of GraphGPS. As presented in Fig. 3(a), Wideformer can effectively reduce the attention entropy of all backbone methods. This reduction allows target nodes to focus on informative source nodes within the global receptive field, effectively preventing the dilution of key messages and mitigating over-aggregating.



(a) Attention Entropy Comparison



(b) Cluster Attention Entropy

Figure 3: **Effectiveness Evaluation for Over-Aggregating.** "+ Ours" denotes using our proposed Wideformer. "+ Ours w/o. Guide" employs Wideformer without attention guidance.

However, Wideformer without the attention-guiding process still exhibits high attention entropy in Fig. 3(a). This partial method only avoids message dilution but fails to effectively guide the attention of the target nodes. To further investigate the attention-guiding process in Wideformer, we conduct an empirical analysis of cluster attention $\bar{\alpha}$. As shown in Fig. 3(b), Wideformer provides low cluster attention entropy for all three backbones, indicating that only a subset of input messages is crucial for the target nodes. The attention-guiding process in Wideformer allows backbone methods to concentrate on these informative subsets, enhancing their ability to mitigate over-aggregating. As the number of nodes increases, GraphGPS and SGFormer exhibit increasingly divergent attention scores across clusters, while Polynormer maintains much lower cluster attention entropy, albeit with a slight upward trend. This indicates that Wideformer consistently supports different backbone methods in prioritizing informative messages, even when the graph size grows.

Table 3: **Comparison Results on Heterophilic Datasets (Measured by ROC-AUC except for accuracy for amazon-ratings and roman-empire: %).** OOM indicates out-of-memory under the experimental setups.

	AMAZON-RATINGS	MINESWEEPER	QUESTIONS	ROMAN-EMPIRE	TOLOKERS
#NODES	24,492	10,000	48,921	22,662	11,758
#FEATURES	300	7	301	300	10
#CLASSES	5	2	2	18	2
GCN	48.70±0.63	89.75±0.52	76.09±1.27	73.69±0.74	83.64±0.67
GRAPHSAGE	53.63±0.39	93.51±0.57	76.44±0.62	85.74±0.67	82.43±0.44
GAT	52.70±0.62	93.91±0.35	76.79±0.71	88.75±0.41	83.78±0.43
GPRGNN	44.88±0.34	86.24±0.61	55.48±0.91	64.85±0.27	72.94±0.97
NAGPHORMER	51.26±0.72	84.19±0.66	68.17±1.53	74.34±0.77	78.32±0.95
NODEFORMER	43.86±0.35	86.71±0.88	74.27±1.46	64.49±0.73	78.10±1.03
DIFFORMER	47.84±0.65	90.89±0.58	72.15±1.31	79.10±0.32	83.57±0.68
GOAT	44.61±0.50	81.09±1.02	75.76±1.66	71.59±1.25	83.11±1.04
EXPHORMER	53.51±0.46	84.19±0.53	73.94±1.06	89.03±0.37	83.77±0.78
GRAPHGPS	49.73±0.11	93.26±0.10	75.48±0.66	81.46±0.40	83.95±0.81
+ENTROPY REG	OOM	93.35±0.23	OOM	OOM	84.23±0.51
+WIDEFORMER	49.96±0.41	93.52±0.07	75.69±1.17	82.12±0.39	84.67±0.66
SGFORMER	52.38±0.22	88.60±0.49	76.81±0.09	75.20±0.89	82.24±0.13
+ENTROPY REG	52.86±0.29	88.87±0.33	76.83±0.18	75.84±0.43	82.39±0.19
+WIDEFORMER	53.47±0.14	89.02±0.22	76.94±0.05	77.03±0.38	82.55±0.03
POLYNORMER	54.71±0.17	97.13±0.17	78.66±0.50	91.83±0.16	85.09±0.21
+ENTROPY REG	54.79±0.23	97.20±0.32	78.69±0.24	91.94±0.34	85.21±0.33
+WIDEFORMER	**55.05**±0.08	**97.26**±0.01	**79.00**±0.20	**92.16**±0.24	**85.33**±0.23

Table 4: **Comparison Results on Datasets with Broader Scales (Measured by accuracy: %).** OOM indicates out-of-memory under the experimental setups.

	OGB-ARXIV	CITESEER	CORA	TWITCH-GAMER
#NODES	169,343	3,327	2,708	168,114
#FEATURES	128	3,703	1,433	7
#CLASSES	40	6	7	2
GCN	71.74±0.29	71.60±0.40	81.60±0.40	62.18±0.26
GRAPHSAGE	71.49±0.27	71.93±0.85	82.68±0.47	64.37±0.39
GAT	71.95±0.36	72.10±1.10	83.00±0.70	59.89±4.12
EXPHORMER	72.44±0.28	71.63±1.19	82.77±1.38	64.30±0.16
GRAPHGPS	70.47±1.56	77.96±0.37	86.03±0.54	64.98±0.26
+ENTROPY REG	OOM	78.19±0.39	86.23±0.32	OOM
+WIDEFORMER	70.66±1.29	**78.61**±0.35	**86.70**±0.83	65.46±0.20
SGFORMER	72.23±0.73	69.93±0.31	80.83±0.52	65.85±0.02
+ENTROPY REG	OOM	70.12±0.41	81.04±0.38	OOM
+WIDEFORMER	**72.58**±0.39	70.63±0.78	81.20±0.37	66.34±0.37
POLYNORMER	70.77±1.16	66.43±1.50	78.70±0.94	67.15±0.03
+ENTROPY REG	OOM	67.21±0.35	79.28±0.51	OOM
+WIDEFORMER	70.88±0.10	68.10±0.42	79.90±0.86	**67.33**±0.15

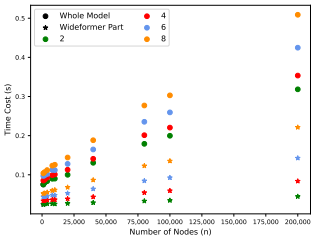


Figure 4: **Time Cost on GraphGPS.** The point shape "." denotes the whole model and "*" denotes the Wideformer part. The color of the points indicates the number of the clusters m.

5.2 Model Comparison

Experimental Setup. We adopt thirteen real-world datasets, including both heterophilic and homophilic graphs. For baseline methods, both GNNs and graph transformers are adopted. Please refer to Appendix D.2 for detailed experimental setups. To address the out-of-memory issue associated with the entropy regularization baseline, we divide the target nodes into mini-batches. Each batch is sequentially processed to compute attention scores with all source nodes. The results of the backbone methods GraphGPS with Performer, SGFormer, and Polynormer are reproduced under the framework of GraphGPS.

Performance. We present the comparison results on datasets where the number of nodes is around 10k in Tab. 2 and Tab. 3. Wideformer consistently enhances the performance of all backbone models across various datasets. Specifically, backbones integrated with Wideformer achieve superior performance compared to baseline methods. For instance, in Tab. 2, Wideformer enables backbone methods to outperform the previous best-performing baselines, including DIFFormer on AmazonComputers and NAGphormer on CoauthorsCS, CoauthorPhysics, and AmazonPhoto. Compared to the entropy regularization method, we can see that regularizing entropy results in inferior performance to Wideformer and even marginal gains on larger graphs. This is consistent with our analysis in Sec. 3.2. Entropy regularization does not divide the aggregation process, and thus suffers from optimization challenges when aggregating over a large number of source nodes. In contrast, Wideformer separates

the aggregation into parallel processes. Each process only involves a small ratio of the source nodes to be aggregated, preventing the dilution of the key messages. We also compare the gains with zero to evaluate the statistical significance of the performance gains. Paired t-tests on all backbones consistently give rise to p-values smaller than 0.05, and the confidence interval results do not contain 0, validating the statistical significance of the performance gains with Wideformer. These results demonstrate that relieving over-aggregating improves the effectiveness of graph attention.

To further verify Wideformer's ability to mitigate over-aggregating, we extend our experiments to datasets with broader node scales, including 1k and 0.1M nodes. Results in Tab. 4 show that Wideformer can benefit backbones with different scales of nodes. Among the three backbones, SGFormer achieves better results on large-scale graphs ogb-arxiv and twitch-gamer, but inferior results on small-scale graphs CiteSeer and Cora. This is due to the trade-off between noise filtering and information loss in SGFormer, which is further investigated in Sec. 5.3.1.

5.3 Model Analysis

5.3.1 Informative Source Nodes

Wideformer empowers backbone methods to focus on informative source nodes within the global receptive field. To investigate the attention mechanism in Wideformer, we conduct additional experiments on source nodes assigned to the cluster with the highest attention score, analyzing both their volumes and their relations with the target nodes.

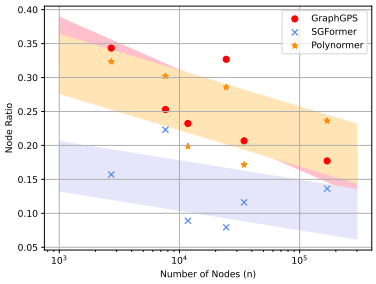


Figure 5: **Informative Source Node Ratio.** The ratio is the number of nodes assigned to the cluster with the highest attention score relative to the total number of graph nodes.

Node Ratio. To analyze the node volume, we calculate the ratio of nodes assigned to the cluster with the highest attention score relative to the total number of graph nodes. As shown in Fig. 5, the node ratio decreases as the total number of nodes increases. This suggests that the proportion of informative nodes diminishes in larger graphs, leading to noisier source messages. Wideformer addresses this by assigning a smaller ratio of nodes to the cluster with the highest attention score, guiding the backbones to focus on the most informative nodes. Among the three backbones, SGFormer focuses on fewer source nodes than GraphGPS and Polynormer. This results in a trade-off between noise filtering and information loss, where SGFormer performs better on large-scale graphs (ogb-arxiv and twitch-gamer) but lags behind on smaller graphs (CiteSeer and Cora).

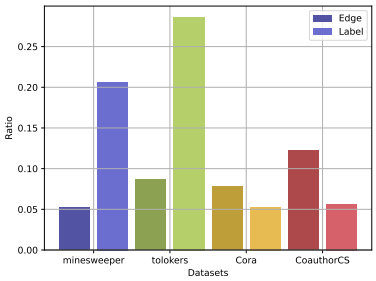


Figure 6: **Node Relations.** Label/Edge denotes the ratio of source nodes sharing the same label/connected with the target nodes.

Node Relations. To analyze the relations between target and source nodes in the cluster with the highest attention score, we compute the ratio of the source nodes (1) that have the same label as the target nodes, and (2) that are connected with the target nodes, both relative to the total number of source nodes in the cluster. The results with GraphGPS are presented in Fig. 6. For the other two backbones, please refer to Appendix E. As shown in Fig. 6, we can see that approximately 10% of source nodes are connected with their target nodes. The label distribution diverges between heterophilic graphs and homophilic graphs, where target nodes assign larger attention scores to the same-class source nodes on heterophilic graphs.

5.3.2 Complexity Analysis

The time and space complexity of Wideformer is $O(nm)$, where n and m denote the number of nodes and clusters, respectively. In practice, m is chosen from $\{2, ..., 8\}$, which ensures the model's scalability to large graphs. We provide a time cost study regarding the number of nodes (n) and clusters (m), comparing the whole model with the part of Wideformer. The results on GraphGPS are presented in Fig. 4, where the time cost of Wideformer scales linearly to n and remains a small ratio of the total cost. For detailed experimental setup and full results, please refer to Appendix F.

5.3.3 Model Ablation

To evaluate the contribution of the aggregation-dividing and attention-guiding process in Wideformer, we conduct an ablation study on the three backbone methods.

Cluster Ablation. Wideformer partitions the source nodes into distinct clusters. To examine the impact of this division, we perform an ablation study on the number of clusters m regarding performance gains. The gain is defined as $\frac{p_m - p_1}{p_1}$, where p_1 and p_m denote the performance with a single cluster and m clusters, respectively. The average performance gains across various datasets for all three backbones are shown in Fig. 7. We can see that increasing the number of clusters initially increases the performance gain. However, when m exceeds 6, the performance gain starts to decline. This is because a relatively larger number of clusters can effectively reduce the input message volume, mitigating the dilution of key information. Nevertheless, excessively partitioning the source nodes disperses critical information into multiple clusters. This leads to the distracted attention of the target nodes, reducing the values of attention scores and increasing attention entropy. In practice, the optimal number of clusters is around $3 \sim 5$.

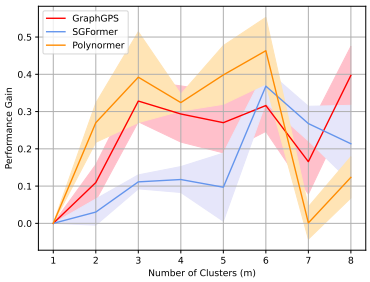


Figure 7: **Comparison with Different Numbers of Clusters.**

Center Selection. To ensure computing efficiency, Wideformer employs a non-iterative strategy for center selection. We now compare this simple choice with (1) multi-step iterative clustering and (2) parameterized centers, to shed light on the future directions. Tab. 5 shows that applying iterative clustering can achieve better results, indicating the efficiency-performance trade-offs. Further modeling centers as parameters can address this trade-off, which does not require iterations and achieves the best results.

Table 5: **Comparison on the Center Selection Methods.** Results are averaged on amazon-ratings, questions, CoauthorCS, and AmazonPhoto.

	GRAPHGPS	SGFORMER	POLYNORMER
ORIGINAL	78.91±0.24	79.54±0.32	80.84±0.45
+WIDEFORMER	79.26±0.59	79.98±0.29	81.07±0.23
+ITER=2	79.93±0.45	80.58±0.39	81.44±0.29
+ITER=4	79.95±0.57	80.42±0.37	81.43±0.21
+ITER=6	79.85±0.44	80.55±0.34	81.52±0.23
+LEARNABLE	**80.05**±0.46	**80.60**±0.39	**81.54**±0.20

Module Ablation. Tab. 6 shows that combining divided aggregation and attention guidance generally achieves better performance. However, using only divided aggregation can also benefit the backbones, such as on CoauthorCS and AmazonPhoto with Polynormer. This is likely due to the simple attention-guiding strategy, which directly determines the importance of each cluster as a whole. Under circumstances where fewer source nodes contain informative messages, this simple strategy leads to limited benefit capability. Exploring alternative guiding strategies will be the focus of our future work.

Table 6: **Module Ablation Results (Measured by accuracy: %).** "Div."/"All" denotes the divided aggregation without/with attention-guidance.

	AMAZON -RATINGS	COAUTHOR CS	AMAZON PHOTO	ROMAN -EMPIRE
GRAPHGPS	49.73±0.11	95.44±0.03	94.98±0.16	81.46±0.40
+DIV.	49.58±0.54	95.56±0.30	94.98±0.61	81.40±0.54
+ALL	**49.96**±0.41	**95.65**±0.07	**95.55**±0.74	**82.12**±0.39
SGFORMER	52.38±0.22	93.50±0.32	95.47±0.66	75.20±0.89
+DIV.	52.42±0.54	93.53±0.21	95.58±0.53	**77.38**±0.54
+ALL	**53.47**±0.14	**93.86**±0.41	**95.64**±0.57	77.03±0.38
POLYNORMER	54.71±0.17	94.53±0.61	95.47±0.53	91.83±0.16
+DIV.	54.82±0.11	**94.90**±0.30	**96.74**±0.30	92.04±0.15
+ALL	**55.05**±0.08	94.60±0.27	95.64±0.36	**92.16**±0.24

6 Conclusion

In this paper, we discovered a phenomenon in graph attention, termed over-aggregating. Attention with over-aggregating assigns uniformly distributed attention scores to the messages. As a result, the key messages are diluted during aggregation, leading to potential information loss. To relieve this problem, we proposed a novel plug-and-play method, Wideformer. Wideformer divides the source nodes for aggregation into distinct clusters and aggregates each cluster separately. To ensure focus on the informative source nodes, Wideformer applies attention guidance to sort and weight the clusters. Experiments on thirteen datasets show that Wideformer can consistently enhance the performance of backbones and effectively relieve over-aggregating. Please refer to Appendix. I for the limitation discussion.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Key R&D Program of China under Grant 2023YFC2508704, in part by the National Natural Science Foundation of China 62236008, and in part by the Fundamental Research Funds for the Central Universities. The authors would like to thank the anonymous reviewers for their helpful comments and suggestions that improved this manuscript.

References

[1] Uri Alon and Eran Yahav. On the Bottleneck of Graph Neural Networks and its Practical Implications. In *International Conference on Learning Representations*, Virtual Only, 2021.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A Video Vision Transformer. In *IEEE/CVF International Conference on Computer Vision*, pages 6816–6826, 2021.

[3] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.

[4] Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. NAGphormer: A Tokenized Graph Transformer for Node Classification in Large Graphs. In *International Conference for Learning Representations*, 2023.

[5] Junxi Chen, Liang Li, Yunbin Tu, Li Su, Zhe Xue, and Qingming Huang. Generalizing single-frame supervision to event-level understanding for video anomaly detection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[6] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 257–266, Anchorage AK USA, 2019. ACM.

[7] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive Universal Generalized PageRank Graph Neural Network. In *International Conference on Learning Representations*, virtual, 2022.

[8] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking Attention with Performers. In *International Conference on Learning Representations*, 2020.

[9] Chenhui Deng, Zichao Yue, and Zhiru Zhang. Polynormer: Polynomial-Expressive Graph Transformer in Linear Time. In *International Conference on Learning Representations*, 2023.

[10] Vijay Prakash Dwivedi and Xavier Bresson. A Generalization of Transformer Networks to Graphs. In *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

[11] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph Neural Networks with Learnable Structural and Positional Representations. 2021.

[12] Matthias Fey and Jan Eric Lenssen. Fast Graph Representation Learning with PyTorch Geometric. In *International Conference on Learning Representations Workshop on Graphs and Manifolds*, 2019.

[13] Benjamin Gutteridge, Xiaowen Dong, Michael M. Bronstein, and Francesco Di Giovanni. DRew: Dynamically Rewired Message Passing with Delay. In *International Conference on Machine Learning*, pages 12252–12267. PMLR, 2023.

[14] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *International Conference on Neural Information Processing Systems*, pages 1025–1035, Red Hook, USA, 2017. Curran Associates Inc.

[15] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Advances in Neural Information Processing Systems*, volume 33, pages 22118–22133. Curran Associates, Inc., 2020.

[16] Nam Hyeon-Woo, Kim Yu-Ji, Byeongho Heo, Dongyoon Han, Seong Joon Oh, and Tae-Hyun Oh. Scratching Visual Transformer's Back with Uniform Attention. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5784–5795, 2023.

[17] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, Toulon, France, 2017.

[18] Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Renkun Ni, C. Bayan Bruss, and Tom Goldstein. GOAT: A Global Transformer on Large-scale Graphs. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17375–17390. PMLR, 2023.

[19] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking Graph Transformers with Spectral Attention. In *Advances in Neural Information Processing Systems*, volume 34, pages 21618–21629. Curran Associates, Inc., 2021.

[20] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-Attention Graph Pooling. In *International Conference on Machine Learning*, pages 3734–3743, Long Beach, California, USA, 2019. PMLR.

[21] Junhyun Lee, Veronika Thost, Bumsoo Kim, Jaewoo Kang, and Tengfei Ma. Understanding and Tackling Over-Dilution in Graph Neural Networks. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pages 1253–1261, New York, NY, USA, 2025. Association for Computing Machinery.

[22] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, 2014.

[23] Yuan Li, Xiaodan Liang, Zhiting Hu, Yinbo Chen, and Eric P. Xing. Graph Transformer. 2018.

[24] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods. In *Advances in Neural Information Processing Systems*, volume 34, pages 20887–20902. Curran Associates, Inc., 2021.

[25] Jinzhe Liu, Junshu Sun, Shufan Shen, Chenxue Yang, and Shuhui Wang. Edit Less, Achieve More: Dynamic Sparse Neuron Masking for Lifelong Knowledge Editing in LLMs, 2025.

[26] Shih-yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-Decomposed Low-Rank Adaptation. In *Forty-First International Conference on Machine Learning*, 2024.

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *IEEE/CVF International Conference on Computer Vision*, pages 9992–10002, 2021.

[28] Diego Mesquita, Amauri H. Souza, and Samuel Kaski. Rethinking pooling in graph neural networks. *arXiv:2010.11418 [cs]*, 2020.

[29] Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. GraphiT: Encoding Graph Structure in Transformers. *arXiv:2106.05667 [cs]*, 2021.

[30] Kenta Oono and Taiji Suzuki. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference for Learning Representations*, 2021.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *International Conference on Neural Information Processing Systems*, pages 8026–8037, Red Hook, NY, USA, 2019. Curran Associates Inc.

[32] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, 2023.

[33] Saeed Rahmani, Asiye Baghbani, Nizar Bouguila, and Zachary Patterson. Graph Neural Networks for Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 24(8):8846–8885, 2023.

[34] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. In *Advances in Neural Information Processing Systems*, volume 35, pages 14501–14515, New Orleans, USA, 2022.

[35] Giorgos Savathrakis and Antonis Argyros. Enact: Entropy-Based Clustering of Attention Input for Reducing the Computational Needs of Object Detection Transformers. In *2025 IEEE International Conference on Image Processing (ICIP)*, pages 295–300, 2025.

[36] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective Classification in Network Data. *AI Magazine*, 29(3):93, 2008.

[37] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of Graph Neural Network Evaluation. *arXiv:1811.05868 [cs, stat]*, 2019.

[38] Shufan Shen, Zhaobo Qi, Junshu Sun, Qingming Huang, Qi Tian, and Shuhui Wang. Enhancing Pre-trained Representation Classifiability can Boost its Interpretability. In *The Thirteenth International Conference on Learning Representations*, 2024.

[39] Shufan Shen, Junshu Sun, Qingming Huang, and Shuhui Wang. VL-SAE: Interpreting and Enhancing Vision-Language Alignment with a Unified Concept Set, 2025.

[40] Shufan Shen, Junshu Sun, Xiangyang Ji, Qingming Huang, and Shuhui Wang. Expanding Sparse Tuning for Low Memory Usage. In *Advances in Neural Information Processing Systems*, volume 37, pages 76616–76642, 2024.

[41] Hamed Shirzad, Honghao Lin, Balaji Venkatachalam, Ameya Velingker, David P. Woodruff, and Danica J. Sutherland. Even Sparser Graph Transformers. In *Advances in Neural Information Processing Systems*, volume 37, pages 71277–71305, 2024.

[42] Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J. Sutherland, and Ali Kemal Sinop. Exphormer: Sparse Transformers for Graphs. In *International Conference on Machine Learning*, volume 202, Honolulu, USA, 2023. PMLR.

[43] Junshu Sun, Shuhui Wang, Xinzhe Han, Zhe Xue, and Qingming Huang. All in a row: Compressed convolution networks for graphs. In *International Conference on Machine Learning*, volume 202, pages 33061–33076, Honolulu, USA, 2023. PMLR.

[44] Junshu Sun, Shuhui Wang, Chenxue Yang, and Qingming Huang. Scalable Graph Compressed Convolutions, October 2024.

[45] Junshu Sun, Chenxue Yang, Xiangyang Ji, Qingming Huang, and Shuhui Wang. Towards Dynamic Message Passing on Graphs. In *Conference on Neural Information Processing Systems*, December 2024.

[46] Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference for Learning Representations*, 2021.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *International Conference on Neural Information Processing Systems*, pages 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, Vancouver, Canada, 2018.

[49] Qitian Wu, Chenxiao Yang, Wentao Zhao, Yixuan He, David Wipf, and Junchi Yan. DIFFormer: Scalable (Graph) Transformers Induced by Energy Constrained Diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

[50] Qitian Wu, Wentao Zhao, Zenan Li, David Wipf, and Junchi Yan. NodeFormer: A Scalable Graph Structure Learning Transformer for Node Classification. In *Advances in Neural Information Processing Systems*, 2022.

[51] Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. SGFormer: Simplifying and Empowering Transformers for Large-Graph Representations. In *Conference on Neural Information Processing Systems*, 2023.

[52] Yue Wu, Zhaobo Qi, Junshu Sun, Yaowei Wang, Qingming Huang, and Shuhui Wang. Video Language Model Pretraining with Spatio-temporal Masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8557–8567, 2025.

[53] Zhanghao Wu, Paras Jain, Matthew Wright, Azalia Mirhoseini, Joseph E Gonzalez, and Ion Stoica. Representing Long-Range Context for Graph Neural Networks with Global Attention. In *Advances in Neural Information Processing Systems*, volume 34, pages 13266–13279. Curran Associates, Inc., 2021.

[54] Yujie Xing, Xiao Wang, Yibo Li, Hai Huang, and Chuan Shi. Less is More: On the Over-Globalizing Problem in Graph Transformers. In *International Conference on Machine Learning*, pages 54656–54672. PMLR, 2024.

[55] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do Transformers Really Perform Bad for Graph Representation? In *Advances in Neural Information Processing Systems*, volume 34, pages 28877–28888. Curran Associates, Inc., 2021.

[56] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical Graph Representation Learning with Differentiable Pooling. In *Advances in Neural Information Processing Systems*, volume 31, pages 4805–4815, Montréal, Canada, 2018. Curran Associates, Inc.

[57] Hao Yuan and Shuiwang Ji. StructPool: Structured Graph Pooling via Conditional Random Fields. In *International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020.

[58] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph Transformer Networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[59] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *International Conference on Neural Information Processing Systems*, pages 17283–17297, Red Hook, NY, USA, 2020. Curran Associates Inc.

[60] Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via Denoising for Molecular Property Prediction. In *The Eleventh International Conference on Learning Representations*, 2023.

[61] Bohang Zhang, Guhao Feng, Yiheng Du, Di He, and Liwei Wang. A Complete Expressiveness Hierarchy for Subgraph GNNs via Subgraph Weisfeiler-Lehman Tests. In *Proceedings of the 40th International Conference on Machine Learning*, pages 41019–41077. PMLR, 2023.

[62] Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. Graph-Bert: Only Attention is Needed for Learning Graph Representations. *arXiv:2001.05140 [cs, stat]*, 2020.

A Proof for Monotonic Lower Bound of Attention Entropy

Lemma A.1 (Lower Bound of Attention Entropy)**.** *Let $\boldsymbol{\alpha}_{i,\cdot} \in \mathbb{R}^n$ be the attention distribution of node v_i over n nodes. Suppose each entry satisfies $\boldsymbol{\alpha}_{i,j} \geq \epsilon > 0$. Then the attention entropy $\mathcal{H}(v_i)$ admits the following strict lower bound:*

$$\mathcal{H}(v_i) \geq -\left[1-(n-1)\epsilon\right]\log\left[1-(n-1)\epsilon\right] - (n-1)\epsilon\log\epsilon.$$

Proof. Assume $\epsilon > 0$ is a lower bound for all elements of $\boldsymbol{\alpha}_{i,\cdot}$. The minimum entropy under this constraint is attained when one entry takes the maximum possible weight and the remaining $(n-1)$ entries equal ϵ, *i.e.*,

$$\hat{\boldsymbol{\alpha}}_{i,\cdot} = \left(1-(n-1)\epsilon, \epsilon, \ldots, \epsilon\right).$$

The attention entropy of this distribution is

$$\mathcal{H}_{\min}(v_i) = -\left[1-(n-1)\epsilon\right]\log\left[1-(n-1)\epsilon\right] - (n-1)\epsilon\log\epsilon.$$

Since the Shannon entropy is strictly concave, any deviation from this distribution increases the entropy. Therefore, for any valid distribution $\boldsymbol{\alpha}_{i,\cdot}$ with elements bounded below by ϵ, it holds that

$$\mathcal{H}(v_i) \geq \mathcal{H}_{\min}(v_i).$$

□

We now provide the proof for Theorem 3.1 based on Lemma A.1.

Theorem 3.1 (Monotonic Lower Bound of Attention Entropy). *Let $\boldsymbol{\alpha}_{i,\cdot} \in \mathbb{R}^n$ be the attention distribution of node v_i over n nodes. The attention entropy $\mathcal{H}(v_i)$ admits a lower bound that increases monotonically with n.*

Proof. To show that $\mathcal{H}_{\min}(v_i)$ increases monotonically with n, we compute its derivative with respect to n

$$\begin{aligned}
\frac{d\mathcal{H}_{\min}(v_i)}{dn} &= \epsilon\log\left[1-(n-1)\epsilon\right] + \left[1-(n-1)\epsilon\right]\frac{\epsilon}{\left[1-(n-1)\epsilon\right]} - \epsilon\log\epsilon \\
&= \epsilon\log\left[1-(n-1)\epsilon\right] + \epsilon - \epsilon\log\epsilon \\
&= \epsilon\log\frac{1-(n-1)\epsilon}{\epsilon} + \epsilon.
\end{aligned}$$

Since $\hat{\boldsymbol{\alpha}}_{i,\cdot} = (1-(n-1)\epsilon, \epsilon, \cdots, \epsilon)$ denoting the most non-uniform distribution $\boldsymbol{\alpha}_{i,\cdot}$ requires $0 < \epsilon \leq 1/n$, it follows that $1-(n-1)\epsilon \geq \epsilon$, and thus

$$\frac{d\mathcal{H}_{\min}(v_i)}{dn} > 0.$$

This confirms that $\mathcal{H}_{\min}(v_i)$ increases monotonically with n. □

B Detailed Derivation

In this section, we provide a detailed derivation for Eq. 3. Let $\mathbf{R} = \mathbf{Q}\mathbf{K}^\top$. Given the attention scores formulated in Eq. 1, the gradient of $\boldsymbol{\alpha}_{i,j}$ with respect to $\mathbf{W}_Q$ can be derived as

$$\begin{aligned}
\frac{\partial\boldsymbol{\alpha}_{i,j}}{\partial\mathbf{W}_Q} &= \sum_{k=1}^{n}\frac{\partial\boldsymbol{\alpha}_{i,j}}{\partial\mathbf{R}_{i,k}}\frac{\partial\mathbf{R}_{i,k}}{\mathbf{W}_Q} \\
&= \sum_{k=1}^{n}\boldsymbol{\alpha}_{i,j}\left(\delta_{j,k}-\boldsymbol{\alpha}_{i,k}\right)\frac{\partial}{\partial\mathbf{W}_Q}\mathbf{X}_{i,\cdot}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}_{k,\cdot}^\top \\
&= \sum_{k=1}^{n}\boldsymbol{\alpha}_{i,j}\left(\delta_{j,k}-\boldsymbol{\alpha}_{i,k}\right)\mathbf{X}_{i,\cdot}^\top\mathbf{X}_{k,\cdot}\mathbf{W}_K^\top \\
&= \boldsymbol{\alpha}_{i,j}\left(\mathbf{X}_{i,\cdot}^\top\mathbf{X}_{j,\cdot}\mathbf{W}_K^\top - \sum_{k=1}^{n}\boldsymbol{\alpha}_{i,k}\mathbf{X}_{i,\cdot}^\top\mathbf{X}_{k,\cdot}\mathbf{W}_K^\top\right),
\end{aligned}$$

where $\delta_{j,k}$ is a sign function. When $k=j$, $\delta_{j,k}=1$, otherwise $\delta_{j,k}=0$.

Table S7: **Comparison Results with Entropy Regulation on the Cluster Attention (Measure by accuracy: %).** "Reg." denotes employing the attention entropy of the cluster attention scores as part of the optimization target.

	ROMAN-EMPIRE	COAUTHORCS
GRAPHGPS	81.46	95.44
+WIDEFORMER	82.12	95.85
+WIDEFORMER W/ REG.	**82.23**	**96.14**
SGFORMER	75.20	93.50
+WIDEFORMER	77.03	93.86
+WIDEFORMER W/ REG.	**77.83**	**94.16**
POLYNORMER	91.83	94.53
+WIDEFORMER	92.16	94.60
+WIDEFORMER W/ REG.	**92.36**	**95.42**

C Over-Aggregating Measurement

In this paper, we propose to employ entropy to quantify over-aggregating. This stems from the observation that the attention scores distribute uniformly among different source nodes. To quantify the uniformity of the distribution, we adopt entropy as the measure following the common practice in the community [56, 6, 32]. Except for entropy, other potential metrics include standard deviation (STD), Kullback-Leibler divergence (KLD), and Jensen-Shannon divergence (JSD). STD focuses on the dispersion of values. When the total number of values is large, but the number of dispersed values is small, the STD may remain small despite the distribution being uneven. Therefore, using STD to assess uniformity is not an accurate approach. KLD and JSD are used to assess the similarity between two distributions. It can measure the uniformity of a distribution by setting the reference distribution to be uniform. However, when the reference distribution is uniform, KLD equals the difference between the maximum entropy and the entropy of the given distribution, while JSD can be viewed as a combination of entropy and KLD. Therefore, we opted for the simpler and more commonly used entropy to directly measure the uniformity.

Entropy also demonstrates effectiveness in benefiting model performance. In addition to the entropy regularization method on the whole attention matrix (Tab. 1), we compute the entropy of the cluster attention scores in Wideformer and employ the results as part of the optimization target. Results on roman-empire and CoauthorCS in Tab. S7 show that optimizing the cluster attention entropy for backbones with Wideformer can also benefit the model performance. This also indicates that the value of the attention entropy properly quantifies over-aggregating.

In Fig. 3(a), a more significant entropy reduction occurs on Polynormer, but its associated performance gains are not superior to those of the other two backbone architectures. This is due to the distinct architectural design of the backbones. Different architectures inherently prioritize different attention patterns for optimal representation learning. While high attention entropy consistently indicates over-aggregating, the optimal entropy level varies across models. Thus, although Polynormer exhibits the most significant entropy reduction, this may still fall short of what is needed for its optimal performance.

D Details on Experiments

D.1 Datasets

We adopt thirteen real-world datasets, including heterophilic graphs (amazon-ratings, minesweeper, questions, roman-empire, tolokers [32], and twitch-gamer [24]) and homophilic graphs (Cora, CiteSeer [36], CoauthorCS, CoauthorPhysics, AmazonComputers, AmazonPhoto [37], and ogb-arxiv [15]).

Questions is derived from Yandex Q, spanning user activity from September 2021 to August 2022. Nodes represent users interested in "medicine" and edges indicate answers to others' questions. The binary classification task predicts whether users remained active without account deletion or blocking. Node features are averaged FastText embeddings of profile descriptions.

Amazon-ratings is based on the Amazon product co-purchase network from the SNAP Datasets [22]. Nodes represent products, and edges capture frequent co-purchase relationships. The task is to predict the average reviewer rating for each product, grouped into five ordinal classes. Node features are the average FastText embeddings of product descriptions.

Tolokers represents crowdsourcing participation data from the Toloka platform. Nodes correspond to contributors ("tolokers") active in at least one of 13 projects, with edges connecting those who completed the same tasks. The binary classification task predicts whether tolokers were banned from projects. Node features include profile attributes and task performance statistics.

Minesweeper is a synthetic 100x100 grid graph where nodes represent grid cells, with 20% randomly assigned as mines. The task is to classify nodes as mines or non-mines. Node features are one-hot vectors encoding the count of neighboring mines, but 50% of nodes have their features reset to unknown values, marked by a binary indicator.

Twitch-gamer is collected from the Twitch website with nodes representing users. Only users with mutual relations are incorporated in the dataset. The task is to predict whether the user accounts contain explicit content.

Cora and CiteSeer are citation graphs. In the Cora dataset, machine learning papers are grouped into seven distinct classes, while the CiteSeer dataset categorizes papers into six classes. Node features are derived from high-frequency words appearing in the content of the papers.

CoauthorCS and CoauthorPhysics are derived from the Microsoft Academic Graph. These datasets represent co-authorship networks, where nodes correspond to researchers, and edges signify their collaborative relationships. The features of each node reflect the frequency of keywords extracted from the publications of the respective authors. Graph labels identify the primary research domain of each author, distinguishing between computer science and physics.

AmazonComputers and AmazonPhoto are co-purchase networks constructed from Amazon. In these graphs, nodes represent products, while edges indicate co-purchase relationships between item pairs. Node features are derived from encoded customer review texts associated with each product, and graph labels classify the products into specific categories.

OGB-arXiv is a citation graph composed of Computer Science papers from arXiv. In this dataset, nodes correspond to individual articles, and directed edges represent citation relationships. Node features are computed as the average of 128-dimensional word embeddings extracted from the title and abstract of each paper. The task involves predicting the primary category of arXiv articles among 40 possible classes.

D.2 Experimental Setup

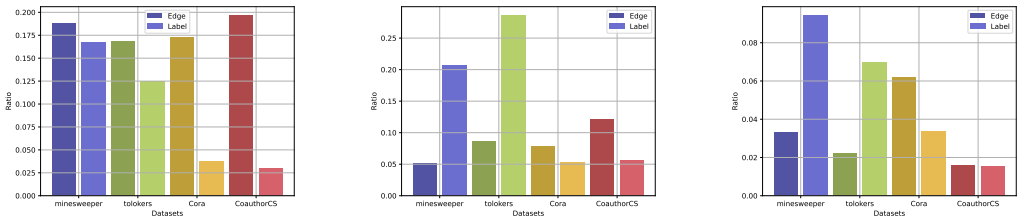
The attention entropy experiments in Fig. 1, Fig. 3, and Fig. 5 are conducted on Cora, Amazon-Photo, AmazonComputers, CoauthorCS, CoauthorPhysics, tolokers, amazon-ratings, minesweeper, and ogb-arxiv. The cluster ablation study in Fig. 7 is conducted and averaged on all the datasets included in this paper. The number of clusters is 4 in Fig. 3, Fig. 5-6, Fig. S8, and Tab. 5-6. Other experiments choose m from $\{2, \cdots, 8\}$. For baseline methods, both GNNs (GCN [17], GraphSAGE [14], GAT [48], GPRGNN [7]) and graph transformers (NAGphormer [4], NodeFormer [50], DIFFormer [49], GOAT [18]) are included. The results of the backbone methods, including GraphGPS with Performer, SGFormer, and Polynormer, are reproduced under the framework of GraphGPS. The framework is implemented with PyTorch [31] and PyTorch Geometric [12], and trained on a single NVIDIA A100. We perform grid search based on the validation performance of the models as follows:

GraphGPS. We search the number of graph transformer layers in $\{1, \cdots, 6\}$, the number of hidden dimensions in $\{64, 80, 128, 256\}$, the number of heads in $\{1, 2, 4\}$, dropout in $\{0.1, 0.2, 0.3, 0.5\}$, and learning rate in $\{5e-4, 1e-3, 1e-2\}$. The rest hyperparameters are fixed as in the original implementation.

SGFormer. The GNN backbone is implemented as GCN [17]. The number of GCN layers is searched in $\{1, \cdots, 10\}$, the number of hidden dimensions in $\{64, 80, 128, 256\}$, the number of

heads in $\{1, 2, 4\}$, dropout in $\{0.1, 0.2, 0.3, 0.5\}$, and learning rate in $\{1e-3, 5e-3, 1e-2\}$. The rest hyperparameters are fixed as in the original implementation.

Polynormer. The GNN backbone is implemented as GAT [48]. We search the number of graph transformer layers in $\{1, \cdots, 6\}$, the number of GAT layers in $\{1, \cdots, 10\}$, the number of hidden dimensions in $\{64, 80, 128, 256\}$, the number of heads in $\{1, 2, 4, 8\}$, dropout in $\{0.1, 0.2, 0.3, 0.5\}$, and learning rate in $\{5e-4, 1e-3\}$. The rest hyperparameters are fixed following the original implementation.

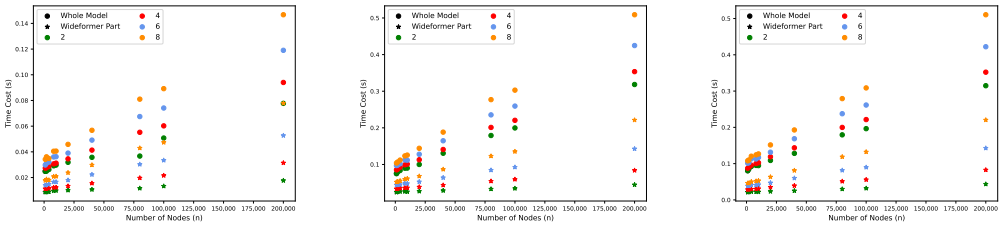


(a) SGFormer (b) GraphGPS (c) Polynormer

Figure S8: **Full Results for Relations between Source and Target Nodes.**

E Full Results for Informative Source Nodes

To analyze the relations between target and source nodes in the cluster with the highest attention score, we compute the ratio of the source nodes (1) that have the same label as the target nodes, and (2) that are connected with the target nodes, both relative to the total number of source nodes in the cluster. The results for all three backbones are presented in Fig. S8. The ratio of source nodes that are connected to their target nodes varies across the different backbones. Among the backbones, SGFormer assigns more attention to source nodes connected with the target nodes than Polynormer and GraphGPS. The label distribution differs between heterophilic and homophilic graphs, yet remains consistent across the backbones. On heterophilic graphs, target nodes assign higher attention scores to source nodes that have the same label as them. This stems from the heterophily of these graphs, where connected nodes often have different labels. Therefore, models only aggregate messages from nodes with different labels via local message passing and turn to global attention to collect messages from nodes with the same label as the target nodes.



(a) SGFormer (b) GraphGPS (c) Polynormer

Figure S9: **Time Cost.** Different colors of the points denote the number of clusters m. The point shape "." denotes the whole model with Wideformer. "*" denotes the Wideformer part.

F Complexity Analysis

We construct synthetic graphs to test the time cost across different graph scales. The average degree is set to 8, with a maximum number of edges equals $100,000$. The number of classes is set to 10. Both the number of input features and hidden dimensions are set to 64. Following the common configuration of the backbones, we set the number of local layers and global layers as (3, 1), (3, 3),

Table S8: **Average Cost Increase Ratio.** Comparing the time and memory cost of 4 clusters against 1 cluster. m denotes the number of clusters. The results are averaged across datasets with different numbers of graph nodes.

	M	TIME (S)	MEMORY (MB)
GRAPHGPS	1	0.0973	13,870
	4	$0.1408_{30.93\%\uparrow}$	$15{,}546_{10.78\%\uparrow}$
SGFORMER	1	0.0258	50,206
	4	$0.0413_{37.56\%\uparrow}$	$51{,}332_{2.19\%\uparrow}$
POLYNORMER	1	0.1042	29,338
	4	$0.1435_{27.37\%\uparrow}$	$30{,}180_{2.78\%\uparrow}$

Table S9: **Comparison Results with ENACT (Measure by accuracy except for ROC-AUC for questions: %).** The bold values denote the best results among comparisons with the same backbone.

	AMAZON-RATINGS	COAUTHORCS	AMAZONPHOTO	QUESTIONS
GRAPHGPS	49.73	95.44	94.98	75.48
+ENACT	48.52	95.16	94.76	**75.74**
+WIDEFORMER	**49.96**	**95.85**	**95.55**	75.69
SGFORMER	52.38	93.50	95.47	76.81
+ENACT	53.32	93.21	95.53	76.53
+WIDEFORMER	**53.47**	**93.86**	**95.64**	**76.94**
POLYNORMER	54.71	94.53	95.47	78.66
+ENACT	54.94	94.60	95.55	78.78
+WIDEFORMER	**55.05**	**94.94**	**95.64**	**79.00**

and (6, 3) for SGFormer, GraphGPS, and Polynormer, respectively. All models have 4 attention heads. From the results in Fig. S9, we can see that the time cost of Wideformer scales linearly with the number of nodes and remains a small ratio of the total time cost as the number of clusters increases.

In Sec 5.3.3, we show that increasing the number of clusters m can benefit the model performance, while the optimal value of m is around $3 \sim 5$. In practice, m is chosen from $\{2, \cdots, 8\}$ to remain a small value. Therefore, although a larger m increases the memory and runtime cost linearly, the additional cost is minor. We further present the average cost increase ratio (calculated as $\frac{\texttt{cost}(m=4)-\texttt{cost}(m=1)}{\texttt{cost}(m=1)}$) across different numbers of graph nodes in Tab. S8. We can see that the additional costs are relatively minor, amounting to only $\sim$ 30% of the original time cost and less than 11% of the memory cost.

G Broader Related Work

Our study is also related to other works beyond graph transformers. Wideformer employs the center selection method to assign source nodes to different clusters. Traditional cluster pooling methods focus on learning hierarchical features of input graphs [56, 20, 57, 28]. They can collect global information by stacking multiple layers. In contrast, Wideformer and other graph transformer models can collect global information in a single layer, giving rise to shallow architectures for large graphs.

To better optimize the learning process of the attention matrix with dense interactions, a previous study proposes to inject uniform attention to vision transformers [16]. However, this solution cannot be directly adopted for graph transformers. It requires explicitly computing the whole attention matrix, which demands a space complexity of $O(n^2)$ and is not scalable to large graphs. Moreover, different from the occasional observation of the uniform attention [16], we discover that uniform attention with high attention entropy consistently exists across different global graph attention methods on different datasets. To further reduce the computing complexity, ENACT proposes to cluster source nodes during attention-based aggregation for vision transformers [35]. However, ENACT cannot effectively alleviate over-aggregating. Specifically, their clustering process is based on the features of each individual node, while Wideformer clusters nodes based on the source-target relation, which better enables the selection of the informative sources for the targets. ENACT also applies attention

aggregation at the cluster level, while Wideformer applies attention aggregation at the node level within each cluster. Node-level aggregation allows the model to adjust the attention scores at a finer level and capture the node feature distribution within each cluster. To further compare Wideformer against ENACT for tackling over-aggregating, we implement ENACT in graph transformers by directly clustering source nodes based on their own features, and only applying cluster-level attention aggregation. Results in Tab. S9 show that ENACT cannot effectively tackle over-aggregating and gains worse performance than Wideformer. This demonstrates the effectiveness of our method in tackling over-aggregating and benefiting backbone performance.

H Societal Impact

This paper presents work whose goal is to advance the field of graph representation learning by proposing a plug-and-play method to tackle over-aggregating problem in graph transformers. The proposed method remains independent of specific downstream applications and can potentially benefit a wide range of domains, such as computational biology [60, 55] and intelligent transportation [33]. At present, we do not anticipate any evident ethical concerns or foreseeable adverse societal impacts.

I Limitation

This paper discovers the over-aggregating phenomenon in graph attention and proposes a plug-and-play method to tackle over-aggregating. To keep the simplicity of our proposed method, we only explore a simple division method based on KMeans++ [3] and an attention-guiding method with cluster weighting. In Sec. 5.3.3, the center selection study primarily shows that a learnable center selection strategy can solve the trade-off between efficiency and model performance. The module ablation study demonstrates that the simple weighting method may fail to benefit backbones under certain circumstances. All these ablation results and the informative source node study in Sec. 5.3.1 indicates that it lacks further exploration of a learnable division strategy and adaptive weighting method.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Appendix I.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Theoretical proof is provided for Theorem 3.1 in Appendix A. Detailed derivation is provided for Eq. 3 in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer to Appendix D.2. Code is provided in https://github.com/sunjss/over-aggregating.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code with instructions for reproduction is provided in https://github.com/sunjss/over-aggregating.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed experimental setups are provided in Appendix D.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Please refer to Tab. 2-Tab. 6

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Appendix D.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to Appendix H.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Please refer to Appendix D.1.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Please refer to https://github.com/sunjss/over-aggregating.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.