A Complexity Analysis of STA

In this subsection, we analyze the time complexity of SubTree Attention (STA). STA has two key components: the feature map and the HopAggregation function. Both of these components offer a wide range of potential options for consideration. Different options will affect the time complexity of STA. In the following analysis, we will adopt the configuration used by STAGNN, i.e., we choose $\phi(x) = elu(x) + 1$ as the feature map and use GPR-like aggregation as the HopAggregation function.

The computation of STA can be seen as an aggregation of $\{\text{STA}_i\}_{i\in[\![1,K]\!]}$, which refer to the attention-based aggregation of each level of the rooted subtree. Therefore, we can start by analyzing the time complexity of STA_k.

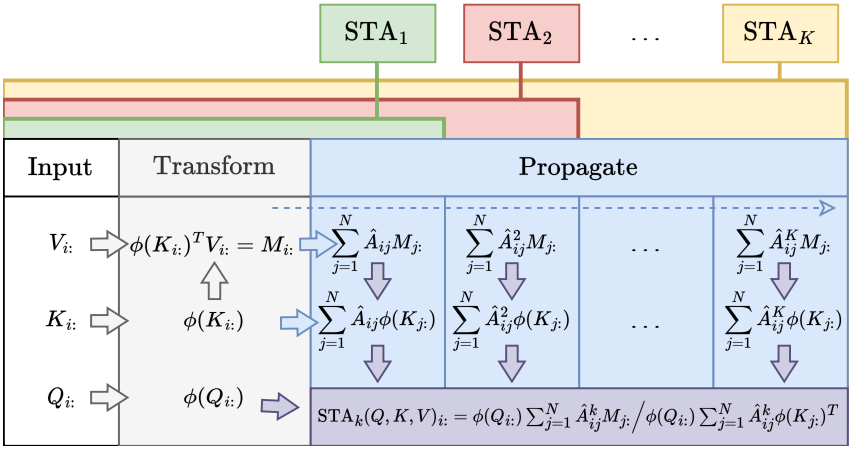

Figure 6: Efficient algorithm of SubTree Attention

The calculation of STA_k can be divided into three steps. In the first step, we compute $\phi(\mathbf{K}_{i:})$ and $\phi(\mathbf{K}_{i:})^T\mathbf{V}_{i:}$ for each node. The time complexity of this step depends on the feature map. In our model, we chose $\phi(x) = elu(x) + 1$ as the feature map. Thus, the time complexity of computing $\phi(\mathbf{K}_{i:})$ is $\mathcal{O}(Nd_k)$. We also need to compute $\phi(\mathbf{K}_{i:})^T\mathbf{V}_{i:}$ for each node, the time complexity of this part is $\mathcal{O}(Nd_kd_v)$. Therefore, the overall time complexity of the first step is $\mathcal{O}(Nd_k + Nd_kd_v)$.

In the second step, we let $\phi(\mathbf{K}_{i:})$ and $\phi(\mathbf{K}_{i:})^T\mathbf{V}_{i:}$ propagate on the graph. For STA_k, we need to propagate k times. The time complexity of propagating $\phi(\mathbf{K}_{i:})$ once is $\mathcal{O}(d_k)$, and the time complexity of $\phi(\mathbf{K}_{i:})^T\mathbf{V}_{i:}$ propagating once is $\mathcal{O}(d_kd_v)$. The message propagation occurs on each edge. Considering that there are in total $|\mathcal{E}|$ edges and k times propagation, the overall time complexity of this step is $\mathcal{O}(k|\mathcal{E}|d_k + k|\mathcal{E}|d_kd_v)$.

In the third step, we use the information $\sum_{j=1}^{N} \hat{\mathbf{A}}_{ij}^k \phi(\mathbf{K}_{j:})^T\mathbf{V}_{j:}$ and $\sum_{j=1}^{N} \hat{\mathbf{A}}_{ij}^k \phi(\mathbf{K}_{j:})^T$ aggregated by each node, along with the node's own query $\phi(\mathbf{Q}_{i:})$, to complete the computation of STA. For each node, we need to calculate $\phi(\mathbf{Q}_{i:}) \sum_{j=1}^{N} \hat{\mathbf{A}}_{ij}^k \phi(\mathbf{K}_{j:})^T\mathbf{V}_{j:}$, the time complexity of this part is $\mathcal{O}(Nd_kd_v)$. At the same time, for each node, we need to calculate $\phi(\mathbf{Q}_{i:}) \sum_{j=1}^{N} \hat{\mathbf{A}}_{ij}^k \phi(\mathbf{K}_{j:})^T$, the time complexity of this part is $\mathcal{O}(Nd_k)$. So the total time complexity of this step is $\mathcal{O}(Nd_k + Nd_kd_v)$.

In summary, the total time complexity of STA_k is $\mathcal{O}(2Nd_k + 2Nd_kd_v + k|\mathcal{E}|d_k + k|\mathcal{E}|d_kd_v)$.

Next, we analyze the time complexity of STA when the height of the rooted subtree is K. It should be noted that $\{\text{STA}_i\}_{i\in[\![1,K]\!]}$ can be viewed as a nested process, calculated one after another. Therefore, the first two steps of the above-mentioned calculation of STA_k do not need to be repeated. We only need to complete the full calculation of STA_K and perform the third step mentioned above K times. Therefore, the time complexity of calculating STA is $\mathcal{O}((K+1)Nd_k + (K+1)Nd_kd_v + K|\mathcal{E}|d_k + K|\mathcal{E}|d_kd_v)$. In general, we can think of the time complexity of STA as $\mathcal{O}(K|\mathcal{E}|d_kd_v)$.

B Proof for Theorem 1

B.1 Proof for Equation 10

Let $\hat{\mathbf{A}}$ denote the random walk matrix of a connected and non-bipartite graph, and let $\mathbf{A}_{\text{sym}}$ denote the symmetric normalized adjacency matrix. Let $1 = \lambda_1 \geq \cdots \geq \lambda_N$ be the eigenvalues of $\hat{\mathbf{A}}$, which are also the eigenvalues of $\mathbf{A}_{\text{sym}}$ [26]. Let $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}$ be the corresponding orthonormal eigenvectors ($\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}$ here are column vectors). Let $\boldsymbol{\pi}_j = \frac{d(j)}{\sum_{i=1}^{N} d(i)}$ and $d(i)$ denotes the degree of the i^{th} node. $\hat{\lambda} = 1 - \max\{\lambda_2, |\lambda_n|\}$ denotes the corresponding spectral gap, and let $\mathbf{D}$ be the diagonal degree matrix. $\overrightarrow{\mathbf{1}}$ denotes an all-ones column vector.

In this subsection, we prove the following results:

$$\forall i, j \in [\![1, N]\!]^2,\ \forall \epsilon > 0,\ \exists K_0 \in \mathbb{N},\ \forall k > K_0,\ |\hat{\mathbf{A}}_{ij}^k - \boldsymbol{\pi}_i| \leq \epsilon$$

And for a given ϵ, the smallest K_0 that satisfies the condition above is at most $\mathcal{O}\left(\frac{\log \frac{N}{\epsilon}}{1-\max\{\lambda_2, |\lambda_n|\}}\right)$.

We begin by considering an arbitrary distribution $\mathbf{p_i} \in \mathbb{R}^N$, which is a column vector and $\|\mathbf{p_i}\|_2 = 1$. Notice that $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}$ form an orthonormal basis, we can rewrite $\mathbf{D}^{-\frac{1}{2}}\mathbf{p_i}$ as:

$$\mathbf{D}^{-\frac{1}{2}}\mathbf{p_i} = \sum_{i=1}^{N} c_i \mathbf{v_i} \tag{13}$$

We next consider the new distribution obtained when $\mathbf{p_i}$ undergoes k-step random walk. Notice that $\hat{\mathbf{A}} = \mathbf{A}\mathbf{D}^{-1} = \mathbf{D}^{\frac{1}{2}}\left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\right)\mathbf{D}^{-\frac{1}{2}} = \mathbf{D}^{\frac{1}{2}}\mathbf{A}_{\text{sym}}\mathbf{D}^{-\frac{1}{2}}$. Thus we have:

$$\begin{aligned}
\hat{\mathbf{A}}^k\mathbf{p_i} &= \left(\mathbf{D}^{\frac{1}{2}}\mathbf{A}_{\text{sym}}\mathbf{D}^{-\frac{1}{2}}\right)^k \mathbf{p_i} \\
&= \mathbf{D}^{\frac{1}{2}}\mathbf{A}_{\text{sym}}^k\mathbf{D}^{-\frac{1}{2}}\mathbf{p_i} \\
&= \mathbf{D}^{\frac{1}{2}}\mathbf{A}_{\text{sym}}^k \sum_{i=1}^{N} c_i \mathbf{v_i} \\
&= \sum_{i=1}^{N} c_i \mathbf{D}^{\frac{1}{2}}\mathbf{A}_{\text{sym}}^k \mathbf{v_i} \\
&= c_1\mathbf{D}^{\frac{1}{2}}\mathbf{A}_{\text{sym}}^k\mathbf{v_1} + \sum_{i=2}^{N} c_i \mathbf{D}^{\frac{1}{2}}\mathbf{A}_{\text{sym}}^k \mathbf{v_i}
\end{aligned} \tag{14}$$

We now consider $c_1\mathbf{D}^{\frac{1}{2}}\mathbf{A}_{\text{sym}}^k\mathbf{v_1}$. As we know that $\mathbf{D}^{\frac{1}{2}}\overrightarrow{\mathbf{1}}$ is an eigenvector of $\mathbf{A}_{\text{sym}}$ with eigenvalue 1. We then have:

$$\mathbf{v_1} = \frac{\mathbf{D}^{\frac{1}{2}}\overrightarrow{\mathbf{1}}}{\|\mathbf{D}^{\frac{1}{2}}\overrightarrow{\mathbf{1}}\|_2} \tag{15}$$

Notice that $\|\mathbf{D}^{\frac{1}{2}}\overrightarrow{\mathbf{1}}\|_2 = \sqrt{\sum_{i=1}^{N} d(i)}$. Using the fact that $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}$ form an orthonormal basis and $\mathbf{D}^{-\frac{1}{2}}\mathbf{p_i} = \sum_{i=1}^{N} c_i \mathbf{v_i}$, we then have:

$$\begin{aligned} c_1 &= \left(\mathbf{D}^{-\frac{1}{2}}\mathbf{p_i}\right)^T v_1 \\ &= \mathbf{p_i}^T \mathbf{D}^{-\frac{1}{2}} \frac{\mathbf{D}^{\frac{1}{2}}\overrightarrow{\mathbf{1}}}{\|\mathbf{D}^{\frac{1}{2}}\overrightarrow{\mathbf{1}}\|_2} \\ &= \frac{\mathbf{p_i}^T \overrightarrow{\mathbf{1}}}{\|\mathbf{D}^{\frac{1}{2}}\overrightarrow{\mathbf{1}}\|_2} \\ &= \frac{1}{\|\mathbf{D}^{\frac{1}{2}}\overrightarrow{\mathbf{1}}\|_2} \\ &= \frac{1}{\sqrt{\sum_{i=1}^{N} d(i)}} \end{aligned} \tag{16}$$

Notice that $\mathbf{A}_{\text{sym}}^k \mathbf{v_1} = \lambda_1^k \mathbf{v_1}$ and $\lambda_1 = 1$. Thus we have:

$$\begin{aligned} c_1 \mathbf{D}^{\frac{1}{2}} \mathbf{A}_{\text{sym}}^k \mathbf{v_1} &= c_1 \mathbf{D}^{\frac{1}{2}} \lambda_1^k \mathbf{v_1} \\ &= c_1 \mathbf{D}^{\frac{1}{2}} \mathbf{v_1} \\ &= \frac{1}{\sqrt{\sum_{i=1}^{N} d(i)}} \mathbf{D}^{\frac{1}{2}} \frac{\mathbf{D}^{\frac{1}{2}}\overrightarrow{\mathbf{1}}}{\|\mathbf{D}^{\frac{1}{2}}\overrightarrow{\mathbf{1}}\|_2} \\ &= \frac{\mathbf{D}\overrightarrow{\mathbf{1}}}{\sum_{i=1}^{N} d(i)} \\ &= \boldsymbol{\pi} \end{aligned} \tag{17}$$

Considering Equation 14 and Equation 17, we have:

$$\hat{\mathbf{A}}^k \mathbf{p_i} = \boldsymbol{\pi} + \sum_{i=2}^{N} c_i \mathbf{D}^{\frac{1}{2}} \mathbf{A}_{\text{sym}}^k \mathbf{v_i} \tag{18}$$

and immediately:

$$\begin{aligned} \|\hat{\mathbf{A}}^k \mathbf{p_i} - \boldsymbol{\pi}\|_2^2 &= \|\sum_{i=2}^{N} c_i \mathbf{D}^{\frac{1}{2}} \mathbf{A}_{\text{sym}}^k \mathbf{v_i}\|_2^2 \\ &= \|\mathbf{D}^{\frac{1}{2}} \sum_{i=2}^{N} c_i \mathbf{A}_{\text{sym}}^k \mathbf{v_i}\|_2^2 \\ &\leq |||\mathbf{D}^{\frac{1}{2}}|||_p^2 \, \|\sum_{i=2}^{N} c_i \mathbf{A}_{\text{sym}}^k \mathbf{v_i}\|_2^2 \end{aligned} \tag{19}$$

where $|||\mathbf{D}^{\frac{1}{2}}|||_p = \sup_{x \in \mathbb{R}^N} \frac{\|\mathbf{D}^{\frac{1}{2}} x\|_2}{\|x\|_2} = \sqrt{d_{\max}}$. Thus we have:

$$\|\hat{\mathbf{A}}^k \mathbf{p_i} - \boldsymbol{\pi}\|_2^2 \leq d_{\max} \, \|\sum_{i=2}^{N} c_i \mathbf{A}_{\text{sym}}^k \mathbf{v_i}\|_2^2 \tag{20}$$

And using the fact that $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}$ are orthonormal and $1 - \hat{\lambda} = \max\{\lambda_2, |\lambda_N|\} = \max\{|\lambda_2|, |\lambda_3|, \ldots, |\lambda_N|\}$, we then have:

$$\begin{aligned}
\|\hat{\mathbf{A}}^k \mathbf{p_i} - \boldsymbol{\pi}\|_2^2 &\leq d_{\max} \|\sum_{i=2}^{N} c_i \lambda_i^k \mathbf{v_i}\|_2^2 \\
&= d_{\max} \sum_{i=2}^{N} c_i^2 \lambda_i^{2k} \\
&\leq d_{\max} (1-\hat{\lambda})^{2k} \sum_{i=2}^{N} c_i^2 \\
&= d_{\max} (1-\hat{\lambda})^{2k} \|\mathbf{D}^{-\frac{1}{2}} \mathbf{p_i}\|_2^2
\end{aligned} \tag{21}$$

Notice that $\|\mathbf{D}^{-\frac{1}{2}} \mathbf{p_i}\|_2^2 \leq |||\mathbf{D}^{-\frac{1}{2}}|||_p^2 \, \|\mathbf{p_i}\|_2^2 = \frac{1}{d_{\min}}$. Therefore:

$$\|\hat{\mathbf{A}}^k \mathbf{p_i} - \boldsymbol{\pi}\|_2^2 \leq \frac{d_{\max}}{d_{\min}} (1-\hat{\lambda})^{2k} \tag{22}$$

and immediately:

$$\begin{aligned}
\|\hat{\mathbf{A}}^k \mathbf{p_i} - \boldsymbol{\pi}\|_2 &\leq \frac{d_{\max}}{d_{\min}} (1-\hat{\lambda})^{k} \\
&\leq \sqrt{N} (1-\hat{\lambda})^{k} \\
&\leq \sqrt{N} e^{-k\hat{\lambda}}
\end{aligned} \tag{23}$$

Using Cauchy–Schwarz, we then have:

$$\|\hat{\mathbf{A}}^k \mathbf{p_i} - \boldsymbol{\pi}\|_1 \leq \sqrt{N} \|\hat{\mathbf{A}}^k \mathbf{p_i} - \boldsymbol{\pi}\|_2 \leq N e^{-k\hat{\lambda}} \tag{24}$$

In conclusion, given an arbitrarily small positive number ϵ, for all k_0 greater than or equal to $\frac{1}{\hat{\lambda}} \log \frac{N}{\epsilon}$, the L1 norm of the difference between $\hat{\mathbf{A}}^k \mathbf{p_i}$ and the vector $\boldsymbol{\pi}$ is less than or equal to ϵ. This result establishes that $\frac{1}{\hat{\lambda}} \log \frac{N}{\epsilon}$ indeed serves as an upper bound.

Notice that the vector $\mathbf{p_i}$ is an arbitrary distribution. Thus, we may consider $\mathbf{p_i}$ to be one of the i^{th} unit basis vector in the N-dimensional space: $\{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_N}\}$, where each vector has only one element equal to 1 (the i^{th} element) and all other elements equal to 0. Thus we have $\|\hat{\mathbf{A}}^k \mathbf{p_j} - \boldsymbol{\pi}\|_1 = \sum_{i=1}^{N} |\hat{\mathbf{A}}_{ij}^k - \boldsymbol{\pi}_i|$. Then given $\epsilon > 0$, we have that:

$$\forall j \in [\![1, N]\!], \; \forall k \geq \frac{1}{\hat{\lambda}} \log \frac{N}{\epsilon}, \; \sum_{i=1}^{N} |\hat{\mathbf{A}}_{ij}^k - \boldsymbol{\pi}_i| \leq \epsilon \tag{25}$$

which demonstrates immediately the first part of Theorem 1:

$$\forall i, j \in [\![1, N]\!]^2, \; \forall k \geq \frac{1}{\hat{\lambda}} \log \frac{N}{\epsilon}, \; |\hat{\mathbf{A}}_{ij}^k - \boldsymbol{\pi}_i| \leq \epsilon \tag{26}$$

B.2 Proof for Equation 11

In this subsection, we prove the following results: if $\mathbf{V}$ is computed by $\mathbf{V} = \sigma(\mathbf{X}\mathbf{W}_V)$ where σ is a non-negative activation function, then:

$$\forall i, j \in [\![1, N]\!]^2, \; \forall \eta \in]0, 1[, \; \exists K_1 \in \mathbb{N}, \; \forall k > K_1, \; \frac{1-\eta}{1+\eta} \leq \frac{\text{STA}_k(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{ij}}{\text{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{ij}} \leq \frac{1+\eta}{1-\eta}$$

holds true when none of the denominators is equal to zero. And for a given η, the smallest K_1 that satisfies the condition shown in Equation 11 is at most $\mathcal{O}\left(\frac{\log \frac{N}{\eta}}{1-\max\{\lambda_2, |\lambda_n|\}}\right)$.

This result can, indeed, be viewed as a straightforward corollary of Equation (10). The crucial prerequisite is that all elements of the vector $\mathbf{V}$ must be positive. Importantly, there are no specific

requirements imposed on the non-negative activation function σ, meaning it can be any function that ensures non-negativity.

STA_k and SA are defined as follows:

$$\begin{aligned}\text{STA}_k(\mathbf{Q},\mathbf{K},\mathbf{V})_{i:} &= \frac{\sum_{j=1}^{N} \hat{\mathbf{A}}_{ij}^{k}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{j:})\mathbf{V}_{j:}}{\sum_{j=1}^{N} \hat{\mathbf{A}}_{ij}^{k}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{j:})}\\ \text{SA}(\mathbf{Q},\mathbf{K},\mathbf{V})_{i:} &= \frac{\sum_{j=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{j:})\mathbf{V}_{j:}}{\sum_{j=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{j:})}\end{aligned} \tag{27}$$

Equation 27 shows their form as row vectors. Their j^{th} elements are:

$$\begin{aligned}\text{STA}_k(\mathbf{Q},\mathbf{K},\mathbf{V})_{ij} &= \frac{\sum_{t=1}^{N} \hat{\mathbf{A}}_{it}^{k}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}}{\sum_{t=1}^{N} \hat{\mathbf{A}}_{it}^{k}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})}\\ \text{SA}(\mathbf{Q},\mathbf{K},\mathbf{V})_{ij} &= \frac{\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}}{\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})}\end{aligned} \tag{28}$$

Hence, we have:

$$\frac{\text{STA}_k(\mathbf{Q},\mathbf{K},\mathbf{V})_{ij}}{\text{SA}(\mathbf{Q},\mathbf{K},\mathbf{V})_{ij}} = \frac{\sum_{t=1}^{N} \hat{\mathbf{A}}_{it}^{k}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}}{\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}} \times \frac{\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})}{\sum_{t=1}^{N} \hat{\mathbf{A}}_{it}^{k}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})} \tag{29}$$

For clarity, we proceed under the assumption that none of the denominators equal zero, which is reasonable considering the context. Let δ_{it} represent the difference between $\hat{\mathbf{A}}_{it}^{k}$ and $\boldsymbol{\pi}_i$: $\hat{\mathbf{A}}_{it}^{k} = \boldsymbol{\pi}_i + \delta_{it}$. Given $\eta \in]0,1[$, we aim to determine an upper bound of the convergence rate between STA_k and SA.

Using Equation 10, we take $\epsilon = \frac{\eta}{N^2}$ and we have immediately:

$$\forall i,t \in [\![1,N]\!]^2,\ \forall k \geq \frac{2\log\frac{N}{\eta}}{1-\hat{\lambda}},\ |\hat{\mathbf{A}}_{it}^{k} - \boldsymbol{\pi}_i| = |\delta_{it}| \leq \epsilon = \frac{\eta}{N^2} \tag{30}$$

We can rewrite Equation 29 as:

$$\frac{\text{STA}_k(\mathbf{Q},\mathbf{K},\mathbf{V})_{ij}}{\text{SA}(\mathbf{Q},\mathbf{K},\mathbf{V})_{ij}} = \frac{\sum_{t=1}^{N} (\boldsymbol{\pi}_i + \delta_{it})\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}}{\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}} \times \frac{\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})}{\sum_{t=1}^{N} (\boldsymbol{\pi}_i + \delta_{it})\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})} \tag{31}$$

Assuming that $k \geq \frac{2\log\frac{N}{\eta}}{1-\hat{\lambda}}$. Considering the fraction $\frac{\sum_{t=1}^{N} \delta_{it}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}}{\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}}$ in the first part of Equation 31. Using Equation 30 and the fact that $\text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})$ and $\mathbf{V}_{tj}$ are all positive, we have:

$$\begin{aligned}|\sum_{t=1}^{N} \delta_{it}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}| &= \sum_{t=1}^{N} |\delta_{it}|\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}\\ &\leq \sum_{t=1}^{N} \frac{\eta}{N^2}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}\end{aligned} \tag{32}$$

Notice that $\forall t \in [\![1,N]\!],\ \boldsymbol{\pi}_i \geq \frac{1}{N^2}$. Hence, we have:

$$|\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}| \geq \sum_{t=1}^{N} \frac{1}{N^2}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj} \tag{33}$$

Therefore:

$$\left|\frac{\sum_{t=1}^{N} \delta_{it}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}}{\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}}\right| = \frac{|\sum_{t=1}^{N} \delta_{it}\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}|}{|\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}|} \leq \eta < 1 \tag{34}$$

Thus we have

$$1-\eta \leq \frac{\sum_{t=1}^{N} (\boldsymbol{\pi}_i + \delta_{it})\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}}{\sum_{t=1}^{N} \boldsymbol{\pi}_i\ \text{sim}(\mathbf{Q}_{i:},\mathbf{K}_{t:})\mathbf{V}_{tj}} \leq 1+\eta \tag{35}$$

Table 3: Statistics on datasets

Dataset	Context	# Nodes	# Edges	# Features	# Classes
Cora	Citation	2,708	5,429	1,433	7
Citeseer	Citation	3,327	4,732	3,703	6
Deezer	Social Connection	28,281	92,752	31,241	2
Actor	Co-occurrence	7,600	29,926	931	5
Pubmed	Citation	19,717	44,324	500	3
CoraFull	Citation	19,793	126,842	8,710	70
Computer	Co-purchasing	13,752	491,722	767	10
Photo	Co-purchasing	7,650	238,163	745	8
CS	Co-authorship	18,333	163,788	6,805	15
Physics	Co-authorship	34,493	495,924	8,415	5

Considering the second part $\frac{\sum_{t=1}^{N} \boldsymbol{\pi}_i \ \mathrm{sim}(\mathbf{Q}_{i:}, \mathbf{K}_{t:})}{\sum_{t=1}^{N} (\boldsymbol{\pi}_i + \delta_{it}) \ \mathrm{sim}(\mathbf{Q}_{i:}, \mathbf{K}_{t:})}$ of Equation 31. Utilizing the same line of reasoning, we can obtain:

$$\left| \frac{\sum_{t=1}^{N} \delta_{it} \ \mathrm{sim}(\mathbf{Q}_{i:}, \mathbf{K}_{t:})}{\sum_{t=1}^{N} \boldsymbol{\pi}_i \ \mathrm{sim}(\mathbf{Q}_{i:}, \mathbf{K}_{t:})} \right| \leq \eta < 1 \tag{36}$$

and

$$\frac{1}{1+\eta} \leq \frac{\sum_{t=1}^{N} \boldsymbol{\pi}_i \ \mathrm{sim}(\mathbf{Q}_{i:}, \mathbf{K}_{t:})}{\sum_{t=1}^{N} (\boldsymbol{\pi}_i + \delta_{it}) \ \mathrm{sim}(\mathbf{Q}_{i:}, \mathbf{K}_{t:})} \leq \frac{1}{1-\eta} \tag{37}$$

Considering Equation 35, Equation 37 and Equation 31, we finally prove that:

$$\frac{1-\eta}{1+\eta} \leq \frac{\mathrm{STA}_k(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{ij}}{\mathrm{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{ij}} \leq \frac{1+\eta}{1-\eta} \tag{38}$$

which proves the second part of Theorem 1.

C Dataset Information

In this section, we present the datasets used in our experiments. These different types of data provide a robust platform to evaluate the performance of our methods.

The detailed information for each dataset is presented in Table 3. These datasets are drawn from the areas of citation networks, co-purchasing networks, co-authorship networks, and social networks: • *Citation Networks*: The citation networks datasets include Cora, Citeseer, Pubmed, and CoraFull. Nodes in these networks correspond to scientific publications, while the edges represent citations between these documents. In addition to the topological structure, each node carries a binary attribute vector, encoding the presence or absence of specific words from a pre-determined dictionary. The dimensionality of these attribute vectors varies from 1,433 in Cora to 8,710 in CoraFull. Moreover, each document node is associated with a unique class label, signifying the document's overarching scientific discipline. • *Co-authorship Networks*: We utilize the CoauthorCSDataset and CoauthorPhysicsDataset that capture co-authorship relationships in Computer Science and Physics domains, respectively. Nodes represent individual authors and edges encode co-authorship relations, thus creating an undirected graph. • *Co-purchasing Networks*: We utilize the AmazonCoBuyComputerDataset and AmazonCoBuyPhotoDataset, derived from Amazon's co-purchasing network. Nodes denote products and edges symbolize frequent co-purchase incidents. Moreover, the nodes can carry diverse product-specific information. • *Social Networks*: The Deezer-Europe dataset is a dataset representing a social network of Deezer users collected via the public API in March 2020. The nodes in this network symbolize Deezer users hailing from various European countries, while the edges represent reciprocal follower relationships between these users. The features of each node are derived from the preferences of the users, specifically, the artists they have expressed an interest in. The task associated with this graph involves binary node classification, wherein the objective is to predict the user's gender. • *Co-occurrence Networks*: We utilize the Actor dataset, a type of co-occurrence network based on the Microsoft Academic Graph. Nodes represent actors, and an edge signifies their co-appearance on the same Wikipedia page.

D Implementation Details

Positional Encoding We use Laplacian positional encoding to capture the structural information. As positional encoding is not the focus of our work, we use a simple approach to combine positional encoding with the original features of the nodes, which is also applied by [6]. Formally, we first calculate the eigenvectors corresponding to the smallest m eigenvalues of the Laplace matrix to construct the matrix $\mathbf{P} \in \mathbb{R}^{n \times m}$. Then we take $\mathbf{X}' = [\mathbf{X}, \mathbf{P}]$ as the new input, where $[\,]$ denotes row-wise concatenation. For all the datasets, we set $m = 3$.

D.1 Node Classification

Training Details We choose two recent studies [43, 6] and we adhere to their experimental configurations. The metrics for the baselines are also derived from these works [43, 6]. For Cora, Citeseer, Deezer and Actor, we apply the same random splits with train/valid/test ratios of 50%/25%/25% as [43]. We conduct 5 runs with different splits and take the mean accuracy and standard deviation for comparison. For Pubmed, Corafull, Computer, Photo, CS and Physics, we apply the same random splits with train/valid/test ratios of 60%/20%/20% as [6]. We conduct 10 runs with different splits and take the mean accuracy and standard deviation for comparison. Specifically, we utilize the ROC-AUC measure for binary classification on the Deezer dataset. For other datasets containing more than two classes, we opt for Accuracy as the metric. We employ the Adam optimizer for gradient-based optimization. The training procedure can at most repeat until a given budget of 3000 epochs and we set the patience of early stop to 200 epochs. We report the test accuracy of the epoch which has the highest accuracy on the validation set.

Hyperparameters For the model configuration of STAGNN, we fix the number of hidden channels at 64. We use grid search for hyper-parameter settings. The learning rate is searched within {0.001,0.01}, dropout probability searched within {0.0,0.2,0.4,0.6}, weight decay searched within {0.0001,0.0005,0.001,0.005}, height of the rooted subtree K searched within {3,5,10}, number of attention heads searched within {1,2,4,6,8}. The best hyper-parameters are provided in supplementary materials.

D.2 Study on the Necessity of SubTree Attention in the Presence of Global Attention

In this experiment, we extend STAGNN by replacing the STA module with global attention enhanced by 0, 1, 2, or 3 hop/hops subtree attention. We now present a detailed mathematical description of the experimental configurations. Formally, we compare the performance of the STAGNN-based model equipped with four different attention strategies: Global Attn Only, 1-hop STA + GA, 2-hops STA + GA and 3-hops STA + GA on six datasets: Pubmed, Corafull, Computer, Photo, CS and Physics, with the same experiment setting described in subsection D.1.

First, we calculate keys, queries and values.

$$\mathbf{Q} = \mathbf{H}\mathbf{W}_Q,\ \mathbf{K} = \mathbf{H}\mathbf{W}_K,\ \mathbf{V} = \mathbf{H}\mathbf{W}_V,\ \mathbf{H} = \text{MLP}(\mathbf{X}) \tag{39}$$

Next, the output of the four different models (equipped with global attention enhanced by subtree attention of different heights) can be described as:

• *Global Attn Only*:

$$\mathbf{O} = \text{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \tag{40}$$

• *1-hop STA + GA*:

$$\mathbf{O} = \alpha_T \text{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \sum_{k=0}^{1} \alpha_k \text{STA}_k(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \tag{41}$$

• *2-hops STA + GA*:

$$\mathbf{O} = \alpha_T \text{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \sum_{k=0}^{2} \alpha_k \text{STA}_k(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \tag{42}$$

• *3-hops STA + GA*:

$$\mathbf{O} = \alpha_T \text{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \sum_{k=0}^{3} \alpha_k \text{STA}_k(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \tag{43}$$

α_T here represents the coefficient of teleportation, because we can regard the global attention enhanced by subtree attention here as the random walk with teleportation. The only difference between these models is that they use subtree attention of different heights as an auxiliary to global attention. As shown in Table 2, we can observe that *2-hops STA + GA* and *3-hops STA + GA* outperform *Global Attn Only* by a large margin.

D.3 Study on HopAggregation Methods

In this experiment, we investigate different choices of the HopAggregation functions within the STA module. We compare GPR-like aggregation with sum, concat [18], and attention-based readout [6]. We now present a detailed mathematical description of the experimental configurations. Formally, we compare the performance of the following four models: STAGNN-GPR (origin STAGNN), STAGNN-SUM, STAGNN-CONCAT and STAGNN-ATTN on four datasets: Cora, Citeseer, Deezer-Europe and Actor, with the same experiment setting described in subsection D.1.

First, we calculate keys, queries and values.

$$\mathbf{Q} = \mathbf{H}\mathbf{W}_Q,\ \mathbf{K} = \mathbf{H}\mathbf{W}_K,\ \mathbf{V} = \mathbf{H}\mathbf{W}_V,\ \mathbf{H} = \text{MLP}(\mathbf{X}) \tag{44}$$

Next, the output of the four different models (STAGNN with different HopAggregation methods) can be described as:

• *STAGNN-GPR (origin STAGNN)*:

$$\mathbf{O} = \sum_{k=0}^{K} \alpha_k \text{STA}_k(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \tag{45}$$

• *STAGNN-SUM*:

$$\mathbf{O} = \sum_{k=0}^{K} \text{STA}_k(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \tag{46}$$

• *STAGNN-CONCAT*:

$$\mathbf{O} = [\text{STA}_0(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \text{STA}_1(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \ldots, \text{STA}_K(\mathbf{Q}, \mathbf{K}, \mathbf{V})]\, \mathbf{W}_O \tag{47}$$

where $\mathbf{W}_O$ is a linear projection matrix.

• *STAGNN-ATTN*:

$$\begin{aligned} \mathbf{O} &= \text{STA}_0(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \sum_{k=1}^{K} \beta_k \text{STA}_k(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \\ \beta_k &= \frac{\exp\left([\text{STA}_0(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \text{STA}_k(\mathbf{Q}, \mathbf{K}, \mathbf{V})]\, \mathbf{W}_a^\top\right)}{\sum_{i=1}^{K} \exp\left([\text{STA}_0(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \text{STA}_i(\mathbf{Q}, \mathbf{K}, \mathbf{V})]\, \mathbf{W}_a^\top\right)} \end{aligned} \tag{48}$$

where $\mathbf{W}_a$ is a linear projection matrix and $[\,]$ denotes row-wise concatenation.

E More Visualizations of GPR Weights

We conduct more visualizations of the GPR weights on Cora and Actor, with heights K of the rooted subtrees ranging from 3 to 75. The results are shown in Figure 7.

In the case of Cora, we observe that as the depth K of the rooted subtree increases, STA keeps increasing the GPR weights of the local neighborhood in order to preserve the local information from being covered up by the global information.

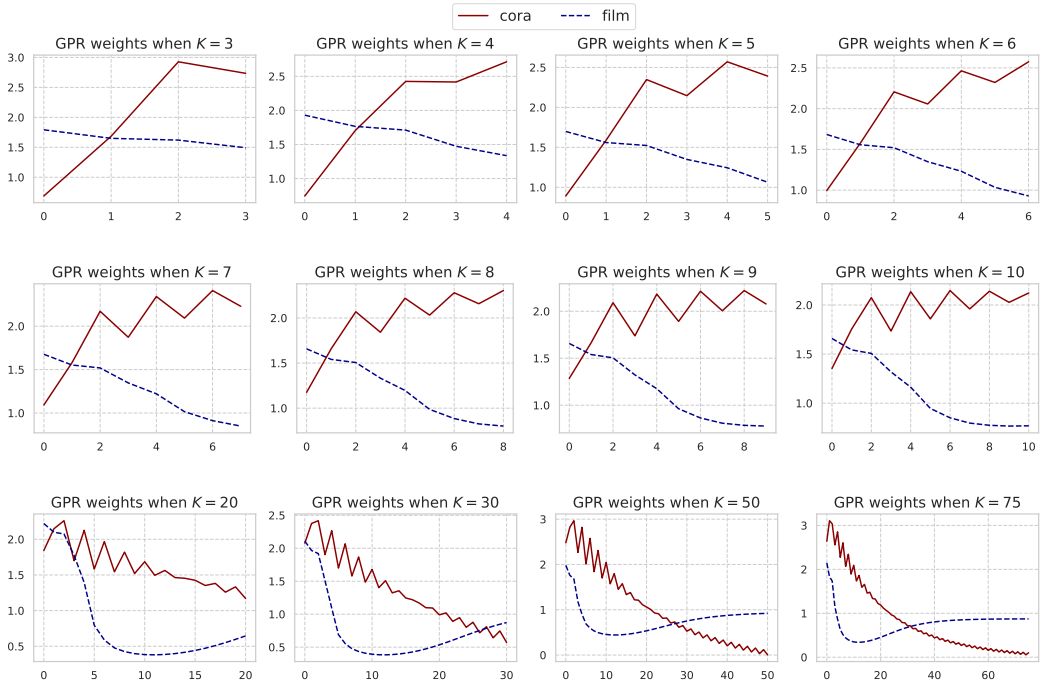

Figure 7: GPR weights of STAGNN when the heights K of the subtree ranging from 3 to 75.

F Further discussion of the Gate Mechanism within the Mixture of Attention Heads

In this subsection, we conduct an ablation study of the gate mechanism within the mixture of attention heads. The sub-tree attention module with multiple attention heads is defined as follows:

• *MSTA w/ gate vector $\boldsymbol{g}_k$, w/ softmax (origin STAGNN)*:

$$\begin{aligned}
&\mathrm{MSTA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{AGGR}\left(\{\mathrm{MSTA}_k(\mathbf{Q},\mathbf{K},\mathbf{V}) \mid k \in [\![0,K]\!]\}\right)\\
&\mathrm{MSTA}_k(\mathbf{Q},\mathbf{K},\mathbf{V}) = \left[\mathrm{head}_k^1,\ldots,\mathrm{head}_k^H\right]\mathbf{W}_O \quad \forall k \in [\![1,K]\!], \quad \mathrm{MSTA}_0(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathbf{V} \qquad (49)\\
&\mathrm{head}_k^h = \hat{g}_k^h\ \mathrm{STA}_k(\mathbf{Q^h},\mathbf{K^h},\mathbf{V^h}) \quad \forall h \in [\![1,H]\!], \quad \hat{\boldsymbol{g}}_k = \mathrm{softmax}(\boldsymbol{g}_k)
\end{aligned}$$

The hop-wise gate vector here $\boldsymbol{g}_k \in \mathbb{R}^H$ is an H-dimensional vector and g_k^h is its h^{th} element. Compared to STA with a single attention head, we introduce in total $H \times K$ additional learnable parameters: $\{\boldsymbol{g}_i\}_{i\in[\![1,K]\!]}$.

For comparison, we consider two variants.

• *MSTA w/ gate vector $\boldsymbol{g}_k$, w/o softmax*:

$$\begin{aligned}
&\mathrm{MSTA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{AGGR}\left(\{\mathrm{MSTA}_k(\mathbf{Q},\mathbf{K},\mathbf{V}) \mid k \in [\![0,K]\!]\}\right)\\
&\mathrm{MSTA}_k(\mathbf{Q},\mathbf{K},\mathbf{V}) = \left[\mathrm{head}_k^1,\ldots,\mathrm{head}_k^H\right]\mathbf{W}_O \quad \forall k \in [\![1,K]\!], \quad \mathrm{MSTA}_0(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathbf{V} \qquad (50)\\
&\mathrm{head}_k^h = g_k^h\ \mathrm{STA}_k(\mathbf{Q^h},\mathbf{K^h},\mathbf{V^h}) \quad \forall h \in [\![1,H]\!]
\end{aligned}$$

• *MSTA w/o gate vector $\boldsymbol{g}_k$*:

$$\begin{aligned}
&\mathrm{MSTA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{AGGR}\left(\{\mathrm{MSTA}_k(\mathbf{Q},\mathbf{K},\mathbf{V}) \mid k \in [\![0,K]\!]\}\right)\\
&\mathrm{MSTA}_k(\mathbf{Q},\mathbf{K},\mathbf{V}) = \left[\mathrm{head}_k^1,\ldots,\mathrm{head}_k^H\right]\mathbf{W}_O \quad \forall k \in [\![1,K]\!], \quad \mathrm{MSTA}_0(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathbf{V} \qquad (51)\\
&\mathrm{head}_k^h = \mathrm{STA}_k(\mathbf{Q^h},\mathbf{K^h},\mathbf{V^h}) \quad \forall h \in [\![1,H]\!]
\end{aligned}$$

The experimental results are shown in Table 4. We find that the performance of *MSTA w/ gate vector $\boldsymbol{g}_k$, w/o softmax* and *MSTA w/o gate vector $\boldsymbol{g}_k$* are almost the same, which means that using the gate

Table 4: Ablation study of the gate mechanism within the mixture of attention heads

Method	Pubmed	CoraFull	Computer	Photo	CS	Physics
STAGNN (origin)	$\mathbf{90.46}_{\pm \mathbf{0.22}}$	$\mathbf{72.65}_{\pm \mathbf{0.36}}$	$91.72_{\pm 0.30}$	$\mathbf{95.64}_{\pm \mathbf{0.27}}$	$\mathbf{95.77}_{\pm \mathbf{0.16}}$	$\mathbf{97.09}_{\pm \mathbf{0.18}}$
w/ gate, w/o softmax	$90.37_{\pm 0.23}$	$71.62_{\pm 0.39}$	$\mathbf{91.89}_{\pm \mathbf{0.27}}$	$95.37_{\pm 0.30}$	$94.72_{\pm 0.19}$	$96.96_{\pm 0.20}$
w/o gate	$90.31_{\pm 0.25}$	$71.67_{\pm 0.36}$	$91.80_{\pm 0.28}$	$95.32_{\pm 0.28}$	$94.70_{\pm 0.18}$	$96.97_{\pm 0.18}$

vector without softmax is approximately equivalent to not using the gate vector. In fact, on closer examination, we find that without softmax, the learned gate vector would be a vector with all equal elements, which means that it is difficult for the model to learn different weights of attention heads at each hop without the help of softmax. Additionally, we observe that for most datasets, using the gating mechanism leads to improvement of the overall performance.

G Potential Impacts

Besides learning better node representations, our proposed Subtree Attention (STA) has potential impacts on various aspects of graph learning. Compared to global attention, STA can help the model to better learn the hierarchical structure of the graph. Therefore, STA can be utilized as a plug-in module for designing local-aware Transformers on graph, acting as a competitor of all the GNN-assisted Transformers. STA opens new avenues for model design by combining the message-passing scheme with fully-attentional architectures, which can significantly enhance both the computational efficiency and expressive power of fully-attentional models on graph data. Furthermore, STA bridges the gap between local and global graph attention methods. This opens up possibilities for the design and application of hierarchical attention models that can leverage both local neighborhood and global structural information from graph data.