Appendix

This supplementary document is organized as follows:

- Section A introduces the symbols utilized in this work.
- Section B analyzes the heterophily and homophily in the Visual Genome dataset.
- Section C shows more explanations of methodological details including the difference between ART and current MP modules in Section C.1.1, and the initialization of γ_u in Section C.1.2.
- Section D provides more details of experiments including implementation details in Section D.1, the details of baseline in Section D.2.

A. Notations

Table 1 summarizes the symbols used in this work. Note that the nodes and edges denote the objects and relationships in a scene graph, respectively.

Symbols	Definitions
$\mathcal{G}$=$(\mathcal{V}, \mathcal{E})$	a graph $\mathcal{G}$ with the node set $\mathcal{V}$ and the edge set $\mathcal{E}$
$\boldsymbol{x}_i$	a feature vector of the node v_i
$\boldsymbol{x}_{ij}$	a feature vector extracted from the union area between two nodes v_i and v_j
r_{ij}	the edge between two nodes v_i and v_j
$\mathcal{N}_i$	the set of neighbors (excluding v_i) of node v_i in graph $\mathcal{G}$
$\mathcal{N}_i^s$	the set of neighbors (excluding v_i) whose class is the same as v_i
$\mathcal{N}_{r_{ij}}$	the set of neighbors (excluding r_{ij}) of the edge r_{ij} in graph $\mathcal{G}$
$\mathcal{N}_{r_{ij}}^s$	the set of neighbors (excluding r_{ij}) whose class is the same as r_{ij}
$h(\mathcal{G}_\mathcal{V})$	the node homophily ratio of the graph $\mathcal{G}$
$h(\mathcal{G}_\mathcal{E})$	the edge homophily ratio of the graph $\mathcal{G}$
$\mathcal{A}_f$	a linear fusion function that obtains the normalized contextual coefficient
$\mathcal{A}_h$	a multi-modal fusion function that obtains the normalized contextual coefficient
σ	a non-linear activation function
U	the number of ART layers
$\|\cdot\|$	the cardinality operator
$\lvert\cdot\rvert$	the absolute operator

Table 1. The definitions of the major symbols.

B. Heterophily and Homophily in Visual Genome

Given a set of node classes, the homophily describes the tendency of a node to have the same class as its neighbors, and the heterophily depicts the tendency of a node to have different classes as its neighbors. Specifically, [6] proposed a metric to measure the level of homophily of nodes in a graph. The metric can be extended to SGG as follows:

$$h(\mathcal{G}_\mathcal{V}) = \frac{1}{\|\mathcal{V}\|} \sum_{i \in \mathcal{V}} \frac{\|\mathcal{N}_i^s\|}{\|\mathcal{N}_i\|}. \tag{1}$$

Low homophily corresponds to high heterophily, and vice versa. Accordingly, $h(\mathcal{G}_\mathcal{V}) \to 1$ corresponds to strong homophily, whereas $h(\mathcal{G}_\mathcal{V}) \to 0$ indicates strong heterophily. The definition could be extended to describe the homophily and heterophily of edges. Accordingly, the level of homophily of edges is defined as

$$h(\mathcal{G}_\mathcal{E}) = \frac{1}{\|\mathcal{E}\|} \sum_{r_{ij} \in \mathcal{E}} \frac{\left\|\mathcal{N}_{r_{ij}}^s\right\|}{\left\|\mathcal{N}_{r_{ij}}\right\|}. \tag{2}$$

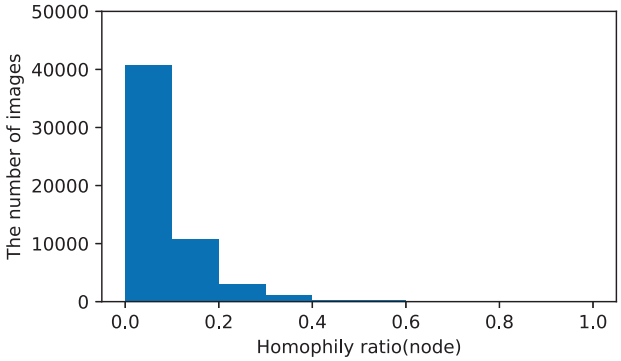

Figure 1. The number of images according to the homophily ratio of nodes $h(\mathcal{G}_{\mathcal{V}})$ in the VG dataset.

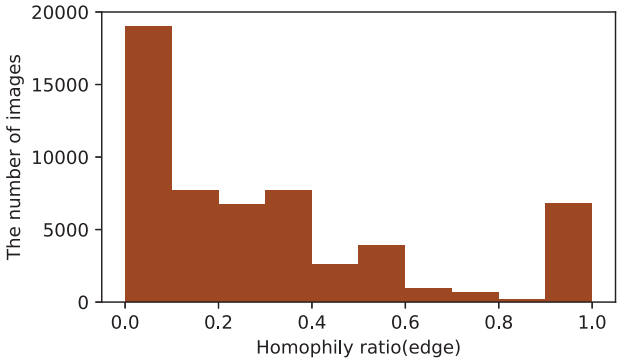

Figure 2. The number of images according to the homophily ratio of edges $h(\mathcal{G}_{\mathcal{E}})$ in the VG dataset.

As illustrated in Figure 1 and Figure 2 according to the homophily ratio of nodes and edges, respectively. The low homophily ratios validate that generating informative scene graphs requires considering heterophily. To verify the effectiveness of our HL-Net for SGG under heterophily, Table 2 and Table 3 compare the performance of HL-Net, Motifs [15], and VtransE [16] considering node homophily and edge homophily using ResNeXt-101-FPN as the backbone, respectively. As we can observe, HL-Net obtains the best performance on nearly all metrics. In particular, HL-Net significantly outperforms other methods when there is high heterophily. Thus, the high performance of HL-Net demonstrates that HL-Net has clear advantages in modeling the heterophily of SGG.

$h(\mathcal{G}_{\mathcal{V}})$	0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
Motifs [9, 15]	40.3	36.7	37.4	32.5	34.1	33.5	36.9	35.3	19.2	37.7
Vtranse [9, 16]	39.5	35.8	37.5	31.7	35.8	**39.4**	36.1	**36.7**	21.4	33.0
HL-Net	**43.8**	**40.1**	**40.4**	**33.4**	**36.4**	37.6	**37.0**	29.6	**29.3**	**33.8**

Table 2. Performance comparisons on R@100 for different $h(\mathcal{G}_{\mathcal{V}})$ in the SGCLS task on the VG dataset.

$h(\mathcal{G}_{\mathcal{E}})$	0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
Motifs [9, 15]	35.4	35.4	38.3	40.3	44.6	44.9	47.3	49.1	57.3	49.4
Vtranse [9, 16]	34.6	34.7	37.3	39.5	44.1	44.7	46.2	48.5	55.7	48.5
HL-Net	**38.4**	**39.2**	**42.0**	**43.7**	**48.5**	**48.4**	**51.0**	**53.5**	**61.7**	**52.8**

Table 3. Performance comparisons on R@100 for different $h(\mathcal{G}_{\mathcal{E}})$ in the SGCLS task on the VG dataset.

C. More Explanations of Methodological Details

This section provides more explanations of the methodological details of the heterophily learning network (HL-Net).

C.1. More explanations of the ART Module

C.1.1 The difference between ART and current MP modules

As illustrated in Figure 3, current message passing networks could be categorized into two types: pairwise-based message passing (P-MP) [7, 14] and union-based message passing (U-MP) [4, 13].

We first review the design of the P-MP [7, 10, 14]. As illustrated in Figure 3 (a), the output of the u-th layer for the representation of the node v_i could be calculated as follows:

$$\boldsymbol{x}_i^{u+1} = \boldsymbol{x}_i^u + \boldsymbol{W}_z \sigma(\sum\nolimits_{j \in \mathcal{N}_i} \mathcal{A}_f(\boldsymbol{x}_i^u, \boldsymbol{x}_j^u) \boldsymbol{W}_v \boldsymbol{x}_j^u), \tag{3}$$

Then, we explain the design of U-MP [4, 13]. U-MP utilizes the features of union areas between two nodes to calculate the correlation between the two nodes. Generally, U-MP adopts a high-order function on context modeling and a Transform

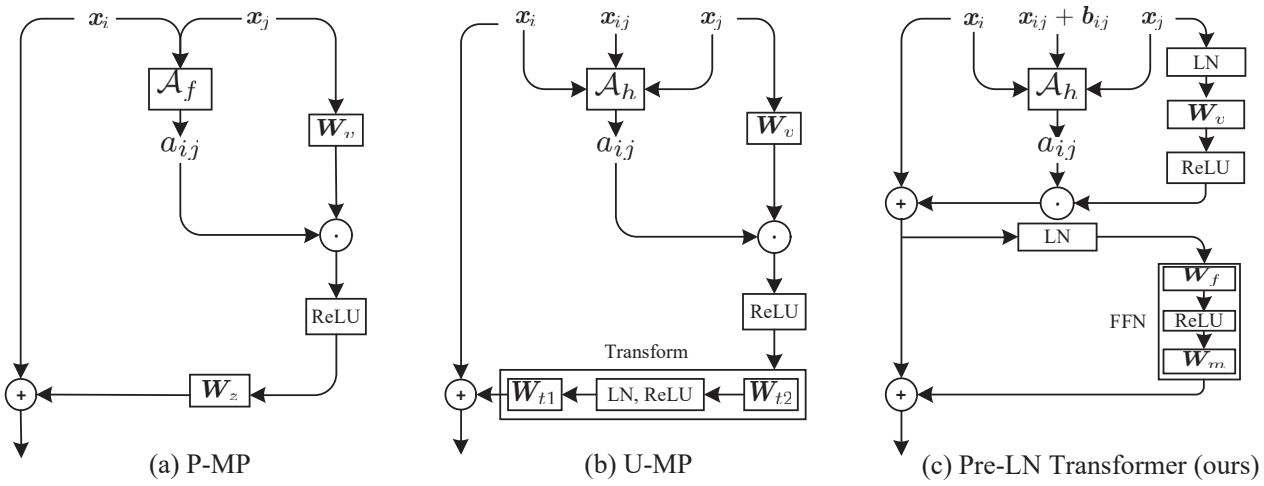

Figure 3. Architecture of three MP modules. $\odot$ and $\oplus$ represent Hadamard product and element-wise addition, respectively.

module [1] to refine the contextual information. Specifically, as shown in Figure 3 (b), the output of the u-th layer for the representation of node v_i could be obtained as follows:

$$\boldsymbol{x}_i^{u+1} = \boldsymbol{x}_i^u + \text{Transform}(\sigma(\sum\nolimits_{j\in\mathcal{N}_i} \mathcal{A}_h(\boldsymbol{x}_i^u, \boldsymbol{x}_j^u, \boldsymbol{x}_{ij})\boldsymbol{W}_v\boldsymbol{x}_j^u)), \tag{4}$$

In contrast, our Pre-LN Transformer utilizes the relative spatial feature of pair-wise node to encode their correlation and delicately arranges the layer normalization, FFN, and residual connection. As depicted in Figure 3 (c), the output of the u-th layer for the representation of v_i could be calculated as follows:

$$\boldsymbol{x}_i^{u+1} = \boldsymbol{z}_i^u + \text{FFN}(\text{LN}(\overbrace{\boldsymbol{x}_i^u + \sum\nolimits_{j\in\mathcal{N}_i} \mathcal{A}_h(\boldsymbol{x}_i^u, \boldsymbol{x}_j^u, \boldsymbol{x}_{ij} + \boldsymbol{B}_{ij})\sigma(\boldsymbol{W}_v\text{LN}(\boldsymbol{x}_j))}^{z_i^u})). \tag{5}$$

Besides, we propose a new approach to obtain the final node representation. As shown in Figure 4 (a), existing works generally refine the node representation by stacking several MP blocks and propagating the message among the layers, namely Stacked Propagation (SP). As explained in [2], SP is related to polynomial graph filtering and not suitable to handle the task under heterophily.

To address the above issue, we propose an adaptive graph filter (AGF) to adjustively aggregate the outputs of different layers. The details are depicted in Figure 4 (b). Specifically, the node representation of each layer contributes to the final output with a weight γ_u. As proven in [2], AGF could enable the model to pass relevant heterophilic information by allowing γ_u to be negative and learned in an end-to-end fashion.

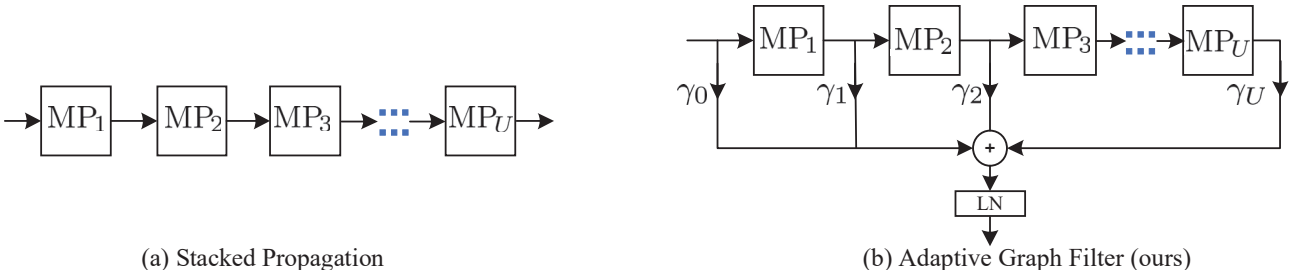

Figure 4. The refinement of the node representation.

C.1.2 The analysis of the initialization of γ_u

As mentioned in Section B, most scene graphs are under high heterophily. In order to better capture this type of high-frequency graph signal, we propose the following function based on high-pass graph filters to initialize γ_u as follows:

$$\gamma_u = \frac{(-\tau)^{u-1}}{\sum_{u=1}^{U} |(-\tau)^{u-1}|}, \tag{6}$$

We proof the Eq. (6) is a high-pass graph filter when $\alpha \in (0, 0.5]$. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_i$ be the eigenvalues of the adjacent matrix. According to [5], GNN-based models can be viewed as graph filters as follows:

$$f_{\gamma,K}(\lambda) = \sum_{k=0}^{K} \gamma_k \lambda^k \tag{7}$$

Thus, we have

$$g(\lambda) = \lim_{K\to\infty} f_{\gamma,K}(\lambda) = \sum_{k=0}^{\infty} \frac{(-\lambda\tau)^k}{1+\tau+\tau^2+\cdots+\tau^k} = \sum_{k=0}^{\infty} \frac{(1-\tau)(-\lambda\tau)^k}{1-\tau^{k+1}} \tag{8}$$

Furthermore, the derivative of $g(\lambda)$ is calculated as follows:

$$\begin{aligned} g'(\lambda) &= \sum_{k=0}^{\infty} \frac{(1-\tau)(k+1)\lambda^k(-\tau)^{k+1}}{1-\tau^{k+2}} \\ &= \sum_{c=0}^{\infty} -\underbrace{\frac{(1-\tau)(2c+1)\lambda^{2c}\tau^{2c+1}}{1-\tau^{2c+2}}}_{b_{2c}} + \underbrace{\frac{(1-\tau)(2c+2)\lambda^{2c+1}\tau^{2c+2}}{1-\tau^{2c+3}}}_{b_{2c+1}} \\ &= \sum_{c=0}^{\infty} -b_{2c} + b_{2c+1} \end{aligned} \tag{9}$$

From basic spectral analysis [11], we know that $\lambda_1 = 1$ and $|\lambda_i| < 1, \forall i \geq 2$. Therefore, we consider three cases: **(1)** $\lambda \in (-1, 0)$, **(2)** $\lambda = 0$, and **(3)** $\lambda \in (0, 1)$.

Case 1: $\lambda \in (-1, 0)$

It is obvious that $-b_{2c} + b_{2c+1} <0$, hence, we have

$$g'(\lambda) < 0. \tag{10}$$

Case 2: $\lambda = 0$

$$g(\lambda) = 1 + \sum_{k=1}^{\infty} \frac{(-\tau\lambda)^k}{1+\tau+\cdots+\tau^k} \Rightarrow g(0) = 1 \tag{11}$$

For λ_1, we have

$$\begin{aligned} g(\lambda_1) &= 1 + \sum_{k=1}^{\infty} \frac{(1-\tau)(-\tau)^k}{1-\tau^{k+1}} \\ &= 1 + \sum_{r=1}^{\infty} -\underbrace{\frac{(1-\tau)\tau^{2r-1}}{1-\tau^{2r}}}_{c_{2r-1}} + \underbrace{\frac{(1-\tau)\tau^{2r+1}}{1-\tau^{2r+1}}}_{c_{2r}} \\ &= 1 + \sum_{r=1}^{\infty} -c_{2r-1} + c_{2r} \end{aligned} \tag{12}$$

$$\frac{c_{2r-1}}{c_{2r}} = \frac{-\tau^{2r+1}}{\tau(1-\tau^{2r})} = \frac{1-\tau^{2r+1}}{\tau-\tau^{2r+1}} > 1 \ (0 < \tau < 1) \Rightarrow -c_{2r-1} + c_{2r} < 0 \Rightarrow g(1) < 1 \tag{13}$$

Similarly, we can rewrite Eq. (12) as follows:

$$\begin{aligned} g(1) &= 1 + \sum_{k=1}^{\infty} \frac{(1-\tau)(-\tau)^k}{1-\tau^{k+1}} \\ &= 1 - \tau + \tau^2 + \sum_{t=1}^{\infty} \underbrace{\frac{(1-\tau)\tau^{2t+1}}{1-\tau^{2t+1}}}_{d_{2t-1}} - \underbrace{\frac{(1-\tau)\tau^{2t+2}}{1-\tau^{2t+2}}}_{d_{2t}} \\ &= 1 - \tau + \tau^2 + \sum_{t=1}^{\infty} c_{2t-1} - c_{2t} > 0 \end{aligned} \tag{14}$$

Therefore, we have

$$0 < g(1) < 1 \Rightarrow \left| \frac{g(0)}{g(1)} \right| > 1. \tag{15}$$

Case 3: $\lambda \in (0, 1)$

$$\begin{aligned} \frac{b_{2c}}{b_{2c+1}} &= \frac{(2c+1)\left(1-\tau^{2c+3}\right)}{\lambda\tau(2c+2)\left(1-\tau^{2c+2}\right)} \\ &> \frac{2c+1}{\lambda\tau(2c+2)} = \frac{1}{\lambda\tau} - \frac{1}{\lambda\tau(2c+2)} = l(c) \end{aligned} \tag{16}$$

It is obvious that $l(c)$ is a monotonically increasing function according to c. Therefore, if $0 < \tau \leq 0.5 \Rightarrow 2\lambda\tau < 1$, we have

$$l(0) = \frac{1}{2\lambda\tau} > 1 \Rightarrow l(\gamma) > 1 \Rightarrow \frac{b_{2c}}{b_{2c+1}} > 1 \overset{(3)}{\Rightarrow} -b_{2c} + b_{2c+1} < 0, \forall c \tag{17}$$

Therefore, we have

$$g'(\lambda) < 0. \tag{18}$$

To sum it up, if $0 < \tau \leq 0.5$, for $|\lambda_i| < 1$, $\forall i \geq 2$, we have

$$\left| \frac{g(\lambda_i)}{g(\lambda_1)} \right| = \left| \frac{\lim_{k\to\infty} f_{\lambda,k}(\lambda_i)}{\lim_{k\to\infty} f_{\lambda,k}(\lambda_1)} \right| > 1 > \left| \frac{\lambda_i}{\lambda_1} \right| \tag{19}$$

Note that Eq. (19) implies that after applying the graph filter, the lowest frequency component, i.e., λ_1, no longer dominates [2]. This concludes the proof that Eq. (6) can be viewed as a high-pass graph filter.

D. The Details of Experiments

D.1. Implementation details

To facilitate a fair comparison with the majority of existing works, we utilized ResNeXt-101-FPN [3, 12] as the backbone for the OI database. We adopted both ResNeXt-101-FPN [3, 12] and VGG-16 [8] as the backbones for the VG database. During training, we froze the layers before the ROIAlign layer and optimized the model jointly, considering the object and relationship classification losses. We optimized HL-Net via Stochastic Gradient Descent (SGD) with momentum, with an initial learning rate of 10^{-3} and a batch size of 6. For the SGDET task, we only predict the relationship between proposal pairs with overlapped bounding boxes. The top-64 object proposals in each image were selected after per-class non-maximal suppression (NMS) with an IoU of 0.3. Moreover, the ratio between pairs without any relationship (background pairs) and those with relationships during training was sampled to 3:1. All experiments were performed on a Linux Machine with 48 cores, 376GB of RAM, and an NVIDIA Tesla V100 GPU with 32GB of GPU memory.

D.2. Details of the baseline

This subsection introduces the details of the baseline that is utilized in ablation studies. Specifically, the output of the u-th layer for the i-th node representation could be obtained as follows:

$$\boldsymbol{x}_i^{u+1} = \boldsymbol{x}_i^u + \boldsymbol{W}_z \sigma \left(\sum\nolimits_{j \in \mathcal{N}_i} \frac{\exp(\boldsymbol{w}_e^T \left[\boldsymbol{x}_i^u, \boldsymbol{x}_j^u\right])}{\sum_{m \in \mathcal{N}_i} \exp(\boldsymbol{w}_e^T \left[\boldsymbol{x}_i^u, \boldsymbol{x}_m^u\right])} \boldsymbol{W}_v \boldsymbol{x}_j^u \right), \tag{20}$$

The edge feature from node v_i to node v_j can be obtained as follows:

$$\boldsymbol{r}_{ij} = \boldsymbol{W}_s \boldsymbol{x}_i^{u+1} \odot \boldsymbol{W}_o \boldsymbol{x}_j^{u+1} \odot \boldsymbol{W}_u (\boldsymbol{x}_{ij} + \boldsymbol{B}_{ij}). \tag{21}$$

References

[1] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu. Global context networks. *TPAMI*, 2020. 3

[2] E. Chien, J. Peng, P. Li, and O. Milenkovic. Adaptive universal generalized pagerank graph neural network. In *ICLR*, 2021. 3, 5

[3] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 5

[4] X. Lin, C. Ding, J. Zeng, and D. Tao. Gps-net: Graph property sensing network for scene graph generation. In *CVPR*, 2020. 2

[5] H. Nt and T. Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019. 4

[6] H. Pei, B. Wei, K. Chang, Y. Lei, and B. Yang. Geom-gcn: Geometric graph convolutional networks. In *ICLR*, 2019. 1

[7] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo. Attentive relational networks for mapping images to scene graphs. In *CVPR*, 2019. 2

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[9] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang. Unbiased scene graph generation from biased training. In *CVPR*, 2020. 2

[10] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019. 2

[11] V. Ulrike. A tutorial on spectral clustering. *Statistics and computing*, 2007. 4

[12] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 5

[13] G. Yang, J. Zhang, Y. Zhang, B. Wu, and Y. Yang. Probabilistic modeling of semantic ambiguity for scene graph generation. In *CVPR*, 2021. 2

[14] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *ECCV*, 2018. 2

[15] R. Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 2

[16] H. Zhang, Z. Kyaw, S. Chang, and T. Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017. 2