p-Laplacian Adaptation for Generative Pre-trained Vision-Language Models

Haoyuan Wu* Xinyun Zhang* Peng Xu Peiyu Liao Xufeng Yao Bei Yu

The Chinese University of Hong Kong

Introduction

In light of the rapidly increasing size of pre-trained VLMs, parameter-efficient transfer learning (PETL) has garnered attention as a viable alternative to full fine-tuning. One such approach is the adapter, which introduces a few trainable parameters into the pre-trained models while preserving the original parameters during adaptation. In this paper, (1) we present a novel modeling framework that recasts adapter tuning after attention as a graph message passing process. (2) Within this framework, tuning adapters in VLMs necessitates handling heterophilic graphs, owing to the disparity between the projected query and value space. (3) Therefore, we propose a new method, p-adapter, to mitigate the heterophilic issue.

Preliminaries

The attention mechanism and adapter tuning in VLMs can be modeled as:

Attention in Transformer

Given query $\boldsymbol{Q} \in \mathbb{R}^{N_1 \times d_k}$, key $\boldsymbol{K} \in \mathbb{R}^{N_2 \times d_k}$ and value $\boldsymbol{V} \in \mathbb{R}^{N_2 \times d_v}$, attention aggregates the features by:

$$\mathrm{Attn}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{M}\boldsymbol{V}, \tag{1}$$

where

$$\boldsymbol{M} = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right) \tag{2}$$

represents the attention weights, N_1 and N_2 are the number of the query and key/value features, respectively.

Adapter Tuning

An adapter is a small learnable module containing two matrices $\boldsymbol{W}_{\text{down}} \in \mathbb{R}^{l_1 \times l_2}$, $\boldsymbol{W}_{\text{up}} \in \mathbb{R}^{l_2 \times l_1}$ and a non-linear function $\sigma(\cdot)$, where l_1 and l_2 are the feature dimensions in pre-trained models and the hidden dimension in adapter (usually $l_2 < l_1$). Given a feature $\boldsymbol{U} \in \mathbb{R}^{N \times l_1}$ in the pre-trained model, the adapter encoding process can be represented as:

$$\boldsymbol{U}' = \sigma(\boldsymbol{U}\boldsymbol{W}_{\text{down}})\boldsymbol{W}_{\text{up}} + \boldsymbol{U}. \tag{3}$$

Modeling Adapter Tuning as Graph Message Passing

From Equation 3 and Equation 1, we can formulate the features sequentially encoded by attention and adapter as:

$$\boldsymbol{U}' = \sigma(\boldsymbol{M}\boldsymbol{V}\boldsymbol{W}_v\boldsymbol{W}_o\boldsymbol{W}_{\text{down}})\boldsymbol{W}_{\text{up}} + \boldsymbol{M}\boldsymbol{V}\boldsymbol{W}_v\boldsymbol{W}_o, \tag{4}$$

where $\boldsymbol{M} \in \mathbb{R}^{N_1 \times N_2}$ is the attention matrix computed by the transformed query $\boldsymbol{Q}\boldsymbol{W}_q$ and key $\boldsymbol{K}\boldsymbol{W}_k$ using Equation 2.

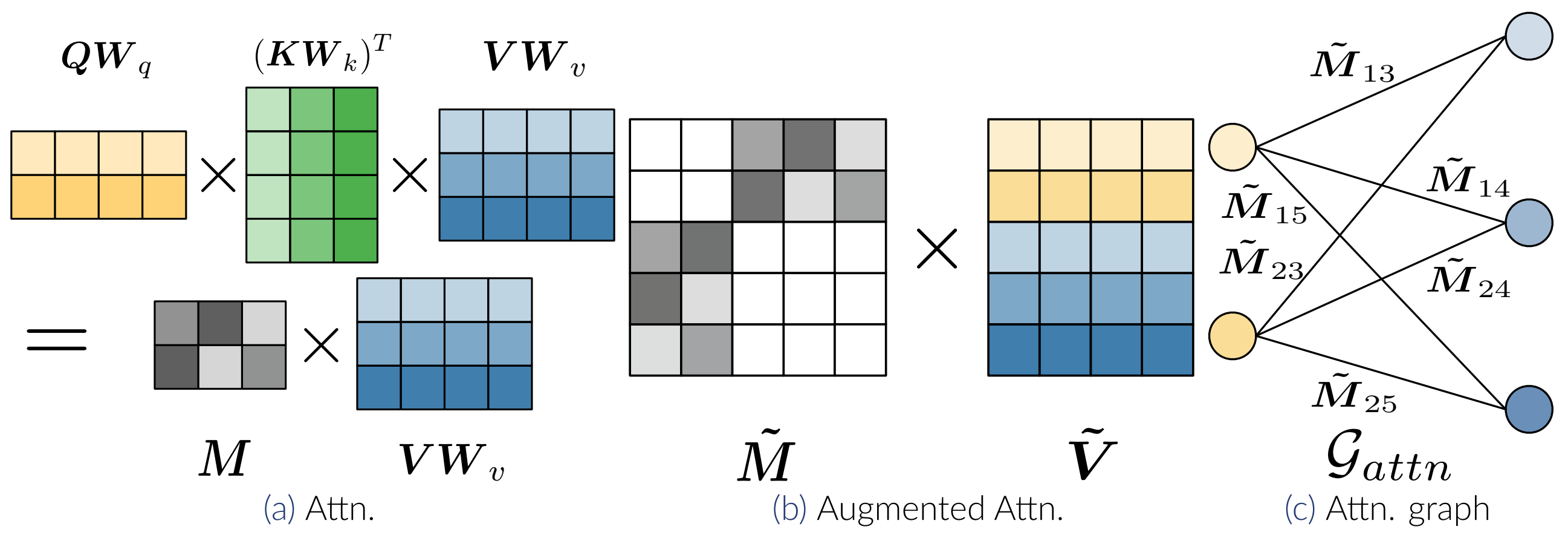
Then, we define the augmented value feature $\tilde{\boldsymbol{V}}$ which concatenates the transformed query and value and the augmented attention matrix $\tilde{\boldsymbol{M}}$ as

$$\tilde{\boldsymbol{V}} = \begin{bmatrix} \boldsymbol{Q}\boldsymbol{W}_q \\ \boldsymbol{V}\boldsymbol{W}_v \end{bmatrix}, \quad \tilde{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{M} \\ \boldsymbol{M}^\top & \boldsymbol{0} \end{bmatrix}. \tag{5}$$

Defining the projected augmented value feature $\hat{\boldsymbol{V}} = \tilde{\boldsymbol{V}}\boldsymbol{W}_o$, with the augmented attention mechanism, we can further define the augmented adapter encoding process by:

$$\tilde{\boldsymbol{U}}' = \sigma(\tilde{\boldsymbol{M}}\hat{\boldsymbol{V}}\boldsymbol{W}_{\text{down}})\boldsymbol{W}_{\text{up}} + \tilde{\boldsymbol{M}}\hat{\boldsymbol{V}}. \tag{6}$$

Comparing Equation 4 and Equation 6, we indicate that the adapter encoding process and the augmented one are equal. Since $\tilde{\boldsymbol{M}}$ is a square and symmetric matrix, we can regard it as the adjacency matrix of the attention graph $\mathcal{G}_{attn}$. With this attention graph, we can transform the adapter encoding process in Equation 4 into spectral graph message passing.

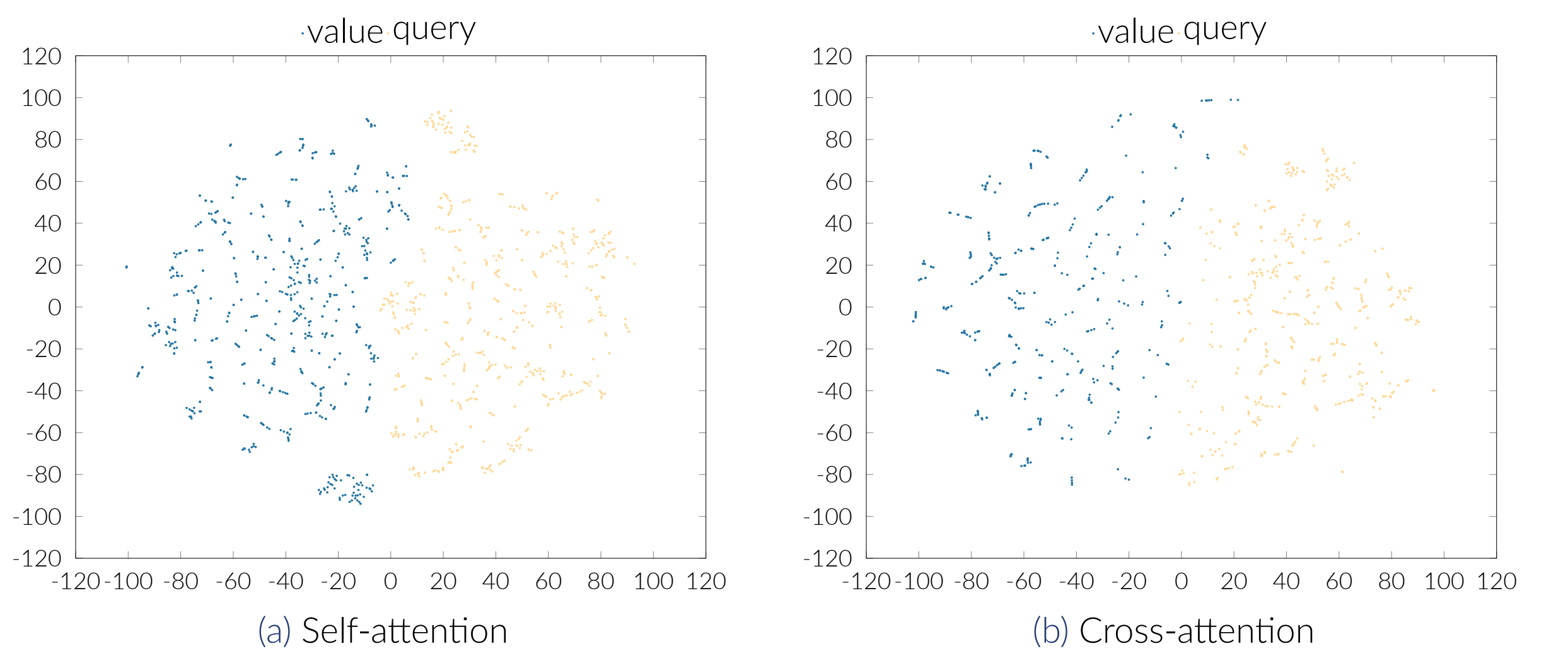

$\boldsymbol{Q}\boldsymbol{W}_q$
$(\boldsymbol{K}\boldsymbol{W}_k)^T$
$\boldsymbol{V}\boldsymbol{W}_v$
×
=
$\boldsymbol{M}$
$\boldsymbol{V}\boldsymbol{W}_v$
$\tilde{\boldsymbol{M}}$
$\tilde{\boldsymbol{V}}$
$\tilde{\boldsymbol{M}}_{13}$
$\tilde{\boldsymbol{M}}_{14}$
$\tilde{\boldsymbol{M}}_{15}$
$\tilde{\boldsymbol{M}}_{23}$
$\tilde{\boldsymbol{M}}_{24}$
$\tilde{\boldsymbol{M}}_{25}$
$\mathcal{G}_{attn}$

(a) Attn. (b) Augmented Attn. (c) Attn. graph

Figure 1. Illustration of the generation of the bipartite attention graph $\mathcal{G}_{attn}$. For simplicity, we omit the scale and softmax functions in attention mechanism.

The Heterophilic in Attention Graph $\mathcal{G}_{attn}$

The attention graph $\mathcal{G}attn$ is a heterophilic graph in which connected nodes have dissimilar features. The visualization of the learned distribution of the projected query and value space is shown in Figure 2.



(a) Self-attention (b) Cross-attention

Figure 2. The t-SNE visualization of the features in the projected query and value space for self- and cross-attention.

p-Adapter

To mitigate the heterophilic issue in attention graph $\mathcal{G}_{attn}$, we propose p-Adapter inspired by p-Laplacian message passing.

p-Laplacian Message Passing [1]

p-Laplacian message passing is proposed for heterophilic graph learning. By denoting $\boldsymbol{\alpha} = \mathrm{diag}(\alpha_{1,1}, \cdots, \alpha_{N,N})$, $\boldsymbol{\beta} = \mathrm{diag}(\beta_{1,1}, \cdots, \beta_{N,N})$, and two hyper-parameters $\mu, p \in \mathbb{R}$, one-layer p-Laplacian message passing can be defined as:

$$\boldsymbol{X}' = \boldsymbol{\alpha}\boldsymbol{D}^{-1/2}\bar{\boldsymbol{A}}\boldsymbol{D}^{-1/2}\boldsymbol{X} + \boldsymbol{\beta}\boldsymbol{X}, \tag{7}$$

where $\bar{\boldsymbol{A}}$ is the p-Laplacian normalized adjacency matrix with entries defined by:

$$\bar{A}_{i,j} = A_{i,j}\left\|\sqrt{\frac{A_{i,j}}{D_{i,i}}}\boldsymbol{X}_{i,:} - \sqrt{\frac{A_{i,j}}{D_{j,j}}}\boldsymbol{X}_{j,:}\right\|^{p-2}, \tag{8}$$

and for all $i, j \in [N]$ we have:

$$\alpha_{i,i} = \left(\sum_{j=1}^{N}\frac{\bar{A}_{i,j}}{D_{i,i}} + \frac{2\mu}{p}\right)^{-1}, \quad \beta_{i,i} = \frac{2\mu}{p}\alpha_{i,i}. \tag{9}$$

p-Adapter Architecture

For p-adapter, we take the attention matrix $\boldsymbol{M}$ and the projected augmented value feature $\hat{\boldsymbol{V}}$, as the output of attention. Then, we augment the attention matrix to $\tilde{\boldsymbol{M}}$, as shown in Equation 5. We then normalize the augmented attention matrix by:

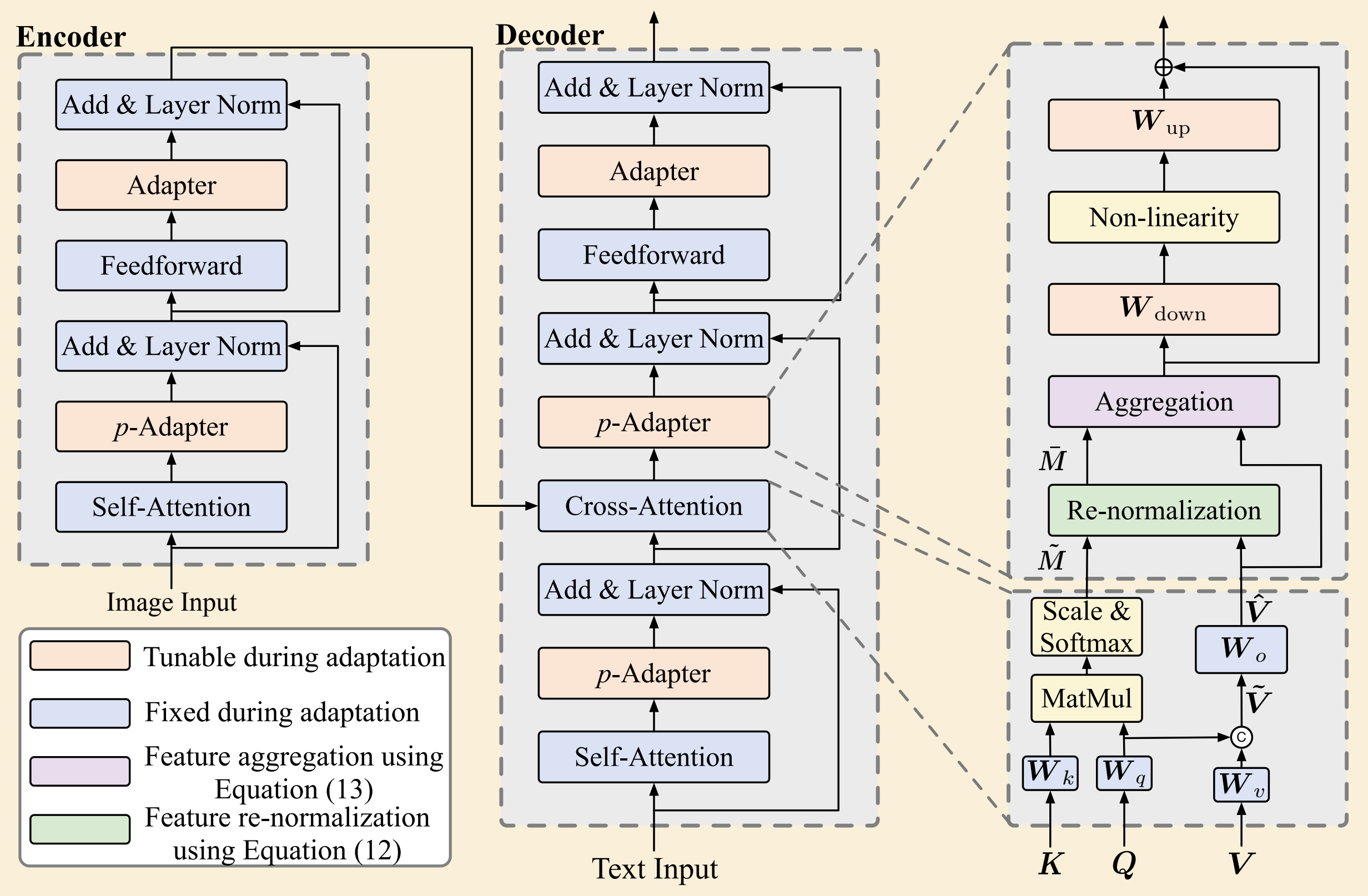
$$\bar{M}_{i,j} = \tilde{M}_{i,j}\left\|\sqrt{\frac{\tilde{M}_{i,j}}{\tilde{D}_{i,i}}}\hat{\boldsymbol{V}}_{i,:} - \sqrt{\frac{\tilde{M}_{i,j}}{\tilde{D}_{j,j}}}\hat{\boldsymbol{V}}_{j,:}\right\|^{p-2}, \tag{10}$$

where $\tilde{\boldsymbol{D}}$ is the degree matrix of $\tilde{\boldsymbol{M}}$. Further, we can aggregate the features using the calibrated attention matrix $\bar{\boldsymbol{M}}$ by

$$\bar{\boldsymbol{U}} = \tilde{\boldsymbol{\alpha}}\tilde{\boldsymbol{D}}^{-1/2}\bar{\boldsymbol{M}}\tilde{\boldsymbol{D}}^{-1/2}\hat{\boldsymbol{V}} + \tilde{\boldsymbol{\beta}}\hat{\boldsymbol{V}}, \tag{11}$$

where $\tilde{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\beta}}$ are caculated according to the algorithm in p-Laplacian message passing. With the aggregated feature $\bar{\boldsymbol{U}}$, we encode it with the learnable adapter weights by:

$$\bar{\boldsymbol{U}}' = \sigma(\bar{\boldsymbol{U}}\boldsymbol{W}_{\text{down}})\boldsymbol{W}_{\text{up}} + \bar{\boldsymbol{U}}. \tag{12}$$



Experiments

Main Experiments

Method	Updated Params (%)	VQA2.0 Karpathy test Acc.(%)	VizWizVQA test-dev Acc.(%)	SNLI_VE test-P Acc.(%)	COCOCaps Karpathy test BLEU@4	COCOCaps Karpathy test CIDEr	TextCaps test-dev BLEU@4	TextCaps test-dev CIDEr	VizWizCaps test-dev BLEU@4	VizWizCaps test-dev CIDEr	Avg.
					BLIP$_{\text{CapFilt-L}}$						
Full fine-tuning	100.00	**70.56**	**36.52**	**78.35**	**39.1**	**128.7**	**27.1**	**91.6**	**45.7**	**170.0**	**76.40**
Prefix tuning	0.71	60.49	22.45	71.82	39.4	127.7	24.8	80.0	40.6	153.3	68.95
LoRA	0.71	66.57	33.39	77.36	38.3	128.3	24.6	82.2	41.3	154.3	71.81
Adapter	6.39	69.53	35.37	78.85	38.9	128.8	25.4	86.7	43.3	160.5	74.15
p-Adapter (Ours)	6.39	**70.39**	**37.16**	**79.40**	**40.4**	**130.9**	**26.1**	**87.0**	**44.5**	**164.1**	**75.54**
					mPLUG$_{\text{ViT-B}}$						
Full fine-tuning	100.00	**70.91**	**59.79**	**78.72**	**40.4**	**134.8**	**23.6**	**74.0**	**42.1**	**157.5**	**75.76**
Prefix tuning	0.71	60.95	47.42	72.11	39.8	133.5	18.8	51.9	35.5	135.6	66.18
LoRA	0.71	66.67	52.49	75.29	39.4	129.4	21.0	64.4	39.5	146.0	70.46
Adapter	6.39	70.65	56.50	78.56	40.3	134.7	22.9	71.5	41.9	155.6	74.73
p-Adapter (Ours)	6.39	**71.36**	**58.08**	**79.26**	**40.4**	**135.3**	**23.2**	**73.3**	**43.1**	**160.1**	**76.01**

Table 1. The main results.

Ablation Study: Different GNNs and Concatenation Patterns

GNN	VQA2.0 Acc.(%)	SNLI_VE Acc.(%)	COCOCaps BLEU@4	COCOCaps CIDEr	Avg.
GCN	69.53	78.85	38.9	128.8	79.02
APPNP	70.22	79.03	39.4	129.1	79.44
GCNII	70.13	79.12	39.7	129.7	79.66
^{p}GNN	**70.39**	**79.40**	**40.4**	**130.9**	**80.27**

Table 2. Ablation study on GNNs.

Concat.	VQA2.0 Acc.(%)	SNLI_VE Acc.(%)	COCOCaps BLEU@4	COCOCaps CIDEr	Avg.
Zero	70.02	79.17	40.2	130.3	79.92
Noise	69.90	78.99	39.9	130.1	79.72
Query	**70.39**	**79.40**	**40.4**	**130.9**	**80.27**

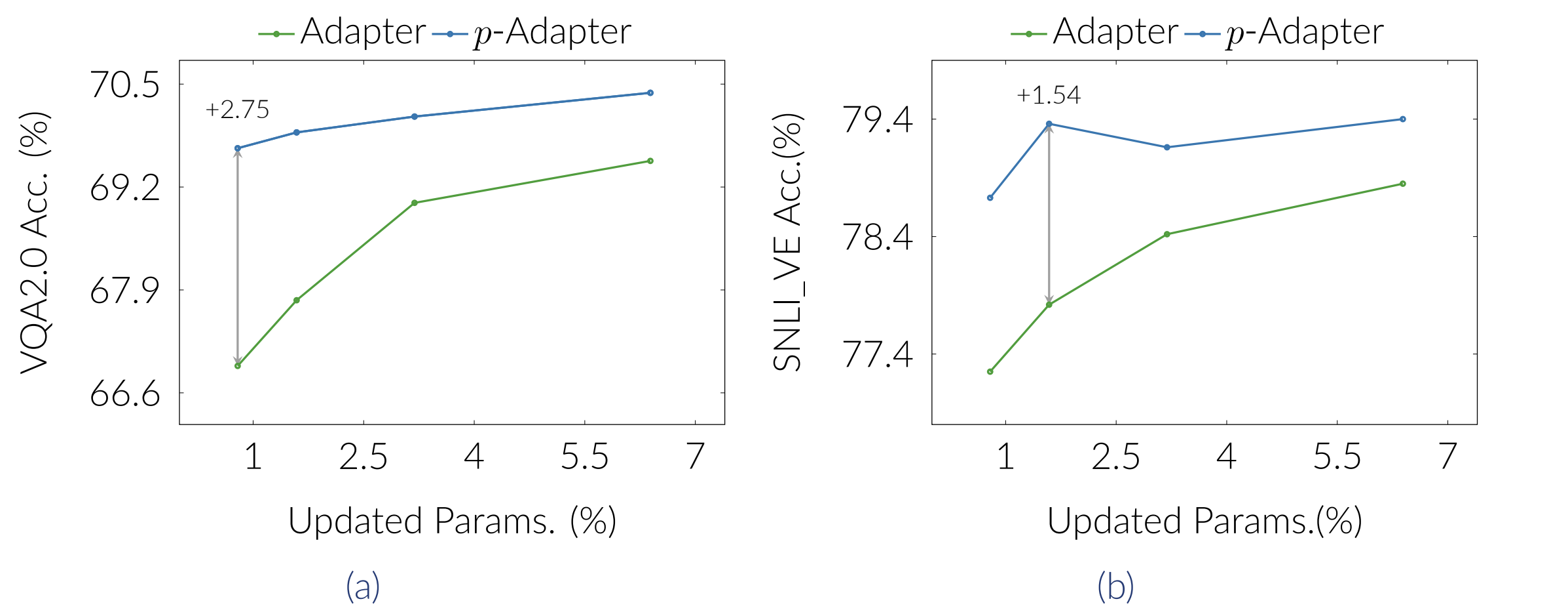
Table 3. Ablation study on concatenation patterns.

Ablation Study: Insertion Positions

Method	FFN	SA	CA	VQA2.0 Acc. (%)	SNLI_VE Acc. (%)	COCOCaps BLEU@4	COCOCaps CIDEr	Avg.
p-Adapter (Imps.)	✓			68.65 (-)	78.21 (-)	38.4 (-)	128.4 (-)	78.41 (-)
	✓	✓		70.11 (**+0.90**)	78.96 (+0.34)	39.9 (+1.4)	130.3 (+1.8)	79.82 (+1.11)
	✓		✓	69.84 (+0.67)	79.17 (**+0.57**)	39.1 (+0.5)	129.4 (+0.7)	79.38 (+0.61)
	✓	✓	✓	**70.39** (+0.86)	**79.40** (+0.55)	**40.4** (**+1.5**)	**130.9** (**+2.1**)	**80.27** (**+1.25**)

Table 4. Ablation study on different insertion positions, including FFN, self- and cross-attention.

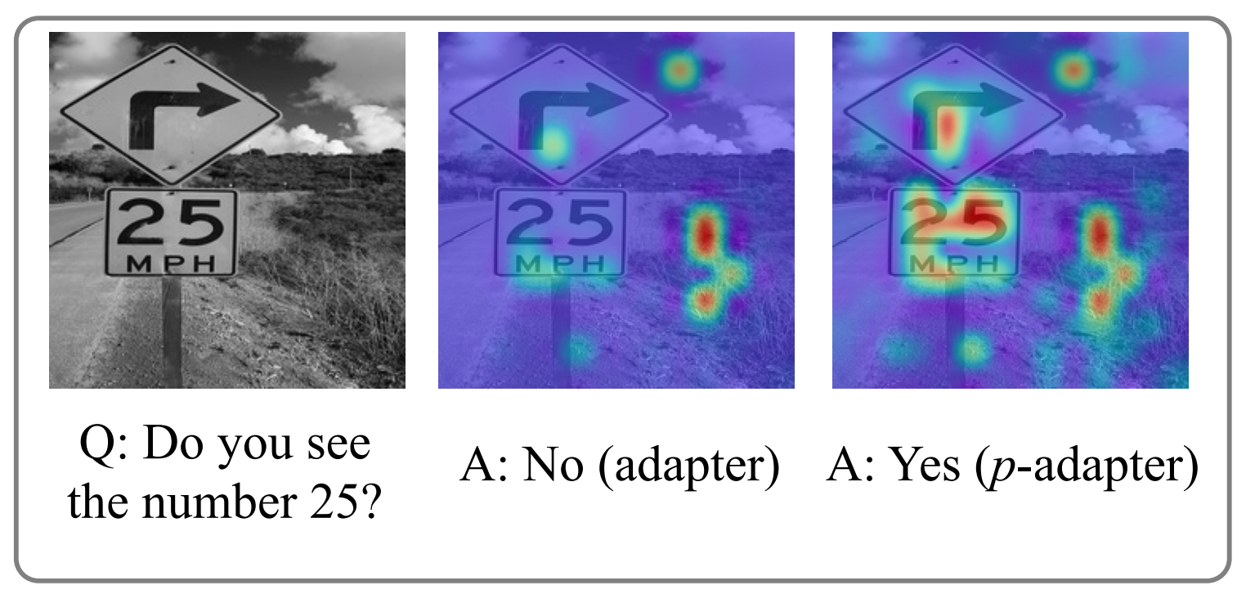
Ablation Study: Adapter Size

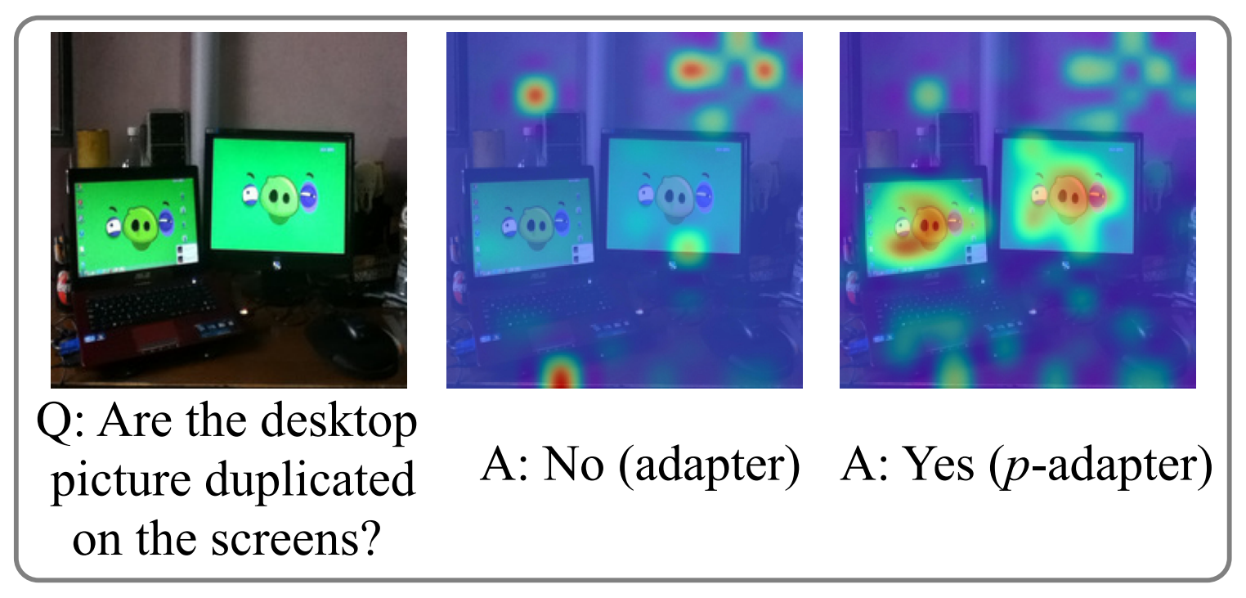


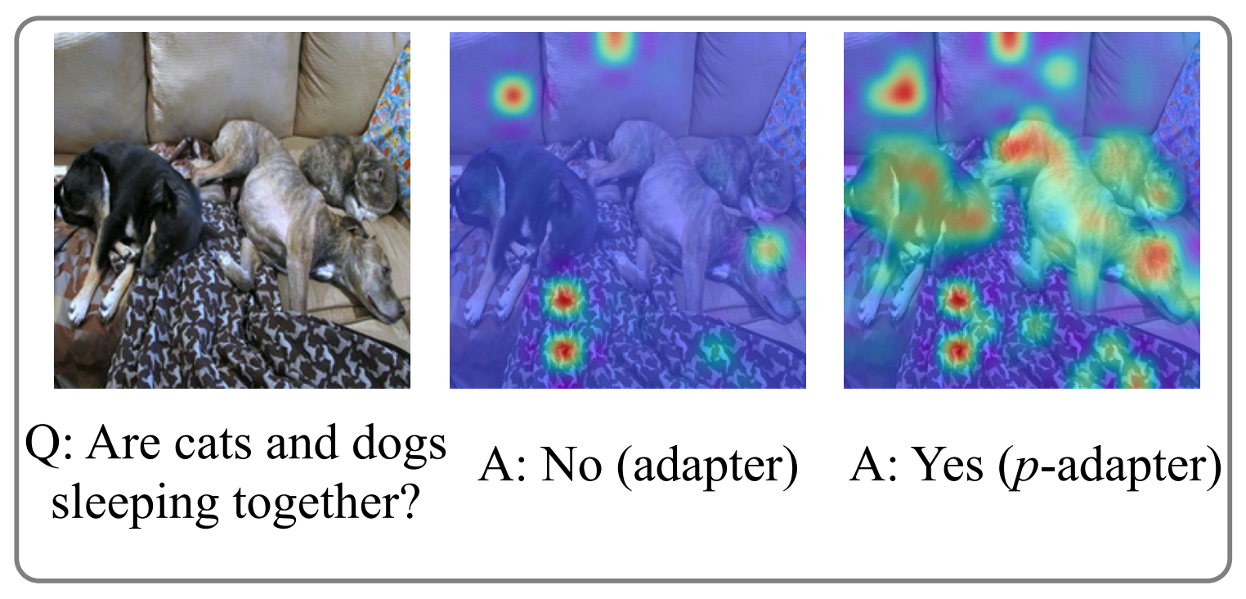
(a) (b)

Figure 3. Ablation study on the adapter size. We report the results on VQA2.0 and SNLI_VE.

Visualization







References

[1] Guoji Fu, Peilin Zhao, and Yatao Bian. p-Laplacian based graph neural networks. In *ICML*, 2022.

The 38th Annual AAAI Conference on Artificial Intelligence, Vancouver, Canada