The 38th Annual AAAI Conference on Artificial Intelligence
FEBRUARY 20-27, 2024 | VANCOUVER, CANADA
VANCOUVER CONVENTION CENTRE – WEST BUILDING

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

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

Department of Computer Science and Engineering
The Chinese University of Hong Kong

Feb. 08, 2024

Introduction

Background

1. By leveraging massive amounts of unlabeled data during training, pre-trained vision-language models can learn highly performant and generalizable representations, leading to improvements on various downstream tasks.
2. As model sizes continue to grow rapidly, fine-tuning is increasingly affected by the parameter-efficiency issue. To address this challenge, researchers proposed parameter-efficient fine-tuning to achieve high parameter efficiency and demonstrated promising results on various downstream tasks.

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[1]

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:

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

[1] Neil Houlsby et al. (2019). "Parameter-efficient transfer learning for NLP". In: *Proc. ICML*. PMLR.

Modeling adapter 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).

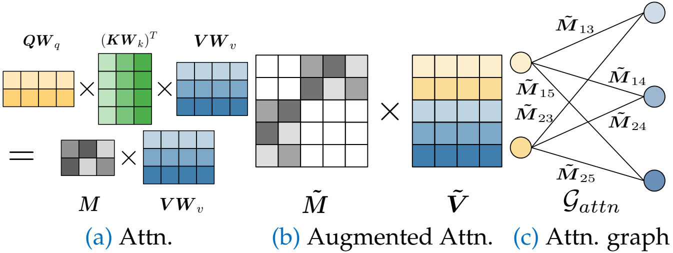


Illustration of the generation of the bipartite attention graph $\mathcal{G}_{attn}$.

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

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

Modeling adapter as graph message passing

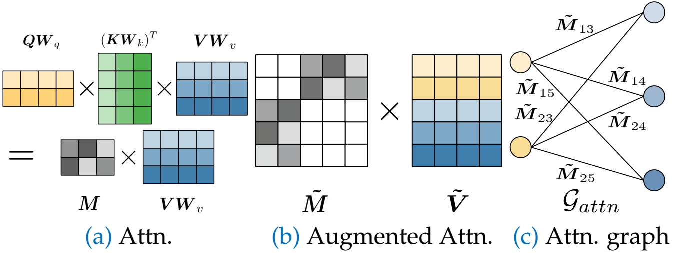
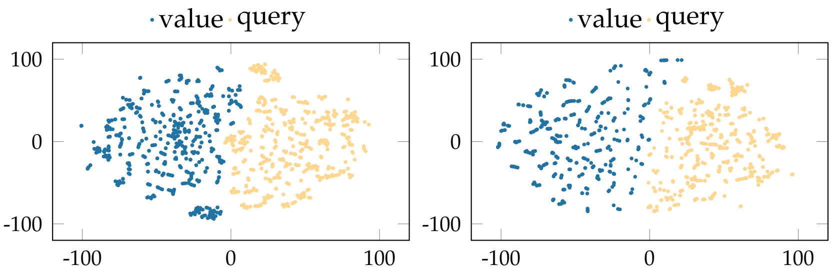


Illustration of the generation of the bipartite attention graph $\mathcal{G}_{attn}$.

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

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

Comparing Equation (4) and Equation (6), we indicate that the adapter encoding process and the augmented one are equal. Since $\tilde{M}$ is a square and symmetric matrix, we can regard it as the adjacency matrix of the attention graph $\mathcal{G}_{attn}$



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

The t-SNE[2] visualization of the features in the projected query and value space for self- and cross-attention. The VLM is $\text{BLIP}_{\text{CapFilt-L}}$[3] and data come from COCO Captions[4].

[2] Laurens Van der Maaten and Geoffrey Hinton (2008). "Visualizing data using t-SNE.". In: *Journal of machine learning research* 9.11.

[3] Junnan Li et al. (2022). "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation". In: *Proc. ICML*.

[4] Tsung-Yi Lin et al. (2014). "Microsoft coco: Common objects in context". In: *Proc. ECCV*. Springer, pp. 740–755.

Method

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. Note that this transformation does not alter any learned parameters in attention. Then, we augment the attention matrix to $\tilde{\boldsymbol{M}}$, as shown in Equation (5). Following p-Laplacian message passing, we normalize the augmented attention matrix by:

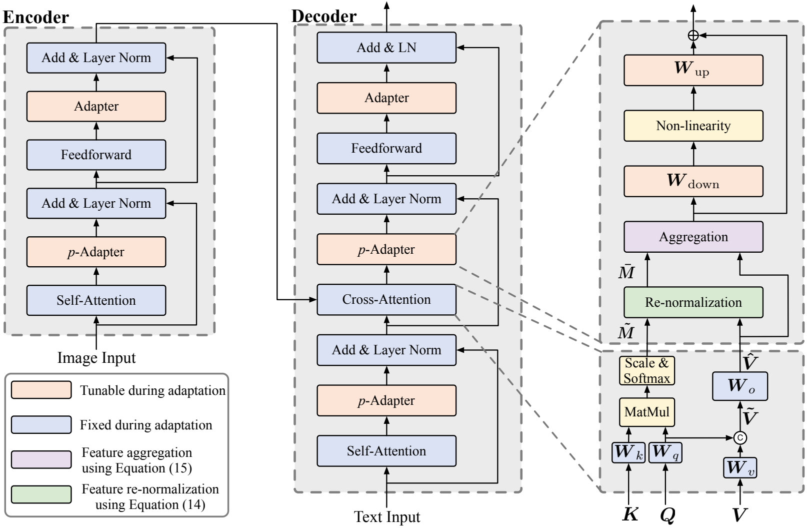
$$\bar{M}_{i,j} = \tilde{M}_{i,j} \left\| \sqrt{\frac{\tilde{M}_{i,j}}{\tilde{D}_{i,i}}} \hat{\boldsymbol{V}}_{i,:} - \sqrt{\frac{\tilde{M}_{i,j}}{\tilde{D}_{j,j}}} \hat{\boldsymbol{V}}_{j,:} \right\|^{p-2}, \tag{7}$$

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

$$\bar{\boldsymbol{U}} = \tilde{\alpha}\tilde{\boldsymbol{D}}^{-1/2}\bar{\boldsymbol{M}}\tilde{\boldsymbol{D}}^{-1/2}\hat{\boldsymbol{V}} + \tilde{\beta}\hat{\boldsymbol{V}}, \tag{8}$$

where $\tilde{\alpha}$ and $\tilde{\beta}$ 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:

$$\bar{\boldsymbol{U}}' = \sigma(\bar{\boldsymbol{U}}\boldsymbol{W}_{\text{down}})\boldsymbol{W}_{\text{up}} + \bar{\boldsymbol{U}}. \tag{9}$$



Overall architecture of *p*-adapter

Experiments

Tasks and datasets

1. For VQA, we consider it as an answer generation problem. We test our model on VQA2.0[5] with the widely-used Karpathy split and VizWizVQA[6].
2. For VE, we adopt SNLI-VE[7] as the evaluation benchmark.
3. For image captioning, we conduct extensive experiments on three benchmarks, i,e., COCO Captions[8] with Karpathy split[9], TextCaps[10], and VizWizCaps[11].

---

[5] Yash Goyal et al. (2017). "Making the v in vqa matter: Elevating the role of image understanding in visual question answering". In: *Proc. CVPR*, pp. 6904–6913.

[6] Danna Gurari, Qing Li, et al. (2018). "Vizwiz grand challenge: Answering visual questions from blind people". In: *Proc. CVPR*, pp. 3608–3617.

[7] Ning Xie et al. (2019). "Visual entailment: A novel task for fine-grained image understanding". In: *arXiv preprint arXiv:1901.06706*.

[8] Tsung-Yi Lin et al. (2014). "Microsoft coco: Common objects in context". In: *Proc. ECCV*. Springer, pp. 740–755.

[9] Andrej Karpathy and Li Fei-Fei (2015). "Deep visual-semantic alignments for generating image descriptions". In: *Proc. CVPR*, pp. 3128–3137.

[10] Oleksii Sidorov et al. (2020). "Textcaps: a dataset for image captioning with reading comprehension". In: *Proc. ECCV*. Springer, pp. 742–758.

[11] Danna Gurari, Yinan Zhao, et al. (2020). "Captioning images taken by people who are blind". In: *Proc. ECCV*. Springer, pp. 417–434.

Implementation details

1. Our experiments are implemented in PyTorch[12] and conducted on 8 Nvidia 3090 GPUs.
2. We validate our method on two generative pre-trained VLMs, $\text{BLIP}_{\text{CapFilt-L}}$[13] and $\text{mPLUG}_{\text{ViT-B}}$[14].

---

[12] Adam Paszke et al. (2019). "Pytorch: An imperative style, high-performance deep learning library". In: *Proc. NeurIPS* 32.

[13] Junnan Li et al. (2022). "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation". In: *Proc. ICML*.

[14] Chenliang Li et al. (2022). "mPLUG: Effective and Efficient Vision-Language Learning by Cross-modal Skip-connections". In: *arXiv preprint arXiv:2205.12005*.

Comparison with transfer learning methods

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.
$BLIP_{CapFilt-L}$											
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**

Table: The main results on various datasets for full fine-tuning, adapter[15], prefix tuning[16], LoRA[17], and our proposed *p*-adapter. We bold the scores for full fine-tuning and the highest scores separately for approaches with PETL methods.

[15] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal (2022). “Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks”. In: *Proc. CVPR*.

[16] Xiang Lisa Li and Percy Liang (2021). “Prefix-Tuning: Optimizing Continuous Prompts for Generation”. In: *Proc. ACL*.

[17] Edward J Hu et al. (2022). “Lora: Low-rank adaptation of large language models”. In: *Proc. ICLR*.

Comparison with transfer learning methods

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.
					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: The main results on various datasets for full fine-tuning, adapter[18], prefix tuning[19], LoRA[20], and our proposed *p*-adapter. We bold the scores for full fine-tuning and the highest scores separately for approaches with PETL methods.

[18] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal (2022). "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks". In: *Proc. CVPR*.

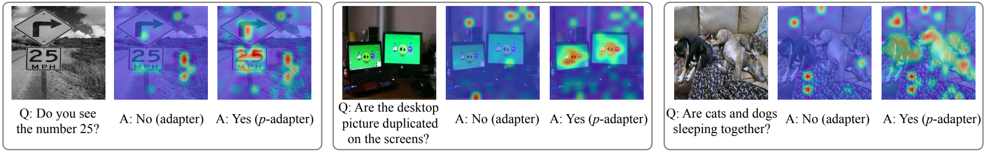
[19] Xiang Lisa Li and Percy Liang (2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proc. ACL*.

[20] Edward J Hu et al. (2022). "Lora: Low-rank adaptation of large language models". In: *Proc. ICLR*.

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: Ablation study on the graph neural networks.

Visualization



Visualization of the attention.

1. To validate the effectiveness of *p*-adapter, we visualize[21] the cross-attention weights at the last transformer layer on some VQA[22] data.
2. We take the `[CLS]` token as the query since it represents the whole question and plot the attention weights on the image features in the key/value space.

---

[21] Hila Chefer, Shir Gur, and Lior Wolf (2021). "Transformer interpretability beyond attention visualization". In: *Proc. CVPR*, pp. 782–791.

[22] Yash Goyal et al. (2017). "Making the v in vqa matter: Elevating the role of image understanding in visual question answering". In: *Proc. CVPR*, pp. 6904–6913.

Conclusion

Conclusion

1. We first propose a new modeling framework for adapter tuning[23] after attention modules in pre-trained VLMs. Within this framework, we can identify the heterophilic nature of the attention graphs, posing challenges for vanilla adapter tuning[24].
2. To mitigate this issue, we propose a new adapter architecture, p-adapter, appended after the attention modules. Inspired by p-Laplacian message passing[25], p-adapters re-normalize the attention weights using node features and aggregate the features with the calibrated attention matrix.
3. Extensive experimental results validate our method's significant superiority over other PETL methods on various VL tasks.

---

[23] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal (2022). "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks". In: *Proc. CVPR*.

[24] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal (2022). "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks". In: *Proc. CVPR*.

[25] Guoji Fu, Peilin Zhao, and Yatao Bian (2022). "p-Laplacian Based Graph Neural Networks". In: *Proc. ICML*.

THANK YOU!