VL-PET: Vision-and-Language Parameter-Efficient Tuning via Granularity Control

Zi-Yuan Hu[1,3] Yanyang Li[1] Michael R. Lyu[1] Liwei Wang[1,2*]
[1]The Chinese University of Hong Kong [2]Centre for Perceptual and Interactive Intelligence
[3]Shanghai Artificial Intelligence Laboratory
{zyhu22,yyli21,lyu,lwwang}@cse.cuhk.edu.hk

Abstract

*As the model size of pre-trained language models (PLMs) grows rapidly, full fine-tuning becomes prohibitively expensive for model training and storage. In vision-and-language (VL), parameter-efficient tuning (PET) techniques are proposed to integrate modular modifications (e.g., Adapter and LoRA) into encoder-decoder PLMs. By tuning a small set of trainable parameters, these techniques perform on par with full fine-tuning. However, excessive modular modifications and neglecting the functionality gap between the encoders and decoders can lead to performance degradation, while existing PET techniques (e.g., VL-Adapter) overlook these critical issues. In this paper, we propose a **V**ision-and-**L**anguage **P**arameter-**E**fficient **T**uning (VL-PET) framework to impose effective control over modular modifications via a novel granularity-controlled mechanism. Considering different granularity-controlled matrices generated by this mechanism, a variety of model-agnostic VL-PET modules can be instantiated from our framework for better efficiency and effectiveness trade-offs. We further propose lightweight PET module designs to enhance VL alignment and modeling for the encoders and maintain text generation for the decoders. Extensive experiments conducted on four image-text tasks and four video-text tasks demonstrate the efficiency, effectiveness and transferability of our VL-PET framework. In particular, our VL-PET$_{\text{large}}$ with lightweight PET module designs significantly outperforms VL-Adapter by 2.92% (3.41%) and LoRA by 3.37% (7.03%) with BART-base (T5-base) on image-text tasks. Furthermore, we validate the enhanced effect of employing our VL-PET designs on existing PET techniques, enabling them to achieve significant performance improvements. Our code is available at https://github.com/HenryHZY/VL-PET.*

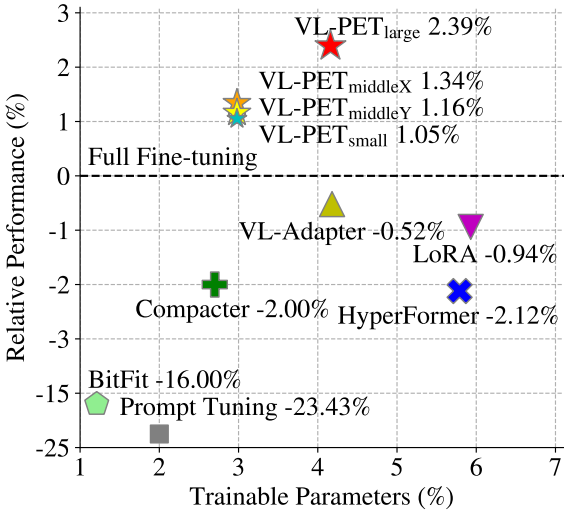

Figure 1. Relative average performance gain of difference PET techniques w.r.t to full fine-tuning. Experiments are conducted with three seeds on four image-text tasks based on BART-base.

1. Introduction

Recently, the paradigm of pre-training transformer-based models on large-scale corpus and then fine-tuning them for downstream tasks has achieved great success in various domains, such as natural language processing (NLP) [50, 9, 33, 27, 45, 2], computer vision (CV) [10, 35, 36, 34, 13, 1], and vision-and-language (VL) [6, 28, 23, 29, 8, 42, 51]. However, as the model size of pre-trained language models (PLMs) and the number of tasks grow rapidly, fine-tuning the entire parameter set of PLMs (i.e., full fine-tuning) and preserving a task-specific copy of PLMs becomes prohibitively expensive for model training and storage.

To mitigate these problems, parameter-efficient tuning (PET) techniques are proposed to save model storage space. As stated in [12], most PET techniques freeze the whole PLM backbone and integrate trainable modular modifications (i.e., additional small trainable PET modules, such as

*Corresponding author.

Adapter [15] and LoRA [16]) into PLM. By only tuning a small set of trainable parameters, these techniques achieve performance comparable to full fine-tuning. Despite the significant achievements of PET in NLP [40, 12, 32, 54, 21, 38, 16, 31] and CV [18, 41, 4, 19, 3, 7, 53], the potential of PET in VL has not been fully explored and requires further VL-specific investigation to bridge the natural modality gap between vision and language. In VL, most PET techniques follow NLP-specific modular modifications [39, 20, 55, 56, 22, 52, 37], while lacking VL-specific designs. Moreover, these techniques mainly focus on discriminative tasks (e.g., image-text retrieval), limiting the generalization ability of PLMs. Although the state-of-the-art PET approach VL-Adapter [49] has studies challenging VL tasks, including discriminative and generative tasks (e.g., image captioning), it directly migrates those NLP-specific modular modifications without deep exploration about the most appropriate design for VL domains.

To conduct a more thorough investigation into VL-specific PET techniques, we raise and analyze **two critical issues neglected by existing PET techniques** in VL: (1) Integrating heavy and excessive modular modifications [12, 46] into PLMs can greatly affect the intermediate output of the PLMs, leading to instability and performance degradation. Therefore, it is crucial to take measures to impose effective control over these modular modifications to achieve better performance on VL tasks. However, state-of-the-art PET techniques (e.g., VL-Adapter) directly integrate modular modifications into PLMs without effective control. (2) For PLMs used in VL tasks, there exists functionality gap between the encoders and decoders [8]. Specifically, the encoders focus on VL alignment and modeling, while the decoders focus on auto-regressive text generation conditioned on the visual-language representations. PLMs rely on the cross-attention modules inside the decoders to bridge the gap between the encoders and decoders. Therefore, it is essential to introduce tailored modular modification designs for each module, thereby enhancing their unique abilities and achieving better performance. However, state-of-the-art PET techniques directly assign identical modular modifications to PLMs without exploring the unique ability of each PLM module, leading to suboptimal performance.

In this paper, we propose a novel **V**ision-and-**L**anguage **P**arameter-**E**fficient **T**uning (VL-PET) framework to address the above issues. We introduce a novel granularity-controlled mechanism to generate a granularity-controlled matrix as effective control over the modular modifications introduced by PET techniques. Considering different granularity control levels, a variety of granularity-controlled matrices are generated by the proposed mechanism with different trainable parameter complexities. With these granularity-controlled matrices and a novel multi-head modular modification, a variety of model-agnostic VL-PET modules can be instantiated from our VL-PET framework for better efficiency and effectiveness trade-offs. Furthermore, conventional PET module designs typically integrate modular modifications (i.e., PET modules such as Adapter) into all self-attention, cross-attention, and feed-forward modules of the PLM backbones. Due to the unique abilities of the encoders and decoders, we propose lightweight PET module designs that facilitate suitable modular modifications integration into the encoders and decoders. For encoders, we integrate our instantiated VL-PET modules into self-attention and feed-forward for better VL alignment and modeling. For decoders, we only integrate our instantiated VL-PET modules into cross-attention to maintain decoder knowledge and enhance text generation. We further assign our instantiated VL-PET module to the value matrix inside the cross-attention, enabling refined and enhanced control over the decoders. Subsequent experiments demonstrate that lightweight designs significantly outperform conventional designs with fewer parameters.

Extensive experiments are conducted on four image-text tasks with BART-base [27], including visual question answering (VQAv2 [11] and GQA [17]), visual reasoning (NLVR^2 [47]) and image captioning (MSCOCO [5]). As shown in Figure 1, all of the proposed VL-PET modules with lightweight PET module designs outperform the state-of-the-art PET techniques. In particular, VL-PET$_{\text{large}}$ (i.e., one of our instantiated VL-PET modules) significantly outperforms VL-Adapter by 2.92% and LoRA by 3.37%. Furthermore, we transfer our model-agnostic VL-PET modules to another larger backbone (i.e., T5-base [45]), where the observed trends of performance improvement remain consistent with those observed in BART-base. Our VL-PET$_{\text{large}}$ still significantly surpasses VL-Adapter by 3.41% and LoRA by 7.03% in the same image-text tasks. However, state-of-the-art PET techniques do not show similar improvements with this larger PLM, and some techniques even exhibit performance degradation due to their heavy and excessive modular modifications integration. For completeness, we also transfer VL-PET modules to four video-text tasks, including video question answering (TVQA [24] and How2QA [29]) and video captioning (TVC [25] and YC2C [57]). Comprehensive experiments and thorough ablation studies demonstrate the efficiency, effectiveness and transferability of our VL-PET framework. Moreover, we validate the enhanced effect of employing VL-PET designs (e.g., granularity-controlled mechanism and lightweight PET module designs) on existing PET techniques (e.g., Compacter [21] and VL-Adapter), enabling them to achieve significant performance improvements.

2. Related Work

Generative Pre-trained Language Model. Fine-tuning pre-trained language models (PLMs) for downstream tasks

has achieved great success in various domains. However, the architecture of PLMs also limits its applicability to downstream tasks. PLMs with encoder-only architecture [9, 33] are effective for discriminative tasks, while PLMs with decoder-only architecture [43, 44, 2] are better suited for generative tasks. PLMs with encoder-decoder architecture [50, 27, 45] are more generalized, as they can handle both discriminative and generative tasks. To demonstrate the effectiveness of our VL-PET framework on challenging downstream tasks (e.g., discriminative and generative tasks), we adopt encoder-decoder generative PLMs (e.g., BART-base [27] and T5-base [45]) as our backbones.

Parameter-efficient Tuning. Parameter-efficient Tuning (PET) techniques are proposed to alleviate the exorbitant cost of model storage. By fine-tuning PLMs with only a small set of trainable parameters, PET techniques perform on par with full fine-tuning. Existing PET techniques can be divided into two research categories: (1) As stated in [12], most PET techniques add new trainable parameters into PLMs (e.g., adapter-based PET techniques [15] and prompt-based PET techniques [26]); (2) Other PET techniques fine-tune a partial parameter set of the original PLMs (e.g., BitFit [54]). Despite the rapid development of PET techniques in NLP [40, 12, 32, 54, 21, 38, 16, 31] and CV [18, 41, 4, 19, 3, 7, 53], most PET techniques in VL are prompt-based or mainly focus on discriminative tasks [39, 20, 55, 56, 22, 52, 37], which limits the generalization ability of PLMs. State-of-the-art PET techniques [49, 48] focus on some challenging VL tasks with encoder-decoder generative PLMs. Regardless of the risk of performance degradation caused by excessive modular modifications and the neglect of the unique abilities of the encoders and decoders, VL-Adapter [49] directly migrates NLP-specific modular modifications without making VL-specific designs. In this work, we propose a VL-PET framework with a granularity-controlled mechanism, multi-head modular modifications and lightweight PET module designs to tackle these issues neglected by existing PET techniques.

3. VL-PET Framework

In this section, we propose a novel **V**ision-and-**L**anguage **P**arameter-**E**fficient **T**uning (VL-PET) framework for encoder-decoder generative PLMs. An illustration of our model is shown in Fig. 2. We propose a novel granularity-controlled mechanism to generate a granularity-controlled matrix at different granularity control levels, which regulates the output of the modular modifications introduced by PET techniques. As shown in Fig. 3, considering different granularity control levels and a multi-head modular modification, a variety of model-agnostic VL-PET modules can be instantiated from the proposed VL-PET framework. We further propose lightweight PET module designs to facilitate suitable VL-PET module integration into the encoders and decoders.

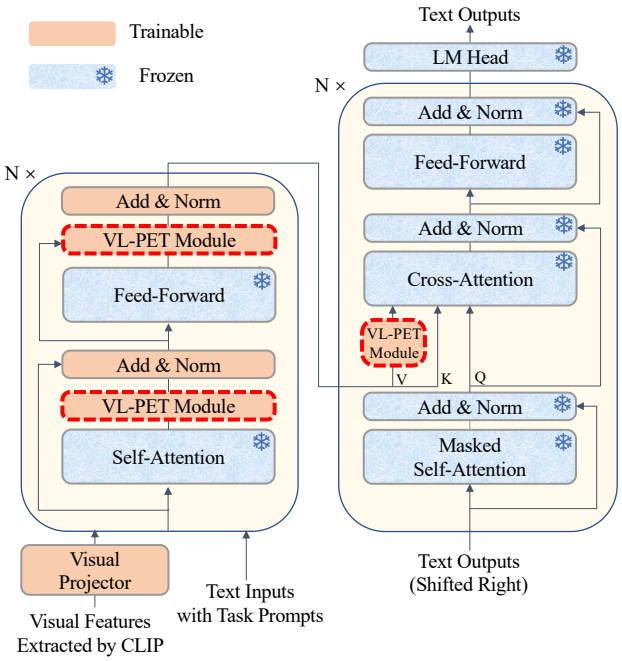

Figure 2. Illustration of an encoder-decoder generative pre-trained language model backbone with model-agnostic VL-PET modules and lightweight PET module designs.

3.1. Preliminary

As stated in [12], most PET techniques can be attributed to introducing trainable modular modifications (e.g., Adapter [15] and LoRA [16]) into PLMs and updating the outputs of frozen PLM modules (e.g., self-attention, feed-forward, cross-attention and value matrix of cross-attention). With this unified perspective, tuning with a trainable PET module can be formulated as follows:

$$\mathbf{H} \leftarrow \mathbf{H} + \Delta\mathbf{H} \tag{1}$$

where $\mathbf{H} \in \mathbb{R}^{N \times d}$ refers to an intermediate hidden state of length N and dimension d from a PLM module, and $\Delta\mathbf{H} \in \mathbb{R}^{N \times d}$ refers to a modular modification introduced by a PET module. In VL tasks, state-of-the-art methods usually migrate PET techniques [21, 38, 15] from NLP and CV without making VL-specific designs. Moreover, these techniques fail to impose effective control over these modular modifications, while excessive modular modifications may lead to performance degradation. Therefore, we aim to tackle this issue with a granularity-controlled mechanism.

3.2. Granularity-controlled Mechanism

To impose effective control over the modular modifications $\Delta\mathbf{H}$, we propose a granularity-controlled mechanism

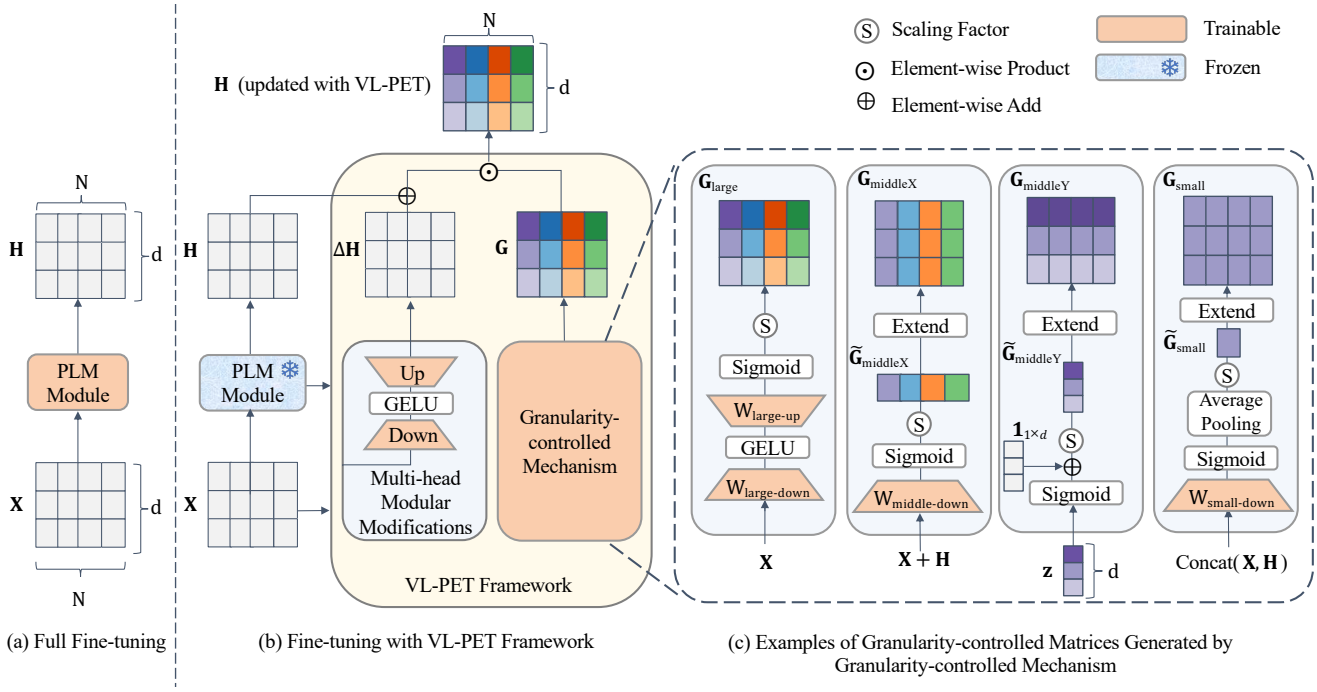

Figure 3. Comparison between full fine-tuning and fine-tuning with VL-PET framework. PLM module refers to a sub-module of PLMs (e.g., self-attention, feed-forward, cross-attention and value matrix of cross-attention). We denote $\mathbf{X}$ of length N and dimension d as the input of a PLM module, $\mathbf{H}$ as the output of a PLM module, $\Delta\mathbf{H}$ as a multi-head modular modification, $\boldsymbol{z}$ as a trainable vector of dimension d, $\mathbf{1}_{1\times d}$ as an all-one matrix and $\mathbf{G}$ as a granularity-controlled matrix generated by the granularity-controlled mechanism.

that assigns a granularity-controlled matrix $\mathbf{G} \in \mathbb{R}^{N\times d}$ to the updating of the intermediate hidden state $\mathbf{H}$. Adopting a unified perspective from Eq. (1), the granularity-controlled mechanism can be expressed as a unified formula:

$$\mathbf{H} \leftarrow \mathbf{G} \odot (\mathbf{H} + \Delta\mathbf{H}) \tag{2}$$

where $\odot$ denotes element-wise product. Given the input $\mathbf{X} \in \mathbb{R}^{N\times d}$ and output $\mathbf{H}$ of a PLM module, a granularity-controlled matrix $\mathbf{G}$ can be generated at different granularity levels according to necessities. In this paper, we generate $\mathbf{G}$ into four granularity control levels (i.e., large, middleX, middleY and small). Specifically, we directly generate a trainable matrix $\mathbf{G}_{\text{large}} \in \mathbb{R}^{N\times d}$ at the large level. At the middleX level, we first generate a trainable matrix $\widetilde{\mathbf{G}}_{\text{middleX}} \in \mathbb{R}^{N\times 1}$ and then extend it to $\mathbf{G}_{\text{middleX}} \in \mathbb{R}^{N\times d}$ without extra trainable parameters. Similarly, we first generate a trainable matrix $\widetilde{\mathbf{G}}_{\text{middleY}} \in \mathbb{R}^{1\times d}$ and $\widetilde{\mathbf{G}}_{\text{small}} \in \mathbb{R}^{1\times 1}$ for the middleY and small, respectively, and then extend them to $\mathbf{G}_{\text{middleY}} \in \mathbb{R}^{N\times d}$ and $\mathbf{G}_{\text{small}} \in \mathbb{R}^{N\times d}$ without additional trainable parameters.

Given a specific granularity-controlled matrix, a specific VL-PET module can be instantiated from our VL-PET framework. Next, we provide one granularity-controlled matrix generation method for each granularity level, respectively, as shown in Fig. 3 and Tab. 1.

Level	Trainable Matrix	Granularity-controlled Matrix	Trainable Parameter Complexity
Large	$\mathbf{G}_{\text{large}} \in \mathbb{R}^{N\times d}$	$\mathbf{G}_{\text{large}} \in \mathbb{R}^{N\times d}$	$\mathcal{O}(dr)$
MiddleX	$\widetilde{\mathbf{G}}_{\text{middleX}} \in \mathbb{R}^{N\times 1}$	$\mathbf{G}_{\text{middleX}} \in \mathbb{R}^{N\times d}$	$\mathcal{O}(d)$
MiddleY	$\widetilde{\mathbf{G}}_{\text{middleY}} \in \mathbb{R}^{1\times d}$	$\mathbf{G}_{\text{middleY}} \in \mathbb{R}^{N\times d}$	$\mathcal{O}(d)$
Small	$\widetilde{\mathbf{G}}_{\text{small}} \in \mathbb{R}^{1\times 1}$	$\mathbf{G}_{\text{small}} \in \mathbb{R}^{N\times d}$	$\mathcal{O}(d)$

Table 1. The proposed granularity-controlled matrices at different granularity control levels, which are extended from trainable matrices. Trainable parameter complexity is determined by the method for generating a trainable matrix.

(1) Large Level. At this level, we generate a granularity-controlled matrix $\mathbf{G}_{\text{large}} \in \mathbb{R}^{N\times d}$ with a bottleneck architecture as follows:

$$\mathbf{G}_{\text{large}} = s \cdot \sigma(\phi(\mathbf{X}\mathbf{W}_{\text{large-down}})\mathbf{W}_{\text{large-up}}) \tag{3}$$

where $\mathbf{W}_{\text{large-down}} \in \mathbb{R}^{d\times r}$ is a down projection layer which projects features from dimension d to projected hidden dimension r, ϕ is a non-linear GELU function [14], $\mathbf{W}_{\text{large-up}} \in \mathbb{R}^{r\times d}$ is an up projection layer, σ is a sigmoid function and s is a scaling factor, which is a hyper-parameter specialized for different PLMs. Therefore, the trainable parameter complexity of $\mathbf{G}_{\text{large}}$ is $\mathcal{O}(dr)$.

(2) MiddleX Level. We first define an all-one matrix as $\mathbf{1}_{m\times n} \in \mathbb{R}^{m\times n}$. The intermediate output of the granularity-controlled mechanism at middleX level is $\widetilde{\mathbf{G}}_{\text{middleX}} \in$

Method	Trainable Params (%)	VQA Acc. (%)	GQA Acc. (%)	NLVR^2 Acc. (%)	COCO Cap. (CIDEr)	Avg.
Backbone: BART-base						
Full Fine-tuning♣	100	$66.88_{0.17}$	$56.79_{0.41}$	$73.66_{0.21}$	$112.01_{0.93}$	$77.33_{0.39}$
BitFit♣ [54]	1.21	$52.94_{0.52}$	$43.15_{0.94}$	$52.29_{0.76}$	$111.44_{0.41}$	$64.96_{0.17}$
Prompt Tuning♣ [26]	2.00	$44.12_{0.45}$	$36.37_{0.35}$	$51.34_{0.83}$	$105.02_{0.24}$	$59.21_{0.21}$
Compacter♣ [21]	2.70	$64.63_{0.09}$	$52.70_{0.24}$	$71.11_{0.35}$	$114.69_{0.42}$	$75.78_{0.21}$
HyperFormer♣ [38]	5.79	$64.62_{0.67}$	$52.55_{0.64}$	$70.74_{1.45}$	$114.84_{0.38}$	$75.69_{0.74}$
LoRA♣ [16]	5.93	$65.15_{0.16}$	$53.66_{0.84}$	$72.58_{0.73}$	$115.01_{0.26}$	$76.60_{0.32}$
VL-Adapter♣ [49]	4.18	$\underline{65.76}_{0.28}$	$54.16_{0.44}$	$\underline{73.19}_{0.71}$	$114.61_{0.26}$	$76.93_{0.25}$
VL-PET$_{\text{small}}$	2.98	$65.43_{0.06}$	$54.03_{0.14}$	$72.43_{0.22}$	$120.68_{0.35}$	$78.14_{0.11}$
VL-PET$_{\text{middleX}}$	2.98	$65.54_{0.09}$	$\underline{54.53}_{0.15}$	$72.66_{0.17}$	$\underline{120.72}_{0.51}$	$\underline{78.37}_{0.14}$
VL-PET$_{\text{middleY}}$	2.98	$65.36_{0.15}$	$53.83_{0.39}$	$\mathbf{73.43}_{0.78}$	$120.31_{0.09}$	$78.23_{0.19}$
VL-PET$_{\text{large}}$	4.16	$\mathbf{66.17}_{0.27}$	$\mathbf{55.11}_{0.17}$	$\mathbf{73.43}_{0.35}$	$\mathbf{122.03}_{0.46}$	$\mathbf{79.18}_{0.14}$
Backbone: T5-base						
Full Fine-tuning♠	100	$67.10_{0.10}$	$56.30_{0.30}$	$74.30_{0.40}$	$112.20_{0.30}$	$77.50_{0.30}$
BitFit♠ [54]	0.83	$55.10_{0.20}$	$45.50_{0.20}$	$51.70_{1.10}$	$101.20_{0.20}$	$63.40_{0.10}$
Prompt Tuning♠ [26]	1.26	$47.40_{0.70}$	$40.60_{0.40}$	$51.00_{0.40}$	$96.10_{0.90}$	$58.80_{0.60}$
LoRA♠ [16]	7.54	$63.70_{0.20}$	$53.30_{0.10}$	$70.00_{0.30}$	$110.30_{0.40}$	$74.30_{0.10}$
VL-Adapter♠ [49]	7.98	$\mathbf{67.10}_{0.10}$	$\underline{56.00}_{0.40}$	$72.70_{0.30}$	$111.80_{0.10}$	$76.90_{0.20}$
LST♠ [48]	7.46	$66.50_{0.10}$	$55.90_{0.10}$	$71.60_{0.30}$	$113.50_{0.30}$	$76.90_{0.10}$
VL-PET$_{\text{small}}$	4.51	$65.88_{0.31}$	$54.96_{1.01}$	$72.64_{0.09}$	$120.05_{0.41}$	$78.38_{0.37}$
VL-PET$_{\text{middleX}}$	4.50	$66.63_{0.14}$	$55.87_{0.25}$	$\mathbf{74.11}_{0.37}$	$\underline{120.41}_{0.31}$	$\underline{79.26}_{0.26}$
VL-PET$_{\text{middleY}}$	4.50	$66.62_{0.20}$	$55.87_{0.13}$	$73.91_{0.45}$	$120.26_{0.40}$	$79.17_{0.08}$
VL-PET$_{\text{large}}$	7.31	$\underline{66.95}_{0.21}$	$\mathbf{56.06}_{0.21}$	$\underline{73.42}_{0.46}$	$\mathbf{121.66}_{0.06}$	$\mathbf{79.52}_{0.21}$

Table 2. Performance on image-text tasks with different PLM backbones. We report the average result with three seeds for a fair comparison, where the subscript is the standard deviation. As analyzed in Sec. 4.2, our special VL-PET designs (e.g., lightweight PET module designs) enable our proposed method to surpass other PET techniques in the downstream tasks (e.g., COCO Captioning). (**Bold** refers to the best result among all PET techniques and underline refers to the second-best result among all PET techniques. ♣: We reproduce the results with three seeds in [49], which only provides results with one seed. ♠: We present the results with three seeds from [48].)

$\mathbb{R}^{N\times 1}$, which can be calculated as follows:

$$\widetilde{\mathbf{G}}_{\text{middleX}} = s \cdot \sigma((\mathbf{X}+\mathbf{H})\mathbf{W}_{\text{middle-down}}) \tag{4}$$

where $\mathbf{W}_{\text{middle-down}} \in \mathbb{R}^{d\times 1}$ is a down projection layer. Next, we use an all-one matrix $\mathbf{1}_{1\times d} \in \mathbb{R}^{1\times d}$ to copy $\widetilde{\mathbf{G}}_{\text{middleX}}$ d times to construct $\mathbf{G}_{\text{middleX}} \in \mathbb{R}^{N\times d}$:

$$\mathbf{G}_{\text{middleX}} = \widetilde{\mathbf{G}}_{\text{middleX}}\mathbf{1}_{1\times d} \tag{5}$$

The trainable parameter complexity of $\mathbf{G}_{\text{middleX}}$ is $\mathcal{O}(d)$.

(3) MiddleY Level. The granularity-controlled matrix $\mathbf{G}_{\text{middleY}} \in \mathbb{R}^{N\times d}$ can be viewed as a variant of $\mathbf{G}_{\text{middleX}}$ as shown in Fig. 3. Instead of utilizing hidden states from the PLM backbone, we adopt a trainable vector $\boldsymbol{z} \in \mathbb{R}^{1\times d}$ to calculate the intermediate output $\widetilde{\mathbf{G}}_{\text{middleY}} \in \mathbb{R}^{1\times d}$:

$$\widetilde{\mathbf{G}}_{\text{middleY}} = s \cdot (\sigma(\boldsymbol{z}) + \mathbf{1}_{1\times d}) \tag{6}$$

The trainable parameter complexity of $\mathbf{G}_{\text{middleY}}$ is $\mathcal{O}(d)$. Similarly, we extend $\widetilde{\mathbf{G}}_{\text{middleY}}$ to $\mathbf{G}_{\text{middleY}}$ as follow:

$$\mathbf{G}_{\text{middleY}} = \mathbf{1}_{N\times 1}\widetilde{\mathbf{G}}_{\text{middleX}} \tag{7}$$

(4) Small Level. Compared to other levels, the granularity-controlled mechanism produces the smallest intermediate output $\widetilde{\mathbf{G}}_{\text{small}} \in \mathbb{R}^{1\times 1}$ at this level. We concatenate $\mathbf{X}$ and $\mathbf{H}$ along the d dimension to compute $\widetilde{\mathbf{G}}_{\text{small}}$:

$$\widetilde{\mathbf{G}}_{\text{small}} = s \cdot \psi(\sigma(\text{Concat}(\mathbf{X},\mathbf{H})\mathbf{W}_{\text{small-down}})) \tag{8}$$

where $\mathbf{W}_{\text{small-down}} \in \mathbb{R}^{2d\times 1}$ is a down projection layer and ψ denotes average pooling along the dimension N. Therefore, the trainable parameter complexity of this level is $\mathcal{O}(d)$. In the end, we expand $\widetilde{\mathbf{G}}_{\text{small}}$ to $\mathbf{G}_{\text{small}} \in \mathbb{R}^{N\times d}$:

$$\mathbf{G}_{\text{small}} = \mathbf{1}_{N\times 1}\widetilde{\mathbf{G}}_{\text{small}}\mathbf{1}_{1\times d} \quad (9)$$

3.3. VL-PET Module with Lightweight Designs

Multi-head Modular Modification. Prior to introducing our lightweight PET module designs, we introduce a novel and more effective modular modification into the PLMs in a multi-head manner. Supposed that N_h is the number of heads and $\mathbf{X}' \in \mathbb{R}^{N\times d}$ is the input, a multi-head modular modification $\Delta\mathbf{H}' \in \mathbb{R}^{N\times d}$ is defined as:

$$\Delta\mathbf{H}' = \phi(\text{Concat}(\mathbf{X}'\mathbf{W}_{\text{down}}^{(1)}, \cdots, \mathbf{X}'\mathbf{W}_{\text{down}}^{(N_h)}))\mathbf{W}_{\text{up}} \quad (10)$$

where $\mathbf{W}_{\text{down}}^{(i)} \in \mathbb{R}^{d\times \frac{r}{N_h}}$ is a down projection layer for head_i and $\mathbf{W}_{\text{up}} \in \mathbb{R}^{r\times d}$ is a up projection layer. Considering a multi-head modular modification and different granularity-controlled matrices describe in Sec. 3.2, a variety of model-agnostic VL-PET modules can be instantiated from our VL-PET framework for better efficiency and effectiveness trade-offs.

Lightweight PET Module Designs. Conventional PET module designs typically apply modular modifications to all self-attention, cross-attention, and feed-forward modules of the PLM backbones. In VL tasks, state-of-the-art PET techniques [49] follow these designs for encoder-decoder PLMs, neglecting the unique abilities of the encoders and decoders. To mitigate this issue, we propose lightweight PET module designs that facilitate suitable modular modifications integration into the encoders and decoders. Specifically, our lightweight designs present a simple yet efficient idea that decoder PET modules should be lightweight and refined compared to encoder PET modules.

Since PLMs are trained on text-only data, adapting PLM encoders to learn unseen visual representation is crucial for VL tasks. Therefore, we aim to enhance the visual-language alignment and modeling ability of the encoders with relatively heavy encoder PET modules. Specifically, we integrate our instantiated VL-PET modules (utilizing $\mathbf{G}_{\text{large}}$, $\mathbf{G}_{\text{middleX}}$, $\mathbf{G}_{\text{middleY}}$ or $\mathbf{G}_{\text{small}}$) into both self-attention and feed-forward modules. As a result, we set $\mathbf{X}'$ as the output of self-attention or feed-forward modules for Eq. (10).

For the decoder VL-PET modules, we want to avoid heavy and excessive modular modification in the PLM decoders as they are already good at text generation. Since PLMs utilize cross-attention to bridge the functionality and modality gap between the encoders and decoders, we only integrate PET modules into cross-attention modules. Specifically, we employ our VL-PET modules (utilizing $\mathbf{G} = \mathbf{1}_{N\times d}$ for parameter-efficiency) to the value matrices of cross-attention only, enabling lightweight and refined control over the decoders. In this case, we set $\mathbf{X}'$ as the input of value matrices (i.e., the final output of the encoders).

Method	Trainable Params (%)	TVQA Acc. (%)	How2QA Acc. (%)	TVC Cap. (CIDEr)	YC2C Cap. (CIDEr)	Avg.
Full Fine-tuning	100	77.69	74.79	50.56	151.71	88.69
BitFit	0.38	66.05	65.42	31.16	115.23	69.47
Prompt Tuning	1.18	24.51	27.76	30.22	108.04	47.63
Compacter	1.89	73.78	72.14	41.39	140.52	81.96
LoRA	5.17	75.51	72.69	44.17	142.72	83.77
VL-Adapter	3.39	77.06	74.73	46.72	<u>153.28</u>	<u>87.95</u>
VL-PET$_{\text{small}}$	2.18	<u>77.69</u>	74.89	47.92	150.24	87.69
VL-PET$_{\text{middleY}}$	2.17	**77.76**	<u>75.40</u>	**48.30**	150.25	87.93
VL-PET$_{\text{middleY}}$	2.17	77.58	75.15	<u>47.93</u>	151.13	<u>87.95</u>
VL-PET$_{\text{large}}$	3.37	76.97	**75.60**	47.53	**154.41**	**88.63**

Table 3. Performance on video-text tasks with BART-base. We report the result with one seed due to the submission limit of VALUE benchmark. (**Bold**: best result among all PET techniques. <u>Underline</u>: second-best result among all PET techniques.)

Method	Params (%)	VQA (%)	GQA (%)	NLVR^2 (%)	COCO (CIDEr)	Avg.
VL-PET w/o $\mathbf{G}$	2.97	$65.22_{0.14}$	$53.35_{0.39}$	$72.65_{0.44}$	$120.19_{0.68}$	$77.85_{0.34}$
VL-PET$_{\text{small}}$	2.98	$65.43_{0.06}$	$54.03_{0.14}$	$72.43_{0.22}$	$120.68_{0.35}$	$78.14_{0.11}$
VL-PET$_{\text{middleX}}$	2.98	<u>65.54</u>$_{0.09}$	<u>54.53</u>$_{0.15}$	<u>72.66</u>$_{0.17}$	<u>120.72</u>$_{0.51}$	<u>78.37</u>$_{0.14}$
VL-PET$_{\text{middleY}}$	2.98	$65.36_{0.15}$	$53.83_{0.39}$	$\mathbf{73.43}_{0.78}$	$120.31_{0.09}$	$78.23_{0.19}$
VL-PET$_{\text{large}}$	4.16	$\mathbf{66.17}_{0.27}$	$\mathbf{55.11}_{0.17}$	$\mathbf{73.43}_{0.35}$	$\mathbf{122.03}_{0.46}$	$\mathbf{79.18}_{0.14}$

Table 4. Effectiveness of granularity-controlled mechanism in BART-base. (**Bold**: best result. <u>Underline</u>: second-best result.)

4. Experiments

4.1. Experimental Settings

Datasets. In this work, we conduct experiments on four image-text downstream tasks and four video-text downstream tasks. Image-text tasks consist of visual question answering (VQAv2 [11] and GQA [17]), visual reasoning (NLVR^2 [47]) and image captioning (MSCOCO [5]). Video-text tasks consist of video question answering (TVQA [24] and How2QA [29]) and video captioning (TVC [25] and YC2C [57]) from VALUE [30] benchmark.

Baselines and Evaluations. Our baselines consist of state-of-the-art PET techniques, including BitFit [54], Prompt Tuning [26], Compacter [21], HyperFormer [38], LoRA [16], VL-Adapter [49] and LST [48]. To compare these baselines with conventional PET module designs, we propose four VL-PET modules (denoted as VL-PET$_{\text{large}}$, VL-PET$_{\text{middleX}}$, VL-PET$_{\text{middleY}}$ and VL-PET$_{\text{small}}$) with lightweight PET module designs described in Sec. 3.2. We also include full fine-tuning (i.e., train the entire PLMs without PET modules) to facilitate a comprehensive comparison. Following [49], we adopt a trainable linear layer as the visual projector and prepend a task-specific prompt to the input sentence for each task, such as "vqa: [Q]". We share the PET module for different tasks and perform multi-task learning via unified text generation [49] to acquire cross-task knowledge. We run each experiment with three seeds on image-text tasks and one seed on video-text tasks due to the submission limit of the VALUE benchmark.

Decoder VL-PET$_{\text{large}}$			Params (%)	VQA (%)	GQA (%)	NLVR^2 (%)	COCO (CIDEr)	Avg.
Self	Cross	FF						
✓	✗	✗	4.16	$66.29_{0.11}$	$54.37_{0.61}$	$72.77_{0.12}$	$117.24_{1.07}$	$77.67_{0.38}$
✗	✓	✗	4.16	$66.24_{0.07}$	$54.62_{0.25}$	$73.26_{0.33}$	$118.68_{0.42}$	$\mathbf{78.20}_{0.25}$
✗	✗	✓	4.16	$66.21_{0.08}$	$54.57_{0.35}$	$73.35_{0.24}$	$116.78_{0.35}$	$77.73_{0.14}$
✓	✓	✗	4.74	$66.66_{0.22}$	$55.12_{0.28}$	$73.16_{0.51}$	$116.90_{0.98}$	$77.96_{0.15}$
✓	✗	✓	4.74	$66.31_{0.21}$	$54.95_{0.06}$	$73.11_{0.29}$	$116.14_{0.13}$	$77.62_{0.05}$
✗	✓	✓	4.74	$66.45_{0.24}$	$55.13_{0.16}$	$73.66_{0.23}$	$116.94_{0.49}$	$78.05_{0.04}$
✓	✓	✓	5.31	$66.57_{0.30}$	$54.77_{0.26}$	$73.54_{0.36}$	$115.65_{0.46}$	$77.63_{0.17}$

Table 5. Experiments on where to integrate VL-PET$_{\text{large}}$ into the PLM decoders. (Self, Cross and FF indicate self-attention, cross-attention and feed-forward modules of the PLM decoders. ✓: insert. ✗: not insert. **Bold**: best average performance.)

Method	Params (%)	VQA (%)	GQA (%)	NLVR^2 (%)	COCO (CIDEr)	Avg.
Decoder VL-PET$_{\text{large}}$ (Cross)	4.16	$66.24_{0.07}$	$54.62_{0.25}$	$73.26_{0.33}$	$118.68_{0.42}$	$78.20_{0.25}$
Decoder VL-PET$_{\text{large}}$ (CrossK)	4.16	$63.25_{0.29}$	$53.32_{0.25}$	$67.14_{1.12}$	$114.52_{0.54}$	$74.55_{0.29}$
Decoder VL-PET$_{\text{large}}$ (CrossV)	4.16	$66.17_{0.27}$	$55.11_{0.17}$	$73.43_{0.35}$	$122.03_{0.46}$	$\mathbf{79.18}_{0.14}$

Table 6. Experiments on how to apply VL-PET$_{\text{large}}$ to the cross-attention modules of the PLM decoders. (Cross: the whole cross-attention module. CrossK: the key matrix of Cross. CrossV: the value matrix of Cross. **Bold**: best average performance.)

LN		Params (%)	VQA (%)	GQA (%)	NLVR^2 (%)	COCO (CIDEr)	Avg.
Encoder LN	Decoder LN						
✗	✗	4.14	$66.17_{0.08}$	$54.68_{0.16}$	$72.42_{0.08}$	$121.09_{0.42}$	$78.59_{0.13}$
✓	✗	4.16	$66.17_{0.27}$	$55.11_{0.17}$	$73.43_{0.35}$	$122.03_{0.46}$	$\mathbf{79.18}_{0.14}$
✗	✓	4.16	$66.14_{0.24}$	$55.06_{0.58}$	$73.08_{0.22}$	$120.09_{0.31}$	$78.59_{0.19}$
✓	✓	4.18	$66.23_{0.12}$	$54.60_{0.57}$	$72.80_{0.26}$	$121.18_{0.22}$	$78.70_{0.13}$

Table 7. Effectiveness of layer normalization (LN). (✓: LN is trainable. ✗: LN is frozen. **Bold**: best average performance.)

The average performance of multiple tasks serves as the criterion for evaluating model performance. To measure the efficiency of the models, we report the percentage of trainable parameters, excluding the frozen vision encoder, which is only used for offline visual feature extraction. We provide the implementation details in the Appendix.

4.2. Main Results on Image-Text Tasks

To valid the efficiency, effectiveness and transferability of our VL-PET framework and model-agnostic VL-PET modules, we conduct experiments on image-text tasks with different PLM backbones (i.e., BART-base and T5-base).

Image-Text Tasks with BART-base. Tab. 2 has shown us the performance of full fine-tuning, state-of-the-art PET techniques and four VL-PET instantiations on image-text tasks with BART-base. All of our four VL-PET modules with lightweight PET module designs significantly outperform other PET techniques, as shown in Fig. 1, which demonstrates the efficiency and effectiveness of the proposed VL-PET framework. Specifically, VL-PET$_{\text{large}}$ outperforms VL-Adapter and LoRA in all downstream tasks. VL-PET$_{\text{large}}$ relatively surpasses VL-Adapter by 2.92% with comparable trainable parameters ($4.16\% < 4.18\%$) and LoRA by 3.37% with fewer trainable parameters ($4.16\% < 5.93\%$). VL-PET$_{\text{small}}$, VL-PET$_{\text{middleX}}$ and VL-PET$_{\text{middleY}}$ also surpass VL-Adapter by 1.57%, 1.87% and 1.69% respectively, while utilizing far fewer trainable parameters ($2.98\% < 4.18\%$). We observe that our VL-PET method performs comparable to or even outperform full fine-tuning, while most of the gains can be attributed to improvements in the COCO captioning task. This phenomenon justifies the necessity of our special VL-PET designs (e.g., lightweight PET module designs) for encoders and decoders, which helps to preserve the text generation ability of the pre-trained decoders.

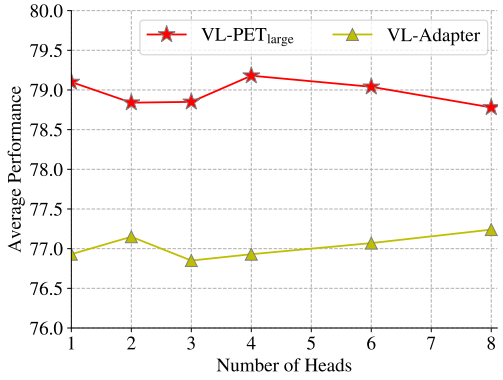

Figure 4. Effectiveness of the multi-head modular modification.

Method	#Params	VQA (%)	GQA (%)	NLVR^2 (%)	COCO (CIDEr)	Avg.
VL-PET$_{\text{large}}$ (BART-base)						
Single-task Learning	584M	$66.13_{0.17}$	$53.68_{0.18}$	$50.36_{1.24}$	$121.63_{0.72}$	$72.95_{0.40}$
Multi-task Learning	146M	$66.17_{0.27}$	$55.11_{0.17}$	$73.43_{0.35}$	$122.03_{0.46}$	$79.18_{0.14}$
VL-PET$_{\text{large}}$ (T5-base)						
Single-task Learning	964M	$66.29_{0.09}$	$54.43_{0.19}$	$60.16_{2.21}$	$122.58_{0.33}$	$75.87_{0.56}$
Multi-task Learning	241M	$66.95_{0.21}$	$56.06_{0.21}$	$73.42_{0.46}$	$121.66_{0.06}$	$79.52_{0.21}$

Table 8. Effectiveness of multi-task learning with different PLMs.

Image-Text Tasks with T5-base. Since the instantiated VL-PET modules are model-agnostic modules, we transfer them to another larger PLM backbone, i.e., T5-base. As shown in Tab. 2, the observed trends of performance improvement of VL-PET modules in T5-base remain consistent with those observed in BART-base. All of four VL-PET modules with lightweight PET module designs significantly outperform other PET techniques. In particular, VL-PET$_{\text{large}}$ outperforms LST, LoRA and VL-Adapter on most downstream tasks with fewer trainable parameters ($7.31\% < 7.46\% < 7.54\% < 7.98\%$), except for slightly lower performance on VQA compared to VL-Adapter. Specifically, VL-PET$_{\text{small}}$, VL-PET$_{\text{middleX}}$, VL-PET$_{\text{middleY}}$ and VL-PET$_{\text{large}}$ surpasses VL-Adapter and LST by 1.92%, 3.07%, 2.95% and 3.41% respectively. They also surpasses LoRA by 5.49%, 6.68%, 6.55% and 7.03% respectively. The performances of the VL-PET modules have shown a significant improvement in a larger PLM. However, other PET techniques do not exhibit a similar improvement and some of them even perform was even worse than their BART-base counterparts. These results demonstrate the effectiveness, efficiency, and transferability of our proposed VL-PET framework.

Method	Params (%)	VQA (%)	GQA (%)	NLVR^2 (%)	COCO (CIDEr)	Avg.
Compacter	2.70	$64.63_{0.09}$	$52.70_{0.24}$	$71.11_{0.35}$	$114.69_{0.42}$	$75.78_{0.21}$
+ $\mathbf{G}_{\text{small}}$ + LW	2.08	$64.14_{0.05}$	$52.84_{0.45}$	$71.04_{0.54}$	$118.77_{0.31}$	$76.70_{0.08}$
+ $\mathbf{G}_{\text{middleX}}$ + LW	2.07	$64.35_{0.10}$	$53.10_{0.73}$	$70.57_{0.44}$	$119.02_{0.37}$	$76.76_{0.17}$
+ $\mathbf{G}_{\text{middleY}}$ + LW	2.07	$64.00_{0.12}$	$52.49_{0.60}$	$70.81_{0.68}$	$117.48_{0.11}$	$76.20_{0.33}$
+ $\mathbf{G}_{\text{large}}$ + LW	3.28	$\mathbf{65.60}_{0.15}$	$\mathbf{54.05}_{0.30}$	$\mathbf{71.66}_{0.17}$	$\mathbf{119.55}_{0.23}$	$\mathbf{77.72}_{0.15}$
VL-Adapter	4.18	$65.76_{0.28}$	$54.16_{0.44}$	$73.19_{0.71}$	$114.61_{0.26}$	$76.93_{0.25}$
+ $\mathbf{G}_{\text{small}}$ + LW	2.98	$65.56_{0.17}$	$54.34_{0.32}$	$71.95_{0.23}$	$\mathbf{119.10}_{0.25}$	$77.74_{0.11}$
+ $\mathbf{G}_{\text{middleX}}$ + LW	2.98	$65.73_{0.12}$	$54.90_{0.10}$	$73.04_{0.50}$	$118.34_{0.91}$	$78.00_{0.31}$
+ $\mathbf{G}_{\text{middleY}}$ + LW	2.98	$65.67_{0.17}$	$54.11_{0.29}$	$73.18_{0.34}$	$117.38_{0.52}$	$77.58_{0.08}$
+ $\mathbf{G}_{\text{large}}$ + LW	4.16	$\mathbf{66.31}_{0.23}$	$\mathbf{55.09}_{0.37}$	$\mathbf{73.46}_{0.37}$	$119.05_{0.83}$	$\mathbf{78.47}_{0.11}$

Table 9. Transferring VL-PET designs to existing PET techniques. Experiments are conducted on image-text tasks with the BART-base backbone. (LW: lightweight designs and trainable encoder LNs. **Bold**: the best result for different PET techniques.)

4.3. Transfer VL-PET Modules to Video-Text Tasks

In Tab. 3, we also test our instantiated VL-PET modules with lightweight PET module designs on video-text tasks. The four VL-PET modules outperform most state-of-the-art PET techniques and attain performance comparable to full fine-tuning with the BART-base backbone. Concretely, VL-PET$_{\text{large}}$ surpasses VL-Adapter by 0.77% with comparable trainable parameters ($3.37\% < 3.39\%$) and LoRA by 5.80% with fewer trainable parameters ($3.37\% < 5.17\%$). VL-PET$_{\text{small}}$, VL-PET$_{\text{middleX}}$ and VL-PET$_{\text{middleY}}$ perform on par with VL-Adapter with fewer trainable parameters ($2.18\% < 3.39\%$) and also outperform LoRA by a large margin. These results reveal the efficiency and effectiveness of our VL-PET framework.

4.4. Ablation Studies

In this section, we conduct ablation studies on image-text tasks with BART-base over three seeds by default. More experiments (e.g., visual projector, scaling factor, task prompt and weight initialization) are provided in the Appendix.

Granularity-controlled Mechanism. In Tab. 4, we study the necessity of the proposed granularity-controlled mechanism. The results indicate that VL-PET modules with granularity control significantly outperform the VL-PET modules without granularity control, which demonstrates the effectiveness of our granularity-controlled mechanism.

Multi-head Modular Modification. We ablate the number of heads for multi-head modular modification of encoder VL-PET$_{\text{large}}$ in Fig. 4 and apply this design to encoder VL-Adapter for comparison. The best number of heads for VL-PET$_{\text{large}}$ and VL-Adapter are 4 and 8, respectively. The superior performance of multi-head over single-head in either VL-PET$_{\text{large}}$ or VL-Adapter indicates the effectiveness and transferability of our multi-head modular modification.

Lightweight PET Module Designs. Tab. 5 shows our exploration of simple yet effective designs. Compared to conventional designs, the results point out that integrating decoder PET modules in the cross-attention modules only is sufficient to achieve the best performance with fewer trainable parameters. Subsequent experiments in Tab. 6 that apply VL-PET to finer-grained PLM modules (i.e., value matrices of cross-attention) further improve the result, indicating the importance of positions of modular modifications.

Layer Normalization (LN). Unlike VL-Adapter [49] which sets all LN as trainable, our experimental results in Tab. 7 indicate that utilizing trainable encoder LNs and frozen decoder LNs is a more effective strategy.

Multi-Task Learning. Multi-task learning fine-tunes a single model for all tasks simultaneously, while single-task learning fine-tunes a single model for each task separately. As shown in Tab. 8, multi-task learning outperforms single-task learning on most tasks with far fewer model parameters for all tasks (BART-base: 146M<584M, T5-base: 241M<964M). In particular, the performance on NLVR^2 under multi-task learning surpasses the one of single-task learning by a large margin, demonstrating that our VL-PET framework can acquire cross-task knowledge under multi-task learning. Therefore, multi-task learning is crucial for enhancing performance and reducing model storage space.

4.5. Applying VL-PET Designs to Existing Methods

In this section, we validate the transferability of some of our VL-PET designs to state-of-the-art PET techniques (e.g., Compacter [21] and VL-Adapter [49]). As described in Sec. 3.2 and Sec. 3.3, we first impose effective control over the modular modifications introduced by these PET techniques. To simplify the validation of lightweight PET module designs, we only retain the decoder PET modules in the cross-attention modules from their conventional de-

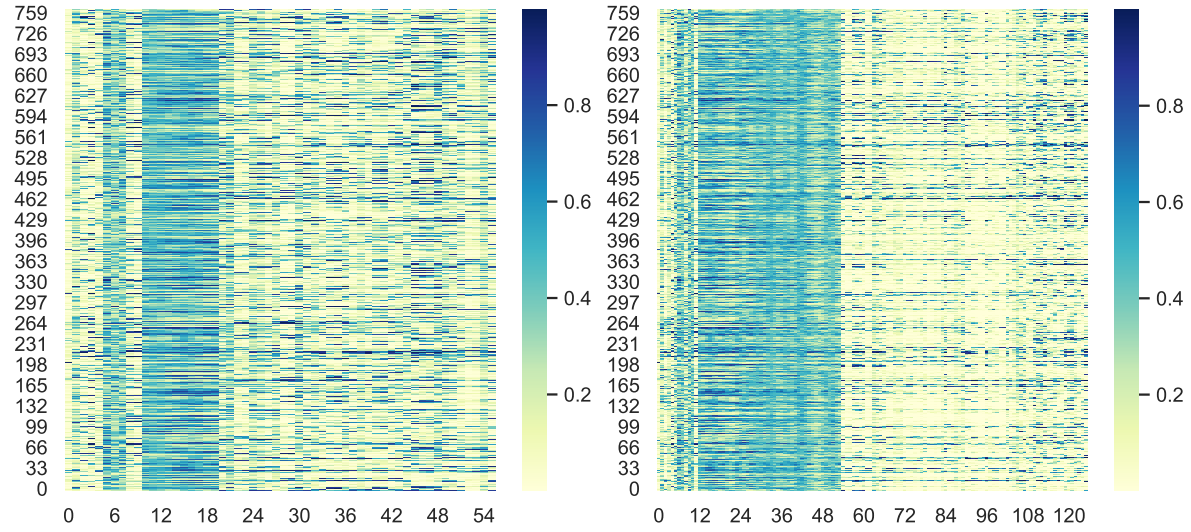

Figure 5. Visualizations of $\mathbf{G}_{\text{large}}$ for two randomly picked inputs from VQAv2 and NLVR^2.

signs. As done in VL-PET, we similarly freeze the PLM backbone, except for the encoder LNs. Results in Tab. 9 demonstrate that applying our VL-PET designs to existing PET techniques leads to significant performance improvements. In particular, Compacter and VL-Adapter with $\mathbf{G}_{\text{middleX}}$ outperform their original versions by 1.29% and 1.39%, respectively, while utilizing fewer trainable parameters ($2.07\% < 2.70\%$ and $2.98\% < 4.18\%$). Compacter and VL-Adapter with $\mathbf{G}_{\text{large}}$ even outperform their original performance by 2.56% and 2.00%, respectively. These results again validate the universality of our VL-PET designs.

4.6. Qualitative Analysis

The granularity-controlled mechanism described in Sec. 3.2 dynamically assigns importance weights to each element in the intermediate hidden states. To gain more insight into how it works, some visualizations are provided in Fig. 5, where we visualize the heatmap of $\mathbf{G}_{\text{large}} \in \mathbb{R}^{N\times d}$ in the first encoder self-attention module of BART-base (hidden dimension d=768). Given two randomly picked inputs from VQAv2 and NLVR^2, $\mathbf{G}_{\text{large}}$ changes dynamically based on the inputs and thus assigns different importance weights to the hidden states. For some text tokens, large weights are densely assigned to almost all of their elements. For other tokens (especially vision tokens), large weights are sparsely distributed on their elements. Such learned weight assignment strategies attest that our granularity-controlled mechanism is a non-trivial method.

5. Conclusion

In this paper, we analyze and tackle some critical issues overlooked by existing PET techniques on VL tasks (e.g., effective control over modular modifications, the encoder-decoder connections and the unique abilities of encoders and decoders). We propose a novel VL-PET framework to effectively control the modular modifications introduced by PET techniques via a granularity-controlled mechanism. Considering different granularity control levels, multi-head modular modifications and lightweight PET module designs, a variety of model-agnostic VL-PET modules can be instantiated from the proposed VL-PET framework for better efficiency and effectiveness trade-offs. Extensive experiments conducted on image-text and video-text tasks demonstrate the efficiency, effectiveness and transferability of our VL-PET framework. Moreover, we validate the universality of our VL-PET designs (e.g., granularity-controlled mechanism and lightweight PET module designs) by transferring them to existing PET techniques, enabling them to achieve significant performance improvements. Although our work focuses on VL tasks, the ideas and designs presented in this work (e.g., granularity-controlled mechanism, multi-head modular modifications and lightweight PET module designs) have the potential to be applied to other domains (e.g., NLP and CV).

Acknowledgments

This research was partially funded by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission (ITC)'s InnoHK. Liwei Wang is a Principal Investigator of CPII under the InnoHK. This work was also partially supported by the UGC under Research Matching Grant Scheme and Direct Grant at CUHK, and Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14206921 of the General Research Fund).

References

[1] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

[3] Hao Chen, Ran Tao, Han Zhang, Yidong Wang, Wei Ye, Jindong Wang, Guosheng Hu, and Marios Savvides. Conv-adapter: Exploring parameter efficient transfer learning for convnets. *CoRR*, abs/2208.07463, 2022.

[4] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *CoRR*, abs/2205.13535, 2022.

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015.

[6] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[7] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *CoRR*, abs/2205.08534, 2022.

[8] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *ICML*, 2021.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[11] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

[12] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *ICLR*, 2022.

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[14] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[15] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

[16] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[17] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

[18] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022.

[19] Shibo Jie and Zhi-Hong Deng. Fact: Factor-tuning for lightweight adaptation on vision transformer. *CoRR*, abs/2212.03145, 2022.

[20] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, 2022.

[21] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In *NeurIPS*, 2021.

[22] Muhammad Uzair Khattak, Hanoona Abdul Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. *CoRR*, abs/2210.03117, 2022.

[23] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021.

[24] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. Tvqa: Localized, compositional video question answering. In *EMNLP*, 2018.

[25] Jie Lei, Licheng Yu, Tamara L. Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*, 2020.

[26] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021.

[27] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.

[28] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

[29] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+language omni-representation pre-training. In *EMNLP*, 2020.

[30] Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, Yu Cheng, Luowei Zhou, Xin Eric Wang, William Yang Wang, et al. Value: A multi-task benchmark for video-and-language understanding evaluation. In *NeurIPS*, 2021.

[31] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021.

[32] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *CoRR*, abs/2205.05638, 2022.

[33] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[34] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *arXiv preprint arXiv:2111.09883*, 2021.

[35] 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 *ICCV*, 2021.

[36] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, 2022.

[37] Haoyu Lu, Mingyu Ding, Yuqi Huo, Guoxing Yang, Zhiwu Lu, Masayoshi Tomizuka, and Wei Zhan. Uniadapter: Unified parameter-efficient transfer learning for cross-modal modeling. *CoRR*, abs/2302.06605, 2023.

[38] Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *ACL*, 2021.

[39] Oscar Mañas, Pau Rodríguez, Saba Ahmadi, Aida Nematzadeh, Yash Goyal, and Aishwarya Agrawal. MAPL: parameter-efficient adaptation of unimodal pre-trained models for vision-language few-shot prompting. *CoRR*, abs/2210.07179, 2022.

[40] Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. Unipelt: A unified framework for parameter-efficient language model tuning. In *ACL*, 2022.

[41] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning for action recognition. *CoRR*, abs/2206.13559, 2022.

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[43] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[44] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[45] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[46] Sebastian Ruder, Jonas Pfeiffer, and Ivan Vulić. Modular and parameter-efficient fine-tuning for nlp models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 23–29, 2022.

[47] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *ACL*, 2019.

[48] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. LST: ladder side-tuning for parameter and memory efficient transfer learning. *CoRR*, abs/2206.06522, 2022.

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

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[51] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *CoRR*, abs/2208.10442, 2022.

[52] Hao Yang, Junyang Lin, An Yang, Peng Wang, Chang Zhou, and Hongxia Yang. Prompt tuning for generative multimodal pretrained models. *CoRR*, abs/2208.02532, 2022.

[53] Bruce X. B. Yu, Jianlong Chang, Lingbo Liu, Qi Tian, and Chang Wen Chen. Towards a unified view on visual parameter-efficient transfer learning. *CoRR*, abs/2210.00788, 2022.

[54] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*, 2022.

[55] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022.

[56] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *Int. J. Comput. Vis.*, 130(9):2337–2348, 2022.

[57] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018.