Preserving Pre-trained Representation Space: On Effectiveness of Prefix-tuning for Large Multi-modal Models

Donghoon Kim, Gusang Lee, Kyuhong Shim, Byonghyo Shim
Dept. of Electrical and Computer Engineering, Seoul National University
{dhkim,gslee,khshim,bshim}@islab.snu.ac.kr

Abstract

Recently, we have observed that Large Multi-modal Models (LMMs) are revolutionizing the way machines interact with the world, unlocking new possibilities across various multi-modal applications. To adapt LMMs for downstream tasks, parameter-efficient fine-tuning (PEFT) which only trains additional prefix tokens or modules, has gained popularity. Nevertheless, there has been little analysis of how PEFT works in LMMs. In this paper, we delve into the strengths and weaknesses of each tuning strategy, shifting the focus from the efficiency typically associated with these approaches. We first discover that model parameter tuning methods such as LoRA and Adapters, distort the feature representation space learned during pre-training, limiting the full utilization of pre-trained knowledge. We also demonstrate that prefix-tuning excels at preserving the representation space, despite of its lower performance on downstream tasks. These findings suggest a simple two-step PEFT strategy called **Prefix-Tuned PEFT (PT-PEFT)**, which successively performs prefix-tuning and then other PEFT (i.e., Adapter, LoRA), combines the benefits of both. Experimental results show that PT-PEFT not only improves performance in image captioning and visual question answering compared to vanilla PEFT methods but also helps preserve the representation space of the four pre-trained models.

1 Introduction

Understanding the visual scene and expressing it with a natural language are two distinct tasks yet the human brain can comprehensively handle both without difficulty. Large multi-modal models (LMMs) mimic such capability by training a deep neural network such that it learns semantically meaningful connections between vision and language from a large number of image-text pairs (Li et al., 2020b; Zhang et al., 2021b; Wang et al., 2022b; Radford et al., 2021). Recently, LMMs have become widely used due to their broad range of applications, including chatbot, robot control, and video generation. (Ouyang et al., 2022; Brohan et al., 2023; Ramesh et al., 2022).

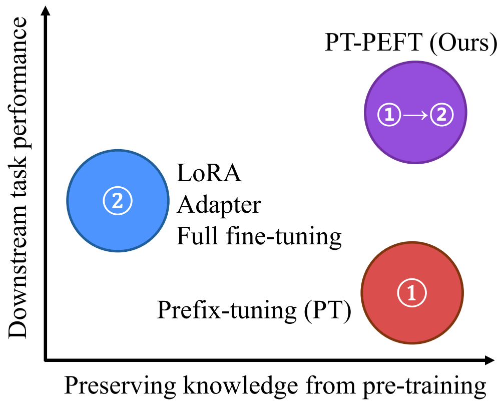


Figure 1: Advantages of the proposed PT-PEFT, which performs 1) prefix-tuning and 2) fine-tuning (i.e., parameter-efficient or full fine-tuning) sequentially.

In the *pre-training*, LMMs are trained to predict the masked words or next words from the image-text pair (Li et al., 2023; Alayrac et al., 2022; Wang et al., 2022a). In the second step called *fine-tuning*, the pre-trained LMMs are tailored to the specific downstream task. It has been shown that fine-tuning provides superior performance in performing various downstream tasks such as image captioning (IC), visual question answering (VQA), and image-text retrieval (Li et al., 2023; Wang et al., 2022a,b; Zhang et al., 2021b). However, fine-tuning often suffers from the loss of generalization capability learned from the pre-training (Sun et al., 2015; Brown et al., 2020a). Since the task-specific dataset is far smaller than

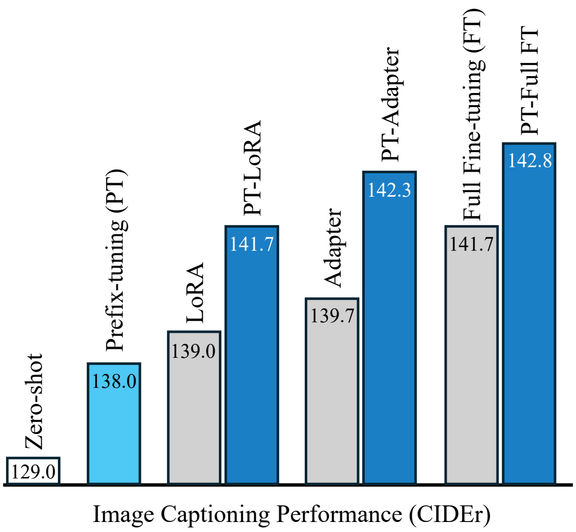


Figure 2: Performance of different task adaptation methods on COCO image captioning dataset. The proposed method (PT-) consistently improves performance when combined with other methods.

the pre-training unlabeled dataset, the pre-trained model can be easily overfitted to the small-sized downstream task dataset, leading to degraded performance on diverse datasets (Kumar et al., 2022). To address the problem, various approaches have been suggested over the years. In prompt-based approaches, human-engineered sentences or trainable continuous embedding vectors are processed alongside the input without modifying the model parameters (Li and Liang, 2021; Liu et al., 2021; Tam et al., 2022; Lester et al., 2021). In knowledge distillation-based fine-tuning approaches, the model is fine-tuned using the distance between the distribution of the pre-trained model and the fine-tuned model (Xu et al., 2020; Sanh et al., 2019; Boschini et al., 2022). The common wisdom behind these approaches is to minimize the modification of the pre-trained model parameters.

One drawback of full model fine-tuning is the huge computational burden caused by the model parameters update. In an effort to reduce the huge training cost, a variety of parameter-efficient fine-tuning (PEFT) techniques have been proposed (Li and Liang, 2021; Houlsby et al., 2019; Hu et al., 2022; He et al., 2021). In this approach, instead of applying full fine-tuning, only a small set of additional modules (e.g., prefix, Adapter, LoRA) is trained. This approach is especially beneficial for training the large pre-trained model like GPT (Brown et al., 2020b), T5 (Raffel et al., 2020), and Llama (Touvron et al., 2023).

Training efficiency is a well-known advantage of prefix-tuning, but what distinguishes prefix-tuning from other PEFT methods is that prefix-tuning does not modify the model's parameters at all, leaving the representation space unchanged. To investigate the changes in the representation space, we analyze the feature representation matrices using singular value decomposition. Notably, we observe that the representation space of a fine-tuned model (in IC and VQA) only utilizes a limited set of effective basis vectors (60 % of the pre-trained model) to express the output compared to that of the pre-trained model, limiting its ability to fully take the advantages gained from pre-training (see Figure 4). In contrast, we discover that all the basis vectors are utilized for the prefix-tuned model, indicating that prefix-tuning effectively preserves the inherited representation space from pre-training. These results suggest that prefix-tuning may address the poor generalization observed in fine-tuning.

While prefix-tuning is effective in preserving pre-trained knowledge, the efficacy of this approach has been somewhat questionable since the reported evaluation results were not conclusive. Some studies claim that the prefix-tuning performs comparable to the model parameter-tuning (e.g., full fine-tuning, LoRA, Adapter), while others argue that the prefix-tuning struggles in the training of relatively small-sized language models (Liu et al., 2021; Tam et al., 2022).

In this paper, we suggest a simple yet effective strategy for grafting two seemingly distinct approaches. The method, henceforth referred to as Prefix-Tuned PEFT (PT-PEFT), performs the prefix-tuning and the model parameter-tuning *sequentially* to combine the merits of both. The key feature of this approach is to preserve the pre-trained feature space through prefix-tuning and then refine the model parameters through other PEFT methods. Intuitively, this approach resembles a language model learning a new task using prompt sentences such as "I will provide example sentences describing the given pictures in a news article style. Please generate the caption for the given images with such style." By providing context suitable for the new task, the model begins

Image	Image Id	Zero-shot	Prefix-tuning	Fine-tuning
	107257	"a stove sitting on the side of the road with a sign that says \"become your dream\" written on it"	"a stove on the side of the road with the words \"become your dream\" written on it"	"an old stove sitting on the side of the road"
	407180	"birds perched on a ledge overlooking a body of water with a city skyline in the background"	“seagulls perched on the edge of a building overlooking a body of water"	"a group of birds perched on a ledge overlooking a body of water"
	518937	"an outdoor patio with two umbrellas and a person sitting under one of the umbrellas"	"a person sitting at a table under a red and yellow umbrella"	"a person sitting at a table under a red umbrella"
	448078	"a car driving down a street with traffic lights and buildings in the background"	"a jeep driving down the street in front of a building"	"a view of a city street from inside a car"

Figure 3: Qualitative image captioning results of zero-shot learning, prefix-tuned, and fine-tuned models. Although fine-tuning provides accurate answers, its results often ignore visual details compared to the other two.

training with high adaptability, allowing for faster convergence and minimal changes to the weights of the pre-trained model.

In our experiments, we show that applying prefix-tuning before LoRA, Adapter, and even full fine-tuning consistently improves the task performance in four IC/VQA public datasets on four pre-trained LMMs including BLIP (Li et al., 2022), BLIP-2 (Li et al., 2023), OFA (Wang et al., 2022a) and VINVL (Zhang et al., 2021b).

Our contributions are as follows:

- We establish the connection between representation space and performance through rank-based analysis. We qualitatively and qualitatively illustrate the effects of representation space collapse.
- We reveal that prefix-tuning differs significantly from other fine-tuning techniques like LoRA, Adapter, and full fine-tuning, highlighting the unique advantage of prefix-tuning in preserving the integrity of the pre-trained knowledge.
- We propose PT-PEFT, a method that sequentially performs prefix-tuning followed by another fine-tuning technique, maximizing the utilization of pre-trained knowledge in LMMs. PT-PEFT outperforms standard fine-tuning methods in image captioning and VQA tasks across four different LMMs.

2 Representation Space Collapse Causes the Loss of Generalization Capabilities

2.1 Zero-shot Sometimes Performs Better than Fine-tune

Model parameter tuning generally shows superior performance over prefix-tuning, however, the full fine-tuned model sometimes generates even worse answer comparing to zero-shot generation for certain samples. Figure 3 presents a qualitative comparison between zero-shot inference, full fine-tuning, and prefix-tuning on IC and VQA tasks. For IC tasks, we find that prefix-tuning capture more detailed descriptions of objects compared to full fine-tuning. Although the IC output from fine-tuning is technically correct, the captions generated through prefix-tuning are richer and more natural. Similarly, for VQA tasks, the Top-5 answers from prefix-tuning are more relevant to the given questions, whereas answers from fine-tuning are often nonsensical or less likely to be correct. This phenomenon arises from the downstream dataset, which consists of a limited range of objects and object's attributes compared to the diverse range encountered during pre-training. Intuitively, fitting the fine-tuning data leads to forgetting word and image representations for other objects and attributes that exist in the pre-trained dataset but not in the downstream dataset (a.k.a., catastrophic forgetting) (Rebuffi et al., 2017; Kalajdzievski, 2024).

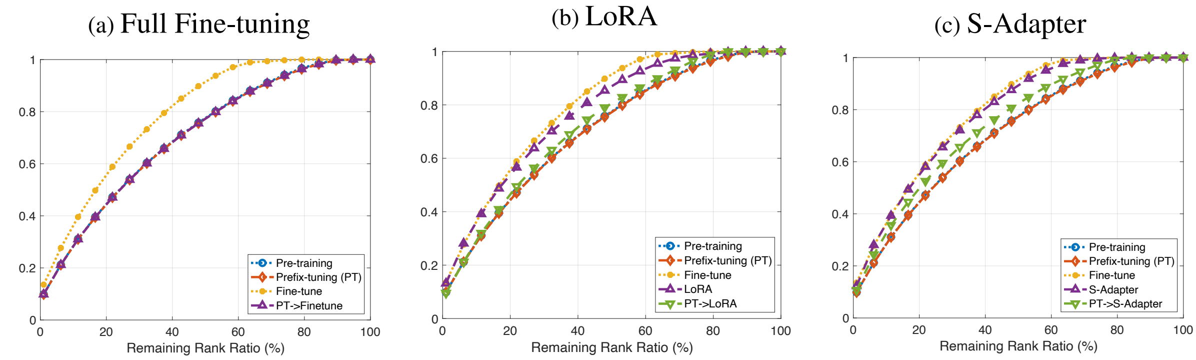


Figure 4: Accumulated and normalized singular values of feature vectors extracted from the last layer of BLIP-2 (OPT-2.7 B). A more concave graph indicates that the singular values are more concentrated, implying the narrower representation space.

	Pre-training	Fine-tuning	Prefix-tuning	S-Adapter	P-Adapter	LoRA	PT→S-Adapter	PT→P-Adapter	PT→LoRA	PT → Finetuning
VINVL	50.2 %	30.0 %	50.2 %	-	-	-	-	-	-	**50.2 %**
BLIP-2	68.2 %	47.0 %	68.2 %	53.0 %	53.7 %	52.0 %	63.5 %	58.4 %	63.5 %	**68.2 %**

Table 1: Effective rank of representation space of various fine-tuning techniques. Note that the effective rank is defined as the remaining rank ratio at which the accumulated singular values equal to 0.9 in Figure 4.

2.2 Relationship Between Semantic Richness and Representation Space

Forgetting the words and image representation further contributes to the simplification of the representation space by rendering the unused vectors become unnecessary. To this end, we argue that the information of the representation matrix is closely related to their rank, following previous studies on the representation space (Zhang et al., 2021a; Bansal et al., 2018; Swaminathan et al., 2020). For instance, low-rank compression methods intentionally reduce the rank of features to distill essential information, such as object class (Sainath et al., 2013; Swaminathan et al., 2020). This rank reduction phenomenon, which we refer to as "representation collapse", can result in a degradation in capturing semantically rich details across a wide range of objects and their attributes. This degradation in capturing semantically rich details potentially harms the generalization ability for downstream tasks.

2.3 Empirical Analysis on Representation Space Collapse

Representation Space Analysis via SVD To quantitatively measure the representation collapse in different model adaptation methods, we apply a singular value decomposition (SVD) on the representation matrices. SVD allows us to quantitatively analyze the average number of basis vectors used to represent a single text or image. For SVD, we use the output (i.e., activation) matrix of the last layer from a single sentence/image. Specifically, LMM processes the text input $x_{txt} = [w_{sos}, w_1, ..., w_N, w_{eos}]$, yielding a sequence of output embedding vectors $\mathrm{F}_{txt} = \mathrm{LMM}(x_{txt})$:

$$\mathrm{F}_{txt} := [\mathbf{f}_{txt}^{sos}, \mathbf{f}_{txt}^{w_1}, ..., \mathbf{f}_{txt}^{w_N}, \mathbf{f}_{txt}^{eos},]. \tag{1}$$

For each sentence embedding matrix F_{txt}, we perform SVD to get the singular values (i.e., the diagonal elements of Σ):

$$\mathrm{F}_{txt} = \mathrm{U}\Sigma\mathrm{V}^{\mathrm{T}}. \tag{2}$$

We sort the singular values $\mathbf{s} = [\sigma_1, ..., \sigma_M]$ in descending order and normalize such that sum of all singular values equals one:

$$\hat{\mathbf{s}} = \frac{1}{\sum_{i=1}^{M} \sigma_i}[\sigma_1, ..., \sigma_M]. \tag{3}$$

After computing singular values on a per-image/per-sentence basis, we average them across the K samples in the dataset:

$$\hat{\mathbf{s}}_{avg} = \frac{1}{K}\sum_{i=1}^{K} \hat{\mathbf{s}}_k. \tag{4}$$

We the accumulated sum of the elements in $\hat{\mathbf{s}}_{avg}$:

$$\mathbf{y} = \left[\hat{\sigma}_{avg,0}, ..., \sum_{j=1}^{i} \hat{\sigma}_{avg,j}, ..., \sum_{j=1}^{M} \hat{\sigma}_{avg,j} \right]. \quad (5)$$

The final $\mathbf{y}$ is ploted in Figure 4 for each model and training technique.

Comparison Between Various Fine-tuning Methods Figure. 4 presents the cumulative sum of singular values of feature vectors extracted from different models. Specifically, we compare the rank of an image and text features extracted from the three models (pre-trained, fine-tuned, and prefix-tuned). The naive fine-tuned model shows the fastest saturation towards the top (see red line in Figure. 4), meaning that most singular values are close to zero (i.e., $\sum_{i=1}^{k} \sigma_i \approx 1$ for small k). This in turn means that the effective rank of the feature matrix extracted from the fine-tuned model is much lower than that of the pre-trained model. In addition, as an accuracy shown in the Figure. 4 legend, the curvature of the singular value plot is also highly correlated to the final performance (e.g., CIDEr, Accuracy) (Daneshmand et al., 2020; Dong et al., 2021). As shown in Table 1, LoRA-tuned and fine-tuned models utilize only 60% basis vectors from the pre-trained representation space, while prefix-tuning utilizes almost all the basis vectors.

3 Prefix-Tuned Parameter-Efficient Fine Tuning (PT-PEFT)

Prefix Implementation Prefix embedding vectors are first processed through the prefix encoder, following standard practices in prefix-tuning (Li and Liang, 2021) (see Appendix for details). The processed prefixes are then concatenated with text and/or image tokens to form the input to the LMMs, as shown in Figure 5. The green boxes in the Figure represent learnable prefix embeddings (tokens) used during the prefix-tuning stage.

Two-stage Optimization As described in the previous Section, we use a two-stage approach: prefix-tuning followed by fine-tuning. During the prefix-tuning step, we only train the prefix embeddings and prefix encoder, keeping the other parameters of LMMs frozen. In the subsequent fine-tuning step (either PEFT or full fine-tuning),

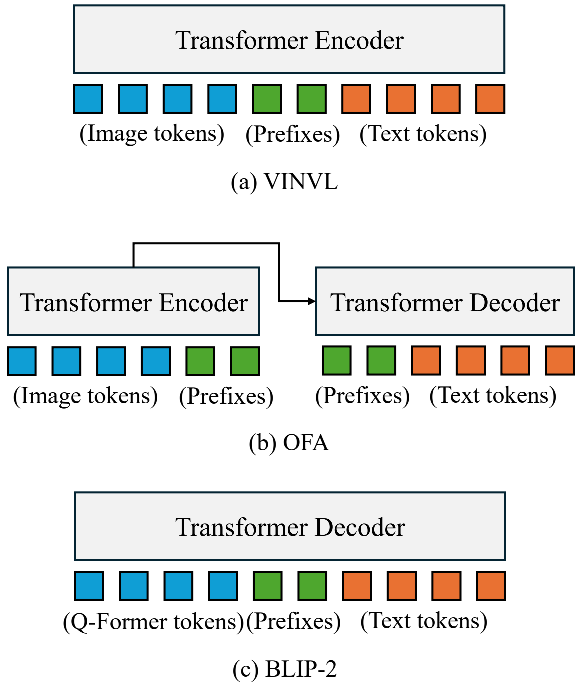


Figure 5: Visualization of where the prefixes are inserted for different LMMs. Proposed method can be applied for general Transformer-based architectures.

we train the corresponding parameters including prefixes, to further adapt the model.

4 Experiments

4.1 Setup

Model To demonstrate the generalization capability of our method, we use various pre-trained LMMs with different architectures and sizes. Specifically, we conduct experiments on VINVL-BASE/LARGE (Zhang et al., 2021b), OFA-BASE (Wang et al., 2022a), BLIP (Li et al., 2022), an BLIP-2 (Li et al., 2023) models.

Dataset We evaluate image captioning (IC) task performance on MS-COCO (Lin et al., 2014) and Flickr30k (Plummer et al., 2015) datasets. For the visual question-answering (VQA) task, we use the VQAv2 (Antol et al., 2015) dataset.

Fine-tuning Methods We take pre-trained LMMs and compare different fine-tuning methods. These include Prefix-tuning (Prefix), LoRA, Parallel-Adapter (P-Adapter), and Sequential-Adapter (S-Adapter) (Hu et al., 2023), and also the full fine-tuning (Full-FT). Adapters usually include

	#Trainable Params	COCO IC			Flickr30k IC			VQAv2	
		B4	C	S	B4	C	S	test-dev	test-std
		OFA$_{\text{BASE}}$ (Wang et al., 2022a)							
Prefix-tuning	0.15 %	35.2	115.6	19.3	27.0	61.4	16.5	72.9	73.2
S-Adapter	3.10%	35.6	119.7	20.9	27.4	62.1	16.8	73.1	73.4
S-Adapter → Prefix	3.15%	38.2	128.2	21.6	27.6	64.8	17.3	73.9	74.1
Prefix → S-Adapter	3.15%	**39.0**	**130.7**	**22.5**	**29.2**	**68.3**	**17.3**	**74.3**	**74.4**
P-Adapter	3.08%	36.8	123.7	21.3	28.5	64.4	17.0	73.4	73.8
P-Adapter → Prefix	3.12 %	38.4	129.7	21.7	28.8	67.2	17.9	74.0	74.2
Prefix → P-Adapter	3.12 %	**39.7**	**132.8**	**23.4**	**31.1**	**73.6**	**18.7**	**75.6**	**75.7**
LoRA	0.26 %	35.3	117.4	19.5	24.7	52.4	15.2	50.1	50.3
LoRA → Prefix	0.45 %	36.6	122.0	21.2	28.5	66.2	17.5	70.9	71.1
Prefix → LoRA	0.45 %	**39.2**	**131.6**	**23.1**	**30.5**	**71.6**	**18.0**	**74.6**	**74.9**
Full fine-tuning	100 %	38.6	127.5	22.8	32.2	74.1	18.5	75.7	75.8
		BLIP-2$_{\text{ViT-g + OPT 2.7B}}$ (Li et al., 2023)							
Prefix-tuning	0.20 %	41.0	138.0	24.9	34.6	92.3	20.6	30.1	29.8
S-Adapter	4.32 %	40.4	140.0	25.0	34.4	93.8	22.6	51.8	52.4
S-Adapter → Prefix	4.52 %	40.7	139.8	24.8	34.9	93.8	22.7	53.2	54.3
Prefix → S-Adapter	4.52 %	**41.0**	**140.6**	**25.0**	**35.6**	**95.4**	**23.4**	**54.3**	**54.4**
P-Adapter	3.23 %	40.1	139.0	24.9	33.6	90.4	22.3	53.1	50.4
P-Adapter → Prefix	3.43 %	40.6	140.6	24.9	35.0	94.1	23.0	53.2	53.7
Prefix → P-Adapter	3.43 %	**41.0**	**140.6**	**25.2**	**35.1**	**95.1**	**23.4**	**53.2**	**54.3**
LoRA	0.34 %	40.3	139.0	25.1	35.2	94.4	22.5	43.8	44.4
LoRA → Prefix	0.54 %	40.6	139.3	25.0	35.7	95.9	23.0	53.2	54.3
Prefix → LoRA	0.54 %	**41.2**	**140.6**	**25.2**	**36.1**	**97.0**	**23.3**	**52.2**	**52.3**
Full fine-tuning	100 %	41.1	141.7	25.0	35.9	97.5	27.6	74.9	74.7

Table 2: Performance comparison between PEFT and our PT-PEFT, applying prefix-tuning followed by other PEFT. B4, C, Simplies BLEU-4, CIDEr, and SPICE scores, respectively.

multi-layer modules, so they generally equip more trainable parameters than LoRA. Prefix-tuning uses the smallest number of trainable parameters among all. Note that our PT-PEFT can be applied to all methods, with prefix-tuning used before other fine-tuning methods as our key innovation.

Additional Details We carefully designed settings for each model and method to achieve the best performance. For more details about the models, datasets, and hyper-parameters, please refer to Appendix B.

4.2 Downstream Task Performance

Prefix-tuned PEFT Table 2 shows the performance of various task adaptation methods, applied to OFA-BASE and BLIP-2 models. Our proposed PT-PEFT consistently outperforms standard PEFT methods across all 8 metrics. PT-PEFT even surpasses full fine-tuning, with a 0.2p/0.1p in BLEU-4 metric for Flickr30k/COCO, along with a 0.2p improvement in SPICE score. Additionally, the results show that applying PEFT before prefix-tuning (i.e., reversing the order) is considerably less effective than PT-PEFT, though it still performs better than not using prefix-tuning at all.

Prefix-tuned Full Fine-tuning Tables 3 and 4 compare prefix-tuning, full fine-tuning, and the sequential combination of both (ours). To ensure the reliability of our results, we conducted three separate runs with different random seeds and reported the mean and standard deviation obtained from these runs. Notably, the standard deviation of the scores is significantly smaller than the improvements over the baseline models. Compared to the full fine-tuning, our prefix-tuned full fine-tuning achieves approximately an 11% increase in the BLEU-4, a 16% increase in SPICE, and a noteworthy 21% improvement in CIDEr. These results highlight the effectiveness of our method, demonstrating that prefix-tuning can help preserve pre-trained knowledge and improve performance

	COCO Image Captioning			Flickr-30k Image Captioning		
	BLEU-4	CIDEr	SPICE	BLEU-4	CIDEr	SPICE
			$\text{VINVL}_{\text{BASE}}$			
Prefix-tuning	37.3	122.5	22.2	28.7	65.5	16.9
Full fine-tuning	40.4	137.2	24.5	33.8	85.5	21.1
Prefix → Full-FT	**41.2 ± 0.08**	**141.1 ± 0.10**	**25.0 ± 0.04**	**35.6 ± 0.13**	**89.7 ± 0.36**	**21.5 ± 0.10**
			$\text{VINVL}_{\text{LARGE}}$			
Prefix-tuning	38.5	128.2	23.2	31.9	72.0	18.3
Full fine-tuning	41.0	139.6	24.8	34.3	85.2	21.1
Prefix → Full-FT	**41.4 ± 0.06**	**141.1 ± 0.12**	**24.9 ± 0.07**	**35.8 ± 0.59**	**89.8 ± 0.14**	**21.9 ± 0.04**
			OFA_{BASE}			
Zero-shot	18.2	62.3	14.8	15.3	23.2	12.1
Prefix-tuning	35.2	115.6	19.3	27.0	61.4	16.5
Full fine-tuning	38.6	127.5	22.8	32.2	74.1	18.9
Prefix → Full-FT	**41.4 ± 0.02**	**136.4 ± 0.16**	**24.3 ± 0.11**	**35.8 ± 0.24**	**89.8 ± 0.21**	**21.9 ± 0.07**
			$\text{BLIP-2}_{\text{ViT-g + OPT 2.7B}}$			
Zero-shot	39.7	129.0	22.6	29.5	74.5	16.8
Prefix-tuning	40.0	138.0	24.9	34.6	92.3	20.6
Full fine-tuning	41.1	141.7	25.0	35.9	97.5	**27.6**
Prefix → Full-FT	**41.8 ± 0.11**	**142.8 ± 0.07**	**25.2 ± 0.04**	**36.5 ± 0.09**	**98.3 ± 0.19**	23.6 ± 0.30

Table 3: Image captioning performance comparison between prefix-tuning, full fine-tuning and ours.

	VQAv2	
	test-std	test-dev
	$\text{VINVL}_{\text{BASE}}$	
Linear-probing	72.7	72.6
Prefix-tuning	73.8	73.4
Full fine-tuning	74.1	74.4
Prefix → Full-FT	**76.2 ± 0.04**	**76.2 ± 0.08**
	$\text{VINVL}_{\text{LARGE}}$	
Linear-probing	73.3	73.7
Prefix-tuning	75.0	74.9
Full fine-tuning	76.5	76.6
Prefix → Full-FT	**77.0 ± 0.04**	**77.9 ± 0.02**
	OFA_{BASE}	
Zero-shot	25.9	25.8
Prefix-tuning	73.2	72.9
Full fine-tuning	75.8	75.7
Prefix → Full-FT	**76.8 ± 0.04**	**76.6 ± 0.04**
	$\text{BLIP}_{\text{LARGE}}$	
Zero-shot	5.0	5.2
Prefix-tuning	30.1	29.8
Full fine-tuning	74.9	74.7
Prefix → Full-FT	**77.0 ± 0.07**	**77.9 ± 0.03**

Table 4: VQAv2 performance comparison.

(a) Performance of the sequential-tuned model.

	COCO IC valid			VQAv2 valid	
	B4	C	S	Acc1	Acc5
w/Prefix	41.3	139.3	24.6	75.2	93.3
-Prefix	22.9	75.0	15.3	36.5	72.6
-Prefix +Noise	25.1	82.9	16.2	31.2	61.4

(b) Performance of the parallel-tuned model.

	COCO IC valid			VQAv2 valid	
	B4	C	S	Acc1	Acc5
w/Prefix	41.0	138.0	24.3	71.6	91.9
-Prefix	23.1	74.3	15.1	72.2	91.7
-Prefix +Noise	23.5	76.8	15.5	62.2	86.6

Table 5: Comparison of (a) sequential and (b) parallel tuning. Unlike PT-PEFT, parallel tuning applies prefix-tuning and fine-tuning together. For noise addition experiments (third rows), we replace learned prefixes with random noise during inference.

in both PEFT and full fine-tuning scenarios.

5 Analysis & Discussion

5.1 Preserving Representation Space

Figure 4 visualizes the accumulated singular values, as described in Section 2.3. The saturation curves for the pre-trained, prefix-tuned, and PT-PEFT models are almost identical, implying that the effective rank is preserved. In contrast, LoRA, Adapter, and full fine-tuning methods show more concave curves, indicating a narrower representation space.

5.2 Ablation Study

Sequential vs. Parallel Instead of sequentially applying prefix-tuning and then fine-tuning, one may consider using both methods together in par-

Model	#Epochs		COCO Image Captioning		
	PT	FT	BLEU-4	CIDEr	SPICE
M1	3	7	35.3	114.2	18.8
M2	5	5	40.2	129.6	23.5
M3	7	3	41.4	136.4	24.3

Table 6: Ablation study on the number of epochs for prefix-tuning (PT) and fine-tuning (FT) stages.

allel. We call this variant *parallel-tuning* and compare its performance to our sequential training. Table 5 (a) and (b) present the downstream task performance of parallel tuning and ours, respectively. The result shows that parallel-tuning performs worse than PT-PEFT in all cases.

To further investigate how parallel-tuning affects the effectiveness of the prefix, we distort the trained prefixes and observe the performance change. Table 5(b) shows that for the parallel-tuned model, even without prefixes, VQA accuracy is almost preserved, meaning that the prefix does not contribute to performance. This finding is further emphasized when replacing the trained prefix with random noise; accuracy only slightly decreases, implying that the prefixes are not very powerful. In contrast, when using prefix tuning first (Table 5(a)), removing prefixes severely hurts the accuracy, showing that they actively contribute to the performance.

Ratio of Each Stage We conduct experiments to find the best number of training steps for the prefix-tuning and fine-tuning stages. As shown in Table 6, we found that prefix-tuning requires a sufficiently long iteration for optimal performance. Within the same training budget, the model achies better performance with fewer fine-tuning epochs if sufficient prefix-tuning precedes.

5.3 Intuitive Explanation of PT-PEFT

Based on our analysis, we conclude that prefix-tuning and other fine-tuning method contributes to the adaptation in different ways. By sequentially performing prefix-tuning and parameter fine-tuning, the model first encodes the representation space as prefix tokens that align with the pre-trained space. This is because the original model parameters remain unchanged during prefix-tuning, so the learned knowledge is not damaged. Once such context is established, the subsequent fine-tuning process can effectively avoid the representation collapse, as the prefixes provide a foundation for a rich representation space.

5.4 Prior Works in Language Domain

Here, we highlight how our work significantly differs from recent studies that combine two fine-tuning techniques in the language domain. The original LoRA paper reported that combining LoRA with Prefix-tuning could improve performance (Appendix E of the paper (Hu et al., 2022)). However, their combination used a "parallel-tuning" approach, in contrast to our "sequential-tuning" approach. In addition, they utilized a much larger number of trainable parameters, making it an unfair comparison between LoRA alone and LoRA with Prefix-tuning.

Around the same time as our work, ProMot (Wang et al., 2024) also suggested using prefix-tuning before model parameter tuning in a sequential manner. They also reported significant performance improvements, which is consistent with our findings. However, our work is very distinct in two key perspectives.

First, our experiments focus on LMMs, demonstrating the effectiveness of PT-PEFT across various vision-language tasks and Transformer-based model architectures. Second, our analyses show that the primary reason for performance gain comes from the preservation of learned knowledge during pre-training, as revealed by our systematic investigation of the effective rank of embeddings. This sets our work apart and highlights the uniqueness of our PT-PEFT.

6 Conclusion

In this paper, we discovered that fine-tuning methods including LoRA, Adapter, and full fine-tuning could cause the loss of learned knowledge from the pre-training stage. We quantified this loss in representation space using a novel rank-bases analysis and identified that prefix-tuning does not result in this critical loss. Based on these findings, we proposed a two-step strategy, PT-PEFT, which first performs prefix-tuning and then applies other fine-tuning methods. Our experiments showed that PT-PEFT not only preserves the representation space preservation but also improves downstream task performance.

7 Limitations

The proposed PT-PEFT can take advantage of both prefix-tuning and fine-tuning. However, there are two practical limitations. Firstly, it leads to an increased computational cost during inference due to the longer input sequence. Managing this increased computational cost in prefix-tuning may become challenging, especially when the portion of prefixes in the total number of input tokens. It's worth noting that the performance gains tend to plateau at around 16 prefixes, which doesn't significantly exacerbate the computational cost (see Appendix C prefix length ablation study). Secondly, we manually determine the best-performing hyper-parameters, such as prefix length, learning rates, and training iterations. We did our best to find the best set for a fair comparison; however, we are aware that such a manual hyperparameter tuning process can be cumbersome, especially when applying our technique to new tasks, datasets, or models.

8 Ethical Statement

In our paper, we analyze various fine-tuning strategies to identify methods for preserving pre-trained knowledge during the fine-tuning process. Rather than having potential risks, we believe that our research can serve as a solution to address ethical issues related to data corruption and safety control in current AI systems. For instance, even if the model is fine-tuned with data corrupted by hacking, our technique can offer robustness to such data corruption by preserving the model's representation space. Our work can be also beneficial for not forgetting the safety guardrails learned during pre-training or instruction tuning. We'd like to note that this representation-preserving have not been studied much in VL models, regardless of the increasing interest on VL applications.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. 2018. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31.

Matteo Boschini, Lorenzo Bonicelli, Angelo Porrello, Giovanni Bellitto, Matteo Pennisi, Simone Palazzo, Concetto Spampinato, and Simone Calderara. 2022. Transfer without forgetting. In *European Conference on Computer Vision*, pages 692–709. Springer.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hadi Daneshmand, Jonas Kohler, Francis Bach, Thomas Hofmann, and Aurelien Lucchi. 2020. Batch normalization provably avoids ranks collapse for randomly initialised deep networks. *Advances in Neural Information Processing Systems*, 33:18387–18398.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. 2021. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, pages 2793–2803. PMLR.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*.

Damjan Kalajdzievski. 2024. Scaling laws for forgetting when fine-tuning large language models. *arXiv preprint arXiv:2401.05605*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *CoRR*, abs/2301.12597.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2020a. What does bert with vision look at? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5265–5275.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020b. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

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

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024.

Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. 2013. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6655–6659. IEEE.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Shiliang Sun, Honglei Shi, and Yuanbin Wu. 2015. A survey of multi-source domain adaptation. *Information Fusion*, 24:84–92.

Sridhar Swaminathan, Deepak Garg, Rajkumar Kannan, and Frederic Andres. 2020. Sparse low rank factorization for deep neural network compression. *Neurocomputing*, 398:185–196.

Weng Lam Tam, Xiao Liu, Kaixuan Ji, Lilong Xue, Xingjian Zhang, Yuxiao Dong, Jiahua Liu, Maodi Hu, and Jie Tang. 2022. Parameter-efficient prompt tuning makes generalized and calibrated neural text retrievers. *CoRR*, abs/2207.07087.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *CoRR*, abs/1908.07490.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022a. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR.

Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit S Dhillon, and Sanjiv Kumar. 2024. Two-stage LLM fine-tuning with less specialization and more generalization. In *The Twelfth International Conference on Learning Representations*.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2022b. Simvlm: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations*.

Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. 2020. Improving bert fine-tuning via self-ensemble and self-distillation. *CoRR*, abs/2002.10345.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. 2021. Florence: A new foundation model for computer vision. *CoRR*, abs/2111.11432.

Aston Zhang, Alvin Chan, Yi Tay, Jie Fu, Shuohang Wang, Shuai Zhang, Huajie Shao, Shuochao Yao, and Roy Ka-Wei Lee. 2021a. On orthogonality constraints for transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 375–382.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021b. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049.

Yichu Zhou and Vivek Srikumar. 2021. A closer look at how fine-tuning changes bert. *arXiv preprint arXiv:2106.14282*.

A Related Work

VL Model Architecture The Transformer and its variants (e.g., BERT, GPT) are widely adopted as VL model architectures due to their powerful attention mechanisms capturing correlations between image and text (Vaswani et al., 2017). Examples include VINVL using a Transformer encoder, OFA employing a Transformer encoder-decoder pair, and BLIP-2 utilizing a Transformer decoder. We evaluate CGFT on these models to demonstrate its robustness and applicability.

VL Unsupervised Pre-training VL models often undergo unsupervised pre-training on large datasets, employing objectives like masked language modeling, image-text matching, and causal language modeling (Li et al., 2023; Alayrac et al., 2022; Wang et al., 2022a; Yuan et al., 2021; Zhang et al., 2021b). This pre-training helps the model understand the relationships between image and text. Tasks include predicting masked words, scoring image-text matching, and predicting the next words from given image-text pairs. The models pre-trained using these approaches are evaluated using CGFT to assess their performance.

Semantic Richness and Rank Assessing the semantic richness of features is crucial for effective Vision-Language (VL) learning. This refers to how well a feature encapsulates fine-grained, dense information from the input. Evaluation often involves linear probing in computer vision. Numerous studies indicate a strong correlation between rank and information content in representations (Bansal et al., 2018; Zhang et al., 2021a). For instance, low-rank compression methods intentionally reduce rank to distill essential information, such as object class (Sainath et al., 2013; Swaminathan et al., 2020).

Fine-tuning Strategies in VL Learning To enhance pre-trained model performance, various transfer learning techniques address domain adaptation challenges. A parameter-efficient fine-tuning approach involves inserting additional modules into pre-trained model layers and optimizing only these modules (Houlsby et al., 2019; Hu et al., 2022).

B Experiments Setup

B.1 Model

Baselines To assess the effectiveness of PT-PEFT, we have employed a diverse set of pre-trained models featuring different architectures and sizes. Specifically, we have tested models such as VINVL base, VINVL large (Zhang et al., 2021b), OFA (base) (Wang et al., 2022a), BLIP (Li et al., 2022)(only for VQA) and BLIP-2 (ViT-g and OPT-2.7B) (Li et al., 2023) as our baseline model due to its good performance on VL sequence generation and classification among many VL model variants (Tan and Bansal, 2019; Lu et al., 2019; Li et al., 2020a; Zhou et al., 2020; Li et al., 2020b; Alayrac et al., 2022), as described in Table 7 (Zhang et al., 2021b; Wang et al., 2022a; Li et al., 2023).

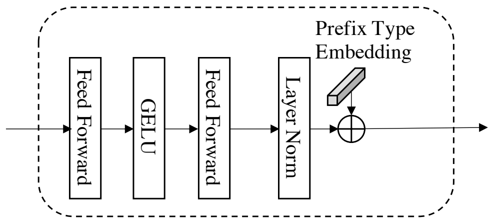


Figure 6: Prefix encoder structure.

Prefix Encoder Figure 6 illustrates the prefix encoder (see Section 3). In contrast to previous re-parameterizations (Li and Liang, 2021), our approach incorporates prefix type embedding to establish a symmetrical setting with token type embedding, as used in previous VL models (Zhang et al., 2021b; Li et al., 2020b). After training, the output of the prefix encoder can be saved as the new prefix, so there is no computational overhead in using this block. In other words, the block is only realized during the training phase.

B.2 Downstream task

Visual Question Answering Visual Question Answering task requires the model to select or generate the correct answer from the given question-image pair. For VINVL (Zhang et al., 2021b; Li et al., 2020b), we train the model to classify the answer given question and image pair sequence from answer sets (i.e., 3129 for VQAv2, 1852 for GQA). For OFA (Wang et al., 2022a) and BLIP-2 (Li et al., 2023), we train the model to generate the answer given question and image pair.

Model	# of Param	Module	Hidden Dim	Number of Layer	Number of Attention Head
VINVL Base	110M	VL Fusion Encoder (BERT-Base)	768	12	12
VINVL Large	340M	VL Fusion Encoder (BERT-Large)	1024	24	16
OFA Base	180M	Vision Encoder (ResNet-101)	2048	101	-
		VL Fusion Encoder (Transformer Enc Base)	768	6	12
		VL Fusion Decoder (Transformer Dec Base)	768	6	12
BLIP-2 (OPT 2.7B)	3.6B	Vision Encoder (ViT-g)	1408	40	16
		Q-Former (BERT-Base)	768	12	12
		VL Fusion Decoder (OPT 2.7B)	2560	32	32

Table 7: Baseline VL pre-trained models specifications.

Model	Module	Prefix-tuning Prefix Length	LoRA Weights
VINVL Base	VL Fusion Encoder (BERT-Base)	16	-
VINVL Large	VL Fusion Encoder (BERT-Large)	16	-
OFA Base	Vision Encoder (ResNet-101)	-	-
	VL Fusion Encoder (Transformer Enc Base)	64 (IC), 16 (VQA)	Q, K, V (r=16, a=32)
	VL Fusion Decoder (Transformer Dec Base)	64 (IC), 16 (VQA)	Q, K, V (r=16, a=32)
BLIP-2 (OPT 2.7B)	Vision Encoder (ViT-g)	-	Q, K, V (r=16, a=32)
	Q-Former (BERT-Base)	8 (IC), 16 (VQA)	Q, K, V (r=16, a=32)
	VL Fusion Decoder (OPT 2.7B)	8 (IC), 16 (VQA)	-

Table 8: Parameter-efficient tuning (Prefix-tuning and LoRA) specifications.

Image Captioning Image captioning task requires the model to generate a natural language description for the given input image. Image captioning fine-tuning is typically a 2-stage process, which consists of cross-entropy (CE) training and self-critical sequence training (SCST) (Rennie et al., 2017).

For the masked language model (VINVL), during CE training, we use masked language modeling for randomly masked tokens with a left-to-right (causal) attention mask to reflect the auto-regressive behavior (see VINVL (Zhang et al., 2021b) for details). And then, the model is trained by optimizing the CIDEr score with SCST which utilizes the score as the reward for REINFORCE algorithm (Rennie et al., 2017). For inference, we utilize a beam size of 5 for beam search, while for SCST, we use a beam size of 1.

For the causal language model (OFA, BLIP-2), we only train the model with the causal language modeling (i.e., next token prediction) with the causal attention mask. Note that we did not use SCST for these models.

B.3 Dataset

Image Captioning For IC experiments, we evaluate the performance of our proposed fine-tuning techniques on MS COCO (Lin et al., 2014) and Flickr30k (Plummer et al., 2015) datasets. We follow the Karpathy split (Karpathy and Fei-Fei, 2015) for a fair comparison. Karpathy split of COCO and Flickr30k datasets contain 83k/5k/5k and 29.8k/1k/1k images for train/val/test split.

Visual Question Answering For VQA experiments, the model is evaluated on the VQAv2 dataset (Antol et al., 2015). VQAv2 dataset contains 83k/41k/81k images and 444k/214k/448k question sets for train/val/test split, respectively.

B.4 Experiment Details

Hyper-parameters For training, we employ a set of hyper-parameters as detailed in Table 13. The table shows the best configurations for prefix-tuning and fine-tuning; these settings are also used for each stage of PT-PEFT. To update the network parameters, we utilize the AdamW optimizer (Loshchilov and Hutter, 2017) with betas set to (0.9, 0.99). For the learning rate schedule, We combine linear warm-up followed by linear decay, gradually increasing the learning rate from 0 to the maximum LR during warm-up epochs and linearly decaying it to 0 for the remaining training epochs.

Evaluation Metrics In evaluating image captioning, we employ the CIDEr, SPICE, and BLEU-4 metrics (Vedantam et al., 2015; Anderson et al., 2016; Papineni et al., 2002) to evaluate the quality of generated captions. The evaluation is performed using the *pycocoevalcap* API available at `https://github.com/salaniz/pycocoevalcap`. For visual question answering, we present accuracy as a performance metric.

Computational Resources We conducted experiments using four A100 (40GB) GPUs.

B.5 Implementation Details

Prefix-tuning In prefix-tuning, the VL model is kept frozen, and only the prefix-encoder block (see Figure 6) and prefix vectors are trained. Our implementation of the prefix-tuning closely follows the original prefix-tuning approach (Li and Liang, 2021), where an MLP is employed as the prefix encoder for stable optimization. The number of prefix vectors is empirically chosen for the best performance based on the experiment in Figure 8 as described in Table 8.

LoRA We implement the low-rank adapter following (Hu et al., 2022). We update all query, key, and value projection matrices in the self-attention module by setting the rank $r = 16$, scaling factor $\alpha = 32$, and dropout probability of 0.05 throughout all the experiments (see Table 8).

PT-PEFT PT-PEFT, proposed in our work, leverages both prefix-tuning and fine-tuning sequentially. For image captioning, we freeze the word embedding layer and the head throughout the training process, including both the prefix-tuning stage and the subsequent fine-tuning stage. In the prefix-tuning stage, we only train the prefix encoder and prefix embedding using CE training. Subsequently, we fine-tune the model using a combination of CE training and SCST (for VINVL COCO-IC only). For visual question answering, we follow a similar procedure. We first train the prefix encoder and prefix embedding (and the CLS head for VINVL) and then proceed with fine-tuning the model.

PT-LoRA PT-LoRA is the parameter-efficient version of prefix-tuning which performs the LoRA instead of the full fine-tuning in the second stage. To ensure a similar number of training parameters (i.e., 0.3 %) with prefix-tuning and LoRA tuning, we train only selected blocks (e.g., only Q-former is trained for the BLIP-2) for the LoRA tuning stage in PT-LoRA. Other than that, all the training processes and settings are the same as the PT-PEFT case.

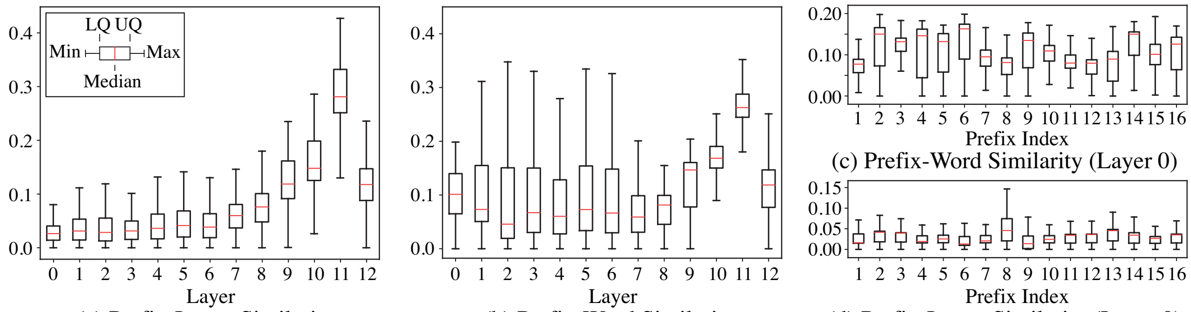
C Additional Experiments

C.1 Ablation Study

Training Step We conduct experiments to determine the best training steps each for prefix-tuning and fine-tuning stages in PT-PEFT. As shown in Table 6, we found that prefix-tuning requires training for a sufficiently long iteration until convergence (see M1 and M4). Otherwise, it may be learned to be ignored rather than providing assistance in the subsequent fine-tuning step. We can achieve high performance in the subsequent fine-tuning stage even with fewer training iterations compared to the preceding prefix-tuning (see M3 and M4). This is because fine-tuning starts from a higher performance level achieved by the already-trained prefix, compared to training solely from the pre-trained model without prefix. However, longer training leads to severe overfitting (see M2).

Prefix Length Longer prefixes (i.e., many prefix tokens) involve more trainable parameters, thus assumed to enhance the performance for prefix-tuning (Li and Liang, 2021). Figure 8 shows that performance indeed improves as the number of prefix tokens increases, but saturates after a certain point. Note that previous works on prefix-tuning often used much longer prefix lengths than our PT-PEFT, but since PT-PEFT refines all the parameters, longer prefix seems to be unnecessary for PT-PEFT.

Prefix Encoder In order to assess the impact of the prefix encoder design, we conducted ablation studies as summarized in Table 10. These experiments were performed on the VQAv2 dataset, following the training step of the PT-PEFT process. We use the same hyper-parameter settings described in Table 13. Notably, the results indicate a slight decrease in top-1 accuracy when the prefix



(a) Prefix-Image Similarity (b) Prefix-Word Similarity (c) Prefix-Word Similarity (Layer 0) (d) Prefix-Image Similarity (Layer 0)

Figure 7: Cosine similarities between prefix-word, and prefix-image feature in image captioning using CGFT.

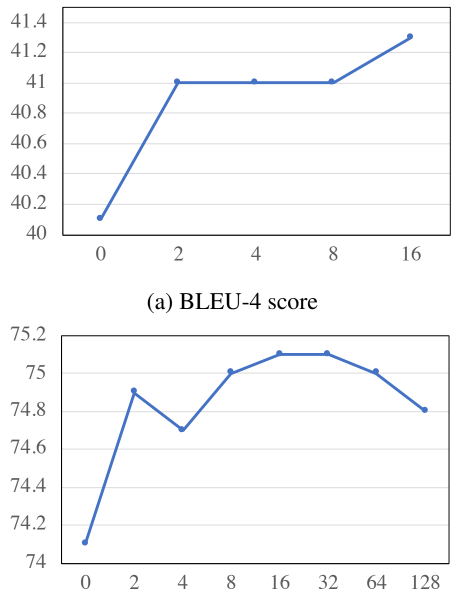
type embedding is removed, but there is a significant drop in top-5 accuracy. This suggests that the prefix type embedding plays an important role in improving performance. Furthermore, when the MLP block is removed, top-5 accuracy experiences a considerable decline. This demonstrates that the prefix encoder contributes to the overall performance of the model, highlighting its importance in capturing and encoding essential information for VQA tasks.

Alternation Training We conduct experiments to see whether the alternation training can further enhance the performance. As shown in Table 9, we found that prefix-tuning fails to learn the context necessary for the task during the alternation training. Even if the initial prefix-tuning is successful (see train alternation step 1), the knowledge learned from the pre-trained model during this phase is lost (see train alternation steps 4, prefix is no longer affecting the output). This loss may be attributed to retraining in the collapsed representation space. Repeated fine-tuning also causes overfitting and performance degradation (see train alternation steps 4 in Table 9).

C.2 Empirical Analysis

Mimicking Pre-trained Representations To gain insights into the learned representations of the prefix during training, we analyze cosine similarity between prefix tokens and image/caption tokens in the PT-PEFT-tuned model (prefix length of 16). We observe that the cosine similarities between 16 prefix tokens are very low, all below 0.09.

Furthermore, we find that the correlation between prefix-image and prefix-word increased across the different layers (see Figure 7 (a) and (b)). Interestingly, the prefix-word similarities (0.1-0.2)



(a) BLEU-4 score

(b) VQAv2 Acc1 (%)

Figure 8: Ablation on the prefix length in image captioning and visual question answering. The x-axis indicates the number of prefix tokens used.

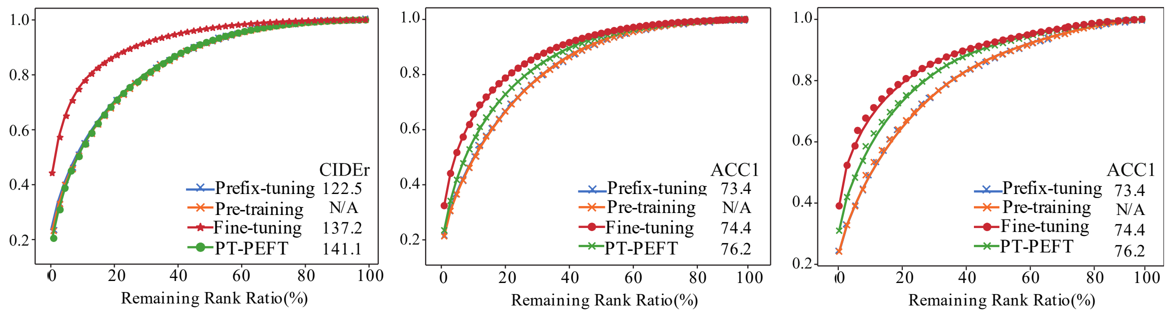
are higher than prefix-image similarities (0.0-0.05), especially in lower layers (see Figure 7 (c) and (d)). This suggests that the prefix maintains its representation space from pre-training by acquiring quasi-orthogonal bases that are relatively closer to pre-trained text features. However, in higher layers, the prefix-image similarities (0.2-0.4) are higher than prefix-text similarities (0.2-0.35) (see Figure 7 (a) and (b)). These results clearly indicate that the feature of the image is converted to language space through the interaction with prefix vectors.

SVD Experiments We experiment with SVD analysis as in Figure 4 on the VINVL(see Figure 9). The results in VINVL also show that rep-

	Alternation Steps											
	1			2			3			4		
	BLEU-4	CIDEr	SPICE	BLEU-4	CIDEr	SPICE	BLEU-4	CIDEr	SPICE	BLEU-4	CIDEr	SPICE
w/ Prefix	41.3	139.3	24.6	33.2	115.1	20.7	23.7	90.0	16.8	20.6	67.4	13.8
- Prefix	22.9	75.0	15.3	21.5	73.8	14.9	21.2	71.3	14.5	20.6	67.4	13.8

Table 9: Alternation training experiments on COCO image captioning.

VINVL (ResNeXt-152 C4 + BERT-Base)



(a) Image Captioning (b) Visual Question Answering (Image) (c) Visual Question Answering (Text)

Figure 9: Accumulated and normalized singular values of feature vectors extracted from the last layer of VINVL.

resentation collapse (i.e., most singular values of the representation matrix are close to zero) in the fine-tuned model while the representation space is preserved (i.e., most singular values are the same) in PT-Full-Finetuning or PT-LoRA model.

	Prefix-tuning Stage		Fine-tuning Stage	
	Acc1	Acc5	Acc1	Acc5
PT-PEFT	**73.8**	**93.1**	**75.2**	**93.3**
- Prefix Type Embedding	73.6	90.6	74.8	91.0
- Prefix MLP	73.3	90.3	74.9	90.8
- Prefix Encoder	73.3	90.3	74.7	90.7

Table 10: Ablation of prefix-encoder in VQAv2 validation split.

More Qualitative Examples Figures 10 and 11 show qualitative examples of generated captions on the COCO Karpathy test split and VQAv2 valid set, respectively. We visualize representative images and corresponding captions generated by two models trained using PT-PEFT and fine-tuning. Compared to the fine-tuned model, the PT-PEFT-tuned model demonstrates a strong ability to capture important details for enriching generated captions. For example, the proposed method enables extracting proper object-related attributes such as ‘cut in half’, ‘in the mirror’, ‘in front of’, and ‘a red and yellow’. Similarly, in VQA, the predictions from PT-PEFT are more consistent with the answer, and there is a high correlation within the top-5 candidates. In contrast, the predicted topmost answers after only applying the fine-tuning are much less similar to each other, implying that the learned word representations are lost. These observations can be attributed to the rank of the feature matrix, as the high-rank features produced by PT-PEFT contain semantically rich information.

Zero-shot Qualitative Example To provide a more comprehensive understanding of the qualitative differences between zero-shot, prefix-tuned, and fine-tuned models, we present additional examples in Table 11. These examples illustrate how fine-tuned models, despite achieving high metric scores, may overlook important visual details, resulting in captions that are shorter and more simplified compared to those generated by prefix-tuning and zero-shot approaches.

D Discussion

D.1 Simply Adding Parameters Helps?

One might assume that the performance enhancement is simply a result of adding additional parameters during fine-tuning. However, it is important to note that increasing the number of parameters (i..e, stacking more layers) does not necessarily expand the representation space. Intuitively, if we consider a linear transformation where $\mathbf{Y} = \mathbf{W}\mathbf{X}$, with $\mathbf{W}$ as the layer weight and $\mathbf{X}$ as the input, then the rank of $\mathbf{Y}$ is limited by the minimum rank between $\mathbf{W}$ and $\mathbf{X}$ (i.e., $\text{rank}(\mathbf{Y}) \leq \min(\text{rank}(\mathbf{W}), \text{rank}(\mathbf{X}))$). This means that simply adding more layers would not contribute to avoiding representation collapse. Moreover, previous research has demonstrated that incorporating more complex layers can lead to a faster collapse in rank (Dong et al., 2021).

D.2 Expressive Power vs. Semantic Richness?

'Expressive power of parameters' refers to a model's ability (complexity and size) to adjust its weights to fit a new downstream task. On the other hand, a 'semantically rich feature representation space' or 'high-rank feature' refers to the capability of a model to capture informative features that exhibit strong generalization across different tasks.

To maximize the downstream performance, both 'expressive power' and 'semantic richness' are important. Our experiments show that prefix-tuning, which only tunes a few parameters, has limited expressive power but is good at preserving a semantically rich feature representation space. In contrast, fine-tuning, an approach to modify all parameters, has greater expressive power but might distort the representation space, resulting in lower rank and reduced semantic richness compared to a pre-trained model.

Our findings (including SVD analysis and task performance comparison) are consistent with the previous analyses on fine-tuning where 'fine-tuning makes the space simpler' (Zhou and Srikumar, 2021) and 'simplified space yields lower performance to out-of-domain (OOD) data (bad generalization)' (Kumar et al., 2022). In summary, the goal of PT-PEFT is to take advantage of both expressive power and the preservation of semantic richness of the feature representation space.

D.3 How Prefix-Tuning Preserves the Representation Space?

To elucidate how prefix-tuning preserves the representation space, we analytically compare the rank of the representation space (i.e., vector space) after applying the attention operation in both fine-tuned and prefix-tuned models.

In a Transformer model, information from the input tokens of the input sequence is mixed exclusively through self-attention. The other components in the Transformer, such as the feed-forward network, are token-wise operators and thus are not affected by prefix tokens. Specifically, for a given input sequence $\mathrm{X} = [\mathbf{x}_0; \dots; \mathbf{x}_N]$, the output of self-attention is the weighted sum of the value matrix XW_V, where the weights are the attention scores:

$$f(\mathrm{X}) = \sigma(\mathrm{W}_Q \mathrm{X} \mathrm{X}^T \mathrm{W}_K^T)\mathrm{X}\mathrm{W}_V \tag{6}$$

where σ denotes the softmax function.

In the case of prefix-tuning, the self-attention function is reformulated to incorporate a learnable prefix matrix P:

$$f_{\text{Prefix}}(\mathrm{X}) = \sigma(\mathrm{W}_Q[\mathrm{X};\mathrm{P}][\mathrm{X};\mathrm{P}]^T \mathrm{W}_K^T)\mathrm{X}\mathrm{W}_V \tag{7}$$

Here, only the number of input tokens increases while the model parameters remain unchanged. Considering the rank of the matrix product, which satisfies the inequality $rank(\mathrm{AB}) \leq \min(rank(\mathrm{A}), rank(\mathrm{B}))$, the rank of the self-attention output is bounded by:

$$\begin{aligned} &rank\left(f(\mathrm{X})\right) \\ &\leq \min\left(|\mathrm{X}|, rank\left(\mathrm{XW}_V\right)\right) \end{aligned} \tag{8}$$

$$\begin{aligned} &rank\left(f_{\text{prefix}}(\mathrm{X})\right) \\ &\leq \min(|\mathrm{X}| + |\mathrm{P}|, rank\left([\mathrm{X};\mathrm{P}]\,\mathrm{W}_V\right)) \end{aligned} \tag{9}$$

Assuming the softmax output is full rank, this indicates that the upper bound of the rank is at least as large as the rank of the pre-trained representation space, provided that the parameters remain unchanged:

$$\begin{aligned} &\min(|\mathrm{X}|, rank(\mathrm{XW}_V)) \\ &\leq \min(|\mathrm{X}| + |\mathrm{P}|, rank([\mathrm{X};\mathrm{P}]\mathrm{W}_V)) \end{aligned} \tag{10}$$

This theoretical analysis suggests that prefix-tuning maintains or even enhances the semantic richness of the feature representation space by preserving the rank, whereas fine-tuning can reduce the rank, thereby diminishing the semantic richness.

COCO Image ID	Zero-Shot	Finetune	Prompt
272117	“a group of people sitting around a table with a birthday cake in front of them”	“a group of people sitting around a table with a cake”	“a group of people sitting around a table with a birthday cake in front of them”
503392	“two horses in an arena with a person riding on the back of one of the horses”	“two horses in an arena with a person riding one of the horses”	“two horses in an arena with a person riding on the back of one of the horses”
60467	“a lunch tray with a breakfast sandwich, orange juice, and a glass of milk”	“a lunch tray with a sandwich, orange juice, and a glass of milk”	“a tray of food on a table”
544471	“a man and a woman sitting on a brick wall with a laptop in front of them”	“a woman and a boy sitting on steps with a laptop”	“a man and a woman posing with a laptop”
117170	“two pizza rolls sitting on a counter with a sign that says ‘pizza rolls’ ”	“two pizza rolls sitting on top of a silver platter”	“two pizza rolls on a silver platter with a sign that says ‘pizza rolls’ ”
235644	“a group of people working on a person on a stretcher at a train station”	“a group of people on a platform next to a train”	“three people helping a person on a stretcher on a train platform”
514607	“an umbrella on a beach with rocks and a body of water in the background”	“an umbrella on a rocky beach with the ocean in the background”	“a beach with a beach umbrella in the foreground and the ocean in the background”
89541	“a container of food with strawberries, blueberries, and a muffin in it”	“a bowl filled with fruit and muffins on a table”	“a yellow container with strawberries, blueberries, and a muffin in it”
477470	“a street at night with traffic lights and a building in the background”	“a traffic light on a city street at night”	“a street at night with traffic lights and a building in the background.”
529004	“a car driving down a road with a herd of cows on the side of the road”	“a herd of cattle crossing a road in front of a car”	“a car driving down a road with a herd of cows on the side of the road”
545407	“an airplane flying in the sky with a clear blue sky in the background”	“an airplane flying through a clear blue sky”	“an airplane flying in the sky with a blue sky behind it”
255036	“an intersection with traffic lights and a building in the background”	“a traffic light sitting on the corner of a street”	“a traffic light at an intersection with a building in the background”
276146	“a pizza on a cutting board with a glass of wine and a bottle of wine”	“a pizza sitting on a cutting board next to a bottle of wine”	“a pizza on a cutting board with a glass of wine next to it”
62554	“some food on a table with a bowl of broccoli and a bowl of asparagus”	“a table topped with bowls of food and plates of food”	“a bowl of broccoli and a bowl of asparagus on a table”
554980	“a red school lunch tray with a sandwich, orange, and a glass of milk”	“a red plastic tray with a sandwich, fruit, and a glass of milk”	“a red tray with food on it”
290951	“people walking in a building with umbrellas hanging from the ceiling”	“people walking under colorful umbrellas in a building”	“umbrellas suspended from the ceiling of a building”
299039	“a plate of food on a table with a vase of flowers in the background”	“a plate of food on a table with a vase of flowers”	“a plate of food on a table with a vase of flowers in the background”
379842	“a wii game with a wii remote and nintendo super mario galaxy 2 game”	“a wii game and controller sitting on a table”	“a wii remote and nintendo super mario galaxy 2 game”

Table 11: Comparison of Captioning Methods on COCO Dataset

Training method	Total train epoch	Warmup epoch	Max LR	Batch size	Weight decay
		COCO IC BASE			
Prefix-tuning	30	3	1.00E-05	1024	0.2
CE	40	12	1.00E-05	1024	0.2
SCST	75	15	3.00E-06	128	0.2
		COCO IC LARGE			
Prefix-tuning	30	3	1.00E-05	512	0.2
CE	30	6	3.00E-06	512	0.2
SCST	50	10	3.00E-06	192	0.1
		Flickr30k IC BASE			
Prefix-tuning	30	0	5.00E-05	512	0.1
Fine-tuning	70	0	1.00E-05	512	0.15
		Flickr30k IC LARGE			
Prefix-tuning	30	0	5.00E-05	512	0.1
Fine-tuning	70	0	3.00E-05	512	0.15
		VQA BASE			
Prefix-tuning	50	0	1.00E-04	512	0.05
Fine-tuning	25	3	1.00E-05	512	0.05
		VQA LARGE			
Prefix-tuning	50	0	5.00E-05	512	0.05
Fine-tuning	25	3	5.00E-06	512	0.05
		GQA BASE			
Prefix-tuning	5	0.5	1.00E-04	512	0.05
Fine-tuning	5	0.5	1.00E-05	512	0.05

Table 12: Training hyper-parameters for VINVL. PT-PEFT is trained with the same hyper-parameter with Fine-tuning (CE) in the table. Image size of 640x480 is used.

Training method	Total train epoch	Warmup epoch	Max LR	Batch size	Weight decay
		COCO IC			
Prefix-tuning	10	0	1.00E-03	16	0.01
LoRA	5	0	1.00E-03	16	0.01
Fine-tuning	5	0	1.00E-03	16	0.15
CGFT (2nd Stage)	10	0	1.00E-05	16	0.15
CGFT-LoRA (2nd Stage)	10	0	1.00E-05	16	0.15
		Flickr30k IC			
Prefix-tuning	5	0	1.00E-03	16	0.01
LoRA	5	0	1.00E-03	16	0.01
Fine-tuning	5	0	1.00E-03	16	0.15
CGFT (2nd Stage)	10	0	1.00E-05	16	0.15
CGFT-LoRA (2nd Stage)	10	0	1.00E-05	16	0.15
		VQA			
Prefix-tuning	50	0	1.00E-04	512	0.05
LoRA	50	0	1.00E-04	512	0.05
Fine-tuning	25	3	1.00E-05	512	0.05
CGFT (2nd Stage)	10	0	1.00E-05	16	0.15
CGFT-LoRA (2nd Stage)	10	0	1.00E-05	16	0.15

Table 13: Training hyper-parameters for OFA. Image size of 480x480 is used.

Training method	Total train epoch	Warmup Steps	Max LR	Batch size	Weight decay
		COCO IC			
Prefix-tuning	5	5000	5.00E-05	128	0.05
LoRA	5	5000	1.00E-04	128	0.05
Fine-tuning	5	5000	1.00E-05	128	0.05
		Flickr30k IC			
Prefix-tuning	5	5000	5.00E-05	128	0.05
LoRA	5	5000	1.00E-04	128	0.05
Fine-tuning	5	5000	1.00E-05	128	0.05
		VQA			
Prefix-tuning	5	0	5.00E-05	512	0.05
LoRA	5	0	1.00E-04	128	0.05
Fine-tuning	5	0	1.00E-03	128	0.05

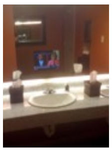
Table 14: Training hyper-parameters for BLIP-2. PT-PEFT and PT-LoRA are trained with the same hyper-parameter with LoRA and Fine-tuning in the table. Image size of 224x224 is used.

GT
a sandwich cut in half on a plate in front of a laptop.

a plate with a sandwich and a mountain dew in the back.

PT-PEFT
a sandwich cut in half on a plate with a bottle of soda.

Fine-tuning
a sandwich on a plate on a table.



GT
bathroom area with multiple sinks and mirrors with television reflected.

a bathroom with a television, sink and two boxes of tissues.

PT-PEFT
a bathroom with two sinks and a television in the mirror.

Fine-tuning
a bathroom with a sink and a mirror.

GT
a woman holding a cake with candles and a man blowing them out.

the man blows out the birthday candles.

PT-PEFT
a man and an older woman blowing out a candle on a cake.

Fine-tuning
a man and a woman holding a cake.

GT
a large man in a top hat is on his phone by an old red ford.

a man in a top hat and suit standing in front of an old truck talking on his cell phone.

PT-PEFT
a man in a top hat talking on a cell phone in front of a red truck.

Fine-tuning
a man in a suit talking on a cell phone.

GT
donuts in baskets are displayed by people sitting at a table.

A blue basket filled with donuts on top of a table.

PT-PEFT
a group of people standing around a blue tray of donuts.

Fine-tuning
a blue tray of donuts on a table.

GT
a person breaking a bottle with a baseball bat.

a boy in yellow shirt swinging a baseball bat.

PT-PEFT
a man is swinging a baseball bat at a fireworks display.

Fine-tuning
a man swinging a golf club at a ball in the water.

GT
a flock of small birds flying in the sky over the water.

a black and white image showing birds flying over a body of water.

PT-PEFT
a group of birds flying in the sky over a beach.

Fine-tuning
a group of birds flying in the sky over a field.

GT
a close up of a woman wearing a shirt and tie.

there is a woman next to water and many factory buildings.

PT-PEFT
a woman in a white shirt and a tie standing in front of a city.

Fine-tuning
a woman standing in front of a cloudy sky.

GT
a couple of people sitting on a bench next to a dog.

a large white dog sits on a bench with people next to a path.

PT-PEFT
a man and a woman sitting on a bench with a white dog.

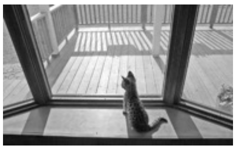
Fine-tuning
a man and a white dog on a bench.

GT
a red and yellow train pulling into a train station.

red/yellow train with people standing nearby waiting to board.

PT-PEFT
a red and yellow train parked at a train station.

Fine-tuning
a red train is parked at a train station.



GT
a cat on the window looking outside next to the balcony.

tiger kitten sitting by french window looking out over sunny balcony.

PT-PEFT
a cat sitting on a porch looking out of a window.

Fine-tuning
a cat sitting on top of a window sill.

GT
three zebras and other wild animals out in a semi-green field.

three zebras and two other animals grazing.

PT-PEFT
a couple of zebras and other animals standing next to a body of water.

Fine-tuning
a group of zebras standing next to a body of water.

Figure 10: Qualitative examples of generated captions on COCO Karpathy test split. **GT**: the ground-truth captions.

Question

“What city is this in?

GT

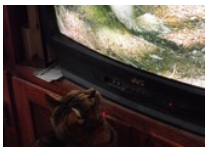
“new york”

PT-PEFT

Top_5_answer:
"new york", "washington", "chicago", "washington dc", "boston

Fine-tuning

Top_5_answer:
"101", "2", "31", "4", "unknown"



Question

“Why is the cat looking at the TV?

GT

“curious”

PT-PEFT

Top_5_answer:
"curious", "watching tv", "yes", "bored", "playing"

Fine-tuning

Top_5_answer:
"yes", "curious", "bark", "it isn't", "dead"

Question

“What’s going on in the wires above the buildings?”

GT

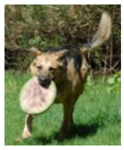
“electricity”

PT-PEFT

Top_5_answer:
"electricity", "power", "nothing", "power lines", "unknown"

Fine-tuning

Top_5_answer:
"advertisement", "for sale", "stop", "no", "nothing"



Question

“What kind of dog is this?

GT

“german shepherd”

PT-PEFT

Top_5_answer:
"german shepherd", "mutt", "lab","goldenretriever", "labrador"

Fine-tuning

Top_5_answer:
"brown", "white", "terrier", "lab", "mutt"

Question

“What is the destination for bus 176?

GT

“pandang”

PT-PEFT

Top_5_answer:
“los angeles”, “Beijing”, “Chicago”, “china”, “unknown”

Fine-tuning

Top_5_answer:
“unknown”, “can’t tell”, “not sure”, “city”, “don’t know”

Question

“What company does the moving truck belong to?

GT

“budget”

PT-PEFT

Top_5_answer:
"fedex", "moving", "ford", "target", "unknown"

Fine-tuning

Top_5_answer:
"unknown", "can't tell", "nike", "not possible", "not sure"

Question

“Where is the sunshine?

GT

“sky”

PT-PEFT

Top_5_answer:
“behind clouds”, “sky”, “in sky”, “above”, “yes”

Fine-tuning

Top_5_answer:
“background”, “in background”, “left”, “behind”, “right”

Question

“What operates this transportation device?

GT

“human”

PT-PEFT

Top_5_answer:
”motor”, “man”, “driver”, “person”, “motorcycle”

Fine-tuning

Top_5_answer:
"seat", "handlebars", "light", "radio", "motorcycle"

Question

“What are the two woman sitting waiting for?

GT

“their flight”

PT-PEFT

Top_5_answer:
"train", "bus", "luggage", “family”, "nothing"

Fine-tuning

Top_5_answer:
"nothing", "child", "luggage", "people", "train"

Question

“What does this cake say?

GT

“congratulations orchard team and happy birthday james”

PT-PEFT

Top_5_answer:
"happy birthday", "bird", "happy", "black", "harry potter"

Fine-tuning

Top_5_answer:
"heart", “stop”, "love", "peace", "cross"

Question

“Which restaurant made the food?

GT

“nathan’s”

PT-PEFT

Top_5_answer:
“nathan’s”, "fast food", "mcdonald’s”, "hot dog", "restaurant"

Fine-tuning

Top_5_answer:
"unknown", "home", “bakery", "kitchen", "nathan's"

Question

“Who has the green poles?

GT

“the man on left”

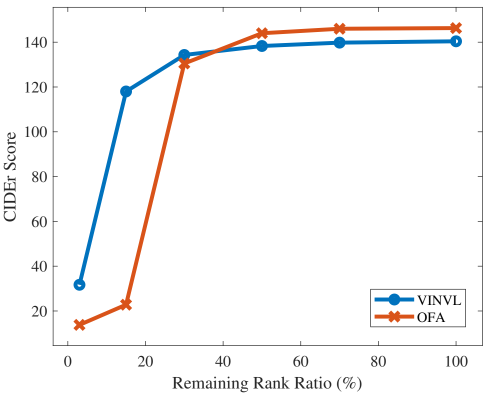
PT-PEFT

Top_5_answer:
"man on left", "woman", "boy", "man on right", “man"

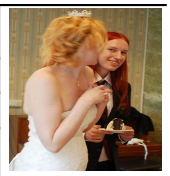
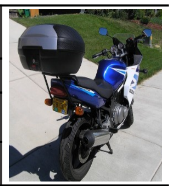
Fine-tuning

Top_5_answer:
"man", "woman", "right", "person", "girl"

Figure 11: Qualitative examples of generated captions on VQAv2 validation split. **GT**: the ground-truth answer.



(a) CIDEr scores on Image Captioning

[GT] " A bride is with long red haired person with cake. "	
[100%] 'a bride and groom standing next to each other holding a piece of cake.'	
[30%] 'a woman in a wedding dress eating a piece of cake.'	
[15%] 'a woman in a bridenatalnatalnatal POSTnatalnatal New Turn'	
[3%] 'ievalhibitedhibitedhibited Sectionievalievalieval POSTievalieval MA MA MAievalieval'	
[GT] "A blue motorcycle with luggage compartment parked at a driveway. "	
[100%] 'a blue motorcycle parked on the sidewalk with luggage on the back.'	
[30%] 'a blue motorcycle parked on the side of street.'	
[15%] 'neauneauneauagherzoszoszosearchzoszosneauneauum ptionzoszosearcher'	
[3%] 'ociationhibitedhibitedinylinylinylictionictionictionneaun eauneauagheragheraghervisor'	

(b) Generated caption examples

Figure 12: The effect of rank reduction on COCO image captioning performance. The percentage in (b) denotes the remaining rank ratio.