

# Applying Prompts and Parameter-Efficient Methods to Enhance Single-Stream Vision-Language Transformers

Xuehao Liu<sup>a</sup>, Sarah Jane Delany<sup>b</sup> and Susan McKeever<sup>c</sup>

*School of Computer Science, Technological University Dublin, Ireland*

**Keywords:** Image Captioning, Prompts, Parameter-Efficient Tuning, Vision-Language Transformer.

**Abstract:** Large-Scale transformer models pose challenges due to resource-intensive training, time, and data requirements for fine-tuning on new tasks, mainly due to their extensive parameter count. To address this, zero-shot and few-shot learning, aided by techniques like prompts and parameter-efficient modules, have emerged. However, these techniques are often tailored for vision-only or language-only tasks, leaving a gap for their effectiveness in multi-modal tasks like image captioning. This paper explores the effectiveness of prompts and parameter-efficient modules in reducing the training effort for image captioning. Rather than extensive fine-tuning, we trained only the prompt and parameter-efficient modules on the pretrained Oscar transformer model using the COCO dataset. We tested five prompt tuning approaches and two parameter-efficient methods. Notably, combining visual prompt tuning (VPT) with Adapter and LoRA led to a 2% Cider score improvement after just one epoch training, with a minimal increase in trainable parameters (5.7%). Our work paves the way towards using single-stream transformer models for a variety of fine-tuned tasks, but with a huge potential reduction in retraining time and processing resources.


## 1 INTRODUCTION


Prominent transformer models, including ChatGPT (Brown et al., 2020), GPT-4 (OpenAI, 2023), BLOOM (Scao et al., 2022), and LLaMA (Touvron et al., 2023), exhibit remarkable capabilities, accommodating over 20 languages, over text, images and audio data. Their performance often surpasses human performance across various tasks. However, these models are characterized by an immense parameter count, typically numbering in the hundreds of billions, and their training necessitates substantial computational resources. For example, the training cost for GPT-3 (Brown et al., 2020) alone amounted to an exorbitant 12 million USD (Floridi and Chiriatti, 2020). When training is required for new downstream tasks, the prevailing approach is to opt for zero-shot learning or few-shot learning methods. This eliminates the full fine-tuning process which is prohibitively expensive. In zero-shot learning, the pre-trained transformer model will use the downstream task data without undergoing any fine-tuning on the downstream task data. While in few-shot learning,


the model receives a limited number of examples as contextual cues to aid in task completion.

Recently, transformer models have used different methods to improve their performance while reducing the computational requirements. These include using prompts to facilitate the zero-shot and few-shot learning (Alayrac et al., 2022; Li et al., 2023) and parameter-efficient tuning methods (Houlsby et al., 2019; Zaken et al., 2021; Aghajanyan et al., 2020; Hu et al., 2021). The incorporation of prompts enables large language models to enhance their zero-shot performance by providing concise task descriptions as a component of the model input. Parameter-efficient tuning methods include a module with a small number of parameters alongside the original model. With the original pre-trained model frozen (i.e. no fine-tuning), these parameter-efficient modules are trained using the target dataset. The new model is the combination of the original model and the parameter-efficient modules together refined for the target task with shorter training time and less training resources.

The majority of prompt tuning and parameter-efficient tuning techniques have been formulated with a single-modal task focus, such as image classification (Jia et al., 2022) and natural language generation (NLG) (Hu et al., 2021). Typically, a prompt

<sup>a</sup>  <https://orcid.org/0000-0001-9815-489X>

<sup>b</sup>  <https://orcid.org/0000-0002-2062-7439>

<sup>c</sup>  <https://orcid.org/0000-0003-1766-2441>

solely influences one modality, either textual (Liu et al., 2021; Li and Liang, 2021; Hounsby et al., 2019; Karimi Mahabadi et al., 2021) or visual (Jia et al., 2022; Zang et al., 2022). Our objective is to evaluate whether these established methods exhibit similar performance enhancements in the multi-modal context such as image captioning, incorporating both visual and textual inputs.

Current prompt methods are typically tailored for multi-stream transformers, which handle visual and language features separately with different transformer blocks. We used the Oscar pre-trained model due to its characteristic as a single-stream transformer which is designed to process both visual and language features within the same transformer block.

We evaluated five different prompts as additional task-specific learnable parameters for Oscar, to identify the optimal prompt approach. We then applied two parameter-efficient tuning techniques, specifically LoRA (Hu et al., 2021) and Adapter (Hounsby et al., 2019), with the objective of evaluating their potential to enhance Oscar’s performance in the domain of image captioning. Additionally, we conducted a comparative analysis to assess the performance of the various prompts in conjunction with LoRA and Adapter. We found that including both prompts and parameter-efficient tuning improved the performance of Oscar on image-captioning. The VPT approach which added trainable prompts to the visual features was found to be the best prompt. The addition of LoRA and Adapter further improved performance, with a small number of trainable parameters, giving an overall 2% increase in the Cider score.

## 2 RELATED WORK

For Large Language Models (LLMs), the conventional training paradigm requires initial pre-training followed by fine-tuning for specific downstream tasks. However, fine-tuning the entire model for new downstream tasks has become prohibitively resource-intensive. We review related works in the two distinct approaches that have emerged to alleviate the resource-intensive process of fine-tuning the entire model: Prompts and Parameter-Efficient methods.

*Prompts:* The first is a paradigm shift in LLMs from fine-tuning to zero-shot learning or few-shot learning, augmented by the use of prompts. In zero-shot or few-shot learning the fine-tuning phase has been replaced by a step that involves pre-training, prompting and predicting (Liu et al., 2023a). For instance, consider the task of predicting the sentiment of a sentence like “I missed the bus today.” In the

absence of prompts, this sentence is presented as the sole input. However, when prompts are employed, the sentence is augmented by, for example, “I felt so ...,” which serves as both a cue for the model to perform sentiment classification and a format for constructing the input. This shift towards prompts has proven to be efficient and adaptable for LLMs across a range of diverse downstream tasks, as demonstrated by the capabilities of GPT-3 (Brown et al., 2020).

More recent developments in LLMs, including *prefix-tuning* (Li and Liang, 2021) and *p-tuning* (Liu et al., 2021), have further refined this approach by transforming English prompts into sets of trainable embeddings. The primary model is held constant with no fine-tuning, while the new downstream dataset is used to train these prompt embeddings, thereby conserving valuable training resources and time. Each downstream task is equipped with its own tailored prompt embeddings.

For the visual modality, the concept of *visual prompt tuning (VPT)* (Jia et al., 2022) has been introduced, using a trainable vector similar to the embeddings in prefix-tuning and p-tuning. Furthermore, VPT introduces trainable prompts at each transformer block layer to further improve the performance.

In the context of vision-language modality, which entail two modalities (vision and language), the *CoOp* (Zhou et al., 2022b) approach uses trainable vectors as prompts specifically designed for the language modality. CoOp builds upon the CLIP (Radford et al., 2021) backbone model (which has learnt visual concepts from natural language supervision), which consists of a two-stream transformer with both text and image encoders. In this setup, the text encoder incorporates the trainable vectors alongside token embeddings of its input. *CoCoOp* (Zhou et al., 2022a), as an enhancement of CoOp, established fully-connect layers between the trainable language prompts and the output from image encoder. *Unified Prompt Tuning (UPT)* (Zang et al., 2022) adopts a similar prompt-building approach to CoCoOp, using the same CLIP backbone transformer. Notably, UPT employs a set of trainable vectors to generate prompts for both vision and language inputs. *Multi-modal Prompt Learning (Maple)* (Khattak et al., 2023) improved prompts further by injecting them into both inputs and each transformer block layer, and fully connecting the prompts between image and language transformer blocks. *Dynamic Prompting* (Yang et al., 2023) introduces a dynamic framework for prompt training, enabling the change of prompt length and positioning to enhance model performance.

*Parameter Efficient Methods* reduce the significant computational resources required for fine-tuning

a LLM by reducing the number of parameters involved in fine-tuning. A fraction of the model will be updated or a limited number of supplementary parameters are introduced (with the original model unchanged) when applied to a new dataset and downstream task. One such approach *Structure Aware Intrinsic Dimension (SAID)* (Aghajanyan et al., 2020) shifted the original pre-trained model parameters to align with the characteristics of the target dataset. In this context, the intrinsic dimension of a subspace denotes the minimum dimensionality required to address the alignment between the new model and the original model. SAID findings indicate that a 200-dimensional subspace can achieve up to 90% of the performance of the original model. Another approach, termed *Bias-Terms Fine-tuning (BitFit)* (Zaken et al., 2021), proposes fine-tuning exclusively on the bias of the original model. This strategy allows for resource savings during training. *Low-Rank Adaptation (LoRA)* (Hu et al., 2021) introduces an external module alongside the original pre-trained model, focused on extracting low-rank intrinsic features from each attention layer. The training efforts are concentrated on this external module, leading to notable savings in training resources. *Adapter* modules (Houlsby et al., 2019), initially designed for Natural Language Processing (NLP) tasks, take the form of bottleneck-shaped layers inserted between the attention blocks within a transformer layer, which achieved a similar performance to fine-tuning with only updating 3.6% of the original model. This architectural concept has found broader applicability in various vision tasks (Gao et al., 2023; Chen et al., 2022). *Compressor* (Karimi Mahabadi et al., 2021) represents an advancement in the Adapter framework by replacing the original multiplication operation with hyper-complex multiplication, resulting in reduced computational demands during training.

Overall there are a variety of prompt approaches and ways to reduce the number of parameters required by LLMs which have been shown to reduce the resources required in training. We will explore how these can assist a single-stream pre-trained transformer on the multi-modal downstream task of image captioning.

### 3 APPROACH

To determine if prompts and parameter-efficient tuning methods can improve the performance of a single-stream transformer used for image captioning, five prompt methods (VPT, CoOp, UPT, Maple, Dynamic Prompt) and two parameter-efficient tuning methods

(LoRA and Adapter) were applied to a multi-modal visual language transformer framework, Oscar for the task of image captioning. We choose Oscar as the exemplar transformer due to its multi-modal single-stream architecture, which has not been tested previously. The pre-trained Oscar model was not fine-tuned in any way. The tuning for image captioning was performed through the prompting and use of the parameter-efficient tuning modules.

Fig.1 shows an overview of the approach illustrating how the prompts and parameter-efficient modules were added to Oscar. The input to Oscar has three parts: caption, the labels of objects in the image and the region features for each object. The two prompt methods, CoOp and VPT are added for illustration. CoOp is added to object labels and VPT is added to region features. Oscar has several transformer layers and multiple attention blocks for each layer. Adapter is added in each transformer layer and LoRA is added in each attention block. This section will discuss Oscar, as the example transformer, and how the prompts and parameter-efficient modules were added into Oscar.

#### 3.1 Example Transformer: Oscar

Oscar is pre-trained to accommodate a range of vision-language downstream tasks, including text retrieval, image retrieval, image captioning, and visual question answering. In our study, we specifically target image captioning as the focal task. Like other vision-language transformers, Oscar can handle two modalities: token embeddings and visual features. Notably, the input structure may vary depending on the specific training and generating phases.

For the experiment, an off-the-shelf Oscar model was used, preserving the original BERT self-attention backbone weights. The notable alterations made included the incorporation of prompts into the Oscar input as well as modifications introduced to the BERT attention structure to align with the LoRA and Adapter module requirements. As we are primarily evaluating the zero-shot learning performance in the context of image captioning, we exclusively implemented the pre-training phase while omitting the fine-tuning step.

The configuration of the input for Oscar comprises a three-part structure, covering three aspects of an image: the caption, the tags, and the object features. Specifically, the caption refers to the word embedding sequence derived from the image caption itself, while the tags correspond to the English terms denoting object labels. The object features are generated from Faster R-CNN (Ren et al., 2015) which also gener-

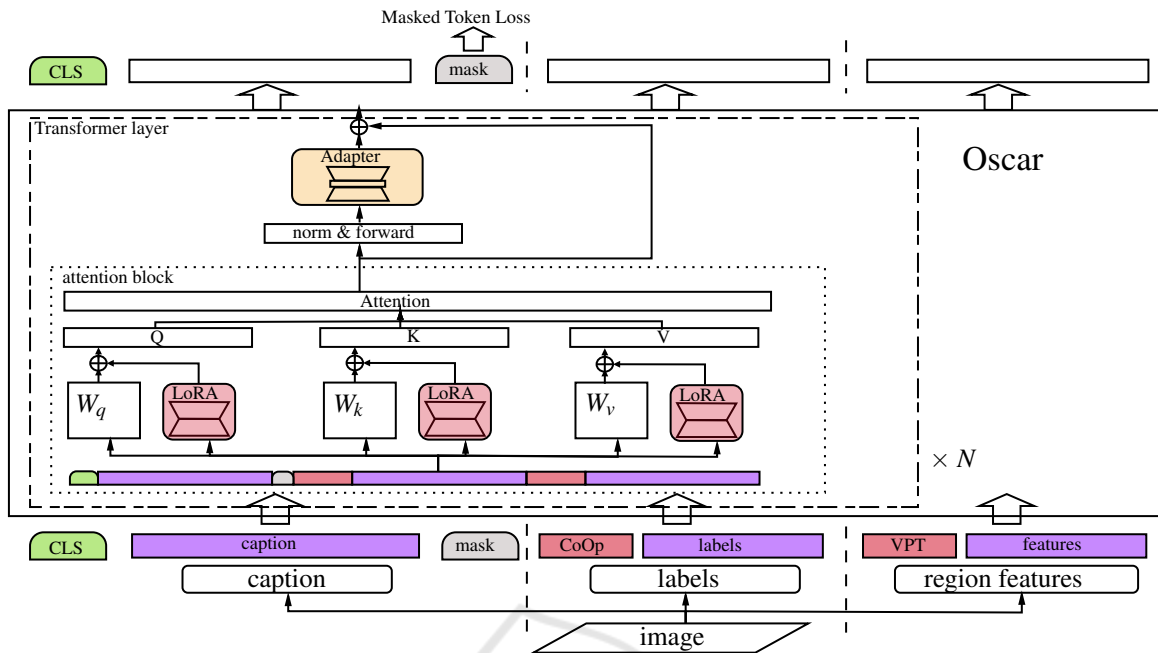


Figure 1: Illustration of the main structure of adding prompts and parameter-efficient modules: The input to Oscar has three parts: caption, the object labels and the region features for each object. CoOp is added to object labels and VPT is added to region features. Adapter is added in the transformer layer and LoRA is added in the attention block.

ates a vector representing each detected object region corresponding to each object label. The distinctive token, “[SEP]”, segregates these three parts within the input sequence, which is initiated with a class token designated as “[CLS].”

In line with the loss objective paradigm established by BERT (Devlin et al., 2018) and adopted by Oscar (Li et al., 2020), we compute the Masked Token Loss (MTL), which predicts the masked tokens. In the context of a language-only environment, the model is expected to infer the missing token given the surrounding tokens. In our hybrid language and vision context, each input sequence undergoes random masking of 15% of the English word tokens in the captions, replacing them with a special token, “[MASK]”. Subsequently, the model predicts the masked tokens in this modified sequence. The training process is to optimize the accuracy of the missing masked token.

### 3.2 Adding Prompts to Oscar

Some modifications were needed for some prompt methods to allow them to work with Oscar, such as the dimension of prompts is changed to adapted to the attention block of Oscar. We selected five prompt methods to test, including VPT (Jia et al., 2022), CoOp (Zhou et al., 2022b), UPT (Zang et al., 2022), Maple (Khattak et al., 2023), and Dynamic Prompting (Yang

et al., 2023). These five prompt methods were selected as they include different modalities in different positions with different structures covering most parts of the Oscar input. VPT applies visual prompts only, whereas CoOp applies textual prompts but both UPT and Maple include prompts for both visual and language modalities. The length of prompts can have a substantial impact on a model’s performance. In line with works in prior research on prompts (Lester et al., 2021; Li and Liang, 2021; Khattak et al., 2023), we experimented to assess the effects of prompt length too.

Fig.2 shows an overview of adding prompts to the Oscar input. The red boxes are the trainable vectors corresponding to the original design and the dotted boxes illustrate inputs that will be processed together. **VPT**: In the original VPT, the trainable prompts are directly added in front of the image patches, which are small parts of the image flattened to a vector. Fig.2 (a) shows how the prompts are added to the region features instead. As there are no image patches in Oscar, the trainable vectors are placed directly before the object region features in Oscar’s input.

**CoOp**: In CoOp, the prompts are added before the text class token. Therefore, in Oscar the trainable vectors were added before the object label tokens as shown in Fig.2 (b).

**UPT**: Original UPT includes the prompts for image patches and English tokens simultaneously (Zang

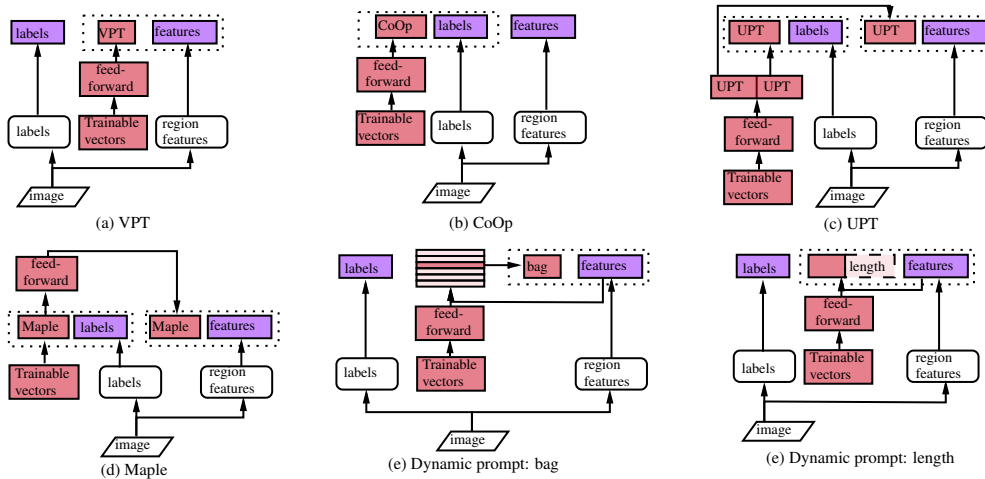


Figure 2: Adding prompts to Oscar: This diagram shows a detailed illustration of the input part of Fig.1. Each subgraph represents a method of adding the different prompts. The red part shows the trainable vectors. The elements in the dotted boxes will be processed together as the same type of modal input.

et al., 2022). To adapt UPT to work with Oscar, we first built a trainable vector that has twice the length of CoOp and VPT. After passing a fully-connected layer, the vector will be divided to two equal-length vectors and added to the region features and object labels separately as illustrated in Fig.2 (c).

**Maple:** Maple prompts are implemented in two parts (Khattak et al., 2023). The first one part fully-connects the text prompts to image patches. The second one fully connects the prompts in each language transformer block layer to the prompts in each visual transformer block layer. Since Oscar is a single-stream transformer, and the prompts between two modalities are already fully-connected by the attention block, we did not implement the second part. We added trainable prompts on object labels, and then fully connected them to the visual features as illustrated in Fig.2 (d).

**Dynamic Prompt:** Dynamic Prompt offers a few choices for implementing prompts (Yang et al., 2023). We selected *dynamic length* and *bag of prompts* approaches as these two methods are distinct from the other prompt methods to be evaluated. Dynamic Prompts generally selects the best prompts as the prompt with the highest attention with the region features. In *dynamic length*, Fig.2 (f), attention between each vector in the prompt and the region features will be calculated by a small attention network. Only the vectors with an attention larger than a threshold will be kept. In *bag of prompts*, the length of the prompts is fixed and there are a set of prompt pools containing several prompts for selection. Fig.2 (e) shows an overview of this selection process.

### 3.3 Adding Parameter-Efficient Tuning

The parameter-efficient tuning methods, LoRA (Hu et al., 2021) and Adapter (Houlsby et al., 2019) were selected for implementation because they are the most widely used. LoRA and Adapter both introduce a set of parameters beside the main transformer block stream. LoRA extracts the low intrinsic rank within the attention block. Adapter inserts the adapter and layer normalization outside the attention block and in the transformer layer. We used the original designs from the LoRA and Adapt papers. Fig.1 shows where these modules were added into the Oscar structure.

## 4 EVALUATION

The primary objective was to assess the influence of five distinct prompts and two parameter-efficient methods on the performance of Oscar, a single-stream vision-language transformer, in the context of zero-shot image captioning.

Given the constraints imposed by our available GPU resources, our initial goal was to establish a baseline implementation of Oscar. Subsequently, we introduced prompt-based and parameter-efficient approaches to demonstrate improvements in comparison to the baseline performance.

In keeping with the original work that introduced Oscar (Li et al., 2020), the image captioning pre-training procedures were conducted using the COCO dataset (Chen et al., 2015). COCO dataset is widely used in Image Captioning performance evaluation (Alayrac et al., 2022; Li et al., 2023; Zhang et al.,

2022; Liu et al., 2023b) including Oscar. The evaluation of all approaches was carried out on the COCO validation set with 5,000 images.

The image captioning with prompts pre-training was executed for 1 epoch with a batch size of 256. The initial Oscar weights were obtained from the original model repository. The prompts were adapted and modified to align with the specifications of the Oscar model and the parameter-efficient modules were integrated as detailed above.

Performance assessment of image captioning, under the conditions of significantly reduced training resource requirements, was conducted using the same metrics as initially used for evaluating Oscar and other works (Li et al., 2020; Chen et al., 2015; Liu et al., 2023b), including the following:

- **Bleu:** Bleu (Papineni et al., 2002) serves as a prevalent metric in the field of machine translation, quantifying the presence of n-grams shared between the reference text and the generated output. In this context, Bleu4 evaluates the precision of 4-gram sequences within the captions produced by the model in comparison to the ground truth references.
- **CIDEr:** For each n-gram in both the reference sentence and the predicted sentence, CIDEr score (Vedantam et al., 2015) computes the Term Frequency-Inverse Document Frequency (TF-IDF) score. The final CIDEr score is determined by the cosine similarity between these sentences.

The aim was to identify the best prompt method for zero-shot learning on image captioning. The baselines included the model without prompts, and the model with two English prompts, the first used “This is” and the second used “a photo of”. These are the English prompts that have been used in previous work that evaluates prompts (Zhou et al., 2022b; Alayrac et al., 2022). We set the length of the prompt to be two, which means there are two trainable vectors used as prompts. We trained the prompts with the COCO dataset for one epoch.

The result are shown in Table.1. The column labelled *Prop of Original Parameters* shows the proportion of parameters used as compared with the number of parameters in the original Oscar. Given that the pre-trained Oscar model was used with no fine-tuning, these reflect the reduction in parameters that need to be trained for the image captioning task instead of fine-tuning the original Oscar model.

The method, CoOp+VPT, is a combination of these two individual methods adding prompts in both language and visual modalities. VPT significantly

outperforms other prompt methods across all evaluation metrics. In contrast, the straightforward English prompts have a competitive performance compared with other prompt methods. All the prompt methods that add prompts in both modalities adversely affected the image captioning performance. These include using VPT with CoOp but also UPT and Maple.

We also explored different prompt lengths, represented by the number of trainable vectors, to find the best length across the different methods using a grid search (between 1 and 20) for each method. As an illustration of this, for a parameter length of five for the VPT approach, we put five trainable vectors before the region features. The results are shown in Fig.3. We found that the length of one or two are the best length for most methods. Specifically, VPT and CoOp perform optimally with a prompt length of two, whereas Maple and CoOp+VPT achieve optimal results with a length of one. The bag of prompts method, on the other hand, performs best with a prompt length of three.

We then added LoRA and Adapter to three prompt methods to see the impact on performance. Typically such parameter efficient methods are used individually, but they have been used together previously (Zhang et al., 2022). The prompt length used was two and the training was done for 35 epochs, with no improvement in performance after 35 epochs. We did not carry out an extensive training strategy involving hundreds of epochs, as outlined in the original paper (Houlsby et al., 2019). This decision was driven by our objective to conserve training resources, as dedicating an extensive amount of time to training does not align with the core focus of our work. Furthermore, we observed that the performance ceased to exhibit significant improvement after 35 epochs of training.

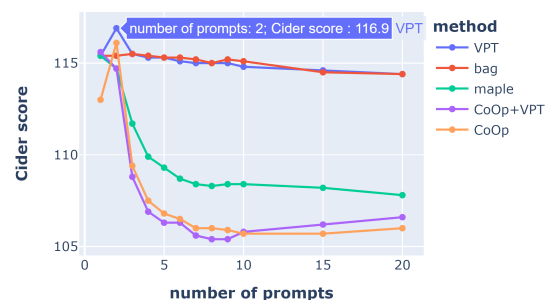


Figure 3: Comparison for different length of prompts.

The results are shown in Table 2. Including Adapter and LoRA with the prompt methods the Cider score of image captioning is further improved, particularly for Maple although VPT achieves a higher overall performance.

Table 1: Comparison of different prompt methods. The prompt type used is cross referenced to the diagram in Figure 2 which shows how it is used.

Method	Prompt Method	Cider	Blue4	Prop of Original Parameters
Baseline, no prompts		115.2	0.342	0%
Baseline, English prompt: "a photo of"		115.4	0.342	0%
Baseline, English prompt: "This is"		115.7	0.343	0%
Add 2 trainable prompts on object labels (b)	CoOp	116.1	0.344	1.367%
Add 2 trainable prompts on object features (a)	VPT	116.9	0.349	1.367%
Add 2 trainable prompts on object labels and features at the same time (a) & (b)	CoOp + VPT	114.7	0.338	2.020%
Add 4 trainable prompts and divide them into object labels and features at the same time (c)	UPT	114.5	0.337	1.369%
Add 2 trainable prompts on object labels, and then fully connect them to visual features (d)	Maple	114.7	0.339	1.367%
Add a bag of prompts for the 2 trainable prompts on object features to select (e)	bag of prompts	115.4	0.343	3.327%
Add dynamic length for the 2 trainable prompts on object features (f)	dynamic length	115.6	0.343	3.296%

The experimental results show that simply adding prompts to a single modality results in enhanced Cider scores. However, introducing prompts to both the visual and language features appears to have a detrimental effect on performance. The main contribution of this work lies in our demonstration of the capacity of prompts to enhance the Cider score in the context of image captioning for a single-stream transformer within the domain of zero-shot learning.

Table 2: Image captioning Cider score for prompts alone and for prompts combined with Adapter and LoRA.

Method	just prompts	with Adapter and LoRA
VPT	116.9	117.4
bag of prompts	115.4	115.9
Maple	114.7	116.3

Our results show that it is important to conduct empirical testing to determine the optimal number of prompts when employing a new model and dataset.

## 5 CONCLUSION

We conducted a comparative analysis on five prompt methods under the same condition to determine the most effective approach of enhancing the performance of image captioning using Oscar on the COCO dataset. VPT stands out as the optimal method of incorporating prompts to Oscar. Furthermore, our study reveals that the prompts length of two, indicating two trainable vectors used, yields the best performance when applied to Oscar and the COCO dataset.

The incorporation of additional parameter-efficient tuning methodologies has the potential to enhance the performance of zero-shot learning image captioning when used alongside prompts. With the addition of Adapter and Lora parameter-efficient approaches, we observed a 2% improvement in the Cider score for image captioning, achieved through just one epoch of retraining requiring just 5.7% of the parameters in training as compared with fine-tuning. In comparison to the full fine-tuning process, the utilization of prompts and parameter-efficient modules represents a substantial resource-saving approach when adapting to new downstream tasks.

Finally, we are the first work applying Adapter and LoRA with prompts on Oscar for image captioning to improve the zero-shot learning performance.

## ACKNOWLEDGEMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

## REFERENCES

Aghajanyan, A., Zettlemoyer, L., and Gupta, S. (2020). Intrinsic dimensionality explains the effectiveness

- of language model fine-tuning. *arXiv preprint arXiv:2012.13255*.
- Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. (2022). Flamingo: a visual language model for few-shot learning. *NIPS*, 35:23716–23736.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *NIPS*, 33:1877–1901.
- Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., and Luo, P. (2022). Adaptformer: Adapting vision transformers for scalable visual recognition. *NIPS*, 35:16664–16678.
- Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015). Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Floridi, L. and Chiriatti, M. (2020). Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694.
- Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., and Qiao, Y. (2023). Clip-adapter: Better vision-language models with feature adapters. *IJCV*, pages 1–15.
- Houlsby, N., Giurghi, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for nlp. In *ICML*, pages 2790–2799. PMLR.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. (2022). Visual prompt tuning. In *ECCV*, pages 709–727. Springer.
- Karimi Mahabadi, R., Henderson, J., and Ruder, S. (2021). Compacter: Efficient low-rank hypercomplex adapter layers. *NIPS*, 34:1022–1035.
- Khattak, M. U., Rasheed, H., Maaz, M., Khan, S., and Khan, F. S. (2023). Maple: Multi-modal prompt learning. In *CVPR*, pages 19113–19122.
- Lester, B., Al-Rfou, R., and Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.
- Li, J., Li, D., Savarese, S., and Hoi, S. (2023). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.
- Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al. (2020). Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, pages 121–137. Springer.
- Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.
- Liu, X., Delany, S. J., and McKeever, S. (2023b). Applying positional encoding to enhance vision-language transformers.
- Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., and Tang, J. (2021). P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.
- OpenAI, R. (2023). Gpt-4 technical report. *arXiv*, pages 2303–08774.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *NIPS*, 28.
- Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. (2022). Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Vedantam, R., Lawrence Zitnick, C., and Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575.
- Yang, X., Cheng, W., Zhao, X., Petzold, L., and Chen, H. (2023). Dynamic prompting: A unified framework for prompt tuning. *arXiv preprint arXiv:2303.02909*.
- Zaken, E. B., Ravfogel, S., and Goldberg, Y. (2021). Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.
- Zang, Y., Li, W., Zhou, K., Huang, C., and Loy, C. C. (2022). Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*.
- Zhang, Y., Zhou, K., and Liu, Z. (2022). Neural prompt search. *arXiv preprint arXiv:2206.04673*.
- Zhou, K., Yang, J., Loy, C. C., and Liu, Z. (2022a). Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825.
- Zhou, K., Yang, J., Loy, C. C., and Liu, Z. (2022b). Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348.