Towards Better Vision-Inspired Vision-Language Models

Yun-Hao Cao[1], Kaixiang Ji[2], Ziyuan Huang[2], Chuanyang Zheng[2],
Jiajia Liu[2], Jian Wang[2], Jingdong Chen[2], Ming Yang[2*]
[1] National Key Laboratory for Novel Software Technology, Nanjing University [2] Ant Group
caoyh@lamda.nju.edu.cn, {kaixiang.jkx, pishi.hzy, zhengchuanyang.zcy}@antgroup.com
{lekun.ljj, bobblair.wj, jingdongchen.cjd, m.yang}@antgroup.com

Abstract

Vision-language (VL) models have achieved unprecedented success recently, in which the connection module is the key to bridge the modality gap. Nevertheless, the abundant visual clues are not sufficiently exploited in most existing methods. On the vision side, most existing approaches only use the last feature of the vision tower, without using the low-level features. On the language side, most existing methods only introduce shallow vision-language interactions. In this paper, we present a vision-inspired vision-language connection module, dubbed as VIVL, which efficiently exploits the vision cue for VL models. To take advantage of the lower-level information from the vision tower, a feature pyramid extractor (FPE) is introduced to combine features from different intermediate layers, which enriches the visual cue with negligible parameters and computation overhead. To enhance VL interactions, we propose deep vision-conditioned prompts (DVCP) that allows deep interactions of vision and language features efficiently. Our VIVL exceeds the previous state-of-the-art method by 18.1 CIDEr when training from scratch on the COCO caption task, which greatly improves the data efficiency. When used as a plug-in module, VIVL consistently improves the performance for various backbones and VL frameworks, delivering new state-of-the-art results on multiple benchmarks, e.g., NoCaps and VQAv2.

1. Introduction

Deep learning has greatly transformed computer vision (CV) and natural language processing (NLP) and led to state-of-the-art results on a series of tasks [3, 8, 9, 15, 49]. The rapid progress in single-modality domains inspires soaring interests in investigating how to join multiple modalities in real-word applications like image captioning [7] and visual question answering [12]. Thus, vision-language (VL) models emerge as the new frontline for both communities.

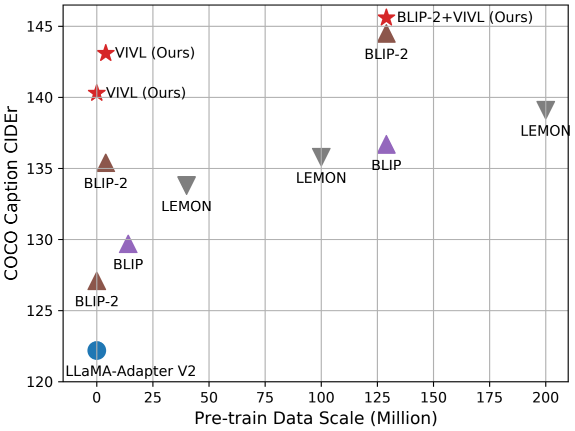
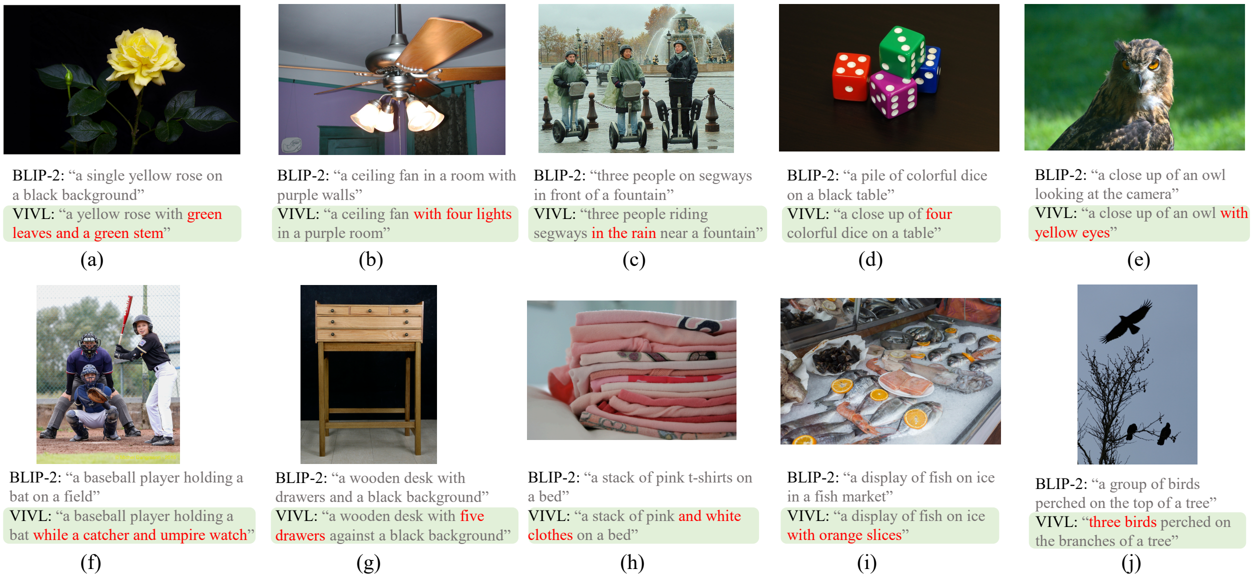


Figure 1. Image captioning performance on COCO when pre-trained using different dataset sizes. Our VIVL significantly exceeds the previous state-of-the-art results [11] on COCO caption without using pre-training data, which is even comparable to the methods that rely on a large amount of pre-training data [17, 23, 24], which demonstrates the data efficiency of our method.

The current mainstream methods [23, 38, 40] use a connection module to bridge a pre-trained image encoder and a pre-trained large language model (LLM) since this is a practical way to reuse both models with manageable training overhead and deliver fairly good performance [2, 23]. The connection module bridges the modality gap which is the key to exerting the capabilities of pre-trained unimodal vision and language models. Many efforts have been made to design the connection module: BLIP-2 [23] designs a BERT-based Q-Former, LLaVA [30] uses a simple fully connected layer, and Flamingo [2] inserts cross-attention layers into LLM. While, these connection modules have not fully take the advantage of abundant visual features in terms of both visual feature extraction and vision-language interaction.

On the image encoder side, previous works [2, 23, 30] primarily use the output from a single layer of the image encoder (*e.g.*, the last layer), without utilizing the early low-level features. In fact, many successful practices from object detection [14, 28] and semantic segmentation [5, 6] indicated

[*] Corresponding author.



BLIP-2: "a single yellow rose on a black background"
VIVL: "a yellow rose with green leaves and a green stem"
(a)

BLIP-2: "a ceiling fan in a room with purple walls"
VIVL: "a ceiling fan with four lights in a purple room"
(b)

BLIP-2: "three people on segways in front of a fountain"
VIVL: "three people riding segways in the rain near a fountain"
(c)

BLIP-2: "a pile of colorful dice on a black table"
VIVL: "a close up of four colorful dice on a table"
(d)

BLIP-2: "a close up of an owl looking at the camera"
VIVL: "a close up of an owl with yellow eyes"
(e)

BLIP-2: "a baseball player holding a bat on a field"
VIVL: "a baseball player holding a bat while a catcher and umpire watch"
(f)

BLIP-2: "a wooden desk with drawers and a black background"
VIVL: "a wooden desk with five drawers against a black background"
(g)

BLIP-2: "a stack of pink t-shirts on a bed"
VIVL: "a stack of pink and white clothes on a bed"
(h)

BLIP-2: "a display of fish on ice in a fish market"
VIVL: "a display of fish on ice with orange slices"
(i)

BLIP-2: "a group of birds perched on the top of a tree"
VIVL: "three birds perched on the branches of a tree"
(j)

Figure 2. Zero-shot test examples on the NoCaps [1] dataset. As highlighted in red, VIVL provides more fine-grained descriptions: including counting capabilities ((b), (d), (g), (j)), colors ((a), (e), (h)), details ((a), (b), (e), (g), (i)), and background descriptions ((c), (i)), etc.

that using multi-scale features can enhance the understanding of image details. Hence, we aim to explore multi-scale features to enrich visual features for VL models.

On the side of LLM, most existing methods [23, 30, 40] have explored shallow vision-language interactions. BLIP-2 [23], LLaVA [30] and other works [38, 40] send visual features as the prompt to the LLM's input layer, which limits the abundance of details in the responses due to its shallow interaction [18]. In contrast, Flamingo [2] enables deep interaction by inserting gated cross attention at each layer of LLM, at the cost of a sharp increase of number of parameters and floating-point operations (FLOPs). Hence, we aim to enable LLM to interact with visual features deeply in a parameter and computation efficient fashion.

Motivated by these, we propose a **V**ision-**I**nspired **V**ision-**L**anguage connection module (VIVL) that enables efficient interactions with richer vision features for LLM. To exploit multi-scale features, we present a feature pyramid extractor (FPE) that provides fine-grained visual cue for VL models, where higher-level features interact with lower-level features in a bottom-up way. To enhance VL interactions, we present deep vision-conditioned prompts (DVCP) by conditioning the prompts on both visual clues and previous outputs. Further, DVCP reduces the number of parameters by sharing weights and FLOPs by using a skip layer strategy. Last but not least, the proposed VIVL is designed for flexible deployment, which could be used as a stand-alone module or readily combined with existing methods such as BLIP-2.

As a stand-alone vision-language bridge, our VIVL significantly reduces the amount of data required for achieving a strong performance. As shown in Fig. 1, VIVL exceeds the previous state-of-the-art (SOTA) method [11] by 18.1 CIDEr when training the connection module from scratch on the COCO caption task [7], which is even comparable to the methods that rely on large-scale pre-training. When serving as a plug-in module, VIVL improves BLIP-2 by 3.4% on the VQAv2 task [12] and LLaVA by 0.6% on the Science QA task [34]. Extensive experiments demonstrate that VIVL brings consistent improvements across different backbones, VL frameworks, and datasets and achieves SOTA results on multiple benchmarks. Qualitative results in Fig. 2 also show that VIVL provides more detailed image descriptions.

2. Related Works

Vision-Language Models. Vision-language pre-training (VLP) aims to learn multimodal foundation models improving the overall performance on a variety of vision-and-language tasks, which has become the highlight of CV research recently. One line of works [36, 41] train both vision and language models from scratch in an end-to-end fashion, which can incur a high computation cost as the increase of model size. Recently, another line of works [2, 23, 38, 48] leverage off-the-shelf pre-trained models and keep them frozen during VLP. LiT [46] utilized a frozen pre-trained image encoder to accelerate CLIP [36] training. Frozen [38] fine-tuned an image encoder and transforms its outputs into LLM's soft prompts. Flamingo [2] inserted new cross-attention layers into pre-trained LLM to inject visual features. BLIP-2 [23] connected pre-trained image encoders and LLMs with a Q-Former. Follow-up works Mini-GPT4 [40] and LLaVA [30] used a linear layer

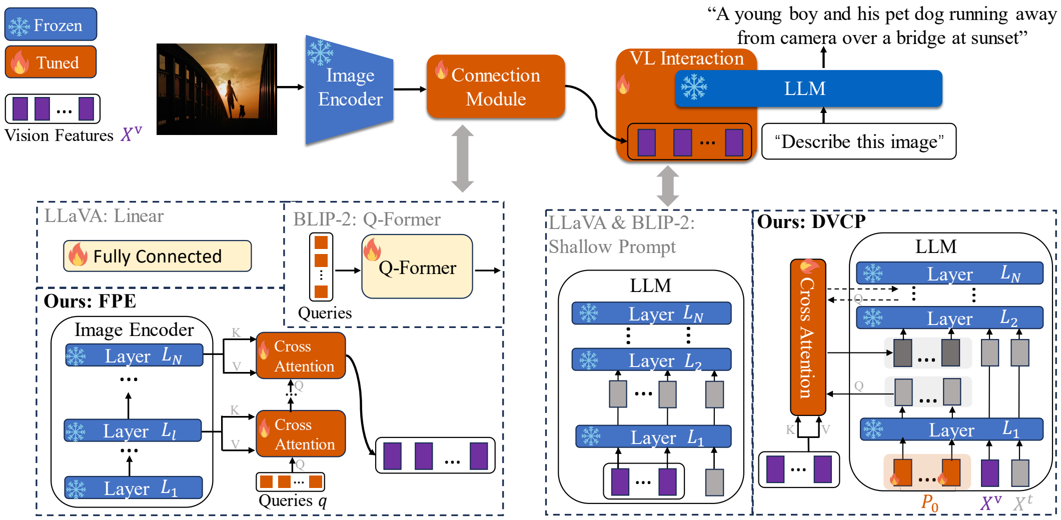


Figure 3. Illustration of our VIVL method, which is composed of FPE and DVCP. VIVL can be used independently (*i.e.*, with no dependency on other pre-trained connection modules), or it can be seamlessly embedded into different frameworks (*e.g.*, BLIP-2 [23] and LLaVA [30]).

to bridge the two modalities, while using more powerful LLMs and well-designed instruction fine-tuning. Our work also falls to the second line of work. In contrast, we (1) utilize the intermediate layer features of the visual encoder to enrich image details; (2) and propose a DVCP method to allow efficient visual-textual interaction at deep levels.

Prompt Tuning. Prompting [31] refers to prepending language instruction to the input text so that a pre-trained LLM can understand the task. With manually chosen prompts, GPT-3 [3] shows strong generalization to downstream transfer learning tasks even in the few-shot or zero-shot settings. Follow-up works proposed to treat the prompts as task-specific continuous vectors and directly optimize them via gradients during fine-tuning, namely Prompt Tuning [22, 27, 32]. Recently, prompting has also been applied to VL models [19, 36, 44, 51]. CoOp [52] applied prompt tuning to CLIP [36]. CoCoOp [51] pointed out that CoOp lacks in generalization to out-of-distribution data, and proposed to alleviate the problem by conditioning the prompt on image inputs. However, previous works mainly have focused on classification tasks using the dual-encoder architectures like CLIP, while few works study generation tasks using the encoder-decoder architectures like BLIP-2 [23]. Moreover, the prompts of previous works [18, 27, 51] did not fully utilize visual clues and the information encoded in the output of previous layers, as shown in Table 1. In this paper, we focus on generative VL models and propose a deep vision-conditioned prompt that can dynamically adapt based on both the visual cue and previous output. Moreover, our method is an efficient deep prompt method, where the number of parameters is largely reduced by sharing weights with manageable computational overhead by skipping layers.

Table 1. Comparisons between different prompt-tuning methods. $\mathcal{O}(1)$ denotes that the number of additional parameters will not increase with the number of network layers N, i.e., remain constant.

Method	Deep Prompt?	Conditioned On Vision?	Conditioned On Output?	Additional Params
P-Tuning [27]	×	×	×	$\mathcal{O}(1)$
CoCoOp [51]	×	✓	×	$\mathcal{O}(1)$
P-TuningV2 [32]	✓	×	×	$\mathcal{O}(N)$
VPT-Deep [18]	✓	×	×	$\mathcal{O}(N)$
Flamingo [2]	×	✓	✓	$\mathcal{O}(N)$
LLaMA-Adapter [48]	✓	✓	×	$\mathcal{O}(N)$
DVCP (Ours)	✓	✓	✓	$\mathcal{O}(1)$

3. Method

We begin with the preliminaries in Sec. 3.1 and then introduce our Feature Pyramid Extractor (FPE) in Sec. 3.2 and Deep Vision-Conditioned Prompt (DVCP) in Sec. 3.3. The overall framework of our VIVL is illustrated in Fig. 3.

3.1. Preliminaries

Transformer. For an N-layer transformer [39], we denote the token length as M and the latent dimension as d. Then, we denote the collection of token embeddings, $\boldsymbol{E}_{i-1} \in \mathbb{R}^{M \times d}$, as inputs to the i-th Transformer layer L_i. The whole Transformer is formulated as:

$$\boldsymbol{E}_i = L_i(\boldsymbol{E}_{i-1}), \qquad i = 1, 2, \ldots, N \tag{1}$$

where $\boldsymbol{E}_0$ denotes the input embeddings.

Notice that most existing VL models [23, 30, 40] send the extracted vision features $\boldsymbol{X}^{\mathrm{v}}$ together with text embeddings $\boldsymbol{X}^{\mathrm{t}}$ into the pre-trained LLM to obtain the final output:

$$\boldsymbol{E}_0 = [\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{X}^{\mathrm{t}}], \tag{2}$$

where $[\cdot, \cdot]$ indicates concatenating along the sequence length dimension. Our VIVL also follows Eq. (2) for the composition of $\boldsymbol{E}_0$ in the LLM, as shown in Fig. 3.

Prompt Tuning. According to the position where the prompt is inserted, it can be divided into shallow prompt [27] and deep prompt [32]. We only study deep prompt here due to its promising performance [18, 32]. Given a pre-trained language model, we introduce a set of K continuous embeddings of dimension d, *i.e.*, prompts, in the input space at each layer. For the i-th layer L_i, we denote the collection of input learnable prompts as $\boldsymbol{P}_{i-1} \in \mathbb{R}^{K\times d}$. As shown in Fig. 5 (a), the deep-prompted Transformer is formulated as:

$$[\boldsymbol{Z}_i, \boldsymbol{E}_i] = L_i([\boldsymbol{P}_{i-1}, \boldsymbol{E}_{i-1}])\,, \qquad i = 1, \cdots, N \tag{3}$$

where $\boldsymbol{Z}_i \in \mathbb{R}^{K\times d}$ denotes the output corresponding to $\boldsymbol{P}_{i-1}$ computed by the i-th layer and will be replaced by the corresponding prompt of the next layer, $\boldsymbol{E}_0$ denotes the input embeddings. Recall that $[\cdot, \cdot]$ indicates the concatenating operation and we have $[\boldsymbol{P}_{i-1}, \boldsymbol{E}_{i-1}] \in \mathbb{R}^{(K+M)\times d}$. Learnable parameters are colored in red.

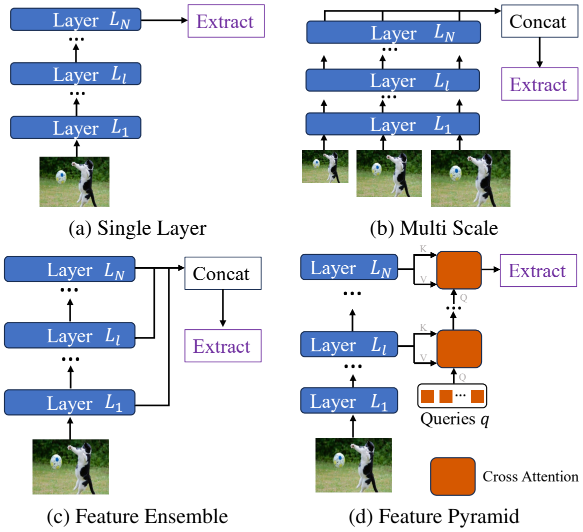
3.2. Feature Pyramid Visual Extractor

As mentioned before, previous works like BLIP-2 [23], Flamingo [2] and LLaVA [30] only used the output of the last layer (or the penultimate layer) of the visual encoder (Fig. 4a), which have not exploited the fine-grained image features. We denote $\boldsymbol{X}_l$ as the feature from the l-th layer of the visual encoder, and the extracted visual embeddings $\boldsymbol{X}^{\text{v}}$ are obtained from $\boldsymbol{X}_N$ in these works. In this paper, we propose to utilize features from the intermediate layers to complement the representation for detailed image contents. As shown in Fig. 4c, a naive solution is to concatenate features from different layers together:

$$\boldsymbol{X}^{\text{v}} = [\{\boldsymbol{X}_l | l \in I\}]\,, \tag{4}$$

where I denotes the collection of indices of the selected layers and the right formula represents stacking all the elements in I along the sequence length dimension. This scheme may bring somewhat improvement, as shown in Sec. 4.6.1. Nonetheless, the granularity difference in semantics embedded in different layers is still not captured in this straightforward implementation in Eq. (4). Moreover, the increased sequence length of the concatenated features will will result in substantial computational overhead.

Inspired by FPN [28] which built feature pyramids for object detection [15], we propose a feature pyramid extractor (FPE) to extract features from the intermediate layers for vision-language models, as shown in Figure 4d. In FPE, the lower-level features interact with higher-level features via cross-attention and generate new features. We denote $\boldsymbol{q}$ as the learnable queries for intermediate features and set the sequence length of $\boldsymbol{q}$ to 32, which greatly lowers the computational overhead compared to Eq. (4). As an example, consider FPE using two features from two layers (the l-th layer and the last layer), and we have:



(a) Single Layer

(b) Multi Scale

(c) Feature Ensemble

(d) Feature Pyramid

Figure 4. (a) Recent Vision-Language models use only single-layer features. (b) [48] uses an image pyramid and features are computed on each of the image scales independently, which is not computational efficient. (c) An alternative is to directly concatenate the features from different layers. (d) Our proposed feature pyramid extractor captures more image contents while runs as fast as (a).

$$\begin{aligned} \boldsymbol{X}_l^{\text{FPE}} &= \text{Attn}(\boldsymbol{q}, \boldsymbol{X}_l, \boldsymbol{X}_l)\,, && (5)\\ \boldsymbol{X}^{\text{v}} &= \text{Attn}(\boldsymbol{X}_l^{\text{FPE}}, \boldsymbol{X}_N, \boldsymbol{X}_N)\,, && (6) \end{aligned}$$

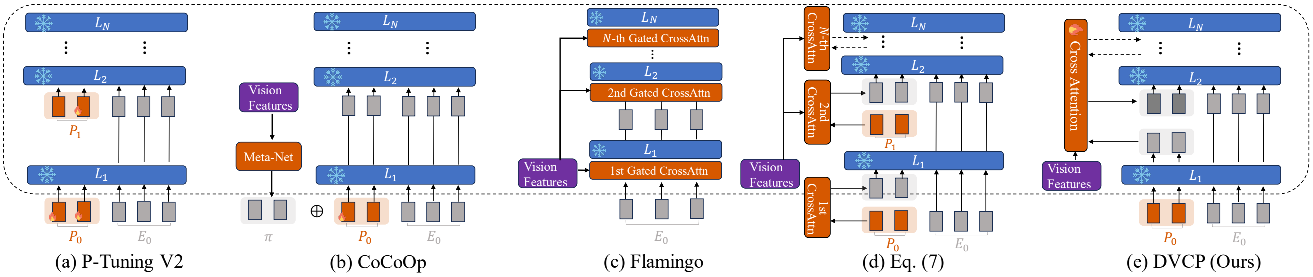
where $\text{Attn}(Q, K, V)$ denotes a cross-attention layer and Q, K and V represent query, key and value for the attention, respectively. Notice that we can also stack more attention layers for Eq. (5) and Eq. (6). Results in Sec. 4.2 show that one-layer cross-attention achieves the best accuracy-efficiency trade-off. Moreover, we can use features from more intermediate layers to construct the feature pyramid as in Eq. (4), where the results in Sec. 4.6.1 show that using more layers in FPE leads to further improvement.

3.3. Deep Vision-Conditioned Prompts

Unlike unimodal vision or language models, the performance of VL models will be affected if the prompts lack interaction with other modalities [51]. That is being said, if we only learn prompts $\boldsymbol{P}_i$ in LLM, this can play the role of Adaptaters [16], even though the visual clue is missing in the prompts. Hence, we propose to inject visual information to assist in generating prompts. As Fig. 5d shows, a feasible solution is to use cross-attention to introduce visual cue to the prompts of each layer, which can be formulated as:

$$[\boldsymbol{Z}_i, \boldsymbol{E}_i] = L_i\left(\left[A_i\left(\boldsymbol{P}_{i-1}, \boldsymbol{X}^{\text{v}}, \boldsymbol{X}^{\text{v}}\right), \boldsymbol{E}_{i-1}\right]\right)\,, \tag{7}$$

where $\boldsymbol{X}^{\text{v}}$ denotes the visual embeddings extracted from the visual encoder and A_i denotes the inserted cross-attention



(a) P-Tuning V2 (b) CoCoOp (c) Flamingo (d) Eq. (7) (e) DVCP (Ours)

Figure 5. Comparisons between our DVCP and other methods. (a) P-Tuning-v2 like methods [18, 32, 52]. (b) CoCoOp-style method [51]. (c) Flamingo-style method [2]. (e) Our proposed variant (Eq. (7)). (d) DVCP (Ours).

module at the i-th LLM layer (*i.e.*, 'Attn'). In this way, we can make the prompts $\boldsymbol{P}_i$ of each layer conditioned on the visual clue to help the language model "see" better. Nevertheless, Eq. (7) still has at least two problems: (1) The number of additional parameters increases linearly with the number of layers N, *i.e.*, an additional attention module A_i and learnable parameters P_i are required for each layer. (2) The prompts are unaware of the output of the previous layer.

DVCP-Plain. In this paper, we propose DVCP-plain to address the above two issues simultaneously. First, we *share* the cross-attention module A_i for all layers, which greatly reduces the amount of extra parameters. Second, we propose to replace $\boldsymbol{P}_i$ with the corresponding output of the previous layer. This benefits in two aspects: (1) the output of the previous layer encodes more information than $\boldsymbol{P}_i$; (2) this saves the number of parameters required for $\boldsymbol{P}_i$. Specifically, DVCP-plain can be formulated as follows:

$$[\boldsymbol{Z}_1, \boldsymbol{E}_1] = L_1([\boldsymbol{P}_0, \boldsymbol{E}_0])\,, \tag{8}$$

$$[\boldsymbol{Z}_i, \boldsymbol{E}_i] = L_i\left(\left[A\left(\boldsymbol{Z}_{i-1}, \boldsymbol{X}^{\text{v}}, \boldsymbol{X}^{\text{v}}\right), \boldsymbol{E}_{i-1}\right]\right),\ i \geq 2\,, \tag{9}$$

where A denotes the *shared* cross-attention layer across different layers in the language model. Actually, Eq. (9) can be seen as a deep prompt method [13] because the learnable A allows the prompts to dynamically change and be updated by gradient backpropagation *in each layer*. Compared with Eq. (7), DVCP-plain reduces the number of extra parameters by 96.7%. We will empirically show the advantages of DVCP-plain over Eq. (7) in Section 4.6.

DVCP-Skip. By designing a shared cross-attention module, the number of parameters is decreased significantly, while the computation cost remains the same. Therefore, we propose DVCP-skip to further reduce the computation, where we execute the module A every S layers and S is a hyper-parameter. We set $S = 5$ in this paper and DVCP-skip reduces extra FLOPs by 80% compared to DVCP-plain. The first layer of DVCP-skip is the same as Eq. (8). For $i \geq 2$, DVCP-skip can be formulated as:

$$[\boldsymbol{Z}_i, \boldsymbol{E}_i] = \begin{cases} L_i\left([\boldsymbol{Z}_{i-1}, \boldsymbol{E}_{i-1}]\right), & (i \bmod S) \neq 0 \\ L_i\left(\left[A\left(\boldsymbol{Z}_{i-1}, \boldsymbol{X}^{\text{v}}, \boldsymbol{X}^{\text{v}}\right), \boldsymbol{E}_{i-1}\right]\right), & \text{otherwise} \end{cases}.$$

Differences from other prompt-tuning methods. In Fig. 5 and Table 1, we compare DVCP with previous prompt methods in various dimensions. Compared with P-Tuning [27], CoCoOp [51] and LLaMA-Adapter [48], etc., our prompts are conditioned on both visual clues and previous outputs and has a constant number of parameters that will not change as the number of layers increases. Compared with Flamingo [2], DVCP reduces the number of parameters and computation from three aspects: (1) we share the weights of cross attention across layers; (2) we only operate on the tokens corresponding to the prompt $\boldsymbol{P}$, while Flamingo processes the whole tokens $\boldsymbol{E}$; (3) we only need to calculate on a small fraction of layers ($\frac{1}{S}$), attributed to our skip layer strategy.

4. Experimental Results

First, we introduce the implementation details in Sec. 4.1. Next, we use our VIVL independently and investigate the data efficiency in Sec. 4.2. After that, we combine our VIVL with the LLaVA framework and experiment on Science QA [34] in Sec. 4.3. Then, we apply VIVL to another framework BLIP-2 and experiment on COCO captioning [7] in Sec. 4.4 and VQAv2 [12] in Sec. 4.5. Then, we also study the transfer performance on NoCaps [1]. Finally, we study the effects of different components and hyper-parameters in our VIVL in Sec. 4.6. All experiments were conducted using PyTorch with 8 A100 GPUs.

4.1. Experimental Details

Backbones. We use CLIP ViT-g/14 [36] from EVA-CLIP [10] as the image encoder. For the frozen language model, we explore the unsupervised-trained OPT model family [49] for decoder-based LLMs and the instruction-trained FlanT5 model family [8] for encoder-decoder-based LLMs.

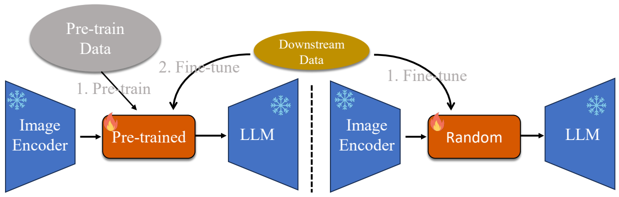
Training details. We use the AdamW [33] optimizer with a weight decay of 0.05 to train for 5 epochs. We use a cosine learning rate decay with an initial learning rate of 1e-5.

4.2. Data Efficiency of Our Method

As shown in Fig. 6a, current mainstream methods fine-tune downstream tasks based on pre-trained connection mod-

Table 2. Comparisons on COCO Caption when training connection modules from scratch. The original BLIP and BLIP-2 require pre-training (PT) on COCO Caption [7], Visual Genome [20], Conceptual Captions [4] and LAION [37], while ClipCap, LLaMA-Adapter V2, and ours only fine-tune (FT) the model on COCO.

Models	Data Scale		COCO Caption	
	PT	FT	BLEU@4	CIDEr
BLIP [24]	14M	0.6M	38.6	129.7
BLIP [24]	129M	0.6M	40.4	136.7
BLIP-2 [23]	129M	0	40.8	136.5
BLIP-2 [23]	129M	0.6M	43.7	145.8
ClipCap [35]	0	0.6M	33.5	113.1
LLaMA-Adapter V2 [11]			36.2	122.2
BLIP-2 [23]			37.4	127.1
VIVL (Ours)			**41.2**	**140.3**
BLIP-2 [23]	4M	0.6M	39.8	135.4
VIVL (Ours)			**42.5**	**143.1**



(a) Fine-tune from pre-trained bridge

(b) Train from scratch

Figure 6. Illustration of our train from scratch paradigm.

ules (*e.g.*, Q-Former in BLIP-2 [23]), which require large-scale pre-training to achieve image-text alignment. Notice that it is very time-consuming to pre-train different connection modules for different combinations of visual encoder and LLM. Moreover, using large-scale data for pre-training also forbids us from fast model iteration (*e.g.*, update a model in 10 minutes). Hence, in this section, we study the scenario where we train from scratch on downstream tasks with *a randomly initialized connection module*, as shown in Fig. 6b. To demonstrate the effectiveness of our VIVL, we only tune VIVL modules while keeping other parameters frozen in this section. As seen from Table 2, VIVL surpasses BLIP [24], even without pre-training on large-scale image-text data, and is comparable to the pre-trained BLIP-2. Moreover, VIVL has achieved *18.1 CIDEr gains* when compared to the latest method LLaMA-Adapter V2 [11]. The results demonstrate the data efficiency of our VIVL, attributed to our effective utilization of visual cue. It also shows that this train-from-scratch paradigm has a great potential.

Table 3. Comparisons of different bridge designs when training from scratch. A-k denotes a k-layer cross attention in (5) and (6).

Bridge	Token Length	Extra Params	COCO Caption BLEU@4	CIDEr
Linear [30]	256	3.6M	27.9	97.9
Q-Former [23]	32	107.1M	37.4	127.1
MLP (2 layers)	32	3.6M	34.0	119.6
A-1	32	3.4M	39.1	133.8
A-2	32	6.8M	38.7	132.6
A-6	32	20.1M	36.7	126.4
A-1	64	3.4M	39.8	134.5
A-1+FPE	64	6.7M	**41.0**	**139.8**

Within this paradigm, we only use downstream data to train the bridge modules, so we can study the impact of different bridge designs comprehensively. For fair comparisons, we use the same visual encoder (ViT-g) and LLM (OPT 2.7B) for all methods in Table 3. We can draw the following conclusions: (1) Cross-attention can extract visual features more effectively than MLP, although the number of parameters is comparable. (2) Increasing the number of parameters alone will not result in performance boost for sure due to the increased risk of overfitting. In fact, one layer of cross-attention achieves the best accuracy-efficiency trade-off. (3) Our FPE can capture more image contents so achieving +6.0 CIDEr gains. Note that the introduction of FPE will increase the token length (from 32 to 64), so for fair comparisons, we also set the token length to 64 for the baseline A-1 (the penultimate line). We can see that simply increasing the token length will not simply lead to significant performance improvements, indicating that the gain brought by FPE is beyond using more number of tokens.

4.3. Science QA

Our VIVL can be used independently as in Sec. 4.2, or in combination with other methods, such as LLaVA [30]. We insert FPE before the pre-trained linear layer and introduce DVCP to the LLM. In Table 4, we study the Science QA [34] dataset, which contains 21k multimodal multiple choice questions with rich domain diversity across 3 subjects, 26 topics, 127 categories, and 379 skills. We consider representative methods including LLaMA-Adapter [48], multimodal chain-of-thoughts (MM-CoT) [50], as well as LLaVA [30], which is the current SoTA method on this dataset. We conduct experiments upon LLaVA and train the model for 12 epochs. As shown in Table 4, our VIVL achieves 91.51% accuracy and yields a 1.24% absolute gain compared with LLaVA for image contexts. The results show that our VIVL can utilize visual information more effectively, and it is also applicable when migrating to other VL frameworks.

4.4. Image Captioning

Now we apply our VIVL to another framework BLIP-2 [23], where we insert FPE before the pre-trained Q-Former and introduce DVCP to the LLM. We fine-tune our models for the image captioning task, which asks the model to generate a text description for the image's visual content. Following BLIP-2, we keep the LLM frozen during fine-tuning and update the parameters of the Q-Former, our

Table 4. Results (accuracy (%)) on the Science QA dataset. Question classes: NAT = natural science, SOC = social science, LAN = language science, TXT = text context, IMG = image context, NO = no context, G1-6 = grades 1-6, G7-12 = grades 7-12.

Method	Subject			Context Modality			Grade		Average
	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	
Human [34]	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
GPT-3.5 [34]	74.64	69.74	76.00	74.44	67.28	77.42	76.80	68.89	73.97
GPT-3.5 w/CoT [34]	75.44	70.87	78.09	74.68	67.43	79.93	78.23	69.68	75.17
GPT-4 [30]	84.06	73.45	87.36	81.87	70.75	90.73	84.69	79.10	82.69
LLaMA-Adapter [48]	84.37	88.30	84.36	83.72	80.32	86.90	85.83	84.05	85.19
MM-CoT$_{\text{Base}}$ [50]	87.52	77.17	85.82	87.88	82.90	86.83	84.65	85.37	84.91
LLaVA [30]	90.36	95.95	88.00	89.49	88.00	90.66	90.93	90.90	90.92
LLaVA+VIVL (Ours)	91.43	95.39	88.55	90.76	89.24	90.87	91.81	90.97	**91.51**

Table 5. Comparison with the state-of-the-art image captioning methods on NoCaps and COCO Caption. For BLIP-2 we use ViT-g and FlanT5$_{\text{XL}}$ as the backbone.

Models	COCO Karpathy test		NoCaps Zero-shot	
	BLEU@4	CIDEr	SPICE	CIDEr
OSCAR [26]	37.4	127.8	11.3	80.9
VinVL [47]	38.2	129.3	13.5	95.5
BLIP [24]	40.4	136.7	14.8	113.2
OFA [41]	43.9	145.3	-	-
Flamingo [2]	-	138.1	-	-
SimVLM [43]	40.6	143.3	-	112.2
BLIP-2 [23]	42.4	144.5	15.8	121.6
BLIP-2+VIVL	42.7	**145.6** (+1.1)	15.8	**122.7** (+1.1)

Table 6. Comparison with the state-of-the-art models fine-tuned for visual question answering.

Models	#Trainable Params	VQAv2 val
Close-ended classification models		
VinVL [47]	345M	76.52
SimVLM [43]	∼1.4B	80.03
CoCa [45]	2.1B	82.30
BEiT-3 [42]	1.9B	84.19
Open-ended generation models		
ALBEF [25]	314M	75.84
BLIP [24]	385M	78.25
OFA [41]	930M	82.00
Flamingo80B [2]	10.6B	82.00
BLIP-2 ViT-g FlanT5$_{\text{XL}}$ [23]	1.2B	81.55
BLIP-2 ViT-g OPT$_{2.7\text{B}}$ [23]	1.2B	81.59
BLIP-2 ViT-g OPT$_{6.7\text{B}}$ [23]	1.2B	82.19
BLIP-2+VIVL ViT-g FlanT5$_{\text{XL}}$	1.2B	**84.84** (+3.29)
BLIP-2+VIVL ViT-g OPT$_{2.7\text{B}}$	1.2B	**85.00** (+3.41)

VIVL, together with the image encoder. We fine-tune on COCO [29] train set and evaluate on COCO test set and also zero-shot transfer to the NoCaps [1] validation set in Table 5. Take FlanT5$_{\text{XL}}$ as an example, our VIVL achieves +1.1 gains in terms of CIDEr on COCO caption compared with the baseline counterpart BLIP-2. More importantly, we further demonstrate that this is not overfitting on the current source dataset, as our VIVL also shows a consistent improvement when transferring the trained model to another dataset, i.e., NoCaps. We also achieve the *state-of-the-art results on NoCaps*, to the best of our knowledge.

4.5. Visual Question Answering

We continue to experiment based on BLIP-2 and switch to Visual Question Answering (VQA) tasks. Given annotated VQA data, we keep the LLM frozen while fine-tuning other parameters as done in Sec. 4.4. Table 6 demonstrates the *SOTA results* of our VIVL among open-ended generation models and our VIVL even surpasses previous SOTA results of close-ended classification method, *i.e.*, BEiT-3 [42]. When comparing the improvement over the baseline, we can see that our VIVL has a greater improvement in VQA than in caption in Table 5. There are two possible reasons for this. One is that richer text prompts (*e.g.*, questions) can more fully utilize the capabilities of our method. The second is that the VQA task has more training data (2.1M vs. 0.6M), allowing our randomly initialized modules to converge better. Hence, we believe that our VIVL can achieve better results under the condition of large-scale data pre-training.

4.6. Ablation Studies

In this section, we first study the different components in our VIVL, *i.e.*, FPE, and DVCP, as shown in Table 7. We keep the same settings as in Sec. 4.4. In addition to training and testing on the source dataset, we also transfer the model trained on the source dataset to a different dataset for zero-shot testing. We can draw the following conclusions: (1) Using FPE or DVCP alone brings improvements, and the combination of the two modules boost the performance contiguously. (2) The improvement brought by VIVL is also consistent when being transferred to other datasets, and shows a good generalization ability.

Then, in Sec. 4.6.1 and Sec. 4.6.2, we present ablation studies about FPE and DVCP on COCO caption and use BLIP-2 ViT-g OPT$_{2.7\text{B}}$ as the backbone. In order to study the effect of our connection module VIVL rigorously, in these following two subsections we only update the parameters of the connection module (*i.e.*, freeze the parameters of the image encoder and LLM), as done in Sec. 4.2. Hence, we

Table 7. Ablation Studies on caption datasets based on BLIP-2. We reproduce BLIP-2 on COCO caption for fair comparisons.

Backbone	FPE	DVCP	COCO Caption		Nocaps Zero-shot	
			BLEU@4	CIDEr	BLEU@4	CIDEr
OPT$_{2.7B}$	×	×	42.8	145.6	47.3	120.0
	✓	×	42.8	**145.8**	47.0	120.6
	×	✓	42.6	145.5	47.5	121.1
	✓	✓	42.6	145.6	**47.7**	**121.3**
FlanT5$_{XL}$	×	×	42.2	144.5	48.0	121.0
	✓	×	42.5	144.6	48.7	122.0
	×	✓	42.3	144.8	48.5	**123.0**
	✓	✓	**42.7**	**145.6**	**48.8**	122.7

Table 8. Ablation study of FPE. I denotes the collection of the indices of the selected layers, as in Eq. (4). 'Baseline' denotes BLIP-2 with frozen visual backbone during fine-tuning.

Method	Layer Index Set I	COCO Caption	
		BLEU@4	CIDEr
Baseline [23]	[38]	41.1	141.2
Feature Ensemble (Eq. (4))	[25, 38]	41.3	141.9
Attentional Pooling [21]	[25, 38]	41.9	142.0
FPE (Ours)	[15, 38]	41.6	142.1
	[25, 38]	**42.1**	**142.7**
	[35, 38]	41.8	142.4
	[37, 38]	41.2	141.6
	[15, 25, 38]	**42.3**	**143.1**

Table 9. Comparisons of prompting methods. 'Baseline' denotes BLIP-2 with frozen visual backbone during fine-tuning.

Method	Extra Params	COCO Caption	
		BLEU@4	CIDEr
Baseline [23]	0	41.1	141.2
P-Tuning [27]	0.1M	41.1	139.8
P-Tuning v2 [32]	2.5M	40.8	139.0
CoCoOp-style [51]	3.7M	40.9	141.6
LLaMA-Adapter-style [48]	3.6M	41.5	141.7
Deep-CoCoOp (Eq. (7))	5.9M	41.5	141.9
Flamingo-style [2]	103.8M	41.8	142.2
DVCP-Plain (Ours)	3.4M	41.5	142.4
DVCP-Skip (Ours)	3.4M	**41.9**	**142.9**

can see that the metrics for the baseline BLIP-2 in Table 8 and Table 9 are lower than those in Table 7, in which the parameters of the image encoder are also tuned.

4.6.1 Competitors of FPE

In this subsection, we will try to answer the following two questions: which layers to choose from the vision encoder, and how to aggregate the features from these selected layers. In Table 8, we investigate different combinations of selected layers in FPE and compare FPE with three baseline methods and we can reach the following conclusions:

- Intermediate layer features are useful. We can see that both the feature ensemble and our FPE outperform the baseline BLIP-2 (the first row) which only uses single-layer features.
- Our proposed FPE is more effective than the feature ensemble baseline (Fig. 4c). When $I = [25, 38]$, our method outperforms it by 0.8 points (142.7 vs. 141.9).
- The selected layers should not be too close or too far apart. When using two layers of features in FPE, $I = [25, 38]$ outperforms both $I = [15, 38]$ and $I = [37, 38]$. When the selected layers are far away, the large semantic gap will affect performance. When the selected layers are very close ($I = [37, 38]$ selects the last two layers), the improvement will be limited due to the lack of diversity.
- Using more layers of features in FPE further improves the results (the last row). Note that we choose $I = [25, 38]$ by default in our paper for better accuracy-efficiency trade-off. It also indicates that we may improve further with more carefully tuned I. In any case, utilizing intermediate layers has a better effect than not using them.

4.6.2 Competitors of DVCP

In Table 9, we compare DVCP with different prompting methods previously mentioned in Fig. 5 and Table 1. For fair comparisons, we set the length of the prompts to 32 for all these methods except for Flamingo-style [2], which does not involve prompts. From Table 9, we can observe:

- Simply increasing the number of parameters will not boost performance. For example, using P-tuning or P-tuning v2 in BLIP-2 even mess the results slightly. This also indicates that making prompts conditioned on modality inputs is very important for vision-language models.
- DVCP outperforms other methods (*e.g.*, CoCoOp, LLaMA-Adapter) that are unaware of previous outputs, proving the importance of our informative adaptation design.
- DVCP-Skip outperforms DVCP-Plain despite using fewer FLOPs (3.3 GFLOPs vs. 16.9 GFLOPs), which shows the effectiveness of our skip layer strategy.

5. Conclusions and Limitations

In this paper, we proposed VIVL for better vision-inspired vision-language models. On the vision side, we proposed a Feature Pyramid Extractor (FPE) to utilize vision features from different intermediate layers effectively. On the language side, we proposed Deep Vision-Conditioned Prompts (DVCP) to allow deep interaction of vision and language features efficiently. Our VIVL can be used independently or seamlessly embedded into other VL frameworks. Experimental results demonstrate the effectiveness of our method and we achieve state-of-the-art results on popular benchmarks including VQAv2 and NoCaps. Moreover, our VIVL only uses downstream data for training and may achieve better results if large-scale data is used for pre-training. However, due to our resource constraints, we cannot do these experiments at the moment and we will take this as future work.

References

[1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *The IEEE International Conference on Computer Vision*, pages 8947–8956, 2019. 2, 5, 7

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736, 2022. 1, 2, 3, 4, 5, 7, 8

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020. 1, 3

[4] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021. 6

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 1

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1

[7] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 1, 2, 5, 6

[8] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 1, 5

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, pages 1–12, 2021. 1

[10] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. 5

[11] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 1, 2, 6

[12] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 6325–6334, 2017. 1, 2, 5

[13] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, pages 1–12, 2022. 5

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *The IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 4

[16] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *The International Conference on Machine Learning*, pages 2790–2799, 2019. 4

[17] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 17980–17989, 2022. 1

[18] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *The European Conference on Computer Vision*, LNCS, pages 709–727. Springer, 2022. 2, 3, 4, 5

[19] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. *arXiv preprint arXiv:2112.04478*, 2021. 3

[20] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2017. 6

[21] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *The International Conference on Machine Learning*, pages 3744–3753, 2019. 8

[22] Brian Lester, Rami Al-Rfou, and Noah Constant. 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, 2021. 3

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1, 2, 3, 4, 6, 7, 8

[24] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *The International Conference on Machine Learning*, pages 12888–12900, 2022. 1, 6, 7

[25] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems*, volume 34, pages 9694–9705, 2021. 7

[26] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *The European Conference on Computer Vision*, LNCS, pages 121–137. Springer, 2020. 7

[27] Xiang Lisa Li and Percy Liang. 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, 2021. 3, 4, 5, 8

[28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 2177–2125, 2017. 1, 4

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *The European Conference on Computer Vision*, volume 8693 of *LNCS*, pages 740–755. Springer, 2014. 7

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 2, 3, 4, 6, 7

[31] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. 3

[32] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021. 3, 4, 5, 8

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *The International Conference on Learning Representations*, pages 1–10, 2019. 5

[34] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, volume 35, pages 2507–2521, 2022. 2, 5, 6, 7

[35] Ron Mokady, Amir Hertz, and Amit H. Bermano. Clipcap: CLIP prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 6

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *The International Conference on Machine Learning*, pages 8748–8763, 2021. 2, 3, 5

[37] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. *arXiv preprint arXiv:2111.02114*, 2021. 6

[38] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *Advances in Neural Information Processing Systems*, volume 34, pages 200–212, 2021. 1, 2

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017. 3

[40] Minigpt-4: Enhancing vision-language understanding with advanced large language models. Deyao zhu and jun chen and xiaoqian shen and xiang li and mohamed elhoseiny. *arXiv preprint arXiv:2304.10592*, 2023. 1, 2, 3

[41] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *The International Conference on Machine Learning*, pages 23318–23340, 2022. 2, 7

[42] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: BEIT pretraining for vision and vision-language tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 19175–19186, 2023. 7

[43] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *The International Conference on Learning Representations*, pages 1–14, 2022. 7

[44] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Cpt: Colorful prompt tun-

ing for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*, 2021. 3

[45] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 7

[46] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022. 2

[47] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. *arXiv preprint arXiv:2101.00529*, 2021. 7

[48] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 2, 3, 4, 5, 6, 7, 8

[49] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 1, 5

[50] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023. 6, 7

[51] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 3, 4, 5, 8

[52] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3, 5