ViPCap: Retrieval Text-based Visual Prompts for Lightweight Image Captioning

Taewhan Kim Soeun Lee Si-Woo Kim Dong-Jin Kim
Hanyang University, South Korea.
{taewhan, soeun, boreng0817, djdkim}@hanyang.ac.kr

Abstract

Recent lightweight image captioning models using retrieved data mainly focus on text prompts. However, previous works only utilize the retrieved data as text prompts, while the visual information relies only on the vision encoder. This leads to a limitation that the image descriptions in the prompt are not sufficiently reflected in the visual representations. To tackle this issue, we propose ViPCap, a novel retrieval text-based visual prompt for lightweight image captioning. ViPCap leverages the retrieved text with image information as visual prompts to enhance the ability of the model to capture relevant visual information. By mapping text prompts into the CLIP space and sampling from Gaussian distributions, we effectively retrieve semantic features containing image information. These retrieved features are integrated into the image and designated as the visual prompt, leading to performance improvements on the datasets such as COCO, Flickr30k, and NoCaps. Experimental results demonstrate that ViPCap significantly outperforms prior lightweight captioning models in efficiency and effectiveness, demonstrating the potential for a plug-and-play solution.

1 Introduction

Vision and language tasks, such as image captioning, have advanced with large-scale models [28, 4, 9, 13, 2, 4]. However, these models require high computational costs. To enhance training efficiency, recent works [24, 25, 29] primarily focus on prompt tuning and using retrieved text from datastore as text prompts. In contrast, there is a limitation in that visual information relies only on the vision encoder. As depicted in Fig. 1, SmallCap [25], which uses retrieval captions as text prompts without visual prompts, challenges to include detailed visual descriptions. We suspect this is because text descriptions are not effectively utilized as visual information.

Figure 1: SmallCap [25] fails to accurately capture local visual information present in the ground truth (GT) or retrieval text (Retrieval). In contrast, ViPCapeffectively captures visual details from both GT and retrieval text.

In this paper, we propose a retrieval text-based visual prompt for lightweight image captioning (**ViPCap**), leveraging retrieved texts with image information as *visual prompts*. First, given that the retrieved text provides a comprehensive image description, we encode the text prompt into the CLIP embedding space and transform it into patch-level hidden representations to extract semantic information. To effectively enhance local visual

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

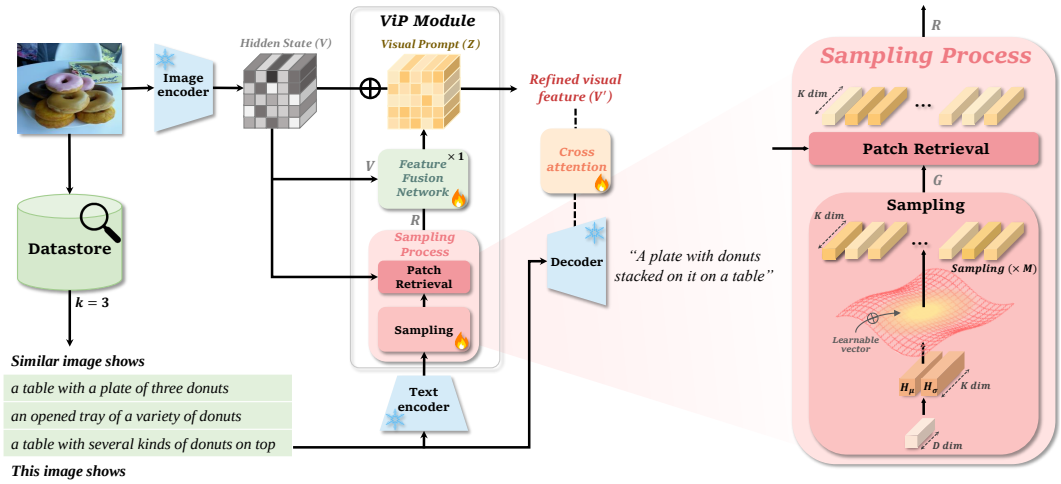

Figure 2: ViPCap leverages the CLIP text encoder to extract retrieval text features for visual prompts generation. The ViP module performs M sampling iterations G from the text embedding distribution to retrieve semantic features closely aligned with image patch features V. The ViP module samples M semantic vectors from the text embedding distribution. From the semantic vector set G, we retrieve semantic vectors closely aligned with image patch features V during patch retrieval. The retrieved semantic features R are fused with image features V within the Feature Fusion Network, and the resulting output is set as the visual prompt Z. Finally, the refined visual feature V' via summation with the visual prompt is fed to the decoder through the cross-attention layer.

representations using a global text representation, we assume that the embedding vectors follow a randomized Gaussian distribution and sample M semantic features from this distribution. Unlike the heuristic approach like CapDec and LinCIR [20, 7], which address the modality gap using Gaussian distributions, our method generates semantic features by sampling from a learnable distribution for a high correlation with visual features. We assume these semantic features contain visual information and expect them to closely resemble the input image features. Then, the retrieved semantic features are combined with image features to generate the *visual prompt*, which are added before decoder input. This approach aims to enhance the model's ability to capture relevant visual representations.

Our approach achieves superior performance on the COCO dataset [16] compared to our baseline model, SmallCap, and it significantly improves performance over previous lightweight models on the NoCaps dataset [1]. In the experiments, we integrate our ViP module into retrieval-based models, text-only training models, and various prompts, resulting in consistent performance improvements. Our contribution can be summarized as follows: (1) We propose a novel visual prompt for lightweight image captioning models named ViPCap, which leverages retrieved texts to generate visual prompts. (2) We introduce the ViP module, which retrieves semantic information from text features and combines it with image features to generate the visual prompt. (3) Extensive experiments demonstrate that our method is efficient and outperforms previous models across datasets like COCO and NoCaps, regardless of the text prompt types used.

2 Proposed Method

Our model adopts SmallCap as a baseline, which retrieves texts from an external datastore and connects the frozen encoder and decoder [22, 23] through trainable cross-attention layers.

As shown in Fig. 2, our work aims to extract semantic information from text prompts and generate it as visual prompts to enhance visual features. ViP module encodes retrieved texts into the CLIP embedding space and converts them into patch-level representations. Given retrieved texts T, our model encodes the retrieved text into D dimensional vector using pretrained CLIP text encoder $\phi(\cdot)$. Also, the CLIP image encoder embeds the input image into K dimensional visual features $V = \{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N\} \in \mathbb{R}^{N \times K}$ representing N number of patch-level visual features.

When generating visual prompts, a single text feature might be insufficient to provide the necessary details to generate a visual prompt with complex patch-level local information. To address this, we employ a random augmentation to sample semantic features from the Gaussian distribution. Also, instead of learning multiple mapping functions for local regions individually, we empirically find that sampling random vectors helps better match with visual local representations. We estimate the parameters of distribution of the text embedding $\vec{\mu}, \vec{\sigma} \in \mathbb{R}^K$ assuming it follows a multivariate Gaussian distribution $\mathcal{N}(\vec{\mu}, \vec{\sigma}^2 I)$. We design functions $\mathcal{H}_\mu(\cdot)$ and $\mathcal{H}_\sigma(\cdot)$ to map text features into the mean and standard deviation of multivariate Gaussian distribution. These functions are implemented as MLP layers to map from D dimensions to K dimensions while sampling the mean and standard deviation ($\mathcal{H} : \mathbb{R}^D \rightarrow \mathbb{R}^K$). We empirically find that adding an additional learnable vector $\vec{\omega_{add}}$ with a hyperparameter α as a scaling factor to the MLP shows better performance and captures complex data structures more effectively. The α is used to expand the range of the learnable vector. Let $\vec{\mu}$ and $\vec{\sigma}^2$ are computed via $\vec{\mu} = \mathcal{H}_m(\phi(T)) + \alpha \cdot \vec{\omega_{add}}$, and $\vec{\sigma} = \mathcal{H}_v(\phi(T))$, respectively.

ViP module samples M number of semantic features from this learnable Gaussian distribution to obtain semantic features that are highly correlated to the local visual embedding. We define the set of semantic representation $G \in \mathbb{R}^{M \times K}$ obtained from the text features as $G = \{\vec{g}_i \sim \mathcal{N}(\vec{\mu}, \vec{\sigma}^2 I; \phi(T))\}_{i=1}^M$. The $\vec{g}$ can be re-formulated by reparameterization trick [11].

2.1 Patch Retrieval Module for Semantic Features

We hypothesize that the semantic features $G = \{\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_M\}$ sampled from the Gaussian distribution contain the textual information describing the image. To effectively leverage G, we compare cosine similarity $\text{Sim}(\cdot, \cdot)$ between image patch-level features V and G to retrieve the most relevant semantic information for each patch $\vec{v}_i \in V$. From M candidates, we select N relevant vectors, one for each patch, to generate R:

$$R = \{\vec{g}_{\mathcal{I}(j)}\}_{j=1}^N \in \mathbb{R}^{N \times K}, \quad \text{where } \mathcal{I}(j) = \text{argmax}_{i \in [1:M]} \text{Sim}(\vec{g}_i, \vec{v}_j). \tag{1}$$

This simple calculation process extracts valuable information through R without any additional training.

2.2 Feature Fusion Network and Visual Prompts

As shown in Fig. 3, after obtaining R contained information relevant to the visual features, we integrate them with image feature to generate the visual prompt Z. We use a Feature Fusion Network (FFN) designed to combine V and R by transformer layers. Unlike previous networks with many layers, a single layer is enough since the features are already aligned.

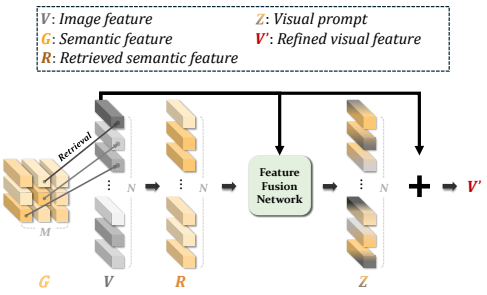

Figure 3: Similarity calculation between input feature V and semantic features G, where V retrieves essential representations from G and fuses them using the proposed fusion network. The fusion network generates visual prompt Z by integrating V, which is then combined with image features to produce refined visual features V'.

Finally, we obtain refined visual features V' with a simple summation ($V' = V + Z$). The FFN generates the visual prompt Z closely aligned with the distribution of image features, enabling the refined visual features through simple summation. This method allows models with vision encoders and language decoders to utilize refined features V' without modifying the decoder, making it compatible across different models using other encoders. The decoder takes the input text embedding, while V' is included conditionally when computing the loss function:

$$L_\theta = -\sum_{i=1}^{Q} \log P_\theta \left(y_i \mid y_{<i}, V'; \theta\right). \tag{2}$$

In the cross-attention layers (θ), weights are optimized by reducing the cross-entropy loss.

Method	Training Param	COCO Test				Flickr30k Test		NoCaps Val			
	θ	B@4	M	C	S	C	S	In	Near	Out	Entire
Large scale training models											
$\text{OSCAR}_{\text{Large}}$ [15]	338M	37.4	30.7	127.8	23.5	-	-	78.8	78.9	77.4	78.6
$\text{LEMON}_{\text{Huge}}$ [8]	675M	41.5	30.8	139.1	24.1	-	-	118.0	116.3	120.2	117.3
$\text{SimVLM}_{\text{Huge}}$ [28]	632M	40.6	33.7	143.3	25.4	-	-	113.7	110.9	115.2	112.2
$\text{BLIP2}_{\text{ViT-g OPT}_{2.7B}}$ [13]	1.1B	43.7	-	145.8	-	-	-	123.0	117.8	123.4	119.7
CogVLM [27]	1.5B	-	-	148.7	-	94.9	-	-	-	132.6	128.3
$\text{PaLI}_{\text{mT5-XXL}}$ [4]	1.6B	-	-	149.1	-	-	-	-	-	-	127.0
Lightweight models											
CaMEL [3]	76M	**39.1**	**29.4**	**125.7**	**22.2**	-	-	-	-	-	-
$\text{I-Tuning}_{\text{Medium}}$ [18]	44M	35.5	28.8	120.0	22.0	**72.3**	**19.0**	89.6	77.4	58.8	75.4
ClipCap [19]	43M	33.5	27.5	113.1	21.1	-	-	84.9	66.8	49.1	65.8
$\text{I-Tuning}_{\text{Base}}$ [18]	14M	34.8	28.3	116.7	21.8	61.5	16.9	83.9	70.3	48.1	67.8
SmallCap [25]	7M	37.0	27.9	119.7	21.3	60.6	-	87.6	78.6	68.9	77.9
$\text{SmallCap}_{\text{d=16, Large}}$ [25]	47M	37.2	28.3	121.8	21.5	-	-	-	-	-	-
ViPCap (Ours)	14M	37.7	28.6	122.9	21.9	66.8	17.2	**93.8**	**81.6**	**71.5**	**81.3**

Table 1: Comparison with large pre-training and lightweight models with existing methods on the COCO test, Flickr30k test, and NoCaps val set. CIDEr score is used for NoCaps evaluation. Our method shows the best performance in most metrics.

3 Experiments

3.1 Experimental Setup

Training dataset. We conduct experiments on image captioning benchmarks, i.e., COCO dataset [16], NoCaps [1], Flickr30k [21]. For COCO and Flickr30k, we follow the Karpathy split [10] used in the image captioning. We evaluate our model on the COCO and Flickr30k test set and NoCaps validation and test datasets, as well as the cross-domain experiments.

3.2 Main Results

In-domain. We evaluate ViPCap on COCO, Flickr30k, and NoCaps datasets in Table 1. On the COCO dataset, ViPCap outperforms in B@4 score compared to the large training models OSCAR while using only 4% of the parameters. Despite having 5 times fewer parameters than CaMEL [3], ViPCap achieves the second-highest CIDEr score among lightweight models. Additionally, ViPCap exceeds the baseline, SmallCap, and outperforms $\text{SmallCap}_{\text{Large}}$ (14M vs. 47M) on COCO, while also demonstrating strong performance on Flickr30k.

Cross-domain. On the Nocaps validation dataset, we achieve a CIDEr score of 93.8 on in-domain data, outperforming all lightweight models. This shows that ViPCap is highly suitable for real-world scenarios. Furthermore, it surpasses $\text{Oscar}_{\text{Large}}$ by over 10 points on in-domain data, indicating superior performance in entire domain.

3.3 ViP Module Capability

We explore different model sizes and prompt styles to evaluate the capabilities of our model.

Plug-and-Play manner. In Table 2, we evaluate the COCO, Flickr30k, and NoCaps test datasets to explicitly demonstrate the visual prompt ability by the ViP module. We combine our module with text-only training models. As a result, applying the ViP module to CapDec and ViECap [20, 6] improves performance, demonstrating that the ViP module can function as both a visual prompt and an image feature. The ViP module easily fuses with the vision encoders without modifying the framework. CapDec and ViECap with the ViP module result in an average increase of 3.5 points in CIDEr score performance on cross-domain in the NoCaps dataset. Our method shows the capability of ViP module in zero-shot tasks across real-world scenarios. We do not test with DeCap [14] due to its focus on memory efficiency, which does not align with our goals.

Method	In-Domain								Cross-Domain											
	COCO				Flickr30k				COCO ⇒ Flickr30k				Flickr30k ⇒ COCO				COCO ⇒ NoCaps			
	B@4	M	C	S	B@4	M	C	S	B@4	M	C	S	B@4	M	C	S	In	Near	Out	Entire
CapDec [20]	26.4	25.1	91.8	-	17.7	20.0	39.1	-	17.3	18.6	35.7	-	9.2	16.3	27.3	-	60.1	50.2	28.7	45.9
CapDec+ViP	27.0	25.6	94.2	18.8	18.6	20.1	44.4	14.4	15.7	18.0	35.8	11.8	9.5	16.3	30.7	9.2	60.2	50.9	33.7	47.8
Δ	**0.6**	**0.5**	**2.4**	-	**0.9**	**0.1**	**5.3**	-	**-1.6**	**-0.6**	**0.1**	-	**0.3**	**-**	**3.4**	-	**0.1**	**0.7**	**5.0**	**1.9**
ViECap [6]	27.2	24.8	92.9	18.2	21.4	20.1	47.9	13.6	17.4	18.0	38.4	11.2	12.6	19.3	54.2	12.5	61.1	64.3	65.0	66.2
ViECap+ViP	27.3	25.1	93.6	18.4	21.2	20.2	48.8	13.9	17.4	18.1	40.2	11.1	13.6	19.3	55.2	12.7	62.2	64.9	67.1	67.2
Δ	**0.1**	**0.3**	**0.7**	**0.2**	**-0.2**	**0.1**	**0.9**	**0.3**	**-**	**0.1**	**1.8**	**-0.1**	**1.0**	**-**	**1.0**	**0.2**	**1.1**	**0.6**	**2.1**	**1.0**

Table 2: In-domain and cross-domain performance of text-only models with the ViP module applied: Results on COCO, Flickr30k test sets, and NoCaps validation set. CIDEr scores for NoCaps evaluation. The ViP module consistently enhances performance across most metrics and base models, with notable improvements in cross-domain CIDEr scores.

Method	Enc.	Dec.	ViP	Ret	CIDEr
ViPCap (Ours)	ViT -B/32	OPT -125M	✓	×	122.0
				✓	122.5 (**0.5** ↑)
		XGLM		×	116.8
				✓	121.2 (**4.4** ↑)
EVCap	EVA-CLIP-g	Vicuna -13B	×	✓	140.1
			✓	✓	141.3 (**1.2** ↑)

Table 3: Performance improvements in CIDEr scores on COCO test using various decoders with the ViP module and retrieved text (**Ret**). Consistent improvements are observed across various encoders and decoders.

Prompt	ViP	
	×	✓
"This image shows"	111.1	116.0 (**4.9** ↑)
Retrieval prompt	117.3	119.9 (**2.6** ↑)

Table 4: CIDEr results of the ViP module on COCO val dataset show the potential of our prompt-agnostic model. Compared to the baseline, performance improvements observed with the ViP module when using "This image shows" as the input text prompt.

Model-agnostic. Table 3 reveals that combining the ViP module with OPT [30] or XGLM [17] as decoders, along with using both ViP and the retrieved text, leads to a notable improvement in performance. This indicates the capability as a model-agnostic and flexible framework. Additionally, similar to SamllCap, combining EVCap [12], which utilizes retrieval data, with the ViP module enhances performance. This means our approach can be effectively applied to models that use retrieval data, as well as large-scale models such as EVA-CLIP and Vicuna [26, 5], based on retrieved data. Therefore, ViP presents a consistent performance improvement when combined with SOTA models regardless of model size.

Prompt-agnostic. Table 4 compares ViPCap performance with and without the retrieval module. SmallCap scores 111.1 without retrieval and 117.3 with retrieval. ViPCap scores 116.0 without retrieval, using simple prompts like "This image shows ..." and 119.9 with retrieval. Additionally, in CapDec, ViECap, and EVCap, we observe notable results by leveraging hard prompts such as "a photo of" and "There are $entity$, ...". This demonstrates that ViPCap addresses competitive performance even with simple hard prompts and suggests its potential as a flexible visual prompt module applicable to various prompt types.

4 Conclusion

In this work, we introduce ViPCap, a novel approach that generates visual prompts by leveraging semantic information from retrieved text embedding through the ViP module. ViPCap performs well across both in-domain and out-of-domain datasets. The ViP module proposes a plug-and-play method that generates visual prompts based on various models and prompt types. Future work will explore using learnable tokens as visual prompts for better flexibility.

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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 8948–8957, 2019.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022.

[3] Manuele Barraco, Matteo Stefanini, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. Camel: Mean teacher learning for image captioning, 2022.

[4] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model, 2023.

[5] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[6] Junjie Fei, Teng Wang, Jinrui Zhang, Zhenyu He, Chengjie Wang, and Feng Zheng. Transferable decoding with visual entities for zero-shot image captioning, 2023.

[7] Geonmo Gu, Sanghyuk Chun, Wonjae Kim, Yoohoon Kang, and Sangdoo Yun. Language-only efficient training of zero-shot composed image retrieval, 2024.

[8] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning, 2022.

[9] Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A. Ross, and Alireza Fathi. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory, 2023.

[10] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions, 2015.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[12] Jiaxuan Li, Duc Minh Vo, Akihiro Sugimoto, and Hideki Nakayama. Evcap: Retrieval-augmented image captioning with external visual-name memory for open-world comprehension, 2024.

[13] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[14] Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. Decap: Decoding CLIP latents for zero-shot captioning via text-only training. In *The Eleventh International Conference on Learning Representations*, 2023.

[15] 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, 2020.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[17] Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. Few-shot learning with multilingual language models, 2022.

[18] Ziyang Luo, Zhipeng Hu, Yadong Xi, Rongsheng Zhang, and Jing Ma. I-tuning: Tuning frozen language models with image for lightweight image captioning, 2023.

[19] Ron Mokady, Amir Hertz, and Amit H. Bermano. Clipcap: Clip prefix for image captioning, 2021.

[20] David Nukrai, Ron Mokady, and Amir Globerson. Text-only training for image captioning using noise-injected clip, 2022.

[21] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, 2016.

[22] 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, 2021.

[23] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[24] Rita Ramos, Bruno Martins, and Desmond Elliott. Lmcap: Few-shot multilingual image captioning by retrieval augmented language model prompting, 2023.

[25] Rita Ramos, Bruno Martins, Desmond Elliott, and Yova Kementchedjhieva. Smallcap: Lightweight image captioning prompted with retrieval augmentation, 2023.

[26] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale, 2023.

[27] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models, 2024.

[28] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision, 2022.

[29] Zhuolin Yang, Wei Ping, Zihan Liu, Vijay Korthikanti, Weili Nie, De-An Huang, Linxi Fan, Zhiding Yu, Shiyi Lan, Bo Li, Ming-Yu Liu, Yuke Zhu, Mohammad Shoeybi, Bryan Catanzaro, Chaowei Xiao, and Anima Anandkumar. Re-vilm: Retrieval-augmented visual language model for zero and few-shot image captioning, 2023.

[30] 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, 2022.