EVCap: Retrieval-Augmented Image Captioning with External Visual–Name Memory for Open-World Comprehension

Jiaxuan Li[1*], Duc Minh Vo[1*], Akihiro Sugimoto[2], Hideki Nakayama[1]
[1]The University of Tokyo, Japan [2]National Institute of Informatics, Japan
{li,vmduc}@nlab.ci.i.u-tokyo.ac.jp sugimoto@nii.ac.jp nakayama@ci.i.u-tokyo.ac.jp

Abstract

*Large language models (LLMs)-based image captioning has the capability of describing objects not explicitly observed in training data; yet novel objects occur frequently, necessitating the requirement of sustaining up-to-date object knowledge for open-world comprehension. Instead of relying on large amounts of data and/or scaling up network parameters, we introduce a highly effective retrieval-augmented image captioning method that prompts LLMs with object names retrieved from **E**xternal **V**isual–name memory (*EVCap*). We build ever-changing object knowledge memory using objects' visuals and names, enabling us to (i) update the memory at a minimal cost and (ii) effortlessly augment LLMs with retrieved object names by utilizing a lightweight and fast-to-train model. Our model, which was trained only on the COCO dataset, can adapt to out-of-domain without requiring additional fine-tuning or re-training. Our experiments conducted on benchmarks and synthetic commonsense-violating data show that* EVCap*, with only 3.97M trainable parameters, exhibits superior performance compared to other methods based on frozen pre-trained LLMs. Its performance is also competitive to specialist SOTAs that require extensive training.*

1. Introduction

Advanced image captioning based on large language models (LLMs) [3, 8, 9, 25] has focused on the approach using big-scale models trained on ever-increasingly large-scale datasets, which is no longer viable. This is because the computational cost to train the models increases exponentially and, more importantly, updating training data is almost impossible to keep pace with the growth of novel objects in our daily lives. Sustaining ever-changing object knowledge with a reasonable cost is a pressing concern in LLMs-based models to truly unlock open-world comprehension.

Retrieval-augmented image captioning [20, 35] is emerging as an alternative since it considerably reduces training costs in both time and data while producing encouraging results. Nonetheless, with their huge datastore, it is obvious that LLMs would imitate the given texts, limiting their ability to describe open-world objects properly. For instance, SmallCap [35] considers the words *"skateboard"* and *"wooden floor"* to be a pair regardless of visual appearances containing a commonsense-violating pair

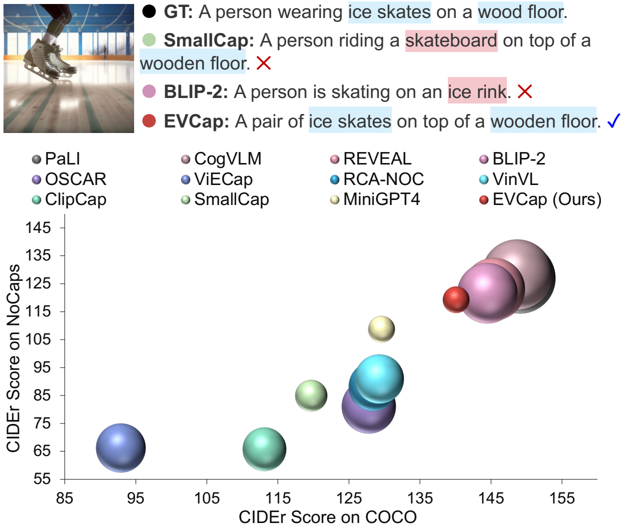


Figure 1. Overall comparison of our EVCap and SOTAs. (Upper) Generated captions by SmallCap, BLIP-2, and our EVCap for a commonsense-violating image from the WHOOPS dataset. ✗ and ✓ indicate incorrect and correct predictions, respectively. Incorrect objects in captions are highlighted in red, while correct ones are in blue. SmallCap and BLIP-2 give incorrect predictions for *"ice skates"* and *"wood floor"*, respectively, while our EVCap utilizes an external visual–name memory to enhance attention to objects within the image, leading to superior performance for image captioning. (Lower) Comparison of the number of trainable parameters, CIDEr score on COCO and NoCaps datasets. The size of each circle reflects the log number of trainable parameters. EVCap (3.97M) has less trainable parameters than others while achieving comparable results with SOTAs at scale.

[*]Equal contributions. Code is available at https://jiaxuan-li.github.io/EVCap.

of *"ice skates"* and *"wood floor"* (Fig. 1, upper). Additionally, prompting the LLMs given a lot of retrieved texts becomes cumbersome, requiring more trainable parameters. Fig. 1 (lower) shows that the CIDEr scores obtained by a lightweight SmallCap [35] with 43M trainable parameters are far away from those obtained by a heavy REVEAL [20] with 2.1B trainable parameters. Beyond that, due to the frequent occurrence of new objects, access to their sample texts is not always feasible, making the memory utilized in [20, 35] difficult to grow. We thus aim to streamline the external memory used in previous work [20, 35] by storing a sufficiently small amount of object information. And, of course, not only does the model not stereotype the example sentences, but the number of trainable parameters would be reduced drastically as a result of the causation (Fig. 1).

We follow [13, 40] to construct a key-value memory where the key is represented by object's features, and the value corresponds to object's name. Unlike [13, 40], which rely on object definition as the key, our method leverages the visual appearance of the object as the key because of the abundance of object images readily available on the internet. We propose an external visual–name memory tailored for ease of expansion and cost-effectiveness in upholding up-to-date object information. We present a highly effective retrieval-augmented LLMs-based image captioning method, called EVCAP, that prompts frozen LLMs with object names retrieved from our proposed memory for open-world comprehension. EVCAP contains a frozen image encoder ViT [14] and Q-Former [25] with *trainable* image query tokens for object retrieval, an attentive fusion module, a *trainable* linear layer for mapping between vision and language latent spaces, and a frozen LLM decoder [10] for generating captions. Specifically, the attentive fusion module feeds retrieved object names and visual features into a customized frozen Q-Former using *trainable* object name query tokens to implicitly reduce the presence of superfluous object names. As a result, EVCAP amounts to only 3.97M trainable parameters. Once trained, the model can be adapted to new domains and large-scale data without further fine-tuning or re-training. Our contributions are as follows:

- We provide an extensible external visual–name memory with minimal but useful object information, which enables LLMs-based models to comprehend the open world.
- We present a highly efficacious retrieval-augmented image captioning EVCAP with 3.97M trainable parameters.

On in-/out-domain benchmarks and synthetic commonsense-violating dataset, EVCAP trained solely on COCO dataset competes with other lightweight methods by a margin while being on par with other specialist SOTAs.

2. Related Work

Image captioning aims to describe the contents of a given image. It can be roughly divided into two approaches: non-LLMs-based methods and LLMs-based ones. The former approaches [4, 22, 42] typically employ a visual encoder and a language decoder in an end-to-end fashion to generate captions. However, they are incapable of describing open-world objects. The latter one leverages pre-trained large-scale vision models (CLIP [32], ViT [12]) and LLMs (GPTs [7, 31], T5 [33], LLaMA [37]) by bridging the gap between two modalities using either pre-training with large-scale data or the learned mapper or prompt techniques. LLMs-based models [8, 9, 25, 29] demonstrate advancements in image captioning challenges, allowing the capacity to describe anything as long as pre-trained vision models can recognize it. Our method belongs to the LLMs-based approaches, but instead of relying fully on the pre-trained vision model, we use object names retrieved from the external memory to augment LLMs-based image captioning.

Novel object captioning is a branch of image captioning that describes images containing objects that were not seen during training. Non-LLMs-based methods explore more objects by learning from unpaired image-sentence sources (DCC [19], NOC [39]) or rely on novel object detectors to recognize novel concepts (NBT [28], OSCAR [26] and VinVL [45]). LLMs-based methods such as ViECap [15] leverage the pre-trained CLIP [32] to obtain object entities Nevertheless, the cut-off in training time of the pre-trained object detector or CLIP prevents it from detecting novel objects that arise quickly in reality. Unlike earlier work, we can readily update our recognition of novel concepts by adding them to external memory, ensuring that we keep any new objects from the past and even the future.

Retrieval-augmented image captioning is a recently popular approach that augments the captioning model with retrieved information for better open-world understanding. AoANet [16] uses a memory bank of image-sentence pairs and target words. SmallCap [35] employs image-to-text retrieval to obtain sampled captions from a captions datastore. RA-CM3 [44] retrieves documents from an external memory of a mixture of text and image via a dense multimodal retriever. EXTRA [34] and Re-ViLM [43] exploit the similarity of the input image and vision candidates to retrieve captions. Different from the previous methods, our external memory contains visual–name pairs to avoid redundant information in the external captions/documents. In addition, we use an attentive fusion module to mitigate the effects of irrelevant retrieved object names on caption generation.

3. Proposed EVCAP

3.1. Idea of EVCAP

We aim to build a retrieval-augmented LLMs-based image captioning model with a sufficiently small yet informative external memory. It involves two challenges: (1) constructing an expandable external memory and (2) building an effective LLMs-based model using retrieved object names.

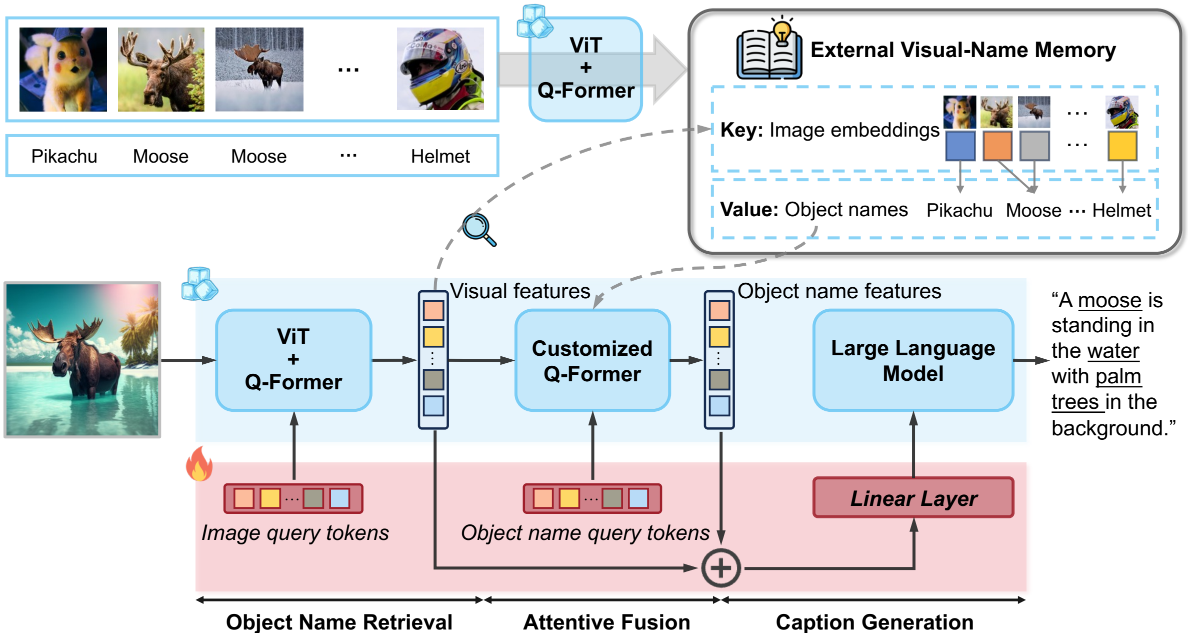


Figure 2. Schematic of our proposed EVCap. It consists of an external visual–name memory with image embeddings and object names (upper), a frozen ViT and Q-Former equipped with *trainable* image query tokens, an attentive fusion module developed by a customized frozen Q-Former and *trainable* object name query tokens, and a frozen LLM with a *trainable* linear layer (lower). The ViT and Q-Former extract learned visual features from the input image, which are then used to retrieve object names from the external memory. These retrieved object names and learned visual features undergo cross-attention in the customized Q-Former, creating refined object name features. Finally, the object name features combined with visual features are fed into the LLM post a linear layer for generating captions.

As discussed above, challenge (1) can be resolved by utilizing the visual appearance of objects. However, if we restrict our memory to only a visual–name pair for each object, our memory will be lacking in diversity. Therefore, we gather several images for each target object. Additionally, we keep the synthetic images in our memory to avoid the harm that synthetic images might cause to our method, as pointed out in [18]. With the capability to collect images from the internet, EVCap can be easily expanded to include novel objects from the real world effortlessly.

We base our method on the frozen pre-trained vision model and LLM with several trainable layers (Fig. 2), giving in a model that is cheap to train. To guide the LLM, we adopt a recently popular approach called prompting as in [11, 25, 29, 35, 46]. We begin by matching the learned visual features from the input image with image embeddings stored in memory, retrieving object names. We also introduce an attentive fusion module designed to implicitly remove irrelevant retrieved names. Finally, following the attentive fusion, we combine the learned visual features and object name features to form a prompt for the LLM to generate a caption, thus addressing challenge (2).

3.2. External visual–name memory

To build the external visual–name memory, we first collect image–name pairs from the external data source. After that, we encode these images into image embeddings, which serve as keys in memory, and use their names as values.

External data source. We utilize object images from LVIS dataset [17] to construct our external visual–name memory $\mathcal{M}$. Specifically, we use 1203 objects in LVIS, where we randomly select from one to ten images for each object, amounting to 8581 object images. Furthermore, as mentioned in Sec. 3.1, we also incorporate synthetic images in our memory construction. Using stable diffusion [36], we generate five additional images for each object, with a prompt of "*a photo of* $\{$*object name*$\}$", resulting in a total of $M = 14596$ $(8581 + 5 \times 1203)$ images. Each object image X^i is associated with an object name v^i. Note that many object images may share the same object name. For the sake of simplicity, we may regard each image as corresponding to a single name. In summary, we have M image–name pairs $\{(X^i, v^i)\}_{i=1}^{M}$ for external memory construction.

External memory construction. For each image X^i, we use a frozen vision encoder $\mathcal{E}(\cdot)$ (see Sec. 3.3 for detail) to project it into 32 embeddings with the size of 1×768 each: $\{\mathbf{k}_1^i, \mathbf{k}_2^i, \cdots, \mathbf{k}_{32}^i\} = \mathcal{E}(X^i)$. We then average 32 embeddings to produce a single embedding $\mathbf{k}^i$ (1×768) that serves as the key (visual) in $\mathcal{M}$. The paired object name v^i acts as its value (name). Consequently, we have the visual–name memory $\mathcal{M} = \{(\mathbf{k}^i, v^i)\}_{i=1}^{M}$ which is indexed using FAISS [21], facilitating rapid searches based on similarity measures. Our memory can be expanded effortlessly by gathering additional visual–name pairs (see Sec. 5.3).

3.3. Object names retrieval

Image encoding. We feed a frozen vision encoder $\mathcal{E}$ image X and image query tokens $\mathbf{T}_{\text{img}}$ to produce visual features $\mathcal{Q}$. To enable the retrieval process controllable, we make image query tokens to be trainable. Thus, the image encoding process can be summarized as $\mathcal{Q} = \mathcal{E}(X, \mathbf{T}_{\text{img}})$. We use the BLIP-2 pre-trained vision encoder [25], which consists of a pre-trained vision transformer ViT-g [14] outputting image features (257×1408), and a Q-Former receiving image features producing $|\mathcal{Q}| = 32$ learned visual features (1×768 each). We denote $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_{32}\}$.

Retrieval. Having obtained $\mathcal{Q}$, we calculate the cosine similarity between the query $\mathbf{q}_j \in \mathcal{Q}$ and the key $\mathbf{k}^i \in \mathcal{M}$. The similarity calculation is given by $\mathrm{SIM}(\mathbf{q}_j, \mathbf{k}^i) = \frac{\mathbf{q}_j^\top \mathbf{k}^i}{\|\mathbf{q}_j\|\|\mathbf{k}^i\|}$, where $i \in [1, M]$, $j \in [1, 32]$. Given each $\mathbf{q}_j$, we select one key with the highest similarity score, resulting in 32 key–value candidates $\{\mathbf{k}_j^{\text{best}}, v_j^{\text{best}}\}_{j=1}^{32}$.

After that, we filter out candidates with repeated object names (values), and then select the top-K values. In particular, we determine the index j from the key that has the highest SIM score. These selected values v_j^{best} are redefined as the new notation v_l in the retrieved top-K object names for the input image, which can be summarized as follows:

$$\{\mathbf{k}_j^{\text{best}}, v_j^{\text{best}}\} = \arg\max_{\mathbf{k}^i} \mathrm{SIM}\left(\mathbf{q}_j, \mathbf{k}^i\right),$$

$$j = \arg\max_{j} \mathrm{SIM}(\mathbf{q}_j, \mathbf{k}_j^{\text{best}}), v_l \leftarrow v_j^{\text{best}},$$

where $l \in [1, \mathrm{K}]$. As a result, the retrieved top-K object names are $\{v_l\}_{l=1}^{\mathrm{K}}$.

3.4. Attentive fusion

Since the object names obtained from the retrieval process may be redundant, we develop an attentive fusion module to selectively distill object name features.

The retrieved object names $\{v_l\}_{l=1}^{\mathrm{K}}$ are concatenated together into a sequence $\mathcal{S}$, each separated by a delimiter: $\mathcal{S} = \{v_1, \texttt{[SEP]}, v_2, \texttt{[SEP]}, \cdots, \texttt{[SEP]}, v_{\mathrm{K}}\}$. The sequence $\mathcal{S}$ and visual features $\mathcal{Q}$ are fed into a customized Q-Former $\mathcal{F}(\cdot)$, which is constructed from the frozen pre-trained Q-Former as we used in vision encoder $\mathcal{E}$. Nonetheless, in order to enable object names to get attention from visual features, we switch the image embedding port and the text instruction port (see the supplement for architecture detail). Like in the image encoding process in Sec. 3.3, we make the object name query tokens $\mathbf{T}_{\text{obj}}$ learnable during training to assist in learning object name features related to the caption. The size of $\mathbf{T}_{\text{obj}}$ is $P \times 768$, where P indicates the number of object name query tokens. We get the object name features $\mathcal{V} = \mathcal{F}(\mathcal{S}, \mathcal{Q}, \mathbf{T}_{\text{obj}})$.

3.5. Caption generation

Before inputting the visual features $\mathcal{Q}$ and object name features $\mathcal{V}$ into the LLM decoder, we concatenate ($\oplus$) them and use a linear layer $\phi(\cdot)$ to project them into the input latent space of the LLM as $\phi(\mathcal{Q} \oplus \mathcal{V})$. The LLM used for caption generation in this work is the pre-trained Vicuna-13B [10], an open-source chatbot constructed from LLaMA [37]. During training and evaluation, we design a prompt in a conversational format, that is similar to [46]:

```
###Human: <Img><ProjFeature></Img>
Describe this image in detail.
###Assistant:
```

in which, `ProjFeature` denotes the projected feature $\phi(\mathcal{Q} \oplus \mathcal{V})$ after the linear layer. In training phase, given input caption tokens $\{c_i\}_{i=1}^{L}$, the LLM decoder concatenates the embedded prompt $\{\mathbf{w}_i\}_{i=1}^{N}$ and the embedded caption tokens $\{\mathbf{c}_i\}_{i=1}^{L}$ as input, and predicts the caption tokens in an autoregressive fashion, while in the evaluation phase, we only need to input the embedded prompt. We train EVCap by minimizing the cross-entropy loss in an end-to-end way: $\mathcal{L}_\theta = -\sum_{i=1}^{L} \log p_\theta \left(c_i \mid \mathbf{w}_1, ...\mathbf{w}_N, \mathbf{c}_1, ..., \mathbf{c}_{i-1}\right)$, in which θ indicates the trainable parameters.

4. Experimental Settings

4.1. Training setup

Implementation. EVCap uses the same image encoder as in BLIP-2 [25], consisting of a ViT-g [14] and their pre-trained Q-Former. Since we intend to obtain object name features through cross-attention between retrieved object names and visual features, we develop a customized Q-Former, which consists of BERT [23] with cross-attention layers inserted at every other transformer block. We use a frozen Vicuna-13B [10] as the caption generator.

Training dataset. For all experiments, we exclusively train EVCap using the training set of **COCO** dataset [27], consisting of 82k images and 5 captions per images. The entire training process takes about 3 hours on 4 A6000 GPUs, using mixed precisions (more details in the supplementary).

4.2. Evaluation setup

Evaluation dataset. We evaluate EVCap, trained using the COCO training set, across four datasets: its test set, two challenging benchmarks – NoCaps validation set and Flickr30k test set, and a synthetic commonsense-violating dataset – WHOOPS. We adhere follow prior work [15, 41] to use the same images of Karpathy split [22] on **COCO** test set, **NoCaps** [2] validation set, and Karpathy split on **Flickr30k** [30] test set. In addition, **WHOOPS** [6] is a synthetic image captioning dataset comprising 500 synthetic commonsense-violating images and 2500 paired captions.

Table 1. Quantitative comparison against SOTA methods on three common image captioning benchmarks. * denotes using a **memory bank**. We report the size of training data and parameters; BLEU@4 (B@4), METEOR (M), CIDEr (C), and SPICE (S) scores on COCO test set; C and S scores on in-domain, near-domain, out-domain and overall data of NoCaps validation set; C and S scores on Flickr30k test set. Higher score is better. **Bold** indicates the best results among compared methods, normal indicates the second best results.

Method	Training		COCO				NoCaps val								Flickr30k	
	Data	Para.	Test				In-domain		Near-domain		Out-domain		Overall		Test	
			B@4	M	C	S	C	S	C	S	C	S	C	S	C	S
Heavyweight-training models																
VinVL [45]	8.9M	110M	38.2	30.3	129.3	23.6	96.8	13.5	90.7	13.1	87.4	11.6	90.9	12.8	–	–
AoANet+MA* [16]	COCO	–	38.0	28.7	121.0	21.8	–	–	–	–	–	–	–	–	–	–
NOC-REK* [40]	COCO	110M	–	–	–	–	104.7	14.8	100.2	14.1	100.7	13.0	100.9	14.0	–	–
RCA-NOC* [13]	COCO	110M	37.4	29.6	128.4	23.1	92.2	12.9	87.8	12.6	87.5	11.5	88.3	12.4	–	–
ViECap $_{\text{GPT2}}$ [15]	COCO	124M	27.2	24.8	92.9	18.2	61.1	10.4	64.3	9.9	65.0	8.6	66.2	9.5	47.9	13.6
InstructBLIP $_{\text{Vicuna-13B}}$ [11]	129M	188M	–	–	–	–	–	–	–	–	–	–	121.9	–	82.8	–
OSCAR [26]	4.1M	338M	37.4	30.7	127.8	23.5	83.4	12.0	81.6	12.0	77.6	10.6	81.1	11.7	–	–
BLIP [24]	129M	446M	40.4	–	136.7	–	114.9	15.2	112.1	14.9	115.3	14.4	113.2	14.8	–	–
BLIP-2 $_{\text{FlanT5-XL}}$ [25]	129M	1.2B	**42.4**	–	144.5	–	**123.7**	**16.3**	**120.2**	**15.9**	**124.8**	**15.1**	121.6	**15.8**	–	–
REVEAL* $_{\text{T5}}$ [20]	1.3B	2.1B	–	–	**145.4**	–	–	–	–	–	–	–	**123.0**	–	–	–
Lightweight-training models																
MiniGPT4 $_{\text{Vicuna-13B}}$ [46]	5M	**3.94M**	38.0	29.6	129.6	23.4	99.0	14.8	106.9	15.3	110.8	14.9	108.8	15.1	78.4	16.9
SmallCap* $_{\text{GPT2}}$ [35]	COCO	7M	37.0	27.9	119.7	21.3	–	–	–	–	–	–	–	–	60.6	–
ClipCap $_{\text{GPT2}}$ [29]	COCO	43M	33.5	27.5	113.1	21.1	84.9	12.1	66.8	10.9	49.1	9.6	65.8	10.9	–	–
EVCAP* $_{\text{Vicuna-13B}}$	COCO	3.97M	41.5	**31.2**	140.1	**24.7**	111.7	15.3	119.5	15.6	116.5	14.7	119.3	15.3	**84.4**	**18.0**
Specialist SOTAs																
Qwen-VL $_{\text{Qwen-7B}}$ [5]	1.4B	9.6B	–	–	–	–	–	–	–	–	–	–	121.4	–	85.8	–
CogVLM $_{\text{Vicuna-7B}}$ [41]	1.5B	6.5B	–	–	148.7	–	–	–	–	–	132.6	–	128.3	–	94.9	–
PaLI $_{\text{mT5-XXL}}$ [9]	1.6B	17B	–	–	149.1	–	–	–	–	–	–	–	127.0	–	–	–
PaLI-X $_{\text{UL2-32B}}$ [8]	2.2B	55B	–	–	149.2	–	–	–	–	–	–	–	126.3	–	–	–

Compared methods. We compare EVCAP with several SOTAs. According to the trainable parameters size, they can be divided into 1) Heavyweight-training (between 100M to 5B): VinVL [45], AoANet [16], NOC-REK [40], RCA-NOC [13], ViECap [15], InstructBLIP [11], OSCAR [26], BLIP [24], BLIP-2 [25], REVEAL [20]; 2) Lightweight-training (less than 100M): MiniGPT4 [46], SmallCap [35], ClipCap [29]; and also 3) Specialist SOTAs with huge trainable parameters (larger than 5B): Qwen-VL [5], CogVLM [41], PaLI [9], PaLI-X [8]. Among these methods, AoANet, NOC-REK, RCA-NOC, REVEAL, and SmallCap are retrieval-augmented captioning methods.

5. Experimental Results

5.1. Results on in-/out-domain benchmarks

We assess EVCAP against SOTAs on both in-domain and out-domain benchmarks. The COCO test set can be considered as in-domain data as we only train our model on the COCO training set. Out-domain benchmarks are the NoCaps validation set and the Flickr30k test set.

Quantitative results. Tab. 1 details our EVCAP's performance in comparison with SOTA methods. We first evaluate training costs in terms of training data sizes and parameters. Similar to various heavyweight-training models that exclude LLMs and the majority of lightweight-training models, EVCAP is trained solely on the COCO training set. It utilizes only 3.97M trainable parameters, positioning it as the second smallest, slightly larger than MiniGPT4 with 3.94M. Among lightweight-training models, our approach outperforms others, achieving the highest scores on all benchmarks. Despite using less training data and nearly identical trainable parameters as MiniGPT4, EVCAP significantly surpasses it, with a marked improvement of 10.5, 10.5, and 6.0 in CIDEr scores for each benchmark. When further compared with heavyweight-training models, the performance of EVCAP stands out among million-level models, nearly matching InstructBLIP, except in NoCaps. Note that since BLIP-2 does not include Vicuna checkpoints, InstructBLIP performs pre-training with Vicuna using the same procedure as BLIP-2, whereas EVCAP does not involve pre-training. Against REVEAL, which also uses external memory, our EVCAP utilizes about 1/3000 training data and 1/500 training parameters yet yields comparable results. Moreover, EVCAP's performance is on par with BLIP-2, the top-performing model with 1.2B trainable parameters. This highlights EVCAP's efficiency and effectiveness despite its significantly smaller training cost, thanks to our external visual–name memory. Regarding specialist SOTAs, they use billion-level training data and over 5B trainable parameters, so it is acceptable that they can achieve exceptionally strong performance, surpassing EVCAP by nearly 10 on all benchmarks in CIDEr scores.

Qualitative results. Fig. 3 presents a comparison of captions generated by our EVCAP and three SOTA models across three benchmarks. The captions of SmallCap are generated by its publicly accessible demo [1]. We generate captions of MiniGPT4 and BLIP-2 using their respec-

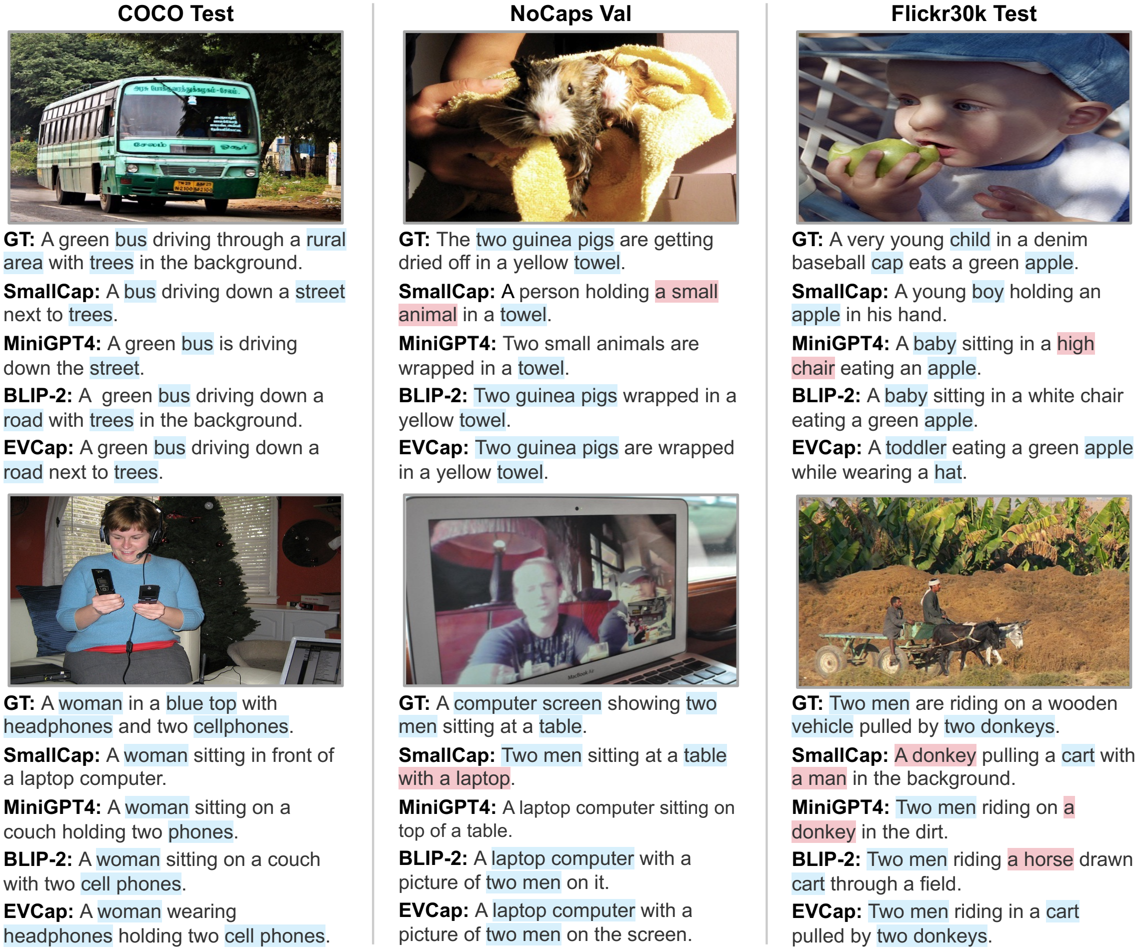

COCO Test
NoCaps Val
Flickr30k Test
GT: A green bus driving through a rural area with trees in the background.
SmallCap: A bus driving down a street next to trees.
MiniGPT4: A green bus is driving down the street.
BLIP-2: A green bus driving down a road with trees in the background.
EVCap: A green bus driving down a road next to trees.
GT: The two guinea pigs are getting dried off in a yellow towel.
SmallCap: A person holding a small animal in a towel.
MiniGPT4: Two small animals are wrapped in a towel.
BLIP-2: Two guinea pigs wrapped in a yellow towel.
EVCap: Two guinea pigs are wrapped in a yellow towel.
GT: A very young child in a denim baseball cap eats a green apple.
SmallCap: A young boy holding an apple in his hand.
MiniGPT4: A baby sitting in a high chair eating an apple.
BLIP-2: A baby sitting in a white chair eating a green apple.
EVCap: A toddler eating a green apple while wearing a hat.
GT: A woman in a blue top with headphones and two cellphones.
SmallCap: A woman sitting in front of a laptop computer.
MiniGPT4: A woman sitting on a couch holding two phones.
BLIP-2: A woman sitting on a couch with two cell phones.
EVCap: A woman wearing headphones holding two cell phones.
GT: A computer screen showing two men sitting at a table.
SmallCap: Two men sitting at a table with a laptop.
MiniGPT4: A laptop computer sitting on top of a table.
BLIP-2: A laptop computer with a picture of two men on it.
EVCap: A laptop computer with a picture of two men on the screen.
GT: Two men are riding on a wooden vehicle pulled by two donkeys.
SmallCap: A donkey pulling a cart with a man in the background.
MiniGPT4: Two men riding on a donkey in the dirt.
BLIP-2: Two men riding a horse drawn cart through a field.
EVCap: Two men riding in a cart pulled by two donkeys.

Figure 3. Examples of captions generated by our EVCap and three SOTA methods on COCO test set, NoCaps validation set, and Flickr30k test set. GT refers to the Ground Truth captions. Incorrect objects in captions are highlighted in red , while correct ones are in blue . Our EVCap correctly generates captions across different datasets, showing performance comparable to BLIP-2.

tive pre-trained models. As a lightweight and retrieval-augmented captioning method, SmallCap struggles to produce accurate captions for given images, primarily because it relies on retrieved captions laden with extraneous information. MiniGPT4, though aligned with the primary content of images, sometimes misses certain objects like *"trees"* and *"headphones"*. This oversight stems from its focus on the main objects in images, without integrating additional cues for other objects provided by the retrieved object names. In contrast, the captions generated by our EVCap are comparable to those of BLIP-2.

5.2. Results on commonsense-violating data

To explore our EVCap's capability in describing contents in open-word settings, we further evaluate it on WHOOPS dataset, which contains commonsense-violating images.

Quantitative results. In Tab. 2, we compare the performance of EVCap, MiniGPT4, BLIP, and BLIP-2 on WHOOPS dataset. This dataset is particularly challenging due to its inclusion of unusual objects [6]. Initially, as an end-to-end trained model, our EVCap exhibits performance similar to MiniGPT4. However, there is a noticeable improvement in the CIDEr score, after the external memory is enriched with 2396 new objects from the WHOOPS dataset, each represented by 5 synthesized images generated using stable diffusion [36]. It highlights the effectiveness of

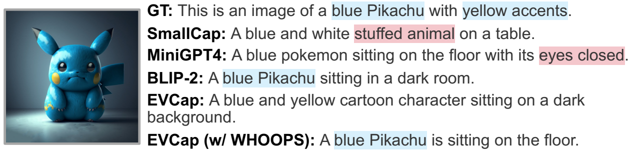

GT: This is an image of a blue Pikachu with yellow accents.
SmallCap: A blue and white stuffed animal on a table.
MiniGPT4: A blue pokemon sitting on the floor with its eyes closed.
BLIP-2: A blue Pikachu sitting in a dark room.
EVCap: A blue and yellow cartoon character sitting on a dark background.
EVCap (w/ WHOOPS): A blue Pikachu is sitting on the floor.

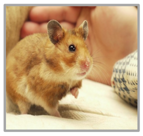
Figure 4. Examples of captions generated by our EVCap, EVCap (w/ WHOOPS), and three SOTAs on WHOOPS dataset. Incorrect objects are highlighted in red , while correct ones are in blue .

Table 2. Quantitative results on commonsense-violating data – WHOOPS dataset. EVCap (w/ WHOOPS) denotes EVCap using the memory expanded by WHOOPS objects. The results reveal the open-world comprehension ability and expandability of EVCap.

Method	B@4	M	C	S
Only pre-trained models				
BLIP [24] (from [6])	13	–	65	–
BLIP-2 FlanT5-XXL [25] (from [6])	31	–	120	–
BLIP-2 FlanT5-XXL [25] (reproduced)	28	26.7	93.1	17.9
Finetuned models on COCO				
MiniGPT4 [46]	24.2	26.7	84.8	18.2
BLIP [24]	22.9	25.0	79.3	17.1
BLIP-2 FlanT5-XL [25]	25.8	27.0	89.1	18.3
End-to-end trained models on COCO				
EVCap	24.1	26.1	85.3	17.7
EVCap (w/ WHOOPS)	24.4	26.1	86.3	17.8

Table 3. Ablation study on components prior to the LLM decoder in EVCap. The result of "+ Attentive fusion" demonstrates the substantial impact of the external visual–name memory.

Method	COCO test		NoCaps val		Flickr30k test	
	C	S	C	S	C	S
ViT + Q-Former (Baseline)	134.4	23.9	108.8	14.2	76.8	17.3
+ Image query tokens (Baseline+)	134.1	23.8	109.0	14.3	77.3	17.2
+ Attentive fusion (EVCap)	140.1	24.7	119.3	15.3	84.4	18.0



GT: A hamster on a blanket with a hand behind it.
Baseline: A hamster sitting on a couch with a person's hand near it.
Baseline+: A close up of a small hamster on a bed.
EVCap: A small hamster sitting on a blanket next to a hand.
(pillow, hamster, grizzly, gameboard, kitten, spectacles, cat, blanket, giraffe, pug-dog)

Figure 5. Visualization of the captions generated from ablation study on the NoCaps validation set. We also show the retrieved object names by EVCap, presented in gray. Incorrect objects in captions are highlighted in red, while correct ones are in blue.

our idea of incorporating an expandable external memory into the captioning model for open-world comprehension.

Qualitative results. Fig. 4 illustrates the captions generated by EVCap, EVCap (w/WHOOPS), and three SOTAs for one image from the WHOOPS dataset. Similar to other methods except for BLIP-2, EVCap can not recognize *"blue cartoon character"* as *"Pikachu"*, while EVCap (w/WHOOPS) successfully predicts it because of the updated memory. SmallCap and MiniGPT4 tend to generate captions with hallucinatory objects, a result of commonsense-violating contents present in the images.

5.3. Detailed analysis

Ablation study. We assess the contribution of each component prior to the LLM decoder in EVCap by incrementally integrating the image query tokens and the attentive fusion module into our baseline model. The baseline model comprises a ViT+Q-Former, a linear layer, and a LLM decoder.

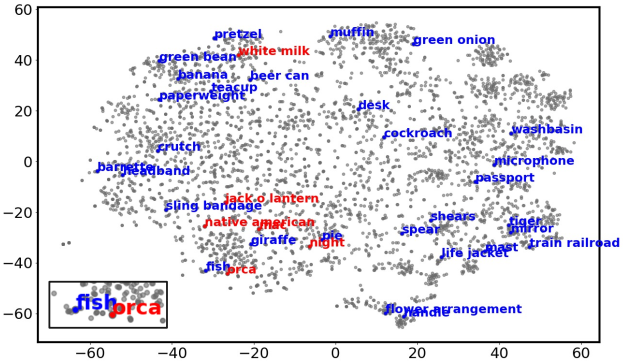


Figure 6. Visualization of the visual features in external memory using t-SNE. For visual features in LVIS dataset's objects (**blue**), the related objects fall in the same cluster. After adding more visual features of synthesized images from WHOOPS' objects, new objects (**red**) are located at appropriate clusters (zoom-in view).

The quantitative results are shown in Tab. 3. When employing only the baseline model (Baseline), CIDEr scores drop notably by 5.7, 10.5, and 7.6 on COCO, NoCaps, and Flickr30k, respectively. The inclusion of trainable image query tokens (Baseline+) brings a marginal improvement on NoCaps and Flickr30k. However, the performance is significantly enhanced with the addition of attentive fusion (along with the introduction of external memory), indicating the pivotal role of the external visual–name memory in the overall effectiveness of EVCap. This is further corroborated by the qualitative results in Fig. 5, where captions from Baseline and Baseline+ inaccurately include objects like *"couch"* and *"bed"*, and Baseline+ overlooks *"hand"*.

Exploration for external memory expandability. To demonstrate the scalability of the external memory in EVCap, we visualize the visual features stored in LVIS external memory, and newly synthesized data from objects appearing in the WHOOPS dataset. We employ t-SNE [38] to plot visual features after reducing their dimensions to 2-D (Fig. 6). For clear visualization, we only randomly display 3649 visual features in LVIS memory, and add 479 visual features from WHOOPS objects. Among them, 35 samples are randomly labeled. The result shows a clear clustering of LVIS objects (blue) in the external memory, as well as the successful integration and appropriate localization of new objects from WHOOPS (red) into these clusters. This pattern not only confirms the distinctiveness of visual features already present in the memory but also demonstrates the potential to accurately incorporate and differentiate new objects introduced from updated data. These findings highlight our external memory's ability to expand and maintain its effectiveness even as new data is incorporated.

Impact of external memory size. We examine the impact of external memory size in Tab. 4. On the one hand, we randomly remove 30%, 60%, and 90% data in the external

Table 4. Impact of the external memory size on the performance of EVCap by evaluation under CIDEr scores. Changes in the size of external memory result in changes in performance.

Method	NoCaps val				Flickr30k
	In	Near	Out	Overall	Test
LVIS objects (EVCap)	111.7	119.5	116.5	119.3	84.4
– 30% LVIS	112.0	119.2	115.3	118.8	85.0
– 60% LVIS	111.4	119.1	116.2	119.0	85.1
– 90% LVIS	110.6	118.2	115.8	118.3	83.6
+ WHOOPS	110.7	118.9	116.7	119.0	84.9

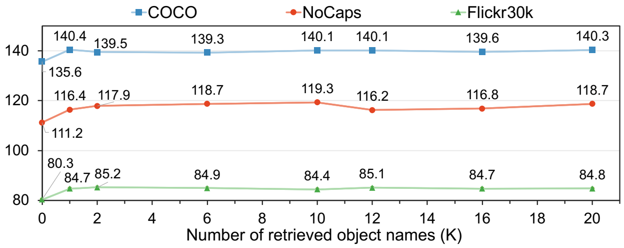


Figure 7. CIDEr scores after training EVCap with the number of retrieved object names K from 0 to 20. The results indicate that the performance is relatively optimal when K is set to be 10.

memory constructed from LVIS objects. The results show the performance gradually degrades on NoCaps as reducing 30% and 90% LVIS. Despite some unexpected increases in certain results on NoCaps (5th row) and Flickr30k (4th - 5th rows), they do not alter the overall downward trend. Similar phenomena are also noted in SmallCap [35], we speculate it is due to data distribution. On the other hand, as we infuse WHOOPS knowledge into LVIS memory, there is a slight improvement on NoCaps (out) and Flickr30k. These observations validate the model's capability to effectively retrieve object names from an updated memory, enhancing its performance in generating captions.

Impact of the number of retrieved object names. We investigate how the number of retrieved object names K (Sec. 3.3) affect EVCap in Fig. 7. We train the model with K from 0 to 20 and evaluate the performance under CIDEr on all three benchmarks. From the results, we can find that the model works worst on the out-domain dataset (NoCaps) with zero object names. It confirms that the performance boost from Baseline+ to EVCap (Tab. 3) is primarily attributed to the retrieval-augmented mechanism, but not the customized Q-Former itself. With more object names, performance fluctuates but improves. Furthermore, we observe that setting K to 10 yields relatively optimal overall performance, validating the choice of $\mathrm{K} = 10$ in EVCap.

Analysis with different decoders. To explore the influence of different LLMs decoders on our EVCap, we experiment by substituting Vicuna-13B with GPT2 and Vicuna-7B, as detailed in Tab. 5. With GPT2 as the decoder, EVCap still markedly surpasses other GPT2-based models, achieving impressive gains of 11.3 and 10.0 under CIDEr on COCO and Flickr30k, compared to SmallCap. When employing Vicuna-7B, the comparison of performance trends mirrors those observed with Vicuna-13B, further attesting to the robustness and adaptability of EVCap across different LLM decoders. Notably, both SmallCap, which retrieves captions, and our GPT2-based EVCap, which retrieves object names, use the same GPT2 decoder. Therefore, their comparison also underscores the effectiveness of our method's object name retrieval and attentive fusion strategy.

Table 5. Analysis with different LLM decoders including GPT2, Vicuna-7B, and Vicuna-13B. The results reveal EVCap is effective when applying it in different LLM decoders.

Method	LLM	COCO test		NoCaps val		Flickr30k test	
		C	S	C	S	C	S
SmallCap [35]	GPT2	119.7	21.3	–	–	60.6	–
ViECap [15]	GPT2	92.9	18.2	66.2	9.5	47.9	13.6
EVCap	GPT2	131.0	23.2	97.6	13.3	70.6	16.1
MiniGPT4 [46]	Vicuna-7B	119.4	23.5	108.7	15.7	73.9	17.2
InstructBLIP [11]	Vicuna-7B	–	–	123.1	–	82.4	–
EVCap	Vicuna-7B	139.0	24.7	116.8	15.3	82.7	18.0
MiniGPT4 [46]	Vicuna-13B	129.6	23.4	108.8	15.1	78.4	16.9
InstructBLIP [11]	Vicuna-13B	–	–	121.9	–	82.8	–
EVCap	Vicuna-13B	140.1	24.7	119.3	15.3	84.4	18.0

Limitations. First, EVCap cannot retrieve all objects that appear in the given image due to the memory coverage limits, leading to incomplete image descriptions (Fig. 4). We will investigate integrating object detection with image captioning to enhance completeness. Second, our focus on object representation restricts consideration of other crucial captioning elements, affecting overall performance. Similar to all models trained with COCO dataset, EVCap has limitations in generating varied styles, which is reflected in our relatively modest performance improvements in Tab. 2, compared to MiniGPT4. We will overcome it by exploring methodologies that encourage style diversity in the future.

6. Conclusion

We further advance image captioning in real-world scenarios by introducing EVCap, a novel image captioning model with object names retrieved from an external visual–name memory. The external memory is easily expandable, allowing for effortless updates with new object visuals and names. We extensively compare EVCap with SOTAs on various benchmarks and commonsense-violating data, demonstrating its significant superiority in performance.

Acknowledgements. This work was supported by JSPS/MEXT KAKENHI Grant Numbers JP22H05015, JP23H03449, JP23KJ0404, and JP22K17947, and the commissioned research (No. 225) by the National Institute of Information and Communications Technology (NICT), Japan, and the Institute of AI and Beyond of the University of Tokyo, and ROIS NII Open Collaborative Research 2023-23FC01, 2024-24S1201.

References

[1] https://huggingface.co/spaces/RitaParadaRamos/SmallCapDemo. 5

[2] 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 *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 4

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[5] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 5

[6] Nitzan Bitton-Guetta, Yonatan Bitton, Jack Hessel, Ludwig Schmidt, Yuval Elovici, Gabriel Stanovsky, and Roy Schwartz. Breaking common sense: Whoops! a vision-and-language benchmark of synthetic and compositional images. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2023. 4, 6, 7

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[8] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. PaLI-X: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023. 1, 2, 5

[9] 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 V Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme Ruiz, Andreas Peter Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 2, 5

[10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023. 2, 4

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3, 5, 8

[12] 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 (ICLR)*, 2021. 2

[13] Jiashuo Fan, Yaoyuan Liang, Leyao Liu, Shaolun Huang, and Lei Zhang. Rca-noc: Relative contrastive alignment for novel object captioning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2023. 2, 5

[14] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 4

[15] Junjie Fei, Teng Wang, Jinrui Zhang, Zhenyu He, Chengjie Wang, and Feng Zheng. Transferable decoding with visual entities for zero-shot image captioning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2023. 2, 4, 5, 8

[16] Zhengcong Fei. Memory-augmented image captioning. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 2, 5

[17] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[18] Ryuichiro Hataya, Han Bao, and Hiromi Arai. Will large-scale generative models corrupt future datasets? In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2023. 3

[19] Lisa Anne Hendricks, Subhashini Venugopalan, Marcus Rohrbach, Raymond Mooney, Saenko Kate, and Trevor Darrell. Deep compositional captioning: Describing novel object categories without paired training data. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[20] 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. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 5

[21] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 3

[22] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 4

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

North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019. 4

[24] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proc. International conference on machine learning (ICML)*, 2022. 5, 7

[25] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proc. International conference on machine learning (ICML)*, 2023. 1, 2, 3, 4, 5, 7

[26] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2, 5

[27] 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 *Proc. European Conference on Computer Vision (ECCV)*, 2014. 4

[28] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[29] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 2, 3, 5

[30] 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. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015. 4

[31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proc. International conference on machine learning (ICML)*, 2021. 2

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 21(1):5485–5551, 2020. 2

[34] Rita Ramos, Desmond Elliott, and Bruno Martins. Retrieval-augmented image captioning. In *Proc. Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2023. 2

[35] Rita Ramos, Bruno Martins, Desmond Elliott, and Yova Kementchedjhieva. Smallcap: Lightweight image captioning prompted with retrieval augmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 3, 5, 8

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 6

[37] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 4

[38] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research (JMLR)*, 9(11), 2008. 7

[39] Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, Raymond Mooney, Trevor Darrell, and Kate Saenko. Captioning images with diverse objects. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[40] Duc Minh Vo, Hong Chen, Akihiro Sugimoto, and Hideki Nakayama. Noc-rek: Novel object captioning with retrieved vocabulary from external knowledge. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 5

[41] 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. *arXiv preprint arXiv:2311.03079*, 2023. 4, 5

[42] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. International conference on machine learning (ICML)*. 2

[43] Zhuolin Yang, Wei Ping, Zihan Liu, Vijay Korthikanti, Weili Nie, De-An Huang, Linxi Fan, Zhiding Yu, Shiyi Lan, Bo Li, et al. Re-ViLM: Retrieval-augmented visual language model for zero and few-shot image captioning. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023. 2

[44] Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Richard James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Retrieval-augmented multimodal language modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[45] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 5

[46] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 3, 4, 5, 7, 8