CopeCap: A lightweight image captioning model with collaborative prompt learning

Xiwei Yu*‡, Guoshun He*‡ and Huijing Zhan†
* Nanyang Technological University, Singapore
E-mail: xiwei001@e.ntu.edu.sg guoshun001@e.ntu.edu.sg
† Singapore University of Social Sciences (Corresponding author) E-mail: hjzhan@suss.edu.sg
‡ These authors contributed equally.

***Abstract*—In recent years, the application of large models in image captioning has become increasingly extensive, with more models enhancing the complexity of the model, resulting in an increase in the trainable parameters of the model and greater difficulty in deployment. To address this issue, we propose a lightweight model called CopeCap, which leverages collaborative prompts that combine both textual and visual features to enhance the image captioning task. More specifically, CopeCap effectively fuses visual features with traditional text-only prompts, enabling the generation of high-quality, precise and diverse descriptions while maintaining lower model complexity and fewer trainable parameters. Extensive experiments are conducted on MSCOCO and nocaps dataset, and the results show that our work could achieve the SOTA performance, which indicates that our model is efficient and easy to train. Our code and full implementation will be made publicly available at https://github.com/Nemovv8/copecap-1.75m upon acceptance of this paper.**

I. Introduction

Image captioning integrates computer vision and natural language processing to automatically generate descriptive text for visual content. This technology has broad real-world impact, such as assisting the blind or visually impaired by converting images into speech or text [1], and improving content accessibility and searchability on social media platforms [2].

Recent advances in image captioning largely benefit from scaling up training datasets and model size [3]–[6], which significantly increases computational and deployment costs. To address this, several lightweight approaches, such as ClipCap [7] and I-Tuning [8], freeze pre-trained visual encoders and language decoders while training only a small connector module, greatly reducing complexity. Although efficient, these methods often struggle to generalize to out-of-domain scenarios without additional retraining. More recently, SmallCap [9] and vipcap [10] alleviates this limitation by introducing external text retrieval to refine prompts, which improves generalization while remaining lightweight; However, purely text-based retrieval still fails to capture essential visual information, such as spatial layouts, object interactions, and subtle visual features that are easily misinterpreted or discarded as noise, all of which are often neglected when relying solely on textual prompts.

To address this limitation, we propose **CopeCap**, a novel lightweight image captioning model which leverages *collaborative prompt learning*. By integrating retrieved textual cues with visual features during prompt construction, CopeCap achieves improved caption quality and greater robustness.

Specifically, CopeCap employs a frozen CLIP visual encoder [11] and GPT-2 language decoder [12], linked by a cross-attention layer with only millions of trainable parameters. During the collaborative prompt learning phase:

1) **Textual cues**: Given an input image, we encode it with CLIP and retrieve the top-k semantically relevant captions from an external corpus using KNN, providing high-level semantic context.
2) **Visual cues**: In parallel, the model incorporates the input image's visual features, offering fine-grained spatial and appearance information typically absent from text.
3) **Collaborative prompt**: By jointly conditioning on retrieved text and visual features, CopeCap generates more precise and diverse captions. For instance, while text may describe "a dog in a park," visual cues can specify the breed or scene lighting.

Our contributions are three-fold: (1) We propose the first collaborative prompt learning framework that achieves a strong balance between efficiency and effectiveness.(2) Copecap experienced a 50% reduction in the number of trainable parameters, leading to a corresponding 43% decrease in training time. (3) We conduct extensive experiments, including scaling different GPT-2 backbone sizes and augmenting MSCOCO with 13 additional text datasets, demonstrating that CopeCap achieves state-of-the-art performance with fewer trainable parameters and improved robustness.

II. Related Work

A. Model Structures and Efficiency Challenges in Image Captioning

Modern image captioning systems predominantly adopt the encoder-decoder architecture, where an image is first encoded into visual features by a pre-trained vision model, followed by an autoregressive language decoder that generates the corresponding caption [13], [14]. State-of-the-art approaches typically rely on general-purpose vision-language (VL) models [3]–[5], [15], which are pre-trained on large-scale image-text corpora to capture rich multimodal representations. These models are then fine-tuned on task-specific datasets for downstream applications such as image captioning. However, this

979-8-3315-7206-8/25/$31.00

paradigm often demands substantial computational resources for both training and deployment, especially when each task requires a dedicated model.

To address these efficiency challenges, recent studies have explored freezing strategies, where pre-trained components are kept intact during training to avoid catastrophic forgetting — the phenomenon in which models lose previously learned knowledge during fine-tuning [16]. By avoiding gradient updates for certain parameters, such methods significantly reduce training time and GPU memory consumption [17].

Building upon this line of work, we propose CopeCap, a novel framework that further reduces the number of trainable parameters while maintaining strong performance through the integration of retrieved textual and visual information. We detail the architecture and advantages of CopeCap in the following section.

B. Retrieval Augmented Generation

Retrieval augmented language generation is a method that involves conditional generation based on additional information retrieved from an external datastore [18]. This approach has been gaining attention in various tasks [19], [20], but remains relatively underexplored in the field of image captioning[21]–[24]. Some relevant work on image captioning is being conducted, such as the retrieval-augmented Transformer captioning models recently proposed by Sarto et al. and Ramos et al. [25]. These models generate new descriptions through cross-attention operations on the encoded retrieved captions. Additionally, the SmallCap model [9] employs a simple prompt-based conditional generation method, using retrieved captions as prompts for the generative language model to complete the captioning task; they were also the first to use retrieval augmentation for training-free domain transfer and generalization in image captioning, effectively enhancing the model's adaptability and performance.

In addition, recent efforts to leverage image features as inputs to prompt large language models have been noted in the literature [26]. These studies have made us realize that using image features as additional prompts can enhance the performance of prompt learning.

Compared to previous models, CopeCap innovatively incorporates image features into the prompts as well, addressing the disconnect between text and images and achieving superior empirical results. Notably, this enhancement in model performance is accompanied by a significant reduction in training costs required to reach a comparable level of performance.

C. Prompt Generation

Prompts are a key concept in pre-trained language models, used to convey specific task instructions or demonstrations to the model, guiding it more clearly when handling specific types of input [12]. In the field of joint vision and language learning, traditional prompt methods generally include only textual information, which guides the model to complete its trained tasks or demonstrate its capabilities in new, unseen tasks - this approach is known as zero-shot learning [27], [28].

Compared to traditional text-only prompts, our innovative prompting method incorporates both text and image information, offering significant benefits. Firstly, by incorporating images in addition to the text, the model gains supplementary visual information, which enhances its understanding of the specific context of the tasks. For instance, in image captioning tasks, text-only prompts might struggle to accurately capture details in images, such as emotional nuances or dynamic scenes, whereas image prompts can directly provide this visual information.

Additionally, combining text and image prompts improves the model's adaptability to complex scenes, particularly in images involving multiple objects or activities. This multimodal prompting not only helps the model more accurately understand and generate descriptions that match the content of the image, but also reduces the risk of model overfitting to specific datasets, thereby enhancing its generalization capabilities.

In summary, by integrating text and image into a composite prompt, our method significantly enhances the model's comprehension and quality of generation, making it more precise and natural in tasks like image captioning. This is an innovative prompt method that provides the model with more comprehensive information, allowing it to exhibit greater adaptability and creativity when faced with diverse inputs.

III. Proposed Approach

As shown in Fig. 1, given an input image, we first encode it using CLIP's vision encoder. Through image-text retrieval, we then retrieve the top-k embedded captions from a RAG vector database built with CLIP's text encoder. These retrieved captions are then collaboratively fused with the last hidden layer of the vision encoder, serving as the prompt for the subsequent decoder.

A. Encoder

Our encoder is based on CLIP [11]. We use its visual encoder to encode the target image and its text encoder to encode the external text database, and return the features of the image and text for subsequent search and prompt generation operations.

B. Retrieving Captions

CopeCap uses an Image-to-text retrieval method [9], in which CopeCap can use any textual material in an external database that is helpful for describing the content of an image. The external database can be manual text annotations of images, subtitles of videos, etc. In this paper, we use the manual annotations of the MsCOCO dataset [29] as our external database. Through the vision encoder and text encoder of CLIP, we encode the image and the text of the external database respectively, and project them into the same vector space for retrieval. In the retrieval process, we use the K Nearest Neighbour algorithm, which sets a k value and uses the cosine value of the angle between vectors to calculate the similarity, which selects the k most similar captions for subsequent prompt generation. Let v be the image embedding

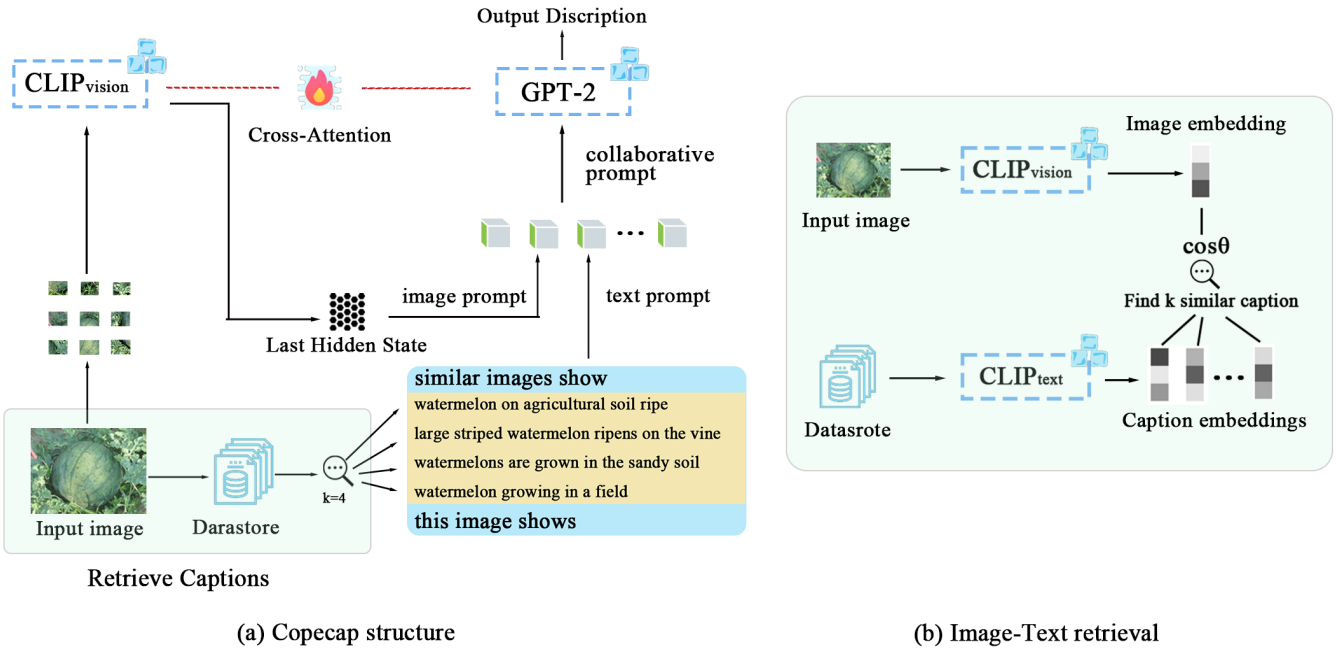

Fig. 1. The structure of CopeCap conducting image captioning task. (a) CopeCap generates captions based on the encoded input image and a set of k retrieved captions, which are provided to the decoder as prompts. (b) The k captions are selected from a datastore containing N captions using an image-to-text retrieval method.

of the given image and t_i be the caption embedding of the text description i, then the value could be calculated as:

$$\cos\theta(v, t_i) = \frac{\mathbf{v} \cdot \mathbf{t}_i}{\|\mathbf{v}\|\|\mathbf{t}_i\|} \tag{1}$$

where the result closer to 1, the higher the similarity.

C. Collaborative Prompt Learning

In the process of prompt generation, we use the features from the k most similar captions, and use the formula:" `Similar images show {caption`$_1$`}, ..., {caption`$_k$`}. This image shows __.` " to construct the text prompt. Plus, we utilize the feature of the image itself as the image prompt, hence forming the collaborative prompt.

D. Decoder

For the decoder, we use a pre-trained GPT-2 model [12] with 768 hidden dimensions and 12 attention heads. GPT-2 has multiple versions, with GPT-2 Base containing approximately 124M parameters, GPT-2 Medium having 355M parameters, and GPT-2 Large comprising 774M parameters. For the comparative experiments evaluating the effects of each version, GPT-2 processes features encoded directly from the input image while simultaneously incorporating the prompt-assisted image captioning task mentioned earlier. Through the use of cross-attention mechanism, the model effectively integrates image features and textual prompts, enabling it to generate a coherent and contextually rich textual description of the input image. The GPT-2 model is used to fill the blank in the prompt and generate the predicted caption $\hat{y}$. During training, the predicted caption $\hat{y}$ is compared to the ground truth caption y using the cross-entropy loss:

$$L = -\sum_{i=1}^{M} y_i \log P_\theta(y_i | y_{<i}, P) \tag{2}$$

where P represents the collaborative prompt, y_i is the ground truth token at position i, and $y_{<i}$ are the tokens generated before y_i. The parameters θ of the model are updated through backpropagation to minimize this loss.

IV. Main Experiments

A. Dataset

Our work uses the MsCOCO dataset [29] to train CopeCap and test the trained model on the nocaps dataset [30]. The introduction of MsCOCO and nocaps could be found in Appendix C. In order to study the benefits of retrieval enhancement on model output, we utilized a substantial volume of web data and a comparatively limited amount of high-quality manually annotated data.

1) Web Data: We consider that as the external database grows, the retrieval effect will be better, so we consider introducing a large amount of web data to expand the external

Fig. 2. Examples of retrieved captions (green) and generated results (black), with green and red highlighting indicating correct and mismatched content, respectively.

database. We used text descriptions from the following datasets to augment the external database: Conceptual Caption [31], Conceptual 12M [32], SBU Captions [33], by introducing these network data, we found that CopeCap performed significantly better on nocaps out-of-domain data. For an introduction to these datasets, see Appendix D.

2) Human-labeled Data: We also consider that introducing a small amount of clean manually annotated data can effectively improve the quality of external databases, thereby improving the effectiveness of the retrieval process, so we introduce the following manually annotated datasets: Flickr30k [34], VizWiz [35], MSRVTT [36], TGIF [37], Clotho [38], LN ADE20k, LN COCO, LN Flickr30k, LN OpenImages [39], by using these manually annotated datasets, the quality of our external database has been improved, and the model performance has been better when tested on the nocaps database. For an introduction to these datasets, see Appendix E.

B. Evaluation metrics

Some widely adopted evaluation metrics are employed to assess the performance of CopeCap: BLEU [40], METEOR [41], and CIDEr [42]. These metrics have been commonly used in previous image captioning studies [9], and we follow their conventions for a fair comparison. For a detailed explanation of each metric, please refer to Appendix A.

C. Experimental Setup

The encoder and decoder of CopeCap use CLIP-ViT-B/32 and GPT-2 Base, with their parameters frozen during training. Only the cross-attention layer parameters are updated. To reduce trainable parameters, we progressively decrease the projection matrix dimensions from 64 to 16, 8, and 4, resulting in model variants with 7M, 3.6M, and 1.8M trainable parameters, respectively. The total number of parameters in CopeCap is 218M, including non-trainable encoder and decoder parameters.

CopeCap is trained on the MSCOCO dataset using the Karpathy splits, optimizing cross-entropy loss with the AdamW optimizer, an initial learning rate of 0.0001, and a batch size of 64 over 10 epochs. The model checkpoint with the highest CIDEr score on the validation set is selected. Training is performed on a RTX4090D. During training, each image is paired with 4 captions retrieved from a datastore of MSCOCO training captions using CLIP representations and FAISS for efficient nearest neighbor retrieval. Inference uses beam search decoding with a beam size of 3.

D. Comparison Results

The test results of our trained model on the MsCOCO dataset are shown in Table 1. It can be seen that CopeCap performs better than many popular lightweight models, which owning to the contribution of our collaborative prompt, our model effectively leverages the information contained in both visual and textual features, enabling a better understanding of the input image. Compared with some large model, CopeCap achieves similar performance but way less in parameter amount, indicating the effectiveness of CopeCap in improving prompt generation.

TABLE I
MODEL PERFORMANCE COMPARISON

Model	$\|\theta\|$	B@4	M	CIDEr
Large Models with V&L pre-training				
$\text{LEMON}_{\text{Huge}}$	675	**41.5**	30.8	139.1
$\text{SimVLM}_{\text{Huge}}$	632	40.6	**33.7**	**143.3**
$\text{OSCAR}_{\text{Large}}$	338	37.4	30.7	127.8
$\text{BLIP}_{\text{CapFilt-L}}$	224	39.7	-	133.3
Lightweight-training models				
CaMEL	76	39.1	29.4	125.7
$\text{CopeCap}_{\text{Large}}$	47	39.5	29.2	126.8
ClipCap	43	33.5	27.5	113.1
$\text{CopeCap}_{\text{Med}}$	22	38.1	29.0	123.5
$\text{SmallCap}_{\text{Base}}$	7	35.1	27.2	119.3
$\text{CopeCap}_{\text{Base}}$	7	36.1	27.6	119.5

$|\theta|$: Number of trainable parameters (millions); B@4: BLEU4; M: METEOR.

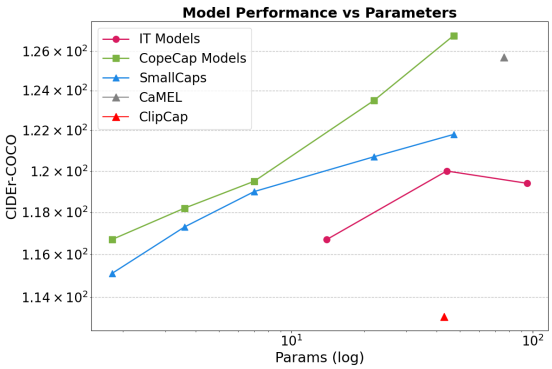

Fig. 3. CopeCap's performance on the MsCOCO dataset, compared with other models with similar amount of trainable parameters. The number of trainable parameters can be adjusted by modifying the dimensionality of the cross-attention layers and the decoder size. CopeCap is well-performed compared to other lightweight models on MsCOCO.

TABLE II
COMPARISON BETWEEN I-TUNING, SMALLCAP AND COPECAP

Model	$\|\theta\|$	B@4	M	CIDEr
I-Tuning$_{\text{Large}}$	95	34.8	29.3	119.4
SmallCap$_{d=16,\text{Large}}$	47	37.2	28.3	121.8
CopeCap$_{d=16,\text{Large}}$	47	**39.5**	**29.2**	**126.8**
I-Tuning$_{\text{Med}}$	44	35.5	28.8	120.0
SmallCap$_{d=16,\text{Med}}$	22	36.5	28.1	120.7
CopeCap$_{d=16,\text{Med}}$	22	**38.1**	**29.0**	**123.5**
I-Tuning$_{\text{Base}}$	14	34.8	28.3	116.7
SmallCap$_{d=8,\text{Base}}$	3.6	35.9	27.6	117.3
CopeCap$_{d=8,\text{Base}}$	3.6	**36.3**	27.6	**118.2**
SmallCap$_{d=4,\text{Base}}$	1.8	35.4	27.1	115.1
CopeCap$_{d=4,\text{Base}}$	1.8	**35.8**	**27.3**	**116.7**

$|\theta|$: the number of trainable parameters in the model (in millions); B@4: BLEU4; M: METEOR.

As shown in Fig. 3, CopeCap achieves up to an 83% reduction in trainable parameters compared to lightweight models like ClipCap [7]. This is accomplished by freezing both the encoder and decoder parameters, optimizing only the cross-attention layer. Additionally, CopeCap outperforms SmallCap [9] across various metrics, particularly in out-of-domain scenarios, due to the proposed collaborative prompt learning.

Table 2 presents the comparison of I-Tuning, SmallCap, and CopeCap under different GPT-2 sizes. The reproduced results show that CopeCap consistently outperforms SmallCap across varying parameter counts, demonstrating that incorporating visual features as prompts effectively complements textual prompts and leads to superior captioning performance.

As shown in Fig.4, we selected 3 images to compare the results generated by the SmallCap and CopeCap models, both with 3.5M parameters. It is evident that the CopeCap model produces more accurate results. For example, in the first image, SmallCap used the word 'holding a banana', which is incorrect. This is because textual prompts alone can only provide information about "hat" and "banana," while the inclusion of visual features enables the model to recognize the association between "hat" and "banana," resulting in better outputs.

The test results of CopeCap on the test set of the nocaps dataset are shown in Table 3. We observe that CopeCap not only achieves excellent performance in the In-domain and Near-domain fields, but also greatly improves the performance in the Out-of-domain field compared with other models. Moreover, with the expansion of external datasets and the introduction of high quality external datasets, the performance of the model, especially in the Out-of-domain field, is further improved. It is because introducing large-scale Web text datasets can expand the external database, allowing for the retrieval of descriptions that are more closely aligned with the given image, thereby generating higher-quality prompts. Additionally, incorporating human-labeled databases introduces more human-like descriptions into the external database, enhancing its overall quality and subsequently improving the quality of the retrieved texts and the generated prompts.

TABLE III
PERFORMANCE COMPARISON ACROSS DIFFERENT SETTINGS

Model	In	Near	Out	Entire
OSCAR$^{\diamond}_{\text{Large}}$	84.8	82.1	73.8	80.9
CaMEL*	88.1	79.1	54.6	75.9
ClipCap*	74.5	65.6	47.1	63.4
SmallCap	83.3	77.1	65.0	75.8
SmallCap$_{\text{+W+H}}$	87.9	84.6	84.4	85.0
CopeCap	93.59	87.39	69.29	84.94
CopeCap$_{\text{+W}}$	95.92	92.02	86.58	91.53
CopeCap$_{\text{+H}}$	**97.42**	**93.53**	83.21	92.18
CopeCap$_{\text{+W+H}}$	96.31	93.28	**86.92**	**92.52**

+W: add Web data; +H: add Human-labeled data.

E. Ablation Study

1) Types of Prompt: In Fig.5, we observe that compared to methods without prompts or those using Prompted with Retrieval Augmentation, our Collaborative Prompt Learning achieves superior performance in image captioning, demonstrating significant improvements and highlighting the effectiveness of our approach. We also conducted experiments to verify the impact of different prompt forms on the results. To this end, we trained a model with 3.6M parameters using only visual feature prompts and compared its performance with models of the same parameter size trained without prompts, with only text prompts, and with our proposed collaborative prompt method. The results are shown in Table 4. As mentioned earlier, the superior results are attributed to the complementary effects of textual and visual features. Utilizing both as prompts significantly enhances the model's performance in image captioning.

2) Feature Fusion: We also considered fusing text features and visual features before using them as prompts for GPT-2. Specifically, we experimented with the Hadamard product and convolution methods (detailed descriptions of these methods

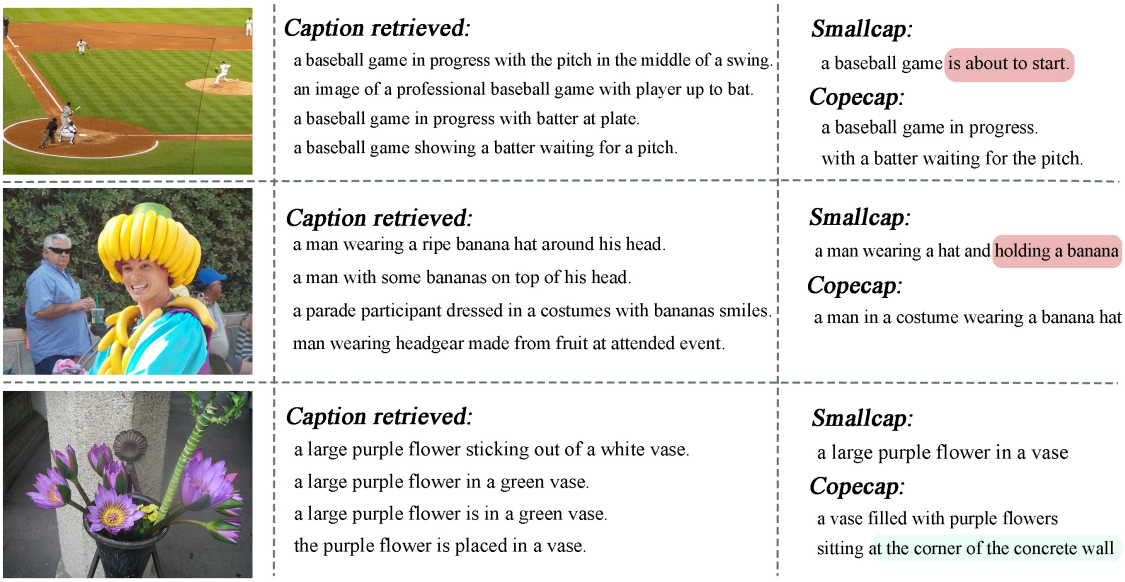

Fig. 4. The performance comparison between two image captioning models, Smallcap and CopeCap. Inaccuracies identified by Smallcap are marked in red, while precise details correctly identified by CopeCap are highlighted in green.

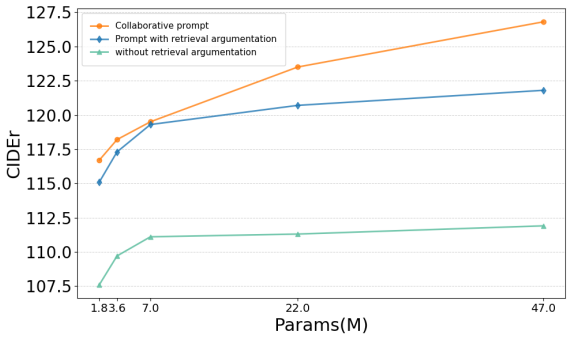

Fig. 5. Ablation experiments results on MsCOCO test set.

TABLE IV
THE IMPACT OF DIFFERENT PROMPT METHODS ON IMAGE CAPTIONING PERFORMANCE.

Type	B@4	M	CIDEr
Without prompt	35.1	27.0	114.3
Text prompt	36.3	27.6	117.3
Visual prompt	34.9	27.2	115.3
Collaborative prompt	**36.3**	**27.6**	**118.2**

B@4: BLEU4; M: METEOR.

can be found in Appendix B), but neither yielded satisfactory results. The performance of both methods is shown in Table 5. The reason behind the bad performance maybe: (1) Simple fusion methods may not effectively model the complex relationships between text and image features. (2) Direct fusion may blur the distinctive features of each modality.

TABLE V
RESULT OF DIFFERENT FUSION METHODS

Fusion Method	B@4	M	CIDEr
Hadamard product	36.0	26.4	114.5
Convolution	34.2	25.4	104.5
CopeCap	**36.1**	**27.6**	**119.5**

B@4: BLEU4; M: METEOR.

V. CONCLUSION

CopeCap addresses key challenges in image captioning by introducing a lightweight model that integrates textual and visual features through Collaborative Prompt Learning. By freezing the pre-trained encoder and decoder while optimizing only the cross-attention layer, CopeCap reduces the number of trainable parameters to just 7 million, significantly lowering training and deployment costs. Evaluations on MsCOCO and nocaps demonstrate its strong performance, particularly in out-of-domain tasks, where it outperforms many larger models. It achieves BLEU4 of 36.1, METEOR of 27.6, and CIDEr of 119.5 on MsCOCO, setting a benchmark for efficiency and effectiveness. CopeCap offers practical solutions for applications such as social media content generation and assistive technology, paving the way for adaptive, low-resource captioning models. However, although Collaborative Prompt Learning leverages both textual and visual features for prompt construction, identifying an optimal method for feature fusion remains an open challenge. Exploring more effective fusion techniques could potentially lead to further performance improvements.

APPENDIX

A. Appendix A

1) BLEU: BLEU (Bilingual Evaluation Understudy) [40] is an n-gram exact match based metric that evaluates quality by calculating the n-gram overlap between the generated text and the reference text.

$$BLEU = BP \cdot \exp(\sum_{n=1}^{N} \omega_n \cdot \log p_n)$$

where BP stands for Brevity Penalty, used to penalize the case where the generated text is too short. It is defined as:

$$BP = \begin{cases} 1, & \text{if} \quad \text{c} > \text{r} \\ \exp(1 - r/c), & \text{if} \quad \text{c} \leq \text{r} \end{cases}$$

where c is the length of the generated text and r is the length of the reference text. p_n stands for the n-gram exact match rate, and the ω_n stands for the weight of n-gram.

2) METEOR: METEOR (Metric for Evaluation of Translation with Explicit Ordering) [41] calculates the similarity between the generated text and the reference text through word-level matching (including synonyms, stems, etc.), while taking into account word order and semantic information.

$$METEOR = F_{mean} \cdot (1 - Penalty)$$

F_{mean} stands for the average F score, is obtained by weighted summing of precision and recall. $Penalty$ is the penalty factor, based on word order errors, is defined as:

$$\text{Penalty} = 0.5 \cdot \left(\frac{\text{number of chunks}}{\text{number of matched words}} \right)^3$$

3) CIDEr: CIDEr (Consensus-based Image Description Evaluation) [42] is designed for the image description generation task and uses TF-IDF weighted n-gram matching to measure the similarity between the generated description and the reference description.

$$CIDEr = \frac{1}{N} \sum_{n=1}^{N} CIDEr_n$$

where $CIDEr_n$ is calculated as:

$$CIDEr_n = \frac{\sum_{g \in G} TF - IDF_g \cdot TF - IDF_g^{'}}{\|TF - IDF_g\| \cdot \left\|TF - IDF_g^{'}\right\|}$$

B. Appendix B

1) Hadamard product: The Hamada product is an element-by-element multiplication operation of corresponding elements of two matrices or tensors. Suppose there are two tensors A and B of the same dimension, their Hamada product is expressed as:

$$C = A \circ B$$

where $C_{ij} = A_{ij} B_{ij}$.

2) Convolution: Convolution is an operation that performs operations on feature maps by sliding convolution kernels (filters) to extract information from local areas. The convolution operation formula is:

$$C(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} K(m,n) X(i+m-1, j+n-1)$$

where $C(i,j)$ is the (i,j)th element of the feature map, $K(m,n)$ is the convolution kernel with the size of $m \times n$, $X(i+m-1, j+n-1)$ is the local area of the input feature map.

C. Appendix C

1) MsCOCO: MsCOCO (Microsoft Common Objects in Context) [29] is an open source dataset widely used in the field of computer vision, released by Microsoft Research in 2014. It is known for its diversity and complexity, and is mainly used to train and evaluate algorithms for tasks such as image recognition, object detection, instance segmentation, key point detection, and image annotation generation. In our work, we combined the training set and validation set as the training set, and use the test set for testing. In our work, we merged the training set and validation set of MsCOCO dataset, and re-divided the training set and test set from this merged dataset. We used the test set of MsCOCO as our test set to verify the model capabilities obtained through training. We used the captions that come with the MsCOCO dataset as our external database for retrieval.

2) nocaps: The nocaps (Novel Object Captioning at Scale) [30] dataset is a dataset designed for the task of "novel object captioning". Released in 2019 by the Allen Institute for AI and the Google AI research team, it aims to evaluate and advance the model's ability to generate image descriptions, even if these images contain new objects not seen in training. The nocaps dataset contains images of three categories: In-domain, Near-domain, and Out-of-domain. In-domain means that the object category is the same as the known category in the training set; Near-domain means that the image contains some categories of objects that are not seen in the training set; Out-of-domain means that all object categories in the image are categories that are not seen in the training set. We use this dataset to test the model and get the performance of the model on the 3 categories.

D. Appendix D

1) Conceptual Caption: Conceptual Caption [31] is an image-text pair dataset released by Google, containing 3.3 million images and their corresponding descriptions. The descriptions are generated by automatically cleaning and filtering HTML image captions from the web to ensure that the descriptions are more diverse and semantically rich.

2) Conceptual 12M: Conceptual 12M [32] is an extended version of the Conceptual Captions dataset, containing 12 million images and their descriptions. The data source and generation method are the same as the original Conceptual Captions, but on a larger scale.

3) SBU Captions: SBU Captions [33] was created by SUNY Stony Brook University and contains about 1 million images and descriptions. The images and text descriptions come from the natural language captions of Flickr images.

E. Appendix E

1) Flickr30k: Flickr30k [34] is an image description dataset containing 31,000 images with 5 manually generated natural language descriptions per image.

2) VizWiz: The VizWiz [35] dataset is used for Visual Question Answering and contains images, questions asked by users, and their answers.

3) MSRVTT: The Microsoft Research Video to Text dataset [36] contains 10,000 short video clips (about 200,000 seconds of video) and their text descriptions.

4) VATEX: VATEX [43] is a bilingual video description dataset containing 41,250 videos and their corresponding English and Chinese descriptions.

5) TGIF: TGIF [37] is a GIF video description dataset containing 100,000+ GIF videos and descriptions.

6) Clotho: Clotho [38] is a multimodal audio description dataset containing 4981 audio clips and corresponding natural language descriptions.

7) LN ADE20k: LN ADE20k is a semantic segmentation dataset that extends the ADE20k dataset and includes fine-grained annotations and scene descriptions.

8) LN COCO: LN COCO is an extension of the MsCOCO dataset that combines semantic segmentation and language description.

9) LN Flickr30k: LN Flickr30k extends the Flickr30k dataset and adds segmentation annotations.

10) LN OpenImages: LN OpenImages [39] extends the OpenImages dataset to include language annotations (e.g. object attributes, scene descriptions).

License and Availability

We will release the source code, pretrained models, and dataset processing scripts under the MIT License upon publication.

References

[1] C. Nishimura, S. Kurita, and Y. Seki, "Text360nav: 360-degree image captioning dataset for urban pedestrians navigation," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 15 783–15 788.

[2] X. Ding, J. Han, H. Xu, W. Zhang, and X. Li, "Hilm-d: Towards high-resolution understanding in multimodal large language models for autonomous driving," *arXiv preprint arXiv:2309.05186*, 2023.

[3] X. Hu, Z. Gan, J. Wang, *et al.*, "Scaling up vision-language pre-training for image captioning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 17 980–17 989.

[4] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International conference on machine learning*, PMLR, 2022, pp. 12 888–12 900.

[5] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao, "Simvlm: Simple visual language model pretraining with weak supervision," *arXiv preprint arXiv:2108.10904*, 2021.

[6] J. Wang, Z. Yang, X. Hu, *et al.*, "Git: A generative image-to-text transformer for vision and language," *arXiv preprint arXiv:2205.14100*, 2022.

[7] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," *arXiv preprint arXiv:2111.09734*, 2021.

[8] Z. Luo, Z. Hu, Y. Xi, R. Zhang, and J. Ma, "I-tuning: Tuning frozen language models with image for lightweight image captioning," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, pp. 1–5.

[9] R. Ramos, B. Martins, D. Elliott, and Y. Kementchedjhieva, "Smallcap: Lightweight image captioning prompted with retrieval augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2840–2849.

[10] T. Kim, S. Lee, S.-W. Kim, and D.-J. Kim, *Vipcap: Retrieval text-based visual prompts for lightweight image captioning*, 2025. arXiv: 2412.19289 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2412.19289.

[11] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[13] K. Xu, "Show, attend and tell: Neural image caption generation with visual attention," *arXiv preprint arXiv:1502.03044*, 2015.

[14] P. Anderson, X. He, C. Buehler, *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[15] X. Li, X. Yin, C. Li, *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, Springer, 2020, pp. 121–137.

[16] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, vol. 24, Elsevier, 1989, pp. 109–165.

[17] J.-B. Alayrac, J. Donahue, P. Luc, *et al.*, "Flamingo: A visual language model for few-shot learning," *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022.

[18] P. Lewis, E. Perez, A. Piktus, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[19] G. Izacard, P. Lewis, M. Lomeli, *et al.*, "Atlas: Few-shot learning with retrieval augmented language models," *Journal of Machine Learning Research*, vol. 24, no. 251, pp. 1–43, 2023.

[20] H. Li, Y. Su, D. Cai, Y. Wang, and L. Liu, "A survey on retrieval-augmented text generation," *arXiv preprint arXiv:2202.01110*, 2022.

[21] R. Ramos, D. Elliott, and B. Martins, "Retrieval-augmented image captioning," *arXiv preprint arXiv:2302.08268*, 2023.

[22] S. Sarto, M. Cornia, L. Baraldi, and R. Cucchiara, "Retrieval-augmented transformer for image captioning," in *Proceedings of the 19th international conference on content-based multimedia indexing*, 2022, pp. 1–7.

[23] C. Xu, W. Zhao, M. Yang, X. Ao, W. Cheng, and J. Tian, "A unified generation-retrieval framework for image captioning," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2313–2316.

[24] S. Zhao, L. Li, H. Peng, Z. Yang, and J. Zhang, "Image caption generation via unified retrieval and generation-based method," *Applied Sciences*, vol. 10, no. 18, p. 6235, 2020.

[25] R. P. Ramos, P. Pereira, H. Moniz, J. P. Carvalho, and B. Martins, "Retrieval augmentation for deep neural networks," in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2021, pp. 1–8.

[26] J. Merullo, L. Castricato, C. Eickhoff, and E. Pavlick, *Linearly mapping from image to text space*, 2023. arXiv: 2209.15162 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2209.15162.

[27] W. Jin, Y. Cheng, Y. Shen, W. Chen, and X. Ren, "A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models," *arXiv preprint arXiv:2110.08484*, 2021.

[28] M. Tsimpoukelli, J. L. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill, "Multimodal few-shot learning with frozen language models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 200–212, 2021.

[29] X. Chen, H. Fang, T.-Y. Lin, *et al.*, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[30] H. Agrawal, K. Desai, Y. Wang, *et al.*, "Nocaps: Novel object captioning at scale," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8948–8957.

[31] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2556–2565.

[32] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3558–3568.

[33] V. Ordonez, G. Kulkarni, and T. Berg, "Im2text: Describing images using 1 million captioned photographs," *Advances in neural information processing systems*, vol. 24, 2011.

[34] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[35] D. Gurari, Y. Zhao, M. Zhang, and N. Bhattacharya, "Captioning images taken by people who are blind," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, Springer, 2020, pp. 417–434.

[36] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5288–5296.

[37] Y. Li, Y. Song, L. Cao, *et al.*, "Tgif: A new dataset and benchmark on animated gif description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4641–4650.

[38] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 736–740.

[39] J. Pont-Tuset, J. Uijlings, S. Changpinyo, R. Soricut, and V. Ferrari, "Connecting vision and language with localized narratives," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, Springer, 2020, pp. 647–664.

[40] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[41] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the ninth workshop on statistical machine translation*, 2014, pp. 376–380.

[42] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation,"

in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[43] X. Wang, J. Wu, J. Chen, L. Li, Y.-F. Wang, and W. Y. Wang, "Vatex: A large-scale, high-quality multilingual dataset for video-and-language research," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4581–4591.