# Question-Aware Image Captioning with LLM for Knowledge-based Visual Question Answering

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Knowledge-based visual question answering (VQA) is a more complicated task than traditional VQA, where image content is insufficient to answer the questions and asks for world knowledge and reasoning. In this paper, with the aim of leveraging the rich world knowledge and strong reasoning abilities of large language models (LLMs), we transform the knowledge-based visual question answering into LLM-based linguistic question answering tasks. We adopt the method that explicitly converts the images in knowledge-based VQA into textual captions, in order to help the LLMs answer the questions. While a general image caption may miss vital clues, our work focuses on the model architecture and training data to provide the LLMs with more question-relevant image descriptions. We propose **Q**uestion-**A**ware image **C**aptioning (QAC) that utilizes the question as guidance to extract correlated visual information from the image and generate a truthful and helpful caption. To enhance this, we carefully construct a target caption dataset using GPT-4. As a result, our zero-shot performance on the OK-VQA dataset surpasses other few-shot similar language-mediated methods, achieving state-of-the-art accuracy of **62.4%**. Our dataset and codes will be released soon.

## 1 Introduction

Visual Question Answering (VQA) refers to the task that given an image and a question about the image, the target model should provide a natural language answer in an open-ended manner. VQA holds significant importance in the development of artificial intelligence, as it bridges the gap between visual perception and language understanding. In other words, this task allows machines to interpret and reason about the real world in a human-like manner. Early VQA tasks [2, 7] only involve the content of the image itself, requiring simple image understanding to answer questions such as object detection and visual attributes. There needs to be more to push the boundaries of AI capabilities toward the final goal of the world model. Therefore, given these considerations, knowledge-based VQA [26, 25, 18, 21] tasks have recently been proposed, which requires reasoning over image content combined with world knowledge. This is a more realistic approach to question-answering that aligns with the working paradigm of humans, as it leads the way to general artificial intelligence.

Knowledge-based VQA requires reasoning over image content and additional world knowledge, sometimes even involving multi-step inference; this makes it challenging for traditional VQA models to provide valid answers. To address this issue, these methods[33, 17, 29] attempt to retrieve knowledge from external knowledge bases to assist in answering questions, but they have shown limited effectiveness due to low efficiency in retrieving relevant knowledge and insufficient language capabilities of the models. The main drawbacks of these methods lie in the inefficiency and instability of knowledge retrieval and limitations in the model's language comprehension and generation capabilities, which are challenging to address.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

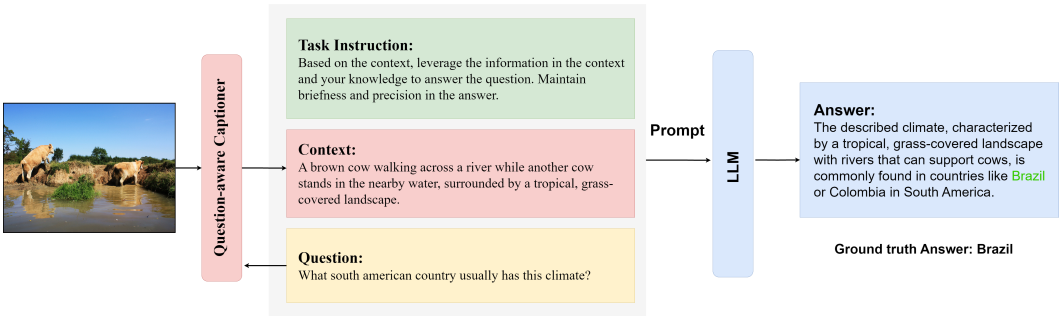


Figure 1: General pipeline of our method. Our model takes in the image and the question and outputs a question-relevant image captioning, serving as the context. The task instruction is elaborately designed to enable the LLM to fully understand its task and what to do. The off-shelf LLM prompt comprises the task instruction, the context, and the question. Then, the LLM generates the answer.

With the rise of large language models, researchers find that LLMs can serve as an implicit engine to solve knowledge-based VQA. Trained on vast corpora, LLMs demonstrate excellent language comprehension abilities, encompassing world knowledge and the ability to reason with text. These all make up for the shortcomings of the previous methods. As a result, researchers have begun leveraging large language models as a core solution for knowledge-based VQA tasks. due to lacking of visual modality in these language models, there are currently two primary methods to address this gap. One popular approach is the use of large vision language models. Prevalent multimodal large language models usually work on the method that aligns multimodal features to the semantic space of text. The typical model structure includes a large language model, one or more visual encoders, and one vision-to-language adapter module. Though showing excellent capabilities on a wide range of multimodal tasks, this end-to-end training approach requires a considerable amount of data and computational resources and lacks flexibility. Once a component (e.g., the large language model or visual encoder) is updated, the entire model must be retrained. At the same time, some of the best large language models remain in a black box state, as they are untrainable, and it is impossible to access their internal structure and parameters directly. Last but not least, even achieving remarkable success on a series of multimodal benchmarks, MLLMs haven't performed well in knowledge-based VQA tasks like OKVQA. This is primarily because multimodal large language models are not explicitly trained to combine image content and their own knowledge to perform multi-step reasoning for obtaining answers. Instead, they are trained within a unified framework. The lack of sufficient high-quality multimodal training data may also be one of the reasons.

Therefore, another alternative approach is explicitly converting images into text and feeding them into the LLMs. Techniques for mapping image content to captions have become relatively mature and have lower training costs. In this way, we can easily evaluate the output of the captioner and connect any large language model for task inference without retraining the entire model. This approach alleviates to some extent the problem of hallucination in MLLMs. Several works are adopting the same method. PICa [31], which only utilizes a general captioner to convert images into text without extracting specific image features for the question, resulting in average performance. Img2LLM [9]'s key insight is generating descriptive captions while also providing sets of question-answer pairs as few-shot examples to the large language model. Similar work includes Prophet [22], which guides the model in generating logical answers to questions. These methods share the commonality of providing additional information by generating few-shot examples or complementary cues to assist the large language model in generating answers. In contrast, our approach focuses more on mining question-related image information, enabling the language model to understand the contextual information of the question fully and, thus, better answer the question.

Figure 1 shows our method's general pipeline. We propose a question-aware caption model that takes the image and the question as input and generates a question-related context of image content. We construct the task instruction prompt to enable the LLMs to understand the question-answering tasks and, most importantly, ask them to utilize the information provided in the given context combined with their knowledge to deduce the answer. To achieve this, we construct a new question-aware caption dataset containing problem-solving assistance information instead of general descriptions, which may lack relevant information. In the captioning model structure, we fuse multimodal information

through cross-attention between the image and textual question, better extracting image features and reasoning based on the question. Notably, previous methods typically adopted few-shot settings, but our research works in a zero-shot manner and achieves comparable, even better performance. We achieve state-of-the-art accuracy of 62.4% on the OKVQA dataset, benefiting from the improved quality of our question-aware caption generation and the capabilities of large language models. We summarize our contribution as follows:

1. We construct a high-quality question-relevant caption dataset based on OK-VQA, synthesized with the help of GPT-4.
2. We propose a lightweight, plug-and-play image captioning model named QAC to connect with frozen LLMs to solve the complicated knowledge-based VQA tasks.
3. Our proposed QAC with GPT-4 achieves state-of-the-art performance on the OK-VQA dataset, i.e.,62.4% accuracy, under the zero-shot setting.

# 2 Related Work

## 2.1 Knowledge-based Visual Question Answering

Knowledge-based VQA is a much more difficult task where answering questions requires additional world knowledge beyond the image content and certain reasoning ability. Early datasets like KB-VQA [26], FVQA [25] annotate questions by selecting a fact (a knowledge triplet such as "dog is mammal") from a fixed knowledge base. OK-VQA dataset [18] is the first large-scale dataset with questions that need to be answered using external knowledge instead of a provided fixed knowledge base, and A-OKVQA provides more questions with Multiple-Choice (MC) as well as Direct Answer evaluation settings.

Recent studies [19, 33, 17, 8, 29] try to retrieve different knowledge from various external knowledge resources, e.g., ConceptNet [23], Wikipedia[24], Google Images[29], etc. These approaches jointly reason over the retrieved knowledge and the image-question pair to form the answer. However, PICa [31] finds this two-step method may be sub-optimal due to the representation mismatches during the retrieving stage and the infer stage, resulting in the potentially limited performance. At the same time, large language models have witnessed rapid advancement and demonstrated powerful capabilities in natural language processing. Researchers resort to LLMs to help accomplish knowledge-based VQA tasks.

## 2.2 LLM for Knowledge-based VQA

Trained on extensive corpora, LLMs exhibit not only excellent language comprehension abilities but also possess rich world knowledge and reasoning capabilities. Therefore, LLMs are considered excellent implicit knowledge engines for addressing knowledge-based VQA. The difficulty lies in that LLMs lack visual modality. To tackle this issue, there are generally two approaches: multimodal pretraining and image-to-text conversion.

### 2.2.1 Multimodal Pretraining Methods

Adding multimodal pretraining to the LLMs helps to build much more general vison-language models that not only solve the Knowledge-based VQA but also a series of multimodal tasks. The main method of developing Multimodel Large Language Models (MLLMs) is to align visual features into the text embedding space, i.e. making the image representation as soft prompt into the LLM. The general architecture of MLLMs is composed of a visual encoder, a language model, and an adapter module that connects visual inputs to the textual space.

The most often employed visual encoders are based on pre-trained Vision Transformer (ViT) models with a CLIP-based objective to exploit the inherent alignment of CLIP embeddings. Popular choices are the ViT-L model from CLIP [20], the ViT-H backbone from OpenCLIP [28], and the ViT-g version from EVA-CLIP [5]. The simultaneous presence of inputs from different modalities emphasizes the need to incorporate a module capable of delineating latent correspondences within these unimodal domains. These modules, termed as "adapters", are intended to facilitate interoperability between the visual and textual domains. A spectrum of different adapters are used in common MLLMs, ranging

from elementary architectures such as linear layers [6, 32] or MLP [16] to advanced methodologies such as Transformer-based solutions, exemplified by the Q-Former model[13], and conditioned cross-attention layers[1] added to the LLM. However, such MLLMs face challenges. On one hand, they require end-to-end training, which consumes significant computational resources and training data. On the other hand, even though achieving good performance on a series of multimodal benchmarks, MLLMs don't achieve as good results in knowledge-based VQA like OKVQA.

#### 2.2.2 Language-mediated Methods

Since the LLMs have extraordinary language understanding abilities, it is natural to consider captioning the input image into text descriptions. PICa [31] is the first method to adopt GPT-3 for solving the KB-VQA task in a few-shot manner by just providing a few in-context VQA examples. [8] propose to use both implicit (i.e. GPT-3) and explicit (i.e. KBs) knowledge based on CLIP retrieval [20] which are combined by a novel fusion module called KAT (based on T5 or Bart). [15] propose to integrate local visual features and positional information (bounding box coordinates), retrieved external and implicit knowledge (using a GPT-3) into a transformer-based question-answering model. [11] propose PromptCap, a novel task-aware captioning model that uses a natural language prompt to control the generation of the visual content that can be used in conjunction with GPT-3 in-context learning. Img2LLM [9] is a zero-shot VQA method that generates image-relevant exemplar prompts for the LLM. Their key insight is that synthetic question-answer pairs can be generated using image captioning and question-generation techniques as in-context exemplars from the provided image. Prophet [22] proposes to prompt GPT-3 with answer heuristics (answer candidates and answer-aware examples) encoded into the prompts to enable GPT-3 to comprehend the task better, thus enhancing its capacity.

## 3 Method

### 3.1 Overview

Unlike other similar language-mediated methods, we primarily focus on two points. Firstly, the model should accomplish the QA tasks more humanly and be more practical in the actual application scene, so we stick to the zero-shot setting. Secondly, better describing the question-related image content is crucial in enhancing the model's performance. As you correctly capture the critical information in the image that is indispensable to answering the question, the LLMs usually behave well and generate the desired answer. Accordingly, to improve the performance under the zero-shot setting, we construct a detailed task instruction prompt to guide the LLMs in tackling the problem. To provide better image captions, we deploy a text-guided captioning architecture that differs from the traditional method. We use GPT-4 to build a new dataset to enhance the model's ability.

Our VQA pipeline is illustrated in Figure 1. The pipeline consists of two components, QAC and LLM. Our QAC model takes in the image and the question and outputs a question-relevant image captioning, serving as the context. The prompt given to the off-shelf LLM comprises the task instruction, the context, and the question. Then, the LLM generates the answer.

Since large language models can not perceive the visual modality, we convert the image into textual descriptions. The main challenge is that a general caption of images may fail to offer the necessary information to answer the question, thus resulting in the LLMs predicting an error answer. To address this issue, our QAC model takes not only the image but also the corresponding question. Under the text guidance, QAC captures relevant visual information and generates a helpful caption. On the same image, the caption content would vary given different questions. As such, we can convert VQA samples into question-answering examples that LLMs can understand.

Having used QAC to convert VQA examples into question-answer examples that LLMs can understand, we use a carefully designed prompt as the task demonstration for LLMs. Since we work under the zero-shot setting, we need examples showing how to solve the tasks to provide the LLMs with detailed guidance. The task instruction prompt tells the large language model that "you are an expert in answering questions based on context. You are required to thoroughly understand the content of the context, combine it with your own knowledge, and utilize appropriate reasoning to arrive at the answer to the question. However, you should avoid overinterpreting the context. If the information in the context is insufficient to answer the question, you are allowed to make reasonable guesses

to some extent. Finally, it is important to maintain conciseness and accuracy in your answers." We concatenate the task instruction, the context generated by QAC, and the question as a whole prompt into the LLMs, and then the LLMs will generate the final answer. Although in-context learning allows the model to learn how to answer questions better, it may not align with the practical application scenario. We often interact with general artificial intelligence systems by asking questions directly rather than providing examples first. That's why we adopt the zero-shot paradigm, which expects the model to answer questions without specific prior examples.

## 3.2 Training Data Synthesis

### 3.2.1 Training Examples Generation with GPT-4

For QAC training, a data sample should consist of an image, a question, and an image caption that helps answer the question. Therefore, we consider constructing corresponding captions for each image-question pair based on the VQA dataset. PromptCap[11] release such a dataset built upon VQAv2[7] and inspired by their work, we use GPT-4 to create a high-quality question-related caption dataset based on the OK-VQA[18]. In general, we follow the PromptCap pipeline to construct the dataset, but we also make several improvements, as stated later. As a result, our dataset captions describe the images more naturally and in more detail and exhibit a greater diversity of expressions.

In the caption synthetic, we also follow the zero-shot setting. For the image instance in OKVQA, which is from the MSCOCO dataset, we can get the original 5 human-annotated COCO captions. We provide them to the GPT-4 along with the question-answer pair and ask it to fully understand the situation that the image depicts and the information contained in the question-answer pair. Then, the GPT-4 should synthesize 3 candidate captions that are semantically consistent with the original captions and should contain useful information to help answer questions.

### 3.2.2 Training Examples Selection and Refinement

To select the best synthetic caption as the training example, we focus on two aspects: helpfulness and truthness. With regard to helpfulness, we suppose the ideal caption will help the LLMs understand the question situation and deduce the correct answer. So, we follow our VQA pipeline to get the LLM's answers for each training data example using candidate captions. The more accurate the answer corresponding to the caption, the more helpful it is, as it contains useful information that assists in problem-solving. On the other hand, truthness also matters. The captions should accurately describe the image's content and should not include information or objects that do not exist in the image. We calculate the clip score of the image and its candidate captions. The higher clip score means the caption is semantically closer to the image. Under these two considerations, we filter out the best target caption for the training instance. The (a) and (b) in Figure 2 shows two successful examples.

However, although we impose several constraints and requirements in the task instructions to generate our target captions, we discover several deficiencies after manually evaluating the synthetic question-aware caption dataset. Firstly, there is a risk of directly injecting the answer information into the caption that is not present in the image. Secondly, the information fusion is done unnaturally, for example, by using weird expressions and deliberately using phrases like 'known for', 'demonstrating' etc. As a case shown in Figure 2 (c), 'a red bottle likely filled with dish soap' is unnatural, and 'a red soap bottle' is good enough. Last but not least, even though we hope the caption can express more diversely and describe the image content vividly, clarity is also important as the model should be able to capture the most question-relevant key information in the image. So, we rewrite those too-complicated examples like (d) in Figure 2 to maintain brevity. To address these issues, we filter out a subset containing 2294 bad examples by both GPT-4 evaluation and human work. We ask GPT-4 to modify or rewrite the bad cases according to their problem. Thus, we create a high-quality and robust question-relevant image caption dataset based on OK-VQA.

## 3.3 Question-Aware Image Captioning

In our pipeline, we need a lightweight visual module that can convert images into question-relevant captions. In light of this, we introduce QAC, a question-aware image captioning model finetuned from BLIP[14]. QAC takes a question and an image as input and outputs a caption about the image

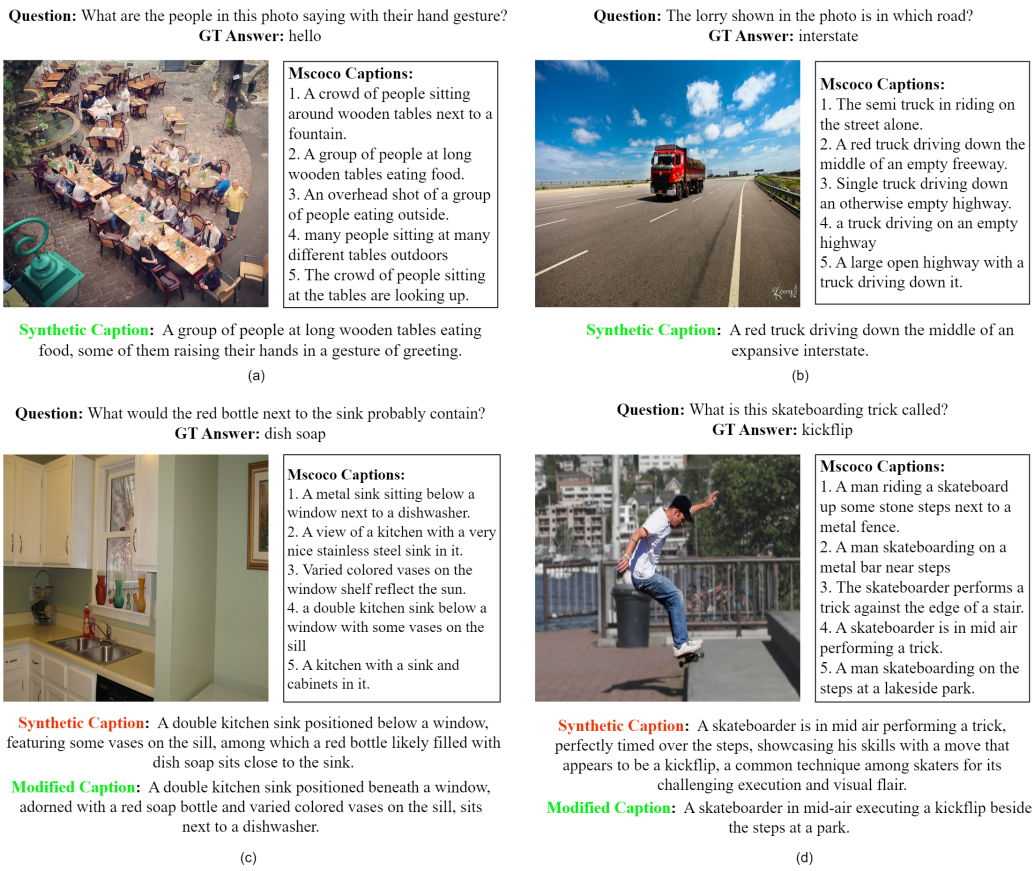


Figure 2: Example synthesis made by GPT-4. Specifically, (a) and (b) are good cases, while (c) and (d) are bad cases that are filtered out to be regenerated.

specific to the question. The caption describes related visual content and provides information for LLMs to infer the answer.

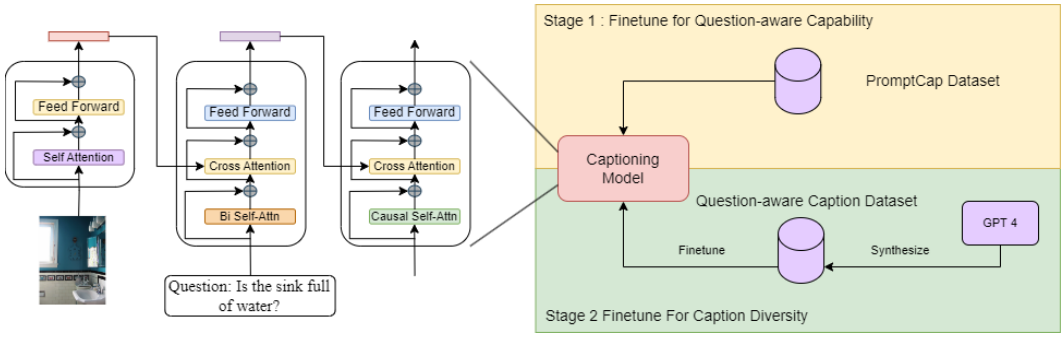


Figure 3: Model architecture and training stages.

BLIP is a multimodal mixture of encoder-decoder framework that is pre-trained on a large amount of image-text pairs. Like most image captioning methods, BLIP has a visual encoder ViT to embed image features and an image-grounded text decoder to generate corresponding captions. Considering the target of capturing useful visual information in the image, which helps to answer the question, we introduce the question as a textual signal to guide the visual feature extraction. Thus, our QAC model consists of the following three parts:

(1) **Visual encoder**, a vision transformer (ViT) that encodes the images.

(2) **Image-text fusion encoder**, this encoder adds one additional cross-attention (CA) layer in each transformer block as the remaining architecture is the same with BERT[4]. The text sequence starts with a special [Encode] token. The question embeddings are Query and image embeddings are Key and Value in the CA layer. The final output embeddings represent the fusion of image-text information.

(3) **Multimodal-grounded text decoder**, which replaces the bidirectional self-attention layers in the image-text fusion encoder with causal self-attention layers. The sequence to be generated autoregressively starts with A [Decode] token.

Since our model should generate different captions for different questions about the same image, we needed to train our model to perceive and respond to various questions. Given that the image-question pairs in OK-VQA are nearly one-to-one, we train our model on the VQAv2 dataset initially provided by the PromptCap project. Subsequently, we enhance the quality of the captions by further fine-tuning on our constructed OK-VQA training dataset. This additional fine-tuning improves the depicting accuracy and expression diversity of the caption generation.

# 4 Experiments

## 4.1 Experiment Setup

**Datasets**. We mainly evaluate our proposed method on knowledge-based VQA dataset OK-VQA. OK-VQA is a commonly used knowledge-based VQA dataset that contains 14,055 image-question pairs associated with 14,031 images from MSCOCO dataset. The questions are manually filtered to ensure all questions require outside knowledge to answer. Each question is annotated with ten open-ended answers. We evaluate our method on the test set.

**Training Details**. We employ a two-stage learning training approach. We perform fine-tuning sequentially on the PromptCap dataset and our own constructed QAC dataset. For the first training state on PromptCap dataset, we initialize the pre-trained model using 'model_base_caption_capfilt_large.pth', the initial learning rate is 2e-5 and the training is performed for 20 epochs. This model is named as QAC-Base. And then we continue to train the QAC-Base on QAC dataset with the same initial learning rate but for 50 epochs. This model is named as QAC-Full.

**Evaluation Metrics**. The commonly used evaluation metric for both OK-VQA is the soft accuracy proposed in VQAv2[2], where

$$\text{Acc}(ans) = \min\{\frac{\#\text{humans that said } ans}{3}, 1\}$$

The predicted answer is deemed 100% accurate if at least 3 humans provided the exact answer. However, with the rapid development of probabilistic generative models, the conventional exact matching evaluation can not meet the requirements of open-ended question answering. To address this, we propose a simple yet effective evaluation method. We ask the model to generate a concise and accurate response, and if the reference answer appears accurately in the generated response, we consider it a correct answer. That is, for the instance {"question: Is the boy swimming or doing another water activity?", "answer: another activity"}, the answer "doing **another** water **activity**." would be recognized as correct.

## 4.2 Results on OK-VQA

We compare our proposed method with the state-of-the-art methods on the OK-VQA dataset in Table 1. The compared methods are categorized into two classes. Methods in the first category are multimodal large language models, and methods in the second category explicitly convert images into text descriptions. Note that PICa, Img2LLM, and Prophet, among others, not only utilize the captions, but also make use of additional text information like tags, answer heuristics, etc. KAT and REVIVE leverage LLM combined with the outside knowledge base, Wikipedia. PromptCap is the primary baseline of our work. QAC-Full surpasses PromptCap by 2% in accuracy and achieves state-of-the-art performance in the language-mediated methods. Figure 4 shows two representative successful case of our method.

Table 1: Performance on OK-VQA.

| Methods | Image Representation | Large Language Model | Shot Number | OK-VQA |
|---|---|---|---|---|
| BLIP-2[13] | Feature | FlanT5 | 0 | 45.9 |
| Flamingo[1] | Feature | Chinchilla | 16 | 57.8 |
| CogVLM[27] | Feature | Vicuna-7B | 0 | 58.9 |
| CogAgent[10] | Feature | Vicuna-7B | 0 | 61.2 |
| InstructBLIP[3] | Feature | Vicuna-7B | 0 | 62.1 |
| PaLI-X-VPD[12] | Feature | PaLI-X | 0 | **66.8** |
| PICa-Full[31] | Caption+Tags | GPT-3 | 16 | 48.0 |
| Img2LLM [9] | Caption+Tags | OPT | 16 | 45.6 |
| KAT[8] | Caption+Tags+Feature | GPT-3+Wikidata | 0 | 54.4 |
| REVIVE[15] | Caption+Feature | GPT-3 +Wikidata | 16 | 58.0 |
| PromptCap[11] | Caption | GPT-3 | 32 | 60.4 |
| TOA[30] | Caption+Visual Evidence | ChatGPT | 24 | 60.6 |
| Prophet[22] | Caption+Answer Heuristics | GPT-3 | 20 | 61.1 |
| QAC-Base | Caption | GPT-4 | 0 | 62.1 |
| QAC-Full | Caption | GPT-4 | 0 | **62.4** |

**Q:** What are these people doing?
**C:** a married couple, dressed in formal attire, are actively slicing into a large cake together at their wedding reception table
**A:** The couple is **cut**ting their **wed**ding **cake**.
**GT:** cut cake/ cut wed cake

**Q:** What activity are they doing?
**C:** two people engaged in a competitive video game, each holding a nintendo wii remote control and actively participating in their gaming activities
**A:** They are playing a **video game** on the Nintendo **Wii**.
**GT:** video game/ wii

Figure 4: Representative successful cases of QAC with GPT-4 on the OK-VQA. *"Q"*, *"C"*, *"A"*, and *"GT"* denote question, context, GPT-4 answer, and ground truth answers respectively.

### 4.3 Ablation Study on QAC dataset

We train the QAC model on two datasets, the PromptCap dataset and the QAC dataset. We calculate the dataset size, the total vocabulary size, and the average sentence lengths of both datasets. The result is shown in the Table 2. The PromptCap dataset consists of 443757 instances, while our dataset containing 9009 examples is relatively small. However, our dataset have a longer average caption length. Then, we collect the model performance on the OK-VQA validation set. In the previous experiment, we utilize the QAC-Base, and QAC-Full to generate captions. On one hand, QAC-Full outperforms the QAC-Base by 0.3% accuracy on OK-VQA. On the other hand, as shown in Table 3, the captions generated by QAC-Base on the OK-VQA validation set have a total vocabulary size of 3185, while that of QAC-Full has 5744. On the same image set, QAC-Full exhibits more diverse expressions and has twice the length of the caption sentence. QAC-Full not only helps the GPT-4 get a performance gain of 0.3% but also achieves a higher average clip score which means the captions are more related with the iamge. Although our dataset's amount is small, the performance gain in caption generation is huge.

Table 2: Training caption dataset quality analysis of PromptCap and QAC.

| Dataset | data instance | vocabulary size | average length |
|---|---|---|---|
| PromptCap | 443757 | 30430 | 10.50 |
| QAC | 9009 | 14920 | 19.22 |

Table 3: Captioning model performance on the OK-VQA validation set.

| Model | vocabulary size | average length | average clip score |
|---|---|---|---|
| QAC-base | 3185 | 9.65 | 24.40 |
| QAC-full | 5744 | 18.46 | 24.90 |

### 4.4 Limitations

As Figure 5 shows two failure cases, we summarize several deficiencies. For the left example, while our QAC model grasp the key clue, i.e., the wine label to describe, QAC fails to capture the correct information due to a weak OCR ability. For the right example, although the GPT-4 gives a nearly correct answer, the case fails due to the simple evaluation method. It is worthwhile to pay attention to the evaluation of open-ended answers, and we believe that more efforts should be dedicated to addressing this problem given its increasing demand in the future.

## 5 Conclusion

We present QAC, a novel question-aware image captioning model that can capture relevant information from the image based on the textual question. To improve QAC's performance, we additionally construct a high-quality caption dataset using GPT-4. The pipeline, QAC with GPT-4, works effectively on knowledge-based VQA tasks. Our system achieves state-of-the-art accuracy of 62.4% on OK-VQA. Ablations show that the QAC dataset gives consistent gains over the baseline PromptCap dataset. QAC is a lightweight, plug-and-play visual module that connects with large language models. Despite the prevalence of multimodal large language models, our work proves to be an effective way to address the intricate knowledge-based VQA.


**Q:** What brand is this wine?
**C:** a wine bottle with a matching wine labels are positioned adjacent to a glass, which is clearly displayed the bottle is crafted by balzac wine
**A:** The brand of this wine is **Balzac**.
**GT:** becker vineyard/ becker

**Q:** What is the horsepower of the truck pulling the load?
**C:** a tow truck is towing a blue freight train, signaling its purpose for transporting
**A:** Based on general knowledge, trucks capable of towing such heavy loads typically have engines that produce anywhere from **300 to 600** horsepower.
**GT:** 510/ 420/ 500

Figure 5: Representative failure cases of QAC with GPT-4 on the OK-VQA.

# References


[1] 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. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[3] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.

[6] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

[7] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[8] Liangke Gui, Borui Wang, Qiuyuan Huang, Alex Hauptmann, Yonatan Bisk, and Jianfeng Gao. Kat: A knowledge augmented transformer for vision-and-language. *arXiv preprint arXiv:2112.08614*, 2021.

[9] Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven Hoi. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10867–10877, 2023.

[10] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.

[11] Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A Smith, and Jiebo Luo. Promptcap: Prompt-guided task-aware image captioning. *arXiv preprint arXiv:2211.09699*, 2022.

[12] Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. *arXiv preprint arXiv:2312.03052*, 2023.

[13] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[14] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[15] Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. Revive: Regional visual representation matters in knowledge-based visual question answering. *Advances in Neural Information Processing Systems*, 35:10560–10571, 2022.

[16] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[17] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14111–14121, 2021.

[18] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[19] Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. *Advances in neural information processing systems*, 31, 2018.

[20] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[21] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer, 2022.

[22] Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. Prompting large language models with answer heuristics for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14974–14983, 2023.

[23] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[24] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[25] Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2413–2427, 2017.

[26] Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony Dick. Explicit knowledge-based reasoning for visual question answering. *arXiv preprint arXiv:1511.02570*, 2015.

[27] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

[28] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7959–7971, 2022.

[29] Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. Multi-modal answer validation for knowledge-based vqa. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2712–2721, 2022.

[30] Xiaoying Xing, Mingfu Liang, and Ying Wu. Toa: Task-oriented active vqa. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[31] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3081–3089, 2022.

[32] 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.

[33] Zihao Zhu, Jing Yu, Yujing Wang, Yajing Sun, Yue Hu, and Qi Wu. Mucko: multi-layer cross-modal knowledge reasoning for fact-based visual question answering. *arXiv preprint arXiv:2006.09073*, 2020.

# NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract accurately reflect the paper's contributions and scope.

   Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See in the Section 4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper has no theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: See in the Section 3 and Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release them as allowed.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: See in the Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: See in the Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: We will provide them soon after.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research is conducted under Code.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: Not discussed yet.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: Our work doesn't involve high risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets used in the paper are explicitly mentioned.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: We will write the document soon after.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.