# Attention Prompting on Image for Large Vision-Language Models

Runpeng Yu, Weihao Yu[†], and Xinchao Wang[†]

National University of Singapore
{r.yu,weihaoyu}@u.nus.edu, xinchao@nus.edu.sg

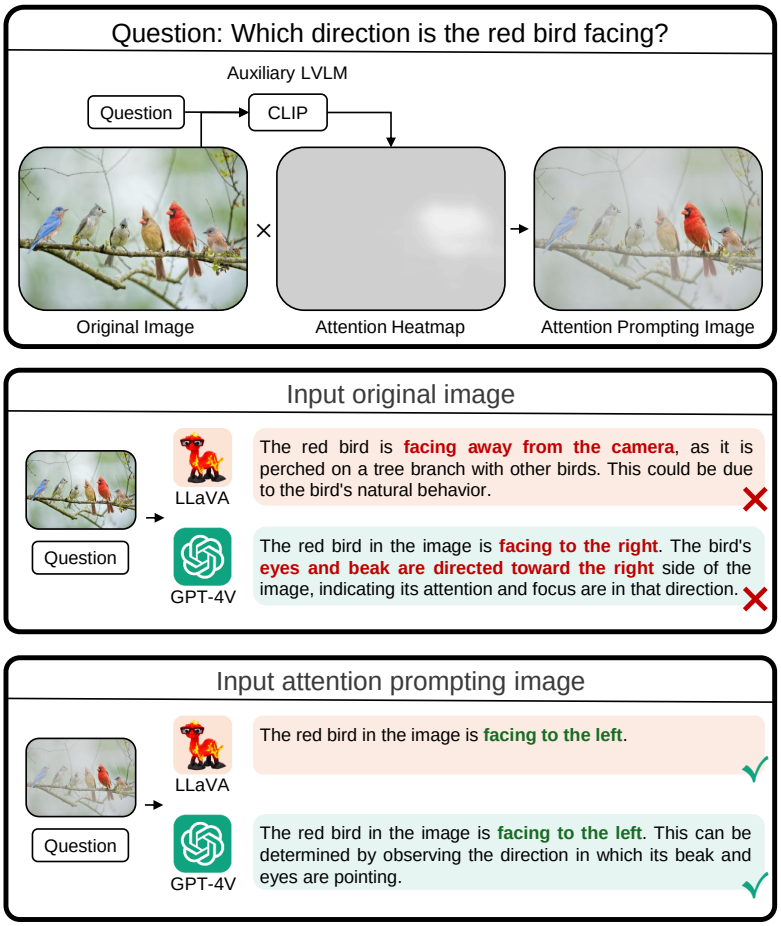


**Fig. 1:** Comparison of the proposed Attention Prompting on Image ($\mathcal{API}$) with the naive VQA.$\mathcal{API}$ provides hints for LVLM by simply overlying a heatmap on the image.

[†] Corresponding author.

**Abstract.** Compared with Large Language Models (LLMs), Large Vision-Language Models (LVLMs) can also accept images as input, thus showcasing more interesting emergent capabilities and demonstrating impressive performance on various vision-language tasks. Motivated by text prompting in LLMs, visual prompting has been explored to enhance LVLMs' capabilities of perceiving visual information. However, previous visual prompting techniques solely process visual inputs without considering text queries, limiting the models' ability to follow text instructions to complete tasks. To fill this gap, in this work, we propose a new prompting technique named Attention Prompting on Image ($\mathcal{API}$), which just simply overlays a text-query-guided attention heatmap on the original input image and effectively enhances LVLM on various tasks. Specifically, we generate an attention heatmap for the input image dependent on the text query with an auxiliary model like CLIP. Then the heatmap simply multiplies the pixel values of the original image to obtain the actual input image for the LVLM. Extensive experiments on various vison-language benchmarks verify the effectiveness of our technique. For example, $\mathcal{API}$ improves LLaVA-1.5 by 3.8% and 2.9% on MM-Vet and LLaVA-Wild benchmarks, respectively.

**Keywords:** Visual Prompting · Large Vision-Language Model · Large Multimodal Model

## 1 Introduction

Benefiting from the great progress of Large Language Models (LLMs) [1,52,53], Large Vision-Language Models (LVLMs) [2,4,8,17,25,31,65,66,79] also advances rapidly, represented by the seminal works GPT-4V [65] and LLaVA [31].[1] They have been widely applied in tasks that involve understanding both visual and linguistic information, such as referring segmentation [71, 72], localization [71], captioning [54], open world 2D/3D understanding [51,56,65,80], and image editing [62, 65].

To enhance the performance of LVLMs, an economical method is to develop prompting techniques to elicit the models' potential. Similar to textual prompting [23, 60], visual prompting[2] [63, 64] is a technique that enhances a model's understanding of images by directly adding annotations such as masks, circles, and marks to the image. This technique provides clear hints for visual perception by highlighting areas relevant to solving the problem, guiding the model's attention to specific parts of the image, thus mitigating issues arising from complex scenes with distractions. It has been demonstrated that even simple visual

[1] Although also referred to as Multimodal Large Language Model (MLLM) or Large Multimodal Model (LMM) [31, 65], we use Large Vision-Language Model (LVLM) to refer to the models discussed in this paper, as we primarily utilizes the model's vision and language capabilities.

[2] In this work, we specifically use "visual prompts" to refer to masks, circles, marks, and other annotations added to images and use "visual prompting" to refer to technologies that employ visual prompts to assist in VQA tasks.

cues like circles [47], arrows [65], or image tiling [29] can improve LVLMs' ability to extract the required information correctly. Unlike methods that improve LVLM performance through adaptation or fine-tuning, visual prompting does not require the training process, thereby reducing the risks of overfitting and knowledge forgetting. Moreover, compared to textual prompts, visual prompting is more direct and precise in guiding the model's focus to specific areas. Textual descriptions cannot succinctly describe an irregular area in an image or accurately indicate the location of a specific object, and they also face issues with aligning textual coordinates with actual image pixels [65]. However, compared to research on textual prompts and LVLM fine-tuning, visual prompting is still underexplored.

Previous visual prompting techniques focused on designing appropriate fine-grained annotations to the image, aiming to highlight important local areas without impairing the model's overall understanding of the image. Remarkably, FGVP [64] and SoM [63] are both based on segmentation masks [63]: The former blurs the image outside the segmentation mask while the latter overlays the image with a set of including alphanumerics, masks, and boxes. However, all these methods sorely process the input images without considering the text query content. In other words, whatever the text query is, an image's visual prompting results are the same. This can easily lead to a mismatch between the prompted image and the text query, as different text queries for the same image require focus on different areas and necessitate different annotations. This mismatch may thereby limit the model's ability to follow instructions accurately.

To address this issue, in this paper, we propose a novel prompting technique named Attention Prompting on Image ($\mathcal{API}$), which just simply overlays a text-query-guided attention heatmap on the original input image. Specifically, to generate text-guided attention heatmap for an image, we utilize an auxiliary LVLM that can accept both image and text as input. For image-text matching type (like CLIP [39]) as auxiliary model, we devised a heatmap generation technique based on the decomposition of cls token similarity score. For the vision-language-input text generation model (like LLaVA [31]), we generate the heatmap based on attention weights. Extensive experiments on various commonly used vision-language (VL) datasets verify the effectiveness of $\mathcal{API}$ in enhancing the VLM's perception of visual information. For example, $\mathcal{API}$ improves LLaVA-1.5 by 3.8%, 2.9%, and 2.3% on MM-Vet, LLaVA-Bench and MMMU benchmarks

Our contributions can be summarized as follows:

1. We find that current visual prompting techniques sorely modify input images without considering the text query, limiting the model's capability to follow instructions accurately.
2. To fill the gap, we propose the $\mathcal{API}$ method, exploring how to derive valuable attribution maps from various types of VLM models and utilize them as visual prompts to offer hints for visual perception, thereby boosting performance.

3. Our experiments demonstrate the effectiveness of our method across a wide range of VLM models on various datasets. Moreover, our approach has also proven effective in addressing the issue of hallucination.

## 2 Related Works

### 2.1 Visual Prompting for LVLM

Originating from language models [32,33,43], the concept of prompting has been widely applied in vision models and vision language models to enhance the transfer learning and adaptation for various tasks (*e.g.,* classification [20,40,75,77,78], detection [12, 19, 57], segmentation [41] and generation [61]) and under various learning settings (*e.g.,* few-shot learning [48], continual learning [58, 59, 76], domain adaptation/generalization [13, 36], unlearning [73], and long-tailed learning [10]). It is crucial to distinguish our work from soft prompts generated through gradient optimization and related prompt-tuning efforts. These prompts, concatenated in the form of continuous vectors to the token sequence of the VL model's transformer layer input [7, 18, 49], or added to the input image as optimizable pixel patches and paddings [21, 44], depend on an additional learning process. Thus, they are strongly coupled with the model and dataset, lack generalizability, and are not intuitively interpretable. Moreover, since (part of) these prompts are incorporated at a shallow layer, their optimization process involves gradient propagation throughout the entire branch, which is costly. Unlike these methods, the visual prompting studied in this paper is manually designed and automatically generated by extra LVLMs. It is interpretable and generalizable across different models and tasks.

Visual prompting is a specialized technique in vision models, especially for segmentation tasks [22,26,38]. Based on an additionally trained prompt encoder, manually annotated points, strokes, boxes, and irregular masks can provide these models with extra instructions to assist in controlling segmentation granularity or in facilitating instance selection. Recently, LVLMs have also been shown to understand manually added circles and color masks in images in a zero-shot manner, focusing attention on highlighted areas without relying on additional encoder components [47, 67]. Unlike these works that explore the LVLM's ability to understand visual prompts, our method discusses how to use pretrained LVLMs to automatically generate visual prompts to enhance image readability.

The two methods most related to ours are [63] and [64], which modify masks generated by segmentation models to construct visual prompts to improve LVLM's performance in segmentation and grounding tasks. Our method differs fundamentally from theirs in that we use LVLMs to construct visual prompts. This leads to two main differences in functionality and applicability. 1) For a single image, the visual prompts generated by [63, 64] are invariant, as these models rely on fixed segmentation models. In contrast, with different text queries, our method can adapt and generate distinct visual prompts to emphasize different areas as required. 2) The visual prompts generated by [63, 64] are essentially instance-specific proposals for segmentation and grounding tasks,

focusing on enhancing the LVLM's grounding capability. Conversely, our visual prompts aim to highlight important areas needed to address text queries, thereby improving the LVLM's performance in general Visual Question Answering tasks.

### 2.2 Self-Reflection and Ensemble

Our method involves LVLM at two stages: once for generating visual prompts and once for performing inference. When the same LVLM is used at both stages, our approach can be seen as a method to enhance LVLM performance using self-reflection technology. The concept of Self-Reflection originated from LLMs [37, 46] but can be directly transferred to LVLMs. Self-Reflection scheme improves model performance by repeatedly answering a query and iteratively updating the answer. The Self-Reflection process involves using self-evaluation [3], self-checking [35], self-feedback [34], feedback from the external environment [6, 45], and even previous answers themselves [74] as hints to input into the model for it to answer the question again. Unlike these works, where the medium of self-reflection is text, our method employs visual prompting to achieve Self-Reflection in the pixel space.

When different LVLMs are used at two stages, our method can be considered a form of model ensemble, where the knowledge of the first VLM is ensembled into the second VLM in the form of visual prompts. In tasks with standard outputs, deep learning model ensemble involves aggregating outputs from multiple models [15]. However, output aggregation is invalid in generation tasks. In LLMs and LVLMs, model ensemble is achieved in the form of sequential or stage-wise use between auxiliary and inference models. The final inference model can enhance its performance by incorporating outputs from other auxiliary models into its input. The auxiliary model outputs used as inputs for the inference model can be responses from another language model [11] or textualized outputs from vision models or vision-language models (image captions, category names) [27, 70]. Unlike these works, our method uses visual rather than textual signals for ensembling. Furthermore, our approach does not ensemble the final hard outputs of auxiliary models but their visual cues used during the inference process. This soft knowledge ensemble provides valuable auxiliary information and mitigates error accumulation introduced by mistakes in auxiliary model inference.

## 3 Method

Large Vision Language Model $f$ takes an image $I \in \mathbb{R}^{H \times W \times 3}$ and a text query $T^i$ as inputs, generating an output text $T^o = f(I, T^i)$. During the inference process using $\mathcal{APT}$, instead of being directly fed into $f$, the original image $I$ undergoes an additional annotation operation $\mathcal{A}$, resulting in an image $I^a = \mathcal{A}(I, T^i)$ that has been overlaid with a heatmap $\Phi$. Subsequently, the annotated image $I^a$ and the original query are input into the LVLM model $f$, producing the output $T^o = f(I^a, T^i)$. The overall framework of the method is shown in Fig. 1.

In our method, the annotation process comprises two steps. The first step involves using an auxiliary LVLM model $g$ to establish an initial attribution map $\Psi$ between the text query $T^i$ and each patch of the image. This attribution map indicates which patches in the image are more relevant to $T^i$ or which patches should be paid more attention to for answering $T^i$. In our method, there are no additional constraints on the LVLM $g$; if the inference LVLM $f$ is accessible and capable of performing the annotation operation $\mathcal{A}$, then the LVLM $g$ used to generate the attribution map can be the same as $f$, *i.e.*, $g = f$. Alternatively, $g$ could be a different LVLM to introduce knowledge from other models to enhance $f$'s functionality, *i.e.*, $g \neq f$. Moreover, due to the diversity of LVLM models, we do not necessarily use the attention map as our attribution map. For example, for the image-text matching model, experiments have shown that using the attention map as the attribution map has suboptimal results. After obtaining the attribution map $\Psi$, the second step in the annotation process is to convert it into a suitable $\Phi$ and apply it to the original image using alpha blending.

Various LVLM models can be utilized to generate attribution maps. We discuss two prevalent and representative LVLM models: CLIP [39], exemplifying image-text matching models, and LLaVA [31], representing vision-language-input text generation models.

### 3.1 Obtaining Attribution Map from CLIP

The CLIP model, $g_{\text{clip}}$, consists of an image encoder and a text encoder, calculating the similarity between an image and a text query in the image-language latent space, $sim(\hat{I}, \hat{T})$, where $\hat{I} = g^{\text{img}}_{\text{clip}}(I)$ and $\hat{T} = g^{\text{text}}_{\text{clip}}(T)$. This similarity measure evaluates the correlation between the entire image and the query. To obtain the attribution map value from the text query to each image patch, we decompose the output image-level similarity $\hat{I}$ and then calculate the similarity of each patch's output with the $\hat{T}$.

The decomposition process is as follows. Due to the presence of residual connections, the final output of the vision tower, $\hat{I}$, actually includes influences from each layer. Consequently, $\hat{I}$ can be expressed as a linear combination of the values at the class token positions from each layer

$$\hat{I} = \mathcal{L}([Z^0_{\text{cls}}]) + \sum_{l=1}^{L} \mathcal{L}(\left[\text{MSA}^l(Z^{l-1})\right]_{\text{cls}}) + \sum_{l=1}^{L} \mathcal{L}(\left[\text{MLP}^l(\hat{Z}^l)\right]_{\text{cls}}), \tag{1}$$

where $L$ denotes the number of transformer layers within the vision encoder, with MSA and MLP representing the Multihead Self-Attention structure and the Multi-Layer Perceptron structure within the transformer, respectively; $\mathcal{L}$ represents the linear transformation that includes the fully-connected layer and the normalization operations performed after the transformer structure, before calculating the similarity score; $Z^l$ signifies the input token sequence for the $l$-th transformer layer; and $[Z]_{\text{cls}}$ indicates the value of the cls token within the token sequence $Z$. These output cls tokens are aggregated through residual connections

to form the output of the vision encoder. As evidenced in [16, 31], among these summation terms, the outputs of the last few layers of MSA play a decisive role, while the contributions from the outputs of the shallow MSA layers, the outputs of MLP, and the $Z^0_{\text{cls}}$ term, which is independent of the input image, can be considered negligible to the final measurement of similarity. Therefore, the similarity $sim(\hat{I}, \hat{T})$ can effectively be approximated by calculating the similarity between $\hat{T}$ and the aggregated outputs of MSAs in the deeper layers :

$$sim(\hat{I}, \hat{T}) \approx sim(\sum_{l=L'}^{L} \mathcal{L}(\Big[\text{MSA}^l(Z^{l-1})\Big]_{\text{cls}}), \hat{T}), \tag{2}$$

where $L'$ represents a predefined starting layer index. To further calculate the attribution of the text query to each patch, inspired by [16], we unfold the operations of the Multihead Self-Attention, obtaining

$$\Big[\text{MSA}^l(Z^{l-1})\Big]_{\text{cls}} = \sum_{h}^{H} \Big[A^{(l,h)} V^{(l,h)} W^{(l,h)}\Big]_{\text{cls}} + B^l \tag{3}$$

$$= \sum_{t=1}^{T} \underbrace{\left[\sum_{h}^{H} A^{(l,h)}_{\text{cls},t} V^{(l,h)}_{t,:} W^{(l,h)} + \frac{1}{HT} B^l\right]}_{\substack{\text{The MSA output corresponding to} \\ \text{the } t\text{-th patch(token)}}} \triangleq \sum_{t=1}^{T} \eta^l_t, \tag{4}$$

where $A^{(l,h)}$, $V^{(l,h)}$ are the attention map and the value matrix in the $l$-th layer corresponding to the $h$-th head, respectively; $W^{(l,h)}$ is the weight matrix in the $l$-th layer used to merge the multiple attention heads and corresponds to the $h$-th head; $B^{(l)}$ is the bias matrix in the $l$-th layer used to merge the multiple attention heads; $A^{(l,h)}_{\text{cls},t}$ denotes the attention value of the class token towards the $t$-th token in $A^{(l,h)}$, and $V^{(l,h)}_{t,:}$ represents the $t$-th row of $V^{(l,h)}$; $H$ and $T$ are the number of attention heads and the number of tokens, respectively; and the value $T$ equals the number of patches $P \times P$ plus one.

Consequently, by summing across layers and incorporating the final linear transformation, we obtain $\psi_t \triangleq \sum_{l=L'}^{L} \mathcal{L}(\eta^l_t)$, which is the direct influence of the $t$-th patch to the similarity in Eq. (2), allowing us to calculate the similarity between text query and the $t$-th image patch. Accordingly, the attribution map $\Psi^{cls} \in \mathbb{R}^{P \times P}$ is defined as

$$\Psi^{cls}_{i,j} \triangleq sim(\psi_t, \hat{T}), \qquad \text{where } t = 1 + j + P * (i-1). \tag{5}$$

By decomposing the cls token, we can identify which patches are more relevant to the query. This approach is particularly effective when the query contains specific entities, allowing for accurate grounding. However, in complex Visual Question Answering (VQA) tasks, there are often no explicit entities mentioned in the query, or the logic and analysis process involved in answering the question may rely on entities that are not explicitly mentioned in the query. To address

this issue, we also define another complementary attribution map $\Psi^{comp}$ using the CLIP model. This map is designed to capture patches that have potential or implicit relevance to the query.

We experimentally observe that, in the vision transformer of CLIP, the similarity score of the query feature $\hat{T}$ and tokens other than the cls token in the final layer can (inversely) select the important regions. Patches corresponding to the image background or large monochrome areas have a significantly higher similarity score with $\hat{T}$ than those tokens representing specific entities (which may not necessarily appear in the query). This phenomenon is similar to observations made in [9]. Drawing on analyses of the transformer's mechanism in [9, 42], a potential explanation is that these "blank" tokens, lacking valuable information themselves, are treated by the transformer as registers. The transformer initially utilizes them to store information from other informative tokens, subsequently filtering and aggregating this stored information to the class token via the attention mechanism to formulate the final prediction. Therefore, tokens other than the class token, with a high similarity score to $\hat{T}$, represent patches with low information content that can be disregarded. We define the complementary attribution map as follows

$$\Psi^{comp}_{i,j} \triangleq 1 - sim(\mathcal{L}(Z^L_t), \hat{T}), \qquad \text{where } t = 1 + j + P * (i-1), \tag{6}$$

where $Z^L_t$ is the $t$-th output token from the last transformer layer. The complementary attribution map is inversely related to similarity, suggesting that patches lacking information are ignored, retaining only those with potential relevance.

Thus, we obtain two attribution maps that complement each other: $\Psi^{cls}$ explicitly identifies patches directly related to entities in the query but may miss some potentially relevant patches. $\Psi^{comp}$ equally identifies all patches with potential relevance but lacks specificity and cannot highlight those directly related to entities in the query.

By integrating the two attribution maps through the following operation, we obtain the final attribution map for CLIP:

$$\Psi_{i,j} \triangleq \Psi^{cls}_{i,j} + \Psi^{comp}_{i,j} - \Psi^{comp}_{i,j} * \Psi^{cls}_{i,j}. \tag{7}$$

This integration can be considered as a soft OR operation (a detailed mathematical explanation is provided in the Appendix). This ensures that the final attribution map highlights patches directly related to entities within the query while retaining those with potential or implicit relevance, merely reducing the weights of patches that do not contain important information for the query. If the function of the final attribution map were described as an algorithm, then this attribution map would, in the first step, apply a mask to all non-informative patches, making them less considered in subsequent VQA processes while leaving other patches unaffected; and, in the second step, for patches not masked, if a patch is directly related to the entities in the query, it further highlights this patch.

### 3.2 Obtaining Attribution Map from LLaVA

The LLaVA model is an auto-regressive vision-language-input text generation model that utilizes Multihead Self-Attention to extract information from text queries and image patches, predicting the following tokens. Given a text token sequence of length $N$, $Z^{\text{text}} = \{Z_t^{\text{text}}\}_{t=1}^{N}$, and an image token sequence of length $P \times P$, $Z^{\text{img}} = \{Z_t^{\text{img}}\}_{t=1}^{P\times P}$, LLaVA generates a new token sequence of length $M$, $Z^{\text{out}} = \{Z_t^{\text{out}}\}_{t=1}^{M}$. We directly use the attention weight between token $Z_t^{\text{out}}$ and each image token as $Z_t^{\text{out}}$'s attribution to that image patch. Similar to the strategy for the CLIP model, we select attention maps from the deeper layer to extract attention weights. The final attribution map is averaged over the entire generated token sequence and all attention heads. Formally, the attribution map $\Psi$ is defined as

$$\Psi_{i,j} \triangleq \frac{1}{MH} \sum_{m=1}^{M} \sum_{h=1}^{H} A_{m,t}^{(\bar{L},h)}, \qquad \text{where } t = j + P * (i-1). \tag{8}$$

In the definition, $A^{(\bar{L},h)}$ is again the attention map in the $\bar{L}$-th layer corresponding to the $h$-th head, where $\bar{L}$ is a set to be a hyper-parameter; for notation simplicity, $A^{(\bar{L},h)}$ here is a submatrix of the entire attention map and only includes cross attention between $Z^{\text{out}}$ and $Z^{\text{img}}$; $A_{m,t}^{(\bar{L},h)}$ still denotes the attention value from the $m$-th token to the $t$-th token.

### 3.3 From Token Space to Pixel Space

The attribution map $\Psi \in \mathbb{R}^{P\times P}$ is generated in the token space. We first resize it back to the pixel space to obtain the raw heatmap $\hat{\Phi} \triangleq \text{Resize}(\Psi)$. Due to the square shape of the patches, the mask pattern in $\hat{\Phi}$ also appears rectangular. To mitigate the issue that the rectangular mask pattern does not align with the object's irregular shape, we apply a mean filter to obtain the final heatmap $\Phi \triangleq \text{Mean}_k(\hat{\Phi})$, where $k$ is the kernel size of the filter. The final heatmap $\Phi$ is then overlaid on the original image by using it as the alpha channel, resulting in the final image after annotation $I^a$.

## 4 Experiments

We show the main experimental results in this section. More experiments and implementation details are in the appendix.

### 4.1 Comprehensive VQA Tasks

**Datasets.** Experiments are conducted on 6 datasets: VisWiz [5], TextVQA [50], MMMU [69], MME [14], MM-Vet [68], and LLaVA-Bench [31]. The performance on the first four datasets is evaluated using matching accuracy with the ground

**Table 1:** Comparison of our method with previous textual and visual prompting methods for various LVLMs. The best result are marked for each model-dataset pair.

| Inference Model | Prompting Method | Dataset | | | | | |
|---|---|---|---|---|---|---|---|
| | | VisWiz | TextVQA | MMMU | MM-Vet | MME | LLaVA-Bench |
| LLaVA | w/o prompt | 60.93 | 48.32 | 35.15 | 32.8 | 85.5 | 71.9 |
| | +Step-by-Step | 60.98 (+0.1) | 48.22 (−0.1) | 35.40 (+0.3) | 33.7 (+0.9) | 84.2 (−1.3) | 73.5 (+1.6) |
| | FGVP (Mask) | 56.89 (−4.0) | 39.38 (<- 5) | 36.14 (+1.0) | 31.0 (−1.8) | 75.8 (<- 5) | 57.4 (<- 5) |
| | FGVP (RBM) | 61.22 (+0.3) | 33.91 (<- 5) | 35.00 (−0.2) | 25.0 (<- 5) | 81.4 (−4.1) | 57.4 (<- 5) |
| | SoM | 54.16 (<- 5) | 18.81 (<- 5) | 35.57 (+0.4) | 26.4 (<- 5) | 75.4 (<- 5) | 56.1 (<- 5) |
| | Ours (CLIP) | 61.26 (+0.3) | 48.78 (+0.5) | **37.52** (+2.4) | 35.3 (+2.5) | **87.2** (+1.7) | 74.1 (+2.2) |
| | Ours (LLaVA) | **61.35** (+0.4) | **48.79** (+0.5) | 36.95 (+1.8) | **36.6** (+3.8) | 86.3 (+0.8) | **74.8** (+2.9) |
| CogVLM | w/o prompt | 53.54 | 78.41 | 36.43 | 49.6 | 81.8 | 50.8 |
| | +Step-by-Step | 28.86 (<- 5) | 42.53 (<- 5) | 29.19 (<- 5) | 48.0 (−1.6) | 63.0 (<- 5) | 40.7 (<- 5) |
| | FGVP (Mask) | 53.55 (+0.0) | 63.69 (<- 5) | 35.34 (−1.1) | 44.1 (<- 5) | 80.4 (−1.4) | 49.1 (−1.7) |
| | FGVP (RBM) | 53.68 (+0.1) | 65.51 (<- 5) | 36.55 (+0.1) | 48.2 (−1.4) | 82.0 (+0.2) | 48.1 (−2.7) |
| | SoM | 51.00 (−2.5) | 36.64 (<- 5) | 35.55 (−0.9) | 31.2 (<- 5) | 78.0 (−3.8) | 38.9 (<- 5) |
| | Ours (CLIP) | 54.01 (+0.5) | **78.99** (+0.6) | **37.05** (+0.6) | **52.5** (+2.9) | 82.3 (+0.5) | **53.3** (+2.5) |
| | Ours (LLaVA) | **54.34** (+0.8) | 78.85 (+0.4) | 36.95 (+0.5) | 52.0 (+2.4) | **82.7** (+0.9) | 52.4 (+1.6) |
| GPT-4V (1106) | w/o prompt | 59.40 | 50.60 | 50.55 | 67.00 | 84.3 | 102.0 |
| | +Step-by-Step | 55.75 (−3.6) | 49.85 (−0.7) | 48.33 (−2.2) | 62.50 (−4.5) | 82.0 (−2.3) | 102.6 (+0.6) |
| | FGVP (Mask) | 69.30 (+9.9) | 45.95 (−4.6) | 43.88 (<−5) | 61.00 (<−5) | 65.0 (<- 5) | 59.2 (<- 5) |
| | FGVP (RBM) | 69.40 (+10.0) | 46.15 (−4.4) | 52.50 (+1.9) | 60.20 (<−5) | 79.6 (−4.7) | 92.5 (<- 5) |
| | SoM | 65.30 (+5.9) | 45.00 (<−5) | 48.33 (−2.22) | 58.90 (<−5) | 65.8 (<- 5) | 56.1 (<- 5) |
| | Ours (CLIP) | 69.50 (+10.1) | **51.50** (+0.9) | 50.96 (+0.4) | **67.70** (+0.7) | **85.3** (+1.0) | 103.3 (+1.3) |
| | Ours (LLaVA) | **71.01** (+11.6) | 50.80 (+0.2) | **51.38** (+0.8) | 67.10 (+0.1) | 84.7 (+0.3) | **103.6** (+1.6) |
| Gemini | w/o prompt | 50.28 | 56.68 | 35.11 | 59.0 | 78.6 | 81.5 |
| | +Step-by-Step | 22.82 (<- 5) | 21.51 (<- 5) | 36.37 (+1.3) | 30.6 (<- 5) | 29.8 (<- 5) | 40.5 (<- 5) |
| | FGVP (Mask) | 52.88 (+2.6) | 40.81 (<- 5) | 34.88 (−0.2) | 45.8 (<- 5) | 71.0 (<- 5) | 64.2 (<- 5) |
| | FGVP (RBM) | 53.01 (+2.7) | 45.67 (<- 5) | 34.08 (−1.0) | 52.0 (<- 5) | 77.4 (−1.2) | 82.3 (+0.8) |
| | SoM | 51.25 (+1.0) | 27.29 (<- 5) | 34.77 (−0.3) | 34.4 (<- 5) | 69.8 (<- 5) | 64.5 (<- 5) |
| | Ours (CLIP) | **58.58** (+8.3) | **59.07** (+2.4) | 37.71 (+2.6) | **60.5** (+1.5) | **80.2** (+1.6) | **85.2** (+3.7) |
| | Ours (LLaVA) | 58.17 (+7.9) | 58.35 (+1.7) | **38.16** (+3.1) | 60.1 (+1.1) | 80.0 (+1.4) | 82.3 (+0.8) |

truth response. The performance of the latter two datasets is measured using the GPT-based evaluation scores.

**LVLMs.** Experiments are conducted using two open-source models: CogVLM [55] and LLaVA [30], and two commercial models: GPT-4V [65] and Gemini [51]. Due to GPT-4V's token limit, following the experiment protocol in the previous work [63] when conducting experiments with GPT-4V, for VisWiz, TextVQA, and MMMU, we randomly selected 200 images from the dataset to verify our method. Because, about 50 questions on MM-Vet are categorised as related to personal identification or brand evaluation due to GPT-4V's safety policy and are refused to answers. Therefore, we evaluated our method's performance only on the remaining questions.

**Comparison.** We compare with the following methods: (1) naively feeding the query and image to the model without any prompt; (2) using "Let's think step by step" as a prompt to trigger the model's chain-of-thought process, a method that has been proven to significantly improve zero-shot reasoning performance for LLMs [24]; and (3) two visual prompting methods designed for LVLMs, FGVP [64] and SoM [63]. FGVP is designed to generate diverse visual prompts. We compared the most straightforward method of using a mask as a visual

**Table 2:** Ablation study on the auxiliary VLM Scale. The best result are marked for each auxiliary model-dataset pair.

| Mask Model | MMMU | MME |
|---|---|---|
| w/o prompt | 35.15 | 85.50 |
| CLIP-ViT-B | 36.03 (+0.88) | 83.50 (−2.00) |
| CLIP-ViT-L | 36.21 (+1.09) | 83.50 (−2.00) |
| CLIP-ViT-L-336 | **37.52** (+2.37) | **87.16** (+1.66) |
| LLaVA-7B | 35.86 (+0.71) | 85.66 (+0.16) |
| LLaVA-13B | **36.95** (+1.80) | **86.34** (+0.84) |

prompt and the best-performing method of using a Reverse Blur Mask (RBM) as a visual prompt. Performance improvements/decrements listed in the table are calculated relative to the "w/o prompt" method.

The main observations from the experimental results are as follows: (1) Our method consistently achieves the best performance across all datasets-LVLM pairs in Tab. 1. Regardless of whether the CLIP or LLaVA is used as the auxiliary model, our method leads to performance improvements. For LLaVA, CogVLM, GPT-4V, and Gemini, the average improvements relative to "w/o prompt" are 1.94%, 1.38%, 1.76%, and 3.42%, respectively. Our method performs particularly well on Gemini+VisWiz, with an average improvement of 8.1%. Excluding it, our method appears more effective for open-ended questions, with an average improvement of 2.20% on MM-Vet and LLaVA-Bench, while the average accuracy increase on multiple-choice and true-false datasets is 1.18%. (2) The "let's think step by step" approach, which is significantly effective in LLMs, does not perform well in VQA tasks. We suspect this is because this method cannot enhance the LVLM's visual perception capabilities and may even exacerbate LVLM's hallucination due to its language-oriented prompt nature. (3) Previous visual prompting methods, lacking the ability to adapt to different queries, do not perform well on VQA tasks. Our method is clearly superior to them. This indicates that indiscriminately annotating objects in an image does not effectively assist the model in performing VQA tasks. Visual prompting methods need the ability to adapt to queries.

### 4.2 Ablation Studies

We identify three important factors affecting the performance of our method and conduct ablation studies on them.

**The Power of the Auxiliary Model.** On the MMMU and MME datasets, we used CLIP models and LLaVA models of different scales to generate heatmaps, with LLaVA serving as the inference model, to compare performance. The results are shown in Tab. 2. As the scale of the auxiliary model increased, the performance of our method also improved. Both increasing the depth of the auxiliary

**Table 3:** Ablation study on the mean filter kernel size.

| Kernel Size | MMMU | MME |
|---|---|---|
| w/o filter (kernel size = 1) | 36.09 (+0.94) | 83.70 (−1.80) |
| 3 | **36.95** (+1.80) | 86.20 (+0.70) |
| 7 | 36.32 (+1.17) | **87.14** (+1.74) |
| w/o prompt (kernel size≥ 2W= 2H) | 35.15 | 85.50 |

**Table 4:** Ablation study on the Transformer layer for attribution map extraction. The best result are marked for each auxiliary model-dataset pair.

| Mask Model | Layer Index | MMMU | MME |
|---|---|---|---|
| w/o prompt | *n/a* | 35.15 | 85.50 |
| CLIP | 23 | 36.32 (+1.17) | **87.16** (+1.66) |
| | 22 | **37.52** (+2.37) | 84.80 (−0.70) |
| | 20 | 37.12 (+1.97) | 83.20 (−1.30) |
| | 15 | 36.14 (+0.99) | 83.16 (−1.34) |
| LLaVA | 23 | 36.15 (+1.00) | 83.10 (−1.40) |
| | 22 | 36.49 (+1.35) | 83.00 (−1.50) |
| | 20 | **36.95** (+1.80) | **86.34** (+0.84) |
| | 15 | 36.32 (+1.17) | 83.16 (−2.34) |

model or reducing the patch size to generate attribution map with finer granularity prove to be effective for improving the performance of our method. When the capability of the auxiliary model is insufficient, the masks generated by it could even be detrimental.

**The Kernel Size of the Mean Filter.** To mitigate the limitations of rectangular mask patterns when highlighting irregularly shaped objects, we incorporated a mean filter into our method. We conducted ablation studies on different kernel sizes on the MMMU and MME datasets with LLaVA as the inference model. The results are shown in Tab. 3. Without the mean filter, heatmaps with rectangular patterns could potentially harm the final task's performance. The optimal kernel size varied across datasets, due to the different image complexity and question complexity.

**The Layer for Attribution Map Extraction.** Another factor affecting our method's performance was the layer used for extracting the attribution map. Although we knew that deeper layers, which contain higher-level semantic information, should be used, the specific choice of layer also impacted our method's performance. We conduct ablation on MMMU and MMe datasets using LLaVA as inference model. The results are shown in Tab. 4. For the CLIP model, the last

**Table 5:** The comparison between our method and textual self-reflection method and their combination.

| Prompt Method | LLaVA-Bench |
|---|---|
| w/o prompt | 71.90 |
| textual self reflection | 72.90 (+1.00) |
| ours (LLaVA) | 74.80 (+2.90) |
| + reflection via re-emphasize | 72.70 (+0.80) |
| + reflection via evaluation | **76.10** (+4.20) |

two layers are more effective. However, for LLaVA, directly using the attention maps from the last two layers do not yield good results; the best performance occurred when a mid-to-late layer was used, such as 20-th layer for LLaVA-13B.

### 4.3 Self-Reflection

When the auxiliary LVLM and the inference LVLM are the same, our method can be seen as having a two-round chat with the LVLM. The first round generates an annotated image, where the highlighted areas represent what the LVLM considers important, embedding the LVLM's process of extracting visual information. The second round conducts inference based on the generated annotated image, allowing the LVLM to perform Self-Reflection and refine its previous process of visual information extraction. Unlike previous Self-Reflection methods using text as a medium in LLMs, under the $\mathcal{API}$ framework, all information related to the first answer is stored in the annotated image, and the text response from the first round is not provided to the model in the second round.

As a new perspective of Self-Reflection, we explore two questions: (1) *Can visual mediums also achieve effective Self-Reflection?* To answer this, we compared text-based Self-Reflection and our method using LLaVA as the inference model on the LLaVA-Bench dataset. The results in Tab. 5 show that our method achieves better performance than text-based Self-Reflection, proving that visual mediums can effectively facilitate Self-Reflection.

The second question is: (2) *Can we more effectively utilize visual mediums for Self-Reflection?* Generally, Self-Reflection techniques involve two steps int the second round: first, evaluating the previous answer, and second, combining the evaluation to re-answer the question. However, in our framework, the evaluation process is not included, and the model directly proceeds to inference. Therefore, we designed a new inference process. We input the annotated image and the question into the VLM, prompting it to judge whether the highlighted areas in the image support the answer to the question. If yes, the answer is generated using the annotated image; if not, the answer is generated using the original image. The result (the last row of Tab. 5) shows that this strategy further improves our method. Conversely, when we do not allow the model to perform evaluation and emphasize that the answer lies within the highlighted areas of the annotated

**Table 6:** The performance of our method on hallucination datasets.

| Prompt Method | VisWiz-Unanswerable | POPE |
|---|---|---|
| w/o prompt | 81.41 | 81.00 |
| Ours (CLIP) | 83.83 (+2.42) | 82.81 (+0.81) |
| Ours (LLaVA) | **85.26** (+3.85) | **83.52** (+2.52) |

image, performance decreases (second to last row in Tab. 5). This also proves the importance and effectiveness of the evaluation process when using visual mediums for Self-Reflection.

### 4.4 Other Discussion

**Hallucination.** We also explore our method's ability to assist LVLM in overcoming hallucinations. We conduct two experiments. First, on VisWiz, we calculated the accuracy with which our method and the baseline identify the unanswerable questions. These questions often involve information that does not exist in the image, thus the responses to these questions are based on hallucination. Second, we conduct experiments on a subset of a commonly used LVLM hallucination dataset POPE [28]. The experimental results presented in Tab. 6 demonstrate that our method also has the ability to mitigate hallucination.

## 5 Conclusion

In this work, we introduce a novel visual prompting technique called Attention Prompting on Image ($\mathcal{API}$), which incorporates an auxiliary LVLM to generate an attention heatmap on the image dependent on text query. Our extensive experiments demonstrate the advantages of our prompting method for different LVLMs on various benchmarks. Additionally, our approach offers new insights into using visual signals for LVLM ensembling and LVLM self-reflection.

## Acknowledgement

This project is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Award Number: MOE-T2EP20122-0006).

## References


1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in Neural Information Processing Systems **35**, 23716–23736 (2022)
3. Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H.: Self-rag: Learning to retrieve, generate, and critique through self-reflection. CoRR (2023)
4. Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S., Sagawa, S., et al.: Openflamingo: An open-source framework for training large autoregressive vision-language models. arXiv preprint arXiv:2308.01390 (2023)
5. Bigham, J.P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R.C., Miller, R., Tatarowicz, A., White, B., White, S., Yeh, T.: Vizwiz: nearly real-time answers to visual questions. In: Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology. p. 333–342 (2010)
6. Chen, X., Lin, M., Schärli, N., Zhou, D.: Teaching large language models to self-debug. CoRR (2023)
7. Chowdhury, S., Nag, S., Manocha, D.: Apollo : Unified adapter and prompt learning for vision language models. In: Conference on Empirical Methods in Natural Language Processing, EMNLP (2023)
8. Dai, W., Li, J., Li, D., Tiong, A., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In: Thirty-seventh Conference on Neural Information Processing Systems (2023), `https://openreview.net/forum?id=vvoWPYqZJA`
9. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. CoRR (2023)
10. Dong, B., Zhou, P., Yan, S., Zuo, W.: LPT: long-tailed prompt tuning for image classification. CoRR (2022)
11. Du, Y., Li, S., Torralba, A., Tenenbaum, J.B., Mordatch, I.: Improving factuality and reasoning in language models through multiagent debate. CoRR (2023)
12. Du, Y., Wei, F., Zhang, Z., Shi, M., Gao, Y., Li, G.: Learning to prompt for open-vocabulary object detection with vision-language model. In: IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR) (2022)
13. Fahes, M., Vu, T., Bursuc, A., Pérez, P., de Charette, R.: Pøda: Prompt-driven zero-shot domain adaptation. CoRR (2022)
14. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Qiu, Z., Lin, W., Yang, J., Zheng, X., Li, K., Sun, X., Ji, R.: MME: A comprehensive evaluation benchmark for multimodal large language models. CoRR **abs/2306.13394** (2023)
15. Ganaie, M.A., Hu, M., Malik, A.K., Tanveer, M., Suganthan, P.N.: Ensemble deep learning: A review. Eng. Appl. Artif. Intell. **115**, 105151 (2022)
16. Gandelsman, Y., Efros, A.A., Steinhardt, J.: Interpreting CLIP's image representation via text-based decomposition. In: International Conference on Learning Representations (ICLR) (2024)
17. Gao, P., Han, J., Zhang, R., Lin, Z., Geng, S., Zhou, A., Zhang, W., Lu, P., He, C., Yue, X., et al.: Llama-adapter v2: Parameter-efficient visual instruction model. arXiv preprint arXiv:2304.15010 (2023)

18. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. In: Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP (2021)
19. Guo, Z., Dong, B., Ji, Z., Bai, J., Guo, Y., Zuo, W.: Texts as images in prompt tuning for multi-label image recognition. In: IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR) (2023)
20. Jia, M., Tang, L., Chen, B., Cardie, C., Belongie, S.J., Hariharan, B., Lim, S.: Visual prompt tuning. In: European Conference on Computer Vision (ECCV) (2022)
21. Jia, M., Tang, L., Chen, B., Cardie, C., Belongie, S.J., Hariharan, B., Lim, S.: Visual prompt tuning. In: European Conference on Computer Vision (ECCV) (2022)
22. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. In: arXiv (2023)
23. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. Advances in neural information processing systems **35**, 22199–22213 (2022)
24. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. Advances in neural information processing systems **35**, 22199–22213 (2022)
25. Li, B., Zhang, Y., Chen, L., Wang, J., Yang, J., Liu, Z.: Otter: A multi-modal model with in-context instruction tuning. arXiv preprint arXiv:2305.03726 (2023)
26. Li, F., Jiang, Q., Zhang, H., Ren, T., Liu, S., Zou, X., Xu, H., Li, H., Li, C., Yang, J., Zhang, L., Gao, J.: Visual in-context prompting. CoRR (2023)
27. Li, S., Du, Y., Tenenbaum, J.B., Torralba, A., Mordatch, I.: Composing ensembles of pre-trained models via iterative consensus. In: International Conference on Learning Representations (ICLR) (2023)
28. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. In: The 2023 Conference on Empirical Methods in Natural Language Processing (2023), `https://openreview.net/forum?id=xozJw0kZXF`
29. Lin, Z., Liu, C., Zhang, R., Gao, P., Qiu, L., Xiao, H., Qiu, H., Lin, C., Shao, W., Chen, K., Han, J., Huang, S., Zhang, Y., He, X., Li, H., Qiao, Y.: SPHINX: the joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. CoRR **abs/2311.07575** (2023)
30. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning (2023)
31. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Conference on Neural Information Processing Systems (NeurIPS) (2023)
32. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Comput. Surv. **55**(9), 195:1–195:35 (2023)
33. Ma, X., Fang, G., Wang, X.: Llm-pruner: On the structural pruning of large language models. Advances in neural information processing systems **36**, 21702–21720 (2023)
34. Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B.P., Hermann, K., Welleck, S., Yazdanbakhsh, A., Clark, P.: Self-refine: Iterative refinement with self-feedback. In: Conference on Neural Information Processing Systems (NeurIPS) (2023)

35. Miao, N., Teh, Y.W., Rainforth, T.: Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. CoRR (2023)
36. Niu, H., Li, H., Zhao, F., Li, B.: Domain-unified prompt representations for source-free domain generalization. CoRR (2022)
37. Pan, L., Saxon, M., Xu, W., Nathani, D., Wang, X., Wang, W.Y.: Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. arXiv preprint arXiv:2308.03188 (2023)
38. Pan, T., Tang, L., Wang, X., Shan, S.: Tokenize anything via prompting. CoRR (2023)
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (ICML) (2021)
40. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (eds.) International Conference on Machine Learning (ICML) (2021)
41. Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., Zhou, J., Lu, J.: Denseclip: Language-guided dense prediction with context-aware prompting. In: IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR) (2022)
42. Reddy, G.: The mechanistic basis of data dependence and abrupt learning in an in-context classification task. International Conference on Learning Representations (ICLR) (2023)
43. Sahoo, P., Singh, A.K., Saha, S., Jain, V., Mondal, S., Chadha, A.: A systematic survey of prompt engineering in large language models: Techniques and applications. CoRR (2024)
44. Shen, S., Yang, S., Zhang, T., Zhai, B., Gonzalez, J.E., Keutzer, K., Darrell, T.: Multitask vision-language prompt tuning. CoRR (2022)
45. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: language agents with verbal reinforcement learning. In: Conference on Neural Information Processing Systems (NeurIPS) (2023)
46. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems **36** (2024)
47. Shtedritski, A., Rupprecht, C., Vedaldi, A.: What does CLIP know about a red circle? visual prompt engineering for vlms. In: International Conference on Computer Vision (ICCV) (2023)
48. Shu, M., Nie, W., Huang, D., Yu, Z., Goldstein, T., Anandkumar, A., Xiao, C.: Test-time prompt tuning for zero-shot generalization in vision-language models. CoRR (2022)
49. Shu, M., Nie, W., Huang, D., Yu, Z., Goldstein, T., Anandkumar, A., Xiao, C.: Test-time prompt tuning for zero-shot generalization in vision-language models. In: Conference on Neural Information Processing Systems 2022, NeurIPS (2022)
50. Singh, A., Natarjan, V., Shah, M., Jiang, Y., Chen, X., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR). pp. 8317–8326 (2019)
51. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)

52. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
53. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
54. Wang, T., Zhang, J., Fei, J., Zheng, H., Tang, Y., Li, Z., Gao, M., Zhao, S.: Caption anything: Interactive image description with diverse multimodal controls (2023)
55. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., Xu, J., Xu, B., Li, J., Dong, Y., Ding, M., Tang, J.: Cogvlm: Visual expert for pretrained language models (2023)
56. Wang, W., Ren, Y., Luo, H., Li, T., Yan, C., Chen, Z., Wang, W., Li, Q., Lu, L., Zhu, X., Qiao, Y., Dai, J.: The all-seeing project v2: Towards general relation comprehension of the open world (2024)
57. Wang, W., Cao, Y., Zhang, J., Tao, D.: FP-DETR: detection transformer advanced by fully pre-training. In: International Conference on Learning Representations (ICLR) (2022)
58. Wang, Y., Huang, Z., Hong, X.: S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. CoRR (2022)
59. Wang, Z., Zhang, Z., Lee, C., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J.G., Pfister, T.: Learning to prompt for continual learning. In: CVPR (2022)
60. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems **35**, 24824–24837 (2022)
61. Wu, C.H., Motamed, S., Srivastava, S., la Torre, F.D.: Generative visual prompt: Unifying distributional control of pre-trained generative models. CoRR (2022)
62. Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., Duan, N.: Visual chatgpt: Talking, drawing and editing with visual foundation models. CoRR (2023)
63. Yang, J., Zhang, H., Li, F., Zou, X., Li, C., Gao, J.: Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V. CoRR (2023)
64. Yang, L., Wang, Y., Li, X., Wang, X., Yang, J.: Fine-grained visual prompting. In: Conference on Neural Information Processing Systems (NeurIPS) (2023)
65. Yang, Z., Li, L., Lin, K., Wang, J., Lin, C.C., Liu, Z., Wang, L.: The dawn of lmms: Preliminary explorations with gpt-4v(ision) (2023)
66. Yang, Z., Li, L., Wang, J., Lin, K., Azarnasab, E., Ahmed, F., Liu, Z., Liu, C., Zeng, M., Wang, L.: Mm-react: Prompting chatgpt for multimodal reasoning and action. arXiv preprint arXiv:2303.11381 (2023)
67. Yao, Y., Zhang, A., Zhang, Z., Liu, Z., Chua, T., Sun, M.: CPT: colorful prompt tuning for pre-trained vision-language models. CoRR (2021)
68. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mm-vet: Evaluating large multimodal models for integrated capabilities (2023)
69. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., Sun, H., Su, Y., Chen, W.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. arXiv preprint arXiv:2311.16502 (2023)
70. Zeng, A., Attarian, M., Ichter, B., Choromanski, K.M., Wong, A., Welker, S., Tombari, F., Purohit, A., Ryoo, M.S., Sindhwani, V., Lee, J., Vanhoucke, V., Florence, P.: Socratic models: Composing zero-shot multimodal reasoning with language. In: International Conference on Learning Representations (ICLR) (2023)

71. Zhang, A., Ji, W., Chua, T.: Next-chat: An LMM for chat, detection and segmentation. CoRR (2023)
72. Zhang, Y., Ma, Z., Gao, X., Shakiah, S., Gao, Q., Chai, J.: Groundhog: Grounding large language models to holistic segmentation (2024)
73. Zhang, Z., Zhou, Y., Zhao, X., Che, T., Lyu, L.: Prompt certified machine unlearning with randomized gradient smoothing and quantization. In: Conference on Neural Information Processing Systems (NeurlPS) (2022)
74. Zheng, C., Liu, Z., Xie, E., Li, Z., Li, Y.: Progressive-hint prompting improves reasoning in large language models. CoRR (2023)
75. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR) (2022)
76. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR) (2022)
77. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. Int. J. Comput. Vis. (2022)
78. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. Int. J. Comput. Vis. (2022)
79. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)
80. Zhu, X., Zhang, R., He, B., Zeng, Z., Zhang, S., Gao, P.: Pointclip V2: adapting CLIP for powerful 3d open-world learning. CoRR **abs/2211.11682** (2022)