

# Unleashing the Potentials of Likelihood Composition for Multi-modal Language Models

Shitian Zhao<sup>1</sup> Renrui Zhang<sup>1,2</sup> Xu Luo<sup>1</sup> Yan Wang<sup>3</sup>  
Shanghang Zhang<sup>4</sup> Peng Gao<sup>1</sup>

<sup>1</sup>Shanghai AI Laboratory <sup>2</sup>CUHK <sup>3</sup>East China Normal University <sup>4</sup>Peking University

## Abstract

Model fusing has always been an important topic, especially in an era where large language models (LLM) and multi-modal language models (MLM) with different architectures, parameter sizes and training pipelines, are being created all the time. In this work, we propose a post-hoc framework, aiming at fusing heterogeneous models off-the-shell, which we call *likelihood composition*, and the basic idea is to compose multiple models' likelihood distribution when doing a multi-choice visual-question-answering task. Here the core concept, *likelihood*, is actually the log-probability of the candidate answer. In *likelihood composition*, we introduce some basic operations: *debias*, *highlight*, *majority-vote* and *ensemble*. By combining (composing) these basic elements, we get the mixed composition methods: *mix-composition*. Through conducting comprehensive experiments on 9 VQA datasets and 10 MLMs, we prove the effectiveness of *mix-composition* compared with simple *ensemble* or *majority-vote* methods. In this framework, people can propose new basic composition methods and combine them to get the new mixed composition methods. We hope our proposed *likelihood composition* can provide a new perspective of fusing heterogeneous models and inspire the exploration under this framework.<sup>1</sup>

## 1 Introduction

Recently numerous multi-modal language models are emerging, *e.g.*, LLaVA (Liu et al., 2023b,a, 2024b), MiniGPT4 (Chen et al., 2023a), BLIP-2 (Li et al., 2023a), Qwen-VL (Bai et al., 2023), InternVL (Chen et al., 2023b), and SPHINX (Lin et al., 2023; Gao et al., 2024), each characterized by different architectures, parameter sizes, training datasets, and pipelines. Consequently, these models exhibit varying strengths across different tasks

and domains. Some works (Ilharco et al., 2023; Wortsman et al., 2021, 2022) have demonstrated that fusing multiple models can enhance performance and generalizability across diverse domains. Thus, several model fusion techniques have been devised to leverage the complementary capabilities of these models.

Many works focus on getting a new model by inheriting the knowledge from multiple parent models. Some of them interpolate several models' weights to get the new model's weight, *e.g.*, WiSE-FT (Wortsman et al., 2021) and model soup (Wortsman et al., 2022). However, in this process, all the parent models and the new derived model need to have the same architecture and parameter sizes, *i.e.*, in most cases, the parent models are the fine-tuning versions of one pretrained model, leading to lack of diversity of these models. There are also some works focusing on distilling the knowledge from several different parent models (Wan et al., 2024a,b). However, the training computation cost makes it hard for researchers to combine parent models freely, *i.e.*, the computation cost of the distillation training process limits researchers to do lots of trial-and-error experiments to get a good parent models recipe.

Considering these issues, a promising line of work (Zhao et al., 2023; Li et al., 2024; Chuang et al., 2023; Wang et al., 2022) fuse different models via manipulating or composing their likelihood distributions<sup>2</sup>, with the advantage of being fully post-hoc, training-free and the same architecture of parent models is not necessary. The basic operation is to average all models' likelihood distribution of the candidate answers, called "ensemble". (Dietterich, 2000) Recently some works

<sup>2</sup>Presicely, "likelihood" actually refers the log-probability of the generated answer, following (Wang et al., 2022), the detailed computing method is shown in Sec.3. Likelihood distribution contains likelihood of multiple candidate answers, here we assume by default that we are discussing a multiple-choice visual question task.

<sup>1</sup>Code is released [zhaoshitian/Likelihood-Composition-Toolkit](https://github.com/zhaoshitian/Likelihood-Composition-Toolkit)

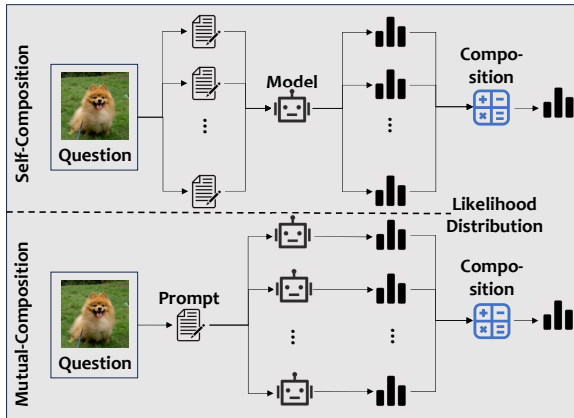


Figure 1: Two categories of likelihood composition: self-composition and mutual-composition.

also tried to combine the likelihood distributions from one model with different prompts as input (Zhao et al., 2023; O’Brien and Lewis, 2023; Leng et al., 2023; Zhang et al., 2023b). Behind these works, the core concept is to fuse different models via combining their likelihood distribution of the candidate answers, to boost the performance of downstream tasks, specifically multi-choice VQA tasks. Thus, we propose a framework named **likelihood composition**. Under this framework, we introduce some basic operations: *debias*, *highlight*, *ensemble* and *majority-vote*, some of which have been proposed in the previous research (Niu et al., 2021; Zhao et al., 2023; Wang et al., 2022). And these basic operations are classified into two classes: “self-composition” and “mutual-composition”. By composing these basic operations, we derive some new likelihood composition methods, e.g., ensemble-debias, ensemble-highlight, majority-debias and majority-highlight, which we call “mix-composition”.

To explore the likelihood composition’s effectiveness on fusing models and boosting the performance on downstream tasks, we conduct the experiments on LLaVA series and other 4 advanced multi-modal language models and 9 VQA datasets. Our experiment results reveal some interesting findings:

- (1) Self-composition can help model improve its performance on VQA tasks, especially for not well-developed models, e.g., *debias* can bring a +12.08% improvement for LLaVA-7B on MMVP.
- (2) Mix-composition performs better compared to mutual-composition with respect to boosting

the performance on VQA tasks, e.g., simply combining *debias* and *ensemble* can bring a +7.93% improvement on MMVP compared to vanilla ensemble method, a +6.93 improvement on MME compared to vanilla majority-vote.

- (3) When fusing models using likelihood composition, models’ quality is more important than models’ quantity, e.g., fusing LLaVA1.5-13B, LLaVA1.6-7B and LLaVA1.6-13B can make a better performance than fusing all models in LLaVA series.

## 2 Related Works

**Multi-modal Language Models** Based on the booming of large language models, lots of multi-modal language models have developed, e.g., LLaVA(Liu et al., 2023b), InternVL(Chen et al., 2023b), Qwen-VL(Bai et al., 2023), Yi-VL, SPHINX(Gao et al., 2024) and CogAgent(Hong et al., 2023b). These models have similar architectures and training pipelines. The composition of the model architecture is to use a MLP layer or Q-Former(Li et al., 2023a) to connect a pre-trained visual encoder, which could be the visual encoder in CLIP(Radford et al., 2021) or pretrained DINO(Caron et al., 2021), to a pretrained large language model, e.g., LLaMA(Touvron et al., 2023), Mistral and InternLM(Team, 2023). The training pipeline mainly follows the two stage design: first align the vision and language modality by training on massive image-text pair data, e.g., CC3M and CC12M(Changpinyo et al., 2021); then fine-tune the model using visual instruction data(Liu et al., 2023b). By doing so, these models show a good performance on multi-modal understanding and VQA tasks.

**Model Ensemble** To boost the performance on downstream tasks, a usual method is to ensemble multiple models. In this literature, many lines of research are developed: weight interpolation(Wortsman et al., 2022, 2021), model collaboration(Besta et al., 2023; Hong et al., 2023a; Shen et al., 2024; Yao et al., 2023) and distillation-based methods(Wan et al., 2024b,a).

**Decoding Methods for Language Models** Given a pretrained large language model, the decoding method also matters a lot. Some works focus on the decoding techniques of LLM, e.g., contrastive

decoding(O’Brien and Lewis, 2023), contrasting a pair of weak and strong LLM’s logits to improve the strong model’s generation quality; proxy tuning(Liu et al., 2024a, 2021), doing arithmetic among three pretrained LLMs to boost the generation ability.

### 3 Preliminary

Before introducing the detailed methodology of likelihood composition officially, it is necessary to make clear some core concepts that may be mentioned in the methodology part. So in this section, we give a clear description of the task formulation, which is the basic setting of our study, and an accurate definition of “likelihood”, the core concept used in our framework.

**Task Formulation** Considering most multi-modal tasks, *e.g.*, visual grounding and image retrieval can be formulated as a visual-question-answering task(Gao et al., 2024). In this paper, we exclusively investigate multi-choice visual question answering (VQA) tasks. The task formulation is as follows: Given a dataset  $\mathcal{D} = \{S_i\}$  comprising numerous VQA samples,  $S_i = (I_i, Q_i, C_i)$  consists of an image  $I_i$  and a question  $Q_i$ . The candidate answers for this image-question pair are represented as  $C_i$ , a list containing  $n$  candidate answers:  $C_i = [c_0^i, \dots, c_n^i]$ . And we need to input  $(I_i, Q_i, C_i)$  into MLM to predict the right answer’s option letter.

**Likelihood Calculation** For a VQA sample  $(I, Q, C)$ , during the normal forward process, both  $I$  and  $Q$  are input into the model.  $C$  is a list of choices, denoted as  $[c_0, c_1, \dots, c_n]$ , where  $n$  is the number of choices. The likelihood of candidate  $c_i$  is calculated as follows:

$$y_i(X) = \exp^{\frac{1}{K_i} \sum_{k=1}^{K_i} \log P(t_k|X, t_1, t_2, \dots, t_{k-1})}, \quad (1)$$

where  $y_i$  represents the likelihood value of  $c_i$  conditioned on  $X$ , the input to the model.  $P(t_k|X, t_1, t_2, \dots, t_{k-1})$  denotes the probability of generating the  $k$ th token  $t_k$  conditioned on  $X$  and the previously generated tokens  $t_1 \sim t_{k-1}$ .  $K_i$  represents the total number of tokens in  $c_i$ .

Once the likelihood of each choice is calculated, we denote the list containing all the choices’ likelihood values as  $\mathbf{Y}$ , upon which composition is

performed.<sup>34</sup>

## 4 Likelihood Composition

First, we organize the basic operations in the likelihood composition framework into two categories: *self-composition* and *mutual-composition*, as illustrated in Fig.1. Then we mix these basic elements, resulting in the *mix-composition*.

### 4.1 Self-Composition

The primary idea behind self-composition involves devising various prompt formats and preprocessing input samples using these prompts to generate different explicit inputs. Considering a VQA sample  $S = (I, Q, C)$ , we design  $m$  prompting methods:  $\text{Prompt}_1, \dots, \text{Prompt}_m$ . By applying these prompting methods to  $S$ , we obtain multiple results:  $X_1, \dots, X_m$ . Subsequently, inputting these results into the model yields corresponding likelihood distributions on  $C$ :  $\mathbf{Y}_1, \dots, \mathbf{Y}_m$ , on which we conduct the composition method. In this section, we introduce two self-composition methods: *Debias* and *Highlight*.

#### Debias

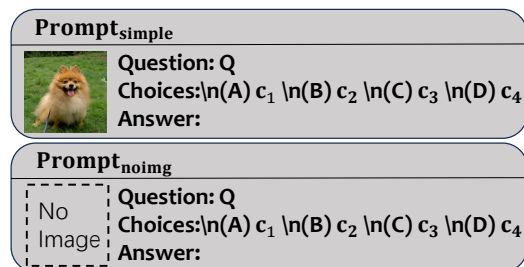


Figure 2: Prompt design of *debias*.

For multi-modal language models, language priors may be modeled during the training process on the multi-modal datasets, *i.e.*, the model will give the hallucinated answer ignoring the actual content of the provided image. (Agrawal et al., 2018; Li et al., 2023b; Niu et al., 2021) For example, when providing a picture containing a black color banana to an MLM and ask “What is the color of the banana?”, the model will say “Yellow”, which is the language prior bias in the training set. To model

<sup>3</sup>Here, after getting  $\mathbf{Y}$ , we actually perform softmax on it to make likelihood distributions from different models at the same scale. For convenience, we use  $\mathbf{Y}$  to illustrate the likelihood composition framework and the experiments in the following sections.

<sup>4</sup>It should be noted that in our practice, the input  $X$  also contains  $C$ , and each option’s likelihood is actually the corresponding option letter’s likelihood.

the language prior bias existing in the MLM, we only input the question into the model, inducing the model to give the most common answer, which reflects the language bias.

The prompt design is shown in Fig.2. Based on these prompt methods, we obtain corresponding likelihood values,  $\mathbf{Y}_{simple}$  and  $\mathbf{Y}_{noimg}$ . We then subtract  $\mathbf{Y}_{noimg}$  from  $\mathbf{Y}_{simple}$  with a coefficient  $\alpha$ , as formalized below:

$$\mathbf{Y} = (1 + \alpha)\mathbf{Y}_{simple} - \alpha\mathbf{Y}_{noimg} \quad (2)$$

Finally, among the likelihood values of the  $n$  choices in  $\mathbf{Y}$ , we select the option corresponding to the highest likelihood as the predicted answer.

### Highlight

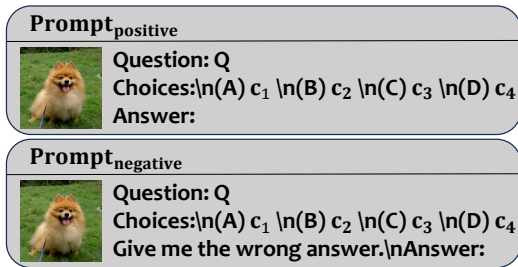


Figure 3: Prompt design of *highlight*.

Based on the idea of highlighting by using the opposite side, we introduce *Highlight*. First, we use the a positive instruction, *e.g.*, “Give me the right answer.”, to instruct the model to produce a high likelihood value of the right candidate. Then we use a negative instruction, *e.g.*, “Give me the wrong answer.”, to prone the model to the wrong candidates, producing a high likelihood value for the wrong answers. Finally, by contrasting these two likelihood distributions, we highlight the right answer.

We list the prompt design of highlight in Fig.3. It should be noted that in  $\text{Prompt}_{positive}$  we do not use a positive instruction, *e.g.*, “Give me the right answer.”, since usually MLM will give the right answer with no special note. Wrap the  $(I, Q, C)$  with  $\text{Prompt}_{negative}$ , we get the explicit input to the model. Then, the corresponding likelihood, termed as  $\mathbf{Y}_{negative}$  is produced conditioned on the  $\text{Prompt}_{negative}(I, Q, C)$ . To highlight the right answer among all the candidates, we subtract  $\mathbf{Y}_{negative}$  from  $\mathbf{Y}_{positive}$ :

$$\mathbf{Y}_{highlight} = (1 + \alpha)\mathbf{Y}_{positive} - \alpha\mathbf{Y}_{negative}, \quad (3)$$

on which the selection of the predicted answer is based.

## 4.2 Mutual-Composition

Except for composing the likelihood distribution produced by one model, we can also compose the likelihood distribution output from multiple different models with varying architectures, sizes and training pipelines, termed as “mutual-composition”.

### Ensemble

Based on the task formulation mentioned in Sec. 3, the most explicit composition method is to averaging all the provided likelihood distribution from different models, conditioned on the same input. Specifically, say there are  $N$  models:  $\{F_i | i = 1, \dots, N\}$ , and conditioned on the given sample,  $(I, Q, C)$ , we can get  $N$  likelihood distribution of the candidate answers from the  $N$  corresponding models, termed as  $\{\mathbf{Y}_i | i = 1, \dots, N\}$ . To ensemble them, we do the simplest averaging:

$$\mathbf{Y}_{ensemble} = \frac{1}{N} \sum_{i=1}^N \mathbf{Y}_i, \quad (4)$$

from which the predicted option is selected.

### Majority-vote

There have been some works focusing on ensemble models’ output, *e.g.*, CoT-SC(Wang et al., 2022). The basic operation used in these works is to do the majority-voting among the outputs despite being produced by one model or multiple heterogeneous models. And majority-vote actually can also be expressed using the likelihood composition language. Compared to ensemble mentioned above, majority-vote actually just adds a mask during the composition process:

$$\begin{cases} \mathbf{Y}_{majority-vote} = \frac{1}{N} \sum_{i=1}^N \mathbf{1} * [MASK]_i, & \text{Unweighted} \\ \mathbf{Y}_{majority-vote} = \frac{1}{N} \sum_{i=1}^N \mathbf{Y}_i * [MASK]_i, & \text{Weighted,} \end{cases} \quad (5)$$

where  $\mathbf{1}$  is an all ones vector having the same length with  $\mathbf{Y}_i$ .  $[MASK]_i$  is a 0-1 vector, where the element with the same index as that of the max element in  $\mathbf{Y}_i$  is 1, and the other elements are 0.

## 4.3 Mix the Self-Composition and Mutual-Composition

The basic idea of likelihood composition is to compose different likelihood distributions from either one model or multiple heterogeneous models. From this perspective, we can mix the two introduced composition methods mentioned in Sec. 4.1

and Sec. 4.2, and we call the mixed method: **mix-composition**.

In detail, say there are  $N$  models,  $\{F_i | i = 1, \dots, N\}$  we first apply the self-composition, either *debias* or *highlight*, to the likelihood distribution output by one model. Then, we use the *ensemble* method to fuse the  $N$  manipulated likelihood distribution of different models. The arithmetic formula is shown below:

$$\mathbf{Y}_{mix} = \frac{1}{N} \sum_{i=1}^N ((1 + \alpha)\mathbf{Y}_{simple,i} - \alpha\mathbf{Y}_{*,i}), \quad (6)$$

where  $*$  is either *noimg* or *negative*, the prompt method used in self-composition.

Similarly, when combining self-composition and majority-vote, the composition would be:

$$\mathbf{Y}_{mix} = \frac{1}{N} \sum_{i=1}^N ((1 + \alpha)\mathbf{Y}_{simple,i} - \alpha\mathbf{Y}_{*,i}) * [MASK]_i, \quad (7)$$

where  $*$  is either *noimg* or *negative*.

## 5 Experiments

To evaluate the efficiency of likelihood composition methods on improving MLM’s performance on VQA tasks, we conduct experiments on 10 advanced multi-modal language models and 9 VQA benchmarks.

### 5.1 Used Multi-modal Language Model

In our early experiments, we find the likelihood composition’s efficiency varied in models with different levels of capabilities and different training schemas. To show the likelihood composition’s general effectiveness, we select the LLaVA series, from LLaVA-7B to LLaVA1.6-13B, with similar architectures, training schemas and stepwise capability increase. Also, we select other 4 well-developed MLMs: Yi-VL, Qwen-VL, InternVL, and Internlm-Xcomposer, to see how likelihood composition performs on heterogeneous models.

- **LLaVA series**(Liu et al., 2023b,a) are multi-modal language models developed on pretrained large language models, *e.g.*, LLaMA(Touvron et al., 2023) and Vicuna(Chiang et al., 2023), and pretrained visual encoder, *e.g.*, vision encoder in pretrained CLIP(Radford et al., 2021). A simple linear layer or MLP layer is used for aligning the vision and language modalities. Models in this family are trained on vision-text pairwise data and visual instruction data. We select LLaVA-7B, LLaVA-13B, LLaVA1.5-7B, LLaVA1.5-13B, LLaVA1.6-7B and LLaVA1.6-13B with increased multi-modal ability in our experiments.

- **Other Heterogeneous MLMs** contains Yi-VL, Qwen-VL(Bai et al., 2023), InternVL(Chen et al., 2023b), and Internlm-Xcomposer(Zhang et al., 2023a), which are all trained on vision-language pairwise data and multi-modal instruction data. These models have good visual understanding and reasoning abilities.

### 5.2 Used Datasets

We include different types of VQA datasets in our experiments, to prove the likelihood composition’s generalizability with respect to data and tasks.

- **Comprehensive VQA benchmarks** include MME(Fu et al., 2023) and MMBench(Liu et al., 2023c), two comprehensive VQA benchmarks containing various tasks, from object existence to code reasoning. And the question format is multi-choice VQA, which is fitted for our task formulation, introduced in Sec. 3.
- **Diagnose VQA Benchmarks** contains VSR, POPE(Li et al., 2023b) and MMVP(Tong et al., 2024). POPE is a Yes-No format VQA benchmark aimed at diagnosing MLM’s object hallucination. MMVP is a dataset exploring the shortcomings of MLMs.
- **Reformed Academic VQA Datasets** include the split and reformed versions of VQAv2(Goyal et al., 2017), OKVQA(Marino et al., 2019), GQA(Hudson and Manning, 2019) and Vizwiz(Gurari et al., 2018) from ReForm-Eval(Li et al., 2023c), and we use a superscript “\*” to represent the reformed versions.

### 5.3 Main Results

We reported the results of applying likelihood composition on LLaVA series MLMs and 4 advanced MLMs with varying hyperparameters in Table.5,2 and Table.3 respectively. In Table.5, baseline is the MLM’s intrinsic performance, while in Table.2 and Table.3, baselines are *ensemble* and *majority-vote*, which we refer as *mutual-composition* in our framework.

**Results on LLaVA Series** As shown in Table.5, applied with self-composition methods mentioned in Sec. 4.1, LLaVA series’ performance on the 9 datasets consistently improved, *e.g.*, +12.08% for LLaVA-7B on MMVP, +4.39% for LLaVA-13B on MMBench, +6.96% for LLaVA1.5-7B on VSR, etc. Overall, for the early models in LLaVA family, *i.e.*, LLaVA-7B and LLaVA-13B, which is not well developed relatively, self-composition methods improve their performance significantly. Also, aggressive self-composition, *i.e.*, with  $\alpha = 1.0$  works better in most cases than that with  $\alpha = 0.1$ , for LLaVA-7B and LLaVA-13B.

For those well-developed models, *i.e.*, LLaVA1.5-7B, LLaVA1.5-13B, LLaVA1.6-7B and LLaVA1.6-13B, the improvement self-composition brings is not as significant as before. In more detail, for the best model, LLaVA1.6-13B, the

	$\alpha$	De bias	High light	MME	MMVP	MMBench	VSR	POPE	VQAv2*	Vizwiz*	GQA*	OKVQA*
7B				991.26	0.67	40.62	54.91	61.83	38.99	36.66	36.83	31.94
	1.0	✓	✓	<b>1069.56</b>	<b>12.75</b>	44.67	57.20	<b>76.76</b>	<b>42.07</b>	<b>38.52</b>	<b>38.19</b>	<b>32.54</b>
				973.94	4.70	<b>46.70</b>	51.47	70.82	38.53	32.02	35.24	31.15
7B	0.1	✓	✓	988.63	2.01	41.81	55.65	61.76	39.69	36.89	37.07	32.34
				987.84	4.03	42.86	<b>59.33</b>	63.35	39.97	36.43	36.20	31.94
13B				1106.00	4.70	41.58	61.62	55.72	37.87	31.79	38.58	28.77
	1.0	✓	✓	1124.50	8.72	38.50	<b>62.60</b>	<b>59.46</b>	38.20	<b>32.95</b>	37.07	29.17
				<b>1197.31</b>	<b>13.42</b>	40.16	55.32	54.64	34.10	25.06	33.09	25.00
13B	0.1	✓	✓	1090.73	8.72	<b>45.97</b>	61.54	57.00	<b>39.46</b>	34.11	<b>38.98</b>	<b>30.56</b>
				1114.30	8.72	41.88	60.31	58.83	36.99	31.32	36.04	27.78
v1.5-7B				1741.14	24.16	71.17	58.92	85.78	73.09	64.04	65.08	73.81
	1.0	✓	✓	1723.09	<b>25.50</b>	70.30	<b>65.88</b>	70.19	<b>73.60</b>	63.34	64.52	73.41
				<b>1804.74</b>	21.48	68.52	54.34	73.82	71.83	61.95	64.52	68.85
v1.5-7B	0.1	✓	✓	1747.96	24.83	<b>71.42</b>	60.80	<b>85.95</b>	73.13	<b>64.27</b>	<b>65.31</b>	<b>73.81</b>
				1756.19	22.82	71.05	58.02	73.43	73.13	64.04	65.08	72.22
v1.5-13B				1782.34	26.17	73.09	68.17	84.70	75.70	75.17	67.70	76.19
	1.0	✓	✓	<b>1833.70</b>	26.17	<b>73.25</b>	<b>73.90</b>	79.83	75.42	<b>75.64</b>	67.70	75.40
				1819.68	26.17	68.61	71.11	59.11	75.79	75.17	66.27	72.22
v1.5-13B	0.1	✓	✓	1789.38	<b>26.85</b>	73.22	69.89	<b>86.04</b>	75.75	75.41	67.70	75.79
				1779.75	25.50	72.97	70.29	60.56	<b>75.84</b>	75.14	<b>68.10</b>	<b>76.19</b>
v1.6-7B				1691.81	13.42	<b>71.30</b>	66.12	67.33	67.07	54.52	59.35	<b>72.62</b>
	1.0	✓	✓	<b>1765.97</b>	14.77	70.46	64.81	71.11	66.84	54.76	58.31	71.43
				1679.07	<b>17.45</b>	51.61	60.23	60.61	67.07	52.90	<b>59.51</b>	68.85
v1.6-7B	0.1	✓	✓	1711.07	13.42	71.14	<b>66.45</b>	<b>68.50</b>	67.16	54.99	59.11	72.42
				1703.37	14.09	70.41	65.96	72.42	<b>69.26</b>	<b>67.02</b>	59.19	72.42
v1.6-13B				1807.45	26.85	74.05	66.94	84.74	76.21	<b>81.67</b>	66.75	75.99
	1.0	✓	✓	1790.79	<b>28.86</b>	73.95	<b>69.89</b>	81.36	75.70	78.42	67.46	75.60
				1726.69	27.52	70.37	68.33	56.83	75.89	77.73	<b>68.18</b>	74.60
v1.6-13B	0.1	✓	✓	1800.68	28.19	<b>74.11</b>	67.92	<b>84.78</b>	<b>76.35</b>	80.97	67.06	<b>76.59</b>
				<b>1814.42</b>	28.86	74.05	68.90	59.47	76.21	81.21	66.91	76.19

Table 1: Self-composition methods bring a consistent improvement to all the LLaVA series models. (1) For *debias*, when  $\alpha$  is set to 1.0, it improves LLaVA-7B’s performance on all 9 datasets and LLaVA-13B’s performance on 7 datasets. When  $\alpha$  is set to 0.1, *debias* improves LLaVA1.5-7B and LLaVA1.5-13B’s performance on 9 and 8 datasets respectively; improves LLaVA1.6-7B and LLaVA1.6-13B’s performance on 5 and 7 datasets respectively. (2) For *highlight*, when setting  $\alpha$  to 0.1, it improves LLaVA1.6-13B, LLaVA1.6-7B and LLaVA-7B’s performance on 5 datasets.

improvement on MMVP is +2.01%, relatively small than that of LLaVA-7B: +12.08%. In some cases, self-composition will cause the performance drop, *e.g.*, the performance of LLaVA1.6-7B on MMBench dropped from 71.30% to 71.14%. Concerning the  $\alpha$ , the value of 0.1 works better.

In Table.2, we reported the results applying mutual-composition and mix-composition. For the mutual-composition, *i.e.*, vanilla ensemble and weighted majority-vote (likelihood as the weight), the performance is significantly higher than that of unweighted majority-vote, which is a mainstream model ensemble and collaboration method. For example, on MMVP, the performance of the vanilla ensemble method is higher than that of unweighted majority-vote by +12.08% and this number for weighted majority-vote is +9.4%. For mix-composition methods, which means mix the two self-composition methods: *debias* or *highlight* into the mutual-composition pipeline, we can see that after the mixing, the performance on most

VQA datasets will be improved further, *e.g.*, mixing *debias* and weighted majority-vote brings a +13.42% improvement on MMVP and +2.13% improvement on VSR.

In general, by conducting experiments on model families with increased abilities, we find:

- Self-composition with a high  $\alpha$  value works better for not well-developed models. While self-composition with a low  $\alpha$  value is suitable for advanced models. More analysis could be found in Sec.6.
- Mix-composition works better than mutual-composition when fusing different models.

**Results on 5 Advanced MLMs** We apply *mutual-composition* and *mix-composition* on 5 advanced MLMs: LLaVA, Yi-VL, Qwen-VL, InternVL and Internlm-Xcomposer. As shown in Table.3, *mutual-composition* brings significant improvement on most datasets, *e.g.*, +4.62% improvement on VQAv2\* and +3.36% improvement on MMVP.

	$\alpha$	De bias	High light	MME	MMVP	MMBench	VSR	POPE	VQAv2*	Vizwiz*	GQA*	OKVQA*	
Majority-vote		unweighted		1751.98	14.09	73.34	67.35	84.77	73.69	68.68	63.64	76.19	
		weighted		1826.62	23.49	74.48	69.64	<b>87.49</b>	77.80	74.48	69.21	79.17	
	1.0	✓	✓	1820.95	<b>36.91</b>	74.71	70.87	82.49	77.66	72.62	69.85	78.37	
				1773.69	24.16	72.72	59.00	61.83	77.66	<b>75.17</b>	68.66	78.77	
		✓	✓	1797.63	30.20	74.37	67.76	70.05	77.99	74.48	69.29	79.17	
				1833.01	<u>34.23</u>	74.87	<b>71.77</b>	85.35	78.08	72.85	70.01	78.17	
	0.5	✓	✓	1833.32	24.16	73.22	63.18	62.26	77.94	<u>74.94</u>	68.89	79.56	
				1797.60	28.19	74.39	69.56	76.53	77.85	74.25	69.93	78.97	
		✓	✓	<u>1836.29</u>	26.17	74.53	70.30	<u>87.30</u>	78.03	74.25	69.93	78.57	
				1840.52	23.49	74.34	68.99	62.90	78.13	73.55	69.05	<b>80.56</b>	
	0.1	✓	✓	1837.14	23.49	74.41	69.89	71.67	78.26	74.48	69.77	<u>79.96</u>	
				1837.33	26.17	74.66	70.13	82.72	<u>77.75</u>	72.85	70.80	78.57	
Ensemble	1.0	✓	✓	1813.03	31.54	74.78	70.87	79.79	77.80	72.85	70.49	78.37	
				1822.79	24.16	71.90	56.87	61.29	78.17	74.01	69.61	78.57	
				1832.44	27.52	73.79	66.12	61.82	78.26	73.78	69.69	78.17	
	0.5	✓	✓	1817.50	33.56	<b>74.94</b>	70.87	80.90	<b>78.36</b>	73.78	70.88	78.37	
				1816.52	24.16	73.25	62.77	61.63	77.66	73.32	69.69	78.57	
		✓	✓	1823.61	27.52	74.37	67.84	62.36	78.22	74.48	70.41	78.97	
				1839.75	28.86	74.82	70.79	82.45	77.80	73.55	71.12	78.37	
	0.1	✓	✓	1836.15	26.17	74.48	69.72	66.51	77.52	73.09	70.56	78.57	
				1834.62	26.17	<u>74.87</u>	70.54	80.25	77.80	73.32	70.64	78.37	
		$\alpha_d = 0.5$ $\alpha_h = 0.1$	✓	✓	1822.77	28.19	74.78	<u>70.95</u>	83.35	<u>78.26</u>	74.25	<b>71.20</b>	78.57

Table 2: Combining self-composition and mutual-composition can significantly further improve the performance over that of mutual-composition. In most cases, combining *debias* with mutual-composition can bring a further improvement. (1) When setting  $\alpha$  to 0.5, combining *debias* with *majority-vote* can bring improvement on 6 datasets. (2) When setting  $\alpha$  to 0.1, combining *debias* with *ensemble* can bring improvement on 7 datasets. (3) In the last line, we combine both *debias* and *highlight* and set the  $\alpha_d$  to 0.5,  $\alpha_h$  to 0.1.

And *mix-composition* brings further improvement, e.g., +9.08 on MME.

## 6 Additional Analysis

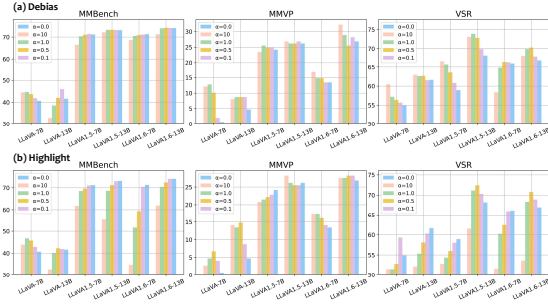


Figure 4: (a) Applying *debias* on LLaVA series models with  $\alpha$  ranging from 1.0 to 0.1. (b) Applying *highlight* on LLaVA series models with  $\alpha$  ranging from 1.0 to 0.1.

**Exploring  $\alpha$ 's Consequence on Self-Composition.** In the previous experiments,  $\alpha$ 's value shows different consequences on models with different levels of ability. **Generally, a high  $\alpha$ 's value works better for not well-developed models and a low  $\alpha$ 's value is suitable for advanced models.** So we conduct a more detailed analysis, ranging the  $\alpha$ 's value from 0.1 to 1.0, and visualize the results in Fig.4.

When setting  $\alpha = 0.0$ , we do not use the self-composition method, which is the baseline. In

subfigure (a), as we can see, *debias* works for all models on VSR. The highest results appear when  $\alpha$  is set low for LLaVA-7B, LLaVA-13B, LLaVA1.5-7B and LLaVA1.5-13B, while for LLaVA1.6-7B and LLaVA1.6-13B, the best performance appears when  $\alpha$  is set relatively low, which is 0.5. On MMBench and MMVP, *debias* works well for LLaVA-7B and LLaVA-13B. But for the other more advanced models, a high  $\alpha$  may bring damage to the performance and  $\alpha$  with low value may bring improvement.<sup>5</sup> In subfigure (b), *highlight* does not work so well as *debias*. But overall, *highlight* works better for LLaVA-7B and LLaVA-13B, which are two relatively weak models.

### Applying self-composition between different models.

To investigate how *debias* and *highlight* works between different models, e.g., subtract likelihood distribution derived by model A using Prompt<sub>noimg</sub> from the likelihood distribution derived by another model B using Prompt<sub>simple</sub>. In Fig.5, the x-axis represents model B and y-axis represents model A, the value is the difference

<sup>5</sup>On MMVP,  $\alpha$  with 1.0 works well for LLaVA1.6-7B and LLaVA1.6-13B, which are two relatively advanced model. But it should be noted that all models do not perform well on MMVP, i.e., on MMVP, all models are actually not "advanced".

	$\alpha$	De bias	MME	MMVP	VSR	POPE	VQAv2*	Vizwiz*	GQA*	OKVQA*
LLaVA			1807.45	26.85	66.94	84.74	76.21	81.67	66.75	75.99
Yi-VL			1977.21	35.57	59.90	81.50	76.96	72.39	72.39	78.57
Qwen-VL			1769.19	26.85	69.80	86.96	76.12	63.80	70.41	74.80
InternVL			1714.89	17.45	69.39	85.97	60.12	46.87	55.93	37.10
Internlm-Xcomposer			1896.67	28.86	77.58	87.47	63.95	56.84	57.84	51.79
Ensemble			2026.23	<b>38.93</b>	77.74	87.13	<b>81.58</b>	78.65	74.14	79.76
	1.0	✓	2021.66	32.89	76.35	86.02	81.06	74.71	74.46	<b>80.56</b>
	0.5	✓	2009.15	36.24	76.51	<b>87.87</b>	<b>81.58</b>	76.8	74.46	80.16
	0.1	✓	<b>2035.31</b>	35.57	77.66	87.63	81.53	78.65	<b>74.54</b>	80.16
	0.05	✓	2030.53	37.58	<b>77.58</b>	87.34	81.53	<b>79.12</b>	74.38	79.76

Table 3: Results of applying *mutual-composition* and *mix-composition* on LLaVA, Yi-VL, Qwen-VL, InternVL and Internlm-Xcomposer. As we can see, in most cases, mixing *debias* with *ensemble* can bring the further improvement.

between the performance of applying *debias* or *highlight* on model A and model B and the performance of model B.<sup>6</sup> Thus, a higher value in the heatmap means *debias* or *highlight* works well between model A and model B.

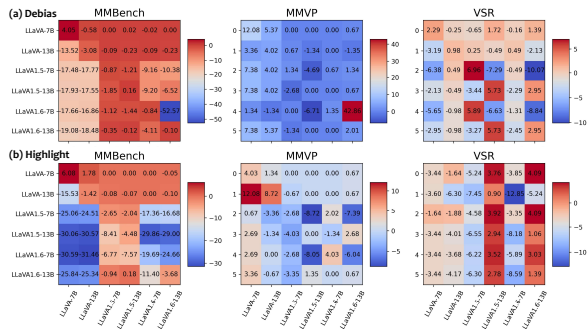


Figure 5: This figure illustrates the results of applying *debias* and *highlight* on model A and model B, which are different models. More details could be found in Sec.6. The x-axis represents model B and the y-axis represents model A.

As we can see, positive values and small negative values appear on the upper right part of the heatmap, *i.e.*, when model B is relatively better than model A, the method we mentioned above can work. Also, the highest value is always derived from two adjacent models on the coordinate axis, *e.g.*, in subfigure (a), the highest value is derived between LLaVA1.6-7B and LLaVA1.6-13B. In summary, **when applying *self-composition* method between two different models, if these two models are similar on the downstream task performance, it may work.**

**Adjusting the Number of Models to Ensemble.** When conducting experiments on LLaVA series, we apply *mutual-composition* and *mix-composition*

<sup>6</sup> $\alpha$  is set to 1.0 in all experiments in Fig.5.

on all the 6 models. However, fusing all the models may not be the best choice. Thus, we reduce the models to fuse and show the performance change in Fig.6. In this figure, each datapoint represents the result of applying *mutual-composition* or *mix-composition* to the model of the datapoint’s x-coordinate and all the models to its left.

As we can see, in most cases, the best performance does not appear at the far right but at “LLaVA1.5-13B” *i.e.*, only fusing LLaVA1.5-13B, LLaVA1.6-7B and LLaVA1.6-13B works better than fusing all 6 models. Thus, when applying *mutual-composition* and *mix-composition*, **models’ quality is more important than models’ quantity.**

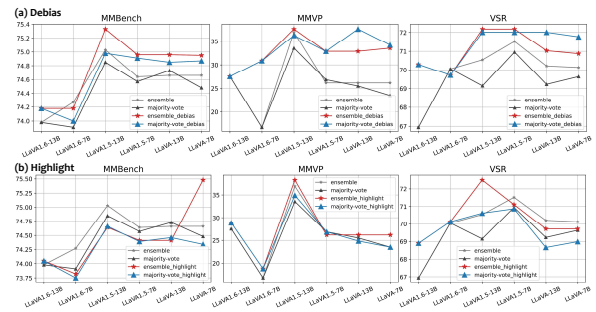


Figure 6: We change the number of models used for *mutual-composition* and *mix-composition*. At each position of the x-axis, we show the results of applying *mutual-composition* and *mix-composition* on the model at this position and the models left to it. It is obvious that models’ quality is more important than models’ quantity.

## 7 Conclusion

In this work, we propose “likelihood composition”, a framework unifying some operation in the model



fusing field. Based on this framework, we further propose “mix-composition”, mixing the “self-composition” and “mutual-composition”. In our experiments, we find “self-composition” can boost the MLM significantly on VQA tasks and “mix-composition” also bring significant improvement compared with “mutual-composition”.

## 8 Limitations

In our work, we did not consider closed source MLMs. For example, we can prompt closed-source MLMs to give their confidence on the list of answers and utilize these likelihood distributions in our proposed composition methods.

## References

- Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don’t just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4971–4980.
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660.
- Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568.
- Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023a. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.
- Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2023b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality.
- Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.
- Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.
- Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.
- Peng Gao, Renrui Zhang, Chris Liu, Longtian Qiu, Siyuan Huang, Weifeng Lin, Shitian Zhao, Shijie Geng, Ziyi Lin, Peng Jin, et al. 2024. Sphinx-x: Scaling data and parameters for a family of multi-modal large language models. *arXiv preprint arXiv:2402.05935*.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. 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.
- Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617.
- Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2023a. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.
- Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. 2023b. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*.
- Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

- Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. [Editing models with task arithmetic](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2023. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. *arXiv preprint arXiv:2311.16922*.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.
- Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024. More agents is all you need. *arXiv preprint arXiv:2402.05120*.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.
- Zejun Li, Ye Wang, Mengfei Du, Qingwen Liu, Bin-hao Wu, Jiwen Zhang, Chengxing Zhou, Zhihao Fan, Jie Fu, Jingjing Chen, et al. 2023c. Reform-eval: Evaluating large vision language models via unified re-formulation of task-oriented benchmarks. *arXiv preprint arXiv:2310.02569*.
- Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. 2023. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*.
- Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024a. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*.
- Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.
- Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. [Llava-next: Improved reasoning, ocr, and world knowledge](#).
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning.
- Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2023c. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.
- Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. 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.
- Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12700–12710.
- Sean O’Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2024. Hugging-gpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36.
- InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities.
- Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. [Eyes wide shut? exploring the visual shortcomings of multimodal llms](#). *Preprint*, arXiv:2401.06209.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024a. [Knowledge fusion of large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Fanqi Wan, Ziyi Yang, Longguang Zhong, Xiaojun Quan, Xinting Huang, and Wei Bi. 2024b. Fusechat: Knowledge fusion of chat models. *arXiv preprint arXiv:2402.16107*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022.

Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2021. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*. <https://arxiv.org/abs/2109.01903>.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models, may 2023. *arXiv preprint arXiv:2305.10601*, 14.

Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al. 2023a. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*.

Yue Zhang, Leyang Cui, Wei Bi, and Shuming Shi. 2023b. Alleviating hallucinations of large language models through induced hallucinations. *arXiv preprint arXiv:2312.15710*.

Shitian Zhao, Zhuowan Li, Yadong Lu, Alan Yuille, and Yan Wang. 2023. Causal-cog: A causal-effect look at context generation for boosting multi-modal language models. *arXiv preprint arXiv:2312.06685*.

## A Appendix

### A.1 Answer with no images.

In our proposed *debias* method, we make the model to produce likelihood distribution conditioned only on the question and choices, with no image provided. We can select the predicted answer based this likelihood distribution in the absence of the image. The results is shown in Fig.7.

We find that the performance of the model with no image provided increases with the increases of the performance conditioned on the image, which is so interesting. **It seems that with the model’s multi-modal understanding ability increasing, the model’s ability of “guessing correctly” increases also.**

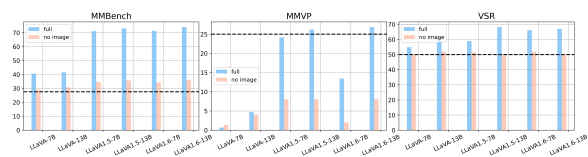


Figure 7: We investigate models’ performance when not providing the image. In the figure, the black dashed line represents the random choice score, “full” represents the normal case and “no image” represents not inputting the image to the model when doing VQA tasks. What’s interesting is that models with higher multi-modal understanding ability “guess” better when not inputting the image.

### A.2 Statistics of Datasets

In Table 4, the statistics of each benchmark, including version and number of samples, are listed.

Benchmark	Version	Number of Samples
MME	-	2373
MMBench	dev	4377
MMVP	-	300
POPE	Popular,Random,Adversarial	8910
VSR	-	1222
OKVQA	ReForm-Eval(Li et al., 2023c)	504
VQAv2	ReForm-Eval	2144
Vizwiz	ReForm-Eval	431
GQA	ReForm-Eval	1257

Table 4: Statistics of each benchmark.

### A.3 Full Results of Debias and Highlight with Different $\alpha$ ’s Values

In Table.5, we show the full results of applying *debias* and *highlight* on LLaVA (Liu et al., 2023b,a, 2024b) series with different  $\alpha$ ’s values.

	$\alpha$	De bias	High light	MME	MMVP	MMBench	VSR	POPE	VQAv2*	Vizviz*	GQA*	OKVQA*
7B				991.26	0.67	40.62	54.91	61.83	38.99	36.66	36.83	31.94
	10	✓		<b>1204.57</b>	12.08	44.46	<b>60.47</b>	67.85	41.14	37.82	<b>39.46</b>	31.15
			✓	980.52	2.68	43.82	51.47	68.90	36.38	27.75	34.37	28.77
	1.0	✓		1069.56	<b>12.75</b>	44.67	57.20	<b>76.76</b>	<b>42.07</b>	38.52	38.19	<b>32.54</b>
			✓	973.94	4.70	<b>46.70</b>	51.47	70.82	38.53	32.02	35.24	31.15
	0.5	✓		1036.94	10.07	43.71	56.38	70.50	40.67	<b>38.75</b>	38.58	31.94
✓			971.97	6.71	45.72	52.78	69.72	39.23	35.73	35.08	31.35	
0.1	✓		988.63	2.01	41.81	55.65	61.76	39.69	36.89	37.07	32.34	
		✓	987.84	4.03	42.86	59.33	63.35	39.97	36.43	36.20	31.94	
13B				1106.00	4.70	41.58	61.62	55.72	37.87	31.79	38.58	28.77
	10	✓		1190.97	8.05	32.81	62.93	<b>62.13</b>	37.22	30.39	35.24	26.79
			✓	<b>1248.54</b>	14.09	32.60	52.13	53.50	33.3	22.04	33.09	23.41
	1.0	✓		1124.50	8.72	38.50	62.60	59.46	38.20	32.95	37.07	29.17
			✓	1197.31	13.42	40.16	55.32	54.64	34.10	25.06	33.09	25.00
	0.5	✓		1099.25	8.72	41.95	<b>62.68</b>	58.59	<b>39.51</b>	<b>35.50</b>	36.75	<b>30.75</b>
✓			1156.91	<b>14.77</b>	42.22	58.10	57.28	34.93	27.38	34.37	25.40	
0.1	✓		1090.73	8.72	<b>45.97</b>	61.54	57.00	39.46	34.11	<b>38.98</b>	30.56	
		✓	1114.30	8.72	41.88	60.31	58.83	36.99	31.32	36.04	27.78	
v1.5-7B				1741.14	24.16	71.17	58.92	85.78	73.09	64.04	65.08	73.81
	10	✓		1626.79	23.49	66.51	<b>66.69</b>	68.56	<b>73.88</b>	62.65	63.33	72.22
			✓	1674.43	20.81	61.78	52.78	74.33	67.35	56.84	62.29	61.31
	1.0	✓		1723.09	<b>25.50</b>	70.30	65.88	70.19	73.60	63.34	64.52	73.41
			✓	<b>1804.74</b>	21.48	68.52	54.34	73.82	71.83	61.95	64.52	68.85
	0.5	✓		1736.78	24.83	70.98	63.58	77.18	73.69	63.57	64.92	73.61
✓			1802.38	22.15	69.61	55.97	73.84	72.48	62.41	65.23	70.04	
0.1	✓		1747.96	24.83	<b>71.42</b>	60.80	<b>85.95</b>	73.13	<b>64.27</b>	<b>65.31</b>	<b>73.81</b>	
		✓	1756.19	22.82	71.05	58.02	73.43	73.13	64.04	65.08	72.22	
v1.5-13B				1782.34	26.17	73.09	68.17	84.70	75.70	75.17	67.70	76.19
	10	✓		1796.79	26.85	72.17	73.08	71.06	75.33	73.78	67.14	74.60
			✓	1740.95	<b>28.19</b>	55.36	61.54	58.88	70.85	73.32	65.31	60.71
	1.0	✓		<b>1833.70</b>	26.17	73.25	<b>73.90</b>	79.83	75.42	<b>75.64</b>	67.70	75.40
			✓	1819.68	26.17	68.61	71.11	59.11	75.79	75.17	66.27	72.22
	0.5	✓		1805.9	26.17	<b>73.27</b>	72.83	84.55	75.51	75.41	67.86	75.60
✓			1791.54	25.50	71.21	72.42	59.27	76.31	75.64	67.14	73.21	
0.1	✓		1789.38	26.85	73.22	69.89	<b>86.04</b>	75.75	75.41	67.70	75.79	
		✓	1779.75	25.50	72.97	70.29	60.56	<b>75.84</b>	75.14	<b>68.10</b>	<b>76.19</b>	
v1.6-7B				1691.81	13.42	<b>71.30</b>	66.12	67.33	67.07	54.52	59.35	<b>72.62</b>
	10	✓		1653.97	16.78	68.59	58.35	72.38	66.09	54.76	57.84	70.83
			✓	1679.73	17.45	34.73	51.55	61.17	50.00	29.00	52.11	58.33
	1.0	✓		<b>1765.97</b>	14.77	70.46	64.81	71.11	66.84	54.76	58.31	71.43
			✓	1679.07	<b>17.45</b>	51.61	60.23	60.61	67.07	52.90	<b>59.51</b>	68.85
	0.5	✓		1754.84	14.77	70.92	<b>66.53</b>	70.54	67.02	54.99	58.55	72.22
✓			1692.89	16.11	59.33	62.44	60.57	67.07	52.90	59.51	71.23	
0.1	✓		1711.07	13.42	71.14	66.45	68.50	67.16	54.99	59.11	72.42	
		✓	1703.37	14.09	70.41	65.96	<b>72.42</b>	<b>69.26</b>	<b>67.02</b>	59.19	72.42	
v1.6-13B				1807.45	26.85	74.05	66.94	84.74	76.21	<b>81.67</b>	66.75	75.99
	10	✓		1803.29	<b>32.21</b>	71.30	68.09	76.65	73.65	71.00	67.14	73.02
			✓	1723.66	27.52	61.96	53.60	56.70	71.88	73.55	<b>68.26</b>	69.64
	1.0	✓		1790.79	28.86	73.95	69.89	81.36	75.70	78.42	67.46	75.60
			✓	1726.69	27.52	70.37	68.33	56.83	75.89	77.73	68.18	74.60
	0.5	✓		1787.40	25.50	<b>74.18</b>	70.29	83.79	76.07	80.04	67.46	76.59
✓			1746.43	28.19	72.38	<b>70.79</b>	57.03	<b>76.49</b>	79.81	67.70	74.80	
0.1	✓		1800.68	28.19	74.11	67.92	<b>84.78</b>	76.35	80.97	67.06	<b>76.59</b>	
		✓	<b>1814.42</b>	28.86	74.05	68.90	59.47	76.21	81.21	66.91	76.19	

Table 5: Full results of *debias* and *highlight*.