



# Vary: Scaling up the Vision Vocabulary for Large Vision-Language Models

Haoran Wei<sup>1\*</sup>, Lingyu Kong<sup>2\*</sup>, Jinyue Chen<sup>2</sup>, Liang Zhao<sup>1</sup>, Zheng Ge<sup>1†</sup>,  
Jinrong Yang<sup>3</sup>, Jianjian Sun<sup>1</sup>, Chunrui Han<sup>1</sup>, Xiangyu Zhang<sup>1</sup>

<sup>1</sup>MEGVII Technology <sup>2</sup>University of Chinese Academy of Sciences

<sup>3</sup>Huazhong University of Science and Technology

<https://varybase.github.io/>

## Abstract

Modern Large Vision-Language Models (LVLMs) enjoy the same vision vocabulary – CLIP, which can cover most common vision tasks. However, for some special vision task that needs dense and fine-grained vision perception, *e.g.*, document-level OCR or chart understanding, especially in non-English scenarios, the CLIP-style vocabulary may encounter low efficiency in tokenizing the vision knowledge and even suffer out-of-vocabulary problem. Accordingly, we propose **Vary**, an efficient and effective method to scale up the Vision vocabulary of LVLMs. The procedures of Vary are naturally divided into two folds: the generation and integration of a new vision vocabulary. In the first phase, we devise a vocabulary network along with a tiny decoder-only transformer to produce the desired vocabulary via autoregression. In the next, we scale up the vanilla vision vocabulary by merging the new one with the original one (CLIP), enabling the LVLMs can quickly garner new features. Compared to the popular BLIP-2, MiniGPT4, and LLaVA, Vary can maintain its vanilla capabilities while enjoying more excellent fine-grained perception and understanding ability. Specifically, Vary is competent in new document parsing features (OCR or markdown conversion) while achieving 78.2% ANLS in DocVQA and 36.2% in MMVet. Our code will be publicly available on the homepage.

## 1 Introduction

Recently, research into vision dialogue robots [1, 19, 25, 33, 55] has been gaining significant traction. These human-like models, mainly relying on two components (large language models (LLMs) [7, 32, 35, 42, 53] and vision vocabulary networks), can not only converse based on user’s input image but also perform well on simple downstream tasks, such as VQA [22, 39], Image caption [43], OCR [30], and so on. Hence, it is undeniable that large vision-language models (LVLMs) are driving the AI community towards the direction of artificial general intelligence (AGI).

Popular GPT-4 [32]-like LVLMs, *e.g.*, BLIP2 [19], MiniGPT4 [55], LLaVA [25], Qwen-VL [4], and *etc.* [12, 50, 54] enjoy a stunning performance in multiple aspects with their own programming paradigm: Based on an LLM [36, 53], BLIP-2 proposes the Q-former, a BERT [11] like network as a vision input embedding layer, aiming to align the image tokens to a text special. Inherited the structure of BLIP-2, MiniGPT-4 introduces 3500 high-quality image-text pairs as self-supervised fine-tuning (SFT) data, allowing it can “talk” like GPT-4. Unlike BLIP-2, LLaVA utilizes a linear layer as the vision embedding layer, which is similar with the text input embedding layer in the text tokenizer, ensuring the consistency in the structure of image and text branches. For Qwen-VL, it

\*Equal contribution

†Project leader

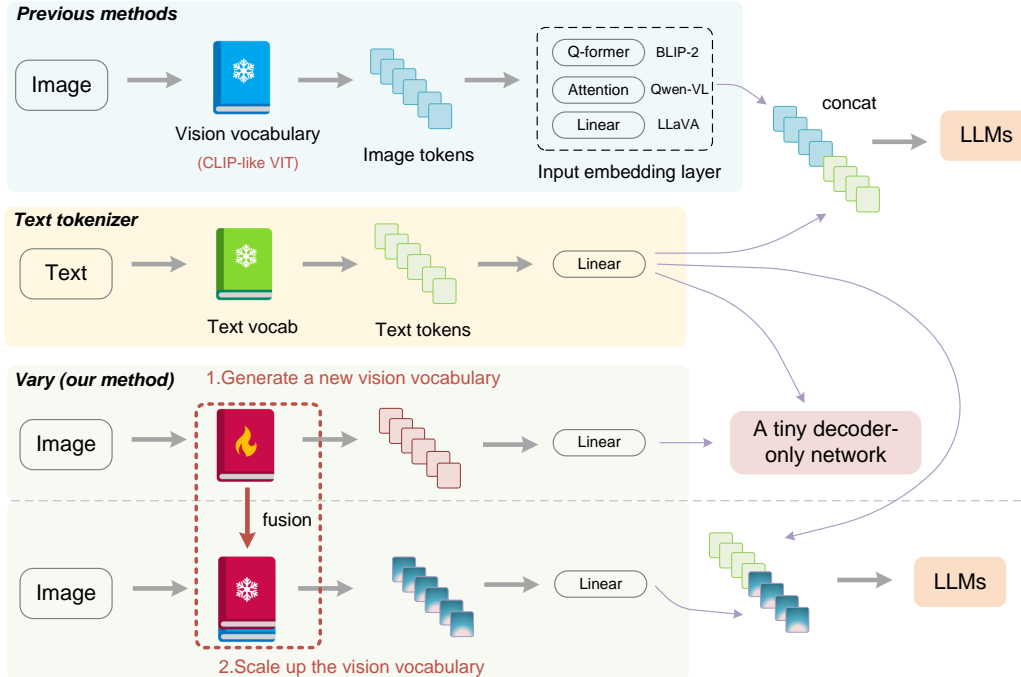


Figure 1: Previous method vs. Vary: Unlike other models that use a ready-made vision vocabulary, the processes of Vary can be divided into two stages: the generation and fusion of vision vocabulary. In the first stage, we use a “vocabulary network” along with a tiny decoder-only network to produce a powerful new vision vocabulary via auto-regression. In the second stage, we fuse the vision vocabulary with the original one to provide new features for the LVLMs efficiently.

utilizes a cross-attention layer to sample and align the image tokens, making the model can accept larger input resolution. Although the above LVLMs’ vision input embedding networks are variable (e.g., MLP, Qformer, Perceiver [1]), their vision vocabulary is almost identical (a CLIP-based [34] VIT) which we argue maybe a bottle-neck.

It is recognized that CLIP-VIT is a tremendous general vision vocabulary, which is trained via contrastive learning upon more than 400M [37] image-text pairs, covering most natural images and vision tasks. However, for some special scenarios, e.g., high-resolution perception, Non-English OCR, Document/Chart understanding, and so on, the CLIP-VIT may regard them as a “foreign language”, leading to inefficient tokenizing, i.e., difficulty in encoding all vision information into a fixed number (usually 256) of tokens. Although mPlug-Owl [49] and Qwen-VL alleviate the above issues by unfreeze its vision vocabulary network (a CLIP-L or CLIP-G), we argue that such manner may not be reasonable due to three aspects: 1) it may overwrite the knowledge of the original vocabulary; 2) the training efficiency of updating a vision vocabulary upon a relative large LLM (7B) is low; 3) it can not allow the vision vocabulary network to “see” an image multiple times (train a dataset with multiple epochs) due to the strong memory ability of LLMs. Therefore, a natural question is: *Is there a strategy that can simplify and effectively intensify the visual vocabulary?*

In this paper, we propose Vary, an efficient and user-friendly approach, to answer the above question. Vary is inspired by the text vocabulary expansion manner in vanilla LLMs [8], i.e., when transferring an English LLM to another foreign language, such as Chinese, it’s necessary to expand the text vocabulary to lift the encoding efficiency and model performance under the new language. Intuitively, for the vision branch, if we feed the “foreign language” image to the model, we also need to scale up the vision vocabulary. In Vary, the process of vocabulary scaling up can be divided into two steps: 1) generate a new vision vocabulary that can make up the old one (CLIP); 2) integrate the new and old vocabularies. As shown in Figure 1, we build a small-size pipeline which is consisting of a vocabulary network and a tiny decoder-only transformer in the first step to train the vocabulary model via predicting the next token. It is worth noting that the autoregressive-based process of generating a

vocabulary is perhaps more suitable for dense perception tasks than that based on contrastive learning like CLIP. On the one hand, the next-token way can allow the vision vocabulary to compress longer texts. On the other hand, the data formats that can be used in this manner are more diverse, such as VQA [5, 29] data with prompt. After preparing the new vision vocabulary, we add it to the vanilla LVLMs to introduce new features. In this process, we freeze both the new and old vocabularies networks to avoid the visual knowledge being overwritten.

Afterward scaling up the vision vocabulary, our LVLM can achieve more fine-grained vision perception, such as document-level Chinese/English OCR, book image to markdown or  $\text{BTeX}$ , Chinese/English chart understanding, and so on, while ensuring its original capabilities (conversation, VQA, caption, *etc.*). Besides, we provide methods for producing synthetic data and validate its importance in document/chart understanding. More importantly, Vary is a useful strategy to strengthen the visual vocabulary of LVLMs, which can be utilized at arbitrary downstream visual tasks that CLIP is not good at. In addition to the document and chart parsing mentioned in this paper, we believe that Vary still enjoys more fine-grained tasks and we appeal to researchers to rethink the design ideas of LVLMs from the perspective of visual vocabulary construction.

## 2 Related Works

### 2.1 Large Language Models

Over the past year, significant attention has been drawn to large language models (LLMs) in the fields of both natural language processing (NLP) and computer vision (CV). This heightened attention stems from LLMs’ outstanding performance in diverse aspects, especially the powerful world knowledge base and universal capabilities. Current LLMs enjoy a unified transformer architecture which is exemplified by BERT [11], GPT-2 [35], T5 [36], *etc.* Subsequently, researchers have uncovered the concept of an "emergent ability" [45] in LLMs. This implies that as language model sizes reach a certain threshold, there may be a qualitative leap in their capabilities. Furthermore, InstructGPT [33] and ChatGPT [31] find that Reinforcement Learning with Human Feedback (RLHF) [9] can further lift the performance of the "talk robot". Motivated by the tremendous success of the GPT series, a multitude of other open-source LLMs have emerged, including OPT [53], LLaMA [42], GLM [52], and so on. Building upon these openly available LLMs, numerous tailored fine-tuned models have been introduced to develop LLMs for diverse applications, especially LLaMA-driven models, *e.g.*, Alphaca [40], Vicuna [8], which have become the de-facto component for a Large Vision-Language Model (LVLM).

### 2.2 LLM-based Large Vision-Language Models

LLM’s robust zero-shot capabilities and logical reasoning make it play the central controller role within an LVLM. There are two primary pipeline styles: plugin-based and end-to-end model. Plugin-based methods [21, 38, 46–48] typically regard LLMs as an agent to invoke various plugins from other foundational or expert models, executing specific functions in response to human instructions. While such methods offer versatility, they have limitations in terms of plugin invocation efficiency and performance. Conversely, end-to-end LVLMs usually rely on a single large multimodal model to facilitate interactions. Following this approach, Flamingo [1] introduces a gated cross-attention mechanism trained on billions of image-text pairs to align vision and language modalities, demonstrating strong performance in few-shot learning. BLIP-2 [19] introduces Q-Former to enhance the alignment of visual features with the language space. More recently, LLaVA [25] proposes using a simple linear layer to replace Q-Former and designed a two-stage instruction-tuning procedure.

Despite the remarkable performance of existing methods, they are confined to the same and limited vision vocabulary – CLIP-VIT [34]. For an LVLM, CLIP-VIT is a tremendous general vision vocabulary that is trained via contrastive learning upon million-level image-texts pairs, which can cover most nature images and vision tasks, *e.g.*, VQA, Caption, Easy English OCR. However, some images under special scenarios, *e.g.*, high-resolution image, Non-English OCR, Document/Chart understanding, and so on, will still be regarded as a "foreign language" by CLIP-VIT, leading to vision out-of-vocabulary problem, which will in turn become a bottleneck for LVLMs.

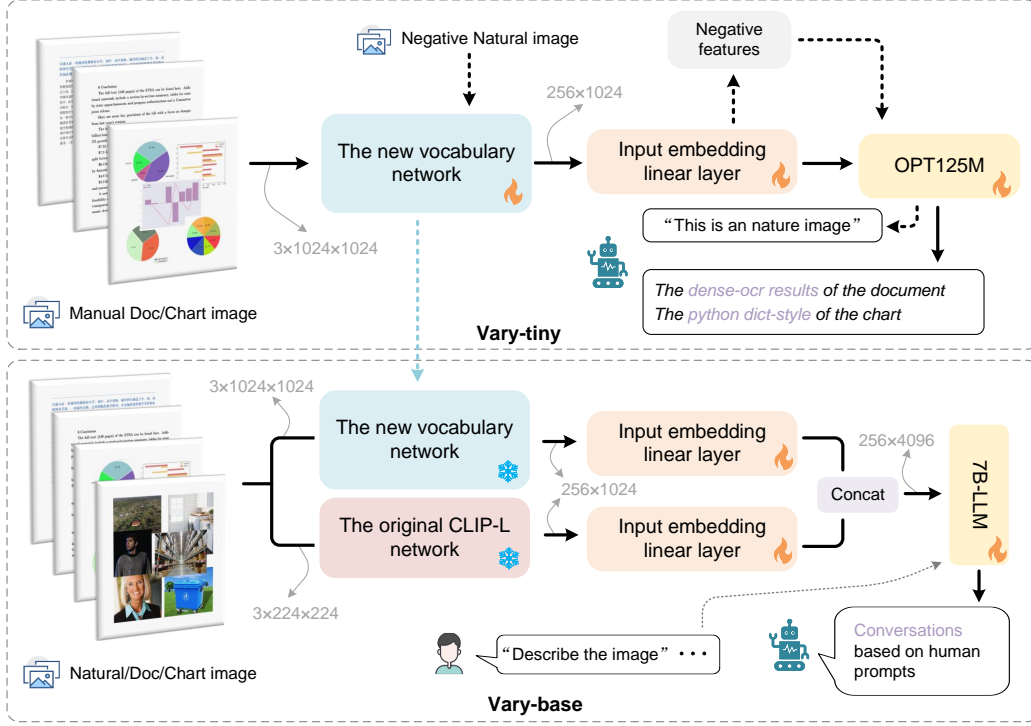


Figure 2: Overview of the Vary. There are two types of Vary form: Vary-tiny and Vary-base. Vary-tiny is mainly focused on generating a new vision vocabulary while Vary-base is our new LVLM aiming to handle various visual tasks based on the new vision vocabulary.

### 3 Method

#### 3.1 Architecture

Vary enjoys two conformations: Vary-tiny and Vary-base, as shown in Figure 2. We devise the Vary-tiny to “write” a new vision vocabulary and the Vary-base to make use of the new vocabulary. Specifically, Vary-tiny is mainly composed of a vocabulary network and a tiny OPT-125M [53]. Between the two modules, we add a linear layer to align the channel dimensions. There is no text input branch in Vary-tiny due to it is a primary focus on fine-grained perception. We hope the new vision vocabulary network can excel in processing artificial images, *i.e.*, documents, and charts, to compensate for CLIP’s shortcomings. At the same time, we also expect that it will not be a noise for CLIP when tokenizing natural images. Accordingly, during generating, we feed the manual document and chart data as positive samples and natural images as negatives to train Vary-tiny. After completing the above process, we extract the vocabulary network and add it to a large model to build the Vary-base. As shown in the lower half of Figure 2, the new and old vocabulary networks enjoy independent input embedding layers and are integrated before the LLM. In such a stage, we freeze both weights of new and old vision vocabulary networks and unfreeze the weights of other modules.

#### 3.2 Towards Generating a New Vision Vocabulary

##### 3.2.1 The new vocabulary network

We use the SAM [15] pretrained ViTDet [20] image encoder (base scale) as the main part of the new vocabulary network of Vary. Due to the input resolution of the SAM-base is  $(1024 \times 1024)$  while the output stride is 16, the feature shape of the last layer is  $(64 \times 64 \times 256)$  for  $H \times W \times C$  that can not be aligned to the output of CLIP-L ( $256 \times 1024$  for  $N \times C$ ). Hence, we add two convolution layers, which we found is a good token merging unit, behind the last layer of the SAM initialized network, as shown in Figure 3. The first convolution layer possesses a kernel size of 3, aiming to transfer the

feature shape to  $32 \times 32 \times 512$ . The setting of the second conv layer is the same as the first one, which can further convert the output shape to  $16 \times 16 \times 1024$ . After that, we flattened the output feature to  $256 \times 1024$  to align the image token shape of CLIP-ViT.

### 3.2.2 Data engine in the generating phrase

**Documnet data.** We select the high-resolution document image-text pairs as the main positive dataset used for the new vision vocabulary pre-train due to the dense OCR can effectively validate the fine-grained image perception ability of the model. To our knowledge, there is no publicly available dataset of English and Chinese documents, so we create our own. We first collect pdf-style documents from open-access articles on arXiv and CC-MAIN-2021-31-PDF-UNTRUNCATED for the English part and collect from e-books on the Internet for the Chinese part. Then we use *fitz* of PyMuPDF to extract the text information in each pdf page and convert each page into a PNG image via *pdf2image* at the same time. During this process, we construct 1M Chinese and 1M English document image-text pairs for training.

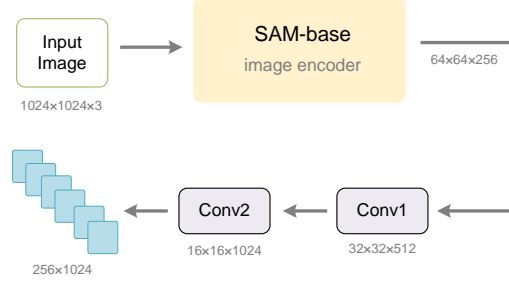


Figure 3: The structure of new vision vocabulary network. We add two convolution layers to convert the output to be similar with CLIP.

**Chart data.** We find current LVLMs are not good at chart understanding, especially Chinese charts, so we choose it as another main knowledge that needs to be “written” into the new vocabulary. For chart image-text pair, we all follow the rendering way. We select both the *matplotlib* and *pyecharts* as the rendering tools. For matplotlib-style chart, we built 250k in both Chinese and English. While for pyecharts, we build 500k for both Chinese and English. Besides, we convert the text ground truth of each chart to a python-dict form. The texts used in the chart, *e.g.*, title, x-axis, and y-axis, are randomly selected from the Natural Language Processing (NLP) corpus downloaded from the Internet.

**Negative natural image.** For natural image data that CLIP-ViT is good at, we need to ensure that the newly introduced vocabulary does not cause noise. Consequently, we construct negative natural image-text pairs to enable the new vocabulary network to encode correctly when seeing natural images. We extract 120k images in the COCO [22] dataset with each image corresponding to a text. The text part is randomly selected from follows sentences: "It's an image of nature"; "Here's a nature picture"; "It's a nature photo"; "This is a natural image"; "That's a shot from nature".

### 3.2.3 Input format

We train all parameters of the Vary-tiny with image-text pairs by autoregression. The input format follows popular LVLMs [13], *i.e.*, the image tokens are packed with text tokens in the form of a prefix. Specifically, we use two special tokens "<img>" and "</img>" to indicate the position of the image tokens as the input of an interpolated OPT-125M (4096 tokens). During training, the output of Vary-tiny is only text, and "</s>" is regarded as the *eos* token.

## 3.3 Towards Scaling Up the Vision Vocabulary

### 3.3.1 The structure of Vary-base

After completing the training of the vocabulary network, we introduce it to our LVLM – Vary-base. Specifically, we parallelize the new vision vocabulary with the original CLIP-ViT. Both two vision vocabularies enjoy an individual input embedding layer, *i.e.*, a simple linear. As shown in Figure 2, the input channel of the linear is 1024 and the output is 2048, ensuring the channel of image tokens after concatenating is 4096, which exactly aligns the input of LLM (Qwen-7B [3] or Vicuna-7B [8]).

### 3.3.2 Data engine in the scaling up phrase

**$\text{\LaTeX}$  rendering document.** Except for the collecting document data in Section 3.2.2, we also need data to enjoy some format, *e.g.*, supporting formula, and table. To this end, we create document

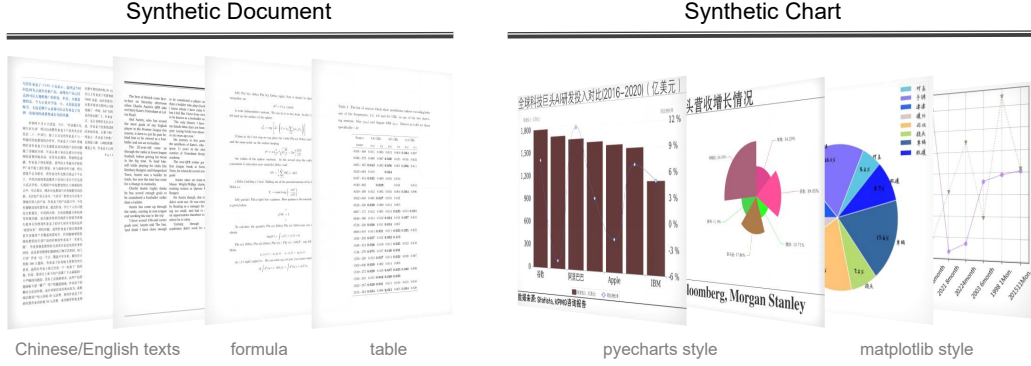


Figure 4: Visualization of synthetic data. We use *pdflatex* to render documents and utilize *pyecharts/matplotlib* to render charts. Document data obtains Chinese/English texts, formulas, and tables. Chart data includes Chinese/English bar, line, pie, and composite styles.

data through *LaTeX* rendering. Firstly, we collected some *.tex* source files on arxiv, and then extracted tables, mathematical formulas, and plain texts using regular expressions. Finally, we re-render these contents with the new template we prepared by *pdflatex*. We collect 10+ templates to perform batch rendering. Besides, we transfer the text ground truth of each document page to a *mathpix* markdown style to unify the format. By this construction process, we acquired 0.5 million English pages and 0.4 million Chinese pages. Some samples are shown in Figure 4.

**Semantic association chart rendering.** In Section 3.2.2, we batch render chart data to train the new vocabulary network. However, the texts (title, x-axis values, and y-axis values) in those rendered charts suffer low correlation because they are randomly generated. This issue is not a problem in the vocabulary-generating process as we only hope that the new vocabulary can efficiently compress visual information. However, in the training stage of the Vary-base, due to unfreezing the LLM, we hope to use higher quality (strongly correlated content) data for training. Therefore, we use GPT-4 [32] to generate some charts using relevant corpus and then we utilize the high-quality corpus to addition render 200k chart data for the Vary-base training.

**General data.** The processes of training Vary-base follows popular LVLMs, *e.g.*, LLaVA [25], including the pretrain and SFT phases. Different from the LLaVA, we freeze all the vocabulary networks and unfreeze both the input embedding layer and LLM, which is more like the pretrain setting of a pure LLM. We use natural image-text pair data to introduce the general concepts to the Vary-base. The image-text pairs are randomly extracted from LAION-COCO [37] with the amount of 4 million. In the SFT stage, we use the LLaVA-80k or LLaVA-CC665k [24] along with the train set of DocVQA [29] and ChartQA [28] as the fine-tuning dataset.

### 3.3.3 Conversation format

When we use the Vicuna-7B as our LLM, the conversation format follows the Vicuna v1 [8], *i.e.*, USER: <img><image></img> "texts input" ASSISTANT: "texts output" </s>. Due to the low efficiency in the text vocabulary of Vicuna to process Chinese, we choose Qwen-7B [2] as the LLM for Chinese processing. When we use the Qwen-7B, we design the conversation style following the LLaVA-MPT [25, 41], which can be described as: <lim\_start>user: <img><image></img> "texts input" <lim\_end> <lim\_start>assistant: "texts output" <lim\_end>.

## 4 Experiments

### 4.1 Datasets and Evaluation Metrics

We evaluate the proposed Vary on multiple datasets, including 1) a document-level OCR test set we created to explore the performance of dense visual perception; 2) DocVQA [29] and ChartQA [28] to test the improvement on downstream tasks; 3) MMVet [51] to monitor changes in the general



performance of the model. Our own document test set contains pure OCR and markdown conversion tasks. In a pure OCR task, the test split includes 100 pages in both Chinese and English, which are randomly extracted from arxiv and ebook. In the markdown conversion task, the test set obtains 200 pages, of which 100 pages contain tables and another 100 pages have mathematical formulas.

We report Normalized Edit Distance [6, 17] and F1-score along with the precision and recall for document parsing. For DocVQA, ChartQA, and MMVet, we use their vanilla metrics for a fair comparison with other LVLMs.

## 4.2 Implementation Details

During the vision vocabulary generating process, we optimize all parameters of Vary-tiny with a batch size of 512 and train the model for 3 epochs. We utilize the AdamW [27] optimizer and a cosine annealing scheduler [26] along with the learning rate of 5e-5 to train Vary-tiny.

In the training stage of the Vary-base, we freeze the weights of both new and vanilla (CLIP-L) vision vocabulary networks and optimize the parameters of input embedding layers and LLM. The initial learning rate is 5e-5 in pretrain while 1e-5 in SFT. Both the pretrain and SFT enjoy a batch size of 256 and an epoch of 1. Other settings are the same as Vary-tiny.

Method	Forms	Pure Document OCR		Markdown Format Conversion		
		Chinese	English	Formula	Table	Average
Nougat [6]	Edit Distance ↓	–	0.126	0.154	0.335	0.245
	F1-score ↑	–	<b>89.91</b>	83.97	75.97	79.97
	Prediction ↑	–	89.12	82.47	75.21	78.84
	Recall ↑	–	<b>90.71</b>	<b>85.53</b>	<b>76.74</b>	81.14
Vary-tiny	Edit Distance ↓	0.266	0.197	–	–	–
	F1-score ↑	86.00	84.25	–	–	–
	Prediction ↑	86.14	89.38	–	–	–
	Recall ↑	85.86	79.67	–	–	–
Vary-base	Edit Distance ↓	0.174	<b>0.106</b>	<b>0.082</b>	<b>0.280</b>	0.181
	F1-score ↑	87.32	88.24	<b>85.94</b>	<b>76.26</b>	81.10
	Prediction ↑	86.59	<b>90.08</b>	<b>87.06</b>	<b>76.81</b>	81.94
	Recall ↑	88.06	86.47	84.84	75.71	80.28

Table 1: Fine-grained text perception compared to Nougat. Vary-tiny is the model based on OPT-125M to generate the vision vocabulary, which enjoys pure OCR ability, including Chinese and English. Vary-base is the model upon Qwen-Chat 7B after scaling up the vision vocabulary, enjoying both pure document OCR and markdown format conversation abilities through prompt control.

## 4.3 Fine-grained Perception Performance

We measure the fine-grained perception performance of Vary through the dense text recognition ability. As shown in Table 1, Vary-tiny gathers both Chinese and English dense OCR ability by the process of vision vocabulary generating. Specifically, it achieves 0.266 and 0.197 edit distance for Chinese and English documents (plain texts) OCR respectively, proving the new vision vocabulary enjoys good fine-grained text encoding capacity. For Vary-base, it can achieve an on-par performance with nougat [6] (a special document parsing model) on English plain text documents. Besides, with different prompts (*e.g.*, Convert the image to markdown format.), Vary-base can realize the document image-markdown format conversion. It is worth noting that in such a task, Vary-base (with 0.181 edit distance and 81.10% F1 on math and table average) is better than nougat (with 0.245 edit distance and 79.97% F1 on average) to some extent, which may be due to the super strong text correction ability of the 7B LLM (Qwen). All the above results indicate that by scaling up the vision vocabulary, the new LVLM can lift its fine-grained perception performance.

## 4.4 Downstream Task Performance

We test the performance improvement on downstream VQA tasks with DocVQA [29] and ChartQA [28]. We use the addition prompt: "Answer the following questions using a single word

Method	DocVQA		ChartQA		
	val	test	human	augmented	Average
Dessurt [10]	46.5	63.2	-	-	-
Donut [14]	-	67.5	-	-	41.8
Pix2Sturct [16]	-	72.1	30.5	81.6	56.0
mPLUG-DocOwl [49]	-	62.2	-	-	57.4
Matcha [23]	-	-	38.2	<u>90.2</u>	64.2
Qwen-VL [3]	-	65.1	-	-	65.7
Vary-base (80k)	<u>78.2</u>	76.3	43.2	87.3	65.3
Vary-base (665k)	78.1	76.3	<u>43.8</u>	88.3	<u>66.1</u>

Table 2: Comparison with popular methods on DocVQA and ChartQA. 80k represents that the SFT data is LLaVA-80k while 665k is the LLaVA-CC665k. The metric of DocVQA is ANLS while the ChartQA is relaxed accuracy following their vanilla papers.

Method	MM-Vet						
	Rec	OCR	Know	Gen	Spat	Math	Total
BLIP-2 [19]	27.5	11.1	11.8	7.0	16.2	5.8	22.4
LLaVA-7B [25]	28.0	17.1	16.3	18.9	21.2	<u>11.5</u>	23.8
MiniGPT-4 [55]	29.9	16.1	20.4	22.1	22.2	3.8	24.4
Otter [18]	27.3	17.8	14.2	13.8	24.4	3.8	24.7
OpenFlamingo [1]	28.7	16.7	16.4	13.1	21.0	7.7	24.8
LLaVA-13B [25]	<u>39.2</u>	22.7	<u>26.5</u>	<u>29.3</u>	29.6	7.7	32.9
LLaVA1.5-7B [24]	-	-	-	-	-	-	30.5
Vary-base (vicuna7B) (665k)	38.7	22.0	23.6	24.1	29.6	7.7	32.9
Vary-base (qwen7B) (80k)	38.9	<u>30.1</u>	22.4	21.7	<u>34.3</u>	7.7	<u>36.2</u>

Table 3: Comparison with popular methods on MMVet. The abbreviations represent: Rec: Recognition; Know: Knowledge; Gen: Language generation; Spat: Spatial awareness.

or phrase:" [24] to allow the model to output short and precise answers. As shown in Table 2, Vary-base (with Qwen-7B as LLM) can achieve 78.2% (test) and 76.3% (val) ANLS on DocVQA upon LLaVA-80k [25] SFT data. With LLaVA-665k [24] data for SFT, Vary-base can reach 66.1% average performance on ChartQA. The performance on both two challenging downstream tasks is comparable to or even better than Qwen-VL [4], demonstrating the proposed vision vocabulary scaling-up method is also promising for downstream.

#### 4.5 General Performance

We monitor the general performance of Vary through MMVet [51] benchmark. As shown in table 3, with the same LLM (Vicuna-7B) and SFT data (LLaVA-CC665k), Vary lifts 2.4% (32.9% vs. 30.5%) of the total metric than LLaVA-1.5, proving that our data and training strategy do not hurt the model’s general ability. Besides, Vary with Qwen-7B and LLaVA-80k can achieve 36.2% performance, further demonstrating the effectiveness of our vision vocabulary scaling-up manner.

## 5 Conclusion

This paper highlights that scaling up the vocabulary in the visual branch for an LVLM is quite significant and we successfully devise a simple method to prove such a claim. According to the experiments, the provided model, Vary, achieves promising scores in multiple tasks, which is mainly profited by the new vocabulary we generated. Despite the satisfactory performance of Vary, we believe that how to effectively scale up the visual vocabulary still enjoys much improvement rooms, especially compared to the mature and relatively simple means of expanding text vocabulary. We hope that the useful and efficient design of Vary will attract more research attention to such a direction.



## 6 Appendix

In this appendix, we present the output results of our model to provide a more intuitive understanding of its performance.

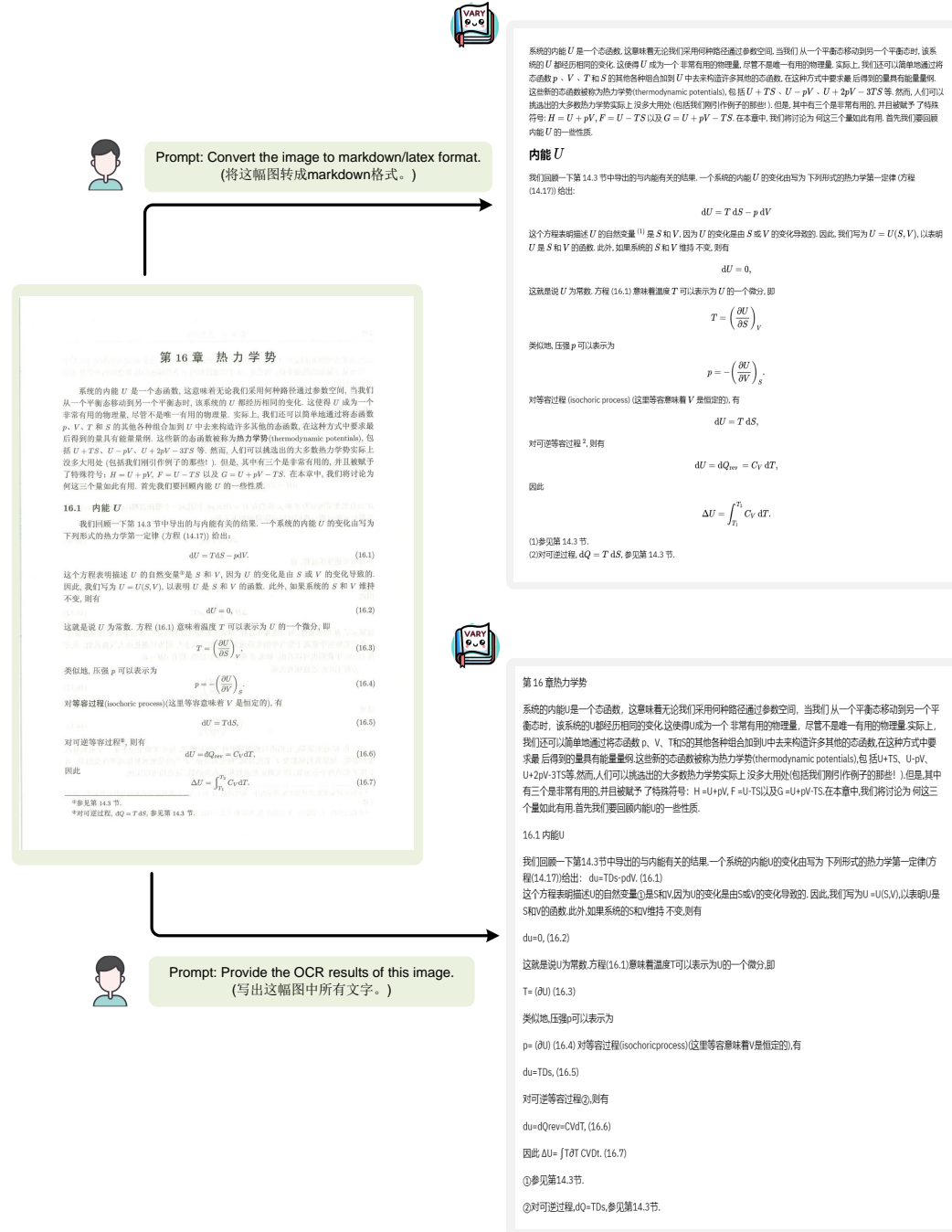


Figure 5: Instruction following ability of Vary-base to excel markdown conversion or pure OCR. Vary-base can control the output format for a document image input upon the user's prompts.



Figure 6: Fine-grained visual perception ability of Vary-base on English document dense OCR. This image is the page 3 of [44].

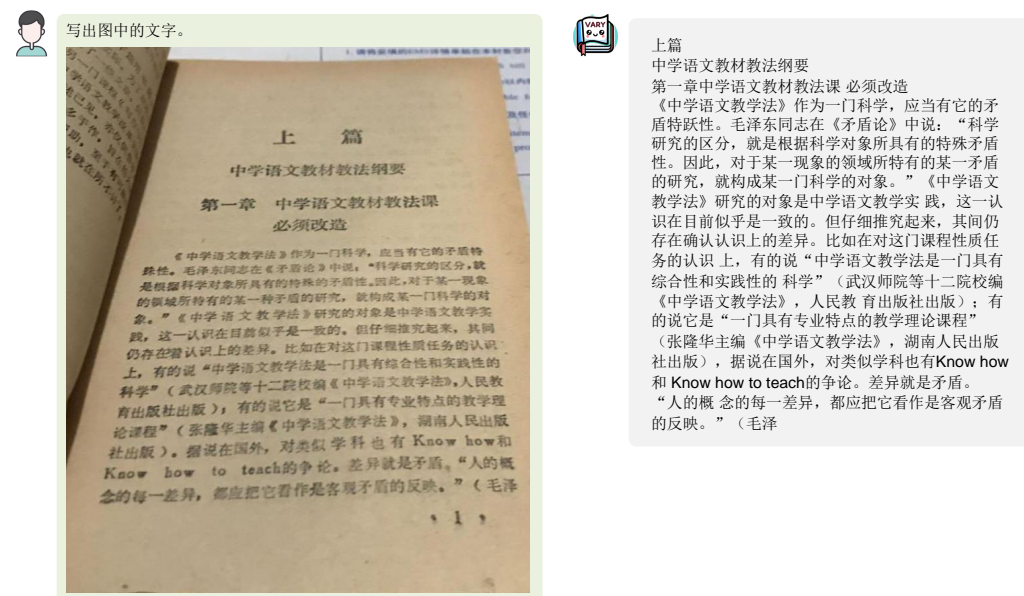


Figure 7: Fine-grained visual perception ability of Vary-base on Chinese book dense OCR. This image is from the Internet.



Convert the image to markdown format.

高二数学导数部分大题练习

5. 已知函数  $f(x) = \ln(x-1) - k(x-1) + 1$ .

- (I) 当  $k=1$  时, 求函数  $f(x)$  的最大值;  
(II) 若函数  $f(x)$  没有零点, 求实数  $k$  的取值范围;

6. 已知  $x=2$  是函数  $f(x) = (x^2 + ax - 2a - 3)e^x$  的一个极值点 ( $e \approx 2.718 \dots$ ).

- (I) 求实数  $a$  的值;  
(II) 求函数  $f(x)$  在区间  $[e, e^2]$  上的最大值和最小值.

7. 已知函数  $f(x) = x^2 - 4x + (2-a)\ln x$  ( $a \in \mathbf{R}, a \neq 0$ ).

- (I) 当  $a=18$  时, 求函数  $f(x)$  的单调区间;  
(II) 求函数  $f(x)$  在区间  $[e, e^2]$  上的最小值.

8. 已知函数  $f(x) = x(x-6) + a\ln x$  在  $x \in (2, +\infty)$  上不具有单调性.

- (I) 求实数  $a$  的取值范围;  
(II) 若  $f(x)$  是  $f(x)$  的导函数, 设  $g(x) = f(x) + \frac{2}{x}$ , 试证明: 对任意两个不相等正数  $x_1, x_2$ , 不等式  $|g(x_1) - g(x_2)| > \frac{2}{25} |x_1 - x_2|$  恒成立.



5. 已知函数  $f(x) = \ln(x-1) - k(x-1) + 1$ .

- (I) 当  $k=1$  时, 求函数  $f(x)$  的最大值;  
(II) 若函数  $f(x)$  没有零点, 求实数  $k$  的取值范围;

6. 已知  $x=2$  是函数  $f(x) = (x^2 + ax - 2a - 3)e^x$  的一个极值点 ( $e \approx 2.718 \dots$ ).

- (I) 求实数  $a$  的值;  
(II) 求函数  $f(x)$  在  $x \in [\frac{3}{2}, 3]$  的最大值和最小值.

7. 已知函数  $f(x) = x^2 - 4x + (2-a)\ln x$  ( $a \in \mathbf{R}, a \neq 0$ ).

- (I) 当  $a=18$  时, 求函数  $f(x)$  的单调区间;  
(II) 求函数  $f(x)$  在区间  $[e, e^2]$  上的最小值.

8. 已知函数  $f(x) = x(x-6) + a\ln x$  在  $x \in (2, +\infty)$  上不具有单调性.

- (I) 求实数  $a$  的取值范围;  
(II) 若  $f(x)$  是  $f(x)$  的导函数, 设  $g(x) = f'(x) + 6 - \frac{2}{x}$ , 试证明: 对任意两个不相等正数  $x_1, x_2$ , 不等式  $|g(x_1) - g(x_2)| \geq \frac{3}{25} |x_1 - x_2|$  恒成立.

Figure 8: Markdown/Latex format conversion ability (on math formula) of Vary-base. This image is from the Internet.



Convert this table to latex format.

Variable	Obs	Mean	Std. dev.	Min	Max
Inflation	874	31.447	213.399	-13.056	4157.317
EFW3a	685	30.736	157.736	-64.6	2294.3
EFW4	690	6.021	1.697	1.660	9.764
EFW4a	698	6.070	2.621	0	10
EFW4d	768	7.374	3.769	0	10
EFW4e	782	2.974	3.289	0	10
EFW5	654	5.366	1.109	2.473	8.755
EFW5a	749	5.839	2.737	0	10
Averschool	558	5.469	2.695	0.2	12.005
Averschoolqua	523	5.715	3.914	0.183	16.677

Variable	Obs	Mean	Std. dev.	Min	Max
Ln(Inflation)	825	2.076	1.326	-2.882	8.333
Ln(EFW3a)	639	2.365	1.178	-1.897	7.738
Ln(EFW4)	690	1.746	0.338	0.507	2.279
Ln(EFW4a)	673	1.719	0.596	-2.302	2.302
Ln(EFW4d)	648	2.087	0.542	-1.609	2.302
Ln(EFW4e)	481	1.360	0.694	-0.956	2.302
Ln(EFW5)	654	1.656	0.229	0.905	2.169
Ln(EFW5a)	697	1.724	0.571	-1.791	2.302
Ln(Averschool)	558	1.545	0.613	-1.609	2.485
Ln(Averschool-qua)	521	1.468	0.797	-1.696	2.814



Variable	Obs	Mean	Std. dev.	Min	Max
Inflation	874	31.447	213.399	-13.056	4157.317
EFW3a	685	30.736	157.736	-64.6	2294.3
EFW4	690	6.021	1.697	1.660	9.764
EFW4a	698	6.070	2.621	0	10
EFW4d	768	7.374	3.769	0	10
EFW4e	782	2.974	3.289	0	10
EFW5	654	5.366	1.109	2.473	8.755
EFW5a	749	5.839	2.737	0	10
Averschool	558	5.469	2.695	0.2	12.005
Averschoolqua	523	5.715	3.914	0.183	16.677

Variable	Obs	Mean	Std. dev.	Min	Max
Ln(Inflation)	825	2.076	1.326	-2.882	8.333
Ln(EFW3a)	639	2.365	1.178	-1.897	7.738
Ln(EFW4)	690	1.746	0.338	0.507	2.279
Ln(EFW4a)	673	1.719	0.596	-2.302	2.302
Ln(EFW4d)	648	2.087	0.542	-1.609	2.302
Ln(EFW4e)	481	1.360	0.694	-0.956	2.302
Ln(EFW5)	654	1.656	0.229	0.905	2.169
Ln(EFW5a)	697	1.724	0.571	-1.791	2.302
Ln(Averschool)	558	1.545	0.613	-1.609	2.485
Ln(Averschool-qua)	521	1.468	0.797	-1.696	2.814

Figure 9: Markdown/Latex format conversion ability (on the table) of Vary-base. The images are from the Internet.

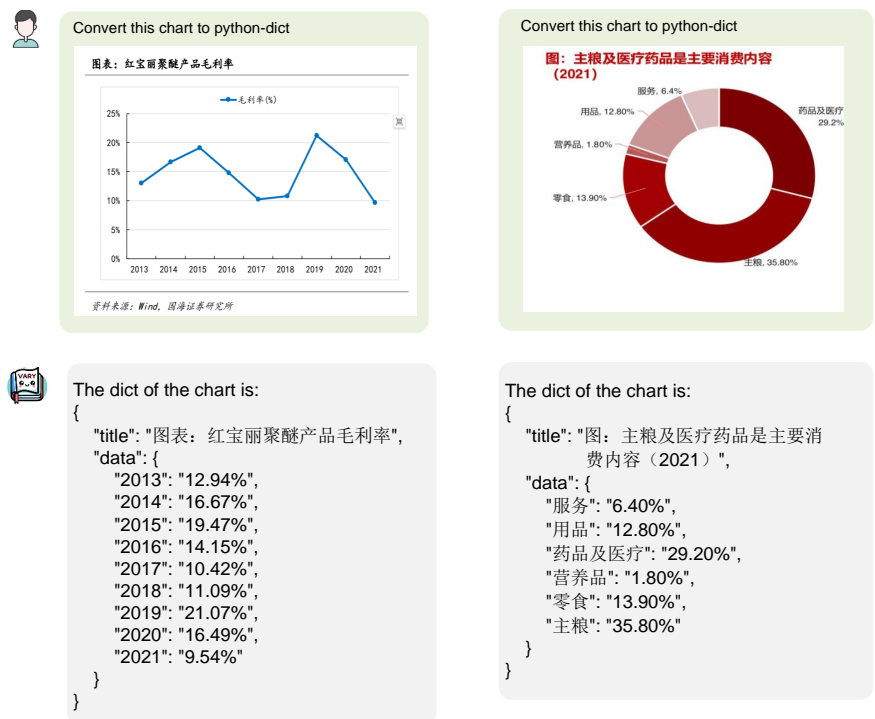


Figure 10: Chart understanding (Chinese) of Vary-base. The images are from the Internet.

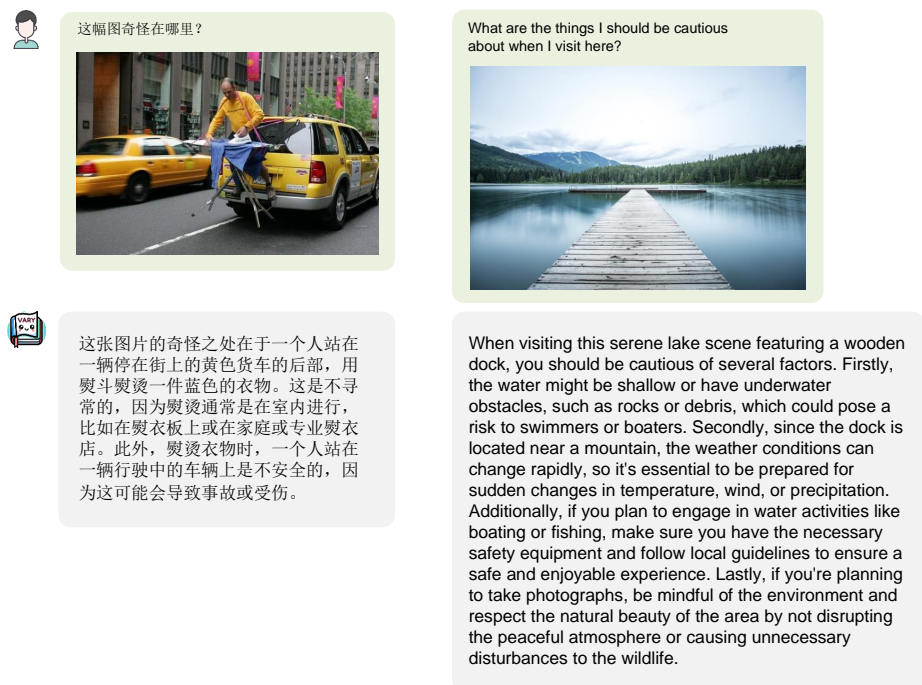


Figure 11: General performance of Vary-base. The images are from LLaVA [25] samples.

## References

- [1] Alayrac, J., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J.L., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., Simonyan, K.: Flamingo: a visual language model for few-shot learning. In: NeurIPS (2022) [1](#), [2](#), [3](#), [8](#)
- [2] Alibaba: Introducing qwen-7b: Open foundation and human-aligned models (of the state-of-the-arts). <https://github.com/QwenLM/Qwen-7B> (2023) [6](#)
- [3] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, S., Yao, Y., Yu, B., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., Zhu, T.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023) [5](#), [8](#)
- [4] Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966 (2023) [1](#), [8](#)
- [5] Biten, A.F., Litman, R., Xie, Y., Appalaraju, S., Manmatha, R.: Latr: Layout-aware transformer for scene-text vqa. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16548–16558 (2022) [3](#)
- [6] Blecher, L., Cucurull, G., Scialom, T., Stojnic, R.: Nougat: Neural optical understanding for academic documents. arXiv preprint arXiv:2308.13418 (2023) [7](#)
- [7] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020) [1](#)
- [8] Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality. <https://lmsys.org/blog/2023-03-30-vicuna/> (2023) [2](#), [3](#), [5](#), [6](#)
- [9] Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. Advances in neural information processing systems **30** (2017) [3](#)
- [10] Davis, B., Morse, B., Price, B., Tensmeyer, C., Wigington, C., Morariu, V.: End-to-end document recognition and understanding with dessurt. In: European Conference on Computer Vision. pp. 280–296. Springer (2022) [8](#)
- [11] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) [1](#), [3](#)
- [12] Dong, R., Han, C., Peng, Y., Qi, Z., Ge, Z., Yang, J., Zhao, L., Sun, J., Zhou, H., Wei, H., et al.: Dreamllm: Synergistic multimodal comprehension and creation. arXiv preprint arXiv:2309.11499 (2023) [1](#)
- [13] Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., Lv, T., Cui, L., Mohammed, O.K., Liu, Q., et al.: Language is not all you need: Aligning perception with language models. arXiv preprint arXiv:2302.14045 (2023) [5](#)
- [14] Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., Park, S.: Ocr-free document understanding transformer. In: European Conference on Computer Vision. pp. 498–517. Springer (2022) [8](#)
- [15] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023) [4](#)
- [16] Lee, K., Joshi, M., Turc, I.R., Hu, H., Liu, F., Eisenschlos, J.M., Khandelwal, U., Shaw, P., Chang, M.W., Toutanova, K.: Pix2struct: Screenshot parsing as pretraining for visual language understanding. In: International Conference on Machine Learning. pp. 18893–18912. PMLR (2023) [8](#)
- [17] Levenshtein, V.I., et al.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710. Soviet Union (1966) [7](#)
- [18] 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) [8](#)

- [19] Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023) 1, 3, 8
- [20] Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: European Conference on Computer Vision. pp. 280–296. Springer (2022) 4
- [21] Liang, Y., Wu, C., Song, T., Wu, W., Xia, Y., Liu, Y., Ou, Y., Lu, S., Ji, L., Mao, S., et al.: Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis. arXiv preprint arXiv:2303.16434 (2023) 3
- [22] Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV. pp. 740–755 (2014) 1, 5
- [23] Liu, F., Piccinno, F., Krichene, S., Pang, C., Lee, K., Joshi, M., Altun, Y., Collier, N., Eisenschlos, J.M.: Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. arXiv preprint arXiv:2212.09662 (2022) 8
- [24] Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning (2023) 6, 8
- [25] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning (2023) 1, 3, 6, 8, 12
- [26] Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 7
- [27] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019) 7
- [28] Masry, A., Long, D.X., Tan, J.Q., Joty, S., Hoque, E.: Chartqa: A benchmark for question answering about charts with visual and logical reasoning. arXiv preprint arXiv:2203.10244 (2022) 6, 7
- [29] Mathew, M., Karatzas, D., Jawahar, C.: Docvqa: A dataset for vqa on document images. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2200–2209 (2021) 3, 6, 7
- [30] Mishra, A., Shekhar, S., Singh, A.K., Chakraborty, A.: Ocr-vqa: Visual question answering by reading text in images. In: 2019 international conference on document analysis and recognition (ICDAR). pp. 947–952. IEEE (2019) 1
- [31] OpenAI: Chatgpt. <https://openai.com/blog/chatgpt/> (2023) 3
- [32] OpenAI: Gpt-4 technical report (2023) 1, 6
- [33] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P.F., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback. In: NeurIPS (2022) 1, 3
- [34] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 2, 3
- [35] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog 1(8), 9 (2019) 1, 3
- [36] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research 21(1), 5485–5551 (2020) 1, 3
- [37] Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021) 2, 6
- [38] Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y.: Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. arXiv preprint arXiv:2303.17580 (2023) 3
- [39] Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8317–8326 (2019) 1
- [40] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca) (2023) 3



- [41] Team, M., et al.: Introducing mpt-7b: A new standard for open-source, commercially usable llms (2023) [6](#)
- [42] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023) [1](#), [3](#)
- [43] Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 (2016) [1](#)
- [44] Wei, H., Guo, P., Zhu, Y., Liu, C., Wang, P.: Humanliker: A human-like object detector to model the manual labeling process. Advances in Neural Information Processing Systems **35**, 2294–2306 (2022) [10](#)
- [45] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al.: Emergent abilities of large language models. arXiv preprint arXiv:2206.07682 (2022) [3](#)
- [46] Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., Duan, N.: Visual chatgpt: Talking, drawing and editing with visual foundation models. arXiv preprint arXiv:2303.04671 (2023) [3](#)
- [47] Yang, R., Song, L., Li, Y., Zhao, S., Ge, Y., Li, X., Shan, Y.: Gpt4tools: Teaching large language model to use tools via self-instruction. arXiv preprint arXiv:2305.18752 (2023)
- [48] 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) [3](#)
- [49] Ye, J., Hu, A., Xu, H., Ye, Q., Yan, M., Dan, Y., Zhao, C., Xu, G., Li, C., Tian, J., et al.: mplug-docowl: Modularized multimodal large language model for document understanding. arXiv preprint arXiv:2307.02499 (2023) [2](#), [8](#)
- [50] Yu, E., Zhao, L., Wei, Y., Yang, J., Wu, D., Kong, L., Wei, H., Wang, T., Ge, Z., Zhang, X., et al.: Merlin: Empowering multimodal llms with foresight minds. arXiv preprint arXiv:2312.00589 (2023) [1](#)
- [51] 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. arXiv preprint arXiv:2308.02490 (2023) [6](#), [8](#)
- [52] Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., et al.: Glm-130b: An open bilingual pre-trained model. arXiv preprint arXiv:2210.02414 (2022) [3](#)
- [53] Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., et al.: Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068 (2022) [1](#), [3](#), [4](#)
- [54] Zhao, L., Yu, E., Ge, Z., Yang, J., Wei, H., Zhou, H., Sun, J., Peng, Y., Dong, R., Han, C., et al.: Chatspot: Bootstrapping multimodal llms via precise referring instruction tuning. arXiv preprint arXiv:2307.09474 (2023) [1](#)
- [55] 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) [1](#), [8](#)