Vary: Scaling up the Vision Vocabulary for Large Vision-Language Model

Haoran Wei[1⋆], Lingyu Kong[2⋆⋆], Jinyue Chen[2], Liang Zhao[1], Zheng Ge[1], Jinrong Yang[3], Jianjian Sun[1], Chunrui Han[1], and Xiangyu Zhang[1]

[1] MEGVII Technology, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Huazhong University of Science and Technology, Wuhan, China
weihaoran18@mails.ucas.ac.cn
https://github.com/Ucas-HaoranWei/Vary/

Abstract. Most Large Vision-Language Models (LVLMs) enjoy the same vision vocabulary, *i.e.*, CLIP, for common vision tasks. However, for some special task that needs dense and fine-grained perception, the CLIP-style vocabulary may encounter low efficiency in tokenizing corresponding vision knowledge and even suffer out-of-vocabulary problems. Accordingly, we propose **Vary**, an efficient and productive method to scale up the **V**ision vocabul**ary** 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 compress rich vision signals. Next, we scale up the vanilla vision vocabulary by merging the new with the original one (CLIP), enabling the LVLMs to garner new features effectively. We present frameworks with two sizes: Vary-base (7B) and Vary-toy (1.8B), both of which enjoy excellent fine-grained perception performance while maintaining great general ability.

Keywords: LVLM · Vision vocabulary · Fine-grained perception

1 Introduction

Recently, research into vision dialogue robots [1, 23, 29, 36, 63] has been gaining significant traction. These human-like models, mainly relying on two components, *i.e.,* large language models (LLMs) [6, 35, 38, 46, 61] and vision vocabulary networks, can not only converse based on user's prompts but also perform well on downstream tasks, such as VQA [26, 43], image caption [47], OCR [34], and so on. It is undeniable that large vision-language models (LVLMs) will drive the AI community towards the direction of artificial general intelligence (AGI).

⋆ The work was supported by the National Science and Technology Major Project of China (2023ZD0121300). The work was also supported by the Research on AI Terminal Computing Power and End-Cloud Framework (R2411B0R) of China Mobile Group Device Co., Ltd.

⋆⋆ Equal contribution

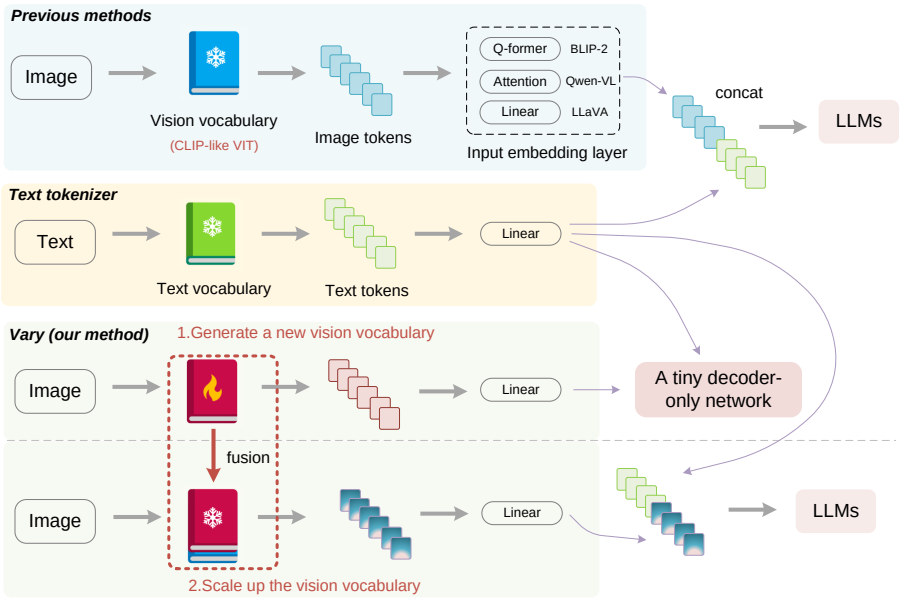

Fig. 1: Previous method *vs.* Vary: Unlike other models that use a ready-made vision vocabulary, the processes of Vary can be divided into two stages: the generation and fusion of vision vocabulary. In the first stage, we use a vocabulary network along with a tiny decoder-only network to produce a powerful 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.

Popular GPT-4 [35]-like LVLMs, *e.g.*, BLIP2 [23], MiniGPT4 [63], LLaVA [29], Qwen-VL [4], and *etc.* [12, 57, 62] enjoy a stunning performance in multiple aspects with own programming paradigm. Based on an LLM [39, 61], 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 latent. 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 to the text input embedding layer in text tokenizer [46, 61], ensuring consistency in the structure of image and text branches. For Qwen-VL, it utilizes a cross-attention layer to compress and align the image tokens, making the model accept a larger input resolution. Although the above LVLMs' vision input embedding networks are variable (*e.g.*, MLP [29], Qformer, Perceiver [1]), their vision vocabulary is almost identical (a CLIP-based [37] 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 [40] image-text pairs, covering most natural images and vision scenes. However, for some special scenarios, *e.g.*, high-resolution perception, non-English OCR, document/chart

understanding, and so on, the CLIP may regard them as "foreign languages", leading to inefficient tokenizing, *i.e.*, difficulty compress rich vision information into 256×1024 tokens. Although mPlug-Owl [56] and Qwen-VL alleviate the above issues by unfreezing its vision vocabulary network (CLIP) in pretraining, we argue that such a manner may not be reasonable due to four 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 train a dataset with multiple epochs (due to the strong "memory" ability of LLMs) which maybe an important setting in computer vision perception tasks. Therefore, a natural question is: *Is there a strategy that can simplify and effectively intensify the vision vocabulary for an LVLM?*

In this paper, we propose Vary, an efficient and effective approach, to answer the above question. Vary is inspired by the text vocabulary expansion manner in vanilla LLMs [8]. 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 LVLMs, 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 to construct a better one. As shown in Fig. 1, we build a small-size pipeline consisting of an encoder network and a tiny decoder-only transformer in the first step to generate the new vocabulary via predicting the next token. It is worth noting that the autoregressive-based process of generating a vocabulary is perhaps more flexible for dense scenes 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 object detection [26, 41] data with the prompt. After preparing the new vision vocabulary, we graft it to the vanilla LVLMs to introduce new features. In this process, we freeze both the new and old vocabulary 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 LaTeX, Chinese/English chart parsing, and so on, while achieving excellent downstream performances. Specifically, Vary-base achieves 79.1% ANLS on DocVQA [33], 66.3% relaxed accuracy on ChartQA [32], 88.6% accuracy on RefCOCO [16]and 36.4% overall accuracy on MMVet [58]. Besides, along with only 1.8B LLM [2], Vary-toy can achieve 65.0% on DocVQA, 59.2% on ChartQA, 88.0% on RefCOCO, and 30.4% on MMVet.

In summary, Vary is a useful strategy to strengthen the vision vocabulary of LVLMs, which can be utilized at arbitrary downstream visual tasks that CLIP is not good at. In addition to the document OCR, chart parsing, and object detection mentioned in this paper, we believe Vary still enjoys more fine-grained tasks. We appeal to researchers to rethink the design ideas of LVLMs from the perspective of vision 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 [38], T5 [39], *etc.* Subsequently, researchers have uncovered the concept of an "emergent ability" [50] in LLMs. This implies that as language model sizes reach a certain threshold, there may be a qualitative leap in their capabilities. Motivated by the tremendous success of the decoder-only GPT series, a multitude of other open-source LLMs have emerged, including OPT [61], LLaMA [46], GLM [59], 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 [44], 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 [25, 42, 51, 53, 55] 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 [23] introduces Q-Former to enhance the alignment of visual features with the language space. More recently, LLaVA [29] 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 [37]. For an LVLM, CLIP 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, scene text OCR. However, some images under special scenarios, *e.g.*, high-resolution image, non-English OCR, document/chart parsing, and so on, will still be regarded as a "foreign language" by CLIP, which may lead to vision "out-of-vocabulary" problem and in turn become a bottleneck.

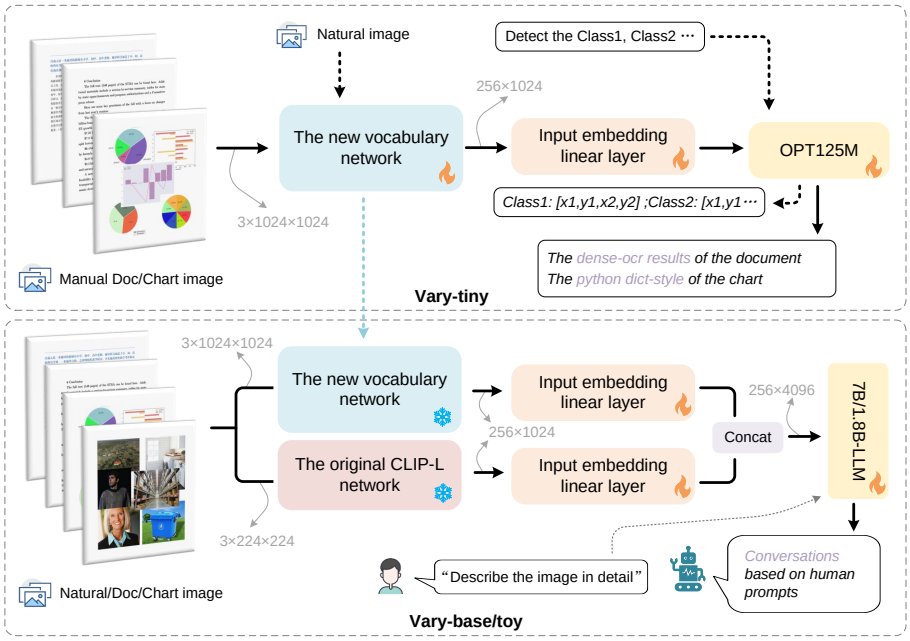

Fig. 2: Overview of the Vary. There are two types of Vary form: Vary-tiny and Vary-base/toy. Vary-tiny is mainly focused on generating a new vision vocabulary while Vary-base/toy is the final Vary body (a new LVLM) aiming to handle various visual tasks based on the new vision vocabulary.

3 Method

3.1 Architecture

Vary enjoys two main conformations: Vary-tiny and Vary-base/toy, as shown in Fig. 2. We devise the Vary-tiny to produce a new vision vocabulary and the Vary-base/toy to make use of the new vocabulary. Specifically, Vary-tiny mainly comprises a vocabulary encoder and a tiny OPT-125M [61] decoder. Between the two modules, we add a linear layer to align the channel dimensions. We hope the new vocabulary network can excel in compressing rich visual signals, *e.g.*, texts within documents/charts, and locations of objects, to compensate for CLIP's shortcomings. Thus we utilize the corresponding three types of data to train the model. Afterward, we extract the new vocabulary network and transplant it to a new language network to build the Vary main body. As shown in the lower half of Fig. 2, the new and original vocabulary (CLIP) networks enjoy independent input embedding layers and are integrated before the LLM. In such a stage, we freeze both weights of two vocabulary networks and unfreeze the weights of others to optimize the whole Vary.

3.2 Towards Generating a New Vision Vocabulary

The New Vocabulary Network We use the SAM [18] initialized ViTDet [24] image encoder (base version with about 80M parameters) as the main body of the new vocabulary network of Vary. Due to the input resolution of the SAM-base encoder being (1024×1024) while the output stride being 16, the feature shape of the last layer is (64×64×256 for H×W×C) that can not be aligned to the output of CLIP-L (256×1024 for N×C). Hence, we introduce two convolution layers, which we found is a good token merging unit, behind the last layer of the SAM encoder, as shown in Fig. 3. The first convolution layer possesses a kernel size of 3 and a stride of 2, aiming to transfer the feature shape to 32×32×512. The setting of the second convolution layer is the same as the first one, which can further compress the output to 16×16×1024. After that, we flattened the output feature to 256×1024 to align the image token shape of CLIP-VIT.

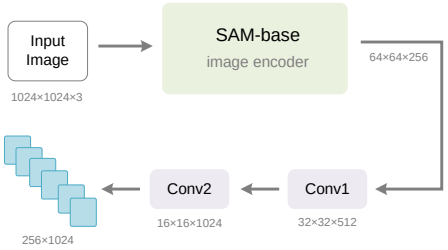

Fig. 3: The structure of new vision vocabulary network of Vary-tiny. We add two convolution layers to convert the output to be similar to CLIP.

Document Data Engine We select the high-resolution document image-text pairs as the main dataset used for the new vision vocabulary generation due to the dense OCR task can effectively validate the fine-grained image perception ability of a model. To our knowledge, there is no publicly available dataset of English and Chinese document-text pairs, so we created our own. We first collect pdf-style documents from open-access ArXiv and CC-MAIN-2021-31-PDF-UNTRUNCATED for the English set and assemble from E-books on the Internet for the Chinese set. 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. For text ground truth, we concatenate the texts within each block (box) of a page according to the order of top to bottom and left to right. For each image, all pages enjoy the same resolution (96 dpi). During this process, we construct 1M Chinese and 1M English document image-text pairs in total.

Chart Data Engine We find current LVLMs are not good at chart parsing, so we choose the chart data as another main knowledge that needs to be "written" into the new vocabulary. For chart image-text pair, we all follow a rendering way. We select both the *matplotlib* and *pyecharts* as the rendering tools. For the matplotlib-style chart, we built 250k samples in both Chinese and English. While for pyecharts, we build 500k for both two languages. 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.

Detection Data Engine To fully utilize the capacity of the vision vocabulary network and gain the natural image perception ability from SAM initialization, we also introduce object detection data in the vision vocabulary generating process. We gather samples from two large open-source datasets, *i.e.,* Object365 [41] and OpenImage [19]. Due to the low efficiency of coordinate (number texts) tokenizing in OPT's [61] text tokenizer, for images with too many objects, the number of tokens in the ground truth may exceed the maximum token length supported by OPT-125M (although we interpolate it to 4096). Therefore, we re-organize the annotations upon two tasks: 1) **Object Detection**: If there are no more than 30 object-boxes in the image, we will allow the Vary-tiny detect all objects with the prompt: "*Detect all objects in this image*". 2) **REC**: If the object-box number is over 30, we will regard this image as a REC task using a prompt template: "*Detect class1, class2, ..., in this image*". The selected classes are random so one image can be used multiple times. Through the above manner, we obtain approximately 3M of detection data.

Input Format We train all parameters of the Vary-tiny with image-text pairs by autoregression. The input format follows popular LVLMs [14, 15], *i.e*, the image tokens are packed with text tokens in the form of a prefix. We use two special tokens "<img>" and "</img>" to indicate the position of the image tokens. The OPT-125M is interpolated to 4096 for long document texts.

3.3 Towards Scaling up the Vision Vocabulary

The Structure of Vary After completing the new vocabulary network, we graft it to our LVLM – Vary-base/toy. 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 layer, as shown in Fig. 2. For Vary-base, 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]). For Vary-toy, both the input and output channels are 1024 to align the Qwen-1.8B [3].

LaTeX Document Rendering Except for the collected coarse document data in Section 3.2, we also need purified data that enjoys a certain format to support formula and table parsing. To this end, we create new document data through LaTeX rendering. Firstly, we collect abundant "*.tex*" source files on Arxiv, and then extracted tables, mathematical formulas, and plain texts using regular expressions. Finally, we re-render these contents with *pdflatex* based on the new templates we prepared. We prepare 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. Through the above processes, we acquired about 0.5 million English and Chinese document pages.

Text Associated Chart Rendering In Section 3.2, 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 situation 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 main body, we hope to use higher quality (strong texts correlated) data due to the LLM being unfrozen. To this end, we use GPT-4 [35] to generate some chart corpus and then we utilize the high-quality chart texts to additionally render 200k chart images for the Vary-base/toy training.

Data Ratio in Pre-training & SFT For Vary, the pretrain stage is a multi-task training stage, wherein we prepare abundant image-text pairs in various formats. We mainly focus on six types of data in such a stage, containing weakly annotated image caption, PDF-dense OCR, chart parsing, object detection, pure text conversation, and VQA. Specifically, for general natural images, we sample 4M image-text pairs from the Laion-COCO [40] dataset, and we also use the BLIP-558K data proposed in LLaVA [29]. For the PDF image-text pair, we prepare two types of data: one is pure dense text OCR, and the other is a task that converts the PDF image to a markdown format. The previous type of data is randomly sampled from the PDF data used in Vary-tiny and the last one is obtained via LaTeX rendering as aforementioned. For the detection data, images are gathered from the COCO [26] dataset. We sample 50K images with fewer objects for the object detection task and use all train data of RefCOCO for the REC task. We normalize the coordinates of each box and magnify the values to 1000 times. To prevent the language ability of the LLM from deteriorating, we also introduce pure NLP conversation data, including ShareGPT, Baize [52], and Alpaca [44]. For the downstream VQA tasks, we choose two challenge datasets (DocVQA and ChartQA [32]) to monitor the dense text perception and reasoning performance of Vary for artificial data. There are at least 10 prompts made through GPT3.5 [6] for each task.

In the SFT stage, we use the LLaVA-80k or 665k [29] to continue training the model. Both LLaVA-80k and 665k are general SFT datasets with detailed descriptions and prompts produced by GPT4 [29, 35].

Conversation Format When using the Vicuna-7B as the LLM, we prepare the conversation format following Vicuna v1 [8], *i.e.*, USER: <img>"image"</img> "texts input" ASSITANT: "texts output" </s>. When utilizing the Qwen-7B or Qwen1.8B [2], we design the conversations following the LLaVA-MPT [45] format, which can be described as: <|im_start|>user: <img>"image"</img> "texts input"<|im_end|> <|im_start|>assistant: "texts output" <|im_end|>. The "image" represents 256 image tokens.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate the proposed Vary on multiple datasets, including 1) a document-level OCR test set that we created to explore the performance of dense vision recognition; 2) DocVQA [33], ChartQA [32], and RefCOCO [16] to test the improvement on downstream tasks; 3) MMVet [58] to monitor fluctuations in the general scenes of the model. The document test benchmark contains pure OCR and markdown conversion tasks. In the pure OCR task, the set includes 100 pages with both Chinese and English, which are randomly extracted from Arxiv and E-book. In the markdown conversion task, the test set obtains 200 pages, of which 100 pages contain tables and another 100 pages enjoy formulas.

We report Normalized Edit Distance [5,21] and F1-score along with the page-level precision and recall for the document parsing task. For downstream tasks, *e.g.,* DocVQA, ChartQA, RefCOCO, 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 [31] optimizer and a cosine annealing scheduler [30] along with the learning rate of 5e-5 to train the model.

In the Vary-base/toy training stage, 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 2e-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.

4.3 Fine-grained Perception Performance

We measure the fine-grained perception performance of Vary through dense text recognition. 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 0.336 and 0.311 edit distance for Chinese and English documents (on plain texts), respectively, proving the new vision vocabulary enjoys good fine-grained text compression capacity. Based on the new vision vocabulary and Qwen-1.8B decoder, Vary-toy further lifts the document OCR accuracy compared to Vary-tiny. For Vary-base, it can achieve an on-par performance with Nougat [5] (a special document parsing model) on English plain-text style documents. Besides, with different prompts (*e.g.*, Convert the image to markdown format), both Vary-toy and Vary-base can realize the document image-markdown format conversion. It is worth noting that in such a task, Vary-base (yields 0.223 edict distance and 80.37% F1 on math and table average) is better than nougat (with 0.245 edict distance and 79.97% F1 on average), which may be due to

Table 1: Fine-grained text perception compared to Nougat. Vary-tiny is the model based on OPT-125M for generating the vision vocabulary, which enjoys pure OCR ability, covering Chinese and English. Vary-base/toy are the models upon Qwen-chat 7B/1.8B upon the new vision vocabulary, enjoying both pure document OCR and markdown format conversation abilities through prompt control.

Method	Forms	Pure OCR		Markdown Conversion		
		Chinese	English	Formula	Table	Average
Nougat [5]	Edit Distance ↓	–	0.126	0.154	0.335	0.245
	F1-score ↑	–	**89.91**	83.97	**75.97**	79.97
	Prediction ↑	–	89.12	82.47	75.21	78.84
	Recall ↑	–	**90.71**	**85.53**	**76.74**	81.14
Vary-tiny	Edit Distance ↓	0.336	0.311	–	–	–
	F1-score ↑	82.89	82.24	–	–	–
	Prediction ↑	83.15	84.37	–	–	–
	Recall ↑	82.63	80.22	–	–	–
Vary-toy	Edit Distance ↓	0.297	0.212	0.182	0.599	0.391
	F1-score ↑	83.84	85.90	79.19	70.67	74.93
	Prediction ↑	84.09	86.74	80.15	72.30	76.23
	Recall ↑	83.60	85.08	78.27	69.12	73.70
Vary-base	Edit Distance ↓	0.201	**0.119**	**0.128**	**0.317**	0.223
	F1-score ↑	86.70	88.10	**85.05**	75.68	80.37
	Prediction ↑	86.46	**89.73**	**85.39**	**77.25**	81.32
	Recall ↑	87.54	86.52	84.72	74.18	79.45

the super strong text correction ability of the 7B LLM decoder. All the above results indicate that by scaling up the vision vocabulary, the new LVLM can gather excellent fine-grained perception performance.

4.4 Downstream Task Performance

We test the performance of Vary on downstream VQA and REC tasks, including DocVQA [33], ChartQA [32], and RefCOCO [16].

For DocVQA and ChartQA, we use the addition prompt: "Answer the following question using a single word or phrase:" [28] to give the model short and precise answers. As shown in Table 2, Vary-base (with Qwen-7B as LLM) can achieve 79.1% (test) and 78.9% (val) ANLS on DocVQA upon LLaVA-80k [29] SFT data. With LLaVA-665k [28] data for SFT, Vary-base can reach 66.3% average accuracy on ChartQA. The performance on both two challenging downstream tasks is much better than Qwen-VL [4], demonstrating the proposed vision vocabulary scaling-up strategy is promising. It is worth noting that along with the only 1.8B language model, Vary-toy can achieve 65.3% ANLS on DocVQA (val) and 59.2% accuracy on ChartQA, further demonstrating the effectiveness of the proposed method.

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	DocVQA		ChartQA		
	val	**test**	**human**	**augmented**	**Average**
Dessurt [9]	46.5	63.2	-	-	-
Donut [17]	-	67.5	-	-	41.8
Pix2Sturct [20]	-	72.1	30.5	81.6	56.0
mPLUG-DocOwl [56]	-	62.2	-	-	57.4
Matcha [27]	-	-	38.2	90.2	64.2
Qwen-VL [3]	-	65.1	-	-	65.7
Qwen-VL-chat [3]	-	62.6	-	-	66.3
Vary-toy (665k)	65.3	65.0	33.1	85.2	59.2
Vary-base (80k)	78.9	79.1	43.7	87.9	65.8
Vary-base (665k)	78.4	78.2	43.9	88.6	66.3

Table 3: Comparison with popular methods on RefCOCO. Benefiting from the new vision vocabulary, Vary-base/toy can achieve 88.6%/88.0% accuracy on RefCOCO val, which is on par with the Qwen-VL-chat-7B.

Type	Method	Size	RefCOCO		
			val	**testA**	**testB**
Traditional	OFA-L [48]	-	80.0	83.7	76.4
	TransVG [10]	-	81.0	82.7	78.4
	VILLA [13]	-	82.4	87.5	74.8
	UniTAB [54]	-	86.3	88.8	80.6
LLM-based	VisionLLM-H [49]	-	-	86.7	-
	Shikra-7B [7]	7B	87.0	90.6	80.2
	Shikra-13B [7]	13B	87.8	91.1	81.7
	Qwen-VL-chat [3]	7B	88.6	92.3	84.5
	Next-chat [60]	7B	85.5	90.0	77.9
	Vary-toy (665k)	1.8B	88.0	90.8	85.1
	Vary-base (665k)	7B	88.6	92.1	84.9

For the REC task, Vary-base/toy can get 88.6%/88.0% accuracy@0.5 on the RefCOCO validation set, which is on par with Qwen-VL-chat (7B) and much better than the Shikra-13B [7]. All the above results show that along with a better vision vocabulary, Vary gathers great natural object perception ability, further proving the effectiveness of using the low-cost Vary-tiny architecture to build a vision vocabulary, allowing us to further reflect on the necessity of CLIP if we add a large amount of weakly labeled image caption data, *e.g.,* Laion-400M [40], during the new vocabulary generating process.

Table 4: Comparison with popular methods on the general vision ability benchmark – MMVet. The abbreviations are as follows: Rec: Recognition; Know: Knowledge; Gen: Language generation; Spat: Spatial awareness.

Method	MM-Vet						
	Rec	**OCR**	**Know**	**Gen**	**Spat**	**Math**	**Total**
BLIP-2 [23]	27.5	11.1	11.8	7.0	16.2	5.8	22.4
LLaVA-7B [29]	28.0	17.1	16.3	18.9	21.2	11.5	23.8
MiniGPT-4 [63]	29.9	16.1	20.4	22.1	22.2	3.8	24.4
Otter [22]	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 [29]	39.2	22.7	26.5	29.3	29.6	7.7	32.9
LLaVA1.5-7B [28]	-	-	-	-	-	-	30.5
Vary-toy (qwen1.8B)(665k)	37.0	19.6	18.5	20.0	25.1	7.7	30.4
Vary-base (vicuna7B) (665k)	39.5	30.0	24.8	23.0	33.7	11.5	36.0
Vary-base (qwen7B) (80k)	36.6	33.4	22.0	22.7	37.2	14.6	36.4

4.5 General Vision Performance

We verify the general vision performance of Vary through MMVet [58] benchmark. As shown in Table 4, with the same LLM (Vicuna-7B) and SFT data (LLaVA-CC665k), Vary lifts 5.5% (36.0% *vs.* 30.5%) of the total accuracy than LLaVA-1.5. Besides, Vary-toy with only Qwen-1.8B can achieve an on-par 30.4% accuracy with 7B-size LLaVA-1.5. The above results fully demonstrate the proposed Vary can maintain the general visual understanding ability well.

4.6 Complementarity of the New and Old Vision Vocabularies

We apply feature normalization after the input embedding layers (Fig. 2) of both the new and original (CLIP) vision vocabulary networks to monitor whether a branch will collapse before inputting the LLM under some scenarios and analyze the complementarity between them. Specifically, we select four types of common scenarios, including natural, scene text, chart, and document images for experiments. We collect 50 samples for each class to calculate the feature normalization and gain their value ranges. As shown in Fig. 4, for natural images, the feature normalization values of the proposed new vision vocabulary are between 437 and 521 while the CLIP values are from 426 to 552, indicating that the new and old vocabularies are nip and tuck in such class, with the new vocabulary mainly used for object localization and CLIP for image description. For scene text, CLIP values are approximately 100 points higher than the new vision vocabulary due to we do not use such images to train new vision vocabulary, and CLIP plays a major role in this scenario. By contrast, the feature normalization values of new vision vocabulary are much higher than the CLIP for chart/document images, especially for document images which CLIP is not good at, proving the two vocabularies complement each other to a certain extent.

Fig. 4: The values of feature normalization of two vision vocabularies. We choose four commonly used scenarios to test the relationship between two vision branches. For each type of image, we prepare 50 samples, and this figure shows one typical example and the value range of feature normalization.

4.7 Ablation Study

In this section, we conduct ablation analyses to further validate the effectiveness of our designs. We perform ablations from two aspects: the data used to generate the new vocabulary and the selection of the new and original vocabulary.

Datasets in Generating the New Vocabulary We use document/chart parsing and object detection data in the vision vocabulary generating process. To test the effect of the object detection dataset, we train a new Vary-tiny without detection data. As shown in Table 5, under this setting, the DocVQA only lifts 0.8% (from 65.3% to 66.1%) while the RefCOCO drops 9.7% (from 88.0% to 78.3%). The results demonstrate that: 1) the introduction of detection data has little harm on document text recognition; 2) "writing" general object localization knowledge into the new vision vocabulary is crucial for boosting LVLM's REC performance.

Effectiveness of the Original CLIP We verify the effect of the old vocabulary (CLIP) of Vary via blocking the new vision vocabulary. As shown in Table 5, when discarding the new vision vocabulary and only keeping the CLIP, the MMVet accuracy is 29.0% (only dropping 0.6%) yet with only 10.9% on DocVQA, proving that the CLIP is crucial for LVLM's general ability, and also indicating that it cannot handle high-resolution document-level VQA task. We also turn off the CLIP branch to further verify its necessity. Under such a setting, we additionally feed 50M image-text pair images sampled from Laion-COCO to make the new vision vocabulary enjoy general vision ability. However, it only yields 16.1% on MMVet. It is difficult to train a general vision vocabulary equiv-

Table 5: Ablation study of Vary. Vary-tiny data means the datasets used in generating the new vocabulary. The "New" represents the new vision vocabulary network. Abbreviations "Doc" and "Det" represent the document and detection data, respectively.

Vary-tiny data			**Vary-toy**		**Performance**		
Doc/Chart	Det	Pair	CLIP	New	DocVQA(val)	RefCOCO(val)	MMVet
✓	✓	×	✓	✓	65.3	88.0	30.4
✓	×	×	✓	✓	66.1	78.3	-
-	-	-	✓	×	10.9	78.6	29.0
✓	✓	×	×	✓	65.5	87.2	-
✓	✓	50M	×	✓	65.1	86.9	16.1

alent to CLIP with limited resources. The above results firmly indicate that our scaling-up vocabulary strategy is very effective at a low cost.

5 Limitation and Conclusion

This paper highlights that scaling up the vocabulary in the vision branch for an LVLM is quite significant and we successfully prove such a claim by a simple framework. Profited by the new vision vocabulary we generated, the proposed model – Vary achieves promising performances in multiple tasks. Nevertheless, Vary still suffers limitations such as potential competition between two vision vocabulary networks. We believe that how to effectively scale up the visual vocabulary at a low cost still enjoys much improvement rooms, especially compared to the mature and relatively simple means of text vocabulary expansion. We hope that the useful and efficient design of Vary will attract more research attention to such a direction.

Acknowledgement

The work was supported by the National Science and Technology Major Project of China (2023ZD0121300). The work was also supported by the Research on AI Terminal Computing Power and End-Cloud Framework (R2411B0R) of China Mobile Group Device Co., Ltd.

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)

2. Alibaba: Introducing qwen-7b: Open foundation and human-aligned models (of the state-of-the-arts). https://github.com/QwenLM/Qwen-7B (2023)
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)
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)
5. Blecher, L., Cucurull, G., Scialom, T., Stojnic, R.: Nougat: Neural optical understanding for academic documents. arXiv preprint arXiv:2308.13418 (2023)
6. 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)
7. Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., Zhao, R.: Shikra: Unleashing multimodal llm's referential dialogue magic. arXiv preprint arXiv:2306.15195 (2023)
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)
9. 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)
10. Deng, J., Yang, Z., Chen, T., Zhou, W., Li, H.: Transvg: End-to-end visual grounding with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1769–1779 (2021)
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)
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)
13. Gan, Z., Chen, Y.C., Li, L., Zhu, C., Cheng, Y., Liu, J.: Large-scale adversarial training for vision-and-language representation learning. Advances in Neural Information Processing Systems **33**, 6616–6628 (2020)
14. Hao, Y., Song, H., Dong, L., Huang, S., Chi, Z., Wang, W., Ma, S., Wei, F.: Language models are general-purpose interfaces. arXiv preprint arXiv:2206.06336 (2022)
15. 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)
16. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 787–798 (2014)
17. 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)

18. 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)
19. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. International Journal of Computer Vision **128**(7), 1956–1981 (2020)
20. 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)
21. 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)
22. 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)
23. 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)
24. 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)
25. 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)
26. 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)
27. 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)
28. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning (2023)
29. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning (2023)
30. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
32. 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)
33. 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)
34. 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)
35. OpenAI: Gpt-4 technical report (2023)
36. 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)

37. 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)
38. 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)
39. 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)
40. 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)
41. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8430–8439 (2019)
42. 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)
43. 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)
44. 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` (2023)
45. Team, M., et al.: Introducing mpt-7b: A new standard for open-source, commercially usable llms (2023)
46. 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)
47. 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)
48. Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In: International Conference on Machine Learning. pp. 23318–23340. PMLR (2022)
49. Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., et al.: Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. arXiv preprint arXiv:2305.11175 (2023)
50. 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)
51. 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)
52. Xu, C., Guo, D., Duan, N., McAuley, J.: Baize: An open-source chat model with parameter-efficient tuning on self-chat data. arXiv preprint arXiv:2304.01196 (2023)

53. 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)
54. Yang, Z., Gan, Z., Wang, J., Hu, X., Ahmed, F., Liu, Z., Lu, Y., Wang, L.: Unitab: Unifying text and box outputs for grounded vision-language modeling. In: European Conference on Computer Vision. pp. 521–539. Springer (2022)
55. 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)
56. 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)
57. 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)
58. 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)
59. 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)
60. Zhang, A., Zhao, L., Xie, C.W., Zheng, Y., Ji, W., Chua, T.S.: Next-chat: An lmm for chat, detection and segmentation. arXiv preprint arXiv:2311.04498 (2023)
61. 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)
62. 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)
63. 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)