A Survey on Multimodal Large Language Models

Shukang Yin*, Chaoyou Fu*†, Sirui Zhao*, Ke Li,
Xing Sun, Tong Xu, and Enhong Chen, *Fellow, IEEE*

Abstract—Recently, Multimodal Large Language Model (MLLM) represented by GPT-4V has been a new rising research hotspot, which uses powerful Large Language Models (LLMs) as a brain to perform multimodal tasks. The surprising emergent capabilities of MLLM, such as writing stories based on images and OCR-free math reasoning, are rare in traditional multimodal methods, suggesting a potential path to artificial general intelligence. To this end, both academia and industry have endeavored to develop MLLMs that can compete with or even better than GPT-4V, pushing the limit of research at a surprising speed. In this paper, we aim to trace and summarize the recent progress of MLLMs. First of all, we present the basic formulation of MLLM and delineate its related concepts, including architecture, training strategy and data, as well as evaluation. Then, we introduce research topics about how MLLMs can be extended to support more granularity, modalities, languages, and scenarios. We continue with multimodal hallucination and extended techniques, including Multimodal ICL (M-ICL), Multimodal CoT (M-CoT), and LLM-Aided Visual Reasoning (LAVR). To conclude the paper, we discuss existing challenges and point out promising research directions. In light of the fact that the era of MLLM has only just begun, we will keep updating this survey and hope it can inspire more research. An associated GitHub link collecting the latest papers is available at https://github.com/BradyFU/Awesome-Multimodal-Large-Language-Models.

Index Terms—Multimodal Large Language Model, Vision Language Model, Large Language Model.

1 Introduction

RECENT years have seen the remarkable progress of LLMs [1], [2], [3], [4], [5]. By scaling up data size and model size, these LLMs raise extraordinary emergent abilities, typically including instruction following [5], [6], In-Context Learning (ICL) [7], and Chain of Thought (CoT) [8]. Although LLMs have demonstrated surprising zero/few-shot reasoning performance on most Natural Language Processing (NLP) tasks, they are inherently "blind" to vision since they can only understand discrete text. Concurrently, Large Vision Models (LVMs) can see clearly [9], [10], [11], [12], but commonly lag in reasoning.

In light of this complementarity, LLM and LVM run towards each other, leading to the new field of Multimodal Large Language Model (MLLM). Formally, it refers to the LLM-based model with the ability to receive, reason, and output with multimodal information. Prior to MLLM, there have been a lot of works devoted to multimodality, which can be divided into discriminative [13], [14], [15] and generative [16], [17], [18] paradigms. CLIP [13], as a representative of the former, projects visual and textual information into a unified representation space, building a bridge for downstream multimodal tasks. In contrast, OFA [16] is a representative of the latter, which unifies multimodal tasks in a sequence-to-sequence manner. MLLM can be classified as the latter according to the sequence operation, but it manifests two representative traits compared with the traditional counterparts: (1) MLLM is based on LLM with billion-scale parameters, which is not available in previous models. (2) MLLM uses new training paradigms to unleash its full potential, such as using multimodal instruction tuning [19], [20] to encourage the model to follow new instructions. Armed with the two traits, MLLM exhibits new capabilities, such as writing website code based on images [21], understanding the deep meaning of a meme [22], and OCR-free math reasoning [23].

Ever since the release of GPT-4 [3], there has been a research frenzy over MLLMs because of the amazing multimodal examples it shows. Rapid development is fueled by efforts from both academia and industry. Preliminary research on MLLMs focuses on text content generation grounded in text prompts and image [20], [24]/video [25], [26]/audio [27]. Subsequent works have expanded the capabilities or the usage scenarios, including: (1) Better granularity support. Finer control on user prompts is developed to support specific regions through boxes [28] or a certain object through a click [29]. (2) Enhanced support on input and output modalities [30], [31], such as image, video, audio, and point cloud. Besides input, projects like NExT-GPT [32] further support output in different modalities. (3) Improved language support. Efforts have been made to extend the success of MLLMs to other languages (*e.g*. Chinese) with relatively limited training corpus [33], [34]. (4) Extension to more realms and usage scenarios. Some studies transfer the strong capabilities of MLLMs to other domains such as medical image understanding [35], [36], [37] and document parsing [38], [39], [40]. Moreover, multimodal agents are developed to assist in real-world interaction, *e.g*. embodied agents [41], [42] and GUI agents [43], [44], [45]. An MLLM timeline is illustrated in Fig. 1.

In view of such rapid progress and the promising results

- *†Chaoyou Fu is the project leader.*
- **Shukang Yin, Chaoyou Fu, and Sirui Zhao contribute equally.*

- *Shukang Yin, Sirui Zhao, Tong Xu, and Enhong Chen are with the Department of Data Science, University of Science and Technology of China, No.96, JinZhai Road Baohe District, Hefei, Anhui, 230026, China. E-mail: sirui@mail.ustc.edu.cn, cheneh@ustc.edu.cn*
- *Chaoyou Fu, Ke Li, and Xing Sun are with the Tencent YouTu Lab, Shanghai 200233, China. E-mail: bradyfu24@gmail.com*

Corresponding author: Chaoyou Fu, Sirui Zhao, and Enhong Chen.

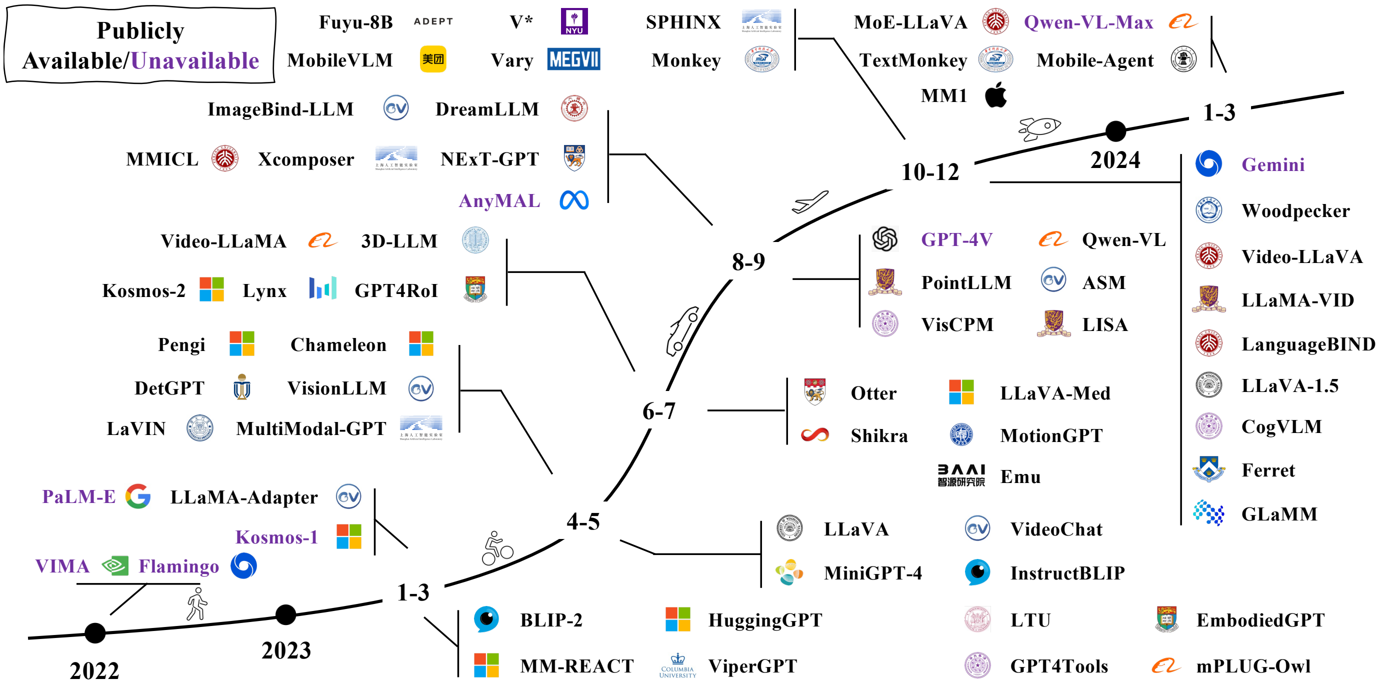

Fig. 1: A timeline of representative MLLMs. We are witnessing rapid growth in this field. More works can be found in our released GitHub page, which is updated daily.

of this field, we write this survey to provide researchers with a grasp of the basic idea, main method, and current progress of MLLMs. Note that we mainly focus on visual and language modalities, but also include works involving other modalities like video and audio. Specifically, we cover the most important aspects of MLLMs with corresponding summaries and open a GitHub page that would be updated in real time. To the best of our knowledge, this is the first survey on MLLM.

The following parts of the survey are structured as such: the survey starts with a comprehensive review of the essential aspects of MLLMs, including (1) Mainstream architectures (§2); (2) A full recipe of training strategy and data (§3); (3) Common practices of performance evaluation (§4). Then, we delve into a deeper discussion on some important topics about MLLMs, each focusing on a main problem: (1) What aspects can be further improved or extended (§5)? (2) How to relieve the multimodal hallucination issue (§6)? The survey continues with the introduction of three key techniques (§7), each specialized in a specific scenario: M-ICL (§7.1) is an effective technique commonly used at the inference stage to boost few-shot performance. Another important technique is M-CoT (§7.2), which is typically used in complex reasoning tasks. Afterward, we delineate a general idea to develop LLM-based systems to solve composite reasoning tasks or to address common user queries (§7.3). Finally, we finish our survey with a summary and potential research directions.

2 Architecture

A typical MLLM can be abstracted into three modules, *i.e.* a pre-trained modality encoder, a pre-trained LLM, and a modality interface to connect them. Drawing an analogy to humans, modality encoders such as image/audio encoders are human eyes/ears that receive and pre-process optical/acoustic signals, while LLMs are like human brains that understand and reason with the processed signals. In between, the modality interface serves to align different modalities. Some MLLMs also include a generator to output other modalities apart from text. A diagram of the architecture is plotted in Fig. 2. In this section, we introduce each module in sequence.

2.1 Modality encoder

The encoders compress raw information, such as images or audio, into a more compact representation. Rather than training from scratch, a common approach is to use a pretrained encoder that has been aligned to other modalities. For example, CLIP [13] incorporates a visual encoder semantically aligned with the text through large-scale pretraining on image-text pairs. Therefore, it is easier to use such initially pre-aligned encoders to align with LLMs through alignment pre-training (see §3.1).

The series of commonly used image encoders are summarized in Table 1. Apart from vanilla CLIP image encoders [13], some works also explore using other variants. For example, MiniGPT-4 [21] adopts an EVA-CLIP [47], [48] (ViT-G/14) encoder, which is trained with improved training techniques. In contrast, Osprey [29] introduces a convolution-based ConvNext-L encoder [46] to utilize higher resolution and multi-level features. Some works also explore encoder-free architecture. For instance, the image patches of Fuyu-8b [49] are directly projected before sending to LLMs. Thus, the model naturally supports flexible image resolution input.

TABLE 1: A summary of commonly used image encoders.

Variants	Pretraining Corpus	Resolution	Samples (B)	Parameter Size (M)
OpenCLIP-ConvNext-L [46]	LAION-2B	320	29	197.4
CLIP-ViT-L/14 [13]	OpenAI's WIT	224/336	13	304.0
EVA-CLIP-ViT-G/14 [47]	LAION-2B,COYO-700M	224	11	1000.0
OpenCLIP-ViT-G/14 [46]	LAION-2B	224	34	1012.7
OpenCLIP-ViT-bigG/14 [46]	LAION-2B	224	34	1844.9

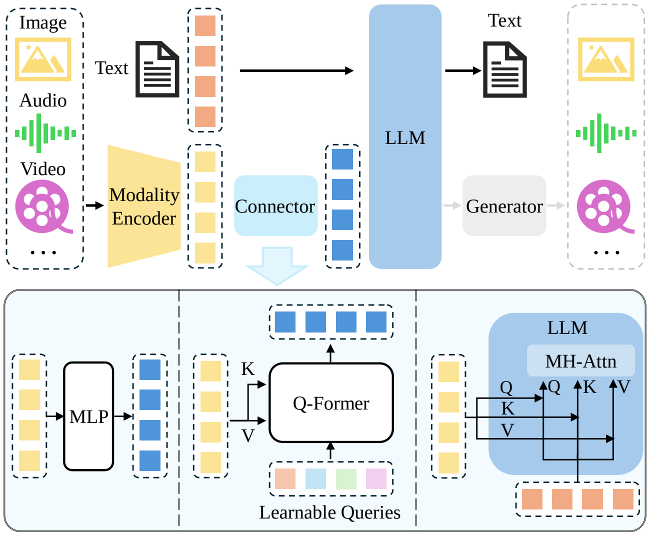

Fig. 2: An illustration of typical MLLM architecture. It includes an encoder, a connector, and a LLM. An optional generator can be attached to the LLM to generate more modalities besides text. The encoder takes in images, audios or videos and outputs features, which are processed by the connector so that the LLM can better understand. There are broadly three types of connectors: projection-based, query-based, and fusion-based connectors. The former two types adopt token-level fusion, processing features into tokens to be sent along with text tokens, while the last type enables a feature-level fusion inside the LLM.

When choosing encoders, one often considers factors like resolution, parameter size, and pretraining corpus. Notably, many works have empirically verified that using higher resolution can achieve remarkable performance gains [34], [50], [51], [52]. The approaches for scaling up input resolution can be categorized into direct scaling and patch-division methods. The direct scaling way inputs images of higher resolutions to the encoder, which often involves further tuning the encoder [34] or replacing a pre-trained encoder with higher resolution [50]. Similarly, CogAgent [44] uses a dual-encoder mechanism, where two encoders process high and low-resolution images, respectively. High-resolution features are injected into the low-resolution branch through cross-attention. Patch-division methods cut a high-resolution image into patches and reuse the low-resolution encoder. For example, Monkey [51] and SPHINX [53] divide a large image into smaller patches and send sub-images together with a downsampled high-resolution image to the image encoder, where the sub-images and the low-resolution image capture local and global features, respectively. In contrast, parameter size and training data composition are of less importance compared with input resolution, found by empirical studies [52].

Similar encoders are also available for other modalities. For example, Pengi [27] uses CLAP [54] model as the audio encoder. ImageBind-LLM [30] uses the ImageBind [55] encoder, which supports encoding image, text, audio, depth, thermal, and Inertial Measurement Unit (IMU) data. Equipped with the strong encoder, ImageBind-LLM can respond to the input of multiple modalities.

2.2 Pre-trained LLM

Instead of training an LLM from scratch, it is more efficient and practical to start with a pre-trained one. Through tremendous pre-training on web corpus, LLMs have been embedded with rich world knowledge, and demonstrate strong generalization and reasoning capabilities.

We summarize the commonly used and publicly available LLMs in Table 2. Notably, most LLMs fall in the causal decoder category, following GPT-3 [7]. Among them, Flan-T5 [56] series are relatively early LLMs used in works like BLIP-2 [59] and InstructBLIP [60]. LLaMA series [5], [57] and Vicuna family [4] are representative open-sourced LLMs that have attracted much academic attention. Since the two LLMs are predominantly pre-trained on English corpus, they are limited in multi-language support, such as Chinese. In contrast, Qwen [58] is a bilingual LLM that supports Chinese and English well.

It should be noted that scaling up the parameter size of LLMs also brings additional gains, similar to the case of increasing input resolution. Specifically, Liu *et al.* [50], [61] find that simply scaling up LLM from 7B to 13B brings comprehensive improvement on various benchmarks. Furthermore, when using a 34B LLM, the model shows emergent zero-shot Chinese capability, given that only English multimodal data are used during training. Lu *et al.* [62] see a similar phenomenon by scaling up LLMs from 13B to 35B and 65B/70B, where the larger model size brings consistent gains on benchmarks specifically designed for MLLMs. There are also works that use smaller LLMs to facilitate deployment on mobile devices. For example, MobileVLM series [63], [64] use downscaled LLaMA [5] (termed as MobileLLaMA 1.4B/2.7B), enabling efficient inference on mobile processors.

Recently, explorations of Mixture of Experts (MoE) architecture for LLMs have garnered rising attention [65], [66], [67]. Compared with dense models, the sparse architecture enables scaling up total parameter size without increasing computational cost, by selective activation of the parameters. Empirically, MM1 [52] and MoE-LLaVA [68] find that MoE implementation achieves better performance than the dense counterpart on almost all the benchmarks.

TABLE 2: A summary of commonly used open-sourced LLMs. en, zh, fr, and de stand for English, Chinese, French, and German, respectively.

Model	Release Date	Pretrain Data Scale	Parameter Size (B)	Language Support	Architecture
Flan-T5-XL/XXL [56]	Oct-2022	-	3/ 11	en, fr, de	Encoder-Decoder
LLaMA [5]	Feb-2023	1.4T tokens	7/ 13/ 33/ 65	en	Causal Decoder
Vicuna [4]	Mar-2023	1.4T tokens	7/ 13/ 33	en	Causal Decoder
LLaMA-2 [57]	Jul-2023	2T tokens	7/ 13/ 70	en	Causal Decoder
Qwen [58]	Sep-2023	3T tokens	1.8 / 7/ 14/ 72	en, zh	Causal Decoder

2.3 Modality interface

Since LLMs can only perceive text, bridging the gap between natural language and other modalities is necessary. However, it would be costly to train a large multimodal model in an end-to-end manner. A more practical way is to introduce a learnable connector between the pre-trained visual encoder and LLM. The other approach is to translate images into languages with the help of expert models, and then send the language to LLM.

Learnable Connector. It is responsible for bridging the gap between different modalities. Specifically, the module projects information into the space that LLM can understand efficiently. Based on how multimodal information is fused, there are broadly two ways to implement such interfaces, *i.e.* token-level and feature-level fusion.

For token-level fusion, features output from encoders are transformed into tokens and concatenated with text tokens before being sent into LLMs. A common and feasible solution is to leverage a group of learnable query tokens to extract information in a query-based manner [69], which first has been implemented in BLIP-2 [59], and subsequently inherited by a variety of work [26], [60], [70]. Such Q-Former-style approaches compress visual tokens into a smaller number of representation vectors. In contrast, some methods simply use a MLP-based interface to bridge the modality gap [20], [37], [71], [72]. For example, LLaVA series adopts one/two linear MLP [20], [50] to project visual tokens and align the feature dimension with word embeddings.

On a related note, MM1 [52] has ablated on design choices on the connector and found that for token-level fusion, the type of modality adapter is far less important than the number of visual tokens and input resolution. Nevertheless, Zeng *et al.* [73] compare the performance of token and feature-level fusion, and empirically reveal that the token-level fusion variant performs better in terms of VQA benchmarks. Regarding the performance gap, the authors suggest that cross-attention models might require a more complicated hyper-parameter searching process to achieve comparable performance.

As another line, feature-level fusion inserts extra modules that enable deep interaction and fusion between text features and visual features. For example, Flamingo [74] inserts extra cross-attention layers between frozen Transformer layers of LLMs, thereby augmenting language features with external visual cues. Similarly, CogVLM [75] plugs in a visual expert module in each Transformer layer to enable dual interaction and fusion between vision and language features. For better performance, the QKV weight matrix of the introduced module is initialized from the pre-trained LLM. Similarly, LLaMA-Adapter [76] introduces learnable prompts into Transformer layers. These prompts are first embedded with visual knowledge and then concatenated with text features as prefixes.

In terms of parameter size, learnable interfaces generally comprise a small portion compared with encoders and LLMs. Take Qwen-VL [34] as an example, the parameter size of the Q-Former is about 0.08B, accounting for less than 1% of the whole parameters, while the encoder and the LLM account for about 19.8% (1.9B) and 80.2% (7.7B), respectively.

Expert Model. Apart from the learnable interface, using expert models, such as an image captioning model, is also a feasible way to bridge the modality gap [77], [78], [79], [80]. The basic idea is to convert multimodal inputs into languages without training. In this way, LLMs can understand multimodality by the converted languages. For example, VideoChat-Text [25] uses pre-trained vision models to extract visual information such as actions and enriches the descriptions using a speech recognition model. Though using expert models is straightforward, it may not be as flexible as adopting a learnable interface. The conversion of foreign modalities into text would cause information loss. For example, transforming videos into textual descriptions distorts spatial-temporal relationships [25].

3 Training Strategy and Data

A full-fledged MLLM undergoes three stages of training, *i.e.* pre-training, instruction-tuning, and alignment tuning. Each phase of training requires different types of data and fulfills different objectives. In this section, we discuss training objectives, as well as data collection and characteristics for each training stage.

3.1 Pre-training

3.1.1 Training Detail

As the first training stage, pre-training mainly aims to align different modalities and learn multimodal world knowledge. Pre-training stage generally entails large-scale text-paired data, *e.g.* caption data. Typically, the caption pairs describe images/audio/videos in natural language sentences.

Here, we consider a common scenario where MLLMs are trained to align vision with text. As illustrated in Table 3, given an image, the model is trained to predict autoregressively the caption of the image, following a standard cross-entropy loss. A common approach for pre-training is to keep pre-trained modules (*e.g.* visual encoders and LLMs) frozen and train a learnable interface [20], [35], [72]. The idea is to align different modalities without losing pre-trained knowledge. Some methods [34], [81], [82] also unfreeze more modules (*e.g.* visual encoder) to enable more trainable parameters for alignment. It should be noted that

Input: <image>
Response: {caption}

TABLE 3: A simplified template to structure the caption data. {<image>} is the placeholder for the visual tokens, and {caption} is the caption for the image. Note that only the part marked in red is used for loss calculation.

the training scheme is closely related to the data quality. For short and noisy caption data, a lower resolution (*e.g.* 224) can be adopted to speed up the training process, while for longer and cleaner data, it is better to utilize higher resolutions (*e.g.* 448 or higher) to mitigate hallucinations. Besides, ShareGPT4V [83] finds that with high-quality caption data in the pretraining stage, unlocking the vision encode promotes better alignment.

3.1.2 Data

Pretraining data mainly serve two purposes, *i.e.* (1) aligning different modalities and (2) providing world knowledge. The pretraining corpora can be divided into coarse-grained and fine-grained data according to granularities, which we will introduce sequentially. We summarize commonly used pretraining datasets in Table 4.

Coarse-grained caption data share some typical traits in common: (1) The data volume is large since samples are generally sourced from the internet. (2) Because of the web-scrawled nature, the captions are usually short and noisy since they originate from the alt-text of the web images. These data can be cleaned and filtered via automatic tools, for example, using CLIP [13] model to filter out image-text pairs whose similarities are lower than a pre-defined threshold. In what follows, we introduce some representative coarse-grained datasets.

CC. CC-3M [84] is a web-scale caption dataset of 3.3M image-caption pairs, where the raw descriptions are derived from alt-text associated with images. The authors design a complicated pipeline to clean data: (1) For images, those with inappropriate content or aspect ratio are filtered. (2) For text, NLP tools are used to obtain text annotations, with samples filtered according to the designed heuristics. (3) For image-text pairs, images are assigned labels via classifiers. If text annotations do not overlap with image labels, the corresponding samples are dropped.

CC-12M [85] is a following work of CC-3M and contains 12.4M image-caption pairs. Compared with the previous work, CC-12M relaxes and simplifies the data-collection pipeline, thus collecting more data.

SBU Captions [86]. It is a captioned photo dataset containing 1M image-text pairs, with images and descriptions sourced from Flickr. Specifically, an initial set of images is acquired by querying the Flickr website with a large number of query terms. The descriptions attached to the images thus serve as captions. Then, to ensure that descriptions are relevant to the images, the retained images fulfill these requirements: (1) Descriptions of the images are of satisfactory length, decided by observation. (2) Descriptions of the images contain at least 2 words in the predefined term lists and a propositional word (*e.g.* "on", "under") that generally suggests spatial relationships.

TABLE 4: Common datasets used for pre-training.

Dataset	Samples	Date
Coarse-grained Image-Text		
CC-3M [84]	3.3M	2018
CC-12M [85]	12.4M	2020
SBU Captions [86]	1M	2011
LAION-5B [87]	5.9B	Mar-2022
LAION-2B [87]	2.3B	Mar-2022
LAION-COCO [88]	600M	Sep-2022
COYO-700M [90]	747M	Aug-2022
Fine-grained Image-Text		
ShareGPT4V-PT [83]	1.2M	Nov-2023
LVIS-Instruct4V [91]	111K	Nov-2023
ALLaVA [92]	709K	Feb-2024
Video-Text		
MSR-VTT [93]	200K	2016
Audio-Text		
WavCaps [94]	24K	Mar-2023

LAION. This series are large web-scale datasets, with images scrawled from the internet and associated alt-text as captions. To filter the image-text pairs, the following steps are performed: (1) Text with short lengths or images with too small or too big sizes are dropped. (2) Image deduplication based on URL. (3) Extract CLIP [13] embeddings for images and text, and use the embeddings to drop possibly illegal content and image-text pairs with low cosine similarity between embeddings. Here we offer a brief summary of some typical variants:

- LAION-5B [87]: It is a research-purpose dataset of 5.85B image-text pairs. The dataset is multilingual with a 2B English subset.
- LAION-COCO [88]: It contains 600M images extracted from the English subset of LAION-5B. The captions are synthetic, using BLIP [89] to generate various image captions and using CLIP [13] to pick the best fit for the image.

COYO-700M [90]. It contains 747M image-text pairs, which are extracted from CommonCrawl. For data filtering, the authors design the following strategies: (1) For images, those with inappropriate size, content, format, or aspect ratio are filtered. Moreover, the images are filtered based on the pHash value to remove images overlapped with public datasets such as ImageNet and MS-COCO. (2) For text, only English text with satisfactory length, noun forms, and appropriate words are saved. Whitespace before and after the sentence will be removed, and consecutive whitespace characters will be replaced with a single whitespace. Moreover, text appearing more than 10 times (*e.g.* "image for") will be dropped. (3) For image-text pairs, duplicated samples are removed based on (image pHash, text) tuple.

Recently, more works [83], [91], [92] have explored generating high-quality fine-grained data through prompting strong MLLMs (*e.g.* GPT-4V). Compared with coarse-grained data, these data generally contain longer and more accurate descriptions of the images, thus enabling finer-grained alignment between image and text modalities. However, since this approach generally requires calling commercial-use MLLMs, the cost is higher, and the data volume is relatively smaller. Notably, ShareGPT4V [83] strikes a balance by first training a captioner with GPT-4V-generated

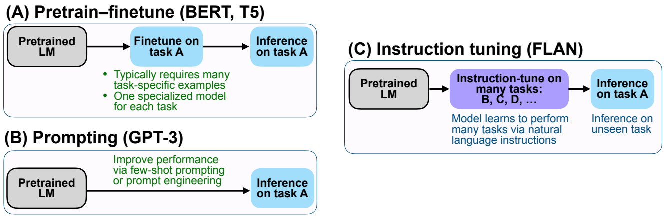
(A) Pretrain–finetune (BERT, T5)
Pretrained LM
Finetune on task A
Inference on task A
• Typically requires many task-specific examples
• One specialized model for each task
(B) Prompting (GPT-3)
Pretrained LM
Improve performance via few-shot prompting or prompt engineering
Inference on task A
(C) Instruction tuning (FLAN)
Pretrained LM
Instruction-tune on many tasks: B, C, D, ...
Model learns to perform many tasks via natural language instructions
Inference on task A
Inference on unseen task

Fig. 3: Comparison of three typical learning paradigms. The image is from [19].

100K data, then scaling up the data volume to 1.2M using the pre-trained captioner.

3.2 Instruction-tuning

3.2.1 Introduction

Instruction refers to the description of tasks. Intuitively, instruction tuning aims to teach models to better understand the instructions from users and fulfill the demanded tasks. Tuning in this way, LLMs can generalize to unseen tasks by following new instructions, thus boosting zero-shot performance. This simple yet effective idea has sparked the success of subsequent NLP works, such as ChatGPT [2], InstructGPT [95], FLAN [19], [56], and OPT-IML [96].

The comparisons between instruction tuning and related typical learning paradigms are illustrated in Fig. 3. The supervised fine-tuning approach usually requires a large amount of task-specific data to train a task-specific model. The prompting approach reduces the reliance on large-scale data and can fulfill a specialized task via prompt engineering. In such a case, though the few-shot performance has been improved, the zero-shot performance is still quite average [7]. Differently, instruction tuning learns how to generalize to unseen tasks rather than fitting specific tasks like the two counterparts. Moreover, instruction tuning is highly related to multi-task prompting [97].

In this section, we delineate the format of instruction samples, the training objectives, typical ways to gather instruction data, and corresponding commonly used datasets.

3.2.2 Training Detail

A multimodal instruction sample often includes an optional instruction and an input-output pair. The instruction is typically a natural language sentence describing the task, such as, "*Describe the image in detail.*" The input can be an image-text pair like the VQA task [99] or only an image like the image caption task [100]. The output is the answer to the instruction conditioned on the input. The instruction template is flexible and subject to manual designs [20], [25], [98], as exemplified in Table 5. Note that the instruction template can also be generalized to the case of multi-round conversations [20], [37], [71], [98].

> Below is an instruction that describes a task. Write a response that appropriately completes the request
>
> Instruction: <instruction>
> Input: {<image>, <text>}
> Response: <output>

TABLE 5: A simplified template to structure the multimodal instruction data. <instruction> is a textual description of the task. {<image>, <text>} and <output> are input and output from the data sample. Note that <text> in the input may be missed for some datasets, such as image caption datasets merely have <image>. The example is adapted from [98].

Formally, a multimodal instruction sample can be denoted in a triplet form, *i.e.* $(\mathcal{I}, \mathcal{M}, \mathcal{R})$, where $\mathcal{I}, \mathcal{M}, \mathcal{R}$ represent the instruction, the multimodal input, and the ground truth response, respectively. The MLLM predicts an answer given the instruction and the multimodal input:

$$\mathcal{A} = f(\mathcal{I}, \mathcal{M}; \theta) \tag{1}$$

Here, $\mathcal{A}$ denotes the predicted answer, and θ are the parameters of the model. The training objective is typically the original auto-regressive objective used to train LLMs [20], [37], [71], [101], based on which the MLLM is encouraged to predict the next token of the response. The objective can be expressed as:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(\mathcal{R}_i | \mathcal{I}, \mathcal{R}_{<i}; \theta) \tag{2}$$

where N is the length of the ground-truth response.

3.2.3 Data Collection

Since instruction data are more flexible in formats and varied in task formulations, it is usually trickier and more costly to collect data samples. In this section, we summarize three typical ways to harvest instruction data at scale, *i.e.* data adaptation, self-instruction, and data mixture.

Data Adaptation. Task-specific datasets are rich sources of high-quality data. Hence, abundant works [60], [70], [76], [82], [101], [102], [103], [104] have utilized existing high-quality datasets to construct instruction-formatted datasets. Take the transformation of VQA datasets for an example, the original sample is an input-out pair where the input comprises an image and a natural language question, and the output is the textual answer to the question conditioned on the image. The input-output pairs of these datasets could naturally comprise the multimodal input and response of the instruction sample (see §3.2.2). The instructions, *i.e.* the descriptions of the tasks, can either derive from manual design or from semi-automatic generation aided by GPT. Specifically, some works [21], [35], [60], [70], [102], [105] hand-craft a pool of candidate instructions and sample one of them during training. We offer an example of instruction templates for the VQA datasets as shown in Table 6. The other works manually design some seed instructions and use these to prompt GPT to generate more [25], [82], [98].

Note that since the answers of existing VQA and caption datasets are usually concise, directly using these datasets for instruction tuning may limit the output length of MLLMs. There are two common strategies to tackle this problem. The first one is to specify explicitly in instructions. For example, ChatBridge [104] explicitly declares *short* and *brief* for short-answer data, as well as *a sentence* and *single sentence* for conventional coarse-grained caption data. The second one is to extend the length of existing answers [105]. For example, M^{3}IT [105] proposes to rephrase the original answer by

- <Image> {Question}
- <Image> Question: {Question}
- <Image> {Question} A short answer to the question is
- <Image> Q: {Question} A:
- <Image> Question: {Question} Short answer:
- <Image> Given the image, answer the following question with no more than three words. {Question}
- <Image> Based on the image, respond to this question with a short answer: {Question}. Answer:
- <Image> Use the provided image to answer the question: {Question} Provide your answer as short as possible:
- <Image> What is the answer to the following question? "{Question}"
- <Image> The question "{Question}" can be answered using the image. A short answer is

TABLE 6: Instruction templates for VQA datasets, cited from [60]. <Image> and {Question} are the image and the question in the original VQA datasets, respectively.

TABLE 7: A summary of popular datasets generated by self-instruction. For input/output modalities, I: Image, T: Text, V: Video, A: Audio. For data composition, M-T and S-T denote multi-turn and single-turn, respectively.

Dataset	Sample	Modality	Source	Composition
LLaVA-Instruct	158K	I + T → T	MS-COCO	23K caption + 58K M-T QA + 77K reasoning
LVIS-Instruct	220K	I + T → T	LVIS	110K caption + 110K M-T QA
ALLaVA	1.4M	I + T → T	VFlan, LAION	709K caption + 709K S-T QA
Video-ChatGPT	100K	V + T → T	ActivityNet	7K description + 4K M-T QA
VideoChat	11K	V+T → T	WebVid	description + summarization + creation
Clotho-Detail	3.9K	A + T → T	Clotho	caption

prompting ChatGPT with the original question, answer, and contextual information of the image (*e.g.* caption and OCR).

Self-Instruction. Although existing multi-task datasets can contribute a rich source of data, they usually do not meet human needs well in real-world scenarios, such as multiple rounds of conversations. To tackle this issue, some works collect samples through self-instruction [106], which utilizes LLMs to generate textual instruction-following data using a few hand-annotated samples. Specifically, some instruction-following samples are hand-crafted as demonstrations, after which ChatGPT/GPT-4 is prompted to generate more instruction samples with the demonstrations as guidance. LLaVA [20] extends the approach to the multimodal field by translating images into text of captions and bounding boxes, and prompting text-only GPT-4 to generate new data with the guidance of requirements and demonstrations. In this way, a multimodal instruction dataset is constructed, called LLaVA-Instruct-150k. Following this idea, subsequent works such as MiniGPT-4 [21], ChatBridge [104], GPT4Tools [107], and DetGPT [72] develop different datasets catering for different needs. Recently, with the release of the more powerful multimodal model GPT-4V, many works have adopted GPT-4V to generate data of higher quality, as exemplified by LVIS-Instruct4V [91] and ALLaVA [92]. We summarize the popular datasets generated through self-instruction in Table 7.

Data Mixture. Apart from the multimodal instruction data, language-only user-assistant conversation data can also be used to improve conversational proficiencies and instruction-following abilities [81], [98], [101], [103]. LaVIN [101] directly constructs a minibatch by randomly sampling from both language-only and multimodal data. MultiInstruct [102] probes different strategies for training with a fusion of single modal and multimodal data, including mixed instruction tuning (combine both types of data and randomly shuffle) and sequential instruction tuning (text data followed by multimodal data).

3.2.4 *Data Quality*

Recent research has revealed that the data quality of instruction-tuning samples is no less important than quantity. Lynx [73] finds that models pre-trained on large-scale but noisy image-text pairs do not perform as well as models pre-trained with smaller but cleaner datasets. Similarly, Wei *et al.* [108] finds that less instruction-tuning data with higher quality can achieve better performance. For data filtering, the work proposes some metrics to evaluate data quality and, correspondingly, a method to automatically filter out inferior vision-language data. Here we discuss two important aspects regarding data quality.

Prompt Diversity. The diversity of instructions has been found to be critical for model performance. Lynx [73] empirically verifies that diverse prompts help improve model performance and generalization ability.

Task Coverage. In terms of tasks involved in training data, Du *et al.* [109] perform an empirical study and find that the visual reasoning task is superior to captioning and QA tasks for boosting model performance. Moreover, the study suggests that enhancing the complexity of instructions might be more beneficial than increasing task diversity and incorporating fine-grained spatial annotations.

3.3 Alignment tuning

3.3.1 *Introduction*

Alignment tuning is more often used in scenarios where models need to be aligned with specific human preferences, *e.g.* response with fewer hallucinations (see §6). Currently, Reinforcement Learning with Human Feedback (RLHF) and Direct Preference Optimization (DPO) are two main techniques for alignment tuning. In this section, we introduce the main ideas of the two techniques in sequence and offer some examples of how they are utilized in addressing practical problems, and finally, give a compilation of the related datasets.

3.3.2 Training Detail

RLHF [110], [111]. This technique aims to utilize reinforcement learning algorithms to align LLMs with human preferences, with human annotations as supervision in the training loop. As exemplified in InstructGPT [95], RLHF incorporates three key steps:

1) **Supervised fine-tuning.** This step aims to fine-tune a pre-trained model to present the preliminary desired output behavior. The fine-tuned model in the RLHF setting is called a *policy model*. Note that this step might be skipped since the supervised policy model π^{SFT} can be initialized from an instruction-tuned model (see §3.2).
2) **Reward modeling.** A *reward model* is trained using preference pairs in this step. Given a multimodal prompt (*e.g*. image and text) x and a response pair (y_w, y_l), the reward model r_θ learns to give a higher reward to the preferred response y_w, and vice versa for y_l, according to the following objective:

$$\mathcal{L}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log(\sigma(r_\theta(x,y_w) - r_\theta(x,y_l))\right] \quad (3)$$

where $\mathcal{D} = \{(x, y_w, y_l)\}$ is the comparison dataset labeled by human annotators. In practice, the reward model r_θ shares a similar structure with the policy model.
3) **Reinforcement learning.** In this step, the Proximal Policy Optimization (PPO) algorithm is adopted to optimize the RL policy model π_ϕ^{RL}. A per-token KL penalty is often added to the training objective to avoid deviating too far from the original policy [95], resulting in the objective:

$$\begin{aligned}\mathcal{L}(\phi) = -\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\phi^{RL}(y|x)}\Big[&r_\theta(x,y) \\ &- \beta\cdot\mathbb{D}_{KL}\Big(\pi_\phi^{RL}(y|x)||\pi^{REF}(y|x)\Big)\Big]\end{aligned} \quad (4)$$

where β is the coefficient for the KL penalty term. Typically, both the RL policy π_ϕ^{RL} and the reference model π^{REF} are initialized from the supervised model π^{SFT}. The obtained RL policy model is expected to align with human preferences through this tuning process.

Researchers have explored using the RLHF techniques for better multimodal alignment. For example, LLaVA-RLHF [112] collects human preference data and tunes a model with fewer hallucinations based on LLaVA [20].

DPO [113]. It learns from human preference labels utilizing a simple binary classification loss. Compared with the PPO-based RLHF algorithm, DPO is exempt from learning an explicit reward model, thus simplifying the whole pipeline to two steps, *i.e*. human preference data collection and preference learning. The learning objective is as follows:

$$\begin{aligned}\mathcal{L}(\phi) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[&\log\sigma\Big(\beta\log\frac{\pi_\phi^{\text{RL}}(y_w|x)}{\pi^{\text{REF}}(y_w|x)} \\ &- \beta\log\frac{\pi_\phi^{\text{RL}}(y_l|x)}{\pi^{\text{REF}}(y_l|x)}\Big)\Big]\end{aligned} \quad (5)$$

RLHF-V [114] collects fine-grained (segment-level) preference data pairs by correcting hallucinations in the model response and uses the obtained data to perform dense DPO. Silkie [115] instead collects preference data via prompting GPT-4V and distills the preference supervision into an instruction-tuned model through DPO.

TABLE 8: A summary of datasets for alignment-tuning. For input/output modalities, I: Image, T: Text.

Dataset	Sample	Modality	Source
LLaVA-RLHF [112]	10K	I + T → T	Human
RLHF-V [114]	5.7K	I + T → T	Human
VLFeedback [115]	380K	I + T → T	GPT-4V

3.3.3 Data

The gist of data collection for alignment-tuning is to collect feedback for model responses, *i.e*. to decide which response is better. It is generally more expensive to collect such data, and the amount of data used for this phase is typically even less than that used in previous stages. In this part, we introduce some datasets and summarize them in Table 8.

LLaVA-RLHF [112]. It contains 10K preference pairs collected from human feedback in terms of honesty and helpfulness. The dataset mainly serves to reduce hallucinations in model responses.

RLHF-V [114]. It has 5.7K fine-grained human feedback data collected by segment-level hallucination corrections.

VLFeedback [115]. It utilizes AI to provide feedback on model responses. The dataset contains more than 380K comparison pairs scored by GPT-4V in terms of helpfulness, faithfulness, and ethical concerns.

4 Evaluation

Evaluation is an essential part of developing MLLMs since it provides feedback for model optimization and helps to compare the performance of different models. Compared with evaluation methods of traditional multimodal models, the evaluation of MLLMs exhibits several new traits: (1) Since MLLMs are generally versatile, it is important to evaluate MLLMs comprehensively. (2) MLLMs exhibit many emergent capabilities that require special attention (*e.g*. OCR-free math reasoning) and thus require new evaluation schemes. The evaluation of MLLMs can be broadly categorized into two types according to the question genres, including closed-set and open-set.

4.1 Closed-set

Closed-set questions refer to a type of question where the possible answer options are predefined and limited to a finite set. The evaluation is usually performed on task-specific datasets. In this case, the responses can be naturally judged by benchmark metrics [20], [60], [70], [76], [101], [102], [103], [104]. For example, InstructBLIP [60] reports the accuracy on ScienceQA [116], as well as the CIDEr score [117] on NoCaps [118] and Flickr30K [119]. The evaluation settings are typically zero-shot [60], [102], [104], [105] or finetuning [20], [35], [60], [70], [76], [101], [103], [105]. The first setting often selects a wide range of datasets covering different general tasks and splits them into held-in and held-out datasets. After tuning on the former, zero-shot performance is evaluated on the latter with unseen datasets or even unseen tasks. In contrast, the second setting is often observed in the evaluation of domain-specific tasks. For example, LLaVA [20] and LLaMA-Adapter [76] report fine-tuned performance on ScienceQA [116]. LLaVA-Med [35] reports results on biomedical VQA [120], [121], [122].

The above evaluation methods are usually limited to a small range of selected tasks or datasets, lacking a comprehensive quantitative comparison. To this end, some efforts have endeavored to develop new benchmarks specially designed for MLLMs [123], [124], [125], [126], [127], [128], [129]. For example, Fu *et al*. [123] construct a comprehensive evaluation benchmark MME that includes a total of 14 perception and cognition tasks. All instruction-answer pairs in MME are manually designed to avoid data leakage. MMBench [124] is a benchmark specifically designed for evaluating multiple dimensions of model capabilities, using ChatGPT to match open responses with pre-defined choices. Video-ChatGPT [130] and Video-Bench [131] focus on video domains and propose specialized benchmarks as well as evaluation tools for assessment. There are also evaluation strategies designed to evaluate a specific aspect of the model [102], as exemplified by POPE [132] for assessment of hallucination degree.

4.2 Open-set

In contrast to the closed-set questions, the responses to open-set questions can be more flexible, where MLLMs usually play a chatbot role. Because the content of the chat can be arbitrary, it would be trickier to judge than the closed-ended output. The criterion can be classified into manual scoring, GPT scoring, and case study. Manual scoring requires humans to assess the generated responses. This kind of approach often involves hand-crafted questions that are designed to assess specific dimensions. For example, mPLUG-Owl [81] collects a visually related evaluation set to judge capabilities like natural image understanding, diagram, and flowchart understanding. Similarly, GPT4Tools [107] builds two sets for the finetuning and zero-shot performance, respectively, and evaluates the responses in terms of thought, action, arguments, and the whole.

Since manual assessment is labor intensive, some researchers have explored rating with GPT, namely GPT scoring. This approach is often used to evaluate performance on multimodal dialogue. LLaVA [20] proposes to score the responses via text-only GPT-4 in terms of different aspects, such as helpfulness and accuracy. Specifically, 30 images are sampled from the COCO [133] validation set, each associated with a short question, a detailed question, and a complex reasoning question via self-instruction on GPT-4. The answers generated by both the model and GPT-4 are sent to GPT-4 for comparison. Subsequent works follow this idea and prompt ChatGPT [81] or GPT-4 [35], [70], [101], [104], [105] to rate results [35], [70], [81], [101], [104] or judge which one is better [103].

A main issue of applying text-only GPT-4 as an evaluator is that the judge is only based on image-related text content, such as captions or bounding box coordinates, without accessing the image [35]. Thus, it may be questionable to set GPT-4 as the performance upper bound in this case. With the release of the vision interface of GPT, some works [77], [134] exploit a more advanced GPT-4V model to assess the performance of MLLMs. For example, Woodpecker [77] adopts GPT-4V to judge the response quality of model answers based on the image. The evaluation is expected to be more accurate than using text-only GPT-4 since GPT-4V has direct access to the image.

A supplementary approach is to compare the different capabilities of MLLMs through case studies. For instance, some studies evaluate two typical advanced commercial-use models, GPT-4V and Gemini. Yang *et al*. [135] perform in-depth qualitative analysis on GPT-4V by crafting a series of samples across various domains and tasks, spanning from preliminary skills, such as caption and object counting, to complex tasks that require world knowledge and reasoning, such as joke understanding and indoor navigation as an embodied agent. Wen *et al*. [136] make a more focused evaluation of GPT-4V by designing samples targeting automatic driving scenarios. Fu *et al*. [137] carry out a comprehensive evaluation on Gemini-Pro by comparing the model against GPT-4V. The results suggest that GPT-4V and Gemini exhibit comparable visual reasoning abilities in spite of different response styles.

5 Extensions

Recent studies have made significant strides in extending the capabilities of MLLMs, spanning from more potent foundational abilities to broader coverage of scenarios. We trace the principal development of MLLMs in this regard.

Granularity Support. To facilitate better interaction between agents and users, researchers have developed MLLMs with finer support of granularities in terms of model inputs and outputs. On the input side, models that support finer control from user prompts are developed progressively, evolving from image to region [28], [138], [139] and even pixels [29], [140], [141]. Specifically, Shikra [28] supports region-level input and understanding. Users may interact with the assistant more flexibly by referring to specific regions, which are represented in bounding boxes of natural language forms. Ferret [141] takes a step further and supports more flexible referring by devising a hybrid representation scheme. The model supports different forms of prompts, including point, box, and sketch. Similarly, Osprey [29] supports point input by utilizing a segmentation model [9]. Aided by the exceptional capabilities of the pre-trained segmentation model, Osprey enables specifying a single entity or part of it with a single click. On the output side, grounding capabilities are improved in line with the development of input support. Shikra [28] supports response grounded in the image with box annotations, resulting in higher precision and finer referring experience. LISA [142] further supports mask-level understanding and reasoning, which makes pixel-level grounding possible.

Modality Support. Increased support for modalities is a tendency for MLLM studies. On the one hand, researchers have explored adapting MLLMs to support the input of more multimodal content, such as 3D point cloud [41], [143], [144], [145]. On the other hand, MLLMs are also extended to generate responses of more modalities, such as image [32], [146], [147], [148], audio [32], [147], [149], [150], and video [32], [151]. For example, NExT-GPT [32] proposes a framework that supports inputs and outputs of mixed modalities, specifically, combinations of text, image, audio, and video, with the help of diffusion models [152], [153] attached to the MLLM. The framework applies an encoder-decoder architecture and puts LLM as a pivot for understanding and reasoning.

Language Support. Current models are predominantly unilingual, probably due to the fact that high-quality non-English training corpus is scarce. Some works have been devoted to developing multilingual models so that a broader range of users can be covered. VisCPM [33] transfers model capabilities to the multilingual setting by designing a multi-stage training scheme. Specifically, the scheme takes English as a pivotal language, with abundant training corpus. Utilizing a pre-trained bilingual LLM, the multimodal capabilities are transferred to Chinese by adding some translated samples during instruction tuning. Taking a similar approach, Qwen-VL [34] is developed from the bilingual LLM Qwen [58] and supports both Chinese and English. During pre-training, Chinese data is mixed into the training corpus to preserve the bilingual capabilities of the model, taking up 22.7% of the whole data volume.

Scenario/Task Extension. Apart from developing common general-purpose assistants, some studies have focused on more specific scenarios where practical conditions should be considered, while others extend MLLMs to downstream tasks with specific expertise.

A typical tendency is to adapt MLLMs to more specific real-life scenarios. MobileVLM [63] explores developing small-size variants of MLLMs for resource-limited scenarios. Some designs and techniques are utilized for deployment on mobile devices, such as LLMs of smaller size and quantization techniques to speed up computation. Other works develop agents that interact with real-world [41], [154], [155], *e.g.* user-friendly assistants specially designed for Graphical User Interface (GUI), as exemplified by CogAgent [44], AppAgent [43], and Mobile-Agent [45]. These assistants excel in planning and guiding through each step to fulfill a task specified by users, acting as helpful agents for human-machine interaction. Another line is to augment MLLMs with specific skills for solving tasks in different domains, *e.g.* document understanding [38], [39], [156], [157] and medical domains [35], [36], [37]. For document understanding, mPLUG-DocOwl [38] utilizes various forms of document-level data for tuning, resulting in an enhanced model in OCR-free document understanding. TextMonkey [39] incorporates multiple tasks related to document understanding to improve model performance. Apart from conventional document image and scene text datasets, position-related tasks are added to reduce hallucinations and help models learn to ground responses in the visual information. MLLMs can also be extended to medical domains by instilling knowledge of the medical domain. For example, LLaVA-Med [158] injects medical knowledge into vanilla LLaVA [20] and develops an assistant specialized in medical image understanding and question answering.

6 Multimodal Hallucination

Multimodal hallucination refers to the phenomenon of responses generated by MLLMs being inconsistent with the image content [77]. As a fundamental and important problem, the issue has received increased attention. In this section, we briefly introduce some related concepts and research development.

6.1 Preliminaries

Current research on multimodal hallucinations can be further categorized into three types [159]:

1) *Existence Hallucination* is the most basic form, meaning that models incorrectly claim the existence of certain objects in the image.
2) *Attribute Hallucination* means describing the attributes of certain objects in a wrong way, *e.g.* failure to identify a dog's color correctly. It is typically associated with existence hallucination since descriptions of the attributes should be grounded in objects present in the image.
3) *Relationship Hallucination* is a more complex type and is also based on the existence of objects. It refers to false descriptions of relationships between objects, such as relative positions and interactions.

In what follows, we first introduce some specific evaluation methods (§6.2), which are useful to gauge the performance of methods for mitigating hallucinations (§6.3). Then, we will discuss in detail the current methods for reducing hallucinations, according to the main categories each method falls into.

6.2 Evaluation Methods

CHAIR [160] is an early metric that evaluates hallucination levels in open-ended captions. The metric measures the proportion of sentences with hallucinated objects or hallucinated objects in all the objects mentioned. In contrast, POPE [132] is a method that evaluates closed-set choices. Specifically, multiple prompts with binary choices are formulated, each querying if a specific object exists in the image. The method also covers more challenging settings to evaluate the robustness of MLLMs, with data statistics taken into consideration. The final evaluation uses a simple watchword mechanism, *i.e.* by detecting keywords "yes/no", to convert open-ended responses into closed-set binary choices. With a similar evaluation approach, MME [123] provides a more comprehensive evaluation, covering aspects of existence, count, position and color, as exemplified in [77].

Different from previous approaches that use matching mechanisms to detect and decide hallucinations, HaELM [161] proposes using text-only LLMs as a judge to automatically decide whether MLLMs' captions are correct against reference captions. In light of the fact that text-only LLMs can only access limited image context and require reference annotations, Woodpecker [77] uses GPT-4V to directly assess model responses grounded in the image. FaithScore [162] is a more fine-grained metric based on a routine that breaks down descriptive sub-sentences and evaluates each sub-sentence separately. Based on previous studies, AMBER [163] is an LLM-free benchmark that encompasses both discriminative tasks and generative tasks and involves three types of possible hallucinations (see §6.1).

6.3 Mitigation Methods

According to high-level ideas, the current methods can be roughly divided into three categories: pre-correction, in-process-correction, and post-correction.

Pre-correction. An intuitive and straightforward solution for hallucination is to collect specialized data (*e.g.* negative data) and use the data for fine-tuning, thus resulting in models with fewer hallucinated responses.

LRV-Instruction [164] introduces a visual instruction tuning dataset. Apart from common positive instructions, the dataset incorporates delicately designed negative instructions at different semantic levels to encourage responses faithful to the image content. LLaVA-RLHF [112] collects human-preference pairs and finetunes models with reinforcement learning techniques, leading to models more aligned with less hallucinated answers.

In-process-correction. Another line is to make improvements in architectural design or feature representation. These works try to explore the reasons for hallucinations and design corresponding remedies to mitigate them in the generation process.

HallE-Switch [159] performs an empirical analysis of possible factors of object existence hallucinations and hypothesizes that existence hallucinations derive from objects not grounded by visual encoders, and they are actually inferred based on knowledge embedded in the LLM. Based on the assumption, a continuous controlling factor and corresponding training scheme are introduced to control the extent of imagination in model output during inference.

VCD [165] suggests that object hallucinations derive from two primary causes, *i.e.* statistical bias in training corpus and strong language prior embedded in LLMs. The authors take notice of the phenomenon that when injecting noise into the image, MLLMs tend to lean towards language prior rather than the image content for response generation, leading to hallucinations. Correspondingly, this work designs an amplify-then-contrast decoding scheme to offset the false bias.

HACL [166] investigates the embedding space of vision and language. Based on the observation, a contrastive learning scheme is devised to pull paired cross-modal representation closer while pushing away non-hallucinated and hallucinated text representation.

Post-correction. Different from previous paradigms, post-correction mitigates hallucinations in a post-remedy way and corrects hallucinations after output generation. Woodpecker [77] is a training-free general framework for hallucination correction. Specifically, the method incorporates expert models to supplement contextual information of the image and crafts a pipeline to correct hallucinations step by step. The method is interpretable in that intermediate results of each step can be checked, and objects are grounded in the image. The other method LURE [167] trains a specialized revisor to mask objects with high uncertainty in the descriptions and regenerates the responses again.

7 Extended Techniques

7.1 Multimodal In-Context Learning

ICL is one of the important emergent abilities of LLMs. There are two good traits of ICL: (1) Different from traditional supervised learning paradigms that learn implicit patterns from abundant data, the crux of ICL is to learn from analogy [168]. Specifically, in the ICL setting, LLMs learn from a few examples along with an optional instruction and extrapolate to new questions, thereby solving complex and unseen tasks in a few-shot manner [22], [169], [170]. (2) ICL is usually implemented in a training-free manner [168] and thus can be flexibly integrated into different frameworks at the inference stage. A closely related technique to ICL is instruction-tuning (see §3.2), which is shown empirically to enhance the ICL ability [19].

```
<BOS> Below are some examples and an instruction that
describes a task. Write a response that appropriately completes
the request

### Instruction: {instruction}
### Image: <image>
### Response: {response}

### Image: <image>
### Response: {response}
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
### Image: <image>
### Response: <EOS>
```

TABLE 9: A simplified example of the template to structure an M-ICL query, adapted from [98]. For illustration, we list two in-context examples and a query divided by a dashed line. {instruction} and {response} are texts from the data sample. <image> is a placeholder to represent the multimodal input (an image in this case). <BOS> and <EOS> are tokens denoting the start and the end of the input to the LLM, respectively.

In the context of MLLM, ICL has been extended to more modalities, leading to Multimodal ICL (M-ICL). Building upon the setting in (§3.2), at inference time, M-ICL can be implemented by adding a demonstration set, *i.e.* a set of in-context samples, to the original sample. In this case, the template can be extended as illustrated in Table 9. Note that we list two in-context examples for illustration, but the number and the ordering of examples can be flexibly adjusted. In fact, models are commonly sensitive to the arrangement of demonstrations [168], [171].

7.1.1 Improvement on ICL capabilities

Recently, a growing amount of work has focused on enhancing ICL performance under various scenarios. In this section, we trace the development of this field and summarize some relevant works.

MIMIC-IT [172] combines in-context learning with instruction tuning by building an instruction dataset formatted with multimodal context. The model instruction tuned on the introduced dataset shows improved few-shot performance in the caption task. Emu [173] extends the idea of Flamingo [74] by introducing extra modalities in model generation and corresponding training corpus. Aided by the introduced vision decoder, *i.e.* Stable Diffusion, the model learns from extra vision supervision and supports more flexibility in output format and in-context reasoning. Specifically, apart from answering in pure text, the model can also give responses in the form of images. Sheng *et* *al*. [174] adopt a similar idea and try to extend output modalities into both text and image. Instead of adopting a specialized encoder for images, the work adopts a unified quantization scheme with a shared embedding layer.

Some other works explore improving few-shot learning performance under specific settings. Link-context learning [175] focuses on strengthening the causal link between image-label pairs and casts a contrast training scheme by formulating positive and negative image-description pairs. MMICL [176] aims to augment the capabilities in reasoning with multiple related images. To strengthen the link between image and text, the work proposes a context scheme to transform interleaved image-text data into a uniform format. Jeong [177] finds that when inserting a small fraction of incoherent images/text as noise, MLLMs can be misled to give responses inconsistent with the context. Based on the observation, the work accordingly proposes a pre-filtering method to remove irrelevant context and facilitate more coherent responses.

7.1.2 Applications

In terms of applications in multimodality, M-ICL is mainly used in two scenarios: (1) solving various visual reasoning tasks [22], [74], [178], [179], [180] and (2) teaching LLMs to use external tools [169], [170], [181]. The former usually involves learning from a few task-specific examples and generalizing to a new but similar question. From the information provided in instructions and demonstrations, LLMs get a sense of what the task is doing and what the output template is and finally generate expected answers. In contrast, examples of tool usage are more fine-grained. They typically comprise a chain of steps that could be sequentially executed to fulfill the task. Thus, the second scenario is closely related to CoT (see §7.2).

7.2 Multimodal Chain of Thought

As the pioneer work [8] points out, CoT is "a series of intermediate reasoning steps", which has been proven to be effective in complex reasoning tasks [8], [182], [183]. The main idea of CoT is to prompt LLMs to output not only the final answer but also the reasoning process that leads to the answer, resembling the cognitive process of humans.

Inspired by the success in NLP, multiple works [184], [185], [186], [187] have been proposed to extend the unimodal CoT to Multimodal CoT (M-CoT). We first introduce different paradigms for acquiring the M-CoT ability (§7.2.1). Then, we delineate more specific aspects of M-CoT, including the chain configuration (§7.2.2) and the pattern (§7.2.3).

7.2.1 Learning Paradigms

The learning paradigm is also an aspect worth investigating. There are broadly three ways to acquire the M-CoT ability, *i.e.* through finetuning and training-free few/zero-shot learning. The sample size requirement for the three ways is in descending order.

Intuitively, the finetuning approach often involves curating specific datasets for M-CoT learning. For example, Lu *et al*. [116] construct a scientific question-answering dataset ScienceQA with lectures and explanations, which can serve as sources of learning CoT reasoning, and finetune the model on this proposed dataset. Multimodal-CoT [185] also uses the ScienceQA benchmark but generates the output in a two-step fashion, *i.e.* the rationale (chain of reasoning steps) and the final answer based on the rationale. CoT-PT [187] learns an implicit chain of reasoning through a combination of prompt tuning and step-specific visual bias.

Compared with finetuning, few/zero-shot learning is more computationally efficient. The main difference between them is that the few-shot learning typically requires hand-crafting some in-context examples so that the model can learn to reason step by step more easily. In contrast, the zero-shot learning does not require any specific example for CoT learning. In this case, models learn to use the embedded knowledge and the reasoning ability without explicit guidance by prompting designed instructions like "Let's think frame by frame" or "What happened between these two keyframes" [184], [186]. Similarly, some works [22], [188] prompt models with descriptions of the task and tool usage to decompose complex tasks into sub-tasks.

7.2.2 Chain Configuration

Structure and length are two critical aspects of the reasoning chains. In terms of structure, current methods can be divided into single-chain and tree-shape methods. Reasoning with a single chain is a paradigm widely used in various methods [116], [185]. Specifically, the step-by-step reasoning process forms a single question-rationale-answer chain. Recently, some methods have explored using a more complicated scheme, *i.e.* tree-shape chain, for reasoning. Specifically, DDCoT [189] breaks down a question into multiple sub-questions, each of which is solved by LLM itself or visual experts to generate rationales. Then the LLM aggregates and reasons with the rationales to form the final answer. With respect for chain length, it can be categorized into adaptive and pre-defined formations. The former configuration requires LLMs to decide on their own when to halt the reasoning chains [22], [116], [169], [170], [185], [188], while the latter setting stops the chains with a pre-defined length [79], [184], [186], [187].

7.2.3 Generation Patterns

How the chain is constructed is a question worth studying. We summarize the current works into (1) an infilling-based pattern and (2) a predicting-based pattern. Specifically, the infilling-based pattern demands deducing steps between surrounding context (previous and following steps) to fill the logical gaps [184], [186]. In contrast, the predicting-based pattern requires extending the reasoning chains given conditions such as instructions and previous reasoning history [22], [116], [169], [170], [185], [188]. The two types of patterns share a requirement that the generated steps should be consistent and correct.

7.3 LLM-Aided Visual Reasoning

7.3.1 Introduction

Inspired by the success of tool-augmented LLMs [190], [191], [192], [193], some researches have explored the possibilities of invoking external tools [22], [107], [169], [170] or vision foundation models [22], [79], [80], [188], [194], [195], [196] for visual reasoning tasks. Taking LLMs as helpers with different roles, these works build task-specific [79], [197], [198] or general-purpose [22], [169], [170], [181], [188] visual reasoning systems.

Compared with conventional visual reasoning models [199], [200], [201], these works manifest several good traits: (1) Strong generalization abilities. Equipped with rich open-world knowledge learned from large-scale pretraining, these systems can easily generalize to unseen objects or concepts with remarkable zero/few-shot performance [169], [170], [195], [197], [198], [202]. (2) Emergent abilities. Aided by strong reasoning abilities of LLMs, these systems can perform complex tasks. For example, given an image, MM-REACT [22] can interpret the meaning beneath the surface, such as explaining why a meme is funny. (3) Better interactivity and control. Traditional models typically allow a limited set of control mechanisms and often entail expensive curated datasets [203], [204]. In contrast, LLM-based systems have the ability to make fine control in a user-friendly interface (*e.g.* click and natural language queries) [79].

For this part, we start with introducing different training paradigms employed in the construction of LLM-Aided Visual Reasoning systems (§7.3.2). Then, we delve into the primary roles that LLMs play within these systems (§7.3.3).

7.3.2 Training Paradigms

According to training paradigms, LLM-Aided Visual Reasoning systems can be divided into two types, *i.e.* training-free and finetuning.

Training-free. With abundant prior knowledge stored in pre-trained LLMs, an intuitive and simple way is to freeze pre-trained models and directly prompt LLMs to fulfill various needs. According to the setting, the reasoning systems can be further categorized into few-shot models [22], [169], [170], [181] and zero-shot models [79], [197]. The few-shot models entail a few hand-crafted in-context samples (see §7.1) to guide LLMs to generate a program or a sequence of execution steps. These programs or execution steps serve as instructions for corresponding foundation models or external tools/modules. The zero-shot models take a step further by directly utilizing LLMs' linguistics/semantics knowledge or reasoning abilities. For example, PointCLIP V2 [197] prompts GPT-3 to generate descriptions with 3D-related semantics for better alignment with corresponding images. In CAT [79], LLMs are instructed to refine the captions according to user queries.

Finetuning. Some works adopt further finetuning to improve the planning abilities with respect to tool usage [107] or to improve localization capabilities [142], [205] of the system. For example, GPT4Tools [107] introduces the instruction-tuning approach (see §3.2). Accordingly, a new tool-related instruction dataset is collected and used to finetune the model.

7.3.3 Functions

In order to further inspect what roles LLMs exactly play in LLM-Aided Visual Reasoning systems, existing related works are divided into three types:

- LLM as a Controller
- LLM as a Decision Maker
- LLM as a Semantics Refiner

The first two roles are related to CoT (see §7.2). It is frequently used because complex tasks need to be broken down into intermediate simpler steps. When LLMs act as controllers, the systems often finish the task in a single round, while multi-round is more common in the case of the decision maker. We delineate how LLMs serve these roles in the following parts.

LLM as a Controller. In this case, LLMs act as a central controller that (1) breaks down a complex task into simpler sub-tasks/steps and (2) assigns these tasks to appropriate tools/modules. The first step is often finished by leveraging the CoT ability of LLMs. Specifically, LLMs are prompted explicitly to output task planning [181] or, more directly, the modules to call [107], [169], [170]. For example, VisProg [170] prompts GPT-3 to output a visual program, where each program line invokes a module to perform a sub-task. In addition, LLMs are required to output argument names for the module input. To handle these complex requirements, some hand-crafted in-context examples are used as references [169], [170], [181]. This is closely related to the optimization of reasoning chains (see §7.2), or more specifically, the least-to-most prompting [206] technique. In this way, complex problems are broken down into sub-problems that are solved sequentially.

LLM as a Decision Maker. In this case, complex tasks are solved in a multi-round manner, often in an iterative way [195]. Decision-makers often fulfill the following responsibilities: (1) Summarize the current context and the history information, and decide if the information available at the current step is sufficient to answer the question or complete the task; (2) Organize and summarize the answer to present it in a user-friendly way.

LLM as a Semantics Refiner. When LLM is used as a Semantics Refiner, researchers mainly utilize its rich linguistics and semantics knowledge. Specifically, LLMs are often instructed to integrate information into consistent and fluent natural language sentences [202] or generate texts according to different specific needs [79], [197], [198].

8 Challenges and Future Directions

The development of MLLMs is still in a rudimentary stage and thus leaves much room for improvement, which we summarize below:

- Current MLLMs are limited in processing multimodal information of long context. This restricts the development of advanced models with more multimodal tokens, *e.g.* long-video understanding, and long documents interleaved with images and text.
- MLLMs should be upgraded to follow more complicated instructions. For example, a mainstream approach to generating high-quality question-answer pair data is still prompting closed-source GPT-4V because of its advanced instruction-following capabilities, while other models generally fail to achieve.
- There is still a large space for improvement in techniques like M-ICL and M-CoT. Current research on the two techniques is still rudimentary, and the related capabilities of MLLMs are weak. Thus, explorations of the underlying mechanisms and potential improvement are promising.

- Developing embodied agents based on MLLMs is a heated topic. It would be meaningful to develop such agents that can interact with the real world. Such endeavors require models with critical capabilities, including perception, reasoning, planning, and execution.
- Safety issues. Similar to LLMs, MLLMs can be vulnerable to crafted attacks [177], [207], [208]. In other words, MLLMs can be misled to output biased or undesirable responses. Thus, improving model safety will be an important topic.

9 Conclusion

In this paper, we perform a survey of the existing MLLM literature and offer a broad view of its main directions, including the basic recipe and related extensions. Moreover, we underscore the current research gaps that need to be filled and point out some promising research directions. We hope this survey can offer readers a clear picture of the current progress of MLLM and inspire more works.

References

[1] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv:2303.18223*, 2023. 1

[2] OpenAI, "Chatgpt: A language model for conversational ai," OpenAI, Tech. Rep., 2023. [Online]. Available: https://www.openai.com/research/chatgpt 1, 6

[3] ——, "Gpt-4 technical report," *arXiv:2303.08774*, 2023. 1

[4] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90% chatgpt quality," 2023. [Online]. Available: https://vicuna.lmsys.org 1, 3, 4

[5] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv:2302.13971*, 2023. 1, 3, 4

[6] B. Peng, C. Li, P. He, M. Galley, and J. Gao, "Instruction tuning with gpt-4," *arXiv:2304.03277*, 2023. 1

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *NeurIPS*, 2020. 1, 3, 6

[8] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," *arXiv:2201.11903*, 2022. 1, 12

[9] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv:2304.02643*, 2023. 1, 9

[10] Y. Shen, C. Fu, P. Chen, M. Zhang, K. Li, X. Sun, Y. Wu, S. Lin, and R. Ji, "Aligning and prompting everything all at once for universal visual perception," in *CVPR*, 2024. 1

[11] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv:2203.03605*, 2022. 1

[12] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv:2304.07193*, 2023. 1

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021. 1, 2, 3, 5

[14] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," *NeurIPS*, 2021. 1

[15] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," in *ECCV*, 2020. 1

[16] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," in *ICML*, 2022. 1

[17] J. Cho, J. Lei, H. Tan, and M. Bansal, "Unifying vision-and-language tasks via text generation," in *ICML*, 2021. 1

[18] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao, "Simvlm: Simple visual language model pretraining with weak supervision," *arXiv:2108.10904*, 2021. 1

[19] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," *arXiv:2109.01652*, 2021. 1, 6, 11

[20] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv:2304.08485*, 2023. 1, 4, 6, 7, 8, 9, 10

[21] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv:2304.10592*, 2023. 1, 2, 6, 7

[22] Z. Yang, L. Li, J. Wang, K. Lin, E. Azarnasab, F. Ahmed, Z. Liu, C. Liu, M. Zeng, and L. Wang, "Mm-react: Prompting chatgpt for multimodal reasoning and action," *arXiv:2303.11381*, 2023. 1, 11, 12, 13

[23] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language model," *arXiv:2303.03378*, 2023. 1

[24] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa *et al.*, "Openflamingo: An open-source framework for training large autoregressive vision-language models," *arXiv:2308.01390*, 2023. 1

[25] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao, "Videochat: Chat-centric video understanding," *arXiv:2305.06355*, 2023. 1, 4, 6

[26] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," *arXiv:2306.02858*, 2023. 1, 4

[27] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang, "Pengi: An audio language model for audio tasks," *NeurIPS*, 2024. 1, 3

[28] K. Chen, Z. Zhang, W. Zeng, R. Zhang, F. Zhu, and R. Zhao, "Shikra: Unleashing multimodal llm's referential dialogue magic," *arXiv:2306.15195*. 1, 9

[29] Y. Yuan, W. Li, J. Liu, D. Tang, X. Luo, C. Qin, L. Zhang, and J. Zhu, "Osprey: Pixel understanding with visual instruction tuning," *arXiv:2312.10032*. 1, 2, 9

[30] J. Han, R. Zhang, W. Shao, P. Gao, P. Xu, H. Xiao, K. Zhang, C. Liu, S. Wen, Z. Guo *et al.*, "Imagebind-llm: Multi-modality instruction tuning," *arXiv:2309.03905*, 2023. 1, 3

[31] S. Moon, A. Madotto, Z. Lin, T. Nagarajan, M. Smith, S. Jain, C.-F. Yeh, P. Murugesan, P. Heidari, Y. Liu *et al.*, "Anymal: An efficient and scalable any-modality augmented language model," *arXiv:2309.16058*, 2023. 1

[32] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua, "Next-gpt: Any-to-any multimodal llm," *arXiv:2309.05519*, 2023. 1, 9

[33] J. Hu, Y. Yao, C. Wang, S. Wang, Y. Pan, Q. Chen, T. Yu, H. Wu, Y. Zhao, H. Zhang *et al.*, "Large multilingual models pivot zero-shot multimodal learning across languages," *arXiv:2308.12038*, 2023. 1, 10

[34] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A frontier large vision-language model with versatile abilities," *arXiv:2308.12966*, 2023. 1, 3, 4, 10

[35] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao, "Llava-med: Training a large language-and-vision assistant for biomedicine in one day," *arXiv:2306.00890*, 2023. 1, 4, 6, 8, 9, 10

[36] M. Moor, Q. Huang, S. Wu, M. Yasunaga, Y. Dalmia, J. Leskovec, C. Zakka, E. P. Reis, and P. Rajpurkar, "Med-flamingo: a multi-modal medical few-shot learner," in *Machine Learning for Health (ML4H)*, 2023. 1, 10

[37] X. Zhang, C. Wu, Z. Zhao, W. Lin, Y. Zhang, Y. Wang, and W. Xie, "Pmc-vqa: Visual instruction tuning for medical visual question answering," *arXiv:2305.10415*, 2023. 1, 4, 6, 10

[38] J. Ye, A. Hu, H. Xu, Q. Ye, M. Yan, Y. Dan, C. Zhao, G. Xu, C. Li, J. Tian *et al.*, "mplug-docowl: Modularized multimodal large language model for document understanding," *arXiv:2307.02499*, 2023. 1, 10

[39] Y. Liu, B. Yang, Q. Liu, Z. Li, Z. Ma, S. Zhang, and X. Bai, "Textmonkey: An ocr-free large multimodal model for understanding document," *arXiv:2403.04473*, 2024. 1, 10

[40] A. Hu, Y. Shi, H. Xu, J. Ye, Q. Ye, M. Yan, C. Li, Q. Qian, J. Zhang, and F. Huang, "mplug-paperowl: Scientific diagram analysis with the multimodal large language model," *arXiv:2311.18248*, 2023. 1

[41] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang, "An embodied generalist agent in 3d world," *arXiv:2311.12871*, 2023. 1, 9, 10

[42] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei, "Kosmos-2: Grounding multimodal large language models to the world," *arXiv:2306.14824*, 2023. 1

[43] Z. Yang, J. Liu, Y. Han, X. Chen, Z. Huang, B. Fu, and G. Yu, "Appagent: Multimodal agents as smartphone users," *arXiv:2312.13771*, 2023. 1, 10

[44] W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Dong, M. Ding *et al.*, "Cogagent: A visual language model for gui agents," *arXiv:2312.08914*, 2023. 1, 3, 10

[45] J. Wang, H. Xu, J. Ye, M. Yan, W. Shen, J. Zhang, F. Huang, and J. Sang, "Mobile-agent: Autonomous multi-modal mobile device agent with visual perception," *arXiv:2401.16158*, 2024. 1, 10

[46] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, "Reproducible scaling laws for contrastive language-image learning," in *CVPR*, 2023. 2, 3

[47] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "Eva-clip: Improved training techniques for clip at scale," *arXiv:2303.15389*, 2023. 2, 3

[48] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva: Exploring the limits of masked visual representation learning at scale," in *CVPR*, 2023. 2

[49] R. Bavishi, E. Elsen, C. Hawthorne, M. Nye, A. Odena, A. Somani, and S. Taşırlar, "Introducing our multimodal models," 2023. [Online]. Available: https://www.adept.ai/blog/fuyu-8b 2

[50] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," *arXiv:2310.03744*, 2023. 3, 4

[51] Z. Li, B. Yang, Q. Liu, Z. Ma, S. Zhang, J. Yang, Y. Sun, Y. Liu, and X. Bai, "Monkey: Image resolution and text label are important things for large multi-modal models," *arXiv:2311.06607*, 2023. 3

[52] B. McKinzie, Z. Gan, J.-P. Fauconnier, S. Dodge, B. Zhang, P. Dufter, D. Shah, X. Du, F. Peng, F. Weers *et al.*, "Mm1: Methods, analysis & insights from multimodal llm pre-training," *arXiv:2403.09611*, 2024. 3, 4

[53] Z. Lin, C. Liu, R. Zhang, P. Gao, L. Qiu, H. Xiao, H. Qiu, C. Lin, W. Shao, K. Chen *et al.*, "Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models," *arXiv:2311.07575*, 2023. 3

[54] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "Clap learning audio concepts from natural language supervision," in *ICASSP*, 2023. 3

[55] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra, "Imagebind: One embedding space to bind them all," in *CVPR*, 2023. 3

[56] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, "Scaling instruction-finetuned language models," *arXiv:2210.11416*, 2022. 3, 4, 6

[57] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv:2307.09288*, 2023. 3, 4

[58] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, "Qwen technical report," *arXiv:2309.16609*, 2023. 3, 4, 10

[59] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv:2301.12597*, 2023. 3, 4

[60] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi, "Instructblip: Towards general-purpose vision-language models with instruction tuning," *arXiv:2305.06500*, 2023. 3, 4, 6, 7, 8

[61] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, "Llava-next: Improved reasoning, ocr, and world knowledge," January 2024. [Online]. Available: https://llava-vl.github.io/blog/2024-01-30-llava-next/ 3

[62] Y. Lu, C. Li, H. Liu, J. Yang, J. Gao, and Y. Shen, "An empirical study of scaling instruct-tuned large multimodal models," *arXiv:2309.09958*, 2023. 3

[63] X. Chu, L. Qiao, X. Lin, S. Xu, Y. Yang, Y. Hu, F. Wei, X. Zhang, B. Zhang, X. Wei *et al.*, "Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices," *arXiv:2312.16886*, 2023. 3, 10

[64] X. Chu, L. Qiao, X. Zhang, S. Xu, F. Wei, Y. Yang, X. Sun, Y. Hu, X. Lin, B. Zhang *et al.*, "Mobilevlm v2: Faster and stronger baseline for vision language model," *arXiv:2402.03766*, 2024. 3

[65] S. Shen, L. Hou, Y. Zhou, N. Du, S. Longpre, J. Wei, H. W. Chung, B. Zoph, W. Fedus, X. Chen *et al.*, "Mixture-of-experts meets instruction tuning: A winning combination for large language models," *arXiv:2305.14705*, 2023. 3

[66] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand *et al.*, "Mixtral of experts," *arXiv:2401.04088*, 2024. 3

[67] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *JMLR*, 2022. 3

[68] B. Lin, Z. Tang, Y. Ye, J. Cui, B. Zhu, P. Jin, J. Zhang, M. Ning, and L. Yuan, "Moe-llava: Mixture of experts for large vision-language models," *arXiv:2401.15947*, 2024. 3

[69] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020. 4

[70] F. Chen, M. Han, H. Zhao, Q. Zhang, J. Shi, S. Xu, and B. Xu, "X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages," *arXiv:2305.04160*, 2023. 4, 6, 8, 9

[71] Y. Su, T. Lan, H. Li, J. Xu, Y. Wang, and D. Cai, "Pandagpt: One model to instruction-follow them all," *arXiv:2305.16355*, 2023. 4, 6

[72] R. Pi, J. Gao, S. Diao, R. Pan, H. Dong, J. Zhang, L. Yao, J. Han, H. Xu, and L. K. T. Zhang, "Detgpt: Detect what you need via reasoning," *arXiv:2305.14167*, 2023. 4, 7

[73] Y. Zeng, H. Zhang, J. Zheng, J. Xia, G. Wei, Y. Wei, Y. Zhang, and T. Kong, "What matters in training a gpt4-style language model with multimodal inputs?" *arXiv:2307.02469*, 2023. 4, 7

[74] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *NeurIPS*, 2022. 4, 11, 12

[75] W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song *et al.*, "Cogvlm: Visual expert for pretrained language models," *arXiv:2311.03079*, 2023. 4

[76] R. Zhang, J. Han, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, P. Gao, and Y. Qiao, "Llama-adapter: Efficient fine-tuning of language models with zero-init attention," *arXiv:2303.16199*, 2023. 4, 6, 8, 9

[77] S. Yin, C. Fu, S. Zhao, T. Xu, H. Wang, D. Sui, Y. Shen, K. Li, X. Sun, and E. Chen, "Woodpecker: Hallucination correction for multimodal large language models," *arXiv:2310.16045*, 2023. 4, 9, 10, 11

[78] J. Guo, J. Li, D. Li, A. M. H. Tiong, B. Li, D. Tao, and S. Hoi, "From images to textual prompts: Zero-shot visual question answering with frozen large language models," in *CVPR*, 2023. 4

[79] T. Wang, J. Zhang, J. Fei, Y. Ge, H. Zheng, Y. Tang, Z. Li, M. Gao, S. Zhao, Y. Shan *et al.*, "Caption anything: Interactive image description with diverse multimodal controls," *arXiv:2305.02677*, 2023. 4, 12, 13

[80] D. Zhu, J. Chen, K. Haydarov, X. Shen, W. Zhang, and M. Elhoseiny, "Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions," *arXiv:2303.06594*, 2023. 4, 12

[81] Q. Ye, H. Xu, G. Xu, J. Ye, M. Yan, Y. Zhou, J. Wang, A. Hu, P. Shi, Y. Shi *et al.*, "mplug-owl: Modularization empowers large language models with multimodality," *arXiv:2304.14178*, 2023. 4, 7, 9

[82] W. Wang, Z. Chen, X. Chen, J. Wu, X. Zhu, G. Zeng, P. Luo, T. Lu, J. Zhou, Y. Qiao *et al.*, "Visionllm: Large language model is also an open-ended decoder for vision-centric tasks," *arXiv:2305.11175*, 2023. 4, 6

[83] L. Chen, J. Li, X. Dong, P. Zhang, C. He, J. Wang, F. Zhao, and D. Lin, "Sharegpt4v: Improving large multi-modal models with better captions," *arXiv:2311.12793*, 2023. 5

[84] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *ACL*, 2018. 5

[85] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," in *CVPR*, 2021. 5

[86] V. Ordonez, G. Kulkarni, and T. Berg, "Im2text: Describing images using 1 million captioned photographs," *NeurIPS*, 2011. 5

[87] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *NeurIPS*, 2022. 5

[88] C. Schuhmann, A. Köpf, R. Vencu, T. Coombes, and R. Beaumont, "Laion coco: 600m synthetic captions from laion2b-en." *https://laion.ai/blog/laion-coco/*, 2022. 5

[89] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *ICML*, 2022. 5

[90] M. Byeon, B. Park, H. Kim, S. Lee, W. Baek, and S. Kim, "Coyo-700m: Image-text pair dataset," https://github.com/kakaobrain/coyo-dataset, 2022. 5

[91] J. Wang, L. Meng, Z. Weng, B. He, Z. Wu, and Y.-G. Jiang, "To see is to believe: Prompting gpt-4v for better visual instruction tuning," *arXiv:2311.07574*, 2023. 5, 7

[92] G. H. Chen, S. Chen, R. Zhang, J. Chen, X. Wu, Z. Zhang, Z. Chen, J. Li, X. Wan, and B. Wang, "Allava: Harnessing gpt4v-synthesized data for a lite vision-language model," *arXiv:2402.11684*, 2024. 5, 7

[93] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *CVPR*, 2016. 5

[94] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *arXiv:2303.17395*, 2023. 5

[95] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *NeurIPS*, 2022. 6, 8

[96] S. Iyer, X. V. Lin, R. Pasunuru, T. Mihaylov, D. Simig, P. Yu, K. Shuster, T. Wang, Q. Liu, P. S. Koura *et al.*, "Opt-iml: Scaling language model instruction meta learning through the lens of generalization," *arXiv:2212.12017*, 2022. 6

[97] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja *et al.*, "Multitask prompted training enables zero-shot task generalization," *arXiv:2110.08207*, 2021. 6

[98] T. Gong, C. Lyu, S. Zhang, Y. Wang, M. Zheng, Q. Zhao, K. Liu, W. Zhang, P. Luo, and K. Chen, "Multimodal-gpt: A vision and language model for dialogue with humans," *arXiv:2305.04790*, 2023. 6, 7, 11

[99] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *ICCV*, 2015. 6

[100] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *CVPR*, 2015. 6

[101] G. Luo, Y. Zhou, T. Ren, S. Chen, X. Sun, and R. Ji, "Cheap and quick: Efficient vision-language instruction tuning for large language models," *arXiv:2305.15023*, 2023. 6, 7, 8, 9

[102] Z. Xu, Y. Shen, and L. Huang, "Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning," *arXiv:2212.10773*, 2022. 6, 7, 8, 9

[103] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue *et al.*, "Llama-adapter v2: Parameter-efficient visual instruction model," *arXiv:2304.15010*, 2023. 6, 7, 8, 9

[104] Z. Zhao, L. Guo, T. Yue, S. Chen, S. Shao, X. Zhu, Z. Yuan, and J. Liu, "Chatbridge: Bridging modalities with large language model as a language catalyst," *arXiv:2305.16103*, 2023. 6, 7, 8, 9

[105] L. Li, Y. Yin, S. Li, L. Chen, P. Wang, S. Ren, M. Li, Y. Yang, J. Xu, X. Sun, L. Kong, and Q. Liu, "M^{3}it: A large-scale dataset towards multi-modal multilingual instruction tuning," *arXiv:2306.04387*, 2023. 6, 8, 9

[106] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language model with self generated instructions," *arXiv:2212.10560*, 2022. 7

[107] R. Yang, L. Song, Y. Li, S. Zhao, Y. Ge, X. Li, and Y. Shan, "Gpt4tools: Teaching large language model to use tools via self-instruction," *arXiv:2305.18752*, 2023. 7, 9, 12, 13

[108] L. Wei, Z. Jiang, W. Huang, and L. Sun, "Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4," *arXiv:2308.12067*, 2023. 7

[109] Y. Du, H. Guo, K. Zhou, W. X. Zhao, J. Wang, C. Wang, M. Cai, R. Song, and J.-R. Wen, "What makes for good visual instructions? synthesizing complex visual reasoning instructions for visual instruction tuning," *arXiv:2311.01487*, 2023. 7

[110] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," *arXiv:1909.08593*, 2019. 8

[111] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize with human feedback," *NeurIPS*, 2020. 8

[112] Z. Sun, S. Shen, S. Cao, H. Liu, C. Li, Y. Shen, C. Gan, L.-Y. Gui, Y.-X. Wang, Y. Yang *et al.*, "Aligning large multimodal models with factually augmented rlhf," *arXiv:2309.14525*, 2023. 8, 11

[113] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *NeurIPS*, 2023. 8

[114] T. Yu, Y. Yao, H. Zhang, T. He, Y. Han, G. Cui, J. Hu, Z. Liu, H.-T. Zheng, M. Sun *et al.*, "Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback," *arXiv:2312.00849*, 2023. 8

[115] L. Li, Z. Xie, M. Li, S. Chen, P. Wang, L. Chen, Y. Yang, B. Wang, and L. Kong, "Silkie: Preference distillation for large visual language models," *arXiv:2312.10665*, 2023. 8

[116] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan, "Learn to explain: Multimodal reasoning via thought chains for science question answering," *NeurIPS*, 2022. 8, 9, 12

[117] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *CVPR*, 2015. 8

[118] H. Agrawal, K. Desai, Y. Wang, X. Chen, R. Jain, M. Johnson, D. Batra, D. Parikh, S. Lee, and P. Anderson, "Nocaps: Novel object captioning at scale," in *ICCV*, 2019. 8

[119] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *TACL*, 2014. 8

[120] X. He, Y. Zhang, L. Mou, E. Xing, and P. Xie, "Pathvqa: 30000+ questions for medical visual question answering," *arXiv:2003.10286*, 2020. 9

[121] J. J. Lau, S. Gayen, A. Ben Abacha, and D. Demner-Fushman, "A dataset of clinically generated visual questions and answers about radiology images," *Sci. Data*, 2018. 9

[122] B. Liu, L.-M. Zhan, L. Xu, L. Ma, Y. Yang, and X.-M. Wu, "Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering," in *ISBI*, 2021. 9

[123] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, Z. Qiu, W. Lin, Z. Qiu, W. Lin *et al.*, "Mme: A comprehensive evaluation benchmark for multimodal large language models," *arXiv:2306.13394*, 2023. 9, 10

[124] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu *et al.*, "Mmbench: Is your multi-modal model an all-around player?" *arXiv:2307.06281*, 2023. 9

[125] W. Yu, Z. Yang, L. Li, J. Wang, K. Lin, Z. Liu, X. Wang, and L. Wang, "Mm-vet: Evaluating large multimodal models for integrated capabilities," *arXiv:2308.02490*, 2023. 9

[126] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan, "Seed-bench: Benchmarking multimodal llms with generative comprehension," in *CVPR*, 2024. 9

[127] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao, "Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts," in *ICLR*, 2024. 9

[128] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun *et al.*, "Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi," *arXiv:2311.16502*, 2023. 9

[129] F. Liu, T. Guan, Z. Li, L. Chen, Y. Yacoob, D. Manocha, and T. Zhou, "Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multi-modality models," in *CVPR*, 2024. 9

[130] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan, "Video-chatgpt: Towards detailed video understanding via large vision and language models," *arXiv:2306.05424*, 2023. 9

[131] M. Ning, B. Zhu, Y. Xie, B. Lin, J. Cui, L. Yuan, D. Chen, and L. Yuan, "Video-bench: A comprehensive benchmark and toolkit for evaluating video-based large language models," *arXiv:2311.16103*, 2023. 9

[132] Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, and J.-R. Wen, "Evaluating object hallucination in large vision-language models," *arXiv:2305.10355*, 2023. 9, 10

[133] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014. 9

[134] M. Li, L. Li, Y. Yin, M. Ahmed, Z. Liu, and Q. Liu, "Red teaming visual language models," *arXiv:2401.12915*, 2024. 9

[135] Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, and L. Wang, "The dawn of lmms: Preliminary explorations with gpt-4v (ision)," *arXiv:2309.17421*. 9

[136] L. Wen, X. Yang, D. Fu, X. Wang, P. Cai, X. Li, T. Ma, Y. Li, L. Xu, D. Shang *et al.*, "On the road with gpt-4v (ision): Early explorations of visual-language model on autonomous driving," *arXiv:2311.05332*. 9

[137] C. Fu, R. Zhang, H. Lin, Z. Wang, T. Gao, Y. Luo, Y. Huang, Z. Zhang, L. Qiu, G. Ye *et al.*, "A challenger to gpt-4v? early explorations of gemini in visual expertise," *arXiv:2312.12436*. 9

[138] S. Zhang, P. Sun, S. Chen, M. Xiao, W. Shao, W. Zhang, K. Chen, and P. Luo, "Gpt4roi: Instruction tuning large language model on region-of-interest," *arXiv:2307.03601*, 2023. 9

[139] S. Xuan, Q. Guo, M. Yang, and S. Zhang, "Pink: Unveiling the power of referential comprehension for multi-modal llms," *arXiv:2310.00582*, 2023. 9

[140] H. Rasheed, M. Maaz, S. Shaji, A. Shaker, S. Khan, H. Cholakkal, R. M. Anwer, E. Xing, M.-H. Yang, and F. S. Khan, "Glamm: Pixel grounding large multimodal model," *arXiv:2311.03356*. 9

[141] H. You, H. Zhang, Z. Gan, X. Du, B. Zhang, Z. Wang, L. Cao, S.-F. Chang, and Y. Yang, "Ferret: Refer and ground anything anywhere at any granularity," *arXiv:2310.07704*, 2023. 9

[142] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia, "Lisa: Reasoning segmentation via large language model," *arXiv:2308.00692*, 2023. 9, 13

[143] R. Xu, X. Wang, T. Wang, Y. Chen, J. Pang, and D. Lin, "Pointllm: Empowering large language models to understand point clouds," *arXiv:2308.16911*, 2023. 9

[144] S. Chen, X. Chen, C. Zhang, M. Li, G. Yu, H. Fei, H. Zhu, J. Fan, and T. Chen, "Ll3da: Visual interactive instruction tuning for omni-3d understanding, reasoning, and planning," *arXiv:2311.18651*, 2023. 9

[145] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan, "3d-llm: Injecting the 3d world into large language models," *NeurIPS*, 2023. 9

[146] Q. Sun, Q. Yu, Y. Cui, F. Zhang, X. Zhang, Y. Wang, H. Gao, J. Liu, T. Huang, and X. Wang, "Generative pretraining in multimodality," in *ICLR*, 2024. 9

[147] J. Zhan, J. Dai, J. Ye, Y. Zhou, D. Zhang, Z. Liu, X. Zhang, R. Yuan, G. Zhang, L. Li *et al.*, "Anygpt: Unified multimodal llm with discrete sequence modeling," *arXiv:2402.12226*, 2024. 9

[148] E. Aiello, L. Yu, Y. Nie, A. Aghajanyan, and B. Oguz, "Jointly training large autoregressive multimodal models," *arXiv:2309.15564*, 2023. 9

[149] D. Zhang, S. Li, X. Zhang, J. Zhan, P. Wang, Y. Zhou, and X. Qiu, "Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities," *arXiv:2305.11000*, 2023. 9

[150] P. K. Rubenstein, C. Asawaroengchai, D. D. Nguyen, A. Bapna, Z. Borsos, F. d. C. Quitry, P. Chen, D. E. Badawy, W. Han, E. Kharitonov *et al.*, "Audiopalm: A large language model that can speak and listen," *arXiv:2306.12925*, 2023. 9

[151] X. Wang, B. Zhuang, and Q. Wu, "Modaverse: Efficiently transforming modalities with llms," *arXiv:2401.06395*, 2024. 9

[152] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, 2020. 10

[153] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022. 10

[154] R. Gong, Q. Huang, X. Ma, H. Vo, Z. Durante, Y. Noda, Z. Zheng, S.-C. Zhu, D. Terzopoulos, L. Fei-Fei *et al.*, "Mindagent: Emergent gaming interaction," *arXiv:2309.09971*, 2023. 10

[155] Y. Mu, Q. Zhang, M. Hu, W. Wang, M. Ding, J. Jin, B. Wang, J. Dai, Y. Qiao, and P. Luo, "Embodiedgpt: Vision-language pre-training via embodied chain of thought," *arXiv:2305.15021*, 2023. 10

[156] A. Hu, H. Xu, J. Ye, M. Yan, L. Zhang, B. Zhang, C. Li, J. Zhang, Q. Jin, F. Huang *et al.*, "mplug-docowl 1.5: Unified structure learning for ocr-free document understanding," *arXiv:2403.12895*, 2024. 10

[157] J. Ye, A. Hu, H. Xu, Q. Ye, M. Yan, G. Xu, C. Li, J. Tian, Q. Qian, J. Zhang *et al.*, "Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model," in *EMNLP*, 2023. 10

[158] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao, "Llava-med: Training a large language-and-vision assistant for biomedicine in one day," *arXiv:2306.00890*, 2023. 10

[159] B. Zhai, S. Yang, X. Zhao, C. Xu, S. Shen, D. Zhao, K. Keutzer, M. Li, T. Yan, and X. Fan, "Halle-switch: Rethinking and controlling object existence hallucinations in large vision language models for detailed caption," *arXiv:2310.01779*, 2023. 10, 11

[160] A. Rohrbach, L. A. Hendricks, K. Burns, T. Darrell, and K. Saenko, "Object hallucination in image captioning," in *EMNLP*, 2018. 10

[161] J. Wang, Y. Zhou, G. Xu, P. Shi, C. Zhao, H. Xu, Q. Ye, M. Yan, J. Zhang, J. Zhu *et al.*, "Evaluation and analysis of hallucination in large vision-language models," *arXiv:2308.15126*, 2023. 10

[162] L. Jing, R. Li, Y. Chen, M. Jia, and X. Du, "Faithscore: Evaluating hallucinations in large vision-language models," *arXiv:2311.01477*, 2023. 10

[163] J. Wang, Y. Wang, G. Xu, J. Zhang, Y. Gu, H. Jia, M. Yan, J. Zhang, and J. Sang, "An llm-free multi-dimensional benchmark for mllms hallucination evaluation," *arXiv:2311.07397*, 2023. 10

[164] F. Liu, K. Lin, L. Li, J. Wang, Y. Yacoob, and L. Wang, "Mitigating hallucination in large multi-modal models via robust instruction tuning," in *ICLR*, 2024. 11

[165] S. Leng, H. Zhang, G. Chen, X. Li, S. Lu, C. Miao, and L. Bing, "Mitigating object hallucinations in large vision-language models through visual contrastive decoding," in *CVPR*, 2024. 11

[166] C. Jiang, H. Xu, M. Dong, J. Chen, W. Ye, M. Yan, Q. Ye, J. Zhang, F. Huang, and S. Zhang, "Hallucination augmented contrastive learning for multimodal large language model," *arXiv:2312.06968*, 2023. 11

[167] Y. Zhou, C. Cui, J. Yoon, L. Zhang, Z. Deng, C. Finn, M. Bansal, and H. Yao, "Analyzing and mitigating object hallucination in large vision-language models," *arXiv:2310.00754*, 2023. 11

[168] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey for in-context learning," *arXiv:2301.00234*, 2022. 11

[169] P. Lu, B. Peng, H. Cheng, M. Galley, K.-W. Chang, Y. N. Wu, S.-C. Zhu, and J. Gao, "Chameleon: Plug-and-play compositional reasoning with large language models," *arXiv:2304.09842*, 2023. 11, 12, 13

[170] T. Gupta and A. Kembhavi, "Visual programming: Compositional visual reasoning without training," in *CVPR*, 2023. 11, 12, 13

[171] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp, "Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity," *arXiv:2104.08786*, 2021. 11

[172] B. Li, Y. Zhang, L. Chen, J. Wang, F. Pu, J. Yang, C. Li, and Z. Liu, "Mimic-it: Multi-modal in-context instruction tuning," *arXiv:2306.05425*, 2023. 11

[173] Q. Sun, Q. Yu, Y. Cui, F. Zhang, X. Zhang, Y. Wang, H. Gao, J. Liu, T. Huang, and X. Wang, "Generative pretraining in multimodality," *arXiv:2307.05222*, 2023. 11

[174] D. Sheng, D. Chen, Z. Tan, Q. Liu, Q. Chu, J. Bao, T. Gong, B. Liu, S. Xu, and N. Yu, "Towards more unified in-context visual understanding," *arXiv:2312.02520*, 2023. 12

[175] Y. Tai, W. Fan, Z. Zhang, F. Zhu, R. Zhao, and Z. Liu, "Link-context learning for multimodal llms," *arXiv:2308.07891*, 2023. 12

[176] H. Zhao, Z. Cai, S. Si, X. Ma, K. An, L. Chen, Z. Liu, S. Wang, W. Han, and B. Chang, "Mmicl: Empowering vision-language model with multi-modal in-context learning," *arXiv:2309.07915*, 2023. 12

[177] J. Jeong, "Hijacking context in large multi-modal models," *arXiv:2312.07553*, 2023. 12, 14

[178] Z. Yang, Z. Gan, J. Wang, X. Hu, Y. Lu, Z. Liu, and L. Wang, "An empirical study of gpt-3 for few-shot knowledge-based vqa," in *AAAI*, 2022. 12

[179] M. Tsimpoukelli, J. L. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill, "Multimodal few-shot learning with frozen language models," *NeurIPS*, 2021. 12

[180] B. Li, Y. Zhang, L. Chen, J. Wang, J. Yang, and Z. Liu, "Otter: A multi-modal model with in-context instruction tuning," *arXiv:2305.03726*, 2023. 12

[181] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface," *arXiv:2303.17580*, 2023. 12, 13

[182] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *arXiv:2205.11916*, 2022. 12

[183] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," *arXiv:2210.03493*, 2022. 12

[184] D. Rose, V. Himakunthala, A. Ouyang, R. He, A. Mei, Y. Lu, M. Saxon, C. Sonar, D. Mirza, and W. Y. Wang, "Visual chain of thought: Bridging logical gaps with multimodal infillings," *arXiv:2305.02317*, 2023. 12

[185] Z. Zhang, A. Zhang, M. Li, H. Zhao, G. Karypis, and A. Smola, "Multimodal chain-of-thought reasoning in language models," *arXiv:2302.00923*, 2023. 12

[186] V. Himakunthala, A. Ouyang, D. Rose, R. He, A. Mei, Y. Lu, C. Sonar, M. Saxon, and W. Y. Wang, "Let's think frame by frame: Evaluating video chain of thought with video infilling and prediction," *arXiv:2305.13903*, 2023. 12

[187] J. Ge, H. Luo, S. Qian, Y. Gan, J. Fu, and S. Zhan, "Chain of thought prompt tuning in vision language models," *arXiv:2304.07919*, 2023. 12

[188] C. Wu, S. Yin, W. Qi, X. Wang, Z. Tang, and N. Duan, "Visual chatgpt: Talking, drawing and editing with visual foundation models," *arXiv:2303.04671*, 2023. 12, 13

[189] G. Zheng, B. Yang, J. Tang, H.-Y. Zhou, and S. Yang, "Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models," in *NeurIPS*, 2023. 12

[190] A. Parisi, Y. Zhao, and N. Fiedel, "Talm: Tool augmented language models," *arXiv:2205.12255*, 2022. 12

[191] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig, "Pal: Program-aided language models," *arXiv:2211.10435*, 2022. 12

[192] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," *arXiv:2302.04761*, 2023. 12

[193] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders *et al.*, "Webgpt: Browser-assisted question-answering with human feedback," *arXiv:2112.09332*, 2021. 12

[194] A. Zeng, A. Wong, S. Welker, K. Choromanski, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke *et al.*, "Socratic models: Composing zero-shot multimodal reasoning with language," *arXiv:2204.00598*, 2022. 12

[195] H. You, R. Sun, Z. Wang, L. Chen, G. Wang, H. A. Ayyubi, K.-W. Chang, and S.-F. Chang, "Idealgpt: Iteratively decomposing vision and language reasoning via large language models," *arXiv:2305.14985*, 2023. 12, 13

[196] V. Udandarao, A. Gupta, and S. Albanie, "Sus-x: Training-free name-only transfer of vision-language models," *arXiv:2211.16198*, 2022. 12

[197] X. Zhu, R. Zhang, B. He, Z. Zeng, S. Zhang, and P. Gao, "Pointclip v2: Adapting clip for powerful 3d open-world learning," *arXiv:2211.11682*, 2022. 13

[198] R. Zhang, X. Hu, B. Li, S. Huang, H. Deng, Y. Qiao, P. Gao, and H. Li, "Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners," in *CVPR*, 2023. 13

[199] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018. 13

[200] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *CVPR*, 2019. 13

[201] P. Gao, Z. Jiang, H. You, P. Lu, S. C. Hoi, X. Wang, and H. Li, "Dynamic fusion with intra-and inter-modality attention flow for visual question answering," in *CVPR*, 2019. 13

[202] A. Zeng, A. Wong, S. Welker, K. Choromanski, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke *et al.*, "Socratic models: Composing zero-shot multimodal reasoning with language," *arXiv:2204.00598*, 2022. 13

[203] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "Stylenet: Generating attractive visual captions with styles," in *CVPR*, 2017. 13

[204] A. Mathews, L. Xie, and X. He, "Senticap: Generating image descriptions with sentiments," in *AAAI*, 2016. 13

[205] P. Wu and S. Xie, "V*: Guided visual search as a core mechanism in multimodal llms," *arXiv:2312.14135*, 2023. 13

[206] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, O. Bousquet, Q. Le, and E. Chi, "Least-to-most prompting enables complex reasoning in large language models," *arXiv:2205.10625*, 2022. 13

[207] Y. Zhao, T. Pang, C. Du, X. Yang, C. Li, N.-M. Cheung, and M. Lin, "On evaluating adversarial robustness of large vision-language models," *arXiv:2305.16934*, 2023. 14

[208] E. Shayegani, Y. Dong, and N. Abu-Ghazaleh, "Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models," in *ICLR*, 2023. 14