A Survey on Multimodal Large Language Models

Shukang Yin[1*], **Chaoyou Fu**[2*‡†], **Sirui Zhao**[1*‡], **Ke Li**[2], **Xing Sun**[2], **Tong Xu**[1], **Enhong Chen**[1‡]
[1]School of CST., USTC & State Key Laboratory of Cognitive Intelligence
[2]Tencent YouTu Lab
{xjtupanda,sirui}@mail.ustc.edu.cn, {tongxu,cheneh}@ustc.edu.cn
{bradyfu24}@gmail.com, {tristanli,winfredsun}@tencent.com

Abstract

Multimodal Large Language Model (MLLM) recently 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 methods, suggesting a potential path to artificial general intelligence. In this paper, we aim to trace and summarize the recent progress of MLLM. First of all, we present the formulation of MLLM and delineate its related concepts. Then, we discuss the key techniques and applications, including Multimodal Instruction Tuning (M-IT), Multimodal In-Context Learning (M-ICL), Multimodal Chain of Thought (M-CoT), and LLM-Aided Visual Reasoning (LAVR). Finally, 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.

1. Introduction

Recent years have seen the remarkable progress of large language models [1–4]. By scaling up data size and model size, these LLMs raise amazing emergent abilities, typically including In-Context Learning (ICL) [5], instruction following [4, 6], and Chain of Thought (CoT) [7]. 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 foundation models make rapid progress in perception [8–10], and the traditional combination with text pays more attention to modality alignment [11] and task unity [12], developing slowly in reasoning.

In light of this complementarity, unimodal LLMs and vision models run towards each other at the same time, ultimately leading to the new field of MLLM. Formally, it refers to the LLM-based model with the ability to receive and reason with multimodal information. From the perspective of developing Artificial General Intelligence (AGI), MLLM may take a step forward from LLM for the following reasons: (1) MLLM is more in line with the way humans perceive the world. Our humans naturally receive multisensory inputs that are often complementary and cooperative. Therefore, multimodal information is expected to make MLLM more intelligent. (2) MLLM offers a more user-friendly interface. Thanks to the support of multimodal input, users can interact and communicate with the intelligent assistant in a more flexible way. (3) MLLM is a more well-rounded task-solvers. While LLMs can typically perform NLP tasks, MLLMs can generally support a larger spectrum of tasks.

GPT-4 [2] ignites a research frenzy over MLLM because of the amazing examples it shows. However, GPT-4 does not open the multimodal interface, and no information about the model has been made public up until now. In spite of this, many efforts have been made by the research community to develop capable and open-sourced MLLMs, and some surprising practical capabilities have been exhibited, such as writing website codes based on images [13], understanding the deep meaning of a meme [14], and OCR-free math reasoning [15]. 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. Specifically, we divide the existing MLLMs into four types with corresponding summarizations and, meanwhile, open a GitHub page that would be updated in real-time. To the best of our knowledge, this is the first survey on MLLM.

*Equal contribution.
†Project leader.
‡Corresponding author.
§Version: v1 (update on June 23, 2023).

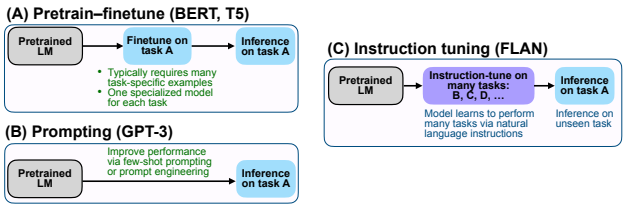

Figure 1. Comparisons of three typical learning paradigms. The image is from [16].

2. Overview

This paper categorizes recent representative MLLMs into four main genres: Multimodal Instruction Tuning (M-IT), Multimodal In-Context Learning (M-ICL), Multimodal Chain-of-Thought (M-CoT), and LLM-Aided Visual Reasoning (LAVR). The first three constitute the fundamentals of MLLMs, while the last one is a multimodal system with LLM as the core. Note that the three techniques are relatively independent and can be utilized in combination. Therefore, our illustration of a concept may also involve others.

We organize the survey according to the four main categories and introduce them sequentially. We start with a detailed introduction of M-IT (§3.1) to reveal how LLMs can be adapted for multimodality in terms of two aspects: architecture and data. Then we introduce M-ICL (§3.2), an effective technique commonly used at the inference stage to boost few-shot performance. Another important technique is the M-CoT (§3.3), which is typically used in complex reasoning tasks. Afterward, we further summarize several roles that LLMs mainly take in LAVR (§3.4), which frequently involves the three techniques. Finally, we finish our survey with a summary and potential research directions.

3. Method

3.1. Multimodal Instruction Tuning

3.1.1 Introduction

Instruction refers to the description of tasks. Instruction tuning is a technique that involves finetuning pre-trained LLMs on a collection of instruction-formatted datasets [16]. 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 works in the realm of NLP, such as ChatGPT [1], InstructGPT [17], FLAN [16, 18], and OPT-IML [19].

The comparisons between instruction tuning and related typical learning paradigms are illustrated in Fig. 1. The supervised finetuning approach usually requires many 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 [5]. 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 [20].

Contrastively, traditional multimodal models are still confined to the first two tuning paradigms, lacking the zero-shot ability. Therefore, many recent works [13, 21, 22] have explored extending the success of instruction tuning in LLMs to multimodality. In order to extend from unimodality to multimodality, the corresponding adaptations are necessary for both the data and the model. For the data, researchers usually acquire M-IT datasets by adapting existing benchmark datasets [23–28] or by self-instruction [13, 21, 29]. Regarding the model, a common approach is to inject the information of foreign modalities into LLMs and treat them as strong reasoners. Relevant works either directly align foreign embeddings to the LLMs [21, 23–25, 27, 28, 30–32] or resort to expert models to translate foreign modalities into natural languages that LLMs can ingest [33, 34]. Formulated in this way, these works transform LLMs into multimodal chatbots [13, 21, 22, 33, 35] and multimodal universal task solvers [23, 24, 26] through multimodal instruction tuning.

In the following parts of this section, we first offer the foundational knowledge (§3.1.2). Before transitioning to the delineation of M-IT, we additionally introduce a common process prior to M-IT, *i.e.*, alignment pre-training (§3.1.3). Then we structure the remaining content as illustrated in Fig. 2: We first introduce how the M-IT data are collected (§3.1.4), followed by a detailed discussion of the model adaption for MLLMs, *i.e.*, various ways of bridging the gap between different modalities (§3.1.5). Finally, we introduce the evaluation methods to assess instruction-tuned MLLMs (§3.1.6).

<BOS> Below is an instruction that describes a task. Write a response that appropriately completes the request

Instruction: <instruction>
Input: {<image>, <text>}
Response: <output><EOS>

Table 1. A simplified template to structure the multimodal instruction data. The <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>. <BOS> and <EOS> are tokens denoting the start and the end of the input to LLM, respectively. The example is adapted from [31].

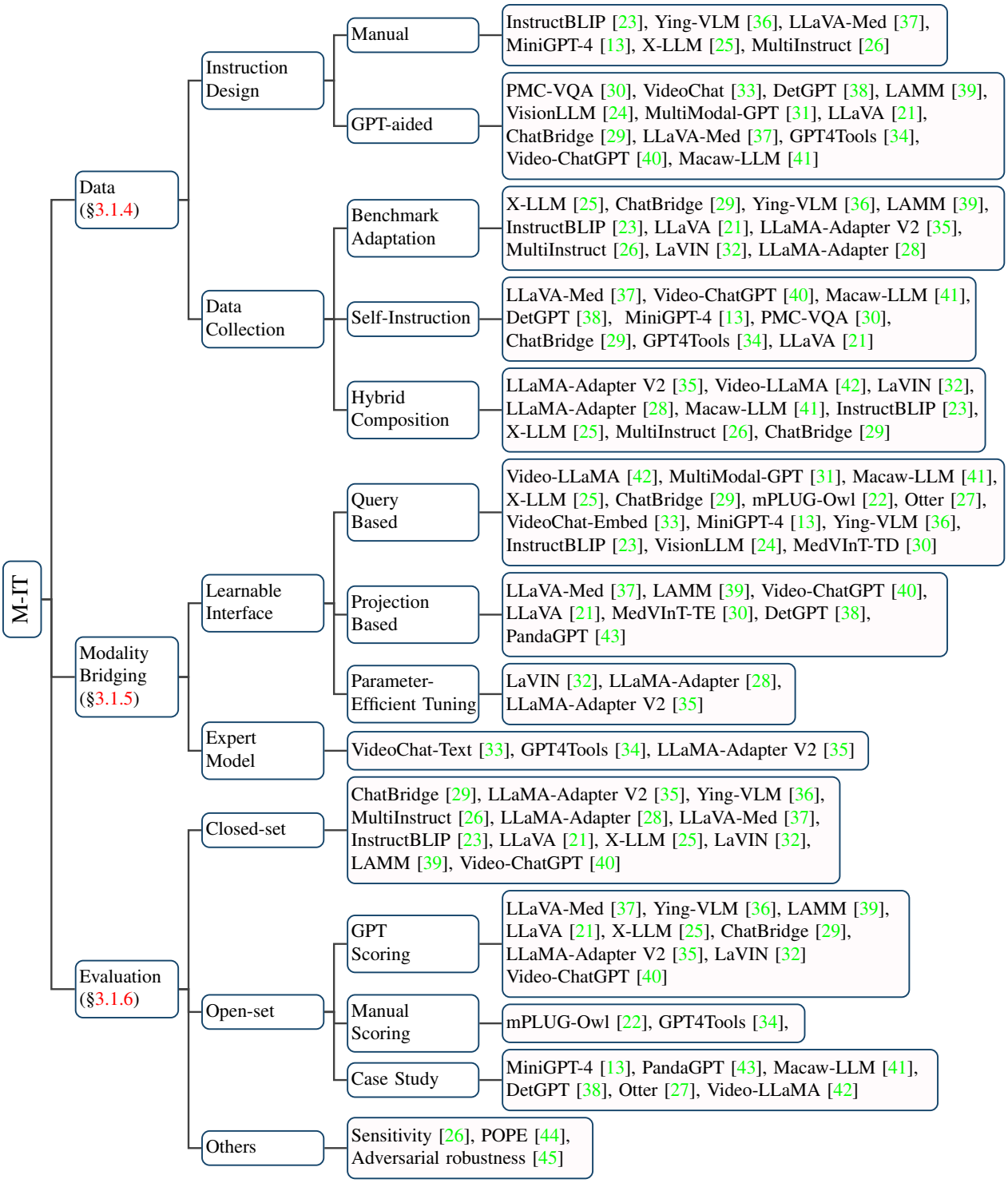

Figure 2. Taxonomy of Multimodal Instruction Tuning (M-IT) that consists of data construction, modality bridging, and evaluation.

3.1.2 Preliminaries

This section briefly illustrates the general structure of multimodal instruction samples and the common process of M-IT.

A multimodal instruction sample often includes an 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 Visual Question-Answering (VQA) task [46] or only an image like the image captioning task [47]. The output is the answer to the instruction conditioned on the input. The instruction template is flexible and subject to manual designs [21, 31, 33], as exemplified in Table 1. Note that the instruction samples can also be generalized to multi-round instructions, where the multimodal inputs are shared [21, 30, 31, 43].

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 the LLMs [21, 30, 32, 43], based on which the MLLM is forced 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.

3.1.3 Modality Alignment

It is common to perform large-scale (compared to instruction-tuning) pre-training on paired data to encourage alignment between different modalities [25, 29, 35, 38], which is prior to the M-IT. The alignment datasets are typically image-text pairs [48–56] or Automatic Speech Recognition (ASR) [57–59] datasets, which all contain text. More specifically, the image-text pairs describe images in the form of natural language sentences, while the ASR datasets comprise transcriptions of speech. A common approach for alignment pre-training is to keep pre-trained modules (*e.g.* visual encoders and LLMs) frozen and train a learnable interface [21, 37, 38], which is illustrated in the following section.

3.1.4 Data

The collection of multimodal instruction-following data is a key to M-IT. The collection methods can be broadly categorized into benchmark adaptation, self-instruction [60], and hybrid composition. We illustrate these three methods sequentially.

Benchmark Adaptation Benchmark datasets are rich sources of high-quality data. Hence, abundant works [23–26, 28, 29, 32, 35] have utilized existing benchmark 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 §3.1.2). 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 [13, 23, 25, 26, 36, 37] 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 2. The other works manually design some seed instructions and use these instructions to prompt GPT to generate more [24, 31, 33].

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 MLLM. There are two common strategies to tackle this problem. The first one is to modify instructions. For example, ChatBridge [29] explicitly declares *short* and *brief* for short-answer data, as well as *a sentence* and *single sentence* for caption data. Similarly, InstructBLIP [23] inserts *short* and *briefly* into instruction templates for public datasets that inherently prefer short responses. The second one is to extend the length of existing answers [36]. For example, M^3IT [36] proposes to rephrase the original answer by prompting ChatGPT with the original question, answer, and context.

Self-Instruction Although existing benchmark datasets can contribute a rich source of data, they usually do not well meet human needs in real-world scenarios, such as multiple rounds of conversations. To tackle this issue, some works collect samples through self-instruction [60], which bootstraps LLMs to generate textual instruction-following data using a few hand-annotated samples. Specifically, some instruction-following samples are hand-crafted as seed examples, after which ChatGPT/GPT-4 is prompted to generate more instruction samples with the seed samples as guidance. LLaVA [21] extends the approach to the multimodal field by translating images into texts of captions and bounding boxes, and prompting GPT-4 to generate new data in the context of seed examples. In this way, an M-IT dataset is constructed, called LLaVA-Instruct-150k. Following this idea, subsequent works such as MiniGPT-4 [13], ChatBridge [29], GPT4Tools [34], and DetGPT [38] develop different M-IT datasets catering for different needs.

Hybrid Composition Apart from the M-IT data, language-only user-assistant conversation data can also be used to improve conversational proficiencies and instruction-following abilities [22, 31, 32, 35]. LaVIN directly constructs a mini-

- <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 2. Instruction templates for VQA datasets, cited from [23]. <Image> and {Question} are the image and the question in the original VQA datasets, respectively.

batch by randomly sampling from both language-only and M-IT data. MultiInstruct [26] 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), sequential instruction tuning (text data followed by multimodal data), and Adapter-based sequential instruction tuning. The empirical results show that mixed instruction tuning is at least not worse than solely tuning on multimodal data.

3.1.5 Modality Bridging

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. Moreover, doing so would take the risk of catastrophic forgetting [61]. Thus, a more practical way is to introduce a learnable interface 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 Interface The learnable interface is responsible for connecting different modalities when freezing the parameters of the pre-trained models. The challenge lies in how to efficiently translate visual content into text that LLM can understand. A common and feasible solution is to leverage a group of learnable query tokens to extract information in a query-based manner [62], which first has been implemented in Flamingo [63] and BLIP-2 [64], and subsequently inherited by a variety of work [23, 25, 42]. Furthermore, some methods use a projection-based interface to close the modality gap [21, 30, 38, 43]. For example, LLavA [21] adopts a simple linear layer to embed image features and MedVInT-TE [30] uses a two-layer multilayer perceptron as a bridge. There are also works that explore a parameter-efficient tuning manner. LLaMA-Adapter [28, 35] introduces a lightweight adapter module in Transformer during training. LaVIN [32] designs a mixture-of-modality adapter to dynamically decide the weights of multimodal embeddings.

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 [35]. Differently, the idea behind the expert models is to convert multimodal inputs into languages without training. In this way, LLMs can understand multimodality by the converted languages indirectly. For example, VideoChat-Text [33] 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 typically cause information loss. As VideoChat-Text [33] points out, transforming videos into textual descriptions distorts spatial-temporal relationships.

3.1.6 Evaluation

There are various metrics to evaluate the performance of the model after M-IT, which can be broadly categorized into two types according to the question genres, including closed-set and open-set.

Closed-set Closed-set questions refer to a type of questions where the possible answer options are predefined and limited to a finite set. The evaluation is usually performed on benchmark-adapted datasets. In this case, the responses can be naturally judged by benchmark metrics [21, 23, 25, 26, 28, 29, 32, 35]. For example, Instruct-

BLIP [23] reports the accuracy on ScienceQA [65], as well as the CIDEr score [66] on NoCaps [67] and Flickr30K [68]. The evaluation settings are typically zero-shot [23, 26, 29, 36] or finetuning [21, 23, 25, 28, 32, 35–37]. 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 downstream tasks. For example, LLaVA [21] and LLaMA-Adapter [28] report finetuned performance on ScienceQA [65]. LLaVA-Med [37] reports results on biomedical VQA [69–71].

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 [39, 40, 72]. For example, Fu *et al*. [73] 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. 10 advanced MLLMs are evaluated with detailed leaderboards and analyses. LAMM-Benchmark [39] is proposed to evaluate MLLMs quantitatively on a variety of 2D/3D vision tasks. Video-ChatGPT [40] proposes a quantitative evaluation framework for video-based conversational models, which incorporates two types of assessments, *i.e*., evaluation of video-based generative performance and zero-shot question-answering.

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 [22] collects a visually related evaluation set to judge capabilities like natural image understanding, diagram and flowchart understanding. Similarly, GPT4Tools [34] 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 [21] proposes to score the responses via GPT-4 in terms of different aspects, such as helpfulness and accuracy. Specifically, 30 images are sampled from the COCO [48] 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 MLLM and GPT-4 are sent to GPT-4 for comparison. Subsequent works follow this idea and prompt ChatGPT [22] or GPT-4 [25, 29, 32, 36, 37] to rate results [22, 25, 29, 32, 37] or judge which one is better [35].

A main issue of GPT-4 based scoring is that currently, its multimodal interface is not publicly available. As a result, GPT-4 can only generate responses based on image-related text content, such as captions or bounding box coordinates, without accessing the image [37]. It thus may be questionable to set GPT-4 as the performance upper bound in this case. An alternative approach is to compare different capabilities of MLLMs through case studies. For example, mPLUG-Owl uses a visually related joke understanding case to compare against GPT-4 [2] and MM-REACT [14]. Similarly, Video-LLaMA [42] offers some cases to demonstrate several capabilities, such as audio-visual co-perception and common-knowledge concept recognition.

Others Some other methods focus on a specific aspect of MLLMs. For instance, MultiInstruct [26] proposes a metric called sensitivity that assesses the model' robustness to varied instructions. Li *et al*. [44] delve into the object hallucination problem and propose a query method POPE to assess performance in this regard. Zhao *et al*. [45] consider safety issues and propose to evaluate the robustness of MLLMs to adversarial attacks.

3.2. 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 [74]. 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 [14, 75, 76]. (2) ICL is usually implemented in a training-free manner [74] and thus can be flexibly integrated into different frameworks at the inference stage. A closely related technique to ICL is instruction-tuning (see §3.1), which is shown empirically to enhance the ICL ability [16].

In the context of MLLM, ICL has been extended to more modalities, leading to Multimodal ICL (M-ICL). Building upon the setting in (§3.1.2), 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 3. 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 [74, 77].

In terms of applications in multimodality, M-ICL is

<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 3. A simplified example of the template to structure an M-ICL query, adapted from [31]. For illustration, we list two in-context examples and a query divided by a dashed line. The {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.

mainly used in two scenarios: (1) solving various visual reasoning tasks [14, 27, 63, 78, 79] and (2) teaching LLMs to use external tools [75, 76, 80]. 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 often text-only and 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 §3.3).

3.3. Multimodal Chain of Thought

As the pioneer work [7] points out, CoT is "a series of intermediate reasoning steps", which has been proven to be effective in complex reasoning tasks [7, 87, 88]. 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 [81, 82, 85, 86] have been proposed to extend the unimodal CoT to Multimodal CoT (M-CoT). We summarize these works as shown in Fig. 3. To begin with, similar to the situation in M-IT (see §3.1), the modality gap needs to be filled (§3.3.1). Then, we introduce different paradigms for acquiring the ability of M-CoT (§3.3.2). Finally, we delineate more specific aspects of M-CoT, including the configuration (§3.3.3) and the formulation of chains (§3.3.4).

3.3.1 Modality bridging

To transfer the success from NLP to multimodal, modality bridging is the first issue to address. There are broadly two ways to achieve this: through the fusion of features or through transforming visual input into textual descriptions. Similar to the case in §3.1.5, we classify them as learnable interface and expert model respectively and discuss them in sequence.

Learnable Interface This approach involves adopting a learnable interface to map visual embedding to the word embedding space. The mapped embeddings can then be taken as a prompt, which is sent to LLMs with other languages to elicit M-CoT reasoning. For example, CoT-PT [81] chains multiple Meta-Nets for prompt tuning to simulate a reasoning chain, where each Meta-Net embeds visual features into a step-specific bias to the prompt. Multimodal-CoT [82] adopts a two-stage framework with a shared Transformer-based structure [89], where visual and textual features interact through cross-attention.

Expert Model Introducing an expert model to translate visual input to textual descriptions is an alternative modality bridging way. For example, ScienceQA [65] adopts an image captioning model and feeds the concatenation of the image captions and original language input to LLMs. Though simple and straightforward, this approach may suffer from information loss in the captioning process [33, 82].

3.3.2 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, ScienceQA [65] constructs a scientific question-answering dataset with lectures and explanations, which can serve as sources of learning CoT reasoning, and finetunes on this proposed dataset. Multimodal-CoT [82] 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 [81] 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.

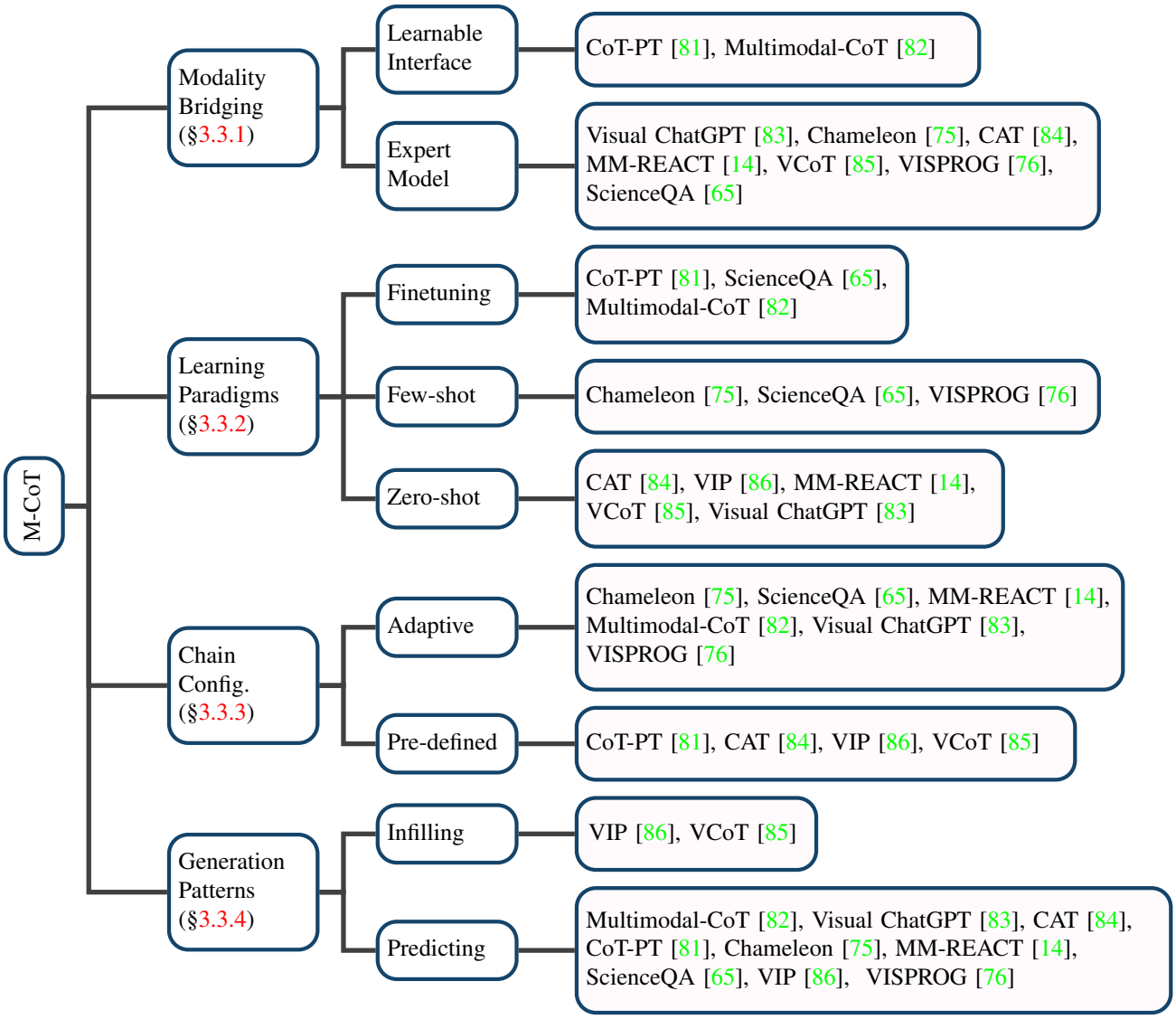

Figure 3. Taxonomy of Multimodal Chain of Thought (M-CoT). Key aspects of M-CoT include modality bridging, learning paradigms, chain configuration, and generation patterns.

In this case, by prompting designed instructions like "Let's think frame by frame" or "What happened between these two keyframes" [85, 86], models learn to leverage the embedded knowledge and the reasoning ability without explicit guidance. Similarly, some works [14, 83] prompt models with descriptions of the task and tool usage to decompose complex tasks into sub-tasks.

3.3.3 Chain Configuration

Chain configuration is an important aspect of reasoning and 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 [14, 65, 75, 76, 82, 83], while the latter setting stops the chains with a pre-defined length [81, 84–86].

3.3.4 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 [85, 86]. In contrast, the predicting-based pattern requires extending the reasoning chains given conditions such as instructions and previous reasoning history [14, 65, 75, 76, 82, 83]. The two types of patterns share

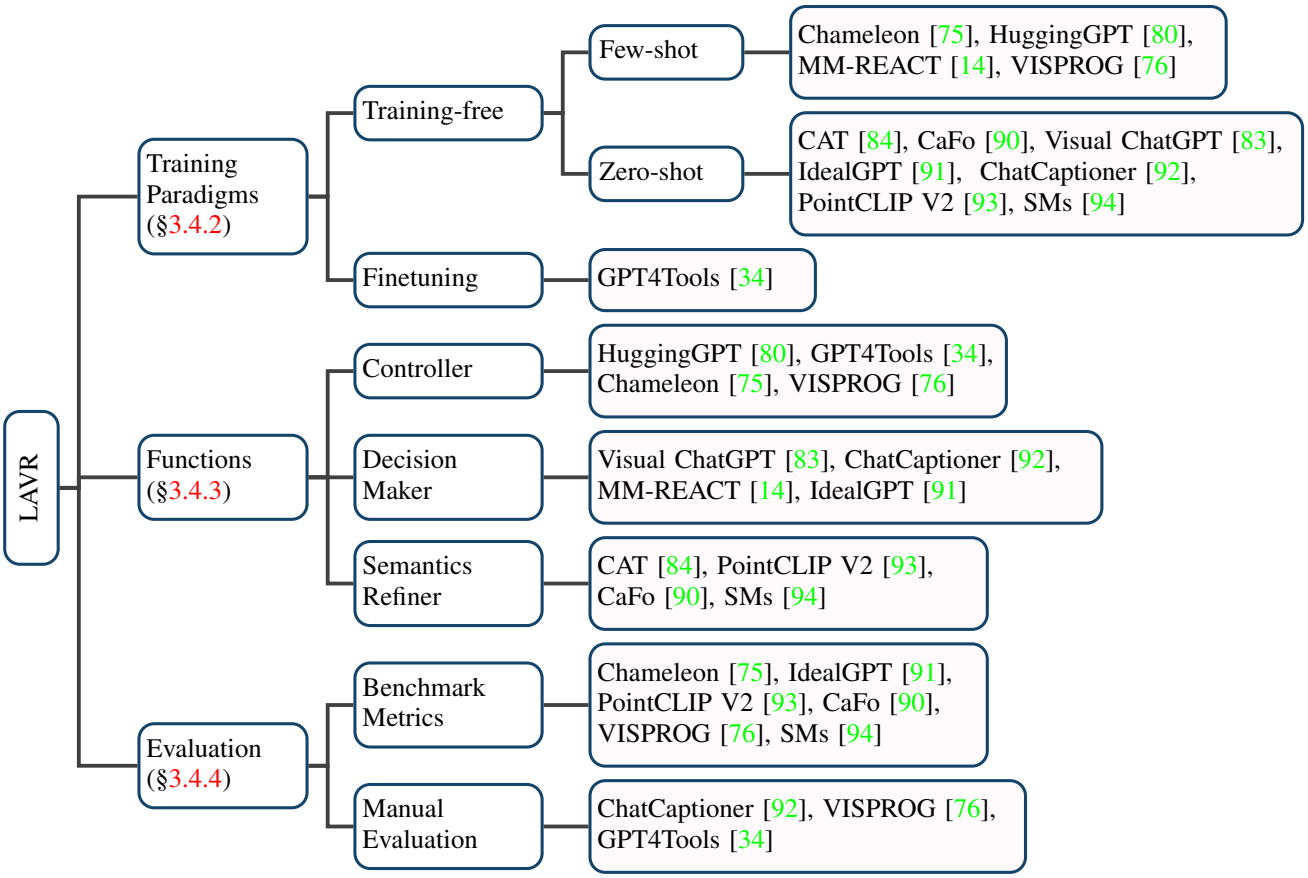

Figure 4. Taxonomy of LLM-Aided Visual Reasoning (LAVR). Key aspects of LAVR include training paradigms, the main functions of LLM, and performance evaluation.

a requirement that the generated steps should be consistent and correct.

3.4. LLM-Aided Visual Reasoning

3.4.1 Introduction

Inspired by the success of tool-augmented LLMs [95–98], some researches have explored the possibilities of invoking external tools [14, 34, 75, 76] or vision foundation models [14, 83, 84, 91, 92, 99] for visual reasoning tasks. Taking LLMs as helpers with different roles, these works build task-specific [84, 90, 93] or general-purpose [14, 75, 76, 80, 83] visual reasoning systems.

Compared with conventional visual reasoning models [100–102], 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 [75, 76, 90, 91, 93, 94]. (2) Emergent abilities. Aided by strong reasoning abilities and abundant knowledge of LLMs, these systems are able to perform complex tasks. For example, given an image, MM-REACT [14] 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 [103, 104]. In contrast, LLM-based systems have the ability to make fine control in a user-friendly interface (*e.g.* click and natural language queries) [84].

The following parts of this section are organized as displayed in Fig. 4: we start with introducing different training paradigms employed in the construction of LLM-Aided Visual Reasoning systems (§3.4.2). Subsequently, we delve into the primary roles that LLMs play within these systems (§3.4.3). Finally, we wrap up our discussion with various types of performance evaluation.

3.4.2 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 and zero-shot models. The few-shot models [14,75,76,80] entail a few hand-crafted in-context samples (see §3.2) 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 [93] prompts GPT-3 to generate descriptions with 3D-related semantics for better alignment with corresponding images. In CAT [84], LLMs are instructed to refine the captions according to user queries.

Finetuning To activate the planning abilities with respect to tool usage and to improve the instruction-following abilities of the system, GPT4Tools [34] introduces the instruction-tuning approach (see §3.1). A new tool-related instruction dataset is collected and used to finetune the model.

3.4.3 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, *i.e.*, the controller and the decision maker, are related to CoT (see §3.3). It is frequently used because complex tasks need to be broken down into intermediate simpler steps. When LLMs act as a controller, 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 [80] or, more directly, the modules to call [34, 75, 76]. For example, VISPROG [76] 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 (see §3.1) examples are used as references [75, 76, 80]. This is closely related to the optimization of reasoning chains (see §3.3), or more specifically, the least-to-most prompting [105] 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 [91]. 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 their rich linguistics and semantics knowledge. Specifically, LLMs are often instructed to integrate information into consistent and fluent natural language sentences [94] or generate texts according to different specific needs [84, 90, 93].

3.4.4 Evaluation

There are two ways to evaluate the performance of LLM-Aided Visual Reasoning systems, namely benchmark metrics [75,76,90,91,93,94] and manual assessment [34,76,92].

Benchmark Metrics A straightforward evaluation way is to test the system on existing benchmark datasets since the metrics can directly reflect how well the model finishes the task. For example, Chameleon [75] is evaluated on complex reasoning benchmarks, including ScienceQA [65] and TabMWP [106]. IdealGPT [91] reports the accuracy on VCR [107] and SNLI-VE [108].

Manual Evaluation Some works adopt manual ratings to evaluate specific aspects of models. For example, ChatCaptioner [92] asks human annotators to judge the richness and correctness of captions generated by different models. GPT4Tools [34] calculates successful rates of thought, action, argument, and the overall successful rate to measure the model's capability in assigning tool usage. VISPROG [76] manually calculates the accuracy when assessing the model on language-guided image editing tasks.

4. 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 still limited in perception capabilities, leading to incomplete or wrong visual information

acquisition [13, 73]. This may be due to the compromise between information capacity and computation burden. More specifically, Q-Former [64] only uses 32 learnable tokens to represent an image, which might induce information loss. Nonetheless, scaling up the token size would inevitably bring a larger computation burden to LLMs, whose input length is usually limited. A potential method is to introduce large vision foundation models like SAM [8] to compress visual information more efficiently [21, 29].

- The reasoning chain of MLLMs may be fragile. For example, Fu *et al*. [73] find that in a math calculation case, although MLLM calculates the correct result, it still delivers a wrong answer due to the broken of reasoning. This indicates that the reasoning ability of a unimodal LLM may not be equal to that of the LLM after receiving visual information. The topic of improving multimodal reasoning is worth investigating.

- The instruction-following ability of MLLMs needs upgrading. After M-IT, some MLLMs fail to generate the expected answer ("yes" or "no") despite an explicit instruction, "Please answer yes or no" [73]. This suggests that instruction tuning may need to cover more tasks to improve generalization.

- The object hallucination issue is widespread [13, 44], which largely affects the reliability of MLLMs. This may be ascribed to insufficient alignment pre-training [13]. Thus, a possible solution is to perform a more fine-grained alignment between visual and textual modalities. The fine granularity refers to the local features of images, which can be obtained by SAM [21, 29], and the corresponding local textual descriptions.

- Parameter-efficient training is needed. Both the existing two modality bridging manners *i.e*., the learnable interface and the expert model, are preliminary explorations to reduce the computation burden. More efficient training methods may unlock more power in MLLMs with limited computational resources.

5. Conclusion

In this paper, we perform a survey of the existing MLLM literature and offer a broad view of its main directions, including three common techniques (M-IT, M-ICL, and M-CoT) and a general framework to build task-solving systems (LAVR). Moreover, we underscore the current research gaps 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 work.

References

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

[2] OpenAI, "Gpt-4 technical report," *arXiv:2303.08774*, 2023. 1, 6

[3] 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

[4] 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

[5] 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, 2

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

[7] 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, 7

[8] 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, 11

[9] 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

[10] 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

[11] 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

[12] 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

[13] 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, 3, 4, 11

[14] 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, 6, 7, 8, 9, 10

[15] 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

[16] 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. 2, 6

[17] 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. 2

[18] 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. 2

[19] 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. 2

[20] 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. 2

[21] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv:2304.08485*, 2023. 2, 3, 4, 5, 6, 11

[22] 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. 2, 3, 4, 6

[23] 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. 2, 3, 4, 5, 6

[24] 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. 2, 3, 4

[25] 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. 2, 3, 4, 5, 6

[26] Z. Xu, Y. Shen, and L. Huang, "Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning," *arXiv:2212.10773*, 2022. 2, 3, 4, 5, 6

[27] 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. 2, 3, 7

[28] 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. 2, 3, 4, 5, 6

[29] 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. 2, 3, 4, 5, 6, 11

[30] 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. 2, 3, 4, 5

[31] 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. 2, 3, 4, 7

[32] 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. 2, 3, 4, 5, 6

[33] 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. 2, 3, 4, 5, 7

[34] 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. 2, 3, 4, 6, 9, 10

[35] 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. 2, 3, 4, 5, 6

[36] 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^3it: A large-scale dataset towards multi-modal multilingual instruction tuning," *arXiv:2306.04387*, 2023. 3, 4, 6

[37] 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. 3, 4, 6

[38] 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. 3, 4, 5

[39] Z. Yin, J. Wang, J. Cao, Z. Shi, D. Liu, M. Li, L. Sheng, L. Bai, X. Huang, Z. Wang *et al.*, "Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark," *arXiv:2306.06687*, 2023. 3, 6

[40] 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. 3, 6

[41] C. Lyu, M. Wu, L. Wang, X. Huang, B. Liu, Z. Du, S. Shi, and Z. Tu, "Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration," *arXiv:2306.09093*, 2023. 3

[42] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," *arXiv:2306.02858*, 2023. 3, 5, 6

[43] 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. 3, 4, 5

[44] 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. 3, 6, 11

[45] 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. 3, 6

[46] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *ICCV*, 2015. 4

[47] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *CVPR*, 2015. 4

[48] 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. 4, 6

[49] V. Ordonez, G. Kulkarni, and T. Berg, "Im2text: Describing images using 1 million captioned photographs," *NeurIPS*, 2011. 4

[50] 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. 4

[51] 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. 4

[52] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, "Laion-400m: Open dataset of clip-filtered 400 million image-text pairs," *arXiv:2111.02114*, 2021. 4

[53] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *IJCV*, 2017. 4

[54] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *ICCV*, 2015. 4

[55] J. Wu, H. Zheng, B. Zhao, Y. Li, B. Yan, R. Liang, W. Wang, S. Zhou, G. Lin, Y. Fu *et al.*, "Ai challenger: A large-scale dataset for going deeper in image understanding," *arXiv:1711.06475*, 2017. 4

[56] J. Gu, X. Meng, G. Lu, L. Hou, N. Minzhe, X. Liang, L. Yao, R. Huang, W. Zhang, X. Jiang *et al.*, "Wukong: A 100 million large-scale chinese cross-modal pre-training benchmark," *NeurIPS*, 2022. 4

[57] 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. 4

[58] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *O-COCOSDA*, 2017. 4

[59] J. Du, X. Na, X. Liu, and H. Bu, "Aishell-2: Transforming mandarin asr research into industrial scale," *arXiv:1808.10583*, 2018. 4

[60] 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. 4

[61] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv:1312.6211*, 2013. 5

[62] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020. 5

[63] 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. 5, 7

[64] 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. 5, 11

[65] 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. 6, 7, 8, 10

[66] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *CVPR*, 2015. 6

[67] 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. 6

[68] 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. 6

[69] X. He, Y. Zhang, L. Mou, E. Xing, and P. Xie, "Pathvqa: 30000+ questions for medical visual question answering," *arXiv:2003.10286*, 2020. 6

[70] 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. 6

[71] 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. 6

[72] P. Xu, W. Shao, K. Zhang, P. Gao, S. Liu, M. Lei, F. Meng, S. Huang, Y. Qiao, and P. Luo, "Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models," *arXiv:2306.09265*, 2023. 6

[73] 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*, 2023. 6, 11

[74] 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. 6

[75] 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. 6, 7, 8, 9, 10

[76] T. Gupta and A. Kembhavi, "Visual programming: Compositional visual reasoning without training," in *CVPR*, 2023. 6, 7, 8, 9, 10

[77] 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. 6

[78] 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. 7

[79] M. Tsimpoukelli, J. L. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill, "Multimodal few-shot learning with frozen language models," *NeurIPS*, 2021. 7

[80] 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. 7, 9, 10

[81] 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. 7, 8

[82] 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. 7, 8

[83] 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. 8, 9

[84] 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. 8, 9, 10

[85] 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. 7, 8

[86] 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. 7, 8

[87] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *arXiv:2205.11916*, 2022. 7

[88] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," *arXiv:2210.03493*, 2022. 7

[89] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017. 7

[90] 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. 9, 10

[91] 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. 9, 10

[92] 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. 9, 10

[93] 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. 9, 10

[94] 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. 9, 10

[95] A. Parisi, Y. Zhao, and N. Fiedel, "Talm: Tool augmented language models," *arXiv:2205.12255*, 2022. 9

[96] 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. 9

[97] 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. 9

[98] 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. 9

[99] 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. 9

[100] 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. 9

[101] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *CVPR*, 2019. 9

[102] 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. 9

[103] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "Stylenet: Generating attractive visual captions with styles," in *CVPR*, 2017. 9

[104] A. Mathews, L. Xie, and X. He, "Senticap: Generating image descriptions with sentiments," in *AAAI*, 2016. 9

[105] 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. 10

[106] P. Lu, L. Qiu, K.-W. Chang, Y. N. Wu, S.-C. Zhu, T. Rajpurohit, P. Clark, and A. Kalyan, "Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning," *arXiv:2209.14610*, 2022. 10

[107] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, "From recognition to cognition: Visual commonsense reasoning," in *CVPR*, 2019. 10

[108] N. Xie, F. Lai, D. Doran, and A. Kadav, "Visual entailment: A novel task for fine-grained image understanding," *arXiv:1901.06706*, 2019. 10