This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

CogAgent: A Visual Language Model for GUI Agents

Wenyi Hong[1*] Weihan Wang[1*] Qingsong Lv[2] Jiazheng Xu[1*] Wenmeng Yu[2]
Junhui Ji[2] Yan Wang[2] Zihan Wang[1*] Yuxiao Dong[1] Ming Ding[2†] Jie Tang[1†]
[1]Tsinghua University [2]Zhipu AI
{hwy22@mails, jietang@}.tsinghua.edu.cn, ming.ding@zhipuai.cn

Abstract

People are spending an enormous amount of time on digital devices through graphical user interfaces (GUIs), e.g., computer or smartphone screens. Large language models (LLMs) such as ChatGPT can assist people in tasks like writing emails, but struggle to understand and interact with GUIs, thus limiting their potential to increase automation levels. In this paper, we introduce CogAgent, an 18-billion-parameter visual language model (VLM) specializing in GUI understanding and navigation. By utilizing both low-resolution and high-resolution image encoders, CogAgent supports input at a resolution of 1120×1120*, enabling it to recognize tiny page elements and text. As a generalist visual language model, CogAgent achieves the state of the art on five text-rich and four general VQA benchmarks, including VQAv2, OK-VQA, Text-VQA, ST-VQA, ChartQA, infoVQA, DocVQA, MM-Vet, and POPE. CogAgent, using only screenshots as input, outperforms LLM-based methods that consume extracted HTML text on both PC and Android GUI navigation tasks—Mind2Web and AITW, advancing the state of the art. The model and codes are available at* https://github.com/THUDM/CogVLM.

1. Introduction

Autonomous agents in the digital world are ideal assistants that many modern people dream of. Picture this scenario: You type in a task description, then relax and enjoy a cup of coffee while watching tasks like booking tickets online, conducting web searches, managing files, and creating PowerPoint presentations get completed automatically.

Recently, the emergence of agents based on large language models (LLMs) is bringing us closer to this dream. For example, AutoGPT [33], a 150,000-star open-source project, leverages ChatGPT [29] to integrate language understanding with pre-defined actions like Google searches and local file operations. Researchers are also starting to develop agent-oriented LLMs [7, 42]. However, the potential of purely language-based agents is quite limited in real-world scenarios, as most applications interact with humans through Graphical User Interfaces (GUIs), which are characterized by the following perspectives:

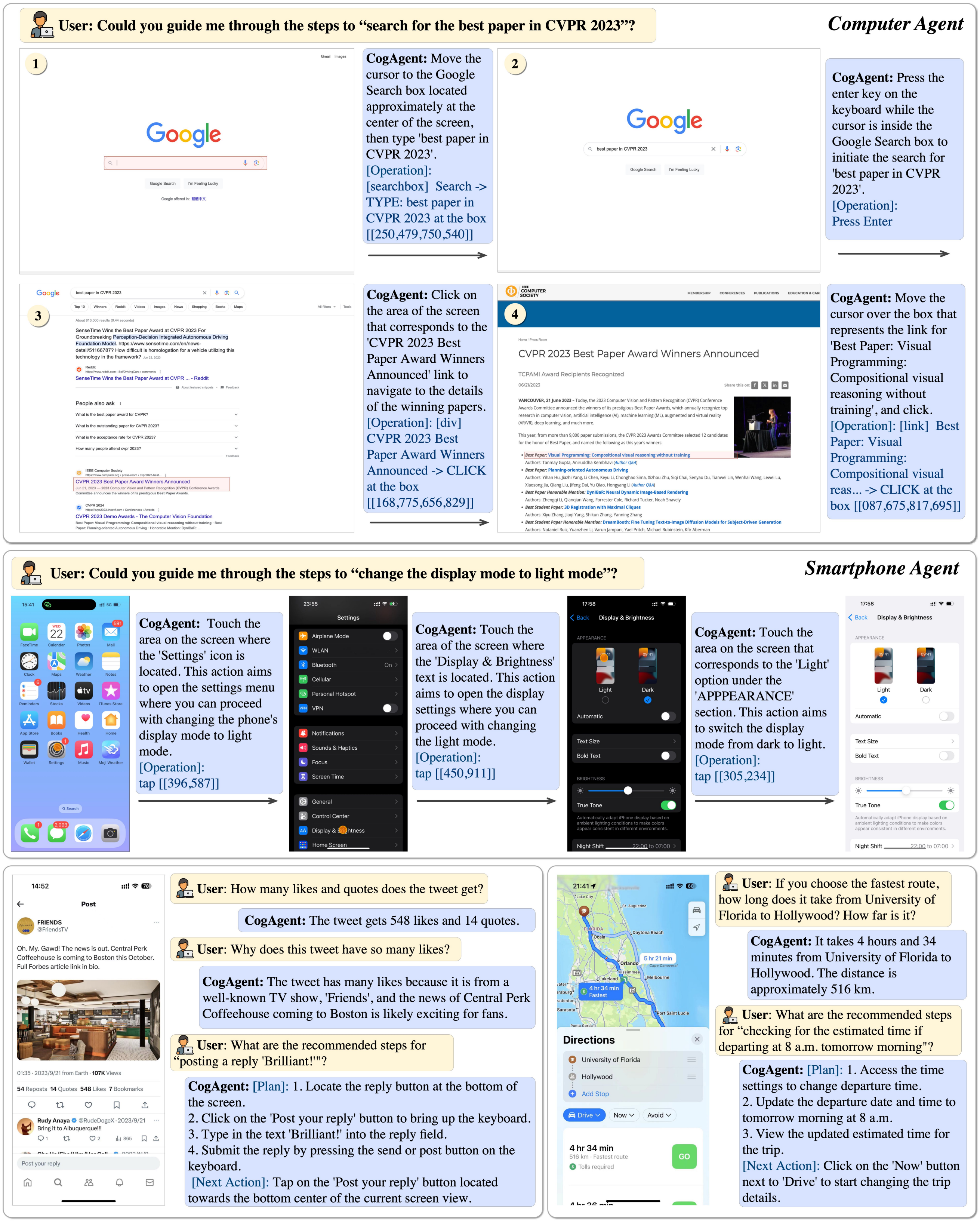
- Standard APIs for interaction are often lacking.
- Important information including icons, images, diagrams, and spatial relations are difficult to directly convey in words.
- Even in text-rendered GUIs like web pages, elements like canvas and iframe cannot be parsed to grasp their functionality via HTML.

Agents based on visual language models (VLMs) have the potential to overcome these limitations. Instead of relying exclusively on textual inputs such as HTML [28] or OCR results [31], VLM-based agents directly perceive visual GUI signals. Since GUIs are designed for human users, VLM-based agents can perform as effectively as humans, as long as the VLMs match human-level vision understanding. In addition, VLMs are also capable of skills such as extremely fast reading and programming that are usually beyond the reach of most human users, extending the potential of VLM-based agents. A few prior studies utilized visual features merely as auxiliaries in specific scenarios. e.g. WebShop [39] which employs visual features primarily for object recognition purposes. With the rapid development of VLM, can we naturally achieve universality on GUIs by relying solely on visual inputs?

In this work, we present CogAgent, a visual language foundation model specializing in GUI understanding and planning while maintaining a strong ability for general cross-modality tasks. By building upon CogVLM [38]—a recent open-source VLM, CogAgent tackles the following challenges for building GUI agents:

- **Training Data.** Most current VLMs are pre-trained on datasets like LAION [32], consisting of natural images on the Web. However, we notice that the GUI

[*] Work was done when interned at Zhipu AI.
[†] Corresponding authors



Computer Agent

User: Could you guide me through the steps to "search for the best paper in CVPR 2023"?

1

CogAgent: Move the cursor to the Google Search box located approximately at the center of the screen, then type 'best paper in CVPR 2023'. [Operation]: [searchbox] Search -> TYPE: best paper in CVPR 2023 at the box [[250,479,750,540]]

2

CogAgent: Press the enter key on the keyboard while the cursor is inside the Google Search box to initiate the search for 'best paper in CVPR 2023'. [Operation]: Press Enter

3

CogAgent: Click on the area of the screen that corresponds to the 'CVPR 2023 Best Paper Award Winners Announced' link to navigate to the details of the winning papers. [Operation]: [div] CVPR 2023 Best Paper Award Winners Announced -> CLICK at the box [[168,775,656,829]]

4

CVPR 2023 Best Paper Award Winners Announced

TCPAMI Award Recipients Recognized

CogAgent: Move the cursor over the box that represents the link for 'Best Paper: Visual Programming: Compositional visual reasoning without training', and click. [Operation]: [link] Best Paper: Visual Programming: Compositional visual reas... -> CLICK at the box [[087,675,817,695]]

Smartphone Agent

User: Could you guide me through the steps to "change the display mode to light mode"?

CogAgent: Touch the area on the screen where the 'Settings' icon is located. This action aims to open the settings menu where you can proceed with changing the phone's display mode to light mode. [Operation]: tap [[396,587]]

CogAgent: Touch the area of the screen where the 'Display & Brightness' text is located. This action aims to open the display settings where you can proceed with changing the light mode. [Operation]: tap [[450,911]]

CogAgent: Touch the area on the screen that corresponds to the 'Light' option under the 'APPPEARANCE' section. This action aims to switch the display mode from dark to light. [Operation]: tap [[305,234]]

Post

FRIENDS @FriendsTV

Oh. My. Gawd! The news is out. Central Perk Coffeehouse is coming to Boston this October. Full Forbes article link in bio.

01:35 · 2023/9/21 from Earth · 107K Views

54 Reposts 14 Quotes 548 Likes 7 Bookmarks

Rudy Anaya @RudeDogeX · 2023/9/21
Bring it to Albuquerque!!!

Post your reply

User: How many likes and quotes does the tweet get?

CogAgent: The tweet gets 548 likes and 14 quotes.

User: Why does this tweet have so many likes?

CogAgent: The tweet has many likes because it is from a well-known TV show, 'Friends', and the news of Central Perk Coffeehouse coming to Boston is likely exciting for fans.

User: What are the recommended steps for "posting a reply 'Brilliant!'"?

CogAgent: [Plan]: 1. Locate the reply button at the bottom of the screen.
2. Click on the 'Post your reply' button to bring up the keyboard.
3. Type in the text 'Brilliant!' into the reply field.
4. Submit the reply by pressing the send or post button on the keyboard.
[Next Action]: Tap on the 'Post your reply' button located towards the bottom center of the current screen view.

Directions

University of Florida
Hollywood
Add Stop

Drive · Now · Avoid

4 hr 34 min
516 km · Fastest route
Tolls required

GO

User: If you choose the fastest route, how long does it take from University of Florida to Hollywood? How far is it?

CogAgent: It takes 4 hours and 34 minutes from University of Florida to Hollywood. The distance is approximately 516 km.

User: What are the recommended steps for "checking for the estimated time if departing at 8 a.m. tomorrow morning"?

CogAgent: [Plan]: 1. Access the time settings to change departure time.
2. Update the departure date and time to tomorrow morning at 8 a.m.
3. View the updated estimated time for the trip.
[Next Action]: Click on the 'Now' button next to 'Drive' to start changing the trip details.

Figure 1. Samples of visual agents generated by CogAgent. More samples are demonstrated in the Appendix.

images share a different distribution from natural images. We thus construct a large-scale annotated dataset about GUIs and OCR for continual pre-training.

- **High-Resolution vs. Compute.** In GUIs, tiny icons and text are ubiquitous, and it is hard to recognize them in commonly-used 224 × 224 resolution. However, increasing the resolution of input images results in significantly long sequence length in language models. For example, a 1120 × 1120 image corresponds to a sequence of 6400 tokens if the patch size is 14, demanding excessive training and inference compute. To address this, we design a cross-attention branch that allows for a trade-off between the resolution and the hidden size within a proper computation budget. Specifically, we propose to combine the original large ViT [12] (4.4B parameters) used in CogVLM [38] and a new small *high-resolution cross-module* (with image encoder of 0.30B parameters) to jointly model visual features.

Our experiments show that:

- CogAgent tops popular GUI understanding and decision-making benchmarks, including AITW [31] and Mind2Web [10]. To the best of our knowledge, this is the first time that a generalist VLM can outperform LLM-based methods with extracted structured text.
- Though CogAgent focuses on GUIs, it achieves state-of-the-art generalist performance on nine visual question-answering benchmarks including VQAv2 [1], OK-VQA [23], TextVQA [34], ST-VQA [4], ChartQA [24], infoVQA [26], DocVQA [25], MM-Vet [41], and POPE [19].
- The separated design of high- and low-resolution branches in CogAgent significantly lows the compute cost for consuming high-resolution images, e.g., the number of the floating-point operations (FLOPs) for CogAgent-18B with 1120 × 1120 inputs is less than half that of CogVLM-17B with its default 490 × 490 inputs.

CogAgent is open-sourced at https://github.com/THUDM/CogVLM. It represents an effort to promote the future research and application of AI agents, facilitated by advanced VLMs.

2. Method

In this section, we will first introduce the architecture of CogAgent, especially the novel high-resolution cross-module, and then illustrate the process of pre-training and alignment in detail.

2.1. Architecture

The architecture of CogAgent is depicted in Fig. 2. We build our model based on a pre-trained VLM (on the right side of the image), and propose to add a cross-attention module to process high-resolution input (on the left side of the image). As our base VLM, We select CogVLM-17B [38], an open-sourced and state-of-the-art large vison-language model. Specifically, We employ EVA2-CLIP-E [35] as the encoder for low-resolution images (224×224 pixels), complemented by an MLP adapter that maps its output into the feature space of the visual-language decoder. The decoder, a pre-trained language model, is enhanced with a visual expert module introduced by Wang et al. [38] to facilitate a deep fusion of visual and language features. The decoder processes a combined input of the low-resolution image feature sequence and text feature sequence, and autoregressively outputs the target text.

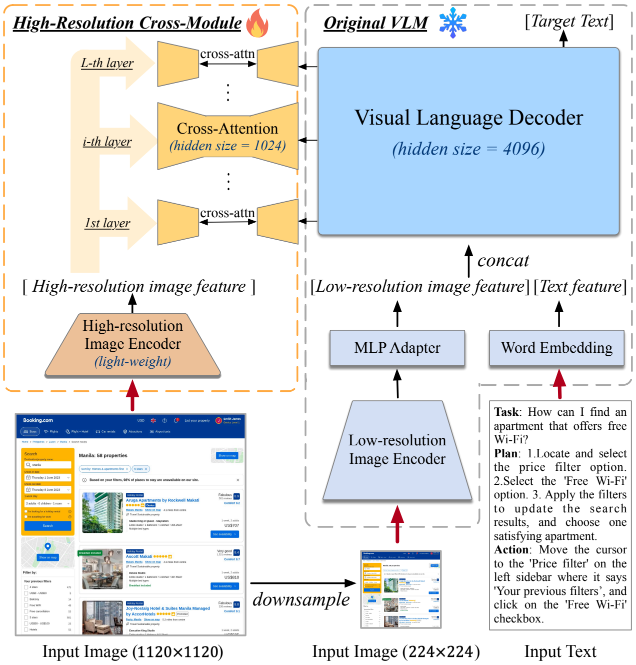
Similar to most VLMs, the original CogVLM can only accommodate images of relatively low resolution (224 or 490), which hardly meets the demands of GUI where the screen resolution of computers or smartphones is typically 720p (1280 × 720 pixels) or higher. It is a common problem among VLMs, e.g. LLaVA [21] and PALI-X [8] are pre-trained at a low resolution of 224 × 224 on the general domain. The primary reason is that high-resolution image brings prohibitive time and memory overhead: VLMs usually concatenate text and image feature sequence as input to the decoder, thus the overhead of self-attention module is quadratic to the number of visual tokens (patches), which is quadratic to the image's side length. There are some initial attempts to reduce costs for high-resolution images. For instance, Qwen-VL [2] proposes a position-aware vision-language adapter to compress image features, but only reduces sequence length by four and has a maximum resolution of 448 × 448. Kosmos-2.5 [22] adopts a Perceiver Resampler module to reduce the length of the image sequence. However, the resampled sequence is still long for self-attention in the large visual-language decoder (2,048 tokens), and can only be applied to restricted text recognition tasks.

Therefore, we propose a novel *high-resolution cross-module* as a potent complement to the existing structure for enhancing understanding at high resolutions, which not only maintains efficiency confronting high-resolution images, but also offers flexible adaptability to a variety of visual-language model architectures.

2.2. High-Resolution Cross-Module

The structural design of *high-resolution cross-module* is mainly based on the following observations:

1. At a modest resolution such as 224 × 224, images can depict most objects and layouts effectively, yet



Task: How can I find an apartment that offers free Wi-Fi?
Plan: 1.Locate and select the price filter option. 2.Select the 'Free Wi-Fi' option. 3. Apply the filters to update the search results, and choose one satisfying apartment.
Action: Move the cursor to the 'Price filter' on the left sidebar where it says 'Your previous filters', and click on the 'Free Wi-Fi' checkbox.

Figure 2. Model architecture of CogAgent. We adopt CogVLM as the original VLM.

the resolution falls short in rendering text with clarity. Hence, our new high-resolution module should emphasize text-related features, which are vital for understanding GUIs.

2. While pre-trained VLMs in general domain often need large hidden sizes (e.g. 4,096 in PALI-X and CogVLM, 5,120 in LLaVA), VLMs tailored for text-centered tasks like document OCR require smaller hidden sizes to achieve satisfying performance (e.g. 1,536 in Kosmos-2.5 and Pix2Struct [16]). This suggests that text-related features can be effectively captured using smaller hidden sizes.

As shown in Fig. 2, the high-resolution cross-module acts as a new branch for higher-resolution input, which accepts images of size 1120 × 1120 pixels in our implementation. Different from the original low-resolution input branch, the high-resolution cross-module adopts a much smaller pre-trained vision encoder (visual encoder of EVA2-CLIP-L [35] in our implementation, 0.30B parameters), and uses cross-attention of a small hidden size to fuse high-resolution image features with every layer of VLLM decoder, thus reducing the computational cost. To be concrete, for an input image, it is resized to 1120 × 1120 and 224×224 and fed into the high-resolution cross-module and the low-resolution branch respectively, then encoded into image feature sequences X_{hi} and X_{lo} with two distinct-sized image encoders in parallel. The visual language decoder retains its original computations, while the only change is to integrate a cross-attention between X_{hi} and hidden states in every decoder layer.

Formally, suppose that the input hidden states of the i-th attention layer in the decoder are $X_{\text{in}_i} \in \mathbb{R}^{B \times (L_{I_{\text{lo}}}+L_T) \times D_{\text{dec}}}$, and the output hidden states of cross-module's image encoder are $X_{\text{hi}} \in \mathbb{R}^{B \times (L_{I_{\text{hi}}}) \times D_{\text{hi}}}$, where B is the batch size, $L_{I_{\text{lo}}}$, $L_{I_{\text{hi}}}$ and L_T are the lengths of the low-resolution image, high-resolution image and text sequences, D_{dec} and D_{hi} is the hidden size of the decoder and high-resolution encoder's output respectively. Each layer's attention procedure can be formulated as

$$X_i' = \text{MSA}(\text{layernorm}(X_{\text{in}_i})) + X_{\text{in}_i}, \tag{1}$$

$$X_{\text{out}_i} = \text{MCA}(\text{layernorm}(X_i'), X_{\text{hi}}) + X_i', \tag{2}$$

where MSA and MCA represent multi-head self-attention with visual expert and multi-head cross-attention, while X_i' and X_{out_i} represent their respective output features with the residual connection. To implement cross-attention between them, we add learnable transformation matrices $W^i_{K_{\text{cross}}}, W^i_{V_{\text{cross}}} \in \mathbb{R}^{D_{\text{hi}} \times D_{\text{cross}}}$ to get $K^i_{\text{cross}} = X_{\text{hi}} W^i_{K_{\text{cross}}}$, $V^i_{\text{cross}} = X_{\text{hi}} W^i_{V_{\text{cross}}} \in \mathbb{R}^{L_{I_{\text{hi}}} \times D_{\text{cross}}}$, and $W^i_{Q_{\text{cross}}} \in \mathbb{R}^{D_{\text{dec}} \times D_{\text{cross}}}$ to get $Q^i_{\text{cross}} = X_i' W^i_{Q_{\text{cross}}} \in \mathbb{R}^{(L_{I_{\text{lo}}}+L_T) \times D_{\text{cross}}}$ in every decoder layer. With the residual connection in Eq. 2, the cross-attention with high-resolution images can be perceived as a complement to the features of low-resolution images, thereby effectively utilizing the previous pre-trained model in low resolution.

Computational complexity. Let the number of attention head be H_{cross} and H_{dec} in cross-attention and self-attention, and the dimension of each head be $d_{\text{cross}} = D_{\text{cross}}/H_{\text{cross}}$ and $d_{\text{dec}} = D_{\text{dec}}/H_{\text{dec}}$. If using our high-resolution cross-module, the computational complexity of attention is

$$\begin{aligned} \text{T}_{\text{improved}} = \mathbf{O}\big((&L_{I_{\text{lo}}} + L_T) L_{I_{\text{hi}}} H_{\text{cross}} d_{\text{cross}} \\ &+ (L_{I_{\text{lo}}} + L_T)^2 H_{\text{dec}} d_{\text{dec}}\big). \end{aligned} \tag{3}$$

Note that d_{cross} and H_{cross} can be flexibly adjusted according to computational budget and model performance. If not utilizing the high-resolution cross-module and directly substituting low-resolution images with high-resolution ones, the computational complexity would be

$$\text{T}_{\text{original}} = \mathbf{O}\big((L_{I_{\text{hi}}} + L_T)^2 H_{\text{dec}} d_{\text{dec}}\big). \tag{4}$$

In our implementation, $d_{\text{cross}} = 32$, $H_{\text{cross}} = 32$, and we inherits $d_{\text{dec}} = 128$, $H_{\text{dec}} = 32$ from CogVLM-17B. Both high- and low-resolution encoders patchify images with 14 × 14-pixel patches, thus $L_{I_{\text{hi}}} = 6400$, $L_{I_{\text{lo}}} = 256$. Our method leads to at least $\frac{L_{I_{\text{hi}}}+L_T}{L_{I_{\text{lo}}}+L_T} = \frac{6400+L_T}{256+L_T} \times$ acceleration which is a stringent lower bound (refer to Appendix for detailed derivation), and reduces memory overhead at the same time.

2.3. Pre-training

To enhance the model's ability to comprehend high-resolution images and adapt it for GUI application scenarios, we focus our pre-training efforts on the following aspects: the capability to recognize texts of various sizes, orientations, and fonts in high-resolution images, the grounding ability of text and objects in the image, and a specialized understanding capability for GUI imagery such as web page. We divide our pre-train data into three parts based on the aforementioned aspects, with samples in the Appendix. All the pre-training data are derived from publicly available datasets. The construction methods are detailed below.

Text recognition. Our data includes (1) Synthetic renderings with text from language pre-training dataset (80M). This is similar to the Synthetic Document Generator in Kim et al. [15], with text of varying font, size, color and orientation, and diverse image background from LAION-2B [32]. (2) Optical Character Recognition (OCR) of natural images (18M). We collect natural images from COYO [6] and LAION-2B [32] and employ Paddle-OCR [13] to extract the texts and their bounding boxes, and filter out images with no text boxes. (3) Academic documents (9M). We follow Nougat [5] to construct image-text pairs including text, formula and tables from the source code (LaTeX) release on arXiv. For (1)(3), we apply the same data augmentation as Nougat. For (2), we additionally employed more aggressive rotation and flipping data augmentation techniques.

Visual grounding. It is imperative for GUI agents to possess the capability to accurately comprehend and locate diverse elements within images. We follow CogVLM [38] to use a constructed visual grounding dataset of 40M images with image-caption pairs sampled from LAION-115M [18], which associate entities in the caption with bounding boxes to indicate their positions. The format of the bounding box is $[[x_0, y_0, x_1, y_1]]$, where (x_0, y_0) and (x_1, y_1) represent the coordinates of upper-left and lower-right corners which are normalized to $[000, 999]$. If multiple objects are indicated by a single noun phrase, their boxes are separated by semicolons in double square brackets.

GUI imagery. Our approach innovatively addresses the scarcity and limited relevance of GUI images in datasets like LAION and COYO, which predominantly feature natural images. GUI images, with their distinct elements such as input fields, hyperlinks, icons, and unique layout characteristics, require specialized handling. To boost the model's capability in interpreting GUI imagery, we have conceptualized two pioneering GUI grounding tasks: (1) GUI Referring Expression Generation (REG) – where the model is tasked with generating HTML code for DOM (Document Object Model) elements based on a specified area in a screenshot, and (2) GUI Referring Expression Comprehension (REC) – which involves creating bounding boxes for given DOM elements. To facilitate robust training in GUI grounding, we have constructed the CCS400K (Common Crawl Screenshot 400K) dataset. This extensive dataset is formed by extracting URLs from the latest Common Crawl data, followed by capturing 400,000 web page screenshots. Alongside these screenshots, we compile all visible DOM elements and their corresponding rendered boxes using Playwright[1], supplementing the dataset with 140 million REC and REG question-answer pairs. This rich dataset ensures comprehensive training and understanding of GUI elements. To mitigate the risk of overfitting, we employ a diverse range of screen resolutions for rendering, selected randomly from a list of commonly used resolutions across various devices. Additionally, to prevent the HTML code from becoming overly extensive and unwieldy, we perform necessary data cleaning by omitting redundant attributes in the DOM elements, following the method outlined in [16].

We also incorporate publicly available text-image datasets including LAION-2B and COYO-700M (after removing the broken URLs, NSFW images, and images with noisy captions and political bias) during pre-training.

We pre-train our CogAgent model for a total of 60,000 iterations with a batch size of 4,608 and a learning rate of 2e-5. We freeze all parameters except the newly added high-resolution cross-module for the first 20,000 steps, resulting in a total number of 646M (3.5%) trainable parameters, then additionally unfreeze the visual expert in CogVLM for the next 40,000 steps. We warm up with curriculum learning by first training on easier text recognition (synthetic renderings and OCR on natural images) and image captioning, then sequentially incorporating harder text recognition (academic document), grounding data and web page data, as we observed that it leads to faster convergence and more stable training in our preliminary experiments.

2.4. Multi-task Fine-tuning and Alignment

To enhance our model's performance for diverse tasks and ensure it aligns with free-form human instructions in the GUI setting, we further fine-tune our model on a broad range of tasks. We manually collected over two thousand screenshots from mobile phones and computers, each annotated with screen elements, potential tasks, and methods of operation in the question-answering format by human annotators (details illustrated in the Appendix). We also utilize Mind2Web [10] and AITW [31], datasets focusing on web and Android behaviors which comprise tasks, sequences of actions and corresponding screenshots, and convert them into a natural language question-and-answer format using GPT-4. Besides, we incorporate multiple publicly available visual question-answering (VQA) datasets encompassing a variety of tasks into our alignment dataset. We unfreeze all model parameters during this stage and train for 10k iterations with a batch size of 1024 and a learning rate of 2e-5.

[1] https://playwright.dev

Method	General VQA		Text-rich VQA					
	VQAv2	OKVQA	OCRVQA	TextVQA	STVQA	ChartQA	InfoVQA	DocVQA
task-specific fine-tuning models								
Pix2Struct [16]	-	-	-	-	-	58.6	40.0	76.6
BLIP-2 [18]	82.2	59.3	72.7	-	-	-	-	-
PALI-X-55B [8]	86.0	66.1	75.0	71.4	79.9	70.9	49.2	80.0
CogVLM$_{\text{task-specific}}$ [38]	84.7	64.7	74.5	69.7	-	-	-	-
generalist models								
UReader [40]	-	57.6	-	-	-	59.3	42.2	65.4
Qwen-VL [2]	79.5	58.6	**75.7**	63.8	-	65.7	-	65.1
Qwen-VL-chat [2]	78.2	56.6	70.5	61.5	-	66.3	-	62.6
Llava-1.5 [20]	80.0	-	-	61.5	-	-	-	-
Fuyu-8B [3]	74.2	60.6	-	-	-	-	-	-
CogVLM$_{\text{generalist}}$ [38]	83.4	58.9	74.1	68.1	-	-	-	-
CogAgent (Ours)	**83.7**	**61.2**	75.0	**76.1**	**80.5**	**68.4**	**44.5**	**81.6**

Table 1. **Performance on Visual Question Answering benchmarks.** Bold text indicates the best score among the generalist category, and underlined text represents the best score across both generalist and task-specific categories.

3. Experiments

To evaluate the foundational capabilities and GUI-related performance of our model, we conduct extensive experiments on a broad range of datasets. First, we conduct evaluations on eight VQA benchmarks, as well as MM-Vet [41] and POPE [19], which validate our model's enhanced ability in visual understanding, especially on those that are reliant on text recognition. Then we evaluate our model on Mind2Web and AITW datasets, as the representative of two major GUI scenarios — computers and smartphones.

3.1. Foundational Visual Understanding

We first extensively evaluate CogAgent's foundational visual understanding capability across eight VQA benchmarks, covering a wide range of visual scenes. The benchmarks can be divided into two categories: general VQA, including VQAv2 [1] and OK-VQA [23], and text-rich VQA, including TextVQA [34], OCR-VQA [27], ST-VQA [4], DocVQA [25], InfoVQA [26] and ChartQA [24]. The latter category emphasizes the understanding of visually-situated text, including documents, charts, photographs containing text, etc. To demonstrate the model's versatility and robustness across tasks, our model is fine-tuned collectively on all datasets simultaneously, yielding a single generalist model which is then evaluated across all datasets.

The results are presented in Tab. 1. For general VQA, CogAgent achieves state-of-the-art generalist results on both datasets. For text-rich VQA, CogAgent achieves state-of-the-art results on 5 out of 6 benchmarks, significantly surpassing generalist competitors (TextVQA+8.0, ChartQA+2.1, InfoVQA+2.3, DocVQA+16.2), even outperforming the task-specific state-of-the-art models on TextVQA(+4.7), STVQA(+0.6) and DocVQA(+1.6). Notably, compared to the generalist results of CogVLM which CogAgent is initially based on, CogAgent demonstrates certain improvements on both general and Text-rich VQA tasks, suggesting the efficacy of our proposed model architecture and training methods.

Furthermore, we conducted zero-shot tests of our model on the challenging MM-Vet [41] and POPE [19] datasets, both of which are instrumental in gauging the multi-modal capabilities and the generalization performance in complex tasks including conversation question-answering, detailed descriptions, complex reasoning tasks. MM-Vet is designed with six core tasks to assess multi-modal models' proficiency in handling intricate assignments, and POPE-adversarial models on their susceptibility to hallucinations. Our experimental results, as detailed in Table 2, showcase that our model significantly outperforms other existing models in both datasets. Notably, on the MM-Vet dataset, our model achieved a remarkable score of 52.8, surpassing the closest competitor, LLaVA-1.5, by a substantial margin (+16.5). On the POPE-adversarial evaluation, our model attained a score of 85.9, demonstrating superior handling of

Method	LLM	MM-Vet	POPE$_{\text{adv}}$
BLIP-2 [18]	Vicuna-13B	22.4	-
Otter [17]	MPT-7B	24.7	-
MiniGPT4 [44]	Vicuna-13B	24.4	70.4
InstructBLIP [9]	Vicuna-13B	25.6	77.3
LLaVA [21]	LLaMA2-7B	28.1	66.3
LLaMA-Adapter v2 [14]	LLaMA-7B	31.4	-
DreamLLM [11]	Vicuna-7B	35.9	76.5
LLaVA-1.5 [20]	Vicuna-13B	36.3	84.5
Emu [36]	LLaMA-13B	36.3	-
CogAgent (Ours)	Vicuna-7B	**52.8**	**85.9**

Table 2. **Evaluation of CogAgent on conversational style QA and hallucination assessment.** Regarding the POPE dataset, we use its adversarial subset for this evaluation.

Method	cross-task	cross-website	cross-domain	overall
Representations of screen inputs: HTML				
GPT-3.5[29] (few-shot)	18.6	17.4	16.2	17.4
GPT-4[30]† (few-shot)	36.2	30.1	26.4	30.9
Flan-T5$_{XL}$ [10]	52.0	38.9	39.6	43.5
LLaMA2-7B[37]	52.7	47.1	50.3	50.1
LLaMA2-70B[37]	55.8	51.6	55.7	54.4
Representations of screen inputs: Image				
Qwen-VL[2]	12.6	10.1	8.0	10.2
CogVLM[38]	37.1	23.4	26.3	23.9
CogAgent (Ours)	**62.3**	**54.0**	**59.4**	**58.2**

Table 3. **Performance on Mind2Web.** † denotes element selection from top-10 element candidates, others from top-50, following Deng et al. [10]. Results for GPT-3.5 and GPT-4 are from Deng et al. [10].

hallucinations compared to other models.

3.2. GUI Agent: Computer Interface

We evaluate CogAgent on Mind2Web, a dataset for web agents that includes over 2,000 open-ended tasks collected from 137 real-world websites across 31 domains. Given the task description, current webpage snapshot and previous actions as inputs, agents are expected to predict the subsequent action. We follow the setting of Deng et al. [10] in our experiments, and report step success rate (step SR) metric.

Several language models were evaluated on this benchmark. For instance, AgentTuning [42] and MindAct [10] evaluated Llama2-70B and Flan-T5-XL in a fine-tuned setting, and GPT-3.5 and GPT-4 in a in-context learning setting. However, limited by the input modality of language models, these models could only use heavily cleansed HTML as the representation of screen inputs. To the best of our knowledge, no visually-based web agents have been experimented with on this benchmark.

We fine-tune our model on the train set and evaluate on three out-of-domain subsets, i.e. cross-website, cross-domain, and cross-task. We additionally fine-tune LLaMA2-7B and LLaMA2-70B as the baseline of fine-tuned LLMs, and adopt the same HTML cleansing process as Deng et al. [10] to construct HTML input. The results are presented in Sec. 3.2. Compared to other methods, our approach achieved significant performance improvements across all three subsets, surpassing LLaMA2-70B, which is nearly 4× the scale of CogAgent, by 11.6%, 4.7%, and 6.6%, respectively. This reflects not only the capability of our model but also the advantages of employing a visual agent in computer GUI scenarios.

3.3. GUI Agent: Smartphone Interface

To evaluate our model on diverse smartphone interfaces and tasks, we utilize Android in the Wild (AITW) dataset [31] , a large-scale dataset for Android device agents. It comprises 715k operation episodes covering varying Android versions and device types. Each episode in the dataset consists of a goal described in natural language, followed by a sequence of actions and corresponding screenshots. The training target is to predict the next action based on the given goal, historical actions, and the screenshot. For each action, models are required to predict the exact action type; for tap, swipe and type, models are further required to predict the position, direction, and content to be typed, respectively.

Method	GoogleApp	Install	WebShop	General	Single	Overall
Representations of screen inputs: textual description (OCR+icon)						
GPT-3.5[29](few-shot)	10.47	4.38	8.42	5.93	9.39	7.72
LLaMA2-7B[37]†	30.99	35.18	19.92	28.56	27.35	28.40
Representations of screen inputs: image						
Auto-UI$_{unified}$[43]	71.37	76.89	70.26	**68.24**	84.58	74.27
CogAgent (Ours)	**74.95**	**78.86**	**71.73**	65.38	**93.49**	**76.88**

Table 4. **Performance on Android in the Wild (AITW) dataset.** † represents models individually fine-tuned on each subset, while others are unified models across all subsets. The results of LLaMA2 and GPT-3.5 are from Zhan and Zhang [43].

We conduct comparisons with two kinds of baselines: language models using the textual description of UI elements provided by the original dataset (text OCR and icon) as the representations of screen inputs[2], and visual-language models using images as the screen inputs. We simultaneously fine-tuned on all the subsets, yielding a unified model which is then evaluated on all test sets. As the GoogleApps subset is 10-100 times larger than other subsets, we downsample it to 10% to avoid data imbalance.

Results are shown in Tab. 4. CogAgent achieves state-of-the-art performance compared to all previous methods. In comparison to language-based methods, our model surpasses both baselines by a large margin. In comparison to the visual-language baseline, Auto-UI, our model achieves +2.61 improvements in the overall performance. In instances of inaccuracies, we randomly sample hundreds of cases, and upon reassessment, more than 40% are determined to be correct (refer to the appendix for details). This diversity arises from the multiple valid pathways inherent in mobile interactions, resulting in a range of responses.

4. Ablation Study

To thoroughly comprehend the impact of various components in the methodology, we conduct ablation studies on two aspects, model architecture and training data. The evaluation is conducted on diverse datasets, including multiple VQA datasets (STVQA, OCRVQA, DocVQA) and a web agent dataset (Mind2Web). For VQA datasets, we fine-

[2] Some Android applications may have View Hierarchy which is more friendly to language-based agents, but most of them tend to be poor quality or missing altogether. Therefore, as a large-scale, general-purpose dataset, AITW retained the results of OCR detection and icon detection as textual representations of screenshots.

tune the model on four datasets together for 3,000 iters with a batch size of 1,280, and report the generalist score; for Mind2Web, models are fine-tuned for 2,400 iters with a batch size of 128 and use top-10 setting. Training iterations are fewer than those in the main experiment, aiming to control variables within the constraints of a limited budget.

4.1. Model Architecture

To ascertain the efficacy of the high-resolution cross-module, we compare it with directly increasing the resolution using the original model architecture of CogVLM, and ablate on two perspectives: computational efficiency and model performance.

To measure computational overhead, we use floating point operations (FLOPs) as the metric, and conduct experiments on multiple resolutions including 224, 490, 756, and 1120. From Fig. 3 we can see that, as the image resolution increases, models that use a high-resolution cross-module experience only a modest rise in computational overhead, demonstrating an almost linear relationship with the number of image patches. In contrast, using the original model structure, i.e. CogVLM, leads to a significant increase in the number of FLOPs at higher resolutions. Its FLOPs can even be more than 10 times higher compared to employing a cross-module at a resolution of 1120, which is the resolution utilized by CogAgent.

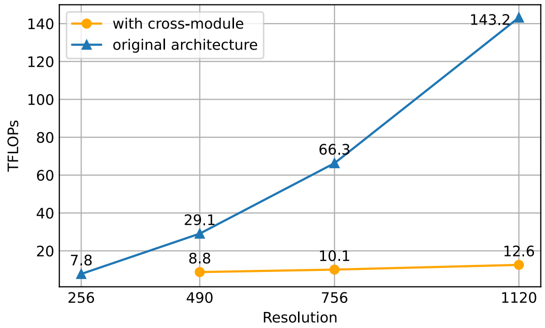


Figure 3. Comparison of FLOPs during forward propagation for different model architectures and resolutions.

We further compare the model performance in Tab. 5, which indicates that models with high-resolution cross-module at the resolution of 756 require only 1/2 of the computational resources used by the original structure at the resolution of 490, while delivering significantly better performance. Additionally, the high-resolution cross-module allows for further increasing models' acceptable resolution within a limited computational budget, thereby yielding additional performance improvements.

4.2. Pre-train Data

We further conduct an ablation study on pre-training data, which is an integral part of training visual agents. Building upon the image-caption data commonly used in visual-language training, we sequentially add OCR data (denoted as Cap+OCR), as well as GUI and grounding data (denoted as All). The results in Tab. 6 indicate that each part of data broadly contributes to enhanced performance. Notably, web and grounding data have a significant impact on the Mind2Web dataset, underscoring the importance of constructing domain-specific pre-train data in the training of GUI agents.

high-res module	base res	cross res	STVQA	OCRVQA	DocVQA	Mind2Web	training time/it (s)	TFLOPs
✗	224	—	48.0	70.2	28.6	34.6	2.36	7.77
✗	490	—	68.1	74.5	57.6	40.7	6.43	29.14
✓	224	756	73.6	74.2	62.3	40.7	3.57	10.08
✓	224	1120	78.2	75.9	74.1	41.4	5.17	12.56

Table 5. Ablation study on model architecture. Training time is evaluated on A800 with the batch size of 8. Models are pre-trained with Caption+OCR data.

pre-train data	base res	cross res	STVQA	OCRVQA	DocVQA	Mind2Web
Cap	490	—	68.1	74.5	57.6	38.6
Cap+OCR	490	—	72.5	75.0	59.8	40.7
Cap+OCR	224	1120	78.2	75.9	74.1	41.4
All	224	1120	79.4	75.6	76.4	54.2

Table 6. Ablation study on pre-train data with sequentially added image captioning, OCR and other pre-train data.

5. Conclusion

We introduce CogAgent, a VLM-based GUI agent with enhanced pre-train data construction and efficient architecture for high-resolution input. CogAgent achieves state-of-the-art performance on a wide range of VQA and GUI benchmarks, and will be open-sourced. CogAgent is an initial exploration of VLM-based GUI agent, and still has some shortcomings, e.g. imprecise output coordinates and incapability of processing multiple images, necessitating further research.

Acknowledgments

This work is supported by Technology and Innovation Major Project of the Ministry of Science and Technology of China under Grant 2022ZD0118600, Natural Science Foundation of China (NSFC) 62277033 and the New Cornerstone Science Foundation through the XPLORER PRIZE. It also got partial support from the National Engineering Laboratory for Cyberlearning and Intelligent Technology, Beijing Key Lab of Networked Multimedia, Daimler Greater China Ltd. -Tsinghua University Joint Institute for Sustainable Mobility,Tsinghua University(Department of Computer Science and Technology)-Siemens Ltd., China Joint Research Center for Industrial Intelligence and Internet of Things (JCIIOT) and a research fund from Zhipu AI.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 3, 6, 11

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 3, 6, 7

[3] Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. Introducing our multimodal models, 2023. 6

[4] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301, 2019. 3, 6, 12

[5] Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*, 2023. 5

[6] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022. 5

[7] Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*, 2023. 1

[8] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023. 3, 6

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 6

[10] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *arXiv preprint arXiv:2306.06070*, 2023. 3, 5, 7, 12

[11] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023. 6

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[13] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020. 5

[14] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 6

[15] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer, 2022. 5

[16] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023. 4, 5, 6

[17] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*, 2023. 6

[18] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 5, 6

[19] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 3, 6, 11

[20] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 6

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 3, 6

[22] Tengchao Lv, Yupan Huang, Jingye Chen, Lei Cui, Shuming Ma, Yaoyao Chang, Shaohan Huang, Wenhui Wang, Li Dong, Weiyao Luo, et al. Kosmos-2.5: A multimodal literate model. *arXiv preprint arXiv:2309.11419*, 2023. 3

[23] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019. 3, 6, 11

[24] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 3, 6, 12

[25] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 3, 6, 12

[26] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa.

In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022. 3, 6, 12

[27] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE, 2019. 6, 11

[28] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. 1

[29] OpenAI. Introducing chatgpt. 2022. 1, 7

[30] OpenAI. Gpt-4 technical report, 2023. 7

[31] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Android in the wild: A large-scale dataset for android device control. *arXiv preprint arXiv:2307.10088*, 2023. 1, 3, 5, 7, 13

[32] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 1, 5

[33] Significant-Gravitas. Autogpt. https://github.com/Significant-Gravitas/AutoGPT, 2023. 1

[34] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 3, 6, 11

[35] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 3, 4

[36] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023. 6

[37] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 7

[38] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023. 1, 3, 5, 6, 7

[39] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022. 1

[40] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *arXiv preprint arXiv:2310.05126*, 2023. 6

[41] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 3, 6, 11

[42] Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. abs/2310.12823, 2023. 1, 7, 12

[43] Zhuosheng Zhan and Aston Zhang. You only look at screens: Multimodal chain-of-action agents. abs/2309.11436, 2023. 7, 13

[44] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 6