Magma: A Foundation Model for Multimodal AI Agents

Supplementary Material

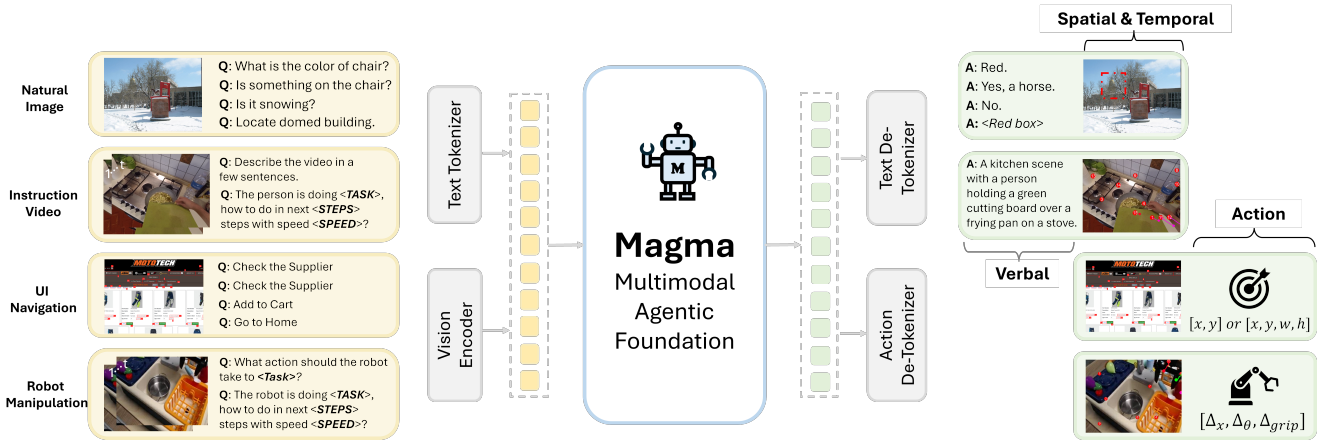


Figure 1. **Magma pretraining pipeline**. For all training data, texts are tokenized into tokens, while images and videos from different domains are encoded by a shared vision encoder. The resulted discrete and continuous tokens are then fed into a LLM to generate the outputs in verbal, spatial and action types. Our proposed method reconcile the multimodal understanding and action prediction tasks.

A. Pretraining and Finetuning

Setting	Pretraining	Finetuning		
		UI	Image/Video	Real Robot
batch size	1024	32		
base learning rate	1e-5	1e-5	1e-5	1e-5
learning rate scheduler	Constant	Cosine	Cosine	Constant
training epochs	3	3	1	20
optimizer	adamw	adamw	adamw	adamw
Image Resolution	512	768	768	256
Number of Crops	4 or 1	4	4 or 1	1

Table 1. Experimental settings pretraining and finetuning of `Magma` models. We maximally use either 32 Nvidia H100s or 64 AMD MI300 GPUs for all training jobs.

We illustrate the pretraining architecture in Fig. 1. It comprises a vision encoder and a text tokenizer to encode the visual and text inputs, respectively. We pretrain our `Magma` model using a combination of natural images, instructional videos, UI navigation and robot manipulation data. For all the model variants, we use the same training recipe as shown in Table 1. To handle different image resolutions from different datasets, we also use a multi-crop strategy to enable batch forward for a given minibatch, though the ConvNext vision backbone can naturally support arbitrary resolutions. Specifically, for our pretraining, we use 512 as the base image size, and resize an input image maximally to 4 crops for UI and image pretraining data, while use 1 crop for video and robotics data.

For downstream finetuning, we following common practice to tune the pretrained magma model as shown in Table 1 right. As mentioned above, the vision encoder can be effortlessly adapted to different image resolutions required for different tasks.

B. Datasets

B.1. Pretraining Data

Due to space constraints, we briefly introduced the datasets for our pretraining in Sec 4.1 of our main submission. To ensure the reproducibility of our pretraining stage, we provide additional details of our pretraining data below.

B.1.1. UI Navigation

Our pretraining data related to UI agent are sourced from two datasets, SeeClick [7] and Vision2UI [15]. We further process these source data by adding marks on screenshots to provide grounded supervisions.

SeeClick. We generally follow the original procedure and make the following modifications to associate with the Set of Mark [46] strategy. For each webpage screenshot, multiple (text, bounding_box) pairs are available. Therefore, we directly overlay all the bounding boxes with corresponding marks on the screenshot. For each mobile screenshot, only a single (text, bounding_box) pair is available in the SeeClick data. To enrich the pairs, we incorporate additional pairs from the RICO dataset [9], and employ an OCR tool to obtain text boxes. Finally, we display the enriched bounding boxes along with their corresponding marks on the mobile screenshot.

Vision2UI. We consider all bounding boxes whose "content" property is not null. To prevent the marks from over-

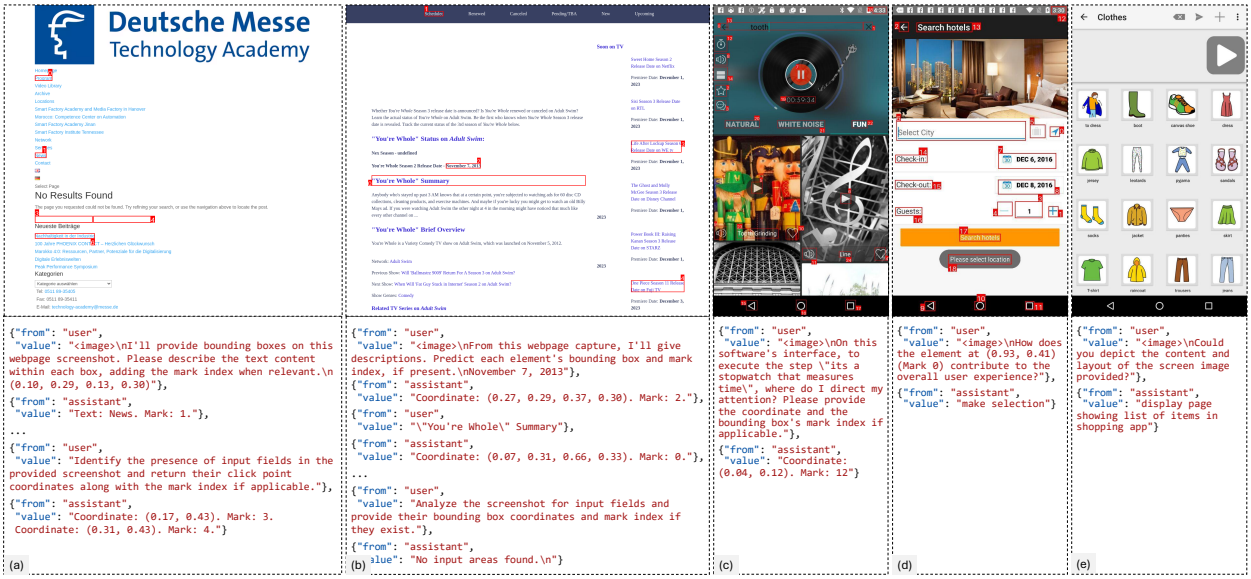


Figure 2. **Training samples in our Magma-PT-UI.** It covers a wide range of action grounding and UI understanding tasks including: (a) Given the bounding box or point coordinates as the query, assistant should return the natural language description or the content. (b) Given the natural language or the exact content as the query, assistant should return the value of the bounding box coordinates.. (c) Given the natural language as the query, assistant should return the value of the point coordinate. (d) Widget captioning. (e) UI summarization.

Source	Task	Size
SeeClick-Web	text_2_point	271K
	text_2_bbox	54K
	point_2_text	54K
	bbox_2_text	54K
SeeClick-Mobile	text_2_point	274K
	text_2_bbox	56K
	UI summarization	48K
	widget captioning	42K
Visison2UI	input_2_point	980K
	input_2_bbox	982K
	text_2_point	794K
	text_2_bbox	774K
	point_2_text	199K
	bbox_2_text	193K
Magma-PT-UI (Ours)	Mixed	2.8M

Table 2. Statistics of UI related pretraining data.

whelming the main content of the webpage, we sample bounding boxes with varying probabilities based on their "type" property. Specifically, we assign a sampling weight of 0.5 to boxes of type `h1`, `h2`, `a`, `button`, `option`, and `nav` with 0.5, while other types are weighted at 0.1. Given the high importance of input areas for interaction, we include boxes of type `input` directly without sampling for mark plotting. After obtaining the elements of high interest, we apply similar tasks as SeeClick [7] to produce the instruction data, including (a) grounding task, which involves two forms: predicting center point coordinates (text_2_point) and predicting bounding box (text_2_bbox); (b) generating text for elements, categorized into predicting text based on the coordinates of center points (point_2_text) or bounding boxes (bbox_2_text); and further introduce the task of (C) locating input fields, including predicting center point coordinates (input_2_point) and bounding box coordinates (input_2_bbox) of the input fields.

Given a webpage, since the first two categories of tasks are grounding or generating texts for the same group of web elements, we further weight the four subtasks, *i.e.*, (text_2_point), (text_2_bbox), (point_2_text), and (bbox_2_text) with [0.4, 0.4, 0.1, 0.1], and sample only one of them to construct the pretraining data. Similarly, we sample one subtask from (input_2_point) and (input_2_bbox) with equal probabilities. We merge the sampled subtasks from the same webpage into one example to improve training efficiency. We denote the full pretraining data related to UI by Magma-PT-UI, and list the sizes of individual subsets in Table 2 and show a few examples in Fig. 2.

B.1.2. Instructional Videos

As mentioned in the main submission, we curate the supervisions from human instructional videos to learn the agentic capability for our model. To cover different scenarios, we considered both 3rd point view videos and egocentric videos. In particular, we start with Epic-Kitchen [8] video data sets considering that their text annotations are relatively high quality. Afterwards, we expand to Something-Soomething v2 [34] to include more human-object interac-

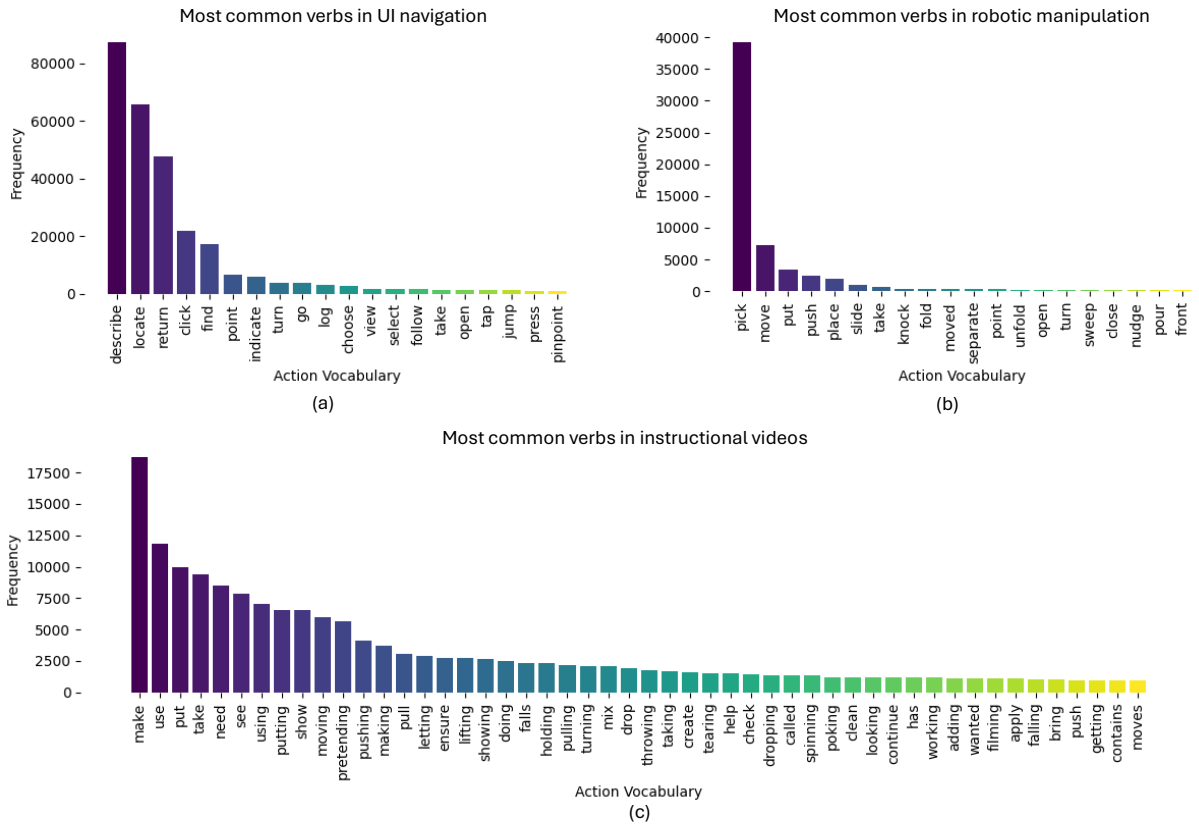


Figure 3. **Action distributions in three types of action-oriented pretraining datasets**. (a) UI Navigation; (b) Robotic Manipulation; (c) Instructional Videos.

tions, and Ego4D [14] and other related instructional videos for scaling up.

Epic-Kitchen [8]. Epic-Kitchen contains 495 egocentric videos recorded by 32 participants in kitchen rooms. Each video contains a number of segments labeled with narrations, start and end frame ids. However, the original video narrations (*e.g.*, "open door") are too coarse to depict the human actions in a certain time frame. For the videos in Epic-Kitchen, we apply the video preprocessing method as discussed in Sec 4.2 of our main submission. Concretely, for each of the original video segments in the dataset, we run PySceneDetect to detect the temporal boundaries and split them into sub-segments. During our model pretraining, the textual annotations are used in two ways. Our model is asked to predict the detailed description in the first frame. In addition, they are used as the task description as input to the model for predicting the traces of marks.

Sth-Sth-v2 [34], **Ego4D** [14]. The Sth-Sth v2 dataset is a comprehensive collection of labeled video clips featuring humans performing predefined actions with everyday objects. The list of action classes spans a wide variety of atomic actions, including but not limited to "pushing something from right to left", "throwing something" and "covering something with something". In total, the dataset contains 220,847 seconds-long video clips. To create our pretraining data, we only leverage the videos in the train and validation splits. This amounts to around 160K video clips. We note that we do not use PySceneDetect for Sth-Sth v2 since the original video clips have been highly curated.

The Ego4D dataset is a large-scale egocentric dataset that contains approximately 3,025 hours of videos. It comprises over 3,670 hours of video footage captured from wearable cameras across a diverse environments and activities. The dataset spans a wide range of real-world scenarios, including daily activities and social interactions. Given the duration of these videos can span over 30 minutes, we leverage the original dense caption annotations that are provided to split each videos into seconds-long segments with consistent views.

Segment and CLIP-score filtering As the point tracking system works in a short time window, we begin by using the annotations provided, curated or otherwise, to split each video into segments, and then run PySceneDetect [4] to further break each segment into short video clips with consis-

tent shots. However, the detected video clips may not always be relevant to their associated text annotations. Thus, we use the pretrained CLIP [40] visual and text encoders to compute the cosine similarity score between each clip and text pair, and filter out clips with < 0.25 scores.

Reliability of CoTracker. To determine the generalizability of such traces, we examine the reliability of CoTracker before running the algorithm on all our pretraining data. We note that CoTracker was already well validated on multiple video datasets such as TAP-Vid [11] and PointOdyssey [53] in the original paper. In this work, we proposed comprehensive strategies to handle scene transition and camera motions in videos (Alg. 2 in main paper), which effectively scale to datasets like Ego4D and other instructional videos (Fig 3). To further validate the reliability of ToM, we quantitatively evaluated the traces on a subset of YouCook2-BB [54] with box annotations by humans. We extract the traces from each annotated box and count the number of future traces still falling into the box 1 second forward. On 1320 clips, we got a precision of **0.89**, indicating that the traces reliably capture temporal motions.

B.1.3. Robotic Manipulation

We follow the training recipe in OpenVLA [23] to prepare our pretraining data for robotics manipulation. Specifically, we take the data mixture "siglip-224px+mx-oxe-magic-soup" as in OpenVLA, which gives us 9.4M image-language-action triplets, extracted from 326K trajectories, from 23 separate datasets.

B.1.4. Multimodal Image Understanding

We simply include the 1.2M synthetic image-text pairs in ShareGPT4V [5] and 665K image instruction tuning data collected by LLaVA-1.5 [29] as our multimodal image pretraining data. The former helps our pretrained model to have a global understanding of visual contents, while the latter helps to get the model familiar with various types of human instructions. We denote this dataset by Magma-PT-Image.

B.1.5. Data Statistics

Given our goal of training a general vision-language-action foundation model, we analyze the distribution of verbs present in the text annotations of the UI and robotic manipulation as well as instructional video datasets in Figure 3. We see that the text annotations in the UI navigation component contain many helpful verbs that help guide agents to achieve a specific task such as "locate" and "turn". This is complemented by the more action-oriented words in the vocabulary of the robot manipulation component, including "pick", "push" and "slide". Such annotations are especially valuable in helping our `Magma` model to learn to reason about interactions with everyday objects. Finally, we also scale up the amount of training data and diversity of verbs by including data from instructional videos (Figure 3c). As evidenced by the relatively high frequency of words such as "lifting" and "throwing", such annotations can be very beneficial for gaining a stronger understanding the of temporal dynamics involved in common activities. More importantly, the diversity of activities present in these datasets can be effective at helping the model generalize better to a larger variety of tasks.

B.2. Downstream Data

B.2.1. UI Agent Navigation

We evaluated the UI grounding and navigation capability mainly on three datasets, ScreenSpot [7], Mind2Web [10] and AITW [41].

ScreenSpot is a benchmark used to evaluate the UI action grounding proposed in [7]. It consists of 600 screenshots images associated with 1.2K instructions spanning iOS, Android, macOS, Windows, and web pages. The evalaution covers both text based elements and a variety of widgets and icons. To evaluate the zero-shot action grounding performance for our model, we use OmniParser [31] to help parse the screenshot and propose actionable regions/icons/buttons. We used the sample code and default settings provided in the official repo. For these candidate regions, we overlay numeric marks and ask our model to pick one.

Mind2Web is first proposed in [10] for text-based web agent. For fair comparison among vision-based web agent, we follow the protocol proposed in SeeClick [7]. Given a webpage, we convert it into a screenshot associated with ground-truth bounding boxes to which the actions should be applied. As the original screenshot of the full website is usually out of the scope of display. We follow a similar way as in [7] to crop the region of interests centering around the ground truth boxes, which gives us a local screenshot as wide as original webpage but with maximal height 1344. To propose the candidate marks for our model, we directly exploit the candidate ranks provided in Mind2Web, and use the top 30 candidates for evaluation.

AITW is a dataset originally collected in [41] for navigation of the android UI. The original dataset contains up to 715K trajectories, resulting in 5.7M screenshots. In our experiments, to examine the efficient finetuning performance, we alteratively follow the same protocol in SeeClick [7] and include a much smaller number of training samples. Specifically, there are 545, 688, 306, 700, 700 instructions from General/Install/GoogleApps/Single/WebShopping, respectively. 80% of each split is used for training and the remainder is used for evaluation. Instead of finetuning our model for each category, we jointly finetune our pretrained `Magma` on the combined data and evaluate across all categories using a single model.

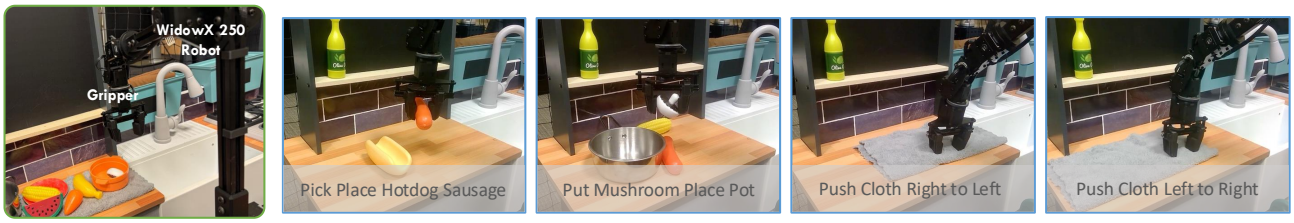


Figure 4. **Real robot setup.** Magma is deployed on a WidowX 250 robot arm to perform a sequence of kitchen manipulation tasks including object pick-place and soft manipulation.

B.2.2. Robot Manipulation

Simulator. We employ SimplerEnv [27] as the main testbed for our learned robot policy. As we do not need to tune our model on the simulated trajectories, we simply report the numbers following the protocol proposed in the original work.

Real-world Setting. We design four tabletop manipulation tasks for our physical WidowX-250 robot setup as shown in Fig. 4. As with BridgeData-v2, the RGB image observations from the robot are captured using a stationary third-person camera, maintaining a resolution of 256×256. For finetuning our pretrained `Magma` model, we collect approximately 50 robot demonstration trajectories for each task as our finetuning dataset. Our experimental design includes classic soft object manipulation and pick-and-place operations tasks. Detailed language instructions for the designed tasks are presented below. For each trial, we randomize the initial location of the target object and include 2-3 random distracting objects (e.g., corn, eggplant) in the scene. For reproducibility, we release the collected robot trajectories.

Tasks included in the finetuning dataset:

- **Hot dog assembly**: Pick up the hot dog sausage from the desk and place it into the bun. The trial is counted as success only when the robot successfully grasps the sausage and accurately places it within the hot dog bun.
- **Mushroom placement**: Pick up the mushroom and place it into the pot. The trial is counted as success only when the robot correctly grasps the mushroom and places it into the cooking pot without dropping or misaligning it.
- **Cloth pushing**: Push the cloth from right to left across the surface. The trial is counted as success only when the robot successfully manipulates the cloth in the specified direction without disturbing other objects on the surface.

Unseen task for evaluating generalization:

- **Bidirectional cloth manipulation**: Push the cloth in both directions while maintaining its shape. This task examines the model's spatial understanding and reasoning capabilities, as it requires generalization from unidirectional pushing in the training data to bidirectional manipulation in novel scenarios.

Dataset	Size	Domain
ShareGPT [43]	40K	Text
ShareGPT4V [5]	39K	General
LLaVA-Instruct [28]	158K	General
LAION-GPT4V [25]	11K	General
VQAv2 [13]	83K	General VQA
GQA [16]	72K	General VQA
OKVQA [42]	9K	Knowledge VQA
OCRVQA [38]	80K	OCR VQA
ChartQA [33]	7K	Chart VQA
DVQA [17]	16K	Chart VQA
DocVQA [35]	10K	Document VQA
AI2D [20]	2K	Infographic VQA
SynthDog-EN [22]	20K	Document Understanding
A-OKVQA	66K	Knowledge VQA
RefCOCO [49]	48K	Grounding Desc.
VG [24]	86K	Referring Exp.
InfographicsVQA [36]	24k	Infographic VQA
ChartQA (Aug) [33]	20k	Chart VQA
FigureQA [18]	20k	Chart/Figure VQA
TQA [21]	1.5k	Textbook VQA
ScienceQA [30]	5k	Textbook VQA
Magma-SFT-Image (Ours)	820k	Mixed

Table 3. A detailed breakdown of our 820k Magma image instruction tuning data used in our multimodal image understanding experiments shown in Table 5 in our main submission.

B.2.3. Image Instruction Tuning

We show a breakdown of our 820k Magma image instruction tuning data in Table 3. As the 760k image instruction tuning data used in LLaVA-1.6 [29] is not released, we follow their guidance to curate 748k public available data including ShareGPT [43], LLaVA-Instruct [28], ShareGPT4V [5], LAION-GPT4V [25], VQAv2 [12], GQA [16], OKVQA [32], OCRVQA [38], ChartQA [33], DVQA [17], DocVQA [35], AI2D [20], SynthDog-EN [22], A-OKVQA [42], RefCOCO [19] and VG [24]. To complement the claimed "improved reasoning, OCR and world knowledge", we resort to a few other open-sourced datasets including InfoGraphicsVQA [36], augmented ChartQA [33], FigureQA [18], TQA [21] and ScienceQA [30]. We denote the full set by Magma-SFT-Image.

Method	Backbone	DoM Tree	Image	General	Install	GoogleApps	Single	WebShopping	Overall
GPT-4V-SeeAct† [52]	GPT-4V [39]		✓	34.1	39.4	40.0	46.2	38.2	39.6
GPT-4V-ReAct† [47]	GPT-4V [39]		✓	36.2	42.5	46.6	49.1	39.2	42.7
GPT-4V-OmniParser [31]	GPT-4V [39]	✓	✓	48.3	57.8	51.6	77.4	52.9	57.7
Fuyu-8B‡	Fuyu-8B [2]		✓	-	45.9	40.0	47.2	40.8	-
Fuyu-8B-GUI [6]	Fuyu-8B [2]		✓	-	50.9	41.6	45.7	43.8	-
MiniCPM-V‡	MiniCPM-V [48]		✓	-	50.2	45.1	56.2	44.0	-
MiniCPM-V-GUI [6]	MiniCPM-V [48]		✓	-	62.3	46.5	67.3	57.5	-
Qwen-VL♮	Qwen-VL [1]		✓	49.5	59.9	46.9	64.7	50.7	54.3
SeeClick [7]	Qwen-VL [1]		✓	54.0	66.4	54.9	63.5	57.6	59.3
Magma-8B (Ours)	LLaMA3 [37]		✓	**61.5**	**73.2**	**62.7**	**77.5**	**61.7**	**67.3**

Table 4. **Efficient finetuning on AITW for mobile UI navigation**. We compared models either using DoM tree or image screenshot. We finetune our Magma jointly and then report the results on individual tasks. † Numbers reported in Zhang et al. [50]. ‡ Numbers reported in Chen et al. [6]. ♮ Numbers reported in Cheng et al. [7].

Model	VQAv2	GQA	MME	POPE	TextVQA	ChartQA	DocVQA
LLaVA-1.5-7B [26]	76.6	62.6	1510.8	85.9	46.1	18.2	28.1
LLaVA-Next-7B [29]	80.1	**64.2**	1519.3	**86.4**	64.9	54.8	74.4
Magma-8B (SFT)	79.5	61.5	1510.1	86.2	67.7	73.0	80.4
Magma-8B ($\text{Act}^{w/o}$)	81.3	63.5	1559.5	86.1	69.8	71.0	84.1
Magma-8B ($\text{Full}^{w/o}$)	81.3	62.9	1576.0	86.3	69.6	71.7	83.8
Magma-8B (Full)	**81.4**	64.0	**1588.7**	86.3	**70.2**	**76.2**	**84.8**

Table 5. **Finetuned performance on multimodal image understanding tasks**. Pretraining on full set with SoM and ToM (last row) attains the overall best performance compared with our own baselines and counterparts of the same model class.

B.2.4. Video Instruction Tuning

For comparisons with state-of-the-art video LMMs, we adopt the LLaVA-Video-178K dataset [51] for instruction tuning. It consists of approximately 1.6M video and text instruction samples from 178K videos. The dataset is compiled from multiple video sources ranging from Charades [44], Sth-SthV2 [34] to Kinetics-700 [3]. We refer interested readers to the original papers for more details.

B.2.5. Details about SoM for training and evaluation

we exploit three ways to extract the candidate bounding boxes for the SoM prompt:

- **DOM Tree**. In addition to the bounding boxes extracted from HTML code [7, 15], we further annotate the mobile screenshots in SeeClick data with bounding boxes derived from Android view hierarchies [45]. These annotations are used during our model pretraining.
- **Vision model.** For zero-shot evaluation on Screenspot [7], we exploit the OmniParser model [31] to make a fair comparison with the state-of-the-art methods [7, 31]. Note that we only use the bounding boxes without local semantics. The original bounding boxes in AITW [41] are identified using an OCR model and IconNet [45].
- **Language model**. For evaluation on As discussed earlier, we directly apply the predictions provided by Mind2Web [10] using a pretrained language model DeBERTa-v3-base. This model gives approximately 85% recall@50.

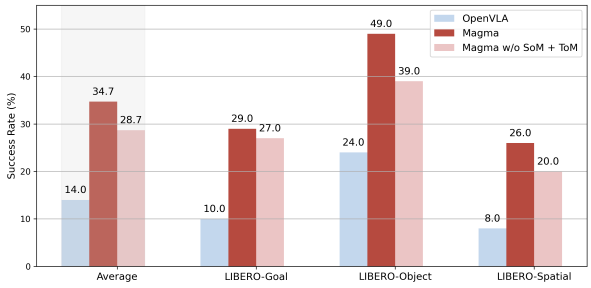


Figure 5. **Few-shot finetuning results on the LIBERO** simulation benchmark, using 10 trajectories per task for fine-tuning.

C. Additional Quantitative Analysis

Efficient finetuning on AITW. We report the results for different models on AITW for UI navigation on mobile in Table. 4. Similarly to the trend on Mind2Web, our Magma model outperforms the SOTA method by a large margin in the five task domains.

Image instruction tuning. To further assess Magma's multimodal understanding capability, we conduct continuous finetuning on our Magma-SFT-820K data. Then, we compare the finetuned Magma model with existing VLMs on a suite of commonly used image reasoning benchmarks, *e.g.* MME and GQA. As shown in Table 5, Magma outperforms recently-proposed VLMs on most of the tasks, with notable gains of ∼5% and ∼22% on TextVQA and ChartQA, respectively. Similarly to our observations in spatial evaluation results shown in our main paper, our ablation study highlights the effectiveness of using SoM and ToM for pre-training, which leads to ∼ 5% improvement in ChartQA.

Efficient finetuning on LIBERO. The efficient adaptation (via finetuning) capability of Magma is further validated through few-shot finetuning evaluations on the LIBERO benchmark. For each task suite in the benchmark, we sample only 10 trajectories for finetuning. During the

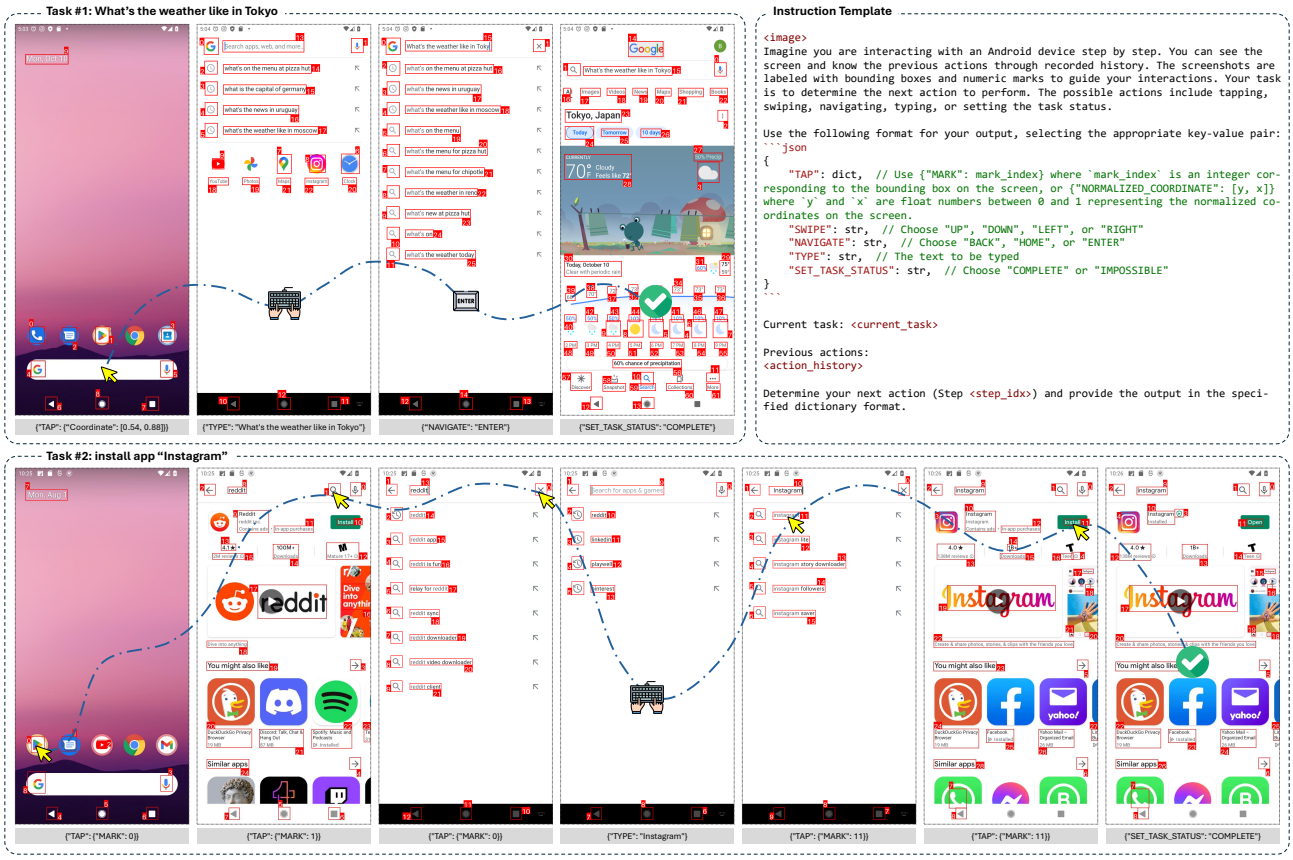


Figure 6. **Examples for mobile UI navigation sample**. We prompt the model with two tasks: "What's the weather like in Tokyo" and "Install app 'Instagram'". The model take actions sequentially given the new observation and history action information.

evaluation, we perform 100 trials per task suite. The results, shown in Fig. 5, indicate that Magma achieves a significantly higher average success rate in all task suites. Additionally, removing SoM and ToM during pretraining has a negative impact on model performance, underscoring the effectiveness of our pretraining method.

D. Qualitative Analysis

D.1. UI Navigation

Given the performant UI navigation performance across different tasks, we show some Mobile UI navigation samples in Fig. 6. We prompt the model to complete two daily tasks starting from the home page: "What's the weather like in Tokyo" and "Install app 'Instagram'". Despite that our model is never trained with the full trajectory, it can handle the tasks in the wild pretty well.

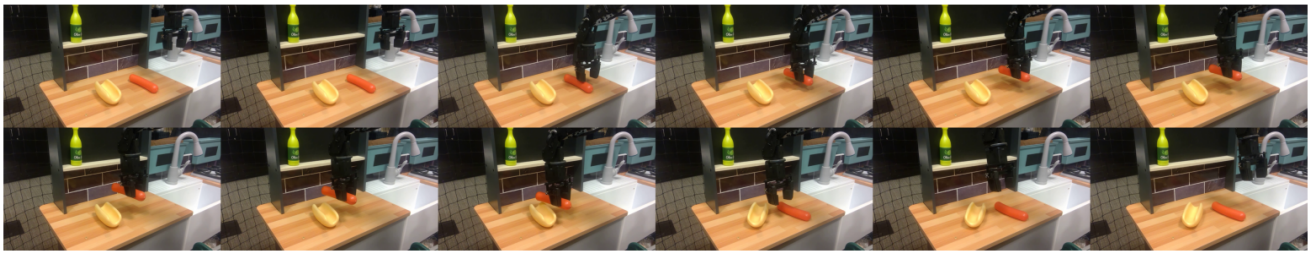
D.2. Robotics Manipulation

We further show the real robot manipulation rollout for OpenVLA and Magma model. As discussed in our main paper, our model exhibits much better generalization ability to different real robot manipulation tasks. In Fig. 7, we qualitatively show how two models handle a complicated task of "Pick up the sausage and put it inside the hotdog". Thanks to the proposed pretraining techniques, our Magma model can not only precisely pick up the sausage but also move smoothly to the top of the hotdog, demonstrating superior spatial understanding and reasoning capability compared with the counterpart.

E. Social Impacts

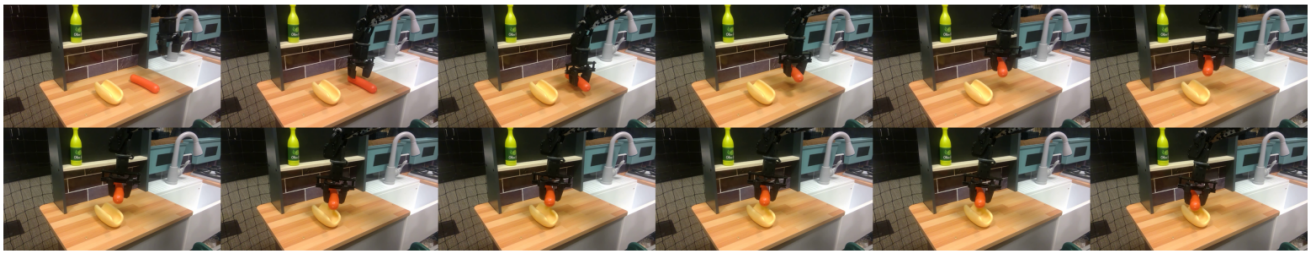
To develop a foundation model with both verbal and spatial intelligence capable of handling diverse agentic tasks in digital and physical environments, we curated a comprehensive pretraining dataset from a wide range of image, video, and robotics domains:

- **UI navigation data**. We leverage two pretraining datasets SeeClick and Vision2UI.
- **Instructional videos**. As our goal was to learn an agentic model that can undertake daily tasks like humans, we compile the videos from Epic Kitchen, Ego4d, Something-Something v2 and other instructional videos.



(a) Robot policy rollout for task "Put the sausage to hotdog" for OpenVLA model. (Failure)

(b) Robot policy rollout for task "Pick up the mushroom to the pot" for OpenVLA model. (Failure)



(c) Robot policy rollout for task "Put the sausage to hotdog" for Magma model. (Success)

(d) Robot policy rollout for task "Pick up the mushroom to the pot" for Magma model. (Success)

Figure 7. **Comparison between OpenVLA (top two rows) and Magma (bottom two rows) for real robot manipulation task.** The two robot policies starts with the same initial stage and asked to perform exactly the same task. The whole task requires precise spatial understanding and planning for the model. For both tasks, OpenVLA failed to accomplish while our model successfully handle.

- **Robotics manipulation data**. For robotics task, we follow OpenVLA to leverage the robotics data in Open-X-Embodiment.
- **Multimodal understanding data**. Lastly, we include a small set of multi modal pretraining data ShareGPT4V, and instruction tuning data LLaVA-1.5 plus a number of other domain-specific data to retain the generic multi-modal understanding capability of the pre-trained model.

The data markup of the robotics and UI navigation data is fairly standardized focusing on generic manipulation tasks ("Place x object on y object") and generic UI navigation tasks ("Click search button"). We, however, performed a detailed data reflection exercise on the video data of people performing certain tasks. The core inferences we took from these videos were the trajectory of objects over time when the tasks were performed.

We note that the distribution of identities and activities in the instructional videos are not representative of the global human population and the diversity in society. We are cognizant of the unintended societal, gender, racial and other biases in training with these data, so we will ensure required disclaimers are in place when publishing the models.

The training dataset, task list and descriptions focus on the next action to perform only – not describe, act on, or perform any analysis on the subject itself. While there can be unintended outputs from the model based on adverse task descriptions, we will ensure to highlight the use cases the model was trained for and its intended use.

Responsible AI. It is important to note that the model is specifically designed for UI navigation in a controlled Web UI and Android simulator, and robotic manipulation tasks and should not be broadly applied to other tasks. The recommended usage is within the settings they were trained on, namely, an enclosure equipped with a robotic arm and everyday objects for robotic manipulation and an android simulator running on a computer for UI manipulation. For UI navigation task, researchers should make sure that a human is in the loop and in control for every action the agentic system generates. Since the model cannot act by itself, the sub-module a researcher uses to actually perform the UI navigation action should ensure that no unintended consequences can occur as a result of performing the UI action proposed by the model. The model by itself demonstrates good-enough capability in UI navigation and robotic manipulation, but is not usable as is for exploitation scenarios. A threat actor, can however use specific training data for a specific malicious task, to leverage the model as a base to perform automated UI navigation. This is a generic risk associated with the agentic models.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. 6

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

[3] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 6

[4] Brandon Castellano. Pyscenedetect: Automated scene detection in videos, 2014–2024. Version 0.6.4, BSD 3-Clause License. 3

[5] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023. 4, 5

[6] Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, Yuan Yao, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Guicourse: From general vision language models to versatile gui agents, 2024. 6

[7] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents, 2024. 1, 2, 4, 6

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 3

[9] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Y. Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 2017. 1

[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, 2023. 4, 6

[11] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022. 4

[12] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 5

[13] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 5

[14] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022. 3

[15] Yi Gui, Zhen Li, Yao Wan, Yemin Shi, Hongyu Zhang, Yi Su, Shaoling Dong, Xing Zhou, and Wenbin Jiang. Vision2ui: A real-world dataset with layout for code generation from ui designs, 2024. 1, 6

[16] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 5

[17] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2018. 5

[18] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Akos Kadar, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning, 2018. 5

[19] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in pho-

tographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014. 5

[20] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision – ECCV 2016*, pages 235–251, Cham, 2016. Springer International Publishing. 5

[21] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5376–5384, 2017. 5

[22] 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

[23] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024. 4

[24] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)*, 123(1):32–73, 2017. 5

[25] LAION-4V. Laion gpt4v-dataset, 2023. 5

[26] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*, 2023. 6

[27] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024. 5

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023. 5

[29] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 4, 5, 6

[30] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 2022. 5

[31] Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. Omniparser for pure vision based gui agent, 2024. 4, 6

[32] 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. 5

[33] 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, 2022. 5

[34] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1049–1059, 2020. 2, 3, 6

[35] 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. 5

[36] 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. 5

[37] Meta. Llama-3. https://ai.meta.com/blog/meta-llama-3/, 2024. 6

[38] 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. 5

[39] OpenAI. Gpt-4v(ision) system card. https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023. 6

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 4

[41] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Android in the wild: A large-scale dataset for android device control, 2023. 4, 6

[42] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pages 146–162. Springer, 2022. 5

[43] ShareGPT. ShareGPT, 2023. 5

[44] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer, 2016. 6

[45] Srinivas Sunkara, Maria Wang, Lijuan Liu, Gilles Baechler, Yu-Chung Hsiao, Jindong Chen, Abhanshu Sharma, and James Stout. Towards better semantic understanding of mobile interfaces. *CoRR*, abs/2210.02663, 2022. 6

[46] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023. 1

[47] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022. 6

[48] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 6

[49] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016. 5

[50] Shaoqing Zhang, Zhuosheng Zhang, Kehai Chen, Xinbei Ma, Muyun Yang, Tiejun Zhao, and Min Zhang. Dynamic planning for LLM-based graphical user interface automation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Miami, Florida, USA, 2024. Association for Computational Linguistics. 6

[51] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 6

[52] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v(ision) is a generalist web agent, if grounded, 2024. 6

[53] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. 4

[54] Luowei Zhou, Nathan Louis, and Jason J Corso. Weakly-supervised video object grounding from text by loss weighting and object interaction. In *British Machine Vision Conference*, 2018. 4