EVALUATING TOOL-AUGMENTED AGENTS IN REMOTE SENSING PLATFORMS

Simranjit Singh, Michael Fore, Dimitrios Stamoulis
CoStrategist R&D Group, Microsoft Corporation, Redmond, WA, USA
{simsingh, mifore, stamoulis.dimitrios}@microsoft.com

ABSTRACT

Tool-augmented Large Language Models (LLMs) have shown impressive capabilities in remote sensing (RS) applications. However, existing benchmarks assume question-answering input templates over predefined image-text data pairs. These standalone instructions neglect the intricacies of realistic *user-grounded* tasks. Consider a geospatial analyst: they zoom in a map area, they draw a region over which to collect satellite imagery, and they succinctly ask "*Detect all objects here*". Where is *here*, if it is not explicitly hardcoded in the image-text template, but instead is implied by the system state, *e.g.*, the *live* map positioning? To bridge this gap, we present GeoLLM-QA, a benchmark designed to capture long sequences of verbal, visual, and click-based actions on a *real* UI platform. Through in-depth evaluation of state-of-the-art LLMs over a diverse set of 1,000 tasks, we offer insights towards stronger agents for RS applications.

1 INTRODUCTION

Large Language Models (LLMs) demonstrate impressive potential in complex geospatial scenarios, augmenting remote sensing (RS) platforms with agents capable of sophisticated planning, reasoning, and task execution. These developments have sparked interest to deploy multimodal models across various RS tasks, including image captioning and visual question answering (VQA) (Yuan et al., 2022). Notably, SkyEyeGPT (Zhan et al., 2024) finetunes state-of-the-art VQA agents (Chen et al., 2023) on RS imagery for unified multimodal responses, while Remote Sensing ChatGPT (Guo et al., 2024) deploys computer-vision models (*e.g.*, land use classification, object detection) via prompting. However, these approaches rely on chatbot-based templates with predefined text-image correlations over specific image files to assess LLM performance (Fig. 1 left), hence failing to capture the nuances of realistic *user-grounded* RS tasks.

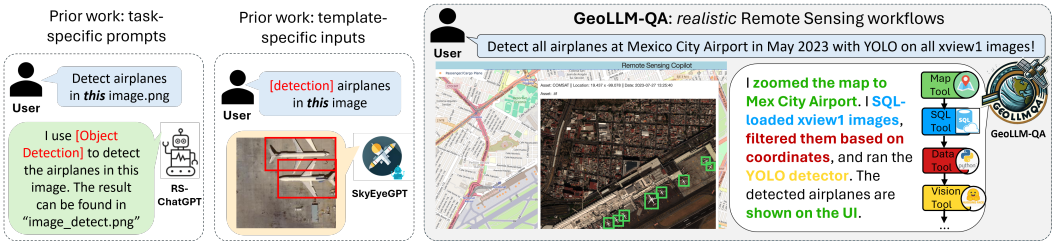

Figure 1: Unlike prior work that assumes task-specific templates (*e.g.*, "[detection]" keyword), GeoLLM-QA requires the agent to follow nuanced instructions and perform multi-step reasoning to accomplish user-defined objectives.

In this work, we aim to bridge this gap with the following contributions: ***first***, we introduce GeoLLM-QA, a novel benchmark of **1,000 diverse tasks**, designed to capture complex RS workflows where LLMs handle complex data structures, nuanced reasoning, and interactions with dynamic user interfaces (Fig. 1 right). To this end, we harness recent advancements in benchmarking work for tool-augmented LLMs (Zhuang et al., 2023; Maini et al., 2024; Koh et al., 2024). ***Second***, we adopt a comprehensive evaluation scheme (Maini et al., 2024) beyond traditional text-based metrics that accurately assesses an agent's proficiency in utilizing external tools for effective problem-

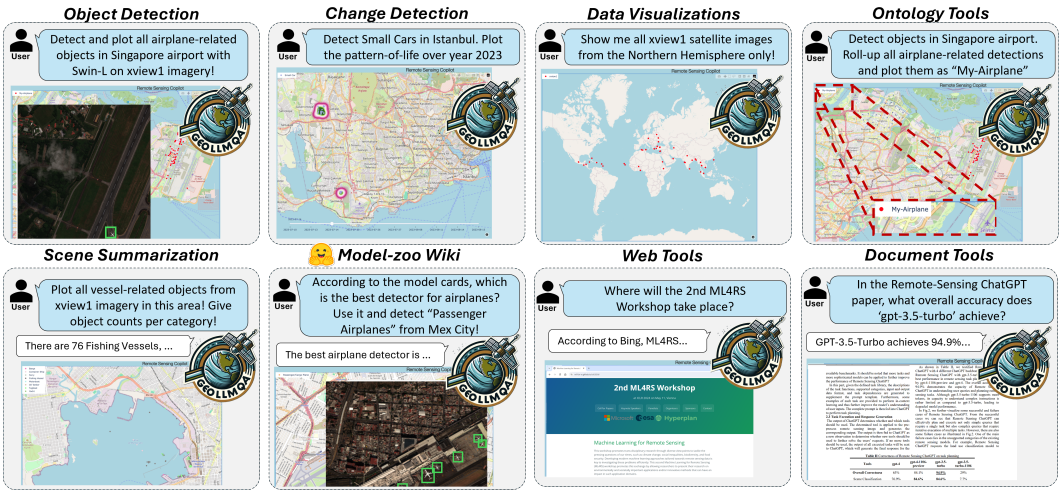

Figure 2: GeoLLM-QA challenges agents to solve complex RS tasks through multimodal reasoning and actions over long sequences of verbal, visual, and click-based actions on a *real* UI platform.

solving. ***Third***, we evaluate several state-of-the-art tool-augmentation and prompting methodologies on our benchmark. We highlight our key takeaways regarding the strengths, weaknesses, and potential of LLMs within geospatial platforms. We strive to motivate future work and help the RS community in unlocking further advancements in this domain.

2 The GeoLLM-QA Framework

Benchmarking Platform: To assess geospatial reasoning in an agent-assisted *platform* context, we draw inspiration from (Zhou et al., 2023) and we implement a benchmarking UI, as a realistic and reproducible standalone web-app that incorporates user-centered tasks with open-source tools and datasets. By leveraging open-source APIs, not only we address challenges of reproducibility and comparison across different systems, but also enable the examination of a wide range of RS use-cases through various input modalities including verbal, visual, and tactile interactions. The complete tool set consists of 117 tools, such as `plotly mapbox` APIs for the map functionality and `LangChain` routines for `FAISS` vectorstores (Douze et al., 2024), to name a few. We intend to release our codebase and benchmark to stimulate future research on geospatial Copilots.

Problem Formulation: To denote RS tasks beyond simplistic VQA data-pairs, we model the problem after the realistic UI experience: intuitively, each interaction consists of the user question, the sequence of tool-calls by the agent, and the final (textual) response to user and platform state. We can therefore denote each task as $\{q, T, r, S\}$, where q is the user prompt, r is the textual response, while T represents the set of tool-calling steps $T = \{t_1, t_2, \dots\}$. At each step i, the agent invokes tool $t_i = \{tool_i, args_i^{**}\} \in \mathcal{T}$ from the available tool space $\mathcal{T}$. Finally, S defines the final system state: *e.g.*, map positioning, loaded database, visible data holdings, *etc*.

Data Sources: Our evaluation framework includes three representative large-scale datasets: `xview1` (Lam et al., 2018), `xview3` (Paolo et al., 2022), `DOTA-v2.0` (Ding et al., 2021). Encompassing both optical and synthetic aperture radar (SAR) imagery, these data holdings offer detailed object annotations across 80 categories from a total of 5,000 images. Notably, these datasets come with valuable metadata, such as dates and coordinates, which greatly enhances the complexity of temporal and spatial RS scenarios in our benchmark. The satellite imagery serves as *task context* for LLM agents to execute function calls and is *not* used for finetuning the LLM or other downstream tasks, enabling our research-purposes investigation.

"Golden" Detector Models: without loss of generality, we employ "oracle detectors," a common practice in foundation-models literature (Yang et al., 2023a), so that we can concentrate on evaluating the agent's proficiency in selecting and utilizing the appropriate tools without confounding

the false positives/negatives of a non-optimal detector. By abstracting out detection errors, we can measure any degradation in performance metrics directly attributable to agents' failures.

For instance, consider a scenario where the LLM is instructed to "*detect all airplanes at the Mexico City airport using the YOLO detector*." We want to verify whether the agent can designate the right detector, filter through the correct imagery, and specify the right classes. Therefore, upon the LLM's selection of an image set, we assume an oracle detector that provides 100% accurate detections, *i.e.*, "gold" results directly from dataset ground truths. We then calculate the recall of these detector "results", attributing any discrepancies solely to the agent's inability to accurately fulfill the task.

Benchmark Creation: To create a representative set of RS tasks, `GeoLLM-QA` adopts the three-step benchmarking process presented in (Zhuang et al., 2023): *1. Reference Template Collection*: we curate a set of 25 template questions that cover the wide range of RS tasks, such as object detection, change detection, *etc*. Several key tasks are shown in Fig. 2. To generate answers for these questions, we guide `GPT-4` to reach the answers via a simple human-in-the-loop mechanism via *feedback* UI buttons (Ouyang et al., 2022). By using previous (un)successful attempts as in-context examples, GPT can quickly help us create the Reference Templates.

2. LLM-guided Question Generation: we generate permutations and perturbations of the Reference Templates. Note here that previous RS benchmarks assume that all LLM tasks are implicitly correct. However, Maini et al. (2024) show that one of the most challenging aspects of agent performance is their ability to handle prompts that maintain the general template of a genuine question but are ***factually incorrect***. We therefore assume a ratio of 9:1 correct:incorrect tasks and we use `GPT-4` to generate variations per template for a total of 1,000 tasks. To allow `GPT-4` to "programmatically" select from real data combinations, we provide in-context prompt with dataset descriptions, *e.g.*, SQL schemas with all eligible category names in the `xview1` database.

Reference Question with Paraphrased and Perturbed Variations

```
Reference Q: Use the YOLO detector to detect fishing vessels in
xview3 images around Ancona.  Plot them on the map.
Paraphrased Q: Use RetinaNet to find yacht detections in xview1
images around Barbados, and show them on the map.
Perturbed Q: Use NoNet to find Zeppelins in images around the
mythical city of Atlantis.
```

3. Human-guided Ground Truth Generation: last, to generate the ground truth answers and tool-set solutions, we task `GPT-4` to solve each question using the available platform tools. To guide the process, we leverage the Reference Templates (questions and solutions) and we augment the LLM by dynamically retrieving similarly correct examples via RAG (Gao et al., 2024). This allows us to accelerate the process, while ensure the human-on-the-loop to validate the overall correctness.

Metrics: Unlike existing VQA-based benchmarks, we consider a a comprehensive set of metrics that capture the LLM's ability for effective tool-calling and reasoning:

a. *Success rate*: the ratio of successfully completed tasks across the entire benchmark. Each task is consider to be completed correctly when the final platform state S matches the $\tilde{S}$ ground-truth. This ratio informs us of the degree to which the agent is able to complete tasks, irrespective of whether it took incorrect or unnecessary intermediate steps.

b. *Correctness ratio*: the ratio of correct function-call operations across the benchmark. Given a ground-truth tool-set $\tilde{T}$ and an LLM solution T, we track *all* applicable LLM error-types as defined in (Zhuang et al., 2023) (*i.e.*, "Infeasible Action", "Function Error", "Argument Error", "Incorrect Data Source", and "Omitted Function"). Given the total number of errors and ground-truth tools, we compute the correctness ratio $R_{correct} = \max(0, 1 - N_{errors}/N_{tools})$ (Maini et al., 2024). This metric captures how likely it is for the agent to invoke the correct functions in the expected order.

c. *ROUGE score*: we use the ROUGE-L recall score (Lin, 2004) to compare model answers a with the ground truth $\tilde{a}$ to assess the ability of the agent to reply to the task at hand.

d. *Cost (Tokens)*: we compute the average number of tokens per task over the entire benchmark.

e. *(Detection) Recall*: over the entire benchmark, we assess the agents ability to correctly return detection tasks by calculating the overall recall R (*i.e.*, detections returned by the method against "gold" ground-truths from oracle detectors).

Table 1: Performance of different agents on `GeoLLM-QA-1k`.

	Success Rate↑	Correctness Rate↑	ROUGE -L↑	Det. Recall↑	Avg. Tokens↓ /Task↓
GPT-3.5 Turbo (0125)					
CoT (Wei et al., 2023) Zero-Shot	30.74%	80.67%	21.42%	91.92%	7.4k
CoT (Wei et al., 2023) Few-Shot	31.65%	89.55%	22.05%	71.17%	9.3k
Chameleon (Lu et al., 2023) Zero-Shot	23.69%	79.88%	23.29%	89.73%	12.1k
Chameleon (Lu et al., 2023) Few-Shot	26.74%	85.70%	24.30%	96.18%	12.9k
ReAct (Yao et al., 2023) Zero-Shot	30.70%	86.26%	22.31%	77.17%	7.5k
ReAct (Yao et al., 2023) Few-Shot	32.95%	89.35%	26.06%	91.78%	11.1k
GPT-4 Turbo (0125)					
CoT (Wei et al., 2023) Zero-Shot	34.99%	94.59%	26.82%	85.81%	8.7k
CoT (Wei et al., 2023) Few-Shot	33.35%	94.93%	27.09%	93.33%	9.2k
Chameleon (Lu et al., 2023) Zero-Shot	29.44%	83.49%	21.57%	88.88%	12.5k
Chameleon (Lu et al., 2023) Few-Shot	31.18%	89.59%	22.56%	90.41%	13.1k
ReAct (Yao et al., 2023) Zero-Shot	33.52%	94.85%	27.82%	87.77%	9.5k
ReAct (Yao et al., 2023) Few-Shot	33.39%	94.98%	27.75%	96.73%	11.6k

3 EXPERIMENTS

In the scope of this analysis, we run various prompting techniques from literature: Chain-of-Thought (Wei et al., 2023), (MM-)ReAct (Yao et al., 2023; Yang et al., 2023b), and Chameleon (Lu et al., 2023). We leave more advanced prompting strategies for future investigation. Our baselines language models include GPT-4 Turbo (`gpt-4-0125-preview`) and GPT-3.5 Turbo (`gpt-3.5-turbo-1106`).

Tab. 1 summarizes our findings. The recent GPT-4 Turbo release exhibits impressive function-calling capabilities, while in terms of methods, CoT and ReAct outperform Chameleon in both correctness and success rates, while being more token efficient. With respect to other metrics, ROUGE-L shows the limitations of text-based scores, as it has been reported by recent work on foundation models comparing closed- and open-vocabulary answers (Majumdar et al., 2024). That is, the distribution of LLM answers is heavily dependent on the prompting method. For instance, answers generated by GPT-3.5 might artificially penalize a different response style by Chameleon if treated ground-truths (*e.g.*, "There are five airplanes" *vs.* "This image contains 5 planes" can result in lower scores despite conveying the same fact). Last, we observe that detection-related metrics, as captured by *recall*, do not necessarily correlate with agent performance. All these findings confirm that, unlike existing RS benchmarks that mainly report detection results or captioning-related scores, a more comprehensive evaluation is required to assess agent performance.

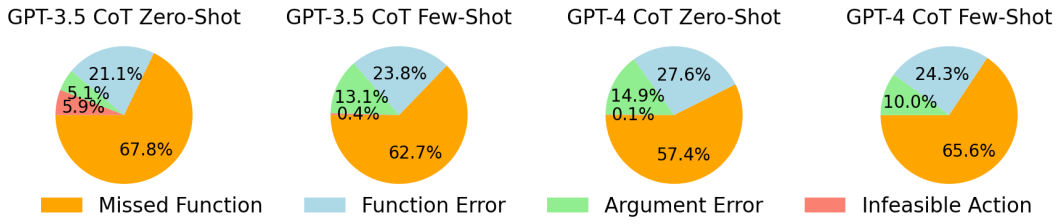

Figure 3: GPT-3.5 *vs.* GPT-4 error analysis for CoT prompting.

Fig. 3 shows the error types for CoT on GPT-3.5 and GPT-4, in both zero-shot and few-shot scenarios. The most common, "Missed Function" (where the agent omits necessary tool calls regardless of the approach used) accounts for more than half of all errors. We expect that dynamic/RAG-augmented (Srinivasan et al., 2023) prompting should improve agent performance by addressing such failures. Last, the consistent distribution across different cases implies that these issues are not method-specific but rather inherent to the current GPT capabilities.

4 Conclusion and Future Work

We presented `GeoLLM-QA`, a benchmark of realistic *user-grounded* tasks aimed at assessing the capabilities of tool-augmented LLMs in geospatial applications. Our hope is that this benchmarking suite will spur the development of new agents that advance the state of the art in remote sensing platforms. To this end, we would like to highlight some particularly exciting and promising areas for future work that we have identified through our research and that we are actively investigating. First, recent advances in multimodal modeling show improved performance compared to MM-ReAct-like prompting. We are currently extending our benchmark to flexibly incorporate open-source GPT-V model families, such as mini-GPT (Zhu et al., 2023; Chen et al., 2023). Additionally, we are expanding our analysis to replace oracle detectors with state-of-the-art models (Jian et al., 2023), to explore how agent errors interact with suboptimal detector performance.

Moreover, a primary bottleneck that we have encountered with our approach, which is common in related work (Zhan et al., 2024), is the overhead of human-guided template generation. In our most recent study (Singh et al., 2024), we demonstrate that by adopting engine-based benchmarking methodologies (Zhou et al., 2023) in the remote sensing domain, we can leverage fully GPT-driven template and ground-truth generation to minimize human-in-the-loop overhead. Lastly, by considering cost- and system-related aspects, our analysis has yielded interesting insights regarding optimizing the overall agent-platform implementation. Our ongoing explorations include methods to improve performance by leveraging state-of-the-art LLM caching and compression techniques (Jiang et al., 2023; Fore et al., 2024).

References

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning, 2023.

Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/TPAMI.2021.3117983.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library, 2024.

Michael Fore, Simranjit Singh, and Dimitrios Stamoulis. Geckopt: Llm system efficiency via intent-based tool selection, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

Haonan Guo, Xin Su, Chen Wu, Bo Du, Liangpei Zhang, and Deren Li. Remote sensing chatgpt: Solving remote sensing tasks with chatgpt and visual models, 2024.

Yanan Jian, Fuxun Yu, Simranjit Singh, and Dimitrios Stamoulis. Stable diffusion for aerial object detection. In *NeurIPS 2023 Workshop on Synthetic Data Generation with Generative AI*, 2023.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models, 2023.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks, 2024.

Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. xview: Objects in context in overhead imagery, 2018.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. 2004.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models, 2023.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

Fernando Paolo, Tsu ting Tim Lin, Ritwik Gupta, Bryce Goodman, Nirav Patel, Daniel Kuster, David Kroodsma, and Jared Dunnmon. xview3-sar: Detecting dark fishing activity using synthetic aperture radar imagery, 2022.

Simranjit Singh, Michael Fore, and Dimitrios Stamoulis. Geollm-engine: A realistic environment for building geospatial copilots, 2024.

Venkat Krishna Srinivasan, Zhen Dong, Banghua Zhu, Brian Yu, Hanzi Mao, Damon Mosk-Aoyama, Kurt Keutzer, Jiantao Jiao, and Jian Zhang. Nexusraven: a commercially-permissive language model for function calling. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023. URL https://openreview.net/forum?id=Md6RUrGz67.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v, 2023a.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action, 2023b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.

Zhiqiang Yuan, Wenkai Zhang, Kun Fu, Xuan Li, Chubo Deng, Hongqi Wang, and Xian Sun. Exploring a fine-grained multiscale method for cross-modal remote sensing image retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022. ISSN 1558-0644. doi: 10.1109/tgrs.2021.3078451. URL http://dx.doi.org/10.1109/TGRS.2021.3078451.

Yang Zhan, Zhitong Xiong, and Yuan Yuan. Skyeyegpt: Unifying remote sensing vision-language tasks via instruction tuning with large language model, 2024.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2023.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset for llm question answering with external tools, 2023.