


ScreenSpot-Pro: GUI Grounding for Professional High-Resolution Computer Use


Kaixin Li¹ Ziyang Meng² Hongzhan Lin³ Ziyang Luo³ Yuchen Tian³
Jing Ma³ Zhiyong Huang¹ Tat-Seng Chua¹

¹National University of Singapore ²East China Normal University

³Hong Kong Baptist University
likaixin@u.nus.edu

 <https://huggingface.co/datasets/likeixin/ScreenSpot-Pro>

 <https://github.com/likeixin2000/ScreenSpot-Pro-GUI-Grounding>

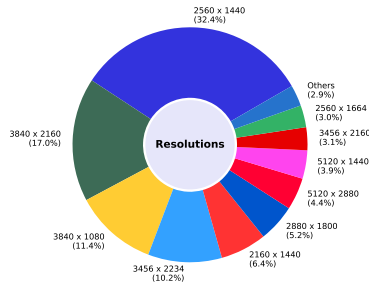
 <https://gui-agent.github.io/grounding-leaderboard/>

Abstract

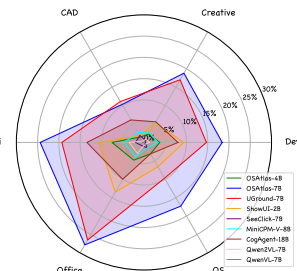
While recent advancements in Multi-modal Large Language Models (MLLMs) have spurred progress in GUI agents for general tasks such as web browsing and mobile phone use, professional applications remain under-explored. These tools, which cater to specialized workflows, present unique challenges for GUI agents, including higher resolution screens, smaller target sizes, and complex working environments. In this paper, we introduce ScreenSpot-Pro, a new benchmark specifically designed to comprehensively assess the capabilities of models in high-resolution professional environments, consisting of authentic high-resolution images and tasks from diverse professional domains annotated by experts. It contains 23 applications in 5 types of industries and common usages in 3 operating systems. Existing GUI grounding models perform poorly on this dataset, with the best model achieving 18.9%. Though our experiments show that strategically shrinking the image size improves performance, the optimal strategy only reaches 40.2%. This remains unsatisfactory, highlighting the need for further progress in this area.



(a) Software categories and the number of tasks.



(b) Resolution distribution.



(c) Results of representative GUI Grounding models across 6 categories of applications.

Figure 1: Task distribution and benchmark results of ScreenSpot-Pro.

1 Introduction

Imagine a future where the everyday burdens of repetitive computer tasks are lifted, unleashing people’s full productivity and creativity. A GUI agent capable of taking over the mundane operations of complex professional applications like Visual Studio Code, AutoCAD, Photoshop, could greatly enable computer users to focus exclusively on the work that truly matters. Recent advancements in Multi-modal Large Language Models (MLLMs) [1, 2, 3, 4] have significantly invigorated this pursuit, driving intensive research efforts in creating pure-vision based GUI agent models that can directly interact with electronic devices that are integral to modern life [5, 6]. These models are capable, to some extent, of directly perceiving device screens in a manner similar to humans, and making decisions on the operations based on the observations.

However, many existing studies primarily address general and everyday tasks performed on electronic devices, such as general computer control [7, 8], web browsing [9, 10, 11], lifestyle and utility apps [12, 13]. In contrast, professional applications remains largely unexplored, with only few works featuring specialized tasks such as coding in VSCode [14]. These software are designed to provide a comprehensive suite of advanced features, catering to specialized tasks and workflows, and are thus fundamental in productivity and creative industries. These programs often involve intricate details, such as high-definition visuals, complex layouts, and data-dense interfaces, challenging GUI agents for higher levels of perception, comprehension, and interaction with the environment.

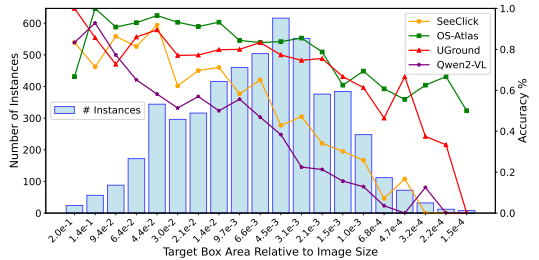
The primary challenge of applying GUI agents to these professional applications is threefold: (1) the significantly greater complexity of professional applications, compared to general-use software, often necessitates the use of higher resolutions that exceed the effective handling capacity of current VLMs; (2) the increased resolution results in smaller relative target sizes in the screenshot, where GUI grounding models generally exhibit worse performance, as demonstrated in Figure 2; (3) professional users frequently rely on additional documents and external tools to complement their workflows, further complicating the screen. Consequently, even if the GUI agents¹ are able to understand user instructions and the user interfaces in the professional work environment, it is difficult for them to ground the instructions into executable actions in such complex screenshots.

This paper explores the key challenge in GUI grounding in professional high-resolution environments. Given a natural language instruction and a screenshot, the models are asked to ground the instruction to a precise location of the target UI element. We introduce ScreenSpot-Pro, a new GUI grounding benchmark that includes 23 applications in 5 types of industries, as well as common usages in 3 operating systems. It contains 1,581 instructions, each paired with a unique screenshot, captured by professional users. These tasks are further categorized into ScreenSpot-Pro differentiates itself from previous grounding benchmarks [8, 17] in that: i) ScreenSpot-Pro includes authentic high-resolution images and tasks captured from a variety of professional applications and domains, thus reflecting the complexity and diversity of real-world scenarios; ii) ScreenSpot-Pro is annotated by professional users, ensuring rigorous quality control to maintain the validity of test samples, guaranteeing reliable and meaningful evaluation results.

Our contribution is summarized as follows:

- We present ScreenSpot-Pro, a new benchmark for GUI grounding designed to facilitate comprehensive evaluation with authentic tasks collected from various high-resolution professional desktop environments.

¹In this work, we use the terms “GUI agent” and “GUI model” interchangeably to refer to the VLMs, as the primary focus of this work is on the grounding capabilities of these models.



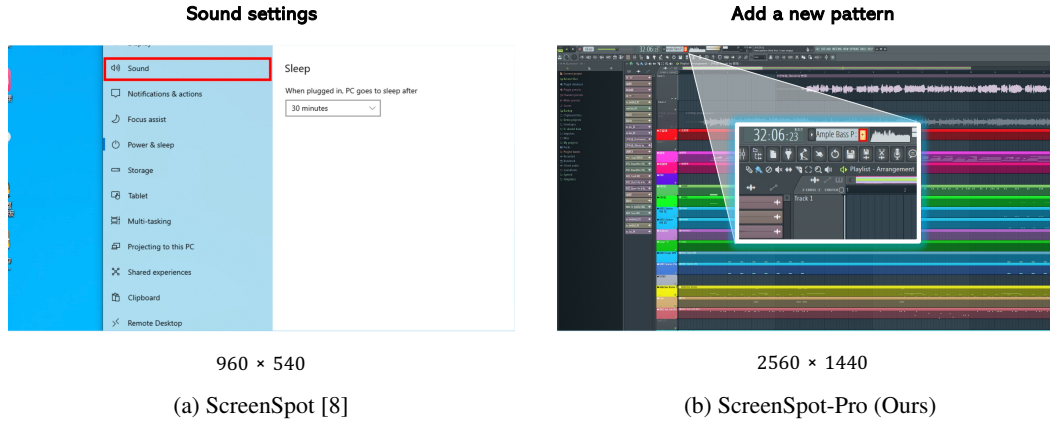


Figure 3: ScreenSpot [8] (left) vs ScreenSpot-Pro (right). ScreenSpot-Pro features screenshots of the entire screen, while ScreenSpot contains unrealistic screenshots cropped to local areas. Targets are highlighted in red boxes.

- We offer a comprehensive evaluation and comparison of common GUI grounding models.
- We identify key challenges in the development of more effective GUI grounding models and introduce baseline methods to tackle the difficulties posed by high-resolution image inputs.

2 ScreenSpot-Pro: A Benchmark for Professional High-Resolution Computer Use

In this section, we introduce the data collection criteria, processing procedure, and quality control measures, and provide a statistical overview of ScreenSpot-Pro.

2.1 Scope of Data Collection



























ScreenSpot-Pro includes six empirical genres of applications, focusing primarily on four types of professional applications. Additionally, it incorporates office productivity tools and operating system commons as supplementary tasks to support the evaluation of GUI agents in high-resolution environments for details about these applications and operating systems.

Development and Programming. Development and programming software supports the entire lifecycle of software development, from writing code to debugging and testing applications. These tools provide integrated environments that enhance productivity and collaboration, offering features like syntax highlighting, version control integration, and debugging tools. The applications in this category include **VSCode** (code editor), **PyCharm** (Python IDE), **Android Studio** (Android app development), and **Quartus** (FPGA programming). Additionally, virtualization is critical for creating scalable computing solutions and managing virtual environments, so we also include **VMware Fusion** (virtual machine management).

Creative Software. Creative software includes applications designed for the creation and editing of visual, audio, and video content. These tools are essential in industries such as graphic design, video production, and music composition, enabling professionals to produce high-quality media for various platforms. The tools in this category include **Photoshop** (image editing), **Premiere** (video editing), **Illustrator** (vector graphic design), **FruitLoops Studio** (music production), **DaVinci Resolve** (color grading and video editing), **Unreal Engine** (game engine and 3D simulation), and **Blender** (3D modeling and animation).

Computer-Aided Design (CAD) and Engineering. CAD and engineering software are used to design and model physical objects and systems. These applications are vital in fields such as engineering, architecture, and product manufacturing, where precision design and simulation are

Table 1: List of software collected in ScreenSpot-Pro.

Icon	Abbr.	Application	Edition & Version	OS	Icons	Texts
Development and Programming						
	VSC	Visual Studio Code	1.95	macOS	22	33
	PyC	PyCharm	2023.3	macOS	38	40
	AS	Android Studio	2022.2	macOS	44	36
	Qrs	Quartus	II 13.0 SP1	Windows	32	13
	VM	VMware	Fusion 13.6.1	macOS	9	32
Creative						
	PS	Photoshop	2020	Windows	25	26
	PR	Premiere	2025	Windows	24	28
	AI	Adobe Illustrator	2025	Windows	19	12
	Bl	Blender	4.0.2	Windows	15	56
	FL	FruitLoops Studio	20.8.3	Windows	31	26
	UE	Unreal Engine	5.4.4	Windows	6	29
	DR	DaVinci Resolve	19.0.3	macOS	23	21
CAD and Engineering						
	CAD	AutoCAD	Mechanical 2019	Windows	7	27
	SW	SolidWorks	Premium 2018 x64	Windows	14	63
	Inv	Inventor	Professional 2019	Windows	11	59
	Vvd	Vivado	2018.3	Windows	32	48
Scientific and Analytical						
	MAT	MATLAB	R2022b	Windows	19	74
	Org	Origin	2018	Windows	43	19
	Stt	Stata	SE 16	Windows	41	8
	Evw	EViews	10	Windows	7	43
Office Suite						
	Wrd	Word	Office 365 (16.90)	macOS	15	69
	PPT	PowerPoint	Home and Student 2019	Windows	25	57
	Exc	Excel	Office 365 (16.82)	macOS	13	51
Operating System Commons						
	Win	Windows	11 Professional	-	47	34
	mac	macOS	Sonoma 14.5	-	23	42
	Lnx	Linux	Ubuntu 24.04	-	19	31

required. They enable professionals to create detailed 2D drawings, 3D models, and simulate the behavior of mechanical structures. The tools in this category include **AutoCAD** (2D/3D design), **SolidWorks** (3D CAD and simulation), **Inventor** (mechanical design), and **Vivado** (circuit design and FPGA programming).

Scientific and Analytical. Scientific and analytical software is designed for data analysis, numerical computation, and mathematical modeling. These applications are indispensable in fields like research, engineering, and data science, providing robust environments for analyzing large datasets, solving complex mathematical problems, and running simulations. The software in this category includes **MATLAB** (numerical computation and algorithm development), **Origin** (data analysis and scientific visualization), **Stata** (statistical analysis), and **EViews** (econometric modeling).

Office Software. Office software includes applications designed to facilitate productivity in tasks such as document creation, data analysis, communication, and presentation. These tools are widely used across various industries to manage workflows and support collaborative environments. Key

applications in this category include **Word** (word processing), **Excel** (spreadsheets and data analysis), **PowerPoint** (presentation design).

Operation System Commons. Apart from professional software, ScreenSpot-Pro also includes basic operating system operations to evaluate models in high-res environments. These samples are referred to as Operating System Commons, encompassing the general use and interaction with a OS. These include file management, system utilities, etc., that are fundamental to day-to-day tasks on any OS. For this category, we include **Windows**, **macOS**, and **Linux**.

2.2 Collection Method and Criteria

ScreenSpot-Pro aims to reflect realistic tasks in real-world challenges across various platforms and applications². To achieve this, it is crucial to capture the authentic workflows of professionals. We invited a total of 14 experts with at least five years of experience using the relevant applications to record the data. They were instructed to perform their regular work routine to ensure the authenticity of the tasks whenever possible. To minimize disruptions to their workflow, we developed a silently running screen capture tool, accessible through a shortcut key. When activated, this tool takes a screenshot and overlays it on the screen, allowing experts to label the bounding boxes and provide instructions directly. This method enhances the consistency and quality of the annotations, as experts can label tasks in real-time without the need to recall the purposes and context of their actions in hindsight. An example of the annotation tool can be found in Figure 4 in the Appendix.

To obtain authentic high-resolution images, we prioritized screens with a resolution greater than 1080p (1920×1080), a configuration commonly found among annotators. Monitor scaling was disabled. In dual-monitor setups, images were captured to span both displays.

Following SeeClick [8], we also specify the type of the target element, categorizing it as either *text* or *icon*. We refined the classification criteria to better discriminate ambiguous cases where icons are accompanied by text labels, which is common in AutoCAD and Office suites. Specifically, a target is classified as *icon* only when no text hints are present. If text labels are present, the target is labeled as *text*, even if an icon is included.

2.3 Quality Control

Task Validity. Each instance in the dataset is reviewed by at least two annotators from the author team to ensure the correctness of the instructions and target bounding boxes. Additionally, we removed instructions that caused ambiguity: each instruction must refer to, and only to, a single area in the image. It is also guaranteed that all instructions can be executed directly on the screenshot without requiring further actions, such as switching to other windows, opening menus, or right-clicking.

Target Box Precision. To ensure precise and reliable annotations, we instructed the annotators to meticulously identify and verify the exact interactable regions of the GUI elements while excluding any irrelevant or non-functional areas. The annotations are required to tightly encompass all parts of each element. For instance, the bounding box for a menu item should not only include the visible text but also extend to cover its full clickable area. This approach minimizes ambiguity in the bounding boxes, providing a more consistent and accurate representation of the elements for rigorous evaluation.

2.4 ScreenSpot-Pro-CN

In the case of professional scenarios, it is common for non-English speakers to operate in both their native languages and English. Consequently, it is essential for GUI agents to efficiently manage tasks that involve switching between languages, while accurately interpreting context and instructions across these languages. To reflect this, every task in the benchmark also includes a Chinese instruction translated by GPT-4 and reviewed by the authors who are fluent in both languages. This allows an assessment of the performance and utility of the GUI agent across different language environments.

²It is important to note that constructing an interactive environment to distribute similar to OSWorld [14] is not feasible due to licensing restrictions.

Table 2: Model Performance by Software. The abbreviations used in the table are defined in Table 1.

Model	Development					Creative					CAD				Scientific			Office			OS			Avg			
	AS	PyC	VSC	VM	UE	PS	BI	PR	DR	AI	FL	CAD SW	Inv	Qrs	Vvd	MAT	Org	Evw	Stt	PPT	Exc	Wrd	Lux		mac	Win	
OS-Atlas-7B	8.8	15.4	25.5	34.1	22.9	17.6	22.5	17.3	27.3	3.2	10.5	2.9	3.9	2.9	13.3	26.3	23.7	11.3	54.0	12.2	22.0	12.5	44.0	20.0	20.0	12.3	18.9
UGround (7B)	7.5	7.7	21.8	31.7	20.0	21.6	25.4	17.3	11.4	0.0	14.0	2.9	0.0	7.1	15.6	28.7	23.7	6.5	46.0	0.0	25.5	15.6	36.9	18.0	12.3	2.5	16.5
AriaUI (MOE, 3.9B active)	0.0	3.8	21.8	2.4	0.0	27.5	26.8	17.3	2.3	0.0	12.3	0.0	1.3	1.4	20.0	17.5	21.5	1.6	44.0	6.1	6.1	1.6	36.9	2.0	3.1	2.5	11.3
ShowUI (2B)	3.8	7.7	5.5	22.0	11.4	5.9	7.0	5.8	0.0	3.2	3.5	0.0	0.0	1.4	15.6	5.0	8.6	12.9	16.0	6.1	9.8	6.3	22.6	4.0	10.8	4.9	7.7
CogAgent (18B)	2.5	5.1	16.4	9.8	2.9	11.8	7.0	7.7	0.0	0.0	5.3	0.0	1.3	0.0	11.1	18.8	16.1	1.6	34.0	2.0	6.1	0.0	21.4	2.0	4.6	2.5	7.7
OS-Atlas-4B	1.3	1.3	12.7	2.4	0.0	0.0	2.8	1.9	2.3	3.2	5.3	0.0	0.0	1.4	2.2	3.8	7.5	3.2	20.0	0.0	4.9	0.0	8.3	6.0	0.0	3.7	3.7
MiniCPM-V (7B)	0.0	2.6	9.1	2.4	0.0	3.9	0.0	3.8	0.0	0.0	0.0	0.0	0.0	0.0	6.7	11.3	2.2	1.6	18.0	0.0	4.9	0.0	3.6	0.0	3.1	3.7	3.0
Qwen2-VL-7B	0.0	0.0	5.5	0.0	2.9	2.0	0.0	0.0	0.0	0.0	1.8	0.0	0.0	0.0	2.2	1.3	2.2	0.0	12.0	2.0	2.4	0.0	6.0	2.0	0.0	0.0	1.6
SeeClick (7B)	0.0	0.0	0.0	2.4	0.0	0.0	1.4	1.9	0.0	0.0	0.0	2.9	0.0	5.7	0.0	0.0	0.0	0.0	8.0	2.0	0.0	0.0	2.4	2.0	1.5	1.2	1.1
GPT-4o	0.0	1.3	0.0	2.4	2.9	2.0	0.0	0.0	0.0	0.0	0.0	0.0	1.3	2.9	0.0	1.3	2.2	0.0	2.0	0.0	0.0	1.6	1.2	0.0	0.0	0.0	0.8
QwenVL-7B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1

2.5 ScreenSpot-Pro Statistics

Table 1 summarizes the collected GUI data, encompassing 23 applications across 3 operating systems, offering a level of diversity unmatched by previous benchmarks. The icons constitute 61.8% of the elements, with the remainder being texts. Notably, targets in ScreenSpot-Pro occupy 0.07% of the screenshot area on average, a significant reduction compared to ScreenSpot’s 2.01%.

3 Experiments

3.1 Settings and Metrics

We experimented on several VLMs that support GUI Grounding: QwenVL-7B [18], Qwen2VL-7B [2], MiniCPM-V-2.6 (8B) [19], CogAgent (18B)³ [6], SeeClick (7B) [8], UGround (7B) [16], OSAtlas-4B, OSAtlas-7B [15], ShowUI (2B) [20] and Aria-UI (MOE, 3.9B active) [21]. We precisely evaluate whether the model’s predictions align with the annotated ground truth boxes. Formally, a prediction is considered correct if $x_{min} \leq x_i \leq x_{max}$ and $y_{min} \leq y_i \leq y_{max}$, where x_i, y_i are the predictions, and $x_{min}, x_{max}, y_{min}, y_{max}$ is the ground truth box. For models generating bounding box outputs, we calculate the center point as its prediction.

3.2 Baseline Methods

We hypothesize that the main challenge for the models is the large resolution of the screenshots. Therefore, we come up with several intuitive baselines to perform multi-round grounding to shrink the image size for a more accurate prediction.

Iterative Zooming. Inspired by V*’s iterative approach [22], Iterative Zooming first performs grounding directly on the whole screenshot, and splits the screenshot into smaller patches with equal sizes. At each step, it chooses the patch the prediction falls into to continue searching within. For the splitting strategy, we always use a 2 row \times 2 column split.

Iterative Narrowing. This baseline operates in the same ground-and-zoom procedure as Iterative Zooming, but the patches are cropped to center the prediction. The patch size is set to half the width and height of the image at each step, and the number of iterations is set to 3 to enable a fair comparison with Iterative Zooming. This approach closely aligns with a concurrent work [23].

ReGround. We assess a simple baseline that crops the region surrounding the initial prediction to re-ground and make a final determination. The size of the crop can be manually configured based on the optimal input size of the models.

3.3 Results and Findings

3.3.1 End-to-end models

Models struggle on ScreenSpot-Pro, even the specialist models. The full results of the GUI grounding models are presented in Table 2. OS-Atlas-7B leads the performance with an accuracy of 18.9%, closely followed by UGround and AriaUI. None of the other models achieved an accuracy

³THUDM/cogagent-chat-hf

Table 3: Performance breakdown of various models across application categories on ScreenSpot-Pro.

Model	Development			Creative			CAD			Scientific			Office			OS			Avg		
	Text	Icon	Avg	Text	Icon	Avg	Text	Icon	Avg	Text	Icon	Avg	Text	Icon	Avg	Text	Icon	Avg	Text	Icon	Avg
OSAtlas-7B	33.1	1.4	17.7	28.8	2.8	17.9	12.2	4.7	10.3	37.5	7.3	24.4	33.9	5.7	27.4	27.1	4.5	16.8	28.1	4.0	18.9
UGround (7B)	26.6	2.1	14.7	27.3	2.8	17.0	14.2	1.6	11.1	31.9	2.7	19.3	31.6	11.3	27.0	17.8	0.0	9.7	25.0	2.8	16.5
AriaUI (MOE, 3.9B active)	16.2	0.0	8.4	23.7	2.1	14.7	7.6	1.6	6.1	27.1	6.4	18.1	20.3	1.9	16.1	4.7	0.0	2.6	17.1	2.0	11.3
CogAgent (18B)	14.9	0.7	8.0	9.6	0.0	5.6	7.1	3.1	6.1	22.2	1.8	13.4	13.0	0.0	10.0	5.6	0.0	3.1	12.0	0.8	7.7
ShowUI (2B)	16.9	1.4	9.4	9.1	0.0	5.3	2.5	0.0	1.9	13.2	7.3	10.6	15.3	7.5	13.5	10.3	2.2	6.6	10.8	2.6	7.7
OSAtlas-4B	7.1	0.0	3.7	3.0	1.4	2.3	2.0	0.0	1.5	9.0	5.5	7.5	5.1	3.8	4.8	5.6	0.0	3.1	5.0	1.7	3.7
MiniCPM-V (7B)	7.1	0.0	3.7	2.0	0.0	1.2	4.1	1.6	3.4	8.3	0.0	4.7	2.8	3.8	3.0	3.7	1.1	2.6	4.5	0.7	3.0
Qwen2-VL-7B	2.6	0.0	1.3	1.5	0.0	0.9	0.5	0.0	0.4	6.3	0.0	3.5	3.4	1.9	3.0	0.9	0.0	0.5	2.5	0.2	1.6
SeeClick (7B)	0.6	0.0	0.3	1.0	0.0	0.6	2.5	0.0	1.9	3.5	0.0	2.0	1.1	0.0	0.9	2.8	0.0	1.5	1.8	0.0	1.1
GPT-4o	1.3	0.0	0.7	1.0	0.0	0.6	2.0	0.0	1.5	2.1	0.0	1.2	1.1	0.0	0.9	0.0	0.0	0.0	1.3	0.0	0.8
QwenVL-7B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1

Table 4: Performance of GUI grounding models with Chinese instructions. The abbreviations used in the table are defined in Table 1.

Model	Development					Creative					CAD				Scientific				Office			OS			Avg		
	AS	PyC	VSC	VM	UE	PS	BI	PR	DR	AI	FL	CAD	SW	Inv	Qrs	Vvd	MAT	Org	Evw	Stt	PPT	Exc	Wrd	Lnx		mac	Win
OS-Atlas-7B	11.3	15.4	21.8	34.1	22.9	11.8	23.9	21.2	11.4	6.5	14.0	5.9	3.9	2.9	8.9	23.8	14.0	11.3	44.0	12.2	17.1	10.9	36.9	16.0	16.9	14.8	16.8
AriaUI (MOE, 3.9B active)	0.0	3.8	18.2	2.4	0.0	23.5	12.7	11.5	0.0	0.0	10.5	0.0	0.0	0.0	13.3	18.8	19.4	1.6	52.0	6.1	2.4	0.0	20.2	2.0	6.2	2.5	9.0
UGround (7B)	3.8	2.6	10.9	14.6	8.6	9.8	11.3	3.8	9.1	3.2	7.0	0.0	0.0	4.3	6.7	12.5	10.8	4.8	30.0	2.0	12.2	4.7	6.0	12.0	7.7	3.7	7.7
ShowUI (2B)	3.8	6.4	5.5	22.0	5.7	7.8	4.2	3.8	0.0	0.0	3.5	5.9	2.6	1.4	15.6	7.5	9.7	11.3	18.0	10.2	9.8	1.6	8.3	4.0	10.8	6.2	7.0
CogAgent (18B)	0.0	5.1	10.9	4.9	0.0	5.9	5.6	5.8	0.0	3.2	3.5	0.0	1.3	0.0	6.7	5.0	7.5	1.6	14.0	2.0	1.2	0.0	2.4	4.0	3.1	2.5	3.7
OS-Atlas-4B	0.0	1.3	7.3	0.0	0.0	2.0	2.8	0.0	4.5	0.0	7.0	5.9	0.0	1.4	0.0	3.8	5.4	4.8	12.0	0.0	4.9	1.6	2.4	4.0	0.0	2.5	2.8
MiniCPM-V (7B)	1.3	2.6	3.6	0.0	0.0	0.0	0.0	1.9	0.0	0.0	3.5	0.0	1.3	0.0	4.4	8.8	0.0	0.0	28.0	0.0	3.7	3.1	0.0	0.0	1.5	2.5	2.5
Qwen2-VL-7B	0.0	0.0	3.6	0.0	0.0	2.0	1.4	3.8	0.0	0.0	1.8	0.0	0.0	0.0	4.4	1.3	2.2	1.6	22.0	6.1	2.4	0.0	2.4	0.0	1.5	0.0	2.0
GPT-4o	2.5	0.0	0.0	0.0	2.9	2.0	1.4	3.8	0.0	0.0	0.0	0.0	2.6	1.4	0.0	0.0	2.2	0.0	2.0	0.0	1.2	0.0	0.0	0.0	1.5	0.0	0.9
SeeClick (7B)	0.0	2.6	0.0	0.0	0.0	0.0	2.8	0.0	0.0	0.0	0.0	0.0	0.0	4.3	0.0	0.0	1.1	0.0	8.0	0.0	1.2	0.0	1.2	0.0	1.5	0.0	0.9
QwenVL-7B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	1.5	0.0	0.2

above 10%. Notably, GPT-4o, despite its advanced capabilities, scored only 0.9%, highlighting its limitations for the GUI grounding task.

Icons targets are more difficult to ground than texts. Table 3 demonstrates that the benchmarked models struggle significantly in identifying and grounding icon elements in the GUI, a consistent finding with [8]. The challenge is exacerbated by the specialization required for professional applications, which introduces several issues: 1) the sheer number of functions makes comprehensive text-based descriptions impractical, e.g. Origin’s toolbar (see Figure 6 in the Appendix); 2) these applications often assume users are familiar with the icons and buttons; and 3) the icons carry unique meanings within professional contexts that are rarely encountered in the web data, on which many models are primarily trained.

Chinese Instructions Pose Greater Challenges. As shown in Table 4, most models experienced a significant performance drop when switching to Chinese instructions, with the SOTA model OS-Atlas-7B achieving only 16.8%. Among these, UGround-7B saw the most severe decline, dropping from 16.4% to 7.7%, emphasizing its limitations in bilingual contexts. Interestingly, the performance of GPT-4o and QwenVL-7B improved, although this increase appears insignificant given their overall low scores.

3.3.2 Multi-round methods

Table 5: Comparison of methods on ScreenSpot-Pro with OS-Atlas-7B.

Model	Dev	Creative	CAD	Scientific	Office	OS	Overall		
							Text	Icon	Avg
Iterative Focusing	33.1	27.3	23.8	25.2	43.9	36.2	43.5	10.8	31.0
Iterative Narrowing	34.4	27.3	20.3	29.5	40.9	43.9	43.5	13.1	31.9
ReGround	37.5	38.1	33.3	37.8	59.1	37.8	55.7	15.1	40.2

The simplest ReGround Method achieves best result. The results of methods are compared in Table 5. Interestingly, the simplest baseline ReGround achieved the highest performance with OS-Atlas-7B, reaching 40.2%. Iterative Narrowing slightly outperformed Iterative Focusing, likely due to its superior image-splitting strategy when the target is positioned near the center of the x or y axes.

Ablations on the crop size of ReGround. Table 6 examines the impact of crop size in ReGround on the two top-performing models, OS-Atlas-7B and UGround. Both models exhibit peak performance within specific resolution ranges, with performance declining as image sizes deviate. OS-Atlas-7B achieves its best score with 1024×1024 crops, while UGround performs optimally with 768×768 crops. This behavior is expected: when images are too small, crucial context is lost [23], whereas images that are too large exceed the model’s processing capacity.

Table 6: ReGround Crop size ablation on ScreenSpot-Pro.

Crop Size	512 × 512	768 × 768	1024 × 1024	1280 × 1280
OS-Atlas-7B	25.1	34.2	40.2	40.1
UGround (7B)	27.0	28.8	28.2	26.3

4 Related Works

4.1 GUI Agent

The aspiration of building autonomous agents to assist humans in daily tasks has long intrigued generations researchers. Some early works explored the feasibility of GUI agents based on visual understanding using computer vision (CV) methods [24, 25], while other works [26, 27] use natural language to guide visual localization, attempting to expand the representation scope of GUI agents by integrating both visual and linguistic cues. Recently, advanced Vision-Language Models (VLMs) like GPT-4V [28] and GPT-4o [1] have demonstrated impressive capabilities in multi-modal understanding and reasoning. These advancements have significantly enhanced the intelligence of GUI agents, enabling them not only to process visual information but also to interpret and respond effectively to complex natural language instructions [28, 29, 30, 31] and contribute to making GUI agents more adaptable, context-aware, and capable of handling a wider range of tasks [32, 33, 5, 34]. For instance, CogAgent [6] enhances the model’s understanding of GUI interfaces by allowing it to learn the correspondence between GUI images and XML content. Ferret constructs GUI description texts as input and distills knowledge from GPT-4 [35], thereby achieving improved interaction capabilities. However, while VLMs perform well in GUI captioning and simple VQA tasks, enhancing their ability to accurately identify and indicate the exact locations of operations within GUIs is a critical step toward developing more capable and reliable agents. To address this, recent research efforts such as SeeClick [8] and UGround [16] have increasingly focused on improving the grounding capabilities of VLMs. Compared to purely vision-based models, VLMs demonstrate an advanced understanding of both instructions and interfaces, excelling in grounding tasks by effectively aligning natural language commands with visual elements. This highlights the necessity of establishing a comprehensive evaluation benchmark to systematically assess and drive further progress in this area. Such a benchmark would not only enable fair comparisons between models but also provide insights into their grounding performance across diverse scenarios.

4.2 GUI Benchmarks

Recently, a large number of work has focused on constructing evaluation metrics for various aspects of GUI agents’ capabilities. They aim to reflect the diverse and complex nature of real-world use by involving web, mobile device and computer in an interactive environment. For instance, Mind2Web [10] and VisualWebArena [9] have concentrated on evaluating GUI agents in the context of their ability to interact with and navigate complex web interfaces. AiTW [36], B-MoCA [37] and LlamaTouch [38] have made strides in assessing GUI agents within mobile interfaces. Moreover, OSWorld [14] combines the aforementioned approaches and extends the benchmark of PC from web to general everyday usage application. However, previous benchmarks neglect the significance of professional scenarios. In real-world professional environments, screens often contain multiple application interfaces. Additionally, to ensure the clear display of information within these interfaces, screen resolutions are usually quite high. Therefore, in this work, we introduce our benchmark that builds on the idea of ScreenSpot [8] to address tasks in high-resolution professional scenarios.

5 Conclusion

In conclusion, this paper addresses critical challenges in the development of GUI agents for professional applications. By introducing ScreenSpot-Pro, we offer a comprehensive evaluation testbed

specifically designed to tackle the complexities of real-world professional workflows. Our work marks a step toward more sophisticated and intelligent systems that can seamlessly support users in creative and productivity-driven tasks. As this field continues to evolve, we hope that ScreenSpot-Pro will inspire further improvements in model accuracy and efficiency, unlocking new opportunities for automation and user empowerment across industries.

References

- [1] OpenAI. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>, 2024. Accessed: 2024-11-01.
- [2] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.
- [3] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.
- [4] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.
- [5] Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. Ferret-ui: Grounded mobile ui understanding with multimodal llms, 2024.
- [6] Wenyi Hong, Weihai Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents, 2023.
- [7] Peter C Humphreys, David Raposo, Tobias Pohlen, Gregory Thornton, Rachita Chhaparia, Alistair Muldal, Josh Abramson, Petko Georgiev, Adam Santoro, and Timothy Lillicrap. A data-driven approach for learning to control computers. In *International Conference on Machine Learning*, pages 9466–9482. PMLR, 2022.
- [8] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclck: Harnessing gui grounding for advanced visual gui agents, 2024.
- [9] 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. *ACL*, 2024.
- [10] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36, 2024.
- [11] 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.
- [12] Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users. *arXiv preprint arXiv:2312.13771*, 2023.
- [13] Yanda Li, Chi Zhang, Wanqi Yang, Bin Fu, Pei Cheng, Xin Chen, Ling Chen, and Yunchao Wei. Appagent v2: Advanced agent for flexible mobile interactions. *arXiv preprint arXiv:2408.11824*, 2024.
- [14] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024.

- [15] Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*, 2024.
- [16] Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for gui agents, 2024.
- [17] Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. Visualwebbench: How far have multimodal llms evolved in web page understanding and grounding? *arXiv preprint arXiv:2404.05955*, 2024.
- [18] 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*, 1(2):3, 2023.
- [19] 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.
- [20] Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. *arXiv preprint arXiv:2411.17465*, 2024.
- [21] Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. Aria-ui: Visual grounding for gui instructions. *arXiv preprint arXiv:2412.16256*, 2024.
- [22] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. *arXiv preprint arXiv:2312.14135*, 2023.
- [23] Anthony Nguyen. Improved gui grounding via iterative narrowing. *arXiv preprint arXiv:2411.13591*, 2024.
- [24] Jieshan Chen, Mulong Xie, Zhenchang Xing, Chunyang Chen, Xiwei Xu, Liming Zhu, and Guoqiang Li. Object detection for graphical user interface: old fashioned or deep learning or a combination? In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2020*, page 1202–1214, New York, NY, USA, 2020. Association for Computing Machinery.
- [25] K. Jaganneshwari and S. Djodilatchoumy. A novel approach of gui mapping with image based widget detection and classification. In *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pages 342–346, 2021.
- [26] Toby Jia-Jun Li, Tom Mitchell, and Brad Myers. Interactive task learning from GUI-grounded natural language instructions and demonstrations. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 215–223, Online, July 2020. Association for Computational Linguistics.
- [27] Tao Li, Gang Li, Jingjie Zheng, Purple Wang, and Yang Li. MUG: Interactive multimodal grounding on user interfaces. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 231–251, St. Julian’s, Malta, March 2024. Association for Computational Linguistics.
- [28] OpenAI. Gpt-4v(ision) system card. https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023. Accessed: 2024-02-03.
- [29] Meng Ziyang, Yu Dai, Zezheng Gong, Shaoxiong Guo, Minglong Tang, and Tongquan Wei. VGA: Vision GUI assistant - minimizing hallucinations through image-centric fine-tuning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1261–1279, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

- [30] Izzeddin Gur, Hiroki Furuta, Austin V. Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. In *ICLR*, 2024.
- [31] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.
- [32] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v(ision) is a generalist web agent, if grounded, 2024.
- [33] Zhuosheng Zhang and Aston Zhang. You only look at screens: Multimodal chain-of-action agents, 2024.
- [34] Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception, 2024.
- [35] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [36] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36, 2024.
- [37] Juyong Lee, Taywon Min, Minyong An, Dongyoon Hahm, Haeone Lee, Changyeon Kim, and Kimin Lee. Benchmarking mobile device control agents across diverse configurations. *arXiv preprint arXiv:2404.16660*, 2024.
- [38] Li Zhang, Shihe Wang, Xianqing Jia, Zhihan Zheng, Yunhe Yan, Longxi Gao, Yuanchun Li, and Mengwei Xu. Llamatouch: A faithful and scalable testbed for mobile ui task automation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–13, 2024.

A Annotator

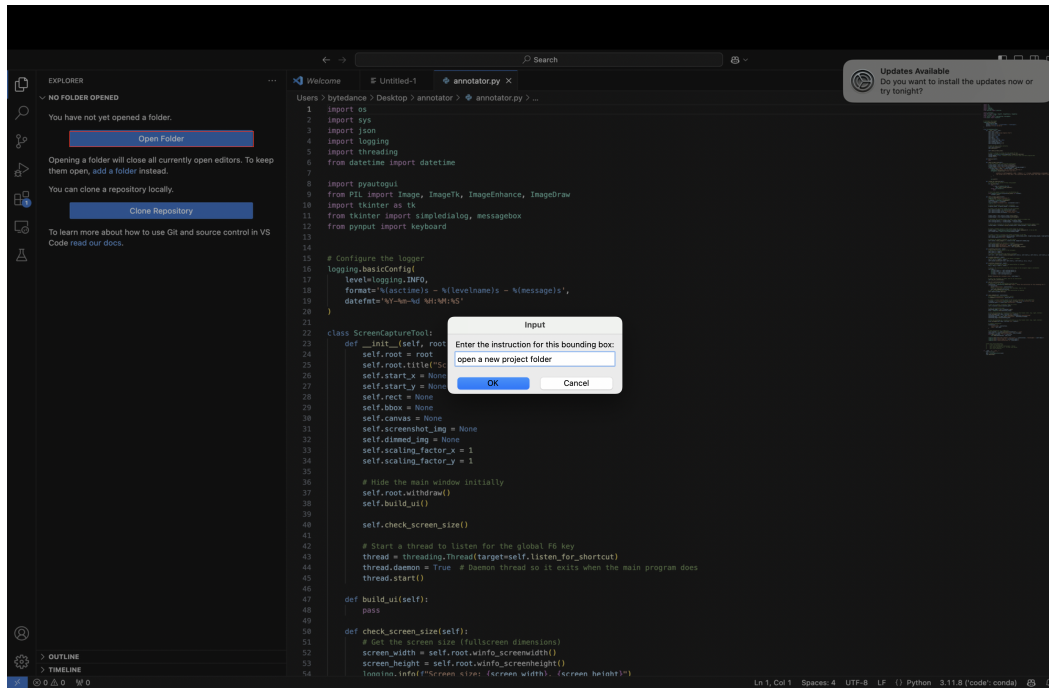
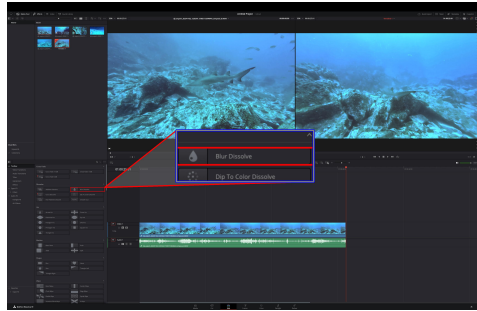
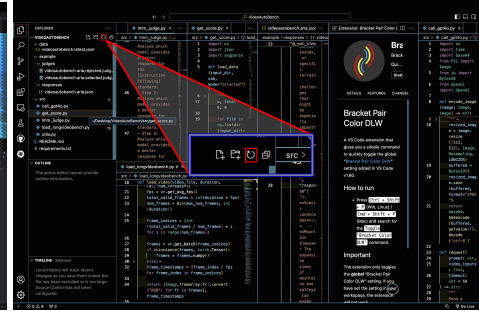


Figure 4: An example of the annotation tool. When activated, the tool captures a screenshot and overlays it on the screen, allowing experts to drag to label the bounding box (the red box around “Open Folder”) and input the instruction in the popup dialog directly.

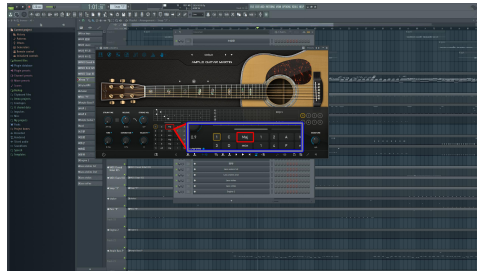
B Case Study



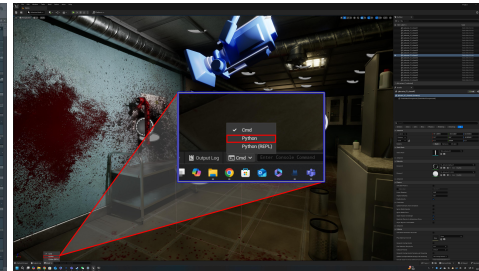
Instruction: *Blur Dissolve.*
Application: *davinci*
Type: *icon*
Bounding Box: [460, 1344, 709, 1375]



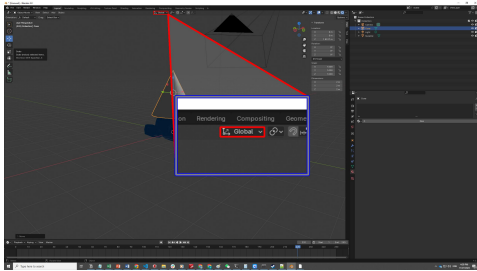
Instruction: *Refresh the file explorer.*
Application: *vscode*
Type: *icon*
Bounding Box: [473, 183, 503, 219]



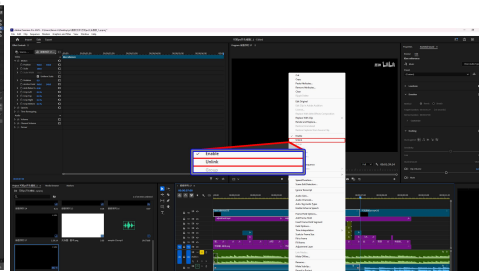
Instruction: *choose chord type for 1.*
Application: *fruitloops*
Type: *text*
Bounding Box: [853, 652, 897, 677]



Instruction: *Execute Python scripts.*
Application: *unreal engine*
Type: *text*
Bounding Box: [246, 2035, 377, 2054]

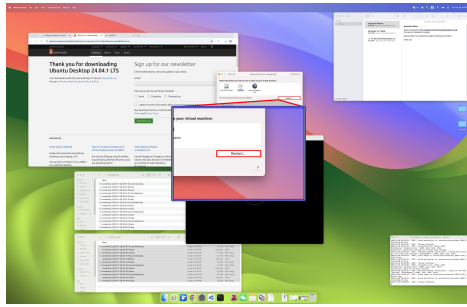


Instruction: *Change the coordinate mode of the object.*
Application: *blender*
Type: *icon*
Bounding Box: [803, 54, 882, 71]

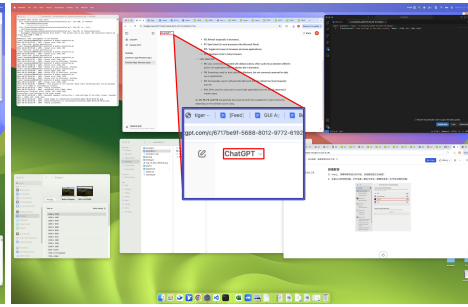


Instruction: *unlink audio and video.*
Application: *premiere*
Type: *text*
Bounding Box: [1499, 592, 1801, 613]

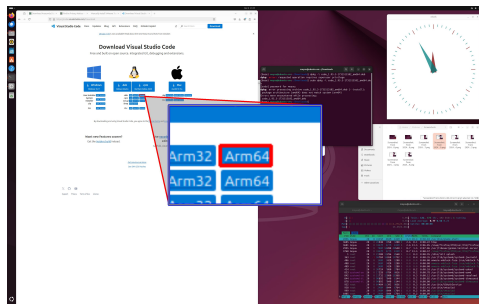
Figure 5: Examples of tasks in ScreenSpot-Pro.



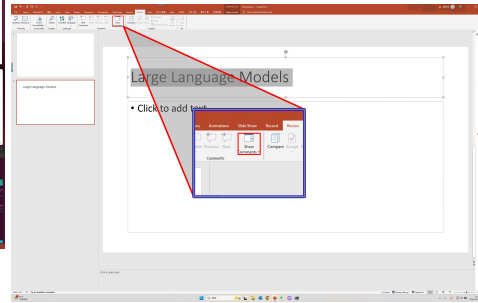
Instruction: restart from CD.
Application: VMWare
Type: text
Bounding Box: [2024, 695, 2188, 718]



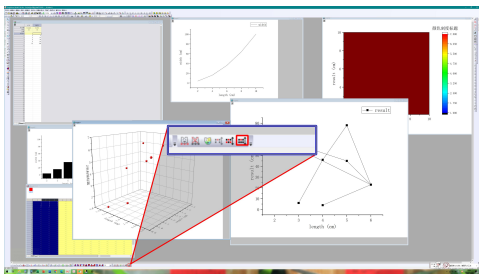
Instruction: Change model.
Application: macOS
Type: text
Bounding Box: [1109, 211, 1209, 236]



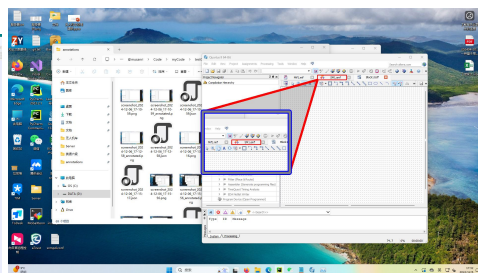
Instruction: select the correct deb package to download according to the error message in the terminal.
Application: linux common
Type: text
Bounding Box: [960, 639, 1001, 655]



Instruction: Show comments.
Application: powerpoint
Type: text
Bounding Box: [614, 72, 681, 136]



Instruction: disable masking.
Application: origin
Type: icon
Bounding Box: [998, 2078, 1021, 2097]



Instruction: select the SMI.smf file in Quartus window.
Application: quartus
Type: text
Bounding Box: [1248, 270, 1365, 289]

Figure 6: More examples of tasks in ScreenSpot-Pro.