FedMABench: Benchmarking Mobile GUI Agents on Decentralized Heterogeneous User Data

Wenhao Wang[1,2,4,*], Zijie Yu[3,4,*], Rui Ye[3,4], Jianqing Zhang[3], Guangyi Liu[1,5], Liang Liu[5], Siheng Chen[4,3,2†], Yanfeng Wang[2,3†]

[1] Zhejiang University [2] Shanghai AI Laboratory [3] Shanghai Jiao Tong University
[4]Multi-Agent Governance & Intelligence Crew (MAGIC) [5] vivo AI Lab

{12321254, guangyiliu@zju.edu.cn}@zju.edu.cn, {zijie.yu, yr991129, tsingz, sihengc, wangyanfeng}@sjtu.edu.cn

Abstract

Mobile GUI agents have attracted tremendous research participation recently. Traditional approaches to mobile agent training rely on centralized data collection, leading to high cost and limited scalability. Distributed training utilizing federated learning offers an alternative by harnessing real-world user data, providing scalability and reducing costs. However, pivotal challenges, including the absence of standardized benchmarks, hinder progress in this field. To tackle the challenges, we introduce FedMABench, the first benchmark for federated training and evaluation of mobile GUI agents, specifically designed for heterogeneous scenarios. FedMABench features 6 datasets with 30+ subsets, 8 federated algorithms, 10+ base models, and over 800 apps across 5 categories, providing a comprehensive framework for evaluating mobile agents across diverse environments. Through extensive experiments, we uncover several key insights: federated algorithms consistently outperform local training; the distribution of specific apps plays a crucial role in heterogeneity; and, even apps from distinct categories can exhibit correlations during training. FedMABench is publicly available at: https://github.com/wwh0411/FedMABench.

1 Introduction

Recent advances in Vision-Language Models (VLMs) (Wang et al., 2021; Jin et al., 2021; Zhou et al., 2022) have significantly propelled the evolution of Graphical User Interface (GUI) agents (Bai et al., 2024; Wang et al., 2024c,a). GUI agents on mobile phones, known as **Mobile Agents**, are capable of automating complex tasks, thereby significantly reducing human workload. Mobile agents have demonstrated promising potential across a wide range of applications (Liu et al., 2024).

The traditional approach for mobile agents largely depends on centralized data collection and training (Hong et al., 2023; Dorka et al., 2024; Chen and Li, 2024), which, although effective, leads to several challenges such as high costs and limited scalability (Sun et al., 2024). Meanwhile, the frequent use of mobile phones by users worldwide naturally generates valuable supervisory information, which serves as a rich data source for training mobile agents. However, this wealth of high-quality data remains underutilized, as it cannot be publicly shared due to privacy concerns (Xiong et al., 2025). Therefore, data from real-world mobile users must be utilized in a distributed manner, where each client locally collects and trains on its own data without direct data transmission.

Continuing to improve the quality and coverage of mobile agents necessitates the development of distributed data collection and training (Wang et al., 2025a). Distributed training mobile agents on user data offers two key advantages: (1) In consideration of the billions of phone users worldwide, collecting data directly from real-world users enables unprecedented scalability. (2) The data collection and annotation costs can be significantly reduced, as user data is an incidental by-product of daily phone usage. Additionally, privacy concerns surrounding the collection of personal data can be effectively mitigated through the application of Federated Learning (FL) (McMahan et al., 2017; Kuang et al., 2023; Wang et al., 2024g), which ensures that sensitive information remains decentralized, thus fostering greater user trust and ensuring compliance with privacy regulations.

Despite the promising potential of training mobile GUI agents on distributed user data via FL, a critical challenge persists: **the absence of standardized benchmarks for federated mobile agents**, which impedes comparisons and advancements in this field. In this context, (1) without diverse and heterogeneous datasets, research efforts cannot effectively address the issue of heterogeneity, which is crucial to utilizing distributed phone

[*]Primary contributing authors. [†]Corresponding authors.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 26399–26420
November 4-9, 2025

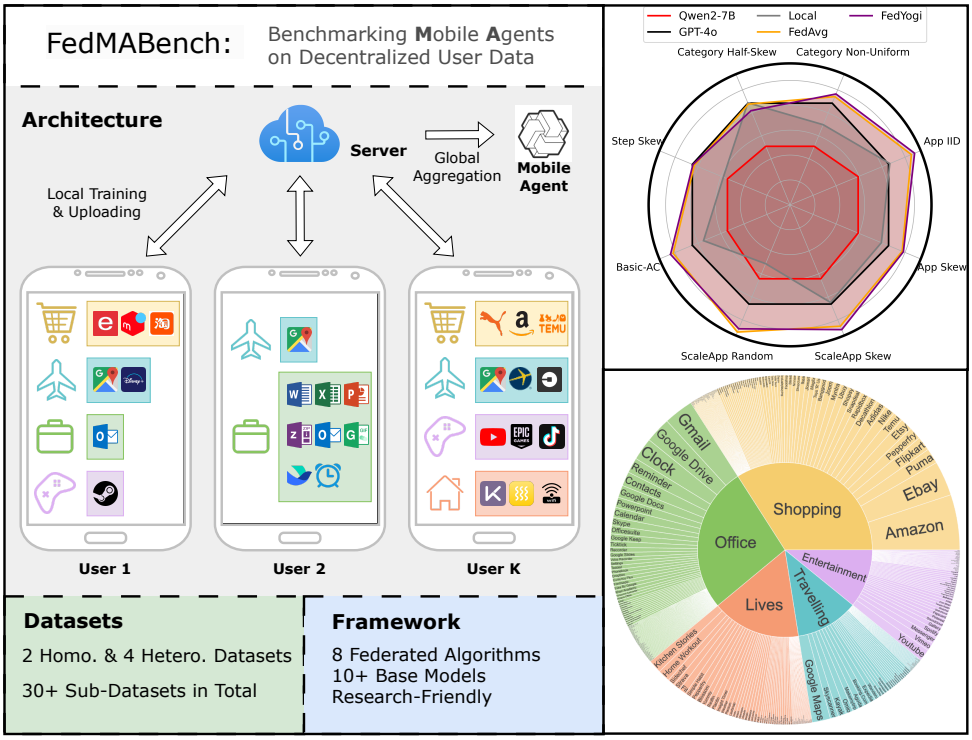


Figure 1: Overview of FedMABench. FedMABench is tailored for benchmarking federated mobile agents trained on distributed mobile user data with diverse types of heterogeneity (top left). We construct two homogeneous dataset and four heterogeneous datasets with 30+ subsets. We also build a research-friendly framework, which integrates eight representative federated algorithms and supports evaluation on more than 10 base models (bottom left). Our datasets cover 877 apps across five categories (bottom right), and the experiments (top right) demonstrate the varying performance of methods under diverse distributions.

usage trajectories. (2) Without an efficient and unified framework, future research may give rise to varied training and evaluation protocols, complicating re-implementations and heightening the risk of unfair comparisons.

To address these challenges, we introduce **FedMABench**, the first benchmark specifically designed for federated training and evaluation of mobile GUI agents, with three key features: (1) **Comprehensiveness**: FedMABench provides a comprehensive framework that integrates eight federated algorithms and supports over ten base models. The evaluation metrics include two performance indicators for both high-level and low-level training, establishing a solid foundation for future research and development. (2) **Diversity**: FedMABench includes thousands of tasks, spanning over 800 apps across five categories from two distinct data sources, yielding substantial diversity. (3) **Heterogeneity**: FedMABench puts strong emphasize on heterogeneous scenarios to promote further research. We incorporate 30+ datasets derived from the original Android Control and Android in the Wild datasets (Rawles et al., 2023; Li et al., 2024b), carefully curated to ensure fair and standardized training and evaluation setups.

Specifically, our datasets address three typical types of heterogeneity, reflecting the diverse mobile usage patterns and preferences of users worldwide: (1) **App Category Distribution**: Each app category addresses a specific type of user need. Since mobile phone usage varies based on users' different needs, the distribution of app categories becomes inherently heterogeneous. (2) **Specific App Preference**: Users exhibit varying preferences for specific apps even with the same function. We construct two series of datasets: one focusing on underlying the differences between apps by selecting the top five apps for experiments and the other expanding the scope with more clients and apps for further validation. (3) **Two-Level Sample Counts**: Mobile agent datasets comprise different number of episodes, where differences in users' tasks and usage patterns lead to additional variations in the number of steps required to complete each episode.

Based on FedMABench, we conduct an exhaustive empirical study to explore federated mobile GUI agents in diverse scenarios, offering new in-

Dataset Name	Distribution Characteristic	N. Subsets	N. Clients	N. Apps	N. Episodes	N. Steps
Basic-AC	Homogeneous	14	10-70	877	7,000+700	47055+4648
Basic-AitW	Homogeneous	5	10-50	-	5,000+500	39394+4447
Step-Episode	Two-Level Sample Counts	4	10	293	1,000+100	6685+635
Category-Level	App Category Distribution	6	5	52	1,000+100	7127+703
App-Level	Specific App Preference	4	5	5	750+100	4456+574
ScaleApp	Specific App Preference	3	30	30	2,500+250	15700+1691

Table 1: Summary of the six dataset series in FedMABench. N. denotes "the number of". The training set and evaluation set are combined by "+". Our datasets span a broad spectrum of homogeneity and heterogeneity, encompassing a variety of apps across five categories.

sights into this area. Through extensive experiments, we make several key observations: (1) FL algorithms consistently outperform local training, providing strong motivation for users to collaborate; (2) The distribution of specific apps is more fundamental to represent heterogeneity than app categories; (3) Even apps from distinct categories can exhibit correlations during training.

In summary, our contributions are:

1. We propose FedMABench, the first benchmark for federated training and evaluation of mobile agents, which is both research-friendly and comprehensive, integrating eight federated algorithms and supporting 10+ base models.
2. We release 6 datasets with 30+ subsets, specifically targeted at three typical types of heterogeneity across various scenarios, simulating real-world user behavior on diverse apps.
3. We conduct extensive experiments to thoroughly investigate the training of federated mobile agents on distributed data with diverse distributions, revealing insightful discoveries.

2 Related Work

2.1 Conventional Centralized Mobile Agents

The emergence of VLMs (Zhang et al., 2024b) has revolutionized phone automation by facilitating more adaptive, contextually aware, and intelligent interactions with mobile devices (Liu et al., 2025). The evolution of mobile agents has undergone several pivotal advancements, with modern models exhibiting enhanced capabilities in processing multi-modal information, discerning user intentions, and autonomously performing intricate user tasks (Zhang et al., 2024d; Nong et al., 2024).

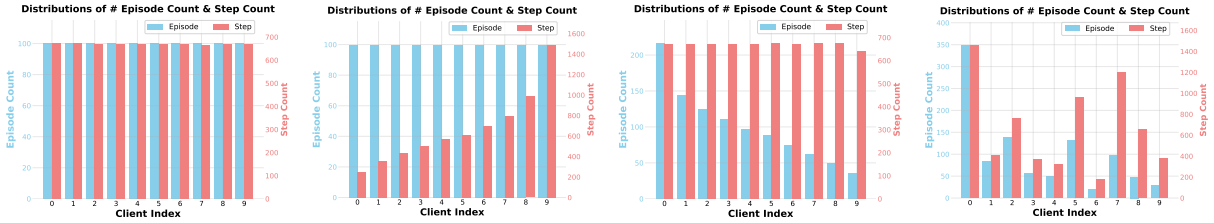
Datasets. Acquiring training trajectories for mobile agents presents considerable challenges. The research community has invested tremendous efforts into constructing high-quality datasets for mobile agents (Rawles et al., 2023; Zhou et al., 2024; Zhang et al., 2024c). However, existing approaches primarily rely on manual curation, rendering data collection both costly and inefficient, and limiting scalability (Gao et al., 2024; Li et al., 2024c).

Benchmarks. Several works have sought to establish efficient benchmarks for mobile GUI agents (Zhang et al., 2024a; Wang et al., 2024a). Yet, none of them is tailored for distributed or federated training. While there are benchmarks for federated Large Language Models (LLMs) (Ye et al., 2024a,b; Wu et al., 2024a), they are not applicable to mobile agent training. This gap significantly obstructs the advancement of federated mobile agents, which offer superior scalability.

2.2 Towards Distributed Mobile Agents

Federated Mobile Agent. FedMobileAgent (Wang et al., 2025a) stands as a pioneering approach that proposes distributed training for mobile agents using self-sourced data from diverse users. It leverages locally deployed VLMs to automatically annotate user instructions and integrates federated learning to collaboratively optimize a global mobile agent. The authors also introduce a novel form of heterogeneity, elaborated in Section 3.3.2. However, the study falls short of further investigating more complexities of heterogeneity, or other real-world scenarios of diverse user phone usage.

Challenges. Federated mobile agents face two major challenges: (1) To facilitate collaboration among a large and diverse set of users with varying usage habits, it is essential to address the issue of heterogeneity (Ye et al., 2023; Qu et al., 2022). This heterogeneity manifests in various forms, such as differing app usage patterns, individual needs, and app preferences for similar functionalities. However, these facets of heterogeneity remain largely unexplored, with vast potential yet to be uncovered. (2) Currently, no publicly available datasets or benchmarks exist for training federated mobile agents. And it is non-trivial to



(a) Step-Episode IID (b) Step Skew (c) Episode Skew (d) Both Skew

Figure 2: Distributions of episode and step counts within the *Step-Episode* Dataset. The four subsets highlight distinct differences in average steps per episode across clients.

effectively capture the heterogeneity that is representative of real-world scenarios by directly downsampling from existing datasets. In this context, FedMABench stands out as the first comprehensive benchmark in the literature, addressing these gaps.

3 FedMABench

3.1 System Overview

FedMABench features a comprehensive framework and six datasets emphasizing on heterogeneity and diversity. As shown in Figure 1 (grey), FedMABench adopts the conventional federated learning protocol and provides an easy-to-use, research-friendly framework that includes eight FL baselines. Specifically, diverse users with heterogeneous data collaboratively train a global mobile GUI agent on their distributed datasets through four iterative steps: server-to-client model broadcasting, local model training, client-to-server model uploading, and global model aggregation.

In real-world scenarios, mobile users exhibit diverse usage habits and preferences, leading to heterogeneous data distributions which are extremely complex and difficult to quantify. To lay the foundation for research on the heterogeneity of distributed data trajectories, we construct two homogeneous datasets and four heterogeneous datasets, addressing diverse aspects of heterogeneity. A summary of the dataset statistics is presented in Table 1.

3.2 Data Collection

Data Composition. To train the core VLM of mobile GUI agents, each data episode, denoted as $\mathcal{D}$, comprises multiple steps, each serving as a basic training unit. A step consists of three components: a task instruction $\mathcal{T}$, a screenshot, and a corresponding action. The data episode is defined as: $\mathcal{D} = \{\langle \mathcal{T}, a_i, s_i \rangle \mid i \in [1, n]\}$, where $\langle \mathcal{T}, a_i, s_i \rangle$ represents the i-th step, with a_i and s_i denoting the action and screenshot respectively. A data example is attached in Figure 6.

Collection. Our datasets are derived from the AndroidControl and Android in the Wild (AitW) datasets, with two key modifications which are labeling and partitioning. Each episode in our datasets is annotated with two app-related attributes: the app name and its corresponding category. Given that the original app and category information is not publicly available in Li et al. (2024b), we are compelled to infer these details based on the actions performed and the instructions provided. We first employ a dual-strategy method, described in Appendix C.1, to extract the related app name for each episode. Following human heuristics, we then categorize the apps into five distinct groups: Shopping, Traveling, Office, Lives, and Entertainment. We employ GPT-4o to automatically assign each app a corresponding category. Details regarding the categorization is presented in Table 19.

Subsequently, we partition each constructed dataset into multiple subsets to simulate the federated learning environment, where each subset represents a distinct data distribution. We specifically control the variables and ensure that subsets are only different in the distribution to provide the fairest possible comparison.

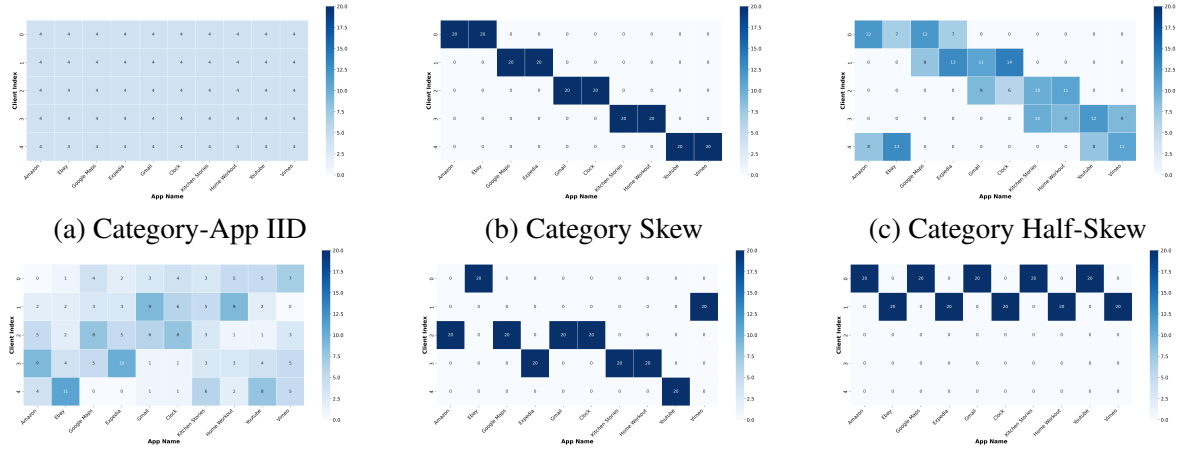
3.3 Datasets Description

To establish a comprehensive foundation for research, we construct six datasets in FedMABench, emphasizing on different forms of homogeneity or heterogeneity. This section provides detailed descriptions and visualizations of these datasets, with additional details available in Appendix C.1.

3.3.1 Basic-AC and Basic-AitW Datasets

Initially, we introduce two basic datasets with homogeneous distributions, to validate general principles and properties of federated mobile agents.

Description of Basic-AC Dataset. *Basic-AC* is constructed from Li et al. (2024b) based on homogeneous distributions, where we disregard the



(a) Category-App IID (b) Category Skew (c) Category Half-Skew

(d) Category Non-Uniform (e) App Random (f) App Skew

Figure 3: Distributions of the top 10 apps across five clients in *Category-Level*. The top two apps from each of the five categories are selected. Our six subsets exhibit diverse patterns across clients.

app attributes of all episodes. Since all episodes are available in this IID setup, we construct six subordinate datasets with increasing data sizes (200-7,000) by random sampling. Additionally, we create five subsets, each consisting of episodes from a single category, to provide more focused scenarios. Basic-AC offers diverse situations with varying data sizes and client participation, enabling the exhaustive evaluation of federated mobile agents under IID settings. For each training set, we sample 10% of the training size to form the test set.

Description of Basic-AitW Dataset. To establish a comprehensive experimental foundation with multiple sources, we construct another homogeneous dataset, named *Basic-AitW*, derived from AitW (Rawles et al., 2023). We sample 1,000 episodes from each category to form five subsets. The Basic-AitW dataset offers distinct data characteristics compared to Basic-AC, adding further diversity for benchmarking federated mobile agents.

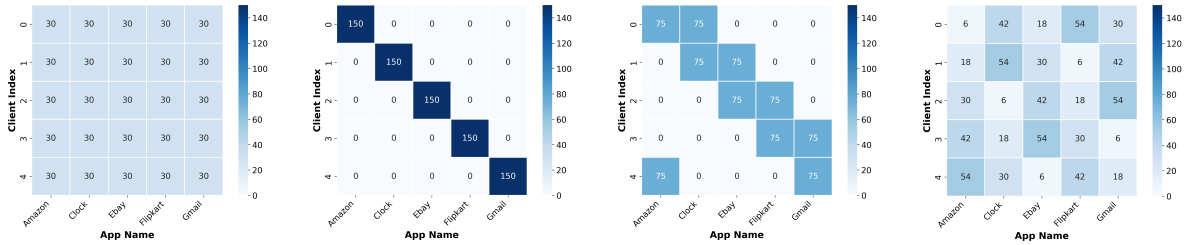
3.3.2 Step-Episode Dataset

Step-Episode Two-Level Heterogeneity. As pointed out in FedMobileAgent (Wang et al., 2025a), the distributed user data for training mobile GUI agents exhibits heterogeneity at two levels: step counts and episode counts, due to variances in users' app usage habits. Unlike traditional federated learning tasks, such as image classification or sentiment analysis, the datasets for training federated mobile agents are characterized by two types of quantity measurements: one based on episode counts and the other based on step counts. As usage habits vary across different users, these two types of measurements do not necessarily align, leading to a unique form of heterogeneity that cannot be adequately captured by the conventional "sample count" perspective. Therefore we refer to this heterogeneity as "step-episode two-level".

Description & Visualization. To evaluate federated mobile agents under *step-episode two-level heterogeneity*, we design four subsets based on a common data pool split among clients using different partition rules. To reduce other heterogeneity factors like app usage, we randomly sample from the pool to create the Step-Episode Dataset. The four subsets are as follows: (1) **Step-Episode IID**: All clients have identical step counts and episode counts. (2) **Episode Skew**: Clients share similar total step counts, but exhibit skewed episode counts. (3) **Step Skew**: All clients have the same episode count, but distinct total step counts. (4) **Both Skew**: Both episode and step counts are heterogeneous across clients. As shown in Figure 2, the four subsets yield distinct step and episode counts, offering valuable signals for evaluating mobile agents under diverse data distributions.

3.3.3 Category-Level Dataset

App Category Heterogeneity. In real-world user phone usage, the users have various app using habits. As showcased in Figure 1 (grey), some users such as "User 1", use mobile phones mostly for shopping and traveling needs, while others such as "User 2" may often utilize phones for office needs. Such using habits and needs result in heterogeneous training data for federated mobile agents as the category distributions differ among users.



(a) App IID (b) App Skew (c) App Half-Skew (d) Non-Uniform

Figure 4: Distributions of the five apps across *App-Level*. Our subsets reveal distinct differences in the heterogeneity of app usage. Note that the numbers represent episode counts, and the episodes are identical for all subsets.

Supported Base Models	
Qwen:	Qwen2-VL-2B/7B-Instruct, Qwen-VL-Chat
Intern:	InternVL2-1B/2B/4B/8B
DeepSeek:	DeepSeekVL2, DeepSeekVL2-tiny/small
OpenAI:	GPT-4o, GPT-4o-mini, GPT-4-Vision
Integrated FL Algorithms	
FedAvg, FedProx, SCAFFOLD, FedAvgM, FedAdam, FedYogi, FedAdagrad, FedMobileAgent	

Table 2: Supported base models and FL algorithms.

Description & Visualization. To investigate how mobile agents using classic FL methods perform, we sample 1,000 episodes from *Basic-AC* to form the *Category-Level* Dataset which consists of 5 categories with 52 apps. To control and monitor the influence of different apps, we select only those apps with a large number of episodes for research efficiency. The sub-datasets are as follows: (1) **IID**: Each app is evenly allocated across all five clients, meaning each client has the same number of episodes for every app and app category. (2) **Category Skew**: The distribution of app categories is highly skewed, as each client possesses only one unique category. (3) **Category Half-Skew**: Similar to Category Skewed, each client has access to two categories, with an even distribution over the two seen categories. (4) **Category Non-Uniform**: All clients have seen all five categories, but the distribution of categories varies across clients. (5) **App Skew**: Each client has five categories of apps, but within each category, a particular app is only seen by one client. In other words, the category distribution is IID across clients, but the specific apps within each category are completely different. (6) **App Random**: Each app is only seen by one client, with apps randomly assigned to clients. Figure 3 highlights the notable distinctions between subsets.

3.3.4 App-Level and ScaleApp Datasets

To evaluate on the app-level heterogeneity instead of categories, we build a concise dataset called *App-Level* targeted at 5 apps, and another dataset *ScaleApp* with scaled app and client numbers.

App Name Heterogeneity. In real life, mobile phone users exhibit distinct preferences for specific apps, even among those that serve similar functions. Therefore, this form of heterogeneity cannot be measured from the perspective of app categories, but rather by app names. As showcased in Figure 1, "User K" prefers Amazon over eBay for purchasing products and Epic over Steam for gaming, resulting in heterogeneity in the specific apps used.

Description of App-Level Dataset. We construct a series of datasets aimed at capturing this diversity in apps. To make the distinction more apparent and straightforward for research comparison, we select five apps with the highest usage frequencies: Amazon, Clock, eBay, Flipkart, and Gmail. Given the limitations in available data samples for each individual app, we sample 150 episodes for each app. Subsequently, we create four representative subsets following similar insights to those described in Section 3.3.3: (1) **App-IID:** All clients share the same number of episodes for each app. (2) **App Skew:** Each client has data collected from only one specific app. (3) **App Half-Skew:** Each client has access to two apps with an equal distribution of episodes. (4) **App Non-Uniform:** All clients have seen all five apps but with varying distributions of data. To facilitate comprehensive research, we provide a test dataset with an equal number of episodes for each of the five apps.

3.4 Framework Description

FedMABench integrates eight typical federated learning algorithms and supports more than ten base models. Our supported models and implemented methods are summarized in Table 2. In addition, we establish an end-to-end pipeline that offers two training paradigms: high-level and low-level training, each can be evaluated using two

Algorithm	Basic-AC			Basic-AitW					
	High-Level	Low-Level							
	Step Acc	Step Acc	Episode Acc	General	Install	G-Apps	Single	WebShopping	**Avg.**
Zero-Shot	27.24	52.13	6	15.90	5.20	15.08	28.38	11.41	15.19
Central	55.59	80.47	27	35.04	54.50	46.65	55.46	39.82	46.29
Local 0	37.64	70.87	20	35.21	52.47	36.03	45.41	32.04	40.23
FedAvg	50.87	78.90	33	36.56	51.84	38.27	54.59	33.59	42.97

Table 3: Experiments on the *Basic-AC* and *Basic-AitW* Datasets. FedAvg consistently surpasses Local Learning, validating the effectiveness of training mobile agents on distributed user data. Local 0 denotes the 0-th client.

metrics: step-level accuracy and episode-level accuracy. In this framework, a low-level instruction refers to a fine-grained, atomic command that corresponds to a single execution step, while a high-level instruction encapsulates the overarching task goal. Compared to high-level training, low-level training provides the agent with additional guidance in the form of explicit subgoals as input at each step.

We build our framework based on the well-known repository `ms-swift` (Zhao et al., 2024). It is important to note that incorporating federated learning support is non-trivial, as we decompose the training pipeline and successfully integrate federated training in a concise manner, which facilitates the easy reproduction of other algorithms.

4 Experiments

4.1 Basic Setups (Details in Appendix C)

Base Model. We employ Qwen2-VL-7B-Instruct (Wang et al., 2024b) as the base model for most of our experiments. We use Low-Rank Adaptation (LoRA) (Hu et al., 2021) for efficient fine-tuning as the resources are limited on mobile phones.

Training Configuration. We train every model for 10 rounds and sample 10% the total dataset at each round. In most settings, we randomly sample 3 clients to participate each round to simulate real-world scenarios where users are occasionally offline (Jiang et al., 2024).

Metrics. We adopt a two-tier evaluation: *Step Accuracy* measures precision at the action level by checking if the predicted response matches the ground truth based on TF-IDF similarity. *Episode Accuracy* evaluates task execution success, requiring all steps in an episode to be correct.

4.2 Experiments on Basic-AC & Basic-AitW

Setups. The experiments are based on the two homogeneous datasets to examine the general properties of federated mobile agents. From all available subsets we choose those with 1,000 episodes as representatives. We evaluated four methodologies on behalf of all baselines, using step-level accuracy as the primary evaluation metric. For Basic-AC, we perform both high-level training and low-level training. Since the episode accuracies of high-level training are close, we omit them for brevity. For Basic-AitW, we experiment on each subset separately and provide the average results as well.

Algorithm	IID	Episode	Step	Both	**Avg.**
Qwen2-VL-7B	27.24				27.24
GPT-4o	42.52				42.52
Central	55.59				55.59
Local 0	37.64	33.39	29.13	**46.77**	36.73
FedAvg	**43.78**	40.63	40.63	40.81	41.46
FedProx	42.36	41.10	40.16	40.16	40.95
FedAvgM	42.00	41.57	41.10	40.47	41.29
FedYogi	42.05	41.10	**41.26**	42.05	41.62
FedAdagrad	43.31	41.42	41.10	41.26	41.77
SCAFFOLD	41.73	41.42	**41.26**	39.84	41.06
FedMobileAgent	42.68	**41.89**	**41.26**	46.53	**43.09**

Table 4: Experiments with multiple baselines on the Step-Episode Dataset. In this setting, FedMobileAgent achieves best performance on average and outperforms GPT-4o, one of the SOTA VLMs.

Results. From Table 3, we draw the following conclusions: (1) Federated learning effectively leverages distributed user data, as evidenced by the noticeable improvement of FedAvg over local training on both the Basic-AC and Basic-AitW Datasets. However, the performance of FedAvg still falls short of centralized training, which aligns with expectations. (2) Federated learning yields varying levels of improvement across different subsets of Basic-AitW, highlighting the impact of different data types and laying the foundation for exploring heterogeneity in the following sections.

4.3 Experiments on Step-Episode Dataset

Setups. We compare seven baselines and two base models on all four subsets: Step-Episode IID, Episode Skew, Step Skew and Both Skew (short for IID, Episode, Step and Both in Table 4 respectively). The evaluation dataset is consistent to provide straightforward comparison, which is why the

Algorithm	Shop	Travel	Office	Lives	Entertain.	Avg.	Algorithm	Shop	Travel	Office	Lives	Entertain.	Avg.
Zero-Shot	26.61	25.33	27.05	24.41	23.81	25.46	Central	57.26	58.67	51.64	55.12	60.95	56.90
Homo.	**Category IID**						**Hetero.**	**Category Skew**					
Local 0	48.39	45.78	36.89	32.28	45.71	42.25	Local 0	50.81	47.56	46.72	38.58	48.57	46.51
FedAvg	**55.65**	52.00	52.46	37.80	**51.43**	50.07	FedAvg	52.42	52.00	**48.36**	41.73	**52.38**	49.64
FedProx	53.23	52.44	51.64	**38.58**	**51.43**	49.79	FedProx	51.61	52.44	47.54	41.73	49.52	49.08
FedAvgM	**54.84**	52.89	50.00	**38.58**	49.52	49.64	FedAvgM	**54.84**	52.89	**48.36**	**42.52**	**52.38**	**50.50**
FedYogi	54.84	**53.78**	**52.46**	**38.58**	50.48	**50.50**	FedYogi	54.03	**53.78**	**48.36**	41.73	51.43	50.36
Hetero.	**Category Half-Skew**						**Hetero.**	**Category Non-Uniform**					
Local 0	41.13	**56.00**	36.89	**40.16**	37.14	44.38	Local 0	38.71	33.78	34.43	34.65	33.33	34.85
FedAvg	46.77	47.11	39.34	36.22	**47.62**	43.81	FedAvg	**50.00**	48.89	47.54	40.94	46.67	47.08
FedProx	**47.58**	49.33	**42.62**	39.37	46.67	**45.66**	FedProx	47.94	**52.42**	**50.22**	**45.90**	42.52	46.67
FedAvgM	45.16	48.89	**42.62**	38.58	45.71	44.81	FedAvgM	48.39	51.56	46.72	43.31	**48.57**	**48.22**
FedYogi	43.55	46.22	36.07	34.65	40.00	40.97	FedYogi	46.77	52.00	47.54	43.31	**48.57**	**48.22**

Table 5: Experiments on *Category-Level*. FL algorithms exhibit diverse behaviors with non-IID distributions result in slightly lower accuracy. Entertain. is short for Entertainment. Colors represent homogeneity and heterogeneity.

results are identical for centralized learning and base models across subsets. Note that we intentionally evaluate FedMobileAgent with the parameter λ set to 7 (around the average steps per episode), which is designed to balance the two-level heterogeneity in both step and episode counts.

Results. As shown in Table 4, the results indicate that: (1) The presence of two-level heterogeneity in step and episode counts is evident, as there is a clear performance drop when the federated trained mobile agents shift from IID scenarios to other non-IID scenarios. (2) Different federated learning algorithms exhibit distinct behaviors in response to this heterogeneity. Overall, FedMobileAgent(Wang et al., 2025a), which leverages a weighted aggregation of each client's total steps and episodes, demonstrates the best performance under these heterogeneous conditions. This approach effectively captures the disparities in data contributions across clients, thereby mitigating the performance drop caused by the two-level sample count heterogeneity. (3) It is surprising at first sight, that Local 0 performs exceptionally well on the Both Skew subset. However, Figure 2 (d) shows that the 0-th client holds a large portion of the total data, which explains its superior performance.

4.4 Experiments on Category-Level Dataset

Setups. We construct 6 subsets to examine how federated mobile agents behave with heterogeneous app category distributions. Due to page limits, we present 4 subsets in Table 5, with the remaining provided in the Appendix (Table 10 and 11). The red color and blue color represent homogeneous and heterogeneous datasets respectively. We evaluate performance across all five category and report the average accuracy across all test samples.

Results. In our constructed hierarchy, heterogeneity escalates from mild to severe as we progress from Category IID $\rightarrow$ Non-Uniform $\rightarrow$ Half-Skew $\rightarrow$ Skew. However, the general accuracy results in Table 5 rank as Category IID $>$ Skew $>$ Non-Uniform $>$ Half-Skew, which does not precisely align with the expected heterogeneity levels. These results suggest that: (1) App category heterogeneity exists and degrades federated learning performance, as nearly all algorithms show a performance drop when transitioning from homogeneous to heterogeneous scenarios. (2) Despite explicit shifts in category distributions, the results on the Category Skew subset remain statistically comparable to those on the Category IID subset. This suggests that category differences lead to domain-invariant representations (i.e., features common across categories, such as temporal usage patterns) which counteract the harmful effects of heterogeneity. In summary, app category differences are not the fundamental cause of heterogeneity.

4.5 Experiments on App-Level Dataset

Setups. The App-Level Dataset encompasses 5 apps: Amazon, Clock, Ebay, Flipkart and Gmail. We evaluate all 5 apps and report their average performance across four subsets. The color scheme follows the same convention as in Section 4.4. Additionally, we include more results from training on the 1-st and 2-nd clients to offer more comparative insights and useful findings.

Results. As shown in Table 6, we conclude the following: (1) The presence of app heterogeneity is evident, as there is a clear performance drop when the model shifts to heterogeneous situations. (2) We further observe a positive correlation between the severity of app name heterogeneity and per-

Algorithm	Amazon	Clock	Ebay	Flipkart	Gmail	Avg.	Algorithm	Amazon	Clock	Ebay	Flipkart	Gmail	Avg.
Zero-Shot	29.75	32.38	28.33	30.00	28.12	29.62	Central	54.55	64.76	58.33	61.00	51.56	57.67
Homo.			**App IID**				**Hetero.**			**App Skew**			
Local 0	44.63	49.52	41.67	50.00	33.59	43.38	Local 0	**56.20**	36.19	42.50	44.00	21.09	39.72
Local 1	46.28	**57.14**	52.50	54.00	39.06	49.30	Local 1	33.06	**60.00**	38.33	31.00	28.91	37.80
Local 2	54.55	53.33	51.67	51.00	38.28	49.48	Local 2	40.50	17.14	45.00	37.00	20.31	32.06
FedAvg	57.02	53.33	52.50	55.00	46.88	52.79	FedAvg	48.76	53.33	**48.33**	52.00	42.97	48.78
FedProx	55.37	53.33	**55.00**	54.00	44.53	52.26	FedProx	48.76	53.33	**48.33**	**54.00**	39.84	48.43
FedAvgM	**58.68**	52.38	54.17	54.00	46.88	53.14	FedAvgM	49.59	53.33	**48.33**	52.00	39.84	48.26
FedYogi	57.02	54.29	54.17	**58.00**	**48.44**	**54.18**	FedYogi	48.76	54.29	47.50	**54.00**	**43.75**	**49.30**
Hetero.			**App Half-Skew**				**Hetero.**			**App Non-Uniform**			
Local 0	52.89	**57.14**	45.00	40.00	36.72	46.17	Local 0	39.67	**58.10**	38.33	48.00	**46.09**	45.64
Local 1	**57.02**	53.33	**50.00**	47.00	28.91	46.86	Local 1	52.89	56.19	38.33	47.00	39.84	46.52
Local 2	50.41	40.95	41.67	**58.00**	28.91	41.64	Local 2	47.11	49.52	45.00	**55.00**	40.62	47.04
FedAvg	54.55	53.33	45.83	55.00	38.28	48.95	FedAvg	56.20	55.24	45.83	51.00	42.19	49.83
FedProx	56.20	55.24	43.33	55.00	38.28	49.13	FedProx	**57.02**	55.24	45.83	50.00	38.28	48.95
FedAvgM	54.55	53.33	45.00	54.00	**42.19**	**49.48**	FedAvgM	55.37	54.29	45.83	50.00	41.41	49.13
FedYogi	54.55	51.43	44.17	55.00	41.41	48.95	FedYogi	55.37	55.24	**46.67**	52.00	42.19	**50.00**

Table 6: Experiments on the App-Level Dataset. We provide evaluation results on all five apps. FL algorithms in skewed app distributions perform significantly lower accuracy compared to IID situations.

formance degradation, confirming that this form of heterogeneity not only exists but critically impacts model effectiveness in real-world deployment contexts. (3) In comparison with the results from *Category-Level*, we find that differences in specific app names contribute more significantly to heterogeneity than app categories. (4) Overall, FedYogi (Reddi et al., 2020) outperforms other representative FL algorithms. (5) Notably, we observe that the 1-st client in the App Half-Skew subset, which only has access to episodes from Clock and Ebay, outperforms all FL baselines on Amazon. We hypothesize that there may be underlying relationships between these apps that warrant further exploration.

4.6 Efficiency Evaluation

We conduct additional evaluations focusing on communication and computation efficiency.

Communication. For approximation, we assume that all episodes have equal data size, and compare the communication overhead of three approaches: (1) centralized training (one round of full dataset transmission), (2) federated training with full fine-tuning (transmitting full model parameters in each round), and (3) federated training with LoRA (transmitting only LoRA adapters in each round). As shown in Table 7, LoRA-based FL is the most communication-efficient method. It is worth noting that the communication cost is identical across different federated algorithms used in our experiments.

Computation. We further report the computation cost of FedAvg using three representative VLMs in Table 8. These results reinforce the training efficiency of FedMABench, which can be fully executed on a single RTX 4090 GPU.

Approach	Overhead
Central + Unpacked data (10k)	≈100 GB
Central + Unpacked data (1k)	≈10 GB
Central + Original TFRecord file	50 GB
FL + Full model	16.57 GB × round
FL + LoRA adapter (rank=8,α=32)	77.06 MB × round

Table 7: Communication overhead using Qwen2-VL-7B on Android Control. Transmitting only LoRA adapters yields achieves the highest communication efficiency.

Base Model	GPU Memory (MB)	Time per Round (mm:ss)
Qwen2-VL-7B	21610	2:59
Intern2-VL-1B	10498	6:56
Phi-3.5-Vision	11560	13:20

Table 8: Computational statistics for training VLMs using FedMABench, showing its efficiency in both GPU memory usage and training time.

5 Conclusion

In this paper, we present FedMABench, the first research-friendly and comprehensive benchmark for federated learning of mobile GUI agents, accompanied by six diverse datasets encompassing over 30 meticulously designed subsets that capture representative patterns of real-world heterogeneity. Our extensive experiments reveal insightful discoveries, such as differences in specific app names contribute more significantly to heterogeneity than app categories. Overall, FedMABench bridges the critical gap between theoretical FL research and practical mobile agent applications, laying a solid foundation for future work.

Limitations

Despite its comprehensive framework and diverse datasets, FedMABench still has some limitations. One major challenge lies in the trade-off between constructing datasets from real user interactions and relying on publicly available open-source data. Using real user data would provide more realistic and representative usage patterns, which are valuable for academic research. However, it raises significant privacy and ethical concerns. In contrast, open-source datasets facilitate direct comparison with existing work and pose no barriers to public release, but may lack the authenticity of real-world usage. Due to ethical considerations and the high cost of acquiring real user data on our own, we adopt the latter approach by leveraging the AndroidControl and Android in the Wild datasets. This strategy inevitably falls short in terms of realism compared to private user data, although it offers a reasonable simulation of actual user trajectories.

Another limitation is that our analysis does not provide an in-depth examination of the linguistic complexity inherent in GUI interactions. Such interactions often involve highly complex instructions and responses that are of high value for research. Because our primary focus is on heterogeneity in application usage rather than on linguistic challenges, we leave this aspect for future work.

References

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. 2024. DigiRL: Training In-The-Wild Device-Control Agents with Autonomous Reinforcement Learning. *Preprint*, arXiv:2406.11896.

Simone Caldarella, Massimiliano Mancini, Elisa Ricci, and Rahaf Aljundi. 2024. The phantom menace: Unmasking privacy leakages in vision-language models. *Preprint*, arXiv:2408.01228.

Wei Chen and Zhiyuan Li. 2024. Octopus v2: On-device language model for super agent. *Preprint*, arXiv:2404.01744.

Shengwen Ding and Chenhui Hu. 2024. efedllm: Efficient llm inference based on federated learning. *arXiv preprint arXiv:2411.16003*.

Nicolai Dorka, Janusz Marecki, and Ammar Anwar. 2024. Training a Vision Language Model as Smartphone Assistant. *Preprint*, arXiv:2404.08755.

Yaxin Du, Rui Ye, Fengting Yuchi, Wanru Zhao, Jingjing Qu, Yanfeng Wang, and Siheng Chen. 2025. FedDQC: Data quality control in federated instruction-tuning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15267–15291, Vienna, Austria. Association for Computational Linguistics.

Wenzhi Fang, Dong-Jun Han, Liangqi Yuan, Seyyedali Hosseinalipour, and Christopher G. Brinton. 2025. Federated sketching lora: On-device collaborative fine-tuning of large language models. *Preprint*, arXiv:2501.19389.

Longxi Gao, Li Zhang, Shihe Wang, Shangguang Wang, Yuanchun Li, and Mengwei Xu. 2024. Mobileviews: A large-scale mobile gui dataset. *Preprint*, arXiv:2409.14337.

Wenyi Hong, Weihan 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. 2023. CogAgent: A Visual Language Model for GUI Agents. *Preprint*, arXiv:2312.08914.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2021. Lora: Low-rank adaptation of large language models. In *ICLR*.

Bargav Jayaraman, Chuan Guo, and Kamalika Chaudhuri. 2024. Déjà vu memorization in vision-language models. *Preprint*, arXiv:2402.02103.

Bonian Jia, Huiyao Chen, Yueheng Sun, Meishan Zhang, and Min Zhang. 2024. Llm-driven multimodal opinion expression identification. In *Proc. Interspeech 2024*, pages 2930–2934.

Zhifeng Jiang, Wei Wang, and Ruichuan Chen. 2024. Dordis: Efficient federated learning with dropout-resilient differential privacy. In *Proceedings of the Nineteenth European Conference on Computer Systems*, pages 472–488.

Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. 2021. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. *arXiv preprint arXiv:2110.08484*.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. 2023. Federatedscope-llm: A comprehensive package for

fine-tuning large language models in federated learning. *arXiv preprint arXiv:2309.00363*.

Peichun Li, Hanwen Zhang, Yuan Wu, Liping Qian, Rong Yu, Dusit Niyato, and Xuemin Shen. 2024a. Filling the missing: Exploring generative ai for enhanced federated learning over heterogeneous mobile edge devices. *IEEE Transactions on Mobile Computing*.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2024b. On the Effects of Data Scale on Computer Control Agents. *Preprint*, arXiv:2406.03679.

Zhangheng Li, Keen You, Haotian Zhang, Di Feng, Harsh Agrawal, Xiujun Li, Mohana Prasad Sathya Moorthy, Jeff Nichols, Yinfei Yang, and Zhe Gan. 2024c. Ferret-UI 2: Mastering Universal User Interface Understanding Across Platforms. *Preprint*, arXiv:2410.18967.

William Liu, Liang Liu, Yaxuan Guo, Han Xiao, Weifeng Lin, Yuxiang Chai, Shuai Ren, Xiaoyu Liang, Linghao Li, Wenhao Wang, Tianze Wu, Yong Liu, Hao Wang, Hongsheng Li, and Guanjing Xiong. 2025. Llm-powered gui agents in phone automation: Surveying progress and prospects. *Preprints*.

Xiao Liu, Bo Qin, Dongzhu Liang, Guang Dong, Hanyu Lai, Hanchen Zhang, Hanlin Zhao, Iat Long Iong, Jiadai Sun, Jiaqi Wang, Junjie Gao, Junjun Shan, Kangning Liu, Shudan Zhang, Shuntian Yao, Siyi Cheng, Wentao Yao, Wenyi Zhao, Xinghan Liu, and 11 others. 2024. AutoGLM: Autonomous Foundation Agents for GUIs. *Preprint*, arXiv:2411.00820.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Guram Mikaberidze, Sayantan Nag Chowdhury, Alan Hastings, and Raissa M D'Souza. 2024. Consensus formation among mobile agents in networks of heterogeneous interaction venues. *Chaos, Solitons & Fractals*, 178:114298.

Songqin Nong, Jiali Zhu, Rui Wu, Jiongchao Jin, Shuo Shan, Xiutian Huang, and Wenhao Xu. 2024. MobileFlow: A Multimodal LLM For Mobile GUI Agent. *Preprint*, arXiv:2407.04346.

Georgios Papoudakis, Thomas Coste, Zhihao Wu, Jianye Hao, Jun Wang, and Kun Shao. 2025. Appvlm: A lightweight vision language model for online app control. *Preprint*, arXiv:2502.06395.

Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. 2022. Rethinking architecture design for tackling data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10061–10071.

Arian Raje. 2024. *Communication-Efficient LLM Training for Federated Learning*. Ph.D. thesis, Carnegie Mellon University Pittsburgh, PA.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2023. Android in the Wild: A Large-Scale Dataset for Android Device Control. *Preprint*, arXiv:2307.10088.

Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Hugh Brendan McMahan. 2020. Adaptive federated optimization. In *International Conference on Learning Representations*.

Laurens Samson, Nimrod Barazani, Sennay Ghebreab, and Yuki M. Asano. 2024. Privacy-aware visual language models. *Preprint*, arXiv:2405.17423.

Qiushi Sun, Kanzhi Cheng, Zichen Ding, Chuanyang Jin, Yian Wang, Fangzhi Xu, Zhenyu Wu, Chengyou Jia, Liheng Chen, Zhoumianze Liu, Ben Kao, Guohao Li, Junxian He, Yu Qiao, and Zhiyong Wu. 2024. OS-Genesis: Automating GUI Agent Trajectory Construction via Reverse Task Synthesis. *Preprint*, arXiv:2412.19723.

Jianfeng Wang, Xiaowei Hu, Pengchuan Zhang, Xiujun Li, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. 2021. Minivlm: A smaller and faster vision-language model. *Preprint*, arXiv:2012.06946.

Luyuan Wang, Yongyu Deng, Yiwei Zha, Guodong Mao, Qinmin Wang, Tianchen Min, Wei Chen, and Shoufa Chen. 2024a. MobileAgentBench: An Efficient and User-Friendly Benchmark for Mobile LLM Agents. *Preprint*, arXiv:2406.08184.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Taiyi Wang, Zhihao Wu, Jianheng Liu, Jianye Hao, Jun Wang, and Kun Shao. 2024c. DistRL: An Asynchronous Distributed Reinforcement Learning Framework for On-Device Control Agents. *Preprint*, arXiv:2410.14803.

WenHao Wang, Xiaoyu Liang, Rui Ye, Jingyi Chai, Siheng Chen, and Yanfeng Wang. 2024d. KnowledgeSG: Privacy-preserving synthetic text generation with knowledge distillation from server. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7677–7695,

Miami, Florida, USA. Association for Computational Linguistics.

Wenhao Wang, Zijie Yu, William Liu, Rui Ye, Tian Jin, Siheng Chen, and Yanfeng Wang. 2025a. Fedmobileagent: Training mobile agents using decentralized self-sourced data from diverse users. *Preprint*, arXiv:2502.02982.

Yuzheng Wang, Zhaoyu Chen, Dingkang Yang, Pinxue Guo, Kaixun Jiang, Wenqiang Zhang, and Lizhe Qi. 2024e. Out of thin air: Exploring data-free adversarial robustness distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5776–5784.

Yuzheng Wang, Zhaoyu Chen, Jie Zhang, Dingkang Yang, Zuhao Ge, Yang Liu, Siao Liu, Yunquan Sun, Wenqiang Zhang, and Lizhe Qi. 2024f. Sampling to distill: Knowledge transfer from open-world data. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 2438–2447.

Zezhou Wang, Yaxin Du, Xingjun Ma, Yugang Jiang, Zhuzhong Qian, and Siheng Chen. 2025b. Optimizing cross-client domain coverage for federated instruction tuning of large language models. *Preprint*, arXiv:2409.20135.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. 2024g. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *Preprint*, arXiv:2409.05976.

Feijie Wu, Zitao Li, Yaliang Li, Bolin Ding, and Jing Gao. 2024a. Fedbiot: Llm local fine-tuning in federated learning without full model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 3345–3355.

Haoqi Wu, Wei Dai, Li Wang, and Qiang Yan. 2025. Cape: Context-aware prompt perturbation mechanism with differential privacy. *CoRR*, abs/2505.05922.

Haoqi Wu, Wenjing Fang, Yancheng Zheng, Junming Ma, Jin Tan, and Lei Wang. 2024b. Ditto: Quantization-aware secure inference of transformers upon MPC. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Baochen Xiong, Xiaoshan Yang, Yaguang Song, Yaowei Wang, and Changsheng Xu. 2025. Pilot: Building the federated multimodal instruction tuning framework. *Preprint*, arXiv:2501.13985.

Zhiwei Yao, Jianchun Liu, Hongli Xu, Lun Wang, Chen Qian, and Yunming Liao. 2024. Ferrari: A personalized federated learning framework for heterogeneous edge clients. *IEEE Transactions on Mobile Computing*.

Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. 2023. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44.

Rui Ye, Rui Ge, Xinyu Zhu, Jingyi Chai, Yaxin Du, Yang Liu, Yanfeng Wang, and Siheng Chen. 2024a. Fedllm-bench: Realistic benchmarks for federated learning of large language models. *Preprint*, arXiv:2406.04845.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. 2024b. Openfedllm: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6137–6147.

Da Yu, Peter Kairouz, Sewoong Oh, and Zheng Xu. 2024. Privacy-Preserving Instructions for Aligning Large Language Models. *Preprint*, arxiv:2402.13659.

Jianguo Zhang, Tian Lan, Rithesh Murthy, Zhiwei Liu, Weiran Yao, Ming Zhu, Juntao Tan, Thai Hoang, Zuxin Liu, Liangwei Yang, Yihao Feng, Shirley Kokane, Tulika Awalgaonkar, Juan Carlos Niebles, Silvio Savarese, Shelby Heinecke, Huan Wang, and Caiming Xiong. 2024a. AgentOhana: Design Unified Data and Training Pipeline for Effective Agent Learning. *Preprint*, arXiv:2402.15506.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024b. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5625–5644.

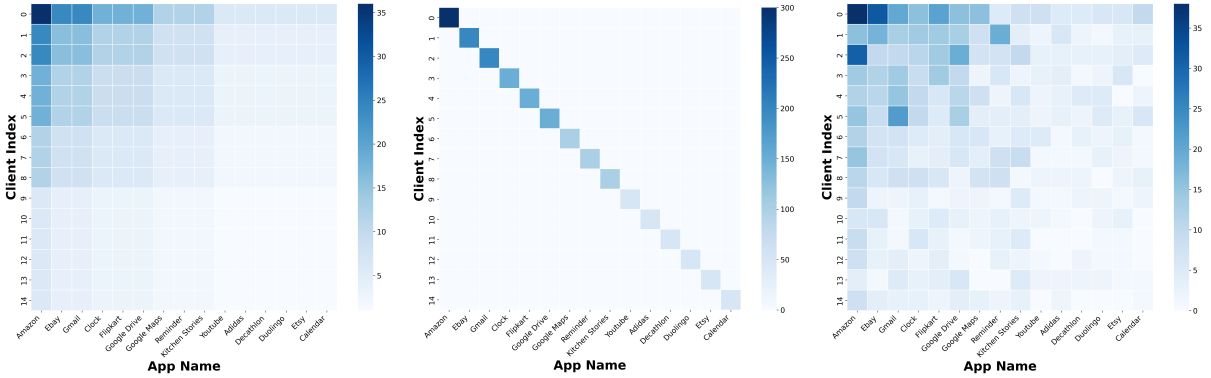
Jiwen Zhang, Jihao Wu, Yihua Teng, Minghui Liao, Nuo Xu, Xiao Xiao, Zhongyu Wei, and Duyu Tang. 2024c. Android in the Zoo: Chain-of-Action-Thought for GUI Agents. *Preprint*, arXiv:2403.02713.

Jiwen Zhang, Yaqi Yu, Minghui Liao, Wentao Li, Jihao Wu, and Zhongyu Wei. 2024d. *UI-Hawk: Unleashing the Screen Stream Understanding for GUI Agents*.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2024. Swift:a scalable lightweight infrastructure for fine-tuning. *Preprint*, arXiv:2408.05517.

Chendi Zhou, Hao Tian, Hong Zhang, Jin Zhang, Mianxiong Dong, and Juncheng Jia. 2021. Tea-fed: time-efficient asynchronous federated learning for edge computing. In *Proceedings of the 18th ACM international conference on computing frontiers*, pages 30–37.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348.



(a) ScaleApp IID (b) ScaleApp Skew (c) ScaleApp Random

Figure 5: Heatmap distribution of the ScaleApp Dataset. We select top 15 apps for visualization.

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. 2024. Webarena: A realistic web environment for building autonomous agents. *Preprint*, arXiv:2307.13854.

A Discussions and Future Directions

Previously, we have shown the promising results achieved by training mobile agents via federated learning. However, this is not the end as there are still emerging challenges and interesting directions that are worth exploring in this field in the future.

A.1 Federated Algorithms for Heterogeneity of Mobile User Data

In FedMABench, we establish the benchmark for evaluating federated mobile agents trained on heterogeneous user data. Our results in Section 4.4 and 4.5 demonstrate that currently no existing federated algorithm can achieve consistently good result when meeting the heterogeneity of diverse app usage. Specifically, FedYogi (Reddi et al., 2020) has noticeable performance drop on the Category Half-Skew subset; FedMobileAgent (Wang et al., 2025a) has no improvement when the distribution is completely skewed.

Recently, there has been some research (Mikaberidze et al., 2024; Yao et al., 2024; Li et al., 2024a) on generative AI and communication optimization for heterogeneous mobile clients. However, none of these studies address app heterogeneity among users. The deployment of federated mobile agents require enhanced performance over diverse data distributions for scalability, which necessitates further research into designing novel FL algorithms to address the heterogeneity of phone usage trajectories, such as privacy-preserving distillation (Wang et al., 2024f,e) and data selection (Du et al., 2025).

A.2 Privacy Preservation in Federated Mobile Agents

Training on user data inevitably raises privacy concerns. While federated learning helps mitigate privacy leakage by keeping private data on the client side and transmitting only LoRA adapters, potential privacy issues remain.

Models with substantial sizes are prone to memorization of their training data (Yu et al., 2024; Wang et al., 2024d). Similar to large LLMs, recent studies (Caldarella et al., 2024; Samson et al., 2024; Jayaraman et al., 2024) reveal that VLMs also inadvertently memorize and potentially expose sensitive information. Dejavu memorization (Jayaraman et al., 2024) proposes a novel measurement for memorization by quantifying the fraction of ground-truth objects in an image that can be predicted from its text description in a training image-text pair. Mobile agents rely on VLMs to perceive the interface and make decisions. Therefore, training directly on user data may lead to leakage of sensitive information.

Federated mobile agents also face the same privacy risks as traditional federated learning, including gradient inversion and membership inference attacks. During transmission, model parameters can be intercepted or exploited, potentially leaking sensitive information about the underlying training data.

The above mentioned issues can be mitigated through techniques such as differential privacy (DP) (Wu et al., 2025) and secure aggregation (Wu et al., 2024b); however, their application to VLMs

Hetero.	Algorithm	Amazon	Clock	Ebay	Flipkart	Gmail	Avg.
-	Zero-Shot	30.68	32.53	39.84	33.33	18.35	32.17
	Central	62.50	68.67	61.72	65.38	63.29	63.10
ScaleApp IID	Local 0	50.00	54.22	51.56	55.13	34.18	46.72
	Local 1	43.75	51.81	39.06	46.15	34.81	44.83
	FedAvg	54.55	60.24	55.47	64.10	47.47	54.35
	FedProx	54.55	59.04	55.47	**65.38**	46.84	54.46
	FedAvgM	54.55	59.04	55.47	61.54	46.84	54.76
	FedYogi	**56.25**	**61.45**	**56.25**	64.10	**48.10**	**55.12**
ScaleApp Random	Local 0	52.27	54.22	53.91	53.85	38.61	49.67
	Local 1	21.59	31.33	28.12	28.21	20.25	25.61
	FedAvg	59.66	**60.24**	57.81	57.69	46.84	55.35
	FedProx	58.52	**60.24**	57.81	60.26	**49.37**	55.59
	FedAvgM	**60.80**	**60.24**	**60.16**	**62.82**	46.84	**55.71**
	FedYogi	59.66	57.83	57.03	60.26	43.04	53.93
ScaleApp Skew	Local 0	**59.66**	39.76	54.69	**61.54**	26.58	42.46
	Local 1	48.30	49.40	48.44	52.56	28.48	41.87
	FedAvg	57.39	57.83	55.47	58.97	40.51	52.81
	FedProx	57.95	57.83	56.25	57.69	42.41	53.40
	FedAvgM	57.95	**59.04**	**58.59**	60.26	43.04	**54.41**
	FedYogi	58.52	57.83	54.69	60.26	**43.67**	54.29

Table 9: Experiments on the ScaleApp Dataset. Skewed app distribution results in lower average accuracy across apps. The long tailed apps with few episodes witness a greater decrease in performance.

and mobile-agent training remains largely underexplored.

A.3 Efficiency and Resources in Federated Mobile Agents

To collaboratively train a global mobile agent on distributed user data, each user needs to locally train a small-sized VLM and communicate with the central server. However, limited computation resources and communication channels on mobile devices may hinder the feasibility of deployment.

With the recent advancement of LLMs, VLMs and diffusion models and their integration into federated learning systems (Zhou et al., 2021; Jia et al., 2024; Wang et al., 2025b), numerous approaches have been proposed to alleviate computational and communication overheads (Ding and Hu, 2024; Raje, 2024; Fang et al., 2025). On the other hand, the proliferation of smaller VLMs has significantly enhanced efficiency. For instance, AppVLM (Papoudakis et al., 2025) specifically targets app control tasks with a lightweight architecture, facilitating rapid and cost-efficient inference for real-time execution.

A.4 Combination of Reinforcement Learning with Federated Mobile Agents

Although our current framework does not yet incorporate reinforcement learning, we identify it as a promising future direction. In a federated mobile agent setting, user feedback can serve as a critical reward signal, enabling agents to adjust their decision-making policies dynamically.

Future work will need to tackle challenges inherent to integrating reinforcement learning into a federated environment, such as handling heterogeneous feedback, ensuring robust and stable learning under variable network conditions, and preserving user privacy. We believe that exploring these issues will pave the way for more adaptive and user-centric mobile agents, ultimately enhancing both their responsiveness and overall utility.

B Additional Experiments

B.1 Experiments on ScaleApp Dataset

Setups. We construct three subsets of the ScaleApp Dataset to further investigate the heterogeneity of specific app preferences. The distribution of subsets are visualized in the heatmaps in Figure 5. We select the top 15 apps to plot as the rest 15 apps have basically the same distribution with the 14-th app. To enhance scalability and increase diversity, we select 30 apps, each with a varying number of episodes, to form a training set consisting of 2,500 episodes. Additionally, we sample 10% of the episodes from each app to form the test set.

Results. From Table 9, we draw the following conclusions: (1) By comparing FedAvg across the

Algorithm	Shopping	Traveling	Office	Lives	Entertain.	Avg.	Algorithm	Shopping	Traveling	Office	Lives	Entertain.	Avg.
Zero-Shot	26.61	25.33	27.05	24.41	23.81	25.46	Central	57.26	58.67	51.64	55.12	60.95	56.90
Homo.	**Category IID**						**Hetero.**	**Category Skew**					
Local 0	48.39	45.78	36.89	32.28	45.71	42.25	Local 0	50.81	47.56	46.72	38.58	48.57	46.51
FedAdagrad	**54.84**	**53.78**	50.00	**39.37**	**50.48**	**50.21**	FedAdagrad	**54.03**	52.44	**48.36**	**42.52**	**51.43**	**50.07**
SCAFFOLD	53.23	52.00	**53.28**	38.58	**50.48**	49.79	SCAFFOLD	**54.03**	**52.89**	47.54	41.73	**51.43**	49.93
Hetero.	**Category Half-Skew**						**Hetero.**	**Category Non-Uniform**					
Local 0	41.13	**56.00**	36.89	**40.16**	37.14	44.38	Local 0	38.71	33.78	34.43	34.65	33.33	34.85
FedAdagrad	**47.58**	46.22	40.98	35.43	40.95	42.82	FedAdagrad	**50.00**	**52.89**	**49.18**	43.31	**48.57**	**49.36**
SCAFFOLD	46.77	48.89	**42.62**	37.80	**42.86**	**44.52**	SCAFFOLD	47.58	52.00	47.54	**44.09**	**48.57**	48.51

Table 10: Supplementary experiments on the Category-Level Dataset with more baselines. Colors represent homogeneity and heterogeneity. FedMA is short for FedMobileAgent (Wang et al., 2025a).

Algorithm	Shopping	Traveling	Office	Lives	Entertain.	Avg.	Algorithm	Shopping	Traveling	Office	Lives	Entertain.	Avg.
Zero-Shot	26.61	25.33	27.05	24.41	23.81	25.46	Central	57.26	58.67	51.64	55.12	60.95	56.90
Homo.	**App Random**						**Hetero.**	**App Skew**					
Local 0	43.55	45.78	35.25	43.31	38.10	41.96	Local 0	44.35	40.44	**48.36**	29.92	35.24	39.83
FedAvg	50.81	51.56	47.54	44.09	48.57	48.93	FedAvg	50.81	53.78	45.90	33.86	53.33	48.22
FedProx	49.19	49.78	46.72	41.73	49.52	47.65	FedProx	51.61	**54.22**	47.54	**38.58**	**54.29**	**49.79**
FedAvgM	50.00	**54.67**	46.72	44.09	**52.38**	**50.21**	FedAvgM	**52.42**	52.00	45.90	37.01	**54.29**	48.65
FedYogi	**53.23**	51.56	**49.18**	**46.46**	48.57	50.07	FedYogi	50.00	52.89	45.08	35.43	49.52	47.37

Table 11: Supplementary experiments on the two other subsets of Category-Level Dataset: App Random and App Skew. Compared to the results in Category Skew, App Skew produces more severe heterogeneity. All FL algorithms demonstrate diverse performances on the two subsets with FedAvgM generally achieves the best results.

three subsets, we further confirm the presence of app-level heterogeneity, as a clear performance drop occurs when the model transitions to more heterogeneous scenarios. (2) Additionally, we observe that in heterogeneous settings, apps with a long-tailed distribution and fewer episodes experience a more significant performance decline compared to apps with more abundant data, such as Amazon and Ebay. (3) The performance of the 0-th local client on Amazon in the ScaleApp Skew subset aligns with expectations, as the client has 300 training episodes of Amazon data. However, it also performs exceptionally well on Flipkart, even though it has not encountered any Flipkart data during training. This remarkable performance suggests that there may be shared patterns between Amazon and Flipkart, contributing to the unexpected yet correlated success.

B.2 Supplementary Experiments on Category-Level and App-Level Datasets

Setups. The experimental settings as the same with the experiments in Section 4.4 and 4.5. Due to page limits, we present more results with different baselines and other subsets in this section for reference. We use "FedMA" to denote FedMobileAgent for spacing. The colors represent homogeneity and heterogeneity.

Results. We draw the following conclusions: (1) As shown in Table 10, we further substantiate that training mobile agents using federated learning yields promising enhancements, as all baselines exhibit remarkable progress compared to local training. (2) From Tables 10 and 5, global aggregation methods based on optimization (FedAdam, FedAdagrad, and FedYogi) consistently manifest subpar performance on the Category Half-Skew subset, but demonstrate exceptional results on the other subsets. This performance discrepancy remains challenging to explain. (3) By comparing the FL results on the two subsets, Category Skew and App Skew, in Tables 11 and 5, we conclude that FL algorithms generally underperform on the App Skew subset, which indicates that app name heterogeneity is more fundamental and severe than app category heterogeneity. (4) As shown in Tables 12 and 6, the eight baselines exhibit diverse performance across different heterogeneous scenarios. FedMobileAgent performs averagely, as it is not specifically designed to handle this type of heterogeneity, and it degrades to standard FedAvg when the app distribution becomes extremely skewed. (5) As reaffirmed, no current FL algorithm effec-

Algorithm	Amazon	Clock	Ebay	Flipkart	Gmail	**Avg.**	**Algorithm**	Amazon	Clock	Ebay	Flipkart	Gmail	**Avg.**
Zero-Shot	29.75	32.38	28.33	30.00	28.12	29.62	Central	54.55	64.76	58.33	61.00	51.56	57.67
Homo.	**App IID**						**Hetero.**	**App Skew**					
Local 0	44.63	49.52	41.67	50.00	33.59	43.38	Local 0	**56.20**	36.19	42.50	44.00	21.09	39.72
FedAdagrad	56.20	**54.29**	54.17	**58.00**	**50.00**	**54.36**	FedAdagrad	45.45	**54.29**	50.83	**55.00**	**46.88**	**50.17**
SCAFFOLD	56.20	**54.29**	**55.00**	53.00	47.66	53.14	SCAFFOLD	48.76	**54.29**	**52.50**	52.00	44.53	**50.17**
FedMA	**58.68**	53.33	53.33	55.00	48.44	53.66	FedMA	47.93	53.33	48.33	54.00	41.41	48.61
Hetero.	**App Half-Skew**						**Hetero.**	**App Non-Uniform**					
Local 0	52.89	**57.14**	**45.00**	40.00	36.72	46.17	Local 0	39.67	**58.10**	38.33	48.00	**46.09**	45.64
FedAdagrad	54.55	54.29	43.33	**55.00**	**42.19**	49.48	FedAdagrad	**56.20**	54.29	**46.67**	50.00	40.62	49.30
SCAFFOLD	54.55	53.33	43.33	54.00	40.62	48.78	SCAFFOLD	55.37	53.33	45.83	50.00	40.62	48.78
FedMA	**55.37**	53.33	45.83	55.00	41.41	**49.83**	FedMA	55.37	55.24	45.83	**52.00**	41.41	**49.65**

Table 12: Supplementary Experiments on the App-Level Dataset. We provide additional evaluation results with four other baselines. The total eight baselines yield diverse performance in different heterogeneous scenarios.

tively addresses the new heterogeneity introduced by federated mobile agents, as all FL algorithms experience a substantial decline from IID to non-IID app distributions, which highlights the need for further advancements in this area.

B.3 Comparison of Base Models

Setups. Built upon ms-swift, FedMABench supports over ten base VLMs and has the potential to accommodate more in the future. We select five models as representatives, encompassing both open-ended and closed-ended models from three distinct model families. Since closed-ended models cannot be fine-tuned, we provide zero-shot results for them. For open-ended models, we fine-tune them on the App IID subset of the App-Level Dataset as a representative case.

Results. As shown in Table 13, we draw the following conclusions: (1) Training on different models yields diverse performance results. (2) Overall, the performance of open-ended models shows a strong positive correlation with their model size. (3) Through federated training on distributed data, even smaller VLMs like Qwen2-VL-2B-Instruct can achieve performance on par with SOTA closed-ended models such as GPT-4o.

B.4 Ablation on Dataset Size

Setups. We conduct experiments on the Basic-AC Dataset with incrementally increasing data sizes to investigate the impact of dataset size on performance, and to examine whether scaling laws hold in the context of federated learning for mobile agent training. To control experimental conditions, we fix the number of clients at 10 and evaluate the mobile agents after 10 communication rounds. Notably, in the FedAvg implementation, 30% of participating clients are randomly sampled per round, leading to a smaller number of sample iterations compared to centralized training.

Base Model	Amazon	Clock	Ebay	Flipkart	Gmail	**Avg.**
Algorithm	**Zero-Shot**					
GPT-4o	40.50	48.57	43.33	45.00	38.28	42.86
GPT-4o-mini	26.45	33.33	30.83	30.00	35.16	31.18
Algorithm	**FedAvg**					
Qwen2-VL-2B	47.11	46.67	35.83	38.00	39.84	41.46
Qwen2-VL-7B	57.02	53.33	52.50	55.00	46.88	52.79
InternVL2-1B	28.93	40.00	27.50	28.00	35.16	31.88
InternVL2-2B	34.71	41.90	30.00	28.00	32.03	33.28

Table 13: Comparison of different base models on the App IID subset. We choose five models as representatives including both open-ended and closed-ended models.

Results. As shown in Table 14, we draw the following conclusions: (1) Performance improvements exhibit a strong positive correlation with dataset scale across all training paradigms, validating the effectiveness of federated learning for scalable mobile agent training. Specifically, FedAvg demonstrates incremental gains from 31.18% to 53.54% as data availability increases. (2) FedAvg shows diminishing returns as the data size reaches a certain threshold, still leaving a gap relative to centralized training. Enhancing the performance of federated trained mobile agents necessitates further efforts into this area.

B.5 Ablation on Clients Number

Setups. We investigate federated learning dynamics under varying client number while maintaining a fixed budget of 100 episodes per client. Mobile agents are evaluated after 100 training rounds with a controlled participation scheme: each round activates 10% of available clients.

Algorithm	200	500	1000	3000	5000	7000
Zero-Shot			27.24			
Central	43.94	42.36	55.59	56.38	59.69	62.05
Local 0	17.80	28.35	37.64	44.25	47.40	52.44
FedAvg	31.18	36.54	43.78	50.39	51.50	53.54

Table 14: Experiments on dataset sizes. Performance improvements exhibit strong positive correlation with dataset scale for all training paradigms.

Client Number	10	30	50	70
Client Sample	1	3	5	7
FedAvg	51.81	56.06	57.17	57.48

Table 15: Experiments with different client numbers. Each client is allocated 100 episodes. As more clients are involved, the dataset scale increases. Performance improvements show a positive correlation with the number of clients, consistent with the results in Table 14.

Results. As shown in Table 15, we conclude that: (1) As reiterated, model performance demonstrates strong positive correlation with client population size, validating federated learning's effectiveness for scalable distributed training. (2) A particularly significant performance leap (51.81% $\rightarrow$ 56.06% step accuracy) occurs when scaling from 10 to 30 clients, suggesting critical mass benefits in collaborative learning.

B.6 Ablation on Clients Participation

Setups. We analyze the impact of client participation rates while keeping the total client population constant and maintaining a fixed global data volume. Specifically, we use the subset of Basic-AC with 3,000 episodes, partitioned across 30 clients. The system is evaluated after 100 training rounds with varying numbers of clients sampled per round, ranging from 1 to 30 participants.

Results. As shown in Table 16, we draw the following conclusions: (1) Cross-referencing with Table 15 reveals an emergent pattern: under equivalent total data budgets, increasing client participation enhances model performance. This suggests distributed learning benefits stem not merely from data accumulation, but crucially from diversified experiential sampling across heterogeneous clients. (2) Moderate participation rates, with 3 clients sampled per round, achieve performance comparable to maximum participation. This phenomenon can be attributed to the fact that as the number of participating clients increases, heterogeneity also rises,

Client Number	30	30	30	30	30	30
Client Sample	1	3	5	10	15	30
FedAvg	41.10	45.35	44.72	44.09	43.94	45.67

Table 16: Experiments with varying client participation rates, with the dataset and its partition kept constant for controlled comparison. A moderate number of clients per round achieves comparable performance to full participation.

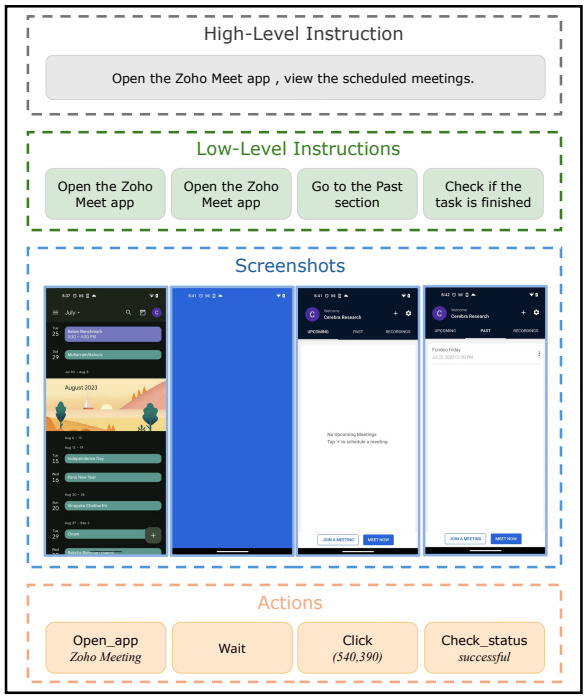


Figure 6: A data episode example for training mobile GUI agents.

which may degrade overall performance despite the higher training cost.

C Data & Experiment Details

C.1 Dataset Details

We provide detailed descriptions of our datasets and the data collection process in this section, including examples and statistics.

Data Episode Example. To provide a clearer understanding of the structure of our dataset and the composition of a data episode, we present a sample as an example in this section. As shown in Figure 6, each episode consists of: (1) A high-level instruction, which is a natural language sentence describing the task to be accomplished; (2) A sequence of low-level instructions, detailing the fine-grained tasks required for the current screenshot; (3) A series of screenshots taken from the start to the end of the task; and (4) A corresponding list of

actions, matching the number of screenshots, indicating what the user does to progress to the next screenshot. All actions belong to an action space containing 7-9 options. We adopt the action spaces defined in (Rawles et al., 2023; Li et al., 2024b; Wang et al., 2025a).

Dual-Strategy App Name Extraction. For each episode from the original dataset of Android Control, we implement a dual-strategy approach for application name extraction based on the "open app" action and regular expression matching.

As demonstrated in the following code snippet, if the actions include the the "open app" action, the application name is directly retrieved from the dedicated `app_name` field, followed by sanitized string processing.

```
for episode in all_episodes:
    // Load the task information for the current
        episode
    data = load("task_info.json")
    // Check if 'open_app' in the actions of this
        episode
    if "open_app" in data["action_type"]:
        // Direct extraction with sanitization
        app_name = data["app_name"].replace("\ufeff"
            , "")
    else:
        // Use semantic pattern matching

        pattern = re.compile(
            r'\bthe\s+(\w+(?:\s+\w+)?)\s+app\b',
            re.IGNORECASE
        )
        match = pattern.search(data["goal"])
        if match exists:
            app_name = match.group(1)
        else:
            // Skip episodes where extraction fails
            continue
```

Otherwise, if the actions of this episode do not contain the "open app" action, which indicates that explicit application identifiers are absent, we attempt to extract potential app name from the `goal` field (i.e., instructions) . This is achieved through a regular expression designed to identify the phrase `"the [app] app"` using semantic pattern matching.

Episodes failing both extraction strategies were systematically excluded to ensure data validity. This dual-strategy filtering process ultimately yielded 8,400 qualified episodes containing unambiguous application identifiers, forming the core dataset for subsequent construction and analysis.

Dataset Statistics. In this part, we provide a detailed enumeration of the specific apps included in each dataset, along with the exact number of instances for each app.

(1) Basic-AC Dataset: The Basic-AC Dataset encompasses comprehensive categories and apps. Detailed statistical information can be found in Table 19. (2) Category-Level Dataset: The Basic Dataset comprises a total of 52 apps that are organized into several categories. In the shopping category, there are 10 apps: Amazon, eBay, Flipkart, Adidas, Nike, Decathlon, Etsy, Puma, Temu, and Snapdeal, with each app contributing 20 instances for a total of 200. The travelling category includes 10 apps, namely Google Maps, Expedia, Omio, Booking.com, Citymapper, Trainline, Kayak, Cruisemapper, MakeMyTrip, and Agoda, where each app again provides 20 instances to reach a sum of 200. The office category follows the same pattern with 10 apps: Gmail, Clock, Google Drive, Google Docs, Calendar, Google Keep, Contacts, Reminder, Recorder, and Voice Recorder, each adding 20 data points for a total of 200. The lives category also consists of 10 apps: Kitchen Stories, Home Workout, Sidechef, Yummly, Blossom, Plantum, Simple Habit, Leafsnap, Medito, and Insight Timer, each contributing 20 instances to make up another 200. In contrast, the entertainment category is slightly different, comprising 12 apps. Eight of these apps, which are YouTube, Vimeo, Artsy, Sketchbook, Messenger, Pinterest, Flipboard, and SoundCloud, each provide 20 instances, while the remaining four apps, namely Snapchat, SmartNews, The Hindu, and CNN, contribute 10 instances each, together totaling 200.

Basic-AC Specifics. We construct 14 subsets in Basic-AC, a detailed description of which is provided in Table 17. The table specifies three key parameters for each subset: number of participating clients, total episodes, and total steps. Subsets 1-9 represent cross-category aggregations with varying scales, while subsets 10-14 correspond to category-specific partitions.

C.2 Training Details

General Parameters. Our implementation leverages the Swift library (Zhao et al., 2024) with parameter-efficient fine-tuning. The LoRA configuration employs a rank of 8 with an alpha scaling factor of 32, incorporating dropout regularization of 0.05 to prevent overfitting. We set the maximum sequence length to 4,096. We set the batch size to 1 and the gradient accumulation step to 4. The learning rate is kept fixed at 5e-5.

Hardware Configuration. The training is conducted on two NVIDIA GeForce RTX 3090 GPUs utilizing CUDA version 12.4. Under this hard-

Dataset	Subset	Category	N. Client	N. Episode	N. Step
Basic-AC	c10n200	all	10	200	7454
	c10n500	All	10	500	20198
	c10n1000	All	10	1000	40112
	c10n3000	All	10	3000	120512
	c10n5000	All	10	5000	201434
	c10n7000	All	10	7000	282332
	c30n3000	All	30	3000	120512
	c50n5000	All	50	5000	201434
	c70n7000	All	70	7000	282332
	Shopping	Shopping	10	2252	79292
	Travelling	Travelling	10	788	47918
	Office	Office	10	1974	76910
	Lives	Lives	10	1136	46070
	Entertainment	Entertainment	10	850	32150

Table 17: Composition details of the 14 subsets in the Basic-AC Dataset. 'N.' denotes the number of instances, and 'Category' refers to the covered categories within each subset.

Configuration	**Value**
Model Architecture	`Qwen2VLForConditionalGeneration`
Model Type	`qwen2_vl`
Torch Dtype	`bfloat16`
Tokenizer	`Qwen2TokenizerFast`
Tokenization Strategy	Greedy search

Table 18: Model and tokenizer configurations.

ware configuration, the training process achieves a throughput of approximately 2 minutes per training round per client when processing 10 episodes.

Federated Algorithms. The framework implements adaptive hyperparameter defaults for various federated algorithms: FedYogi (Reddi et al., 2020) employs momentum factors ($\beta_1 = 0.9, \beta_2 = 0.999$) with learning rate $\eta = 10^{-3}$ and stabilization constant $\tau = 10^{-6}$. FedAvgM (Hsu et al., 2019) uses $0.9/0.1$ ratio for historical/current model interpolation. FedProx (Li et al., 2020) applies proximal regularization with $\mu = 0.2$ through $||w - w^t||^2$ penalty terms. SCAFFOLD (Karimireddy et al., 2020) configurations maintain server learning rate $\eta_s = 1.0$ with client momentum compensation, while FedAdam and FedAdagrad (Reddi et al., 2020) share base parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) with adaptive learning rate scaling. All algorithms expose tunable coefficients through the framework's unified parameter interface.

This section outlines the specific configurations used for our vision-language model (VLM) experiments, ensuring reproducibility and clarity.

Model and Tokenization Our primary experiments are conducted using the Qwen2-VL model. The specific configuration is detailed in Table 18.

C.3 Prompt Format

Following FedMobileAgent (Wang et al., 2025a), we designed a structured prompt format (Figure 7) that provides the necessary context for decision-making. The prompt template includes the high-level goal, the visual context (i.e., the screen screenshot), and a list of available actions. This structured approach allows the model to ground its decision-making process in both the overall objective and the immediate, actionable elements visible on the screen.

Prompt 1: High-Level Training Prompt

You are a smartphone assistant tasked with helping users complete actions by interacting with apps. I will provide you with one screenshot, representing the UI state before an operation is performed.

For the screenshot, you need to identify and output a specific action required to complete the **User Instruction**.

User Instruction
{ high-level instruction $\mathcal{T}^{high}$ }

Response Requirements
For each screenshot, you need to decide just one action on the current screenshot.
You must choose one of the actions below:

1. **Click on button with the text "⟨UI element text⟩"**
If the button has no text related, output "Click on the location ⟨x,y⟩".
2. **Long press on button with the text "⟨UI element text⟩"**
If the button has no text related, output "Long press on the location ⟨x,y⟩".
3. **Type text: "⟨ input text ⟩"**
Type the ⟨**input text**⟩ in the current input field or search bar.
4. **Scroll ⟨direction⟩**
Scroll the UI element by ⟨direction⟩.
If the current UI includes scrollers but lacks the necessary elements for the task, try scrolling down to reveal elements below or scrolling up to uncover elements above. Similarly, scroll right to reveal elements on the right or scroll left to uncover elements on the left.
5. **Return to the home page**
Return to the home page. If you want to exit an app, use this action.
6. **Go back to the previous page**
Go back to the previous page. If you need to return to the previous step or undo an action, use this action to navigate back.
7. **Open App: ⟨app name⟩**
If you wish to open an app, use this action to open ⟨app name⟩.
8. **Wait for response**
Pause for a moment to allow any background processes to complete or for elements to load before proceeding with the next action.
9. **Check status: ⟨successful/infeasible⟩**
If you think all the requirements of the user's instruction have been completed successfully and no further operation is required, you can choose "successful" to terminate the operation process. If the task cannot be completed due to missing elements or any other issue, you can use "infeasible" to indicate that the action cannot be performed.

Your Response

Figure 7: Prompt template for the high-level training of federated mobile agents within FedMABench.

App	Num	App	Num	App	Num	App	Num	App	Num	App	Num
Shopping											
amazon	302	ebay	225	flipkart	151	adidas	83	decathlon	82	etsy	76
nike	64	temu	64	puma	59	shopsy	53	snapdeal	52	ikea	47
shopclues	43	ubuy	40	banggood	38	industrybuy.	37	myntra	37	tata cliq	37
zara	37	jiomart	33	dhgate	32	blinkit	30	moglix	30	bigbasket	26
asos	25	joom	20	tata neu	20	dmart ready	19	ajio	17	hardware sh.	17
nnnow	17	pepperfry	17	edmunds	15	houzz	13	footshop	12	hamleys	12
limeroad	12	rapidbox	12	mywarehouse	11	nykaaman	11	toys 'r' us	11	yalla toys	11
coolblue	10	freshtohome	10	lazada	10	mega hardwa.	10	shoppers st.	10	barakat	9
cartrade	8	furlenco	8	nykaafashion	8	autoscout24	7	lovelocal	7	cars24	6
carwale	6	nykaa	6	olx india	6	spinny	6	toyspoint	6	woodenstreet	6
dookanti	5	uniqlo	5	urbanic	5	albertsons	4	hardware sh.	4	jd	4
louis vuitt.	4	max fashion	4	nature's ba.	4	pdffiller	4	sports dire.	4	true value	4
urban outfi.	4	zalando	4	1800 flowers	3	abercrombie	3	adani one	3	bechdo	3
bewakoof	3	carguru	3	dunzo	3	globalsourc.	3	homzmart	3	igp	3
khelmart	3	nykaa fashi.	3	peter engla.	3	pizza max	3	reliance di.	3	shoptime	3
spencers	3	sportsuncle	3	westside	3	cardekho	2	colourpop c.	2	coop	2
ferns n pet.	2	flower aura	2	funeasylearn	2	furniture o.	2	instashop	2	jaquar	2
louis phili.	2	love local	2	m&s india	2	magzter	2	massimo dut.	2	milkbasket	2
moira	2	namshi	2	noon	2	p louise co.	2	pantaloons	2	pepper	2
cosme.											
redbubble	2	royal	2	safeway	2	sports bazar	2	sportsdirect	2	sportspar	2
super note	2	top-most ha.	2	topmost har.	2	weather rad.	2	yoox	2	zappo	2
zappo	2	acme	1	apkpure	1	app market	1	character c.	1	dubizzle	1
brands											
ebay app	1	electronics.	1	estee lauder	1	farfetch	1	fernsnpetals	1	goat	1
gostor	1	ikea app	1	industry ub.	1	insaraf - s.	1	iplan.ai	1	jd sports	1
jollee	1	kicks crew	1	luxuryestate	1	massimo du.	1	mikbasket	1	mytrip	1
nnnnow	1	nobroker	1	same temu	1	samsung shop	1	sanitary ba.	1	second cale.	1
sun & sand .	1	tesco	1	thriftbooks	1	toys shoppi.	1	tradet mark.	1	vijetha live	1
winni	1	woodland	1	woodlands	1	zomato	1				
Travelling											
google maps	111	expedia	55	omio	47	booking.com	46	kayak	40	citymapper	37
cruisemapper	29	makemytrip	27	trainline	27	airbnb	25	skyscanner	25	agoda	23
wanderu	21	alltrails	20	rail planner	20	guardian	15	moovit	14	traillink	13
hopper	11	momondo	11	rome2rio	11	trip.com	11	yatra	10	cruisedeals	9
goibibo	9	amtrak	8	easemytrip	8	ixigo	8	klook	8	flixbus	7
foursquare	7	talabat	7	time zone c.	6	trainpal	6	schedule pl.	5	cleartrip	4
kiwi.com	4	shipatlas	4	traveloka	4	getby	3	hiking proj.	3	hotels.com	3
immobiliare	3	lambus	3	maxmilhas	3	prestigia	3	rail europe	3	riyadh bus	3
travel life	3	wego flight.	3	bookaway	2	eurostar	2	gotogate	2	greyhound	2
hhr train	2	hiiker	2	klm	2	lner	2	orbitz	2	passporter	2
sbb mobile	2	sncf connect	2	sygic travel	2	trovit	2	cheapflights	1	egy train w.	1
farefirst	1	maps go	1	mytrip	1	roadtrippers	1	sncb intern.	1	thalys	1
trivago	1										
Office											
gmail	189	clock	158	google drive	127	reminder	101	calendar	72	contacts	69
google keep	65	google docs	52	recorder	48	voice recor.	47	google slid.	45	ticktick	37
khan aca.	36	skype	36	chat	33	powerpoint	33	settings	31	files by go.	30
dropbox	28	officesuite	22	todoist	22	phonebook	21	polaris off.	20	clockbuddy	19
all currenc.	18	memrise	17	microsoft w.	17	onedrive	17	outlook	17	smart recor.	16
google news	15	taskito	15	tasks	15	jotform	14	myrecorder	14	any.do	13
readera	13	translate	13	currency pl.	12	easy voice .	12	migros	12	merriam.	11
to do remin.	11	to do list	10	formsapp	9	notein	9	presentatio.	9	colornote	8
coursera	8	easy dialer	8	easy notes	8	easy timezo.	8	xodo	8	zoho meeting	8
calculator	7	note	7	spck editor	7	webex	7	alarmy	6	dictionary	6
duocards	6	habitica	6	meet	6	microsoft p.	6	mondly lang.	6	moon+ reader	6
pdf reader .	6	whiteboard	6	notebook	5	pcloud	5	schedule pl.	5	simple calc.	5
timezone co.	5	alarm clock.	4	calendar pl.	4	code editor	4	digital ala.	4	easynotes	4
forms app	4	plantapp	4	savvy time	4	sheets	4	sublime text	4	tododo	4
vocab.com	4	webex meet	4	winzip	4	word office	4	zarchiver	4	alarm clock.	3
clevnote	3	contact	3	cursa	3	cx file exp.	3	deftpdf	3	digical	3

Continued on next page.

App	Num	App	Num	App	Num	App	Num	App	Num	App	Num
doodle	3	forms.app	3	math tests	3	my money	3	pdfelement	3	pull&bear	3
spendee	3	udemy	3	voice recor.	3	weather xl	3	calcu	2	calendar pro	2
carrot	2	ereader pre.	2	flipsnack	2	funeasylearn	2	giant stopw.	2	google tasks	2
letter temp.	2	math learni.	2	microsoft 3.	2	multi calcu.	2	munimobile	2	mycurrency	2
papago	2	power point	2	pro 7-zip	2	quip	2	simple cont.	2	smartcal	2
super note	2	timezones	2	unit conver.	2	voice recor.	2	world clock	2	xe converter	2
zoho show	2	7z	1	blaze wordp.	1	bookscape	1	calculator .	1	currencycon.	1
docx - all .	1	drawing pad	1	everand ebo.	1	exchange ra.	1	focus to-do	1	g-forms	1
internal fi.	1	iplan.ai	1	lists	1	math learni.	1	maths test	1	monefy	1
office: pre.	1	oppia	1	pdf extra	1	radio u.s.	1	rar	1	setting	1
simple clock	1	smartify	1	step tracke.	1	telegram	1	upgrad	1	webcode	1
Lives											
kitchen sto.	91	home work.	51	fit	50	sidechef	44	yummly	44	insight tim.	38
leafsnap	30	redfin	30	blossom	28	weather	27	plantum	26	opentable	25
google fit	24	simple habit	24	plantin	23	strava	20	dmart ready	19	fitai	19
fitbit	19	meditopia	19	idanim	18	artier	17	medito	17	pepperfry	17
calm	15	jefit	15	grubhub	13	mindfulness	13	tasty	13	cookpad	12
deliveroo	12	evolve	12	migros	12	trovit homes	12	breethe	11	lifestyle	11
photos	11	supercook	11	all recipes	10	bigoven	10	lunch recip.	10	notes	10
doordash	9	home centre	9	99acres	8	all recipes.	8	rentberry	8	urban ladder	8
fitpro	7	heartfulness	7	phases of t.	7	talabat	7	bbc news	6	budgetbytes	6
daff moon	6	moon	6	pizza hut	6	withings	6	baby tracker	5	balance	5
gym workout	5	martinoz pi.	5	mi fitness	5	my moon pha.	5	plant ident.	5	realtor.com	5
runkeeper	5	housing	4	moonx	4	serenity	4	flo	3	headspace	3
healthifyme	3	ovia pregna.	3	planta	3	pregnancy	3	smiling mind	3	trulia	3
carrot	2	hatch baby	2	home garden	2	immoscout24	2	moonly	2	plantora	2
property fi.	2	recime	2	vivareal	2	what to exp.	2	babycenter	1	cult.fit	1
freshto home	1	good food	1	immobiliare.	1	indian reci.	1	luxuryestate	1	mojopizza	1
my workout .	1	nobroker	1	workout pla.	1						
Entertainment											
youtube	95	vimeo	64	gallery	36	artsy	35	messenger	32	pinterest	31
spotify	27	sketchbook	26	soundcloud	23	flipboard	20	snapchat	17	the weather.	16
cnn	15	google news	15	guardian	15	arts & cult.	13	tunein radio	13	wynk music	12
audiomack	11	deviantart	11	nytimes	11	photos	11	pocketbook	11	show	11
smartnews	11	youtube mus.	11	coolblue	10	mytuner rad.	10	the hindu	10	sgraffito	9
skyview free	9	behance	8	reuters	7	sketchar	7	bbc news	6	moon+ reader	6
radio garden	6	time zone c.	6	toi	6	washington .	6	webnovel	6	whiteboard	6
color	5	euronews	5	gaana	5	hindu	5	kobo books	5	mi fitness	5
thefork	5	wattpad	5	cafeyn	4	cna	4	cnn news	4	dolby on	4
domino's	4	hindu news	4	hungama	4	usa today	4	anghami	3	dailymotion	3
fox news	3	headspace	3	mojarto	3	nbc news	3	peggy	3	rtistiq	3
toi news	3	daily art	2	dailyart	2	hiiker	2	magzter	2	msn weather	2
paint	2	radio	2	radio fm	2	readly	2	readwhere m.	2	sky tracker	2
startracker	2	zinio magaz.	2	app market	1	artly	1	bbdaily	1	deccan hera.	1
expert pape.	1	hipaint	1	messages	1	newyork tim.	1	radio u.s.	1	readly maga.	1
sky view	1	skyview	1	smartify	1	winni	1				

Table 19: Application categorization and statistics for the Basic-AC Dataset. Due to the limited table width, app names that are too long will be truncated, with the truncated portion replaced by a dot(.).