

MobileWorldBench: Towards Semantic World Modeling For Mobile Agents

Shufan Li ¹, Konstantinos Kallidromitis ², Akash Gokul ³, Yusuke Kato ²
Kazuki Kozuka ², Aditya Grover ¹

¹UCLA ²Panasonic ¹NVIDIA

Correspondence: jacklishufan@cs.ucla.edu

Abstract

World models have shown great utility in improving the task performance of embodied agents. While prior work largely focuses on pixel-space world models, these approaches face practical limitations in mobile agent settings, where predicting complex visual GUI elements in future states is often difficult. In this work, we explore an alternative formulation of world modeling for mobile agents, where state transitions are described in natural language rather than predicting screenshot pixels. First, we introduce MobileWorldBench, a benchmark that evaluates the ability of vision-language models (VLMs) to function as world models for mobile agents. Second, we release MobileWorld, a large-scale dataset consisting of 1.4M samples, that improves the world modeling capabilities of VLMs. Finally, we propose a novel framework that integrates VLM world models into the planning framework of mobile agents, demonstrating that semantic world models can directly benefit mobile agents by improving task success rates. Code and data is available at <https://github.com/jacklishufan/MobileWorld>.

1 Introduction

World models are capable of predicting future states of a system given current observations, making them particularly useful in a wide range of applications such as robotics, physical simulations, video games, and autonomous driving (Mazzaglia et al., 2024; Yue et al., 2025; Gu, 2025; Guo et al., 2025; Zheng et al., 2025). The most common approach for world modeling is action-conditioned causal video modeling (Chen et al., 2025c; Gu, 2025; Bruce et al., 2024), where a neural network is employed to predict pixels in future video frames based on current and past frames, and action inputs. The predictions of these video world models can be naturally integrated into model-based policies, which have demonstrated strong utility both in real-world tasks and in simulated environments (Ren

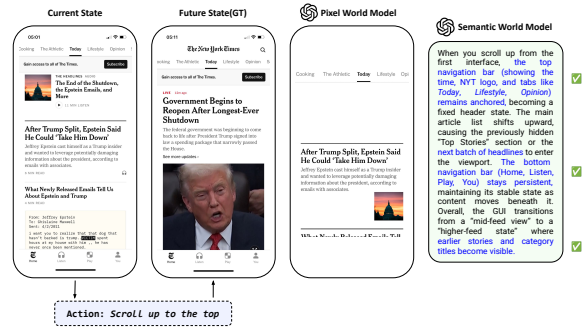


Figure 1: **Advantages of Semantic World Modeling.** Pixel-space world modeling is particularly challenging as the model needs to identify changes, come up with the correct app content, and render them accurately. By contrast, Semantic World Modeling only focuses on abstracting relevant changes in GUI semantics, while being useful for decision-making. In the example shown, the frontier visual generative model GPT-Image struggles to accurately render GUI states, while frontier VLMs (GPT-4o (Hurst et al., 2024)) can accurately describe the expected GUI changes in text.

et al., 2023; Chen et al., 2025c).

Despite their success in many domains, world models for mobile agents remain under-explored. A key challenge in applying naive next-frame prediction to mobile world modeling lies in the intrinsic complexity of the pixel-level prediction task. Let’s use a news app, *e.g.* the New York Times app, as an example (Fig 1). Suppose a user is browsing a list of news articles, and will perform the action “*scroll to the top*”. To predict the next state, a pixel-space world model needs to: 1) Understand the high-level semantics of the action, namely that the scroll action will likely result in additional articles showing up, while keeping the majority of the UI layout unchanged (*e.g.* bottom navigation bar, top tabs, subscribe button), 2) Predict the visual layout of the future page: *e.g.* Does the new app present thumbnails and abstracts? Or does it only show headlines?, 3) Predict the exact content, which in-

volves generating plausible news, 4) Render all texts and icons based on the predicted content and layout.

Even if a world model masters all four capabilities, it remains highly unlikely that the predicted next state will be visually similar to the actual next state, since it is improbable that the predicted news is the same as the real news at the given time. Similarly, it is also hard to predict if new articles include thumbnails, short summaries, or are just titles. In general, 2), 3), and 4) are exceptionally hard tasks that are challenging for state-of-the-art visual generative models. Moreover, these details are typically not essential for model-based policies. For instance, if an agent wants to perform a different action such as “*subscribe*” or “*navigate to view sports news*”, rendering the exact content of new articles that will show up is unnecessary.

Rather than relying on pixel-level predictions, we hypothesize that mobile agents can achieve far more efficient and generalizable world modeling by representing state changes as structured, semantically meaningful textual representations. Notably, traces of this idea appear in the chain-of-thought behavior of large language models (LLMs), which often verbalize predictions about future states in their reasoning. Our contribution is to formalize this phenomenon as an explicit modeling framework, treating textual state descriptions as world models. In doing so, we transform what was previously a by-product of reasoning into a principled mechanism for planning and control. Concretely, this work introduces three key contributions to facilitate the paradigm of semantic world modeling for mobile agents:

First, we propose MobileWorldBench, a comprehensive benchmark that explicitly evaluates VLMs’ world modeling capabilities. Unlike existing GUI understanding and grounding tasks, which focus on interpreting elements on the current screen, MobileWorldBench evaluates VLMs’ ability to predict future states from the current screen and a given user action. Notably, MobileWorldBench involves two tasks: Next-State-Generation and Next-State-QA. In the Next-State-Generation task, the model produces free-form text describing predicted state transitions, which are then evaluated by a VLM judge that compares the description with the ground-truth screenshots of the next state. In Next-State-QA, the model answers a series of yes-no questions about future states, and its performance is quantified using an accuracy metric that directly measures world

modeling ability.

Second, to facilitate the training of semantic world models for mobile agents, we curate MobileWorld, a large-scale world modeling dataset consisting of the triplets: *current state*, *user action*, and *future state*. Future states are represented in three forms: pixels of screenshots, QA pairs, and natural language descriptions of state transitions. We construct MobileWorld by sourcing existing trajectory data and leveraging advanced VLMs to generate QA pairs and annotate state transitions using ground-truth next states.

Finally, we analyze the effectiveness of semantic world models by finetuning an open-source VLM using MobileWorld. Experimental results show that mobile agents that use semantic world models perform better on the AndroidWorld (Rawles et al., 2023) benchmark (+7.4% increase in success rate).

2 Related Works

2.1 World Modeling

Prior work on world modeling has largely focused on predicting future observations either at the pixel level or in latent space. Pixel-based approaches include GUI simulators and video generation systems such as NeuralOS (Rivard et al., 2025), MineWorld (Guo et al., 2025), GEM (Hassan et al., 2025), Cosmos (Gu, 2025), Genie (Bruce et al., 2024), and ViMo (Luo et al., 2025). Latent-space approaches such as V-JEPA (Bardes et al., 2024; Assran et al., 2025) improve efficiency by predicting compact representations instead of reconstructing pixels. More recent work explores semantic world modeling through language-based abstractions, including SWM (Berg et al., 2025) and WMA (Chae et al., 2024), which formulate future-state prediction as textual reasoning. Other studies further demonstrate that language can serve as an effective abstraction for planning and decision-making (Jiang et al., 2019; Chen et al., 2025b,a).

Unlike prior work that relies on pixel reconstruction, we focus directly on semantic future-state prediction. Compared with prior semantic world modeling works on web navigation, our work uniquely address the semantic world modeling for mobile GUI agents, which has more diverse tasks and requiring additional long-horizontal planning capabilities because they involves many persistent states like file systems and system setting.

2.2 Mobile Agents

Mobile agents have evolved from systems relying on structured UI representations (Wen et al., 2023, 2024), to multimodal agents that operate directly on screenshots (Zhang et al., 2025). Recent works improve decision-making using memory, decomposition, or reinforcement learning (Wang et al., 2024; Li et al., 2024b; Ye et al., 2025; Bai et al., 2024; Liu et al., 2024), while others enhance GUI grounding and perception through specialized architectures and supervised finetuning (Wang et al., 2025; Hong et al., 2024; Ma et al., 2024; Cheng et al., 2024; Lu et al., 2025). Existing agents typically select actions through UI element grounding or coordinate prediction (Wen et al., 2023; Wang et al., 2024; Li et al., 2024b) and operate in a reactive observation-action loop. In contrast, our work studies predictive modeling of future GUI states to support planning and long-horizon reasoning.

2.3 GUI Datasets & Benchmarks

Existing GUI datasets and benchmarks primarily focus on interface understanding, grounding, or agent execution. Representative datasets include Rico (Deka et al., 2017), Screen2Words (Wang et al., 2021), AITW (Rawles et al., 2023), and GUICourse (Chen et al., 2025d). Popular benchmarks include Mind2Web (Deng et al., 2023), WebArena (Zhou et al., 2023), Mmina (Tian et al., 2025), Spotlight (Li and Li, 2022), Ferret-UI (You et al., 2024), OmniACT (Kapoor et al., 2024), AgentStudio (Zheng et al., 2024), OSWorld (Xie et al., 2024), AndroidWorld (Rawles et al., 2024), and GUI-World (Chen et al., 2024). Unlike these works, which mainly evaluate action selection or understanding of observed states, our work focuses specifically on predicting future GUI states conditioned on user actions.

3 Method

3.1 Semantic World Modeling for Mobile Agents

Given a current GUI state X_t at timestep t and a user action input a_t , classic world modeling $p(X_{t+1}|X_t, a_t)$ directly predicts the pixel representation of the future state X_{t+1} . We break down this process into two steps. Given (X_t, a_t) , we first predict the high-level changes that will occur, such as “the search bar will appear”, or “a new word is typed in the input box”. We represent these changes as a latent variable z_{t+1} . Then we can

render the pixel-level details based on the predicted changes z_{t+1} and previous screen X_t . Formally, we factorize the classic world modeling objective $p(X_{t+1}|X_t, a_t)$ as:

$$\begin{aligned} p(X_{t+1}|X_t, a_t) \\ = \sum_{z_{t+1}} p(X_{t+1}|z_{t+1}, X_t, a_t) p(z_{t+1}|X_t, a_t) \end{aligned} \quad (1)$$

where $p(X_{t+1}|z_{t+1}, X_t, a_t)$ is assumed to be independent of the action a_t and can be written as $p(X_{t+1}|z_{t+1}, X_t)$, *i.e.* once we know the changes that will occur, we can use it to determine the next state without knowing the action. We refer to the distribution $p(z_{t+1} | X_t, a_t)$ as the semantic world model.

In practice, we use natural language descriptions to represent actions. We assume z_{t+1} lies in the hidden representation of a VLM. To extract interpretable information from a latent z , we define two queries on the latent variable. The first query $p(y|z)$ generates a text description y that summarizes the GUI changes encoded in z . Additionally, we also provide a series of statements $\{q^i\}_{i=1,2..N}$ about the semantics of the next state. The second query $p(o^i|q^i, z)$ can be used to obtain the likelihood of q^i being true in the next GUI state. These two queries allow us to evaluate the quality of z and meaningfully utilize them in decision-making. These setups are illustrated in fig. 2 (Left).

To utilize semantic world models for decision-making, we adopt a model-based policy approach shown in fig. 2 (Right). Given current state X_t , a high-level goal G , and action proposals $a^1 \dots a^k$, we first use the world model to predict $z_{t+1}^1 \dots z_{t+1}^k$, and then use a value model to obtain corresponding scores $v_{t+1}^1 \dots v_{t+1}^k$. Finally, we use $\arg \max$ to select the action a_t . In our setup, the action proposal model and value model are implemented using VLMs. For the value model, we first query text descriptions $y \sim p(y|z)$ and pass y alongside a high-level goal G to obtain value scores.

3.2 MobileWorldBench

We can evaluate a semantic world model by measuring how well the latent z_{t+1} captures the high-level semantics of the state transition $X_t \rightarrow X_{t+1}$. This can be assessed in two ways: (1) how well the text $y \sim p(y | z_{t+1})$ describes the observed state transition in pixel space, and (2) how accurately the model estimates the likelihood $p(o_{t+1}^i | q^i, z_{t+1})$

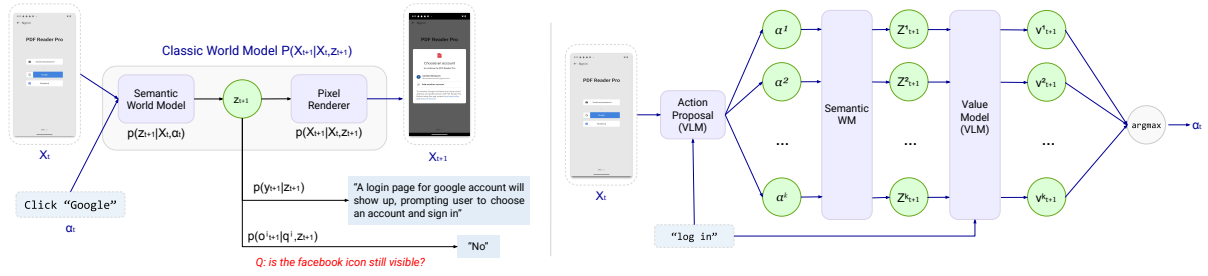


Figure 2: **The Semantic World Model Paradigm.** (Left): we factorize the classic pixel world models into two components. We call the first component the semantic world models. It predicts the latent distribution $p(z_{t+1}|X_t, a_t)$ encoding high-level semantics. z_{t+1} can be queried via $p(y|z)$ to produce text descriptions, or through $p(o^i|q^i, z)$ to produce yes-no answers. (Right). To use semantic world models for decision-making, we employ a model-based policy framework that combines a semantic WM with an action proposal model and value model.

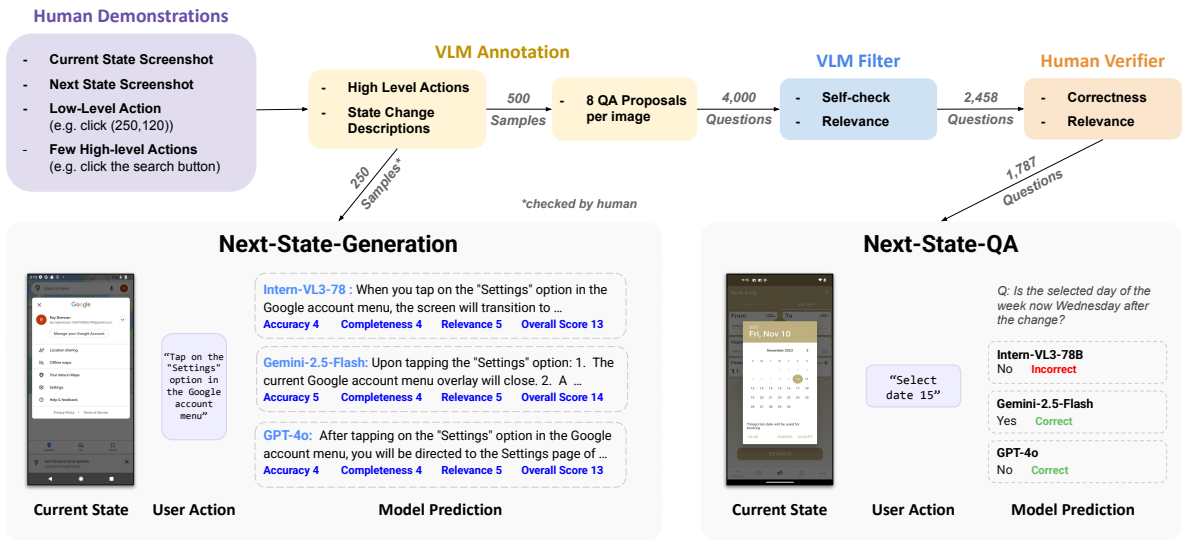


Figure 3: **Data Pipeline for MobileWorldBench.** Our data pipeline consists of 5 steps: 1) Curating raw trajectories, 2) Using VLM to convert low-level actions to high-level actions and annotate state-change descriptions for reference, 3) We generate QA candidates for each state transition sampled, 4) We use VLM to filter these QA pairs through self-check and relevance metrics, 5) For the filtered data, we additionally use human verification to further filter the data based on its correctness and relevance.

that statements q^i hold in the next state. These two aspects are measured using two separate tasks: Next-State-Generation and Next-State-QA.

3.2.1 Task Definitions.

Next-State-Generation tasks prompt a semantic world model to generate text descriptions y_{t+1} describing the expected state changes given inputs (X_t, a_t) . We then use a VLM judge model (GPT-4o) to evaluate these predictions. The judge model is provided with inputs X_t, a_t, y_{t+1} , and ground-truth next state X_{t+1} . It then rates the quality of y_{t+1} based on three key metrics: **Accuracy**. Does the prediction accurately describe the changes in GUI states from X_t to X_{t+1} ? A model is penalized on this metric if it outputs objectively false

statements. **Relevance**. Does the prediction involve changes that are relevant to the action being performed? A model is penalized for correct but irrelevant predictions such as "the system time will advance." **Completeness**. Does the prediction involve a detailed description of all relevant changes?

For each metric, the judge assigns a score in the range of 0-5 for each output y_{t+1} . The overall score for Next-State-Generation is the sum of all three metrics in the range of 0-15.

Next-State-QA is a visual question answering (VQA) task that poses yes-or-no questions about the future state X_{t+1} given current state X_t and action a_t . This setup differs from existing GUI understanding and grounding tasks, which focus

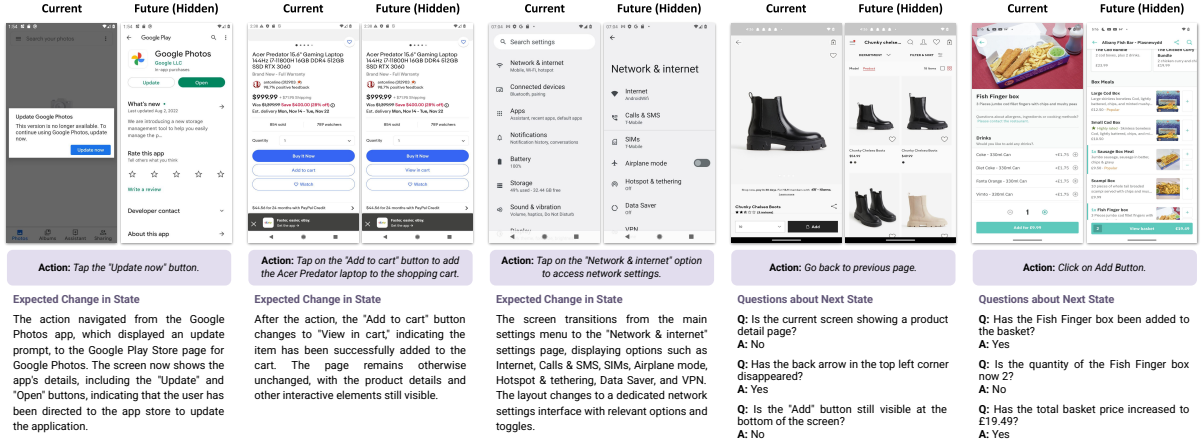


Figure 4: **Examples from MobileWorld Training Set.** We show qualitative examples of Next-State-Generation and Next-State-QA tasks in the training set of MobileWorld, highlighting the effectiveness of our data pipeline.

only on the *observed* GUI state or user interactions. We report answer accuracy as the evaluation metric for this task.

3.2.2 Data Generation and Filtering Pipeline

Our pipeline for creating MobileWorldBench involves three steps: trajectory sourcing, VLM annotation, and quality-based filtering.

Trajectory Sourcing. To build MobileWorldBench, we source trajectories from the test split of the AndroidControl dataset (Li et al., 2024a), which contains human demonstration trajectories with (X_t, a_t, X_{t+1}) triplets. The actions a_t come in the form of natural language descriptions, e.g. “Click on the OK option”. We also source trajectories from Android in the Wild (AiTW) (Rawles et al., 2023), which contains low-level user actions such as “Click (233, 324)”. In total, 250 (X_t, a_t, X_{t+1}) triplets are sampled for the Next-State-Generation and 500 triplets are sampled for Next-State-QA.

VLM Annotation. Since some trajectories only contain low-level actions, we employ a frontier VLM (Qwen3-VL-235B-A22B) to convert these low-level actions to high-level action descriptions. We find that naively passing in pixel coordinates leads to low performance, as the model struggles to correctly interpret these coordinate values. Instead, we create visualizations of actions by overlaying markers indicating the action on the screen (Yang et al., 2023). We provide additional processing details in the Appendix.

After we obtain high-level actions for all samples, we proceed to create QA proposal. We provide X_t, a_t, X_{t+1} to GPT-4o and prompt the model to propose 8 question-answer pairs based on the

observed differences between X_t and X_{t+1} , this leads to 4,000 question-answer candidates.

Quality-based Filtering. We first conduct human verification of the quality of high-level action descriptions. We find that these descriptions are of high quality and there are no obvious errors in all generated examples. After this step, the 250 samples for Next-State-Generation task can be used without further filtering, since the trajectories are sampled from real human demonstrations and the quality of action annotations has been checked. For Next-State-QA tasks, we heavily filter the machine-generated QA pairs to ensure only meaningful, non-trivial questions are retained. The filtering process consists of three steps. First, we prompt GPT-4o to answer its own questions by providing it with (X_t, a_t, X_{t+1}) triplets, and removing the questions that it fails to answer even with ground truth images (Self-Check). Second, we use GPT-4o as a judge to filter out questions about irrelevant GUI elements such as system time, battery level, *etc.* (unless the element is directly relevant to the user action). Finally, we perform human verification on the remaining question-answer pairs. Human evaluators are instructed to provide a ground truth answer for the question based on (X_t, a_t, X_{t+1}) , and asked to determine whether the question is relevant. We remove all instances where human answers disagree with the GPT-4o annotation and remove examples that are deemed irrelevant by annotators. The final filtered QA dataset contains 1,787 questions.

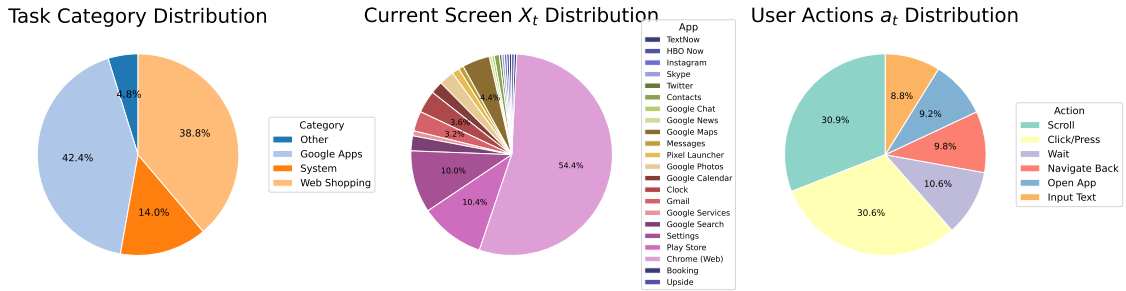


Figure 5: **Distribution of Tasks, Apps, and Actions in MobileWorldBench.** MobileWorldBench covers a wide range of task categories, Apps, and action types. We visualize their distributions for reference.

3.2.3 Coverage of Tasks, Apps, and Actions.

We followed the task distribution of AiTW and Android Control to ensure fair coverage. This is visualized in fig. 5. Specifically, each (X_t, a_t, X_{t+1}) comes from a multi-turn human demonstration whose tasks fall into four broad categories: Google Apps (e.g. Mail, News, Calendar), System (e.g. install), Web Shopping (e.g. Amazon) and Other third-party applications. The sampled screenshots X_t can also come from 22 distinct apps, including Google Apps such as Google Maps and third-party apps like HBO Now, Skype, Twitter (X). We note that there is no strict mapping between task categories and Apps, and the distinction of Apps is not a subdivision of more general task categories. We also visualize the distribution of user actions a_t , which include common mobile GUI interactions such as scroll, click, wait, etc.

3.3 MobileWorld

Our second contribution is MobileWorld, a large foundational dataset for training semantic world models, which consists of 1.4M samples. It includes state transition triplets X_t, a_t, X_{t+1} sourced from human demonstrations, and text descriptions y_{t+1} describing the changes between X_t and X_{t+1} resulting from action a_t , as well as question-answer pairs about X_{t+1} .

Trajectory Sourcing. We source triplets of (X_t, a_t, X_{t+1}) from human demonstrations in AiTW and the Android Control dataset largely following the same pipeline as MobileWorldBench described in 3.2. Unlike MobileWorldBench, we source from the training split of these datasets.

Annotations. We obtain raw text descriptions of state changes by prompting a VLM to describe the observed differences between X_t and X_{t+1} . We also obtain question-answer pairs by prompting the LLM to generate them based on observed state

changes. We generate 3 text descriptions and 8 question-answer candidates per state transition. Unlike the MobileWorldBench benchmark, we do not use GPT-4o as our VLM due to cost concerns. Instead, we use a strong open-sourced model Qwen3-VL-235B-A22B and Qwen3-VL-8B to generate annotations. 90% of the data is annotated using the 8B model, while 10% of the data is annotated using the 235B model. We denote these splits as “pretraining” and “finetuning split”.

Post-processing. For each state transition $X_t \rightarrow X_{t+1}$, we use VLM-as-a-judge to pick the best of three candidate text descriptions using the same criteria (accuracy, completeness, relevance) as MobileWorldBench. We also apply the same VLM filtering techniques for question-answer pairs. The final filtered dataset consists of 543k question-answer pairs and 942k state change descriptions. Due to cost concerns, the training set is not filtered by humans.

Task Coverage. The task coverage and distribution of categories is similar to that of MobileWorld-Bench (fig. 5). We provide the full distribution of our training set in the Appendix.

Quality Assessment. To test the quality of the training data pipeline, we run the two pipelines on the test set of Next-State-Generation task. Unlike standard evaluation, our data pipeline also allows the model to access ground truth next state images. We report the same benchmark scores, including accuracy, completeness, and relevance metrics. Further, we conducted human verification on a subset, collecting 1,800 human responses. Results show an error rate of 13.8% before our automatic filtering pipeline and 4.7% after filtering.

In summary, we conducted rigorous quality control procedures when building MobileWorldBench and MobileWorld. MobileWorldBench is fully human-verified, and MobileWorld employs auto-

matic quality filtering that is validated by humans on a small subset.

4 Experiments

We conduct extensive experiments to evaluate our benchmark, training dataset, and the proposed semantic world modeling paradigm.

Evaluation. We evaluate state-of-the-art VLMs on the Next-State-Generation and Next-State-QA tasks, including closed-source models (Comanici et al., 2025; Anthropic, 2025, 2024; OpenAI, 2024) and open-source models (Zhu et al., 2025; Yang et al., 2025).

Model Training. To further evaluate if the proposed training data can be used to improve model performance, we finetune Qwen3-VL-8B-Instruct using the MobileWorld dataset. Further details can be found in the Appendix.

4.1 Main Results on MobileWorldBench

We report evaluation results for Next-State-Generation (table 1) and Next-State-QA (table 2) tasks, and highlight key insights below.

Existing models have considerable room for improvement. We observe a significant performance gap between the best and worst models on both benchmarks. Notably, we find the best-performing model on the Next-State-QA benchmark, Gemini-2.5-Pro, is also one of the worst performing models on the Next-State-Generation benchmark. After careful inspection, we find that Gemini-2.5-Pro tends to generate long outputs with highly detailed, hallucinated descriptions of future states, leading to low accuracy scores. However, Gemini-2.5-Pro also has one of the highest relevance (4.48) and completeness (3.49) scores. This finding suggests that there are tradeoffs between different metrics, and Gemini-2.5-Pro prioritizes completeness and relevance over accuracy. We hypothesize that this behavior might emerge from its reasoning finetuning, as Gemini-2.5-Flash from the same family achieves a much higher generation score (+0.10) while maintaining strong QA accuracy.

Training Data Pipelines Yield High-Quality Annotations. In table 1, both the pretraining data pipeline, which uses Qwen3-VL-8B-Instruct as the annotator, and the finetuning pipeline, which uses Qwen3-VL-235B-A22B as the annotator, outperforms all existing models, suggesting that the pipeline was able to utilize the provided next-state

observations X_{t+1} to generate high-quality descriptions of state transitions. These improvements in terms of accuracy and completeness scores are more pronounced than the improvements in relevance score, which is to be expected. Most notably, the gap between the pretraining and finetuning data pipeline is relatively small, suggesting that Qwen3-VL-8B-Instruct has strong enough GUI understanding capabilities to serve as a high-quality annotator.

Finetuning on MobileWorld improves performance on both tasks. We find that our finetuned Qwen3-VL-8B-Instruct improves considerably on both generation and QA tasks, highlighting the utility of the MobileWorld dataset. Notably, on generation tasks, the finetuned model achieves the highest accuracy (71.4) score and overall score (12.39).

4.2 Online Evaluation.

Recent mobile GUI agents often rely on intricate scaffolds (*e.g.* memory, RAG) to maximize performance. To isolate the contribution of semantic world models, we conduct a simple experiment comparing agents with and without a semantic world model. We use M3A (Rawles et al., 2024) as the base agent with Qwen3-VL-235B-A22B and Gemini-3.5-Pro the VLM. We evaluate four setups: (1) no world model, (2) directly prompt the VLM to score the actions without explicit next-state prediction (3) using the pretrained VLM as both policy and world model, and (4) our finetuned Qwen3-VL-8B-Instruct as the world model. For (3)(4), we implement a model-based policy agent using the semantic world modeling framework discussed in section 3.1. We report task success rates in Table 3. We find that using either world model outperforms the M3A baseline, with our fine-tuned world model leading to the best performance.

We also generalize this setup to web-navigation in a zero-shot setting and report results on WebVoyager (He et al., 2024) in Table 4. At the time of experiments, 4 websites triggers captcha verification and cause the agent to fail. For fair comparison, we reproduced the baseline and report results in Table 4. Additional details of these experiments are included in the appendix.

4.3 Human Evaluation.

To further verify the performance of our finetuned model, we conduct an LM-Arena-style (Chiang et al., 2024) human evaluation over 3,000 randomly sampled “matches” between models. Each match pairs outputs from two different models for the

Table 1: **Quantitative Results on Next-State-Generation Tasks.** We report the performance of frontier VLMs and our finetuned model. We also reported the performance of our training data annotation pipeline (*Annotator*.)

Name	Per Category Score				Breakdown Score			Overall
	General	Google Apps	System	Web Shopping	Accuracy	Completeness	Relevance	
With GT Next State Image								
Qwen3-VL-235B-A22B-Annotator	13.17	13.12	12.34	13.30	4.45	4.02	4.61	13.08
Qwen3-VL-8B-Instruct-Annotator	13.58	13.09	12.60	13.09	4.42	4.02	4.61	13.05
Without GT Next State Image								
Intern-VL3-78B (Zhu et al., 2025)	12.92	11.96	11.26	11.70	4.02	3.41	4.38	11.81
Qwen3-VL-235B-A22B (Yang et al., 2025)	13.08	12.34	11.09	12.23	4.11	3.53	4.51	12.16
Gemini-2.5-Flash (Comanici et al., 2025)	13.08	12.56	10.77	11.91	4.14	3.48	4.45	12.08
Gemini-2.5-Pro (Comanici et al., 2025)	12.92	11.96	11.23	12.16	4.04	3.49	4.46	11.98
Claude-Sonnet-4.5 (Anthropic, 2025)	12.75	12.43	11.03	12.06	4.14	3.49	4.48	12.11
Claude-Sonnet-4 (Anthropic, 2024)	13.25	12.26	11.20	12.03	4.07	3.59	4.41	12.07
GPT-4o (Hurst et al., 2024)	12.75	12.38	11.23	11.77	4.11	3.44	4.44	12.00
Qwen3-VL-8B-Instruct (Yang et al., 2025)	12.83	12.26	10.80	11.64	4.04	3.42	4.38	11.84
+SFT (Ours)	12.83	12.40	11.63	12.61	4.19	3.70	4.50	12.39
Δ% (vs Baseline)	+0.0%	+1.1%	+7.7%	+8.3%	+3.7%	+8.2%	+2.7%	+4.7%

Table 2: **Quantitative Results on Next-State-QA Tasks.** We report the performance of VLMs and human evaluation. Acc-TestMini refers to the result on the TestMini subset with 500 randomly chosen samples.

Name	Acc-Full	Acc-TestMini
<i>With GT Next State Image</i>		
GPT-4o	100.0	100.0
Human	100.0	100.0
Qwen3-VL-8B-Instruct	94.9	93.2
<i>Without GT Next State Image</i>		
Human	84.4	89.2
Intern-VL3-78B	61.0	61.8
Qwen3-VL-235B-A22B	65.1	66.2
Gemini-2.5-Flash	76.9	77.6
Gemini-2.5-Pro	79.1	79.0
Claude-Sonnet-4.5	71.7	72.8
Claude-Sonnet-4	70.3	72.8
GPT-4o	66.0	69.6
Qwen3-VL-8B-Instruct	67.3	67.6
+SFT (Ours)	71.4	74.8
$\Delta\%$ (vs Baseline)	+4.1	+6.4

same question from the Next-State-Generation task of MobileWorldBench. We ask a human judge to pick a winner based on how helpful the model outputs are for decision-making. The human judge also has access to ground truth next state images X_{t+1} . We report ELO scores which assigns a numerical score to each model based on match results, and is widely used to evaluate LLMs, VLMs, image generators, and other types of generative AI models (Chiang et al., 2024; Jiang et al., 2024; Chi et al., 2025). These results are shown in fig. 6. Overall, human evaluation shows a considerable performance gain after finetuning on MobileWorld. Further, we also verified the correlation between GPT-4o judge scores in Table 1 and human preferences. We find that the accuracy score achieves 83% acc in predicting human preferences. Human preference is less correlated to completeness scores

Table 3: **Online Evaluation on AndroidWorld.**

Model	SR
Qwen3-VL-235B-A22B	46.9
+Value Function for Actions	49.1
+Semantic WM (Zero Shot)	50.8
+Semantic WM (Ours)	54.3
Gemini-2.5-Pro	63.4
+Semantic WM (Ours)	66.1

Table 4: **Zero-shot Generalization on WebVoyager.**

Model	SR
WebVoyager (GPT-4o)	45.4
+Semantic WM (Ours)	49.3

5 Conclusion.

In this work, we advocate for adopting semantic world models that predict state transitions at a higher level of abstraction than pixel-based world models. To support this paradigm, we introduce MobileWorldBench, a high-quality, human-verified benchmark that evaluates semantic world models’ ability to accurately predict future states. We also present MobileWorld, a large-scale dataset for training semantic world models, and demonstrate that training on this dataset leads to substantial performance gains. Finally, we showcase that semantic world models can be effectively integrated into mobile agents through a simple model-based policy, seamlessly translating world-modeling capabilities into task success in real-world environments.

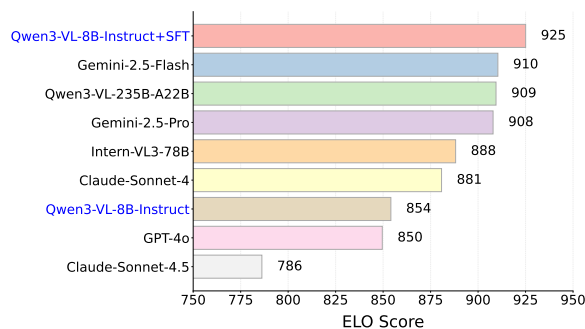


Figure 6: **Model ELO ratings from human evaluation.**

6 Limitation

Our work has limitations. For instance, both MobileWorld and MobileWorldBench consist solely of human demonstrations on Android, as there iOS is a close-sourced systems with complicated legal implications. We note that this limitation is common in mobile agent literature.

7 Ethic Considerations

Use of human annotators are approved and reviewed following respective procedures of institution and local law jurisdictions. All human annotators are properly compensated according regional income level.

8 Acknowledgement

AG would like to acknowledge the support from Schmidt Sciences and NSF Career Award #2341040. SL is in part supported by Amazon Fellowship.

References

Anthropic. 2024. [Introducing claude 4.0](#). Blog post announcement.

Anthropic. 2025. [Introducing claude sonnet 4.5](#). Blog post announcement.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zhohus, and 1 others. 2025. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*.

Hao Bai, Yifei Zhou, Jiayi Pan, Mert Cemri, Alane Suhr, Sergey Levine, and Aviral Kumar. 2024. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *Advances in Neural Information Processing Systems*, 37:12461–12495.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. 2024. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*.

Jacob Berg, Chuning Zhu, Yanda Bao, Ishan Durugkar, and Abhishek Gupta. 2025. Semantic world models. *arXiv preprint arXiv:2510.19818*.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, and 1 others. 2024. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*.

Hyunjoo Chae, Namyoung Kim, Kai Tzu-iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. 2024. Web agents with world models: Learning and leveraging environment dynamics in web navigation. *arXiv preprint arXiv:2410.13232*.

Delong Chen, Willy Chung, Yejin Bang, Ziwei Ji, and Pascale Fung. 2025a. Worldprediction: A benchmark for high-level world modeling and long-horizon procedural planning. *arXiv preprint arXiv:2506.04363*.

Delong Chen, Theo Moutakanni, Willy Chung, Yejin Bang, Ziwei Ji, Allen Bolourchi, and Pascale Fung. 2025b. Planning with reasoning using vision language world model. *arXiv preprint arXiv:2509.02722*.

Dongping Chen, Yue Huang, Siyuan Wu, Jingyu Tang, Liuyi Chen, Yilin Bai, Zhigang He, Chenlong Wang, Huichi Zhou, Yiqiang Li, and 1 others. 2024. Gui-world: A video benchmark and dataset for multi-modal gui-oriented understanding. *arXiv preprint arXiv:2406.10819*.

Taiye Chen, Xun Hu, Zihan Ding, and Chi Jin. 2025c. Learning world models for interactive video generation. *arXiv preprint arXiv:2505.21996*.

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, and 1 others. 2025d. Guicourse: From general vision language model to versatile gui agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21936–21959.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. Seeclck: Harnessing gui grounding for advanced visual gui agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9313–9332.

Wayne Chi, Valerie Chen, Anastasios Nikolas Angelopoulos, Wei-Lin Chiang, Aditya Mittal, Naman Jain, Tianjun Zhang, Ion Stoica, Chris Donahue, and Ameet Talwalkar. 2025. Copilot arena: A platform for code llm evaluation in the wild. *arXiv preprint arXiv:2502.09328*.

- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. [Chatbot arena: An open platform for evaluating llms by human preference](#). *Preprint*, arXiv:2403.04132.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hirschman, Daniel Aferegan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 845–854.
- Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.
- Jinwei Gu. 2025. Cosmos world foundation models for physical ai. In *Proceedings of the 3rd International Workshop on Rich Media With Generative AI*, pages 39–39.
- Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. 2025. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*.
- Mariam Hassan, Sebastian Stapf, Ahmad Rahimi, Pedro Rezende, Yasaman Haghighi, David Brüggemann, Isinsu Katircioglu, Lin Zhang, Xiaoran Chen, Suman Saha, and 1 others. 2025. Gem: A generalizable ego-vision multimodal world model for fine-grained ego-motion, object dynamics, and scene composition control. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22404–22415.
- Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. Webvoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890.
- Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and 1 others. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Dongfu Jiang, Max Ku, Tianle Li, Yuansheng Ni, Shizhuo Sun, Rongqi Fan, and Wenhao Chen. 2024. Genai arena: An open evaluation platform for generative models. *Advances in Neural Information Processing Systems*, 37:79889–79908.
- Yiding Jiang, Shixiang Shane Gu, Kevin P Murphy, and Chelsea Finn. 2019. Language as an abstraction for hierarchical deep reinforcement learning. *Advances in neural information processing systems*, 32.
- Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem AlShikh, and Ruslan Salakhutdinov. 2024. Omniaact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In *European Conference on Computer Vision*, pages 161–178. Springer.
- Gang Li and Yang Li. 2022. Spotlight: Mobile ui understanding using vision-language models with a focus. *arXiv preprint arXiv:2209.14927*.
- Wei Li, William E Bishop, Alice Li, Christopher Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2024a. On the effects of data scale on ui control agents. *Advances in Neural Information Processing Systems*, 37:92130–92154.
- Yanda Li, Chi Zhang, Wenjia Jiang, Wanqi Yang, Bin Fu, Pei Cheng, Xin Chen, Ling Chen, and Yunchao Wei. 2024b. Appagent v2: Advanced agent for flexible mobile interactions. *arXiv preprint arXiv:2408.11824*.
- Xiao Liu, Bo Qin, Dongzhu Liang, Guang Dong, Hanyu Lai, Hanchen Zhang, Hanlin Zhao, Iat Long Iong, Jiadai Sun, Jiaqi Wang, and 1 others. 2024. Autoglm: Autonomous foundation agents for guis. *arXiv preprint arXiv:2411.00820*.
- Quanfeng Lu, Wenqi Shao, Zitao Liu, Lingxiao Du, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, and Ping Luo. 2025. Guiodyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22404–22414.
- Dezhao Luo, Bohan Tang, Kang Li, Georgios Papoudakis, Jifei Song, Shaogang Gong, Jianye Hao, Jun Wang, and Kun Shao. 2025. Vimo: A generative visual gui world model for app agents. *arXiv preprint arXiv:2504.13936*.
- Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2024. Coco-agent: A comprehensive cognitive mllm agent for smartphone gui automation. *arXiv preprint arXiv:2402.11941*.

- Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron Courville, and Sai Rajeswar. 2024. Multimodal foundation world models for generalist embodied agents. In *Multi-modal Foundation Model meets Embodied AI Workshop@ ICML2024*.
- OpenAI. 2024. [Hello gpt 4o](#). Blog post announcement.
- Christopher Rawles, Sarah Clinckemaulle, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, and 1 others. 2024. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*.
- Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2023. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36:59708–59728.
- Pengzhen Ren, Kaidong Zhang, Hetao Zheng, Zixuan Li, Yuhang Wen, Fengda Zhu, Mas Ma, and Xiaodan Liang. 2023. Surfer: Progressive reasoning with world models for robotic manipulation. *arXiv preprint arXiv:2306.11335*.
- Luke Rivard, Sun Sun, Hongyu Guo, Wenhui Chen, and Yuntian Deng. 2025. Neuralos: Towards simulating operating systems via neural generative models. *arXiv preprint arXiv:2507.08800*.
- Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. 2022. Meta-gui: Towards multi-modal conversational agents on mobile gui. *arXiv preprint arXiv:2205.11029*.
- Shulin Tian, Ziniu Zhang, Liang-Yu Chen, and Ziwei Liu. 2025. Mmina: Benchmarking multihop multi-modal internet agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13682–13697.
- Sagar Gubbi Venkatesh, Partha Talukdar, and Srin Narayanan. 2022. Ugif: Ui grounded instruction following. *arXiv preprint arXiv:2211.07615*.
- Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. 2021. Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 498–510.
- Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Juntong Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, and 1 others. 2025. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*.
- Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *Advances in Neural Information Processing Systems*, 37:2686–2710.
- Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. 2024. Autodroid: Llm-powered task automation in android. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 543–557.
- Hao Wen, Hongming Wang, Jiaxuan Liu, and Yuanchun Li. 2023. Droidbot-gpt: Gpt-powered ui automation for android. *arXiv preprint arXiv:2304.07061*.
- Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, and 1 others. 2024. Oworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.
- Jiabo Ye, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Zhaoqing Zhu, Ziwei Zheng, Feiyu Gao, Junjie Cao, Zhengxi Lu, and 1 others. 2025. Mobile-agent-v3: Fundamental agents for gui automation. *arXiv preprint arXiv:2508.15144*.
- Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. 2024. Ferret-ui: Grounded mobile ui understanding with multimodal llms. In *European Conference on Computer Vision*, pages 240–255. Springer.
- Hu Yue, Siyuan Huang, Yue Liao, Shengcong Chen, Pengfei Zhou, Liliang Chen, Maoqing Yao, and Guanghui Ren. 2025. Ewmbench: Evaluating scene, motion, and semantic quality in embodied world models. *arXiv preprint arXiv:2505.09694*.
- Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2025. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20.
- Longtao Zheng, Zhiyuan Huang, Zhenghai Xue, Xinrun Wang, Bo An, and Shuicheng Yan. 2024. Agentstudio: A toolkit for building general virtual agents. *arXiv preprint arXiv:2403.17918*.
- Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, and 1 others. 2025. World4drive: End-to-end autonomous driving via intention-aware physical latent world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28632–28642.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

Supplementary Material

A Additional Technical Details

A.1 Dataset Statistics

Comparison with Other Datasets. We compare the size and task coverage in table 5. As shown, MobileWorld is the first dataset focusing on next-state prediction tasks. Its scale is also larger than most of existing datasets. We emphasize that while some datasets such as GUI-World also involves questions about multi-step trajectories, they still fall in the category of GUI understanding tasks and are fundamentally different from our world-modeling next-state prediction tasks.

As a concrete examples, GUI-World contains questions such as “After moving the Steam window to the center, what did the user do next in the Edge browser?” A model is expected to answer this question based on a list of video frames from screen recording, including all past and future actions. This is a different setup than world modeling, where future states are not provided to the model.

Distribution of MobileWorld. We provided the distribution of apps and actions of MobileWorld-Bench in Figure 5 of the main paper. The distribution of apps and actions of MobileWorld is similar to that of MobileWorldBench since they are sampled from the same source of human demonstrations. For completeness, we also include a detailed account of these distributions in fig. 7. A major difference is that there are more click operations in the training data, which can be attributed to the differences in the train and test split of source data.

A.2 Data Pipeline

A.2.1 VLM Annotation

In this section, we provide details of the VLM annotation process. Recall that the VLM annotators serve three goals. First, they convert low level actions such as “Click (200,312)” to high-level ones such as “Click the return button”. Second, they generate text descriptions of the visual differences in GUI state before and after an action is performed. Finally, they generate candidate QA pairs.

Action Representation. Instead of only providing low level actions as text, we also annotate the image visually because we find VLMs struggle to map screen coordinates to GUI elements. An example is shown in fig. 8, where we annotate click

actions with a cross and swipe or scroll actions with an arrow.

High-level Action and Change Description.

We provide a total of three images in this annotation process. Image 1 and Image 2 are screenshots of mobile phones before a certain action. Image 2 is Image One plus the action visualization. Image 3 is the screenshot after the action is performed. The full prompt is documented as below

Annotation Prompt for High-level Action and Change Description

You are given three images which are screenshots of mobile phones before and after a certain action. Image 1 and Image 2 are states before the action. Image 2 is image one with an action annotated.

The action coordinate is marked with a blue cross. If the action is swipe, the swipe direction is marked with a blue arrow. The end of the arrow indicates the end point of the swipe.

Image 3 is the screenshot after the action is performed.

The input images have the dimension of $[[\text{Height}]] \times [[\text{Width}]]$ (Height x Width). The action will be provided in the prompt. The overall goal will also be provided.

There are two tasks:

- You should also provide a natural language description of the action based on the images. For example, if the raw action provided is Tap at (200, 1112), and the image shows that the user tapped on the “Settings” icon, you should describe the action as “Tap on the ‘Settings’ icon to open the setting menu”. If action is already descriptive enough, (e.g, Input text “usb-c to usb-a”) without any coordinate, you can just repeat it. Feel free to add additional context based on the image to make the action more understandable, such as Input the text “usb-c to usb-a” in the search box to search for usb-c to usb-a adapters.
- Your task is to describe the difference between the two images in detail, focusing on the changes that occurred as a result of the action. Please pro-

Dataset	Size	Sem.	Task Coverage
Rico (Deka et al., 2017)	72,219	Low	UI Code/Layout Generation
Screen2Words (Wang et al., 2021)	22,417	High	UI Summarization
MetaGUI (Sun et al., 2022)	1,125	Low	Mobile Navigation
UGIF (Venkatesh et al., 2022)	523	High	GUI Parsing & Understanding
AITW (Rawles et al., 2023)	715,142	High	Action Selection
Ferret-UI (You et al., 2024)	123,702	Low	GUI Grounding & Understanding
Spotlight (Li and Li, 2022)	2.5M	Low	GUI Understanding
GUI-WORLD (Chen et al., 2024)	12,379	Both	GUI Understanding
MobileWorld (Ours)	1.4M	Both	Next-State Prediction

Table 5: **Overview of existing mobile GUI datasets.** Sem. Semantic level on instructions.

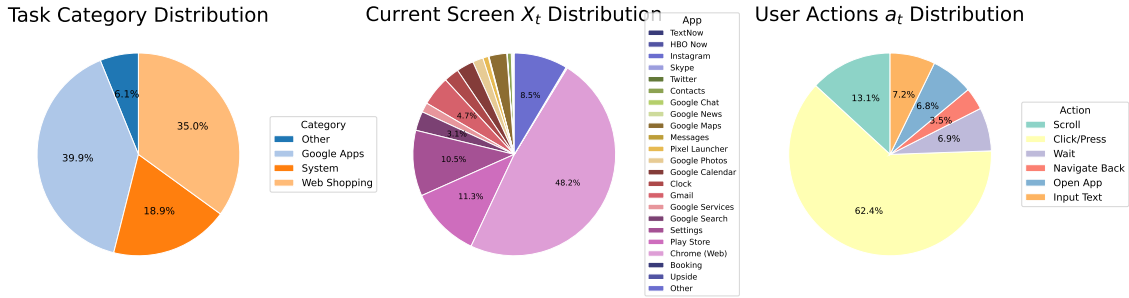


Figure 7: **Distribution of Tasks, Apps, and Actions in MobileWorld.** MobileWorld covers a wide range of task categories, Apps, and action types. We visualize their distributions for reference.

vide a comprehensive description of the differences, including any visual changes, layout modifications, color alterations, text changes, or any other noticeable differences. The overall goal is for reference only, do not include it in your answer or rely on it too much. It may not be fully accurate. Limit your response to one small paragraph less than 200 words. Only describe the changes, no need to say which things are not changed. Also feel free to ignore system status such as wifi, battery, etc. If action is not successful or nothing is changed, just say No changes observed after the action. And explain why it fails

The action is: "{low level action}"

The goal is "{goal}"

Your response should be in the following format:

Action Description: [Your description of the action based on the images.]

Change Description: [Your description of

the changes between Image 1 and Image 3.]

QA proposals. We sample 8 question-answer (QA) proposals for each (X_t, a_t, X_{t+1}) triplet. The prompts are documented as follows

Annotation Prompt for Creating QA Candidates

You are given two images representing the before and after states of an Android device screen. The action being performed is {action}.

Now generate 5 QA pairs that test a model's understanding of the changes between these two images. Specifically, you should provide yes/no questions about UI states after the change, as well as ground truth answers based on the changes observed.

Do you include changes that are not directly related to the action, such as system time, battery level, etc. For each QA pair, format it as:

Q: [question]

A: [answer]

Provide 8 such QA pairs in total.

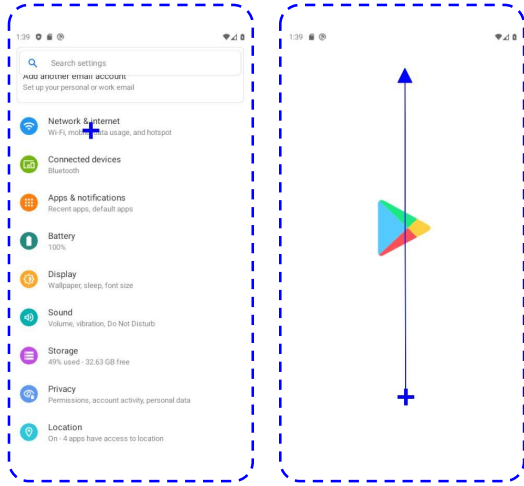


Figure 8: **Action Annotation via Visual Overlay.** To better allow VLM annotators to understand the action being performed, in addition to integer coordinates, we provide additional visual annotations on the coordinates of user gestures visually.

Provide a balanced mix of true and false answers.
Use exactly the format provided, do not add any extra text or formats (e.g. bold)

Given 500 state transitions, the total number of generated QA candidates are 4,000.

A.2.2 VLM Filtering

We perform additional filtering to 1) remove questions whose answer does not accurately reflect state-changes 2) remove irrelevant question-answer pairs relating to system time, battery level, etc. In particular, we find 2) is necessary even though we have instructed the model not to produce this kind of questions during the generation process, as we find that the model will occasionally ignore these instructions and create trivial or irrelevant questions regardless.

To achieve 1) removing questions whose answer does not accurately reflect state-changes, we perform self-check verification where the model is provided with the current state, user action, and ground-truth next state to answer a question. The prompt is documented as follows:

Prompt for VLM Self-Check

You are an intelligent GUI agent capable of understanding GUIs and actions on mobile devices. Given the current GUI screen-

shot and input action, answer the following questions based on your predictions of the changes that will occur on the next screen after the action is performed.

The action is {action}

Answer with yes or no.

The question is {question}.

You will also be given ground truth next state image as image 2.

To achieve 2) removing irrelevant question-answer pairs, we employ the following prompt.

Prompt for Relevance Filter

You are given two images representing the before and after states of an Android device screen. The action being performed is {action}.

A question-answer proposal is given as follows:

Q: {question}

A: {Answer}

Please decide if the question asks about expected state changes that are relevant to the action. Non-relevant actions include system time, battery level, Wi-Fi status, signal strength, etc., unless these aspects are directly related to the current action (e.g. enabling airplane mode will disable cellular signal).

Note that even if a question asks about some UI elements that did not actually change, it may still be relevant. For example, when the user is typing the city column of an address form, it is ok to ask if the actions will result in any changes in the street address column. Answer with a simple yes-or-no.

After the filtering process, we are left with 2,458 QA pairs out of 4,000 proposals.

A.2.3 Human Filtering

For the remaining 2,458 QA pairs, we conduct additional human filtering to further ensure the quality of the benchmark. We employ Amazon Mechanical Turk for this process. To ensure the quality of the worker, the authors annotated a small 100-sample test set manually and only employ workers with high performance on this test set to work on the full set. The human annotation process consists of the following two steps:

Filtering Incorrect or Irrelevant QA pairs.

The purpose of this step is similar to the VLM filtering process. We ask the human annotator to judge 1) if the ground truth answer is consistent with the observed state changes observed in screenshots, 2) If the question is relevant to the action being performed. We list the prompt in Figure 9a.

Ambiguity. While a question can be correct and relevant, it may be ambiguous or under-defined if it asks about something that is impossible to reasonably predict. For example, when opening the "sports" tab of a new app, it is impossible to predict what the headline will be. Hence, we additionally ask human annotators to judge if the questions can be reasonably answered by an average user. The interface is shown in fig. 9b

After this process, we are left with 1,787 QA pairs as our benchmark.

B Additional Experiment Details and Results

B.1 Training Setup

We finetune the base model Qwen3-VL-8B-Instruct (Yang et al., 2025) on the proposed MobileWorld dataset for 2 epochs, with a learning rate of 2×10^{-6} on 8 A6000 GPUs. The detailed training setup is listed in table 6

Table 6: **Training configurations on MobileWorld.** We report the relevant hyperparameters for training, including the learning rate (LR), number of training steps, optimizer setup, image resolution for understanding and generation tasks.

	SFT
Learning Rate (LLM Backbone)	2×10^{-6}
Learning Rate (Vision Encoder)	2×10^{-7}
Epochs	2
β_1	0.99
β_2	0.999
Optimizer	AdamW
Image resolution	1280px
Batch Size	128
Warmup	6%
LR Schedule	Cosine

B.2 Evaluation Setup

B.2.1 Model Choice and Sampling Parameters

In this section, we document the exact model version used to reproduce our main results, as well as the sampling parameters. The model ids of closed source models are listed in table 7. For open-sourced models, we use their Hugging Face releases. For next-state-generation tasks, we use the default sampling parameters of respective models and do not pass in extra parameters. For next-state-QA tasks, we set temperature to 0.0 for reproducibility. We noticed that setting the temperature to 0.0 will lead to text repetitions in some models for next-state-generation tasks. Hence, we set temperature to 0.0 only for short QA tasks.

B.2.2 Automatic VLM Evaluation

We employ GPT-4o as the judge model. We employ the following prompt to obtain the scores for next-state-generation results

Prompt for Automatic Evaluation

You are a judge to judge a VLMs's ability to understand the action on mobile device GUI. Given the current GUI screenshot and input action, the model will describe the changes that will occur on the next screen after the action is performed.

The action is {action}.

You are provided with the model's response, an input image of the current state (first image), as well as ground truth next state image (second image).

You are also given a reference text that describes the actual changes that happened after performing the action. Your task is to evaluate the model's response based on the following criteria:

1. Accuracy: Does the model's description accurately reflect the changes that occurred in the GUI after the action?
 2. Completeness: Does the model mention all significant changes that are visible in the next state image?
 3. Relevance: Are all the changes mentioned by the model relevant to the action performed, or are there extraneous details?
- For each criterion, assign a score from 1 to 5, where 1 is poor and 5 is excellent.

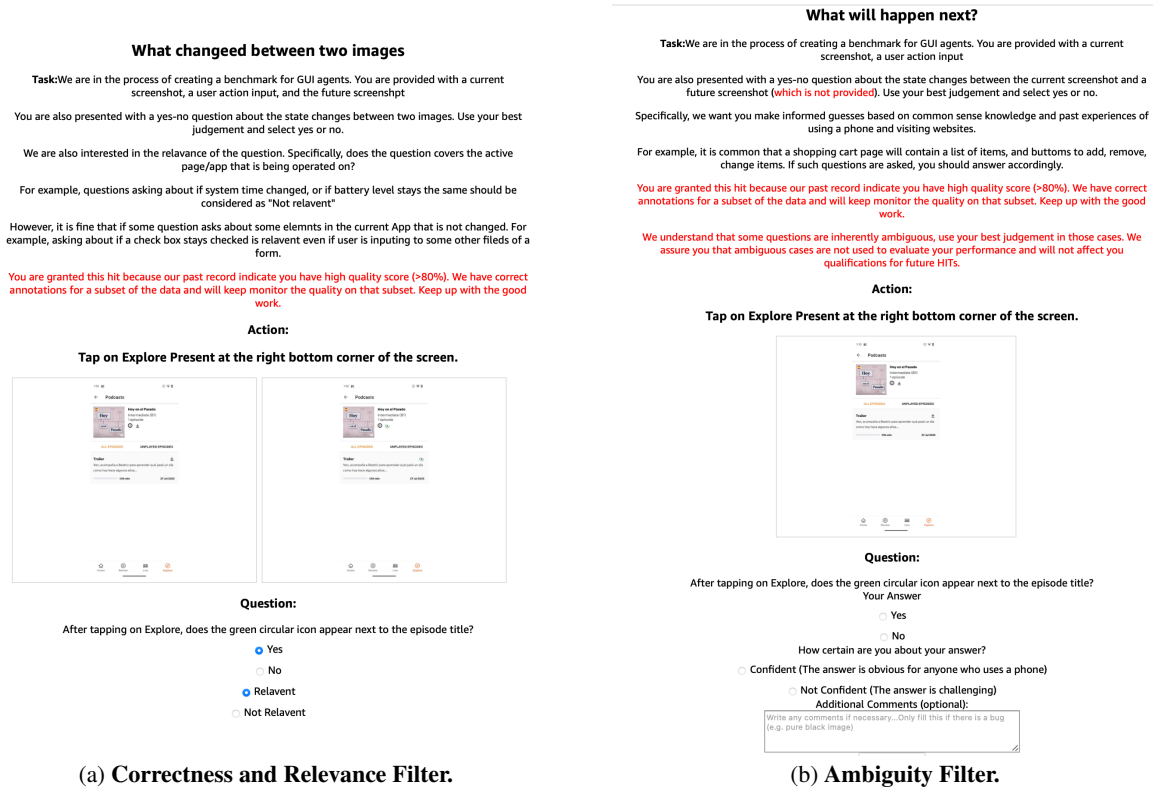


Figure 9: **Example Screenshots of the User Interface for Human Annotators Used in the Data Filtering Process.** (a) Interface for applying correctness and relevance filters. (b) Interface for detecting ambiguous responses.

Table 7: **Closed-Source Model Version Mapping.** We list the precise model identifiers corresponding to each name used in Table 1.

Name in Table	Specific Version / Model ID	License
Gemini-2.5-Flash	gemini-2.5-flash (stable)	Google APIs Terms of Service
Gemini-2.5-Pro	gemini-2.5-pro (stable)	Google APIs Terms of Service
Claude-Sonnet-4	claude-sonnet-4-20250514	Commercial Terms (Claude)
Claude-Sonnet-4.5	claude-sonnet-4-5-20250929	Commercial Terms (Claude)
GPT-4o	gpt-4o-2024-08-06 (default)	Terms of Use (OpenAI)

After evaluating all three criteria, provide an overall score out of 15, along with a brief justification for your scores.

The model’s response is {response}

The reference changes are {changes}

Format your evaluation as follows:

—Begin of response—

Accuracy: [score]

Completeness: [score]

Relevance: [score]

Overall Score: [total score]

—End of response—

use the exact format without any additional text.

B.2.3 Human Evaluation

In human evaluation, we find that forcing human evaluators to give a numerical score leads to inconsistent behavior across different annotators, whose score range varies. To address this, we instead ask human evaluators to compare outputs from two models.

We illustrated the human evaluation interface in fig. 10. In total, we create 3,000 “matches” by randomly selecting benchmark entries and model pairs. We compute ELO scores following the setup of LM-Arena and Gen-AI-Arena (Chiang et al., 2024; Zhou et al., 2023; Jiang et al., 2024)

B.2.4 AndroidWorld

For our AndroidWorld baseline, we use M3A implemented via the official AndroidWorld codebase (Rawles et al., 2024). For our model-based policy, a value model is needed to evaluate the predicted next states. We use the following prompt to obtain value scores for future states:

Prompt for Value Model

```
[...Omitted Background Description Copied
From Action Selection Prompt of M3A ...]
Now based on the goal and expected state
changes caused by the proposed 8 actions,
reason about the value of the resulting states
from these actions and score each of them
in the range of 1-10. Higher score means
the resulting state is more likely on the right
trajectory to reach the final goal.
Think carefully before giving the final answer.
Your Final answer should be the following
format (Expected_Change and Action should be
copied from above)
Action 1:  {{ "action_type":... }} Expected_Change: ... Score_Reason:... Score: ...
Action 2:  {{ "action_type":... }} Expected_Change: ... Score_Reason:... Score: ...
```

For better presentation, we omit the long background description that describes the current environment, past action history, overall goal, and some hard-coded tips like “Sometimes you may need to navigate the phone to gather information needed to complete the task.” Note that this portion is directly copied from M3A agents.

WebVoyager The standard WebVoyager agent provides a VLM (GPT-4V) with current screen observation and a list of possible actions. It asks the VLM to directly select an action from the provided list and provide reasoning. We modify this process by first generate next-state predictions using our semantic world model and then run a value model using the same setup as AndroidWorld Experiment.

In our experiments, we countered Captcha verification on the following websites: Google Search, Booking, Cambridge Dictionary, Allrecipes. These four websites are removed from benchmark since encountering Captcha necessitates task failure.

B.3 Additional Results

Statistic Significance Analysis. To verify the statistic significance of our results, we report the 95% confidence interval of next-state-generation task in Table 8. Results show that our improvements are statistic significance ($p < 0.05$).

B.3.1 More Qualitative Examples.

In fig. 11, we provide additional qualitative examples on screenshots, user actions, and model outputs on MobileWorld, as well as the GPT judge score. Model performance varies across different samples. Our finetuned model is among the top performing models for most of the samples.

B.3.2 Model-Scaling

To further analyze the difficulty of the proposed tasks and investigate the model scaling behavior, we conducted an additional experiment by finetuning a smaller model Qwen3-VL-2B-Instruct following the same setup. We report results in table 9. We observe that while the relative improvements of finetuning is larger for the 2B model (+5.4) versus the 8B model (+4.1), the 2B model significantly underperforms the 8B model. We believe this gap is largely caused by limited GUI understanding capabilities, of the 2B model. We verify this hypothesis by additionally evaluate the models’ performance when provided with the ground truth next-state image. This process converts next-state-QA to standard VQA tasks where the model only needs to compare the differences in the two provided images. We find that GPT4o, Human annotator, and Qwen3-VL-8B-Instruct achieves high scores in this setup ($> 90\%$), while Qwen3-VL-2B-Instruct only achieves 70.0% accuracy, indicating its limited capacity. These findings indicate that the proposed world modeling task is challenging and is beyond the capacity of small VLMs.

B.3.3 Combining Semantic World Models to Pixel-Space World Models

While we argue that semantic world models are conceptually more useful and less expensive to train than pixel world models, we emphasize that conceptually semantic world models can be considered as a sub-process of the classic world models, as noted in section 3.1. To validate this paradigm, we can pass the output of our semantic world model to a state-of-the-art pixel renderer to achieve pixel world models. We illustrate this approach in Figure 12, where we combine our finetuned VLM with

Model	Accuracy	Completeness	Relevance	Overall Score
Qwen3-VL-8B-Instruct	4.05 ± 0.07	3.37 ± 0.06	4.35 ± 0.06	11.77 ± 0.19
+SFT (Ours)	4.22 ± 0.07	3.72 ± 0.06	4.55 ± 0.06	12.49 ± 0.18

Table 8: Statistics Significance analysis. To further verify the significance of our results, we conducted additional analysis by sampling 5 times and report the 95% confidence interval. Due to token cost issue, we only perform this validation for open-sourced Qwen3-VL-8B-Instruct and our finetuned model. Results show that the improvements from finetuning on MobileWorld dataset are statistically significant.

Table 9: **Model Scaling Results.** We provide qualitative results on next-state-QA tasks of finetuned Qwen3-VL-2B-Instruct and Qwen3-VL-8B-Instruct.

Model	Acc
<i>With GT Next State Image</i>	
GPT-4o	100.0
Human	100.0
Qwen3-VL-8B-Instruct	94.9
Qwen3-VL-2B-Instruct	70.0
<i>Without GT Next State Image</i>	
Qwen3-VL-8B-Instruct	67.3
+SFT(Ours)	71.4 (+4.1)
Qwen3-VL-2B-Instruct	60.6
+SFT(Ours)	66.0 (+5.4)

frontier image generators such as Nano-Banana (Comanici et al., 2025) to produce pixel outputs. We find that accurate text prediction can be used to create highly plausible next-state screenshots. We emphasize that this works focus on the semantic level, we left more explorations in pixel world modeling for future works.

C Reproducibility Statements

All artifacts, including fine-tuned model weights, evaluation dataset, and training dataset will be open-sourced. We will also release the evaluation codebase for Next-State-Generation, Next-State-QA, and AndroidWorld experiments.

D Additional Discussion with Related Works.

ViMo. We note that a line of literature tangent to this work, notably ViMo (Luo et al., 2025) focuses on building pixel-space GUI world models by addressing challenges such as text rendering and GUI consistency. We are unable to compare with ViMo because the authors did not release the model checkpoints, training data, or evaluation data.

Language-based Abstraction for Decision-Making. Using language-based abstractions has been widely studied for decision-making problems. WorldPrediction (Chen et al., 2025a) provide a theoretical framework that recognizes the world state has "task-relevant components" and "task-irrelevant" components, and suggest "change captioning" can be used as an alternative to raw image inputs. Vision Language World Model (Chen et al., 2025b) performs world modeling in language space by prediction captions of future video frames. However, most of these works focus on real physical world with embodied agents. Our work specifically focus on introducing the semantic world modeling paradigm for mobile agents.

E Additional Limitations

Our work share many common limitations of VLM-based works, such as hallucinations in the generated output. We conducted rigorous human verification for our benchmark to ensure its quality. However, such approach is not fesaible for training data due to cost issue. As discussed in Section 3.3, human validation on a subset of training data showed that our automatic filtering pipeline reduced the error rate from 13.8% to 4.7%, but there may still be rooms for improvements for future works (e.g by using more costly models for filtering.)

Evaluate GUI Agents

Task: Your task is to evaluate the GUI world modeling capabilities of frontier LLMs. Specifically, these models are provided with a current screenshot and a user action, and they are asked to generate the expected changes that will occur as a result of user action

We also provide you with an actual image of future states. You should check which model's output is more preferred.

You should use the consider the following aspects:

Accuracy: Does the model predict the correct changes?

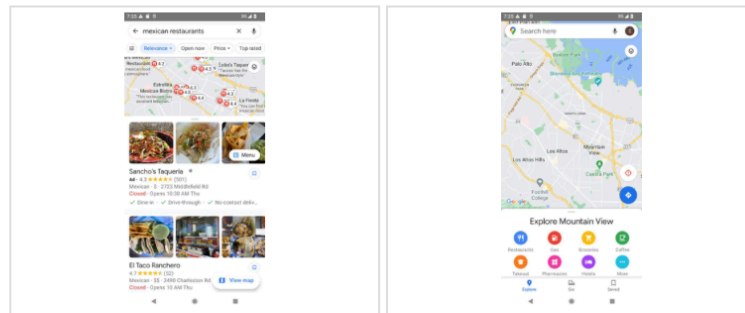
Relevance: Are the predicted changes relevant to user actions?

Completeness: Does the model detailed descriptions or just trivial descriptions

You should focus on high-level semantics and do not over penalize the model if it makes mistakes on small details. In general, think about which model's output would be more useful for decision making

Action:

Tap the back button at the top left corner of the screen to return to the main Maps interface.



Model A Output

Tapping the back button at the top left corner will exit the current search results for "mexican restaurants" and return to the main Google Maps interface. This will likely show the default map view centered on the user's current location or the last viewed location, with no active search query. The search bar will be cleared or reset, and any filters or sorting options (like "Relevance," "Open now," etc.) will no longer be active. The list of restaurant results and their associated photos will disappear, and the screen will revert to the general map view with possible nearby points of interest or directions if previously set.

Model B Output

When you tap on the Google search bar at the bottom of the screen, the following changes will occur: 1. The keyboard will appear on the screen, allowing you to type in a search query. 2. The search bar will likely enlarge, providing a clear text entry area. 3. An animated cursor or highlighted text field will indicate where you can start typing. 4. Depending on the device and settings, voice search or predictive text options might become visible. 5. Icons or buttons related to search (such as microphone for voice search) may appear above or around the search bar. This setup allows you to input a search query and perform a Google search.

☐ Model A is better

☐ Model B is better

Additional Comments (optional):

Write any comments if necessary...Only fill this if there is a bug (e.g. pure black image)

Submit

Figure 10: **Human Evaluation Interface of Comparing Two Models.**

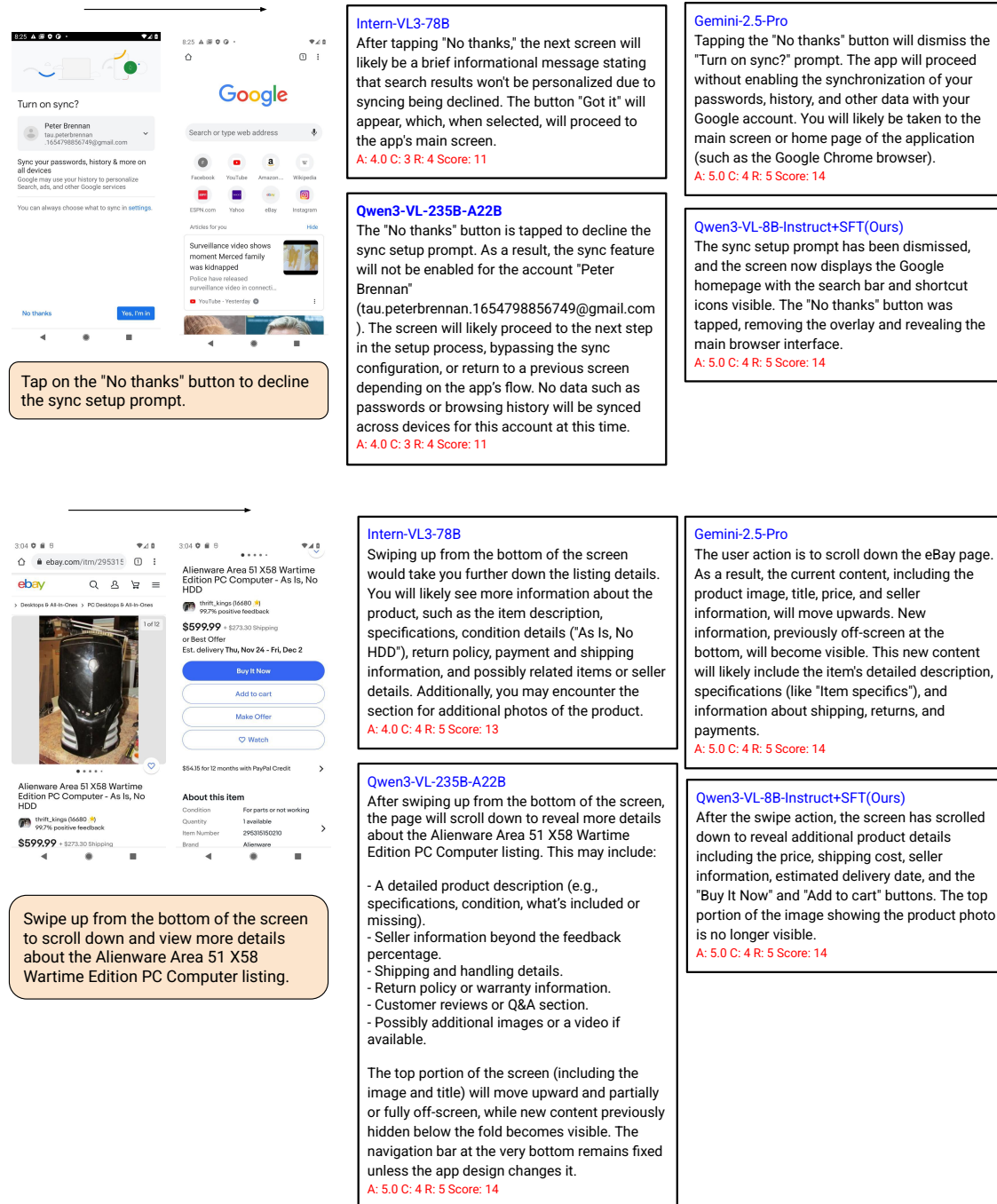


Figure 11: Qualitative Examples of MobileWorld Evaluation Results.

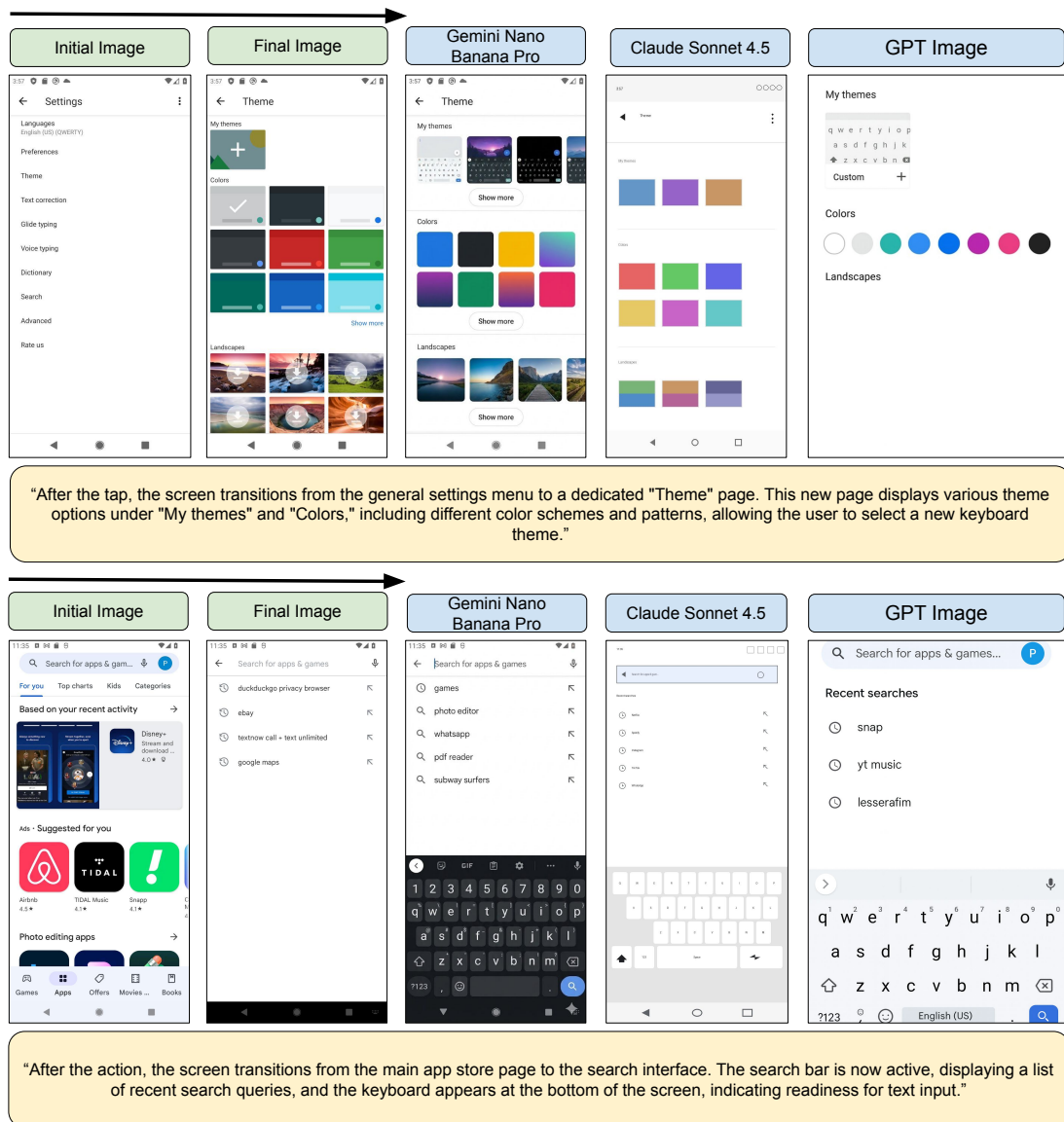


Figure 12: Qualitative Results of Pixel Modelling We combine the output of semantic world models with pixel renders to obtain image predictions. Gemini Nano Banana Pro achieves the best qualitative results. We demonstrate that using text as guidance in the generation process results in a prediction much closer to the ground truth.