ViMo: A Generative Visual GUI World Model for App Agent

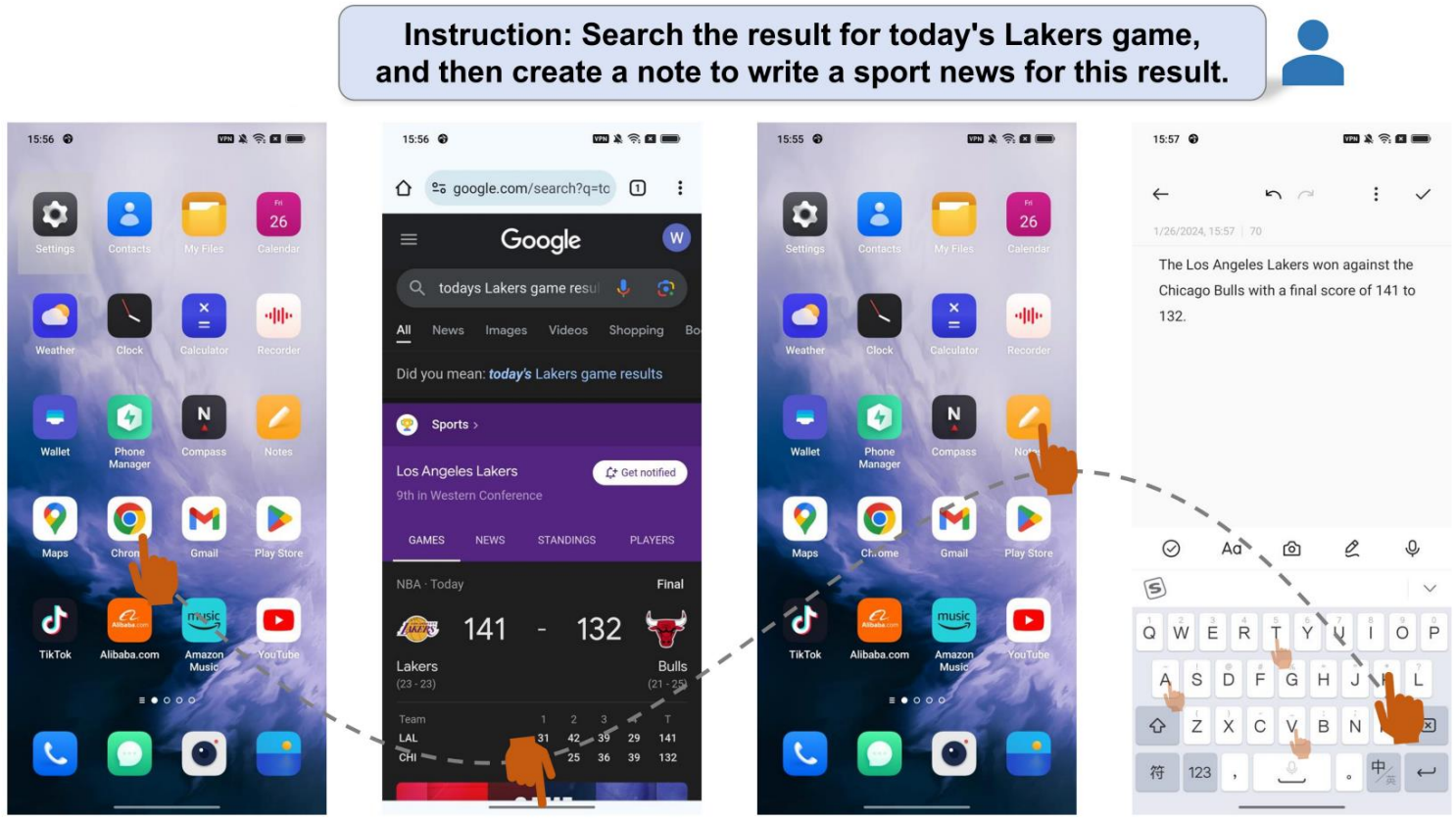
Generative World Models:
By observing the real world, world models can predict how the environment evolves in response to user actions

World model for game engine[1]:

[1] DIFFUSION MODELS ARE REAL-TIME GAME ENGINES, ICLR 2025

ViMo: A Generative Visual GUI World Model for App Agent

App Agent

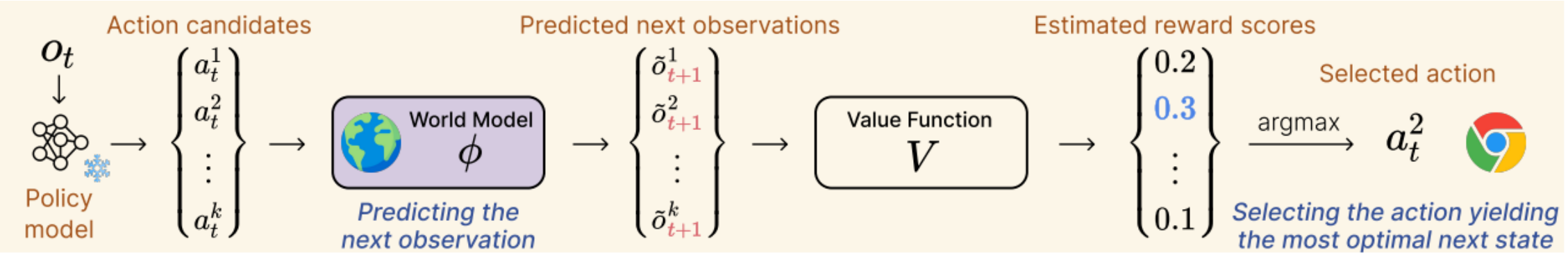


(Given an instruction, app agent automatically decides what to do in the app to achieve the task)

[2] MOBILE-AGENT: AUTONOMOUS MULTI-MODAL MOBILE DEVICE AGENT WITH VISUAL PERCEPTION, Arxiv 2024

ViMo: A Generative Visual GUI World Model for App Agent

Pipeline on App Agent with world model



Existing world models are language-based, lack details to fully represent the visual GUI states..

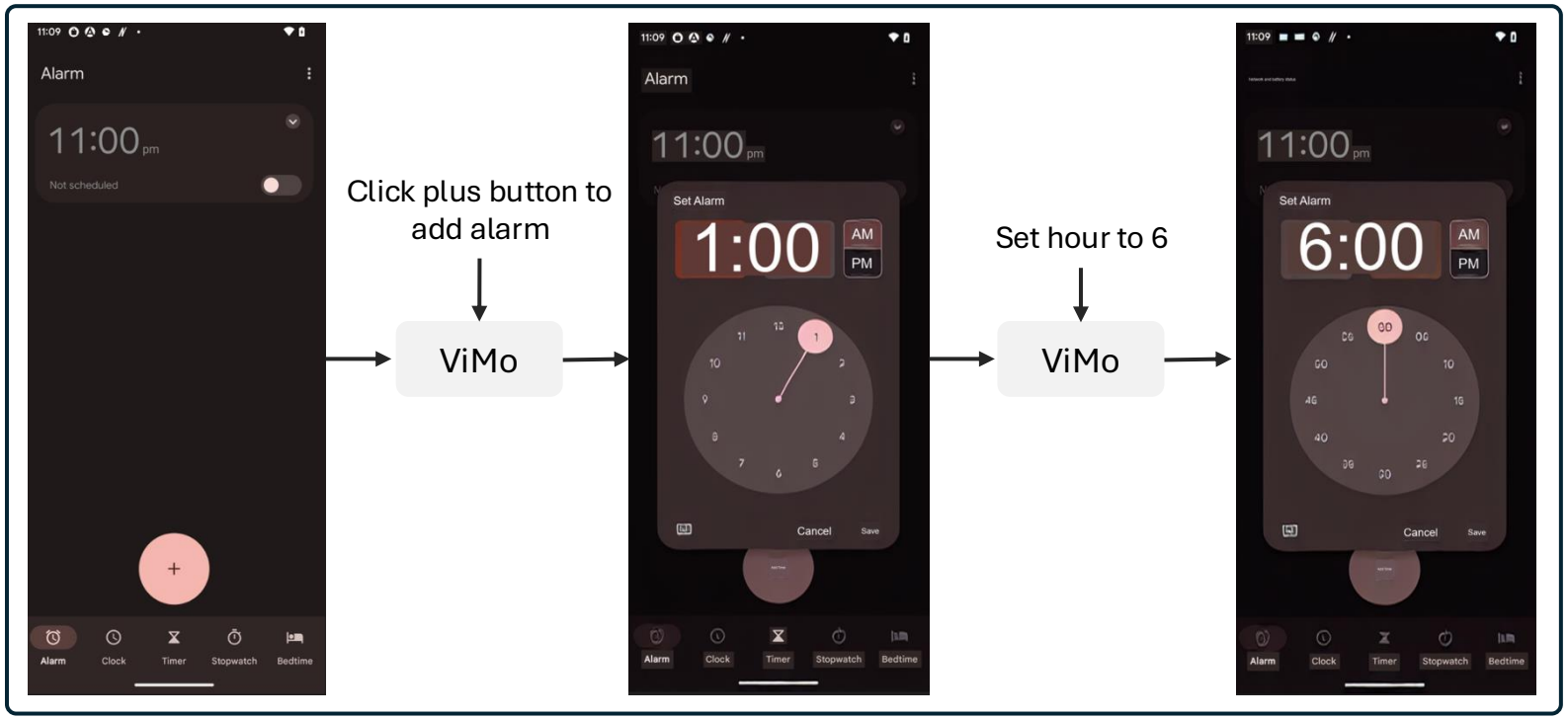
Why not use emulators / real-world environment?

a) Emulators carry out real-world interaction, not irreversible (send message/make payment)

b) Emulator is not easy to backtrack from O_{t+1} to O_t

ViMo: A Generative Visual GUI World Model for App Agent

Our target: A visual world model that predicts the next state of the GUI screenshot :



Problem: the generation of both graphics (UI element) and text

Computation resources limitations:
To generate plausible text without distortions, high-resolution generation is required:

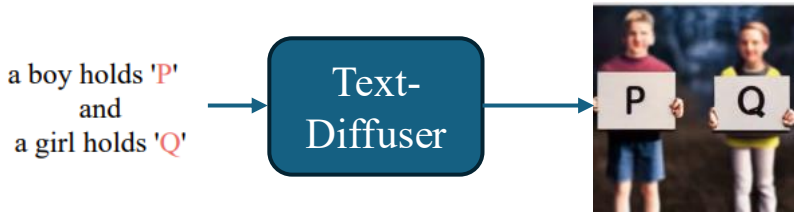
- Stable-Diffusion 3.5 requires 1024&1024 ~1 megapixel to generate small-sized text .
- But finetuning on 512*512 resolution already reaches 80G memory.

Need to apply for a more efficient solution for visual text generation in GUI

Related Works:

(1) GUI generation (UI-Diffuser [1]): text-to-image diffusion model fine-tuned on GUI data

(2) Visual text generation (Text-Diffuser [2]):



(3) Image-image editing (IP2P[3]):

[1] Jialiang Wei, Anne-Lise Courbis, etc, . On ai-inspired ui-design. arXiv 2024
[2] Jingye Chen, Yupan Huang, etc . Textdiffuser-2: Unleashing the power of language models for text rendering. ECCV 2024
[3] Tim Brooks, Aleksander Holynski, etc,. Instructpix2pix: Learning to follow image editing Instructions, CVPR 2023

Related works on GUI generation :

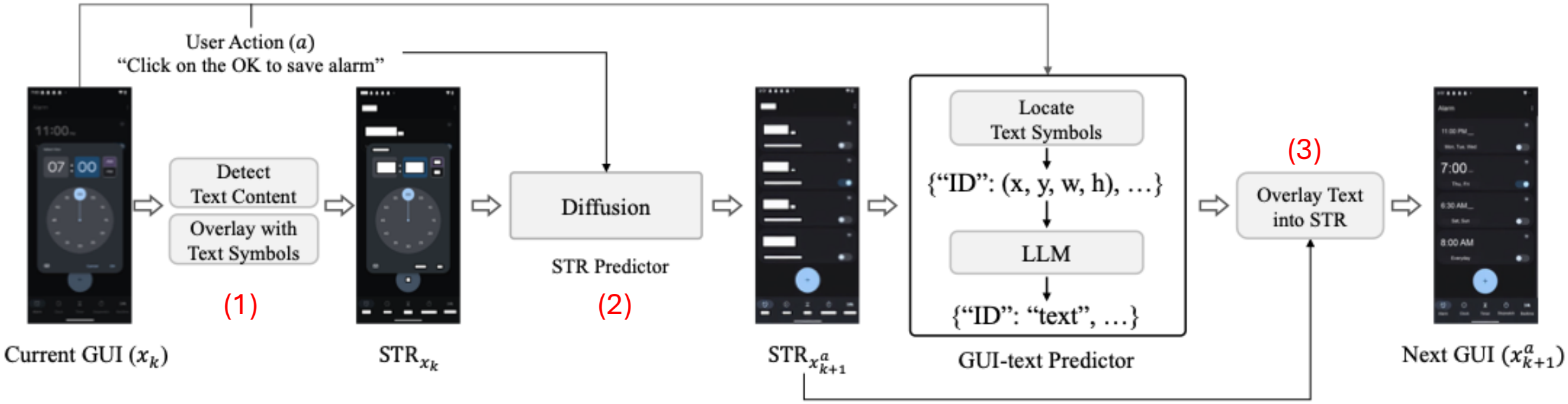
User Action:
Enter the Email as dbwscratch.test.id5@gmail.com

Current GUI | UI-Diffuser [2] | Textdiffuser-2 [3] | IP2P* [4] | ViMo (Ours)

[2][3] they don’t have constains on previous GUI state
[4] generate text in pixel , bring text distortion

Our solution:

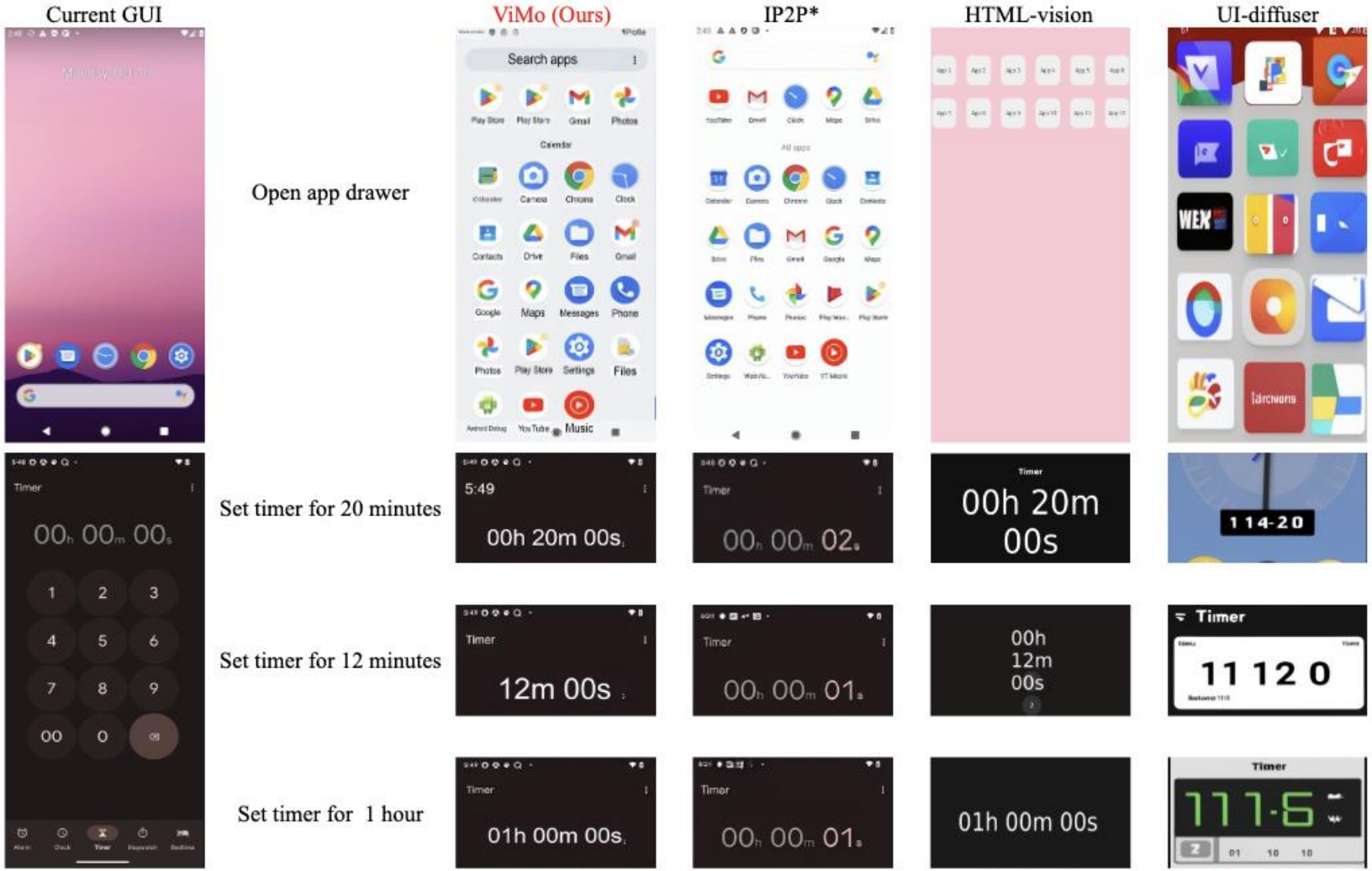
- GUI graphics generation on diffusion
- GUI text generation on LLM



(1) Detect GUI text with OCR, and overlay with a text placeholder
(2) Diffusion to predict GUI graphics (including text placeholder)
(3) LLM predict text for each text placeholder

By generating text from LLM, we can reduce the image resolution to 256*256,16 times more efficient

Visualisation:



Experiment:

GUI quality evaluation:

s_{gc} GUI consistency: ***visual similarity*** between the ground truth and the generated next GUI

s_{ia} Instructional accuracy score: determined whether the ***generated GUI*** adheres ***to the user action***

s_{ar} Action readiness score: whether the ***generated GUI retains valid elements*** essential for subsequent actions required to achieve the user goal

Method	Automatic Metric			
	s_{gc}	s_{ia}	s_{ar}	s_h
HTML-vision	0.70	**85.77**	62.79	0.72
IP2P*	**0.74**	63.57	70.15	0.69
UI-diffuser	0.60	39.61	38.75	0.44
ViMo (Ours)	**0.74**	75.39	**78.68**	**0.76**

Experiment on VIMO-empowered app agents:

(Vimo is used as the world model in slide 3, each app agent is the policy model in slide 3, value function in slide 3 is llm-as-judge)

Step accuracy on offline App agent Dataset

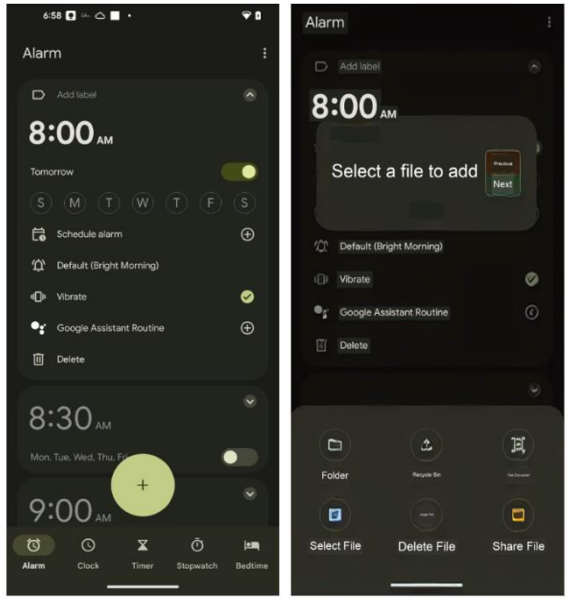
Agent Type	App Agent	Leisure	Work	System	Overall
Language-Based	ER	31.76	46.15	34.13	34.50
	AutoDroid	35.81	46.15	31.75	35.46
	T3A	41.22	51.28	42.86	43.13
	T3A + ViMo (**Ours**)	50.00	**58.97**	**45.24**	49.20
Multi-Modality-Based	APP-Agent	43.24	51.28	39.68	42.81
	Mobile-Agent-v2	43.92	53.85	39.68	43.45
	M3A	46.62	51.28	43.65	46.01
	M3A + ViMo (**Ours**)	**53.38**	53.85	**45.24**	**50.16**

Task accuracy on online App agent Dataset:

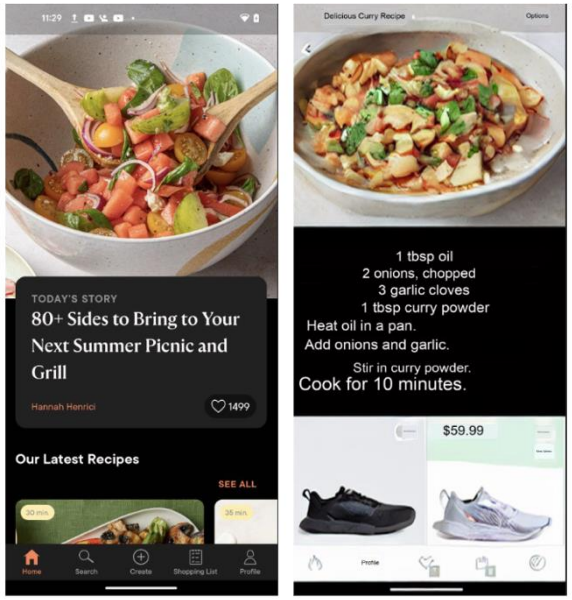
App Agent	LLM	Task Acc.
SeeAct	GPT-4-Turbo	15.50
M3A	GPT-4-Turbo	25.40
M3A	Gemini-1.5-Pro	22.80
T3A	GPT-4-Turbo	30.60
T3A	Gemini-1.5-Pro	19.40
T3A	Gemini-2.0-Flash	33.19
T3A + ViMo	Gemini-2.0-Flash	**40.95**

Discussion:

(a) What's the generalisation, do we want to learn specific apps?



Click the plus icon to add a file



Swipe up to view the curry recipe and some shoes

(Novel combination of app and task, we learn state-transition, not overfitting to specific app screenshots)

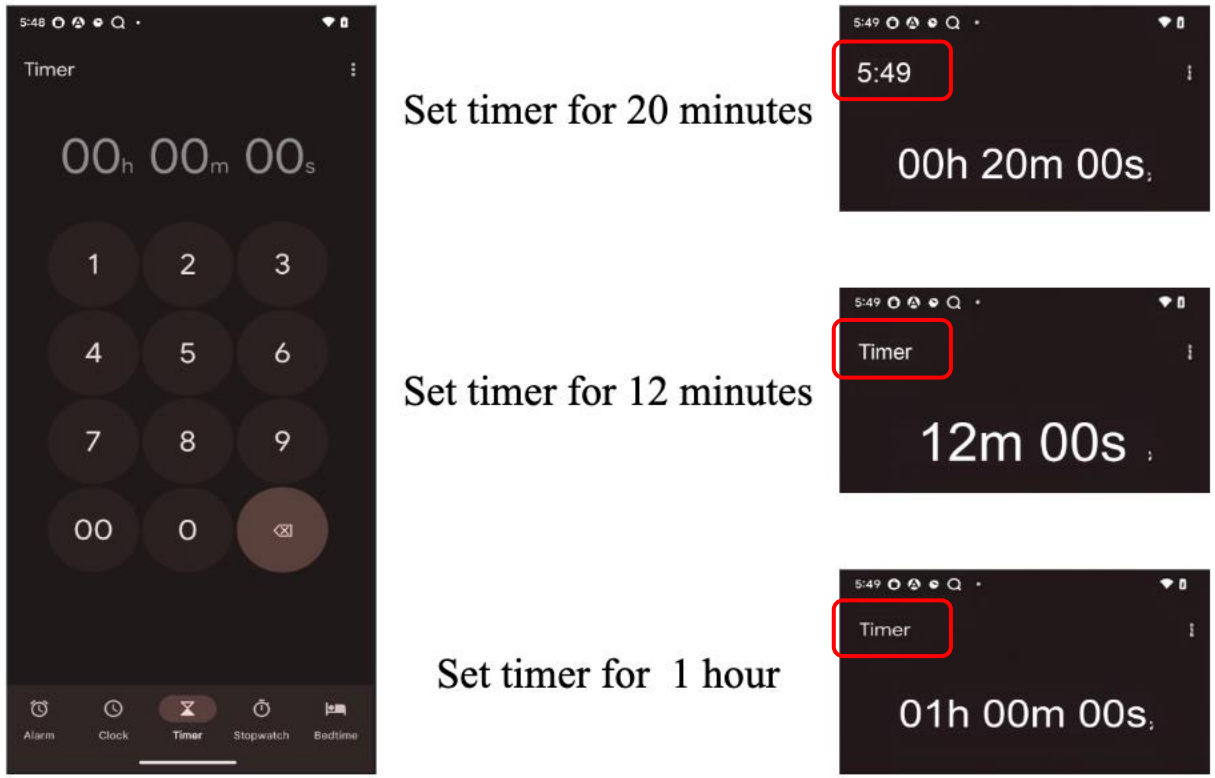
Table 4: Zero-shot Evaluation.

App Agent	LLM	Step Acc.
SeeAct	GPT-4-Turbo	33.9
M3A	GPT-4-Turbo	42.1
ER	Gemini 1.5 Pro	24.4
T3A	Gemini-2.0-Flash	41.4
T3A+ViMo	Gemini-2.0-Flash	46.8
M3A	Gemini-2.0-Flash	44.2
M3A+ViMo	Gemini-2.0-Flash	**47.6**

(Setup: we split our dataset (19 apps) into 16 training apps and 3 testing apps.)

Discussion:

(b) Is randomness a concern?



(The LLM is prompted to generate plausible textual content, and in some cases, multiple reasonable options can be produced. For example, in this figure it shows "5:49 "on the top left corner for "set timer for 20 minutes" command and shows "Timer" for "set timer for 12 minutes", both are plausible and valid in the given context. However, the key functional element, the timer itself, is consistent with the user instructions in all cases.)

Method	Automatic Metric			
	s_{gc}	s_{ia}	s_{ar}	s_h
HTML-vision	0.70	**85.77**	62.79	0.72
IP2P*	**0.74**	63.57	70.15	0.69
UI-diffuser	0.60	39.61	38.75	0.44
r1	0.7421	75.08	78.29	0.7582
r2	0.7323	75.63	77.64	0.7546
r3	0.7423	75.39	78.68	0.7605
STD	0.0057	0.23	0.42	0.0025

- r1-r3 denotes 3 independent runs on ViMo
- STD between the performances is low
- Better than previous methods consistently

(Low STD denotes that Randomness is not a concern)

Visualisation of ViMo in generating GUIs given a single current GUI paired with different actions.

Input

Click the "Add label" text field

Click the checkbox for Monday

Press home button to go to home page

Example of how ViMo helps the App agent to select the correct action.
a) Input
b) Ground truth
c) Action candidates along with the ViMo-predicted future GUI for each candidate. The selected action is highlighted in red.
d) Action selected with out ViMo

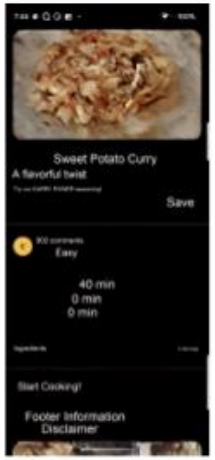
<table>
<tr>
<td>(a) Task Inputs

"After using the Kitchen Stories app for a while and trying many different dishes, I want to set notifications for new recipes because I've been missing the most current updates."

User Goal

Input Image</td>
<td>(c) The potential actions and the correlated predicted GUI of the M3A with ViMo. The selected action is highlighted in red.

{“action_type”: “scroll”, "direction": "down"}

{"action_type": "navigate_back"}

{“action_type”: "click", "index": 18}</td>
</tr>
<tr>
<td>(b) Ground Truth

{"action_type": "navigate_back"}</td>
<td>(d) Output of M3A without ViMo

{“action_type”: “scroll”, "direction": "down"}</td>
</tr>
</table>