💻ShowUI:
One Vision-Language-Action Model for GUI Visual Agent

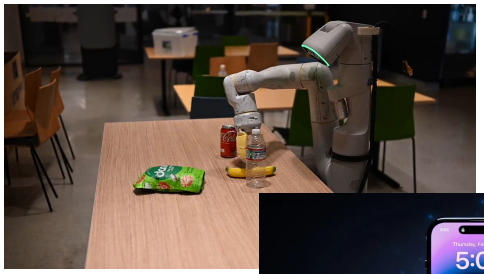
Kevin Qinghong Lin♥, Linjie Li♠, Difei Gao♥, Zhengyuan Yang♠,
Shiwei Wu♥, Zechen Bai♥, Weixian Lei♥, Lijuan Wang♠, Mike Zheng Shou♥
♥Show Lab, National University of Singapore, ♠Microsoft

Multimodal Assistants: from Chatbots to Agents

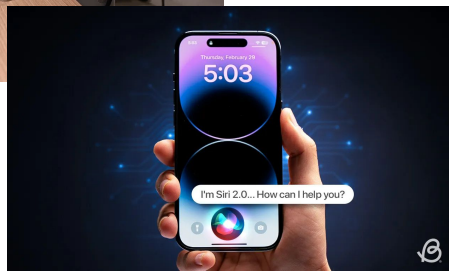
What is unusual about this image?

The image shows a man ironing clothes on an ironing board placed atop a taxi, ...

Chatbot – Perception

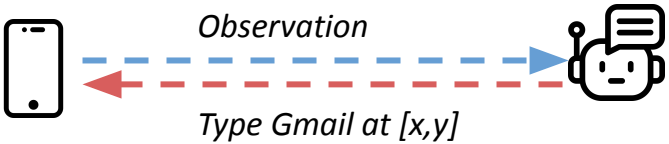


Physical
e.g., Robotic



Digital
e.g., GUI

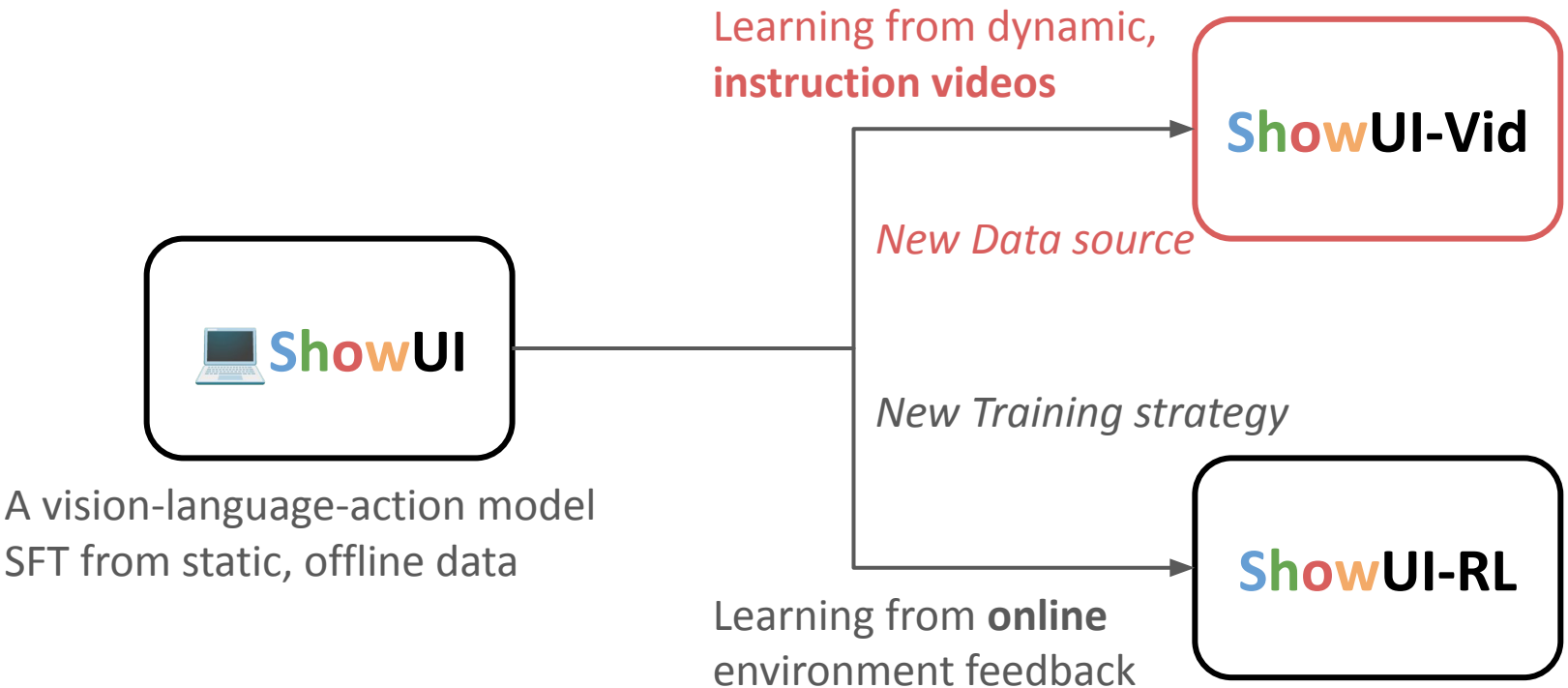
Please help me send a message to Kevin?



Agent – Action

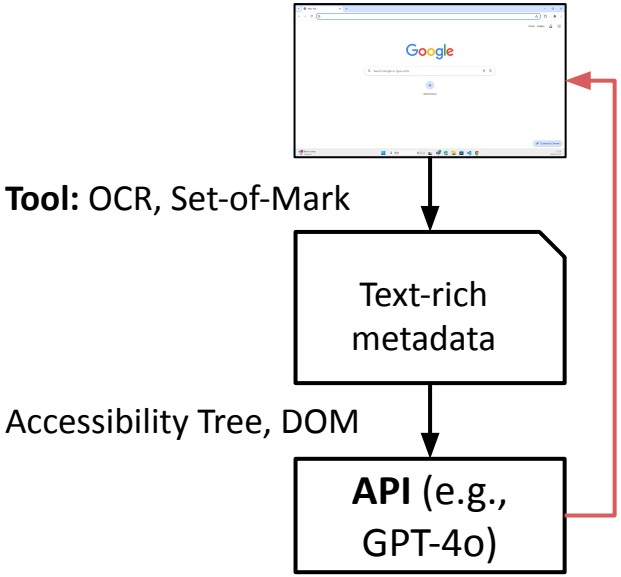
Figures credit to GPT-4V, RT-2, Siri

Project Overview

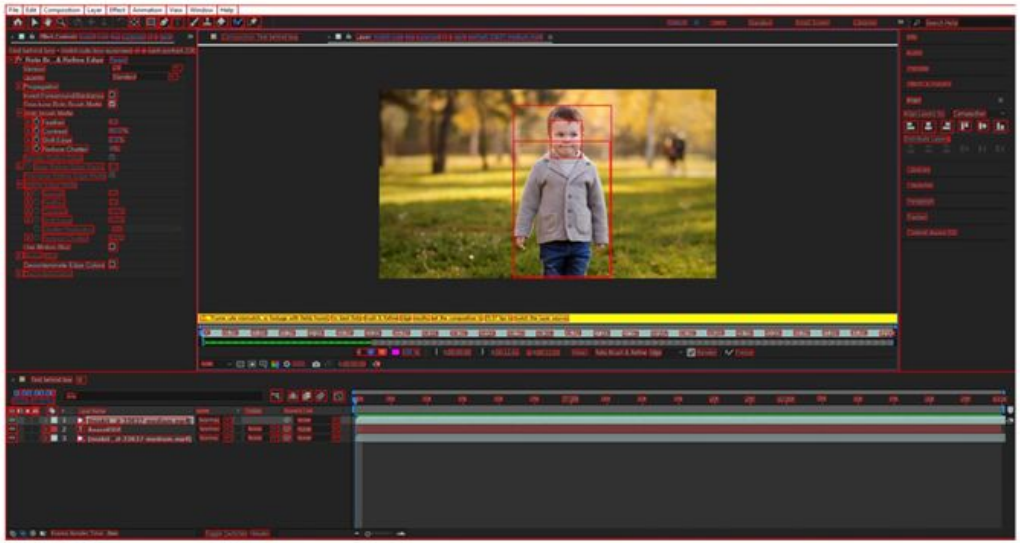


Language Agent v.s. Visual Agent

- **Language Agent: API + Tools**



- Rely on API + Tools, expensive
- AssistGUI (CVPR'24), MindAct (ICML'24)



panel:
name:
Effect Controls
rectangle: [0, 68, 363, 721]
elements:
Effect Controls [100, 82], mixkit [155, 82], cute [184, 82], boy [208, 82], surprised [243, 83], in [274, 83], a [286, 83], park [304, 83]
Text behind boy [49, 109], mixkit cute boy surprised [172, 109], in [250, 109], a [262, 109], park portrait 336 [315, 109]
expand_icon [12, 126], fx Roto Br ... & Refine Edge [101, 126], Reset [214, 127]
Version [54, 144], 20 [210, 142], expand_icon [326, 142]
expand_icon [318, 159]
...

Figures credit to AssistGUI

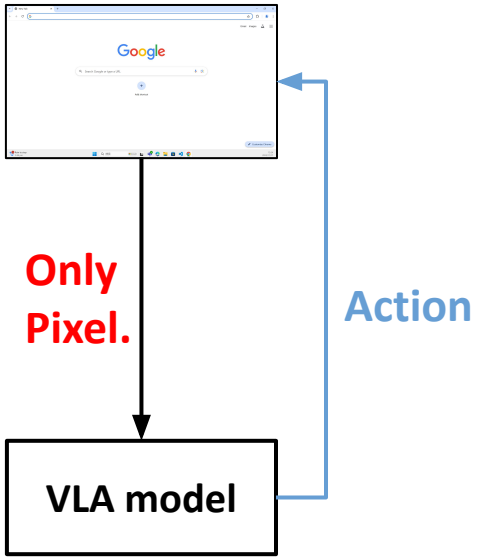
Language Agent v.s. Visual Agent

See and Act Like Humans?

*Building **One End2end***

***Vision-Language-Action Model** in Digital World.*

- **Visual Agent**



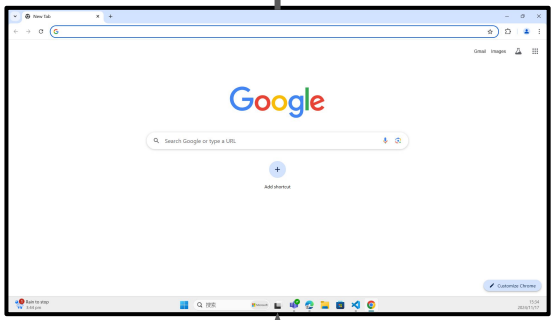
- Training one purely vision input model
- CogAgent (CVPR'24), SeeClick (ACL'24)

Figures credit to kslnewsradio

Challenges by Developing GUI Visual-centric Agent

Perception

Q1: How to model UI visually?



VLA model

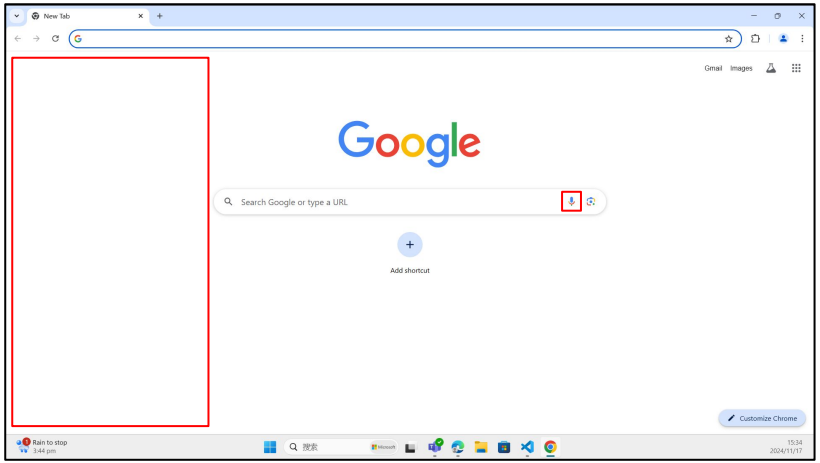
Q3: How to prepare training data?

Action

Q2: How to model Action (with visual)?

Q1: What differentiates UI from natural vision?

V.S.

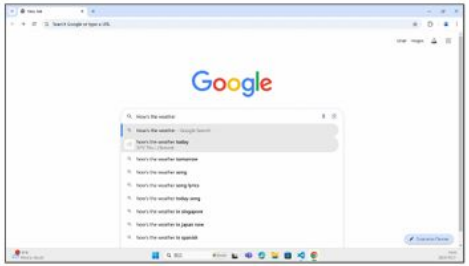


- Unpredictable patterns
- Rich in ***Semantic***

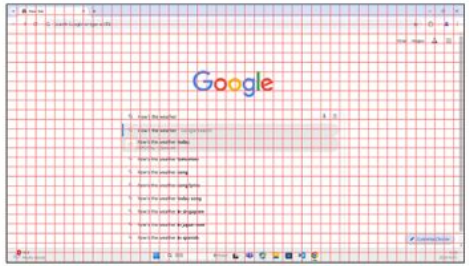
- Structured layout
- ***Key Elements*** with ***Redundancy***

Q1: What differentiates UI from natural vision?

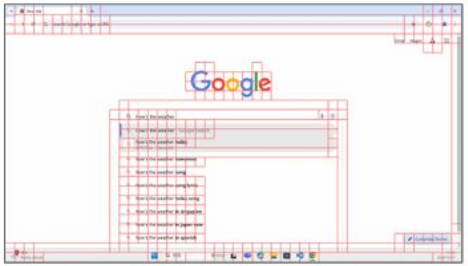
- Building *Patch-wise* UI Graph on *RGB* space



Screenshot
1344 x 756

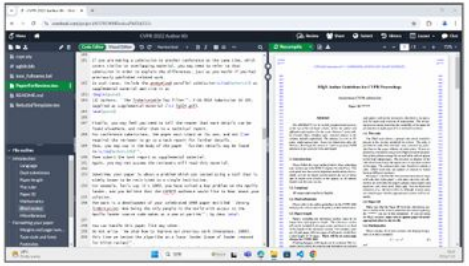


Patchified (28 x 28)
#1296 Tokens

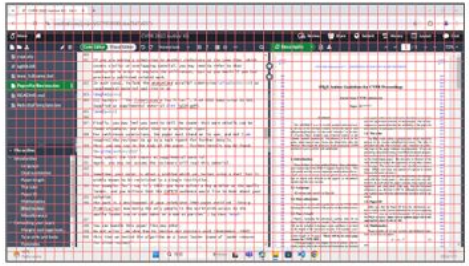


UI Connected Graph
#291 Components

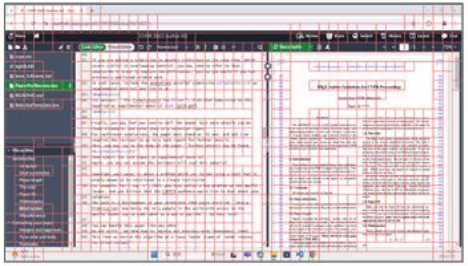
Example1: Google Search



Screenshot
1344 x 756



Patchified (28 x 28)
#1296 Tokens



UI Connected Graph
#986 Components

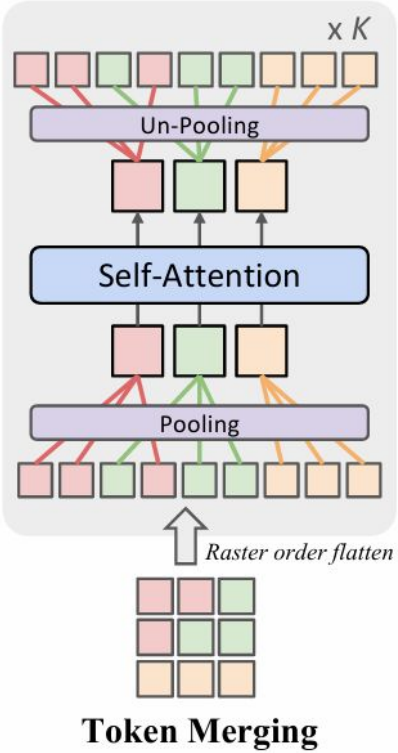
Example2: Overleaf Template

Q1: How to model visual redundancy?

- Computation bottleneck
  - self-attention block O(L^2)
- **Token Merging** by UI-Graph?
  - Lose position relationship!
  - Harm UI Grounding

Method	Strategy	#Vis.Ctx.	Train.Speedup	Test-time?	Screenspot
Baseline	N/A	1344.0	1×	N/A	70.8
Token Merge	UI-Graph	852.8	1.6×	✓	42.3 34.7
Token Selection	Random	941.5	1.5×	✓	65.3 56.2
	UI-Graph	947.4	1.5×	✓	**70.4** 64.9

(a) Comparison between different visual tokens compression methods. '#Vis.ctx' represents the avg. length of visual tokens across all layers. 'Train.Speedup' denotes the training efficiency improvement over the baseline. 'Inference' denotes whether this method is applicable at test time.



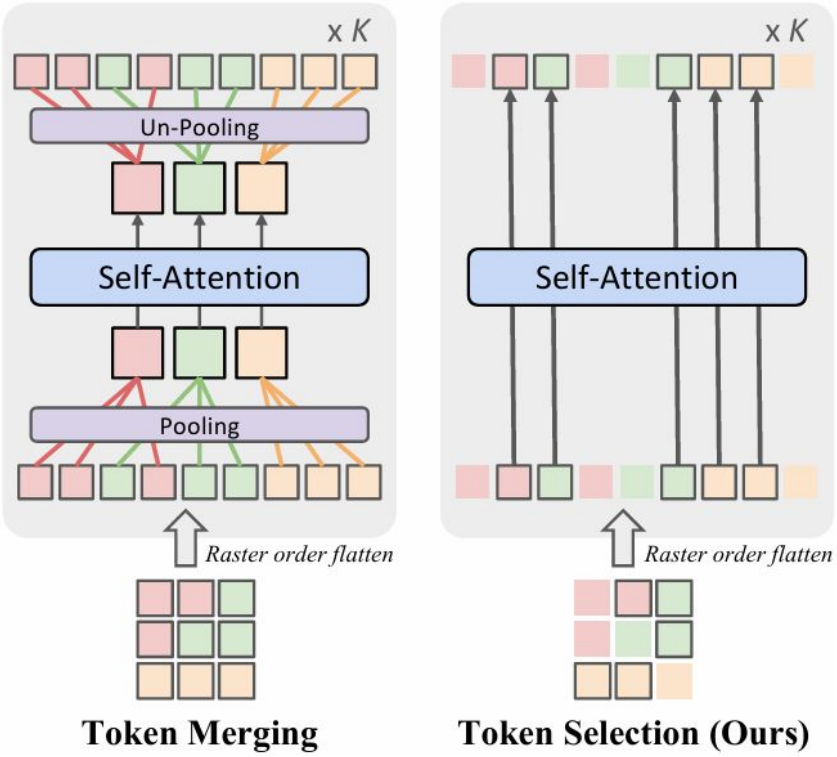
Token Merging

Q1: How to model visual redundancy?

- Computation bottleneck
  - Self-attention block O(L^2)
- ❌Token Merging by UI-Graph
- ✅**Token Selection** by UI-Graph
  - Maintain original position relationship

Method	Strategy	#Vis.Ctx.	Train.Speedup	Test-time?	Screenspot
Baseline	N/A	1344.0	1×	N/A	70.8
Token Merge	UI-Graph	852.8	1.6×	✓	42.3 34.7
Token Selection	Random	941.5	1.5×	✓	65.3 56.2
	UI-Graph	947.4	1.5×	✓	**70.4** 64.9

(a) Comparison between different visual tokens compression methods. '#Vis.ctx' represents the avg. length of visual tokens across all layers. 'Train.Speedup' denotes the training efficiency improvement over the baseline. 'Inference' denotes whether this method is applicable at test time.



Q2: What differentiates Action from natural text?

- Action is a **tuple**
  - TYPE: CLICK / TYPE
  - Position: [x,y] for CLICK
  - Value: string for TYPE, direction for SCROLL
- Actions are **diverse** across
  1. Device: CLICK on PC, PRESS_HOME on Mobile.
  2. Parameter: SCROLL {up/down} on PC, {up/down/left/right} on Mobile
  3. Novel Unseen: PASTE

Q2: What differentiates Action from natural text?

- Action is a **tuple**
  - TYPE: CLICK / TYPE
  - Position: [x,y] for CLICK
  - Value: string for TYPE, direction for SCROLL

Structure Action as JSON

{'action': 'action type',
'value': 'element',
'position': [x,y]}

- Actions are **diverse** across
  1. Device: CLICK on PC, PRESS_H
  2. Parameter: SCROLL {up/down}
  3. Novel Unseen: PASTE

Document Action space by README

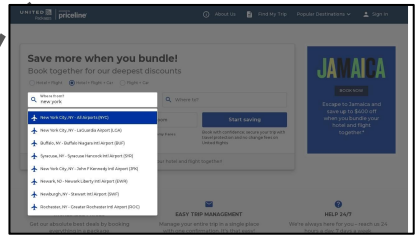
You are an assistant trained to navigate the {device}. Given a task instruction, a screen observation, and an action history sequence, output the next action and wait for the next observation.

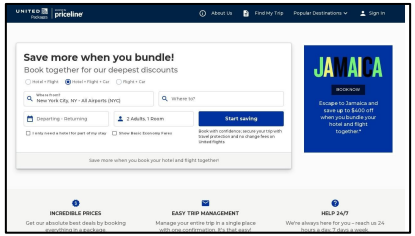
Here is the action space:
 # templated by action_type with action description.
1. 'CLICK': Click on an element, value is the element to click and the position [x,y] is required.
2. 'TYPE': Type a string into an element, value is the string to type and the position [x,y] is not applicable.
...

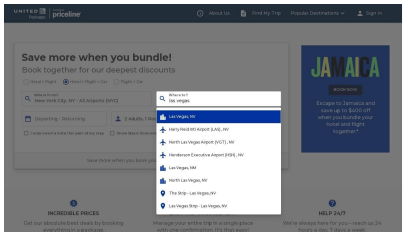
Format the action as a dictionary with the following keys:
{'action':'action_type', 'value':'element', 'position':[x,y]}
Position represents the relative coordinates on the screenshot and should be scaled to a range of 0-1.

Q2: How to model Action with visual?

- UI navigation conditioned on *HISTORY*
  - Past Action Sequence
  - Past Observation








```
{'action':'CLICK',
'value':
'New York City, NY -
All Airports (NYC)',
'position':
[0.21,0.50]}
```

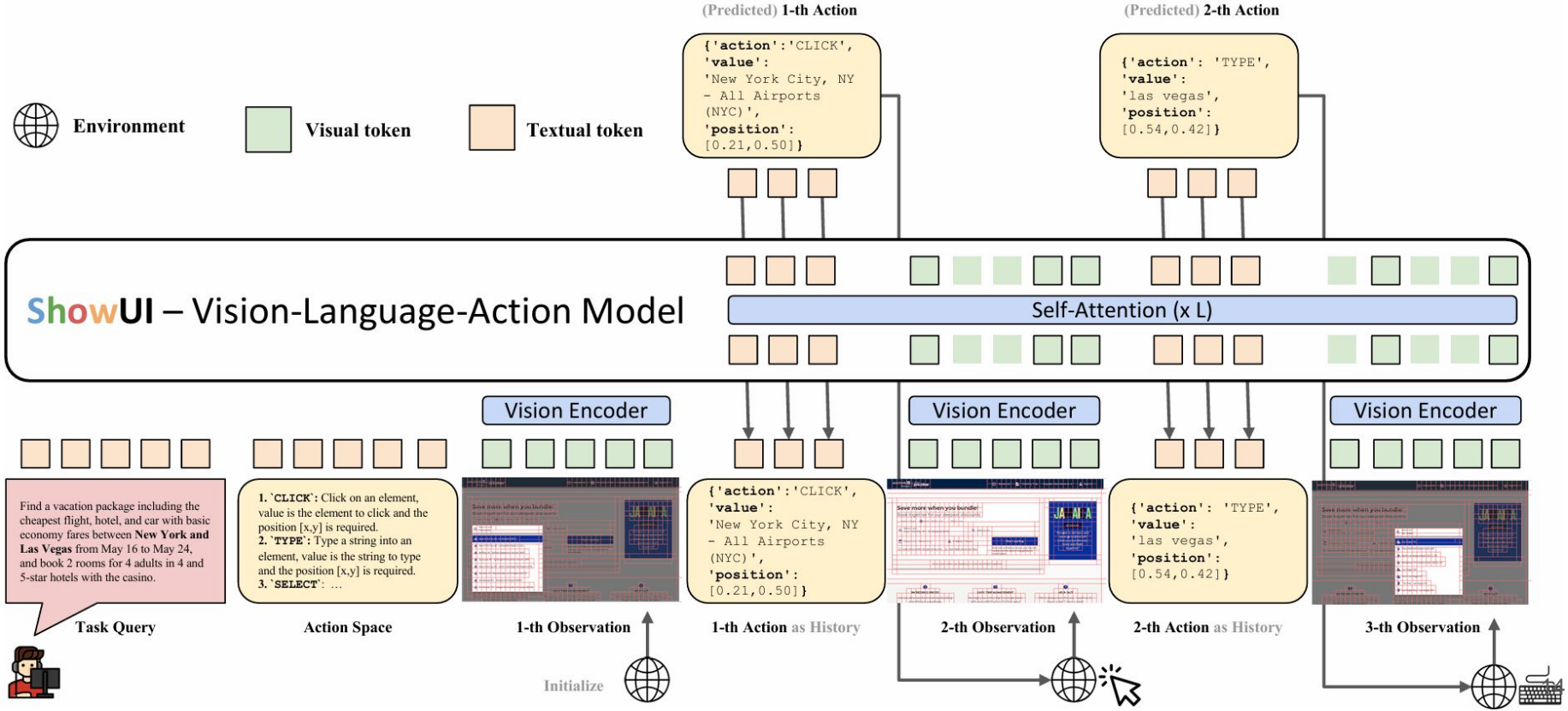
```
{'action': 'TYPE',
'value':
'las vegas',
'position':
[0.54,0.42]}
```

?

HISTORY

Q2: How to manage Action with other modalities?

- Interleaved Vision-Language-Action Streaming



Q2: Gains by Interleaved VLA Streaming

- Improve Training data utilization

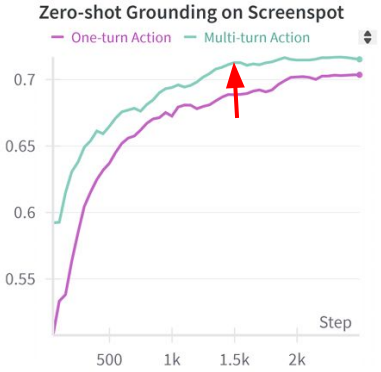


Figure 10. Impact by Interleaved action-query streaming on Grounding task: trained with 119K grounding data, Eval with Screenspot.

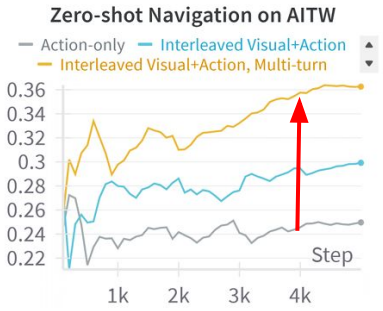


Figure 11. Impact by Interleaved action-visual streaming on Navigation task: trained with GUIAct, Eval with AITW.

History by	History Len.	AITW Avg.
N/A	0	66.8
[Action]	1	67.9
[Action]	2	68.5
[Action]	4	68.6
[Visual][Action]	1	69.4
[Visual][Action]	2	**70.0**

Table 2. Ablation studies on GUI navigation history modeling, including history composition and history length.

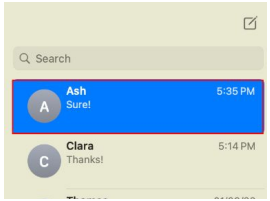
Q3: How to Prepare the Training data?

- Massive Data v.s. A well-selected small training set
- Augment original UI Annotations by GPT-4o
- Data rebalancing is important

Usage	Device	Source	#Sample	#Ele.	#Cls. (len.)	Highlights
Grounding	Web	Self-collected	22K	576K	N/A	Visual-based
	Mobile	AMEX [8]	97K	926K	N/A	Functionality
	Desktop	OmniAct [22]	100	8K	N/A	Diverse query
Navigation	Web	GUIAct [10]	72K	569K	9 (7.9)	One / Multi-step
	Mobile	GUIAct [10]	65K	585K	5 (9.0)	Multi-step
Total	Diverse		256K	2.7M		

Table 1. Overview of our instruction-tuning data. **#Sample** indicates the number of the task instance (screenshot in grounding, task in navigation); **#Ele.** indicates the number of the element (*i.e.*, bbox in grounding); **#Cls.** represents the number of action classes, and **len.** indicates the average trajectory length per task.

Original anno: message_ash



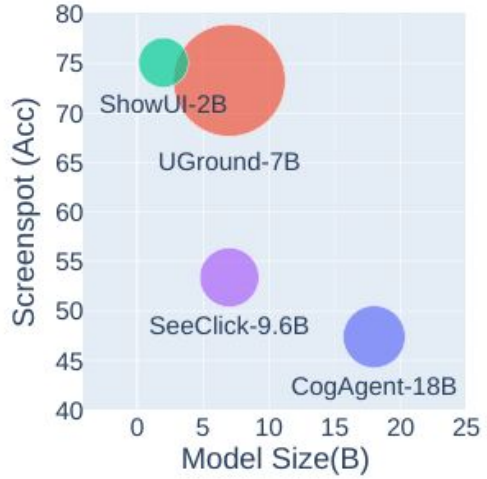
Query by GPT-4o:

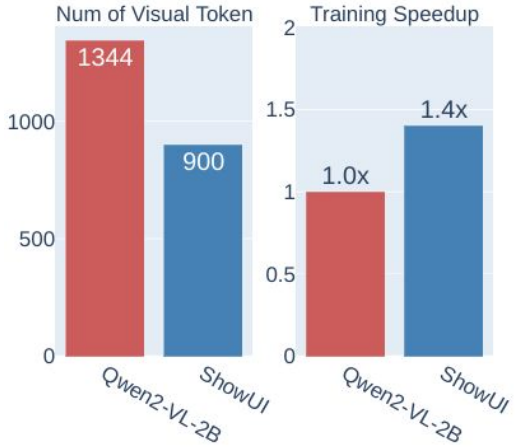
Appearance: A blue chat card contains a capital letter A.

Spatial-relationship: Located above Clara's chat box.

Intention: Send a picture to Ash.

ShowUI – a Lightweight (2B) Vision-Language-Action Model





Method	Size	#Train	Mobile Text	Mobile Icon	Desktop Text	Desktop Icon	Web Text	Web Icon	**Avg.**
Qwen2-VL-2B [41]	2B	–	24.2	10.0	1.4	9.3	8.7	2.4	9.3
Fuyu [5]	8B	–	41.0	1.3	33.0	3.6	33.9	4.4	19.5
CogAgent [17]	18B	400K	67.0	24.0	74.2	20.0	70.4	28.6	47.4
SeeClick [11]	9.6B	364K	78.0	52.0	72.2	30.0	55.7	32.5	53.4
OmniParser [31]	*	–	93.9	57.0	91.3	63.6	81.3	51.0	73.0
UGround [15]	7B	1.3M	82.8	60.3	82.5	63.6	80.4	70.4	73.3
ShowUI-G	2B	119K	91.6	69.0	81.8	59.0	83.0	65.5	74.9
ShowUI	2B	256K	92.3	75.5	76.3	61.1	81.7	63.6	**75.1**

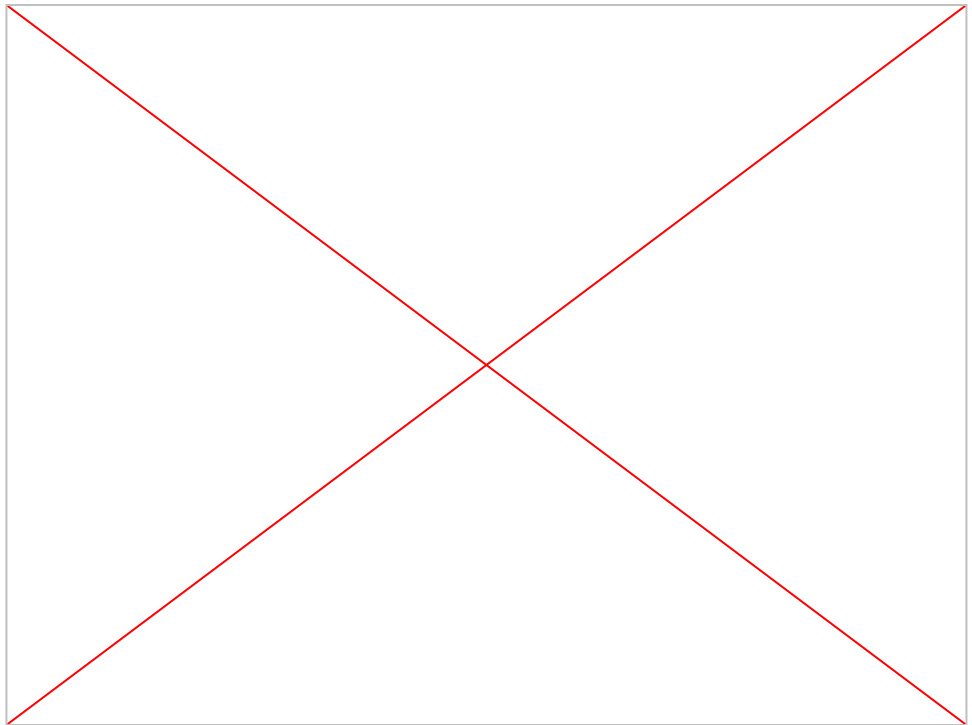
Table 2. Zero-shot grounding on Screenspot. * means Omniparser use

Method	FT?	General	Install	G.Apps	Single	WebShop.	**Overall**
ChatGPT-CoT [53]	–	5.9	4.4	10.5	9.4	8.4	7.7
PaLM2-CoT [37]	–	–	–	–	–	–	39.6
OmniParser [31]	*	48.3	57.8	51.6	77.4	52.9	57.7
SeeClick [11]	✓	54.0	66.4	54.9	63.5	57.6	59.3
Qwen2-VL-2B [41]	✓	61.4	71.8	62.6	73.7	66.7	67.2
ShowUI †	✓	63.5	72.3	66.0	72.3	65.8	68.3
ShowUI	✓	63.9	72.5	69.7	77.5	66.6	**70.0**
ShowUI-ZS	✗	32.1	47.7	42.0	20.1	37.4	35.9

Table 3. Performance of Mobile Navigation [36], where gray color in-

Open-source ShowUI for Local Computer Use

- 🤗Model&Demo are live: https://github.com/showlab/ShowUI (10K+ download)
- 🖥️Easy Computer Use Solution: https://github.com/showlab/computer_use_ootb



Thanks for Listening!

https://github.com/showlab/ShowUI