Deep Learning Project

PIGG: Personalized Interactive GUI Grounding

CSE 20201118 Jeonghoon Park
CSE 20201290 Seungmin Cho

CSE403: Deep Learning
Fall Semester, 2025, UNIST

Contents

Project Workflow

Problem Definition

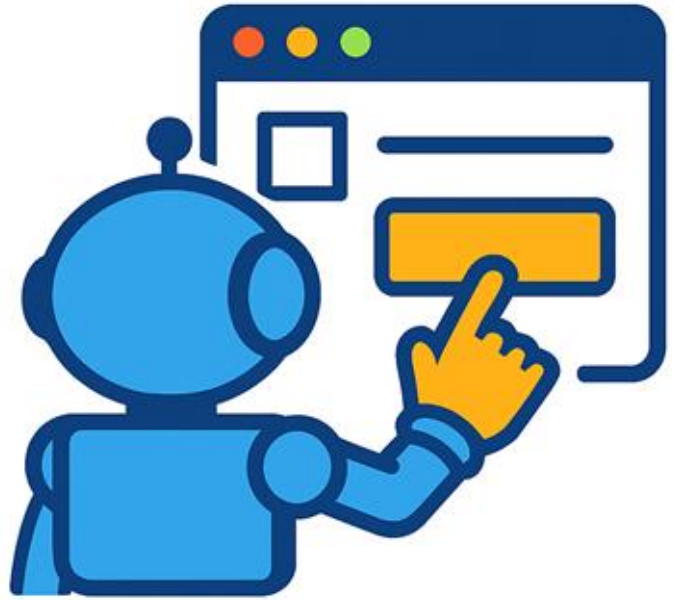
What is GUI Grounding?

Problem Definition

- Definition: the task that enables an AI to **see** Graphical User Interfaces (GUIs), **understand** the function of visual elements, and **link** them to abstract language commands
  - Moves beyond simple object detection to functional understanding
  - Mapping abstract instructions (e.g. booking hair salon) to UI elements

- Core Technologies: A fusion of Computer Vision (Object Detection, OCR) and Multimodal Language Model

- Goal: Create AI that can **control** any application just like us



Limitation of Current Models

Problem Definition

- The app universe is **infinite**
  - In 2024, Google Play store has **1.6M+ apps**
  - Impossible to train a single agent for every app in world

- "Personal Gap" in Generic Models
  - Current models are trained on common generic UIs
  - Fail at indie app, customizing, or corporate software

- Adapting to the Personalized Environment
  - Average korean user has only **~102 apps**
  - requires **fine-tuning** the agent on each individual user

Google Play Store: number of apps 2024| Statista
Average korean user has only... | donga.com

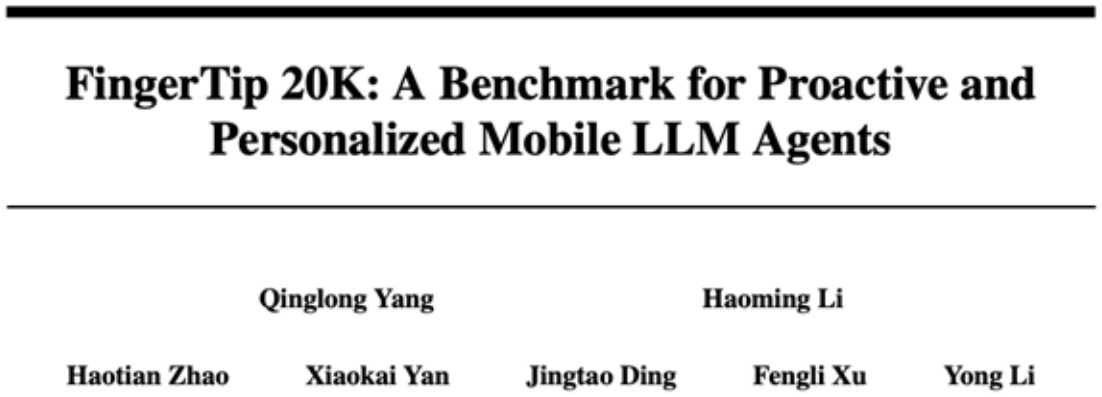
Datasets Used

FingerTip 20K

Datasets Used

- The only user-labeled mobile GUI interaction dataset, collected from 91 real users across 506 apps over long-term daily app usage (29GB)

- Each sample includes user profile, intent, scenario, and action logs, making it ideal for personalized GUI grounding and action modeling



FingerTip 20K: A Benchmark for Proactive and Personalized Mobile LLM Agents

Qinglong Yang Haoming Li

Haotian Zhao Xiaokai Yan Jingtao Ding Fengli Xu Yong Li

Directory Structure

Datasets Used

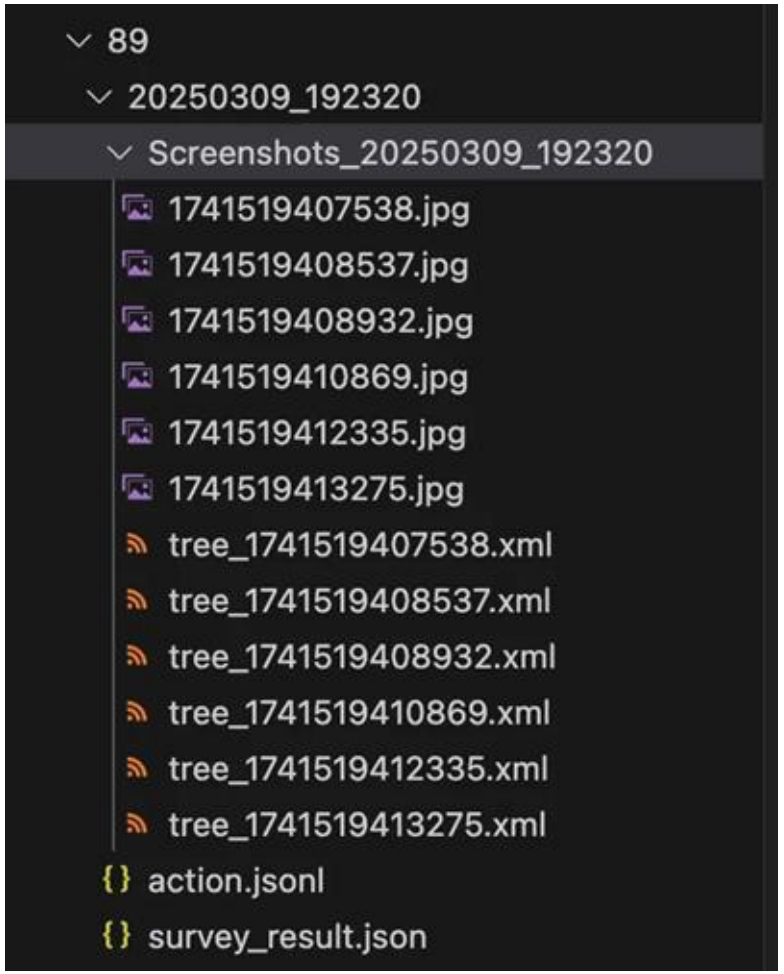
→ User id

→ Timestamp

→ Screenshots

→ Action

→ Survey_result



Screenshots_[TIMESTAMP]

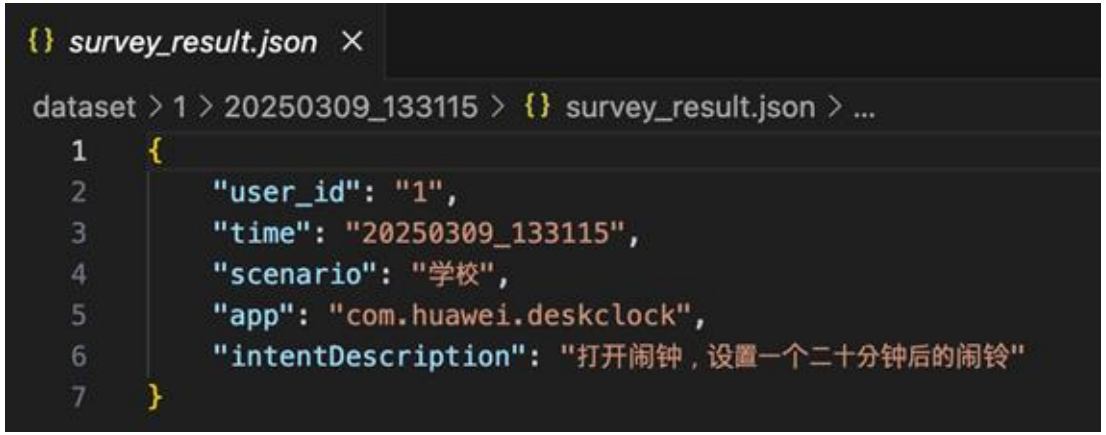
Datasets Used

survey_result.json

Datasets Used

This file stores metadata describing the user's **GUI interaction scenario**

user_id	the identifier of the user participating in the data collection
time	the timestamp of the recorded session
scenario	the context or situation (e.g., school, home, office)
app	the package name of the application being used
intentDescription	**a natural-language description of the user's intended action or goal**



tree_[TIMESTAMP].xml

Datasets Used

This file provides **the View Hierarchy** corresponding to each screenshot in XML

class	the type of UI element
text	the visible text or description
bound	The exact (x, y) coordinates of the element
boolean properties	True/false flags, defining the current functional state of a UI element

```
<node
  class="android.widget.FrameLayout"
  text=""
  content_desc=""
  view_id=""
  bounds="[0,0][1080,2340]"
  hint=""
  tooltip=""
  checked="false"
  enabled="true"
  focused="false"
  selected="false"
  clickable="false"
  long_clickable="false"
  checkable="false"
  focusable="false"
  editable="false"
  scrollable="false"
  password="false"
  visible="true">
```

action.jsonl

Datasets Used

This file is a log file **recording a sequence of user actions** performed on the GUI

action	“click()”, “scroll()”, “press_back()”
coordinates	indicate the click or scroll position
content	describes the UI element or text at that location
direction	specifies the scroll direction (if applicable)

In our experiment

label : target click point (x, y) for each user instruction.
input feature : Screenshot + Natural language instruction
output feature : Element description (referring) + Click coordinates (grounding)

Appropriate Deep Learning Algorithms

Base Deep Learning Architecture: Qwen3-VL

Appropriate Deep Learning Algorithms

- **Qwen3-VL-8B-Instruct**
  - VL transformer model with visual encoder + LLM decoder through cross-modal attention

- Suitable for the **GUI grounding task**
  - Gives both **visual grounding** and **semantic reasoning**
  - Jointly learn to interpret **textual instructions** and **visual layouts**
  - Enabling coordinate prediction based on multimodal reasoning
  - Capture contextual cues from both GUI and instruction, unlike CNN-based detectors

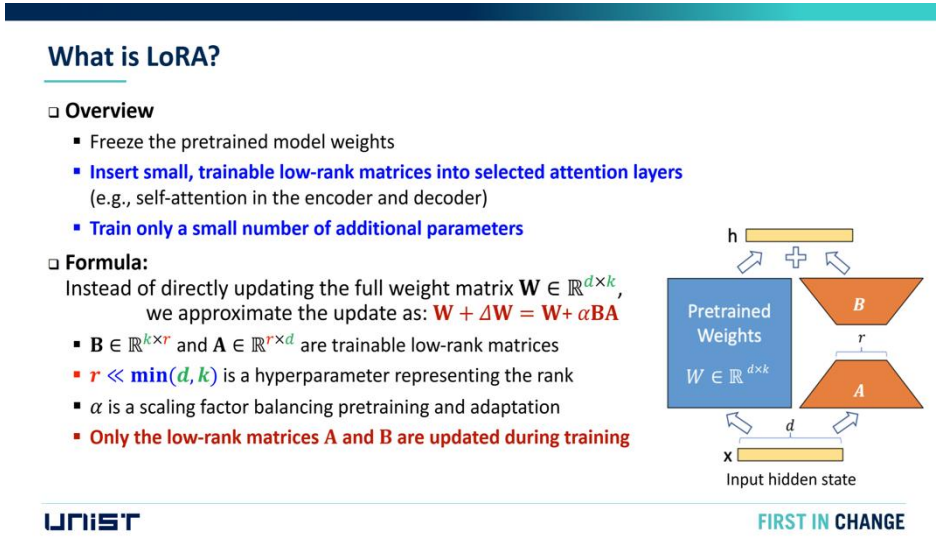
Fine-Tuning Strategy: LoRA (Low-Rank Adaptation)

Appropriate Deep Learning Algorithms

- **LoRA (Low-Rank Adaptation)**
  - Insert small, trainable low-rank matrices into selected attention layers

- Suitable for the **personalization**
  - Reduce memory and computation costs
  - Preserve the representation power of the base model
  - Efficiently provides personalization without full retraining

From Lecture of Prof. Yeon-Chang Lee



Experiments and Results

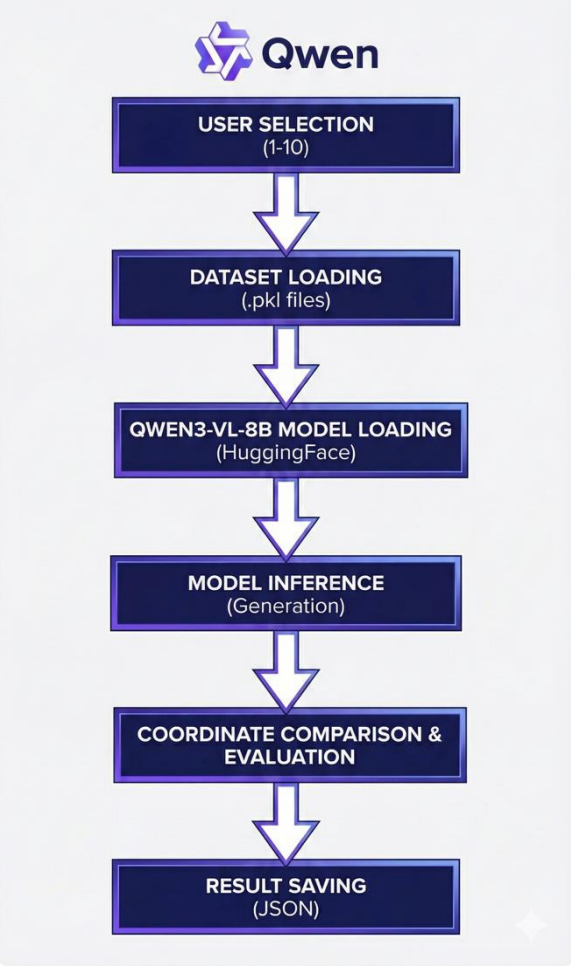
Experiment Setup & Evaluation

Experiments and Results

Backbone	Qwen3-VL-8B-Instruct (Vision-Language Model)
Tuning Method	Full FT & LoRA (r = 8, α = 32)
Dataset Size	90727 samples (73 users) for training, 10063 samples (10 users) for test
Task	GUI Grounding (Point Prediction)
Dataset	FingerTip 20K
Metrics	Click Accuracy @14%, Mean/Median L2 Error
Frameworks & Libraries	PyTorch (with CUDA 12.1) Transformers (Hugging Face) PEFT (Parameter-Efficient Fine-Tuning for LoRA)
Hardware	4 NVIDIA A100 80GB

Experiment Method: Vanilla Qwen3-VL

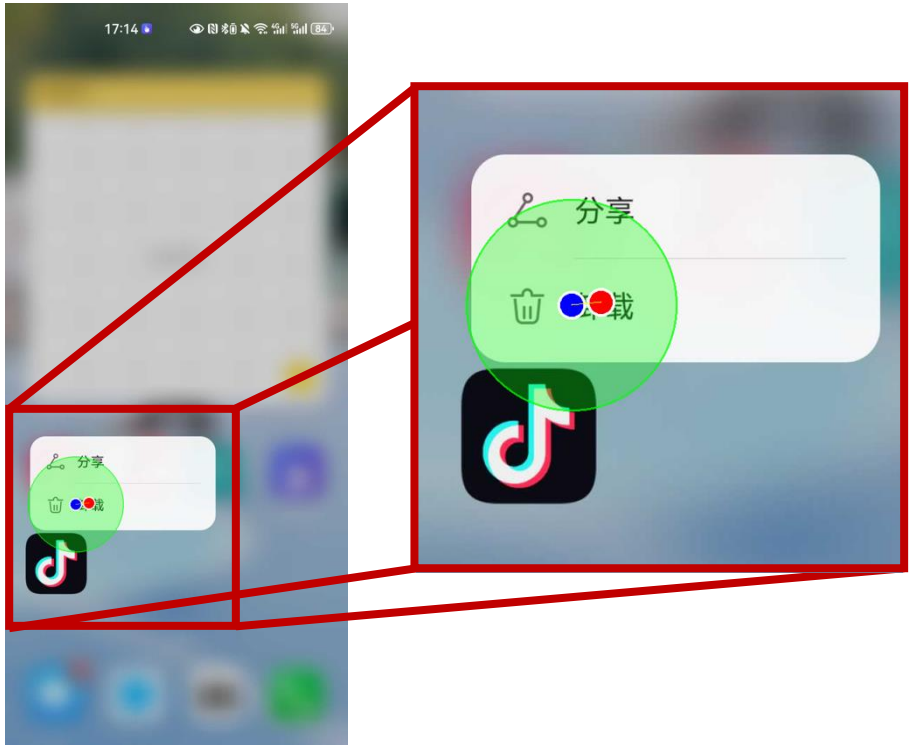
Experiments and Results

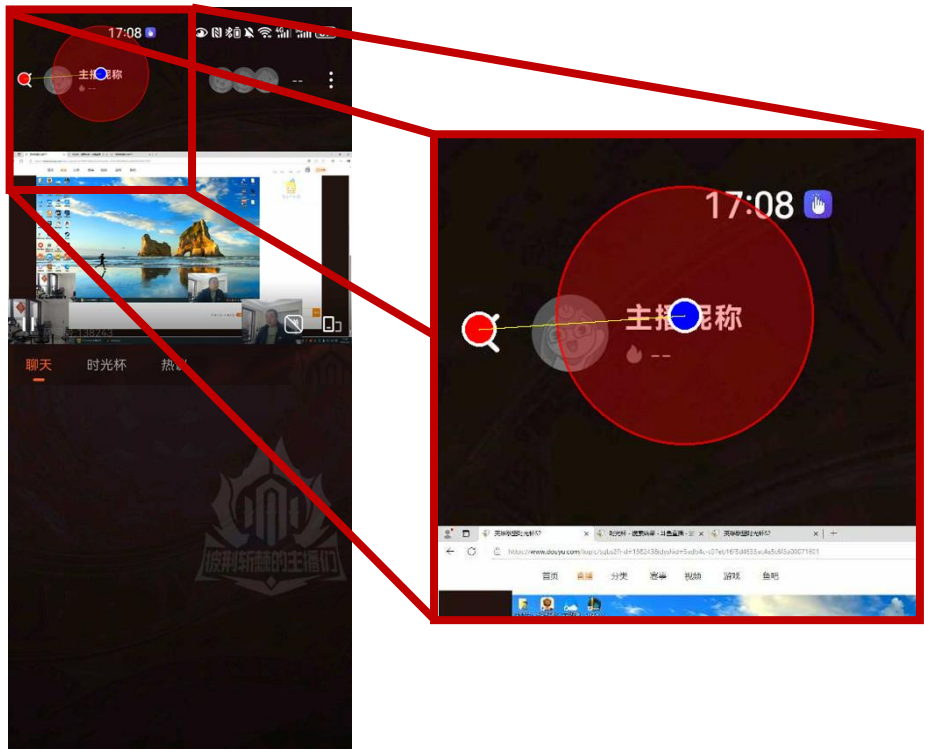


- **Overview**
  - Model: Qwen3-VL-8B-Instruct
  - Goal: Zero-shot baseline for Text-to-Coordinate prediction (No Fine-tuning)
- **Dataset & Setup**
  - Data: 10,063 samples (10 users)
  - Task: Image + Command (Text) → Referring + Grounding (Coordinates)
- **Pipeline**
  - Process: Resize (1280×28×28) → Inference → Regex Parsing
  - Output: Target description & [x, y] extraction
- **Metrics**
  - Click Accuracy: Success if L2 distance ≤ 14% tolerance
  - L2 Error: Mean / Median distance

Qualitative Examples

Experiments and Results



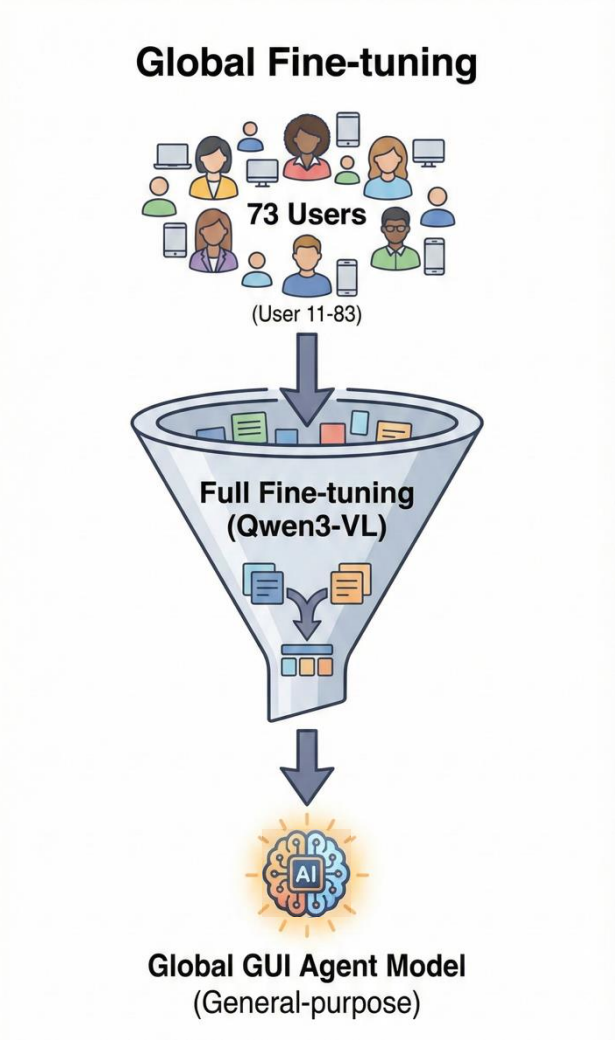


Case	Instruction	Instruction (Eng-Translated)	Pred (x,y)	GT (x,y)	Result
#1	卸载抖音app	Uninstall the TikTok app	(264, 1577)	(222, 1581)	O
#2	打开斗鱼，进入“洞主”直播间	Open Douyu and enter the “Cave Master” livestream	(49, 226)	(290, 210)	X

Experiment Method: Global Fine-Tuning (Full)

Experiments and Results

- **Overview**
  - Goal: Build a GUI Agent & Base for Personalization
  - Base Model: Initialized from Qwen3-VL-8B-Instruct
  - Data: 90,727 samples from 73 users (User 11-83)
- **Methodology: Full Fine-tuning**
  - Approach: Updates ~99% of parameters (Entire LLM) for best performance
  - Frozen Components: Vision Encoder & Projector preserved
- **Training Strategy**
  - Label Masking: Assistant-only training (Ignore user prompt tokens)
  - Target: Learn both Referring (Description) & Grounding (Coordinates)
  - Efficiency: Gradient Checkpointing & BFloat16 Mixed Precision
  - Config: 1 Epoch, Learning Rate $1e^{-6}$, Batch size 1 (Grad Accum 8)



Vanilla vs Global FT

Experiments and Results

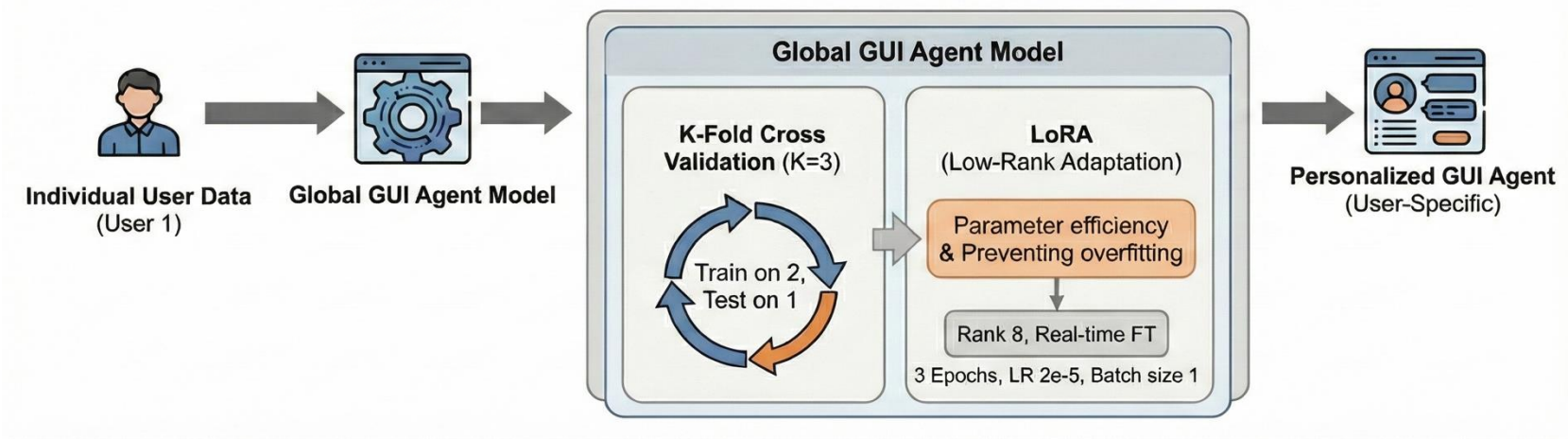
Table 1: Comparison of Vanilla vs. Global Fine-tuned Accuracy at 14% Click Budget

User	Vanilla Acc. (%)	Global FT Acc. (%)	$\Delta\text{Acc}_{(\text{Van}\rightarrow\text{Glob})}$. (%)
1	36.19	45.90	26.80
2	17.32	22.52	29.96
3	22.10	28.88	30.65
4	26.37	38.66	46.59
5	29.75	33.92	13.97
6	29.51	36.72	24.44
7	14.76	24.24	64.29
8	27.06	38.82	43.48
9	27.17	34.53	27.08
10	30.08	37.99	26.28
Avg.	26.03	34.22	31.44

Experiment Method: Personalized Fine-Tuning (LoRA)

Experiments and Results

- **Overview**
  - Goal: Optimize GUI Agent for individual users
  - Base Model: Global FT
- **K-Fold Cross Validation (K=3)**
  - Why: To obtain a stable and less biased validation estimate under small per-user datasets
  - Process: Split data into 3 folds → Train on 2, Test on 1 (Iterate 3 times)
- **LoRA (Low-Rank Adaptation)**
  - Parameter efficiency (Rank 8)
  - Config: 3 Epochs, Learning Rate $2e^{-5}$, Batch size 1 (Grad Accum 4)



Global FT vs Personalized FT

Experiments and Results

Table 2: Comparison of Global FT vs. Personalized FT at 14% Click Budget

User	Global FT Acc. (%)	Personalized FT Acc. (%)	ΔAcc$_{(\text{Glob}\rightarrow\text{Per})}$ (%)
1	45.90	51.09	11.32
2	22.52	38.86	72.56
3	28.88	38.68	33.95
4	38.66	51.13	32.28
5	33.92	42.23	24.51
6	36.72	38.69	5.37
7	24.24	37.30	53.85
8	38.82	40.39	4.04
9	34.53	46.44	34.51
10	37.99	59.36	56.26
Avg.	34.22	44.42	29.81

Overall Performance Summary

Experiments and Results

Table 3: Overall Performance Comparison Across Model Types

Model Type	**Click Acc. @14% ↑**	**Mean L2 ↓**	**Median L2 ↓**
Vanilla	26.03	373.39	338.25
Global	34.22	322.69	290.15
Global + Personalized	44.42	260.50	185.29

Effect of Exclusive App

Experiments and Results

Table 4: Performance Improvements and User-specific Data Proportion

User	$\Delta\text{Acc}_{(\text{Van}\rightarrow\text{Glob})}$ (%)	$\Delta\text{Acc}_{(\text{Glob}\rightarrow\text{Per})}$ (%)	Exclusive App Data Ratio (%)	#Samples
1	+26.80	+11.32	0.0	268
2	+29.96	+72.56	11.4	1,310
3	+30.65	+33.95	1.5	561
4	+46.59	+32.28	9.4	1,058
5	+13.97	+24.51	1.4	457
6	+24.44	+5.37	2.4	305
7	+64.29	+53.85	3.0	3,700
8	+43.48	+4.04	1.7	255
9	+27.08	+34.51	5.8	530
10	+26.28	+56.26	4.1	1,619

Exclusive App Data Ratio: proportion of samples from apps that appear only in the target user

Conclusions

Key Takeaways

Conclusions

- Addressing the “Personal Gap”
  - Confirmed that generic models struggle with user-specific UIs
  - Highlighting the necessity of personalization
- Validated Two-Stage FT Strategy
  - Global FT for general & Personalized FT for individual
- Superior Performance of both Global FT and Personalized FT
  - Global FT significantly outperformed the Vanilla model
  - Successfully adapting to general UI layouts
  - Personalized FT significantly outperformed the Global FT
  - Verifying personalization captured user-specific features well
- Efficient FT in Personalization
  - Used LoRA as a cost-effective solution for personalization

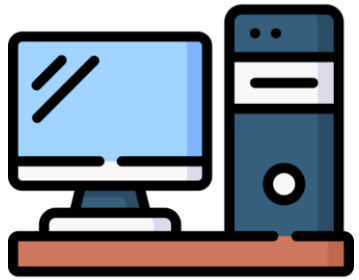
Limitation & Future Work

Conclusions

1 Single dataset only

We evaluate only on FingerTip20K, so generalization to other devices or OS is unclear.

→ Web or desktop personalization datasets experiments



2 No bounding-box ground truth

The dataset does not provide bounding-box labels for clickable UI elements,

so our accuracy relies on a 14% distance threshold rather than precise region matching.

→ Use OCR-based detection to extract bounding boxes and evaluate with region-level accuracy.

3 Privacy issue

User data is too sensitive for server-side training

→ On-device fine-tuning personalization with full privacy

Thank You

CSE403: Deep Learning

Fall Semester, 2025, UNIST

Project Page: https://hoonably.github.io/cse-archive/pigg