iSHIFT : Lightweight slow fast GUI Agent with Adaptive Perception

Supplementary Material

Contents

S.1. Dataset Enhancement

Algorithm 2 describes how each dataset sample is reformatted to include latent thinking and latent perception tokens. For every triplet (I, T, A), we apply a simple rule-based classifier f_{cat}, implemented as an `if-else` condition, to determine whether the action A requires precise visual grounding. If grounding is required, the sample is routed to the *slow path*; otherwise, it follows the *fast path*.

All the instructions are enriched with a short sequence of latent thinking tokens that supervise implicit reasoning. Samples assigned to the slow path additionally receive a bounded block of latent perception tokens, encouraging the model to rely more heavily on the localized image features. The resulting sequence $\mathcal{S}$, combined with the original image and action label, is added to the enhanced dataset $\mathcal{D}'$.

This process transforms the original dataset into a unified format in which all examples contain lightweight latent-thinking supervision, while only grounding-critical instances include additional perception cues, enabling the model to learn adaptive slow fast reasoning.

S.2. Reproducibility and Implementation Details

To facilitate reproducibility and provide clarity on our experimental setup, we detail our training pipeline, hyperparameter choices, prompt template, and datasets used in iSHIFT. Our goal is to ensure that all implementation decisions including model initialization, optimization strategy, action-space normalization, and dataset splits are transparent. *All code, configuration files, and data-processing scripts will be made publicly available upon acceptance.* Below, we outline the computation setup, phase-wise training procedure, and dataset-specific considerations followed in our experiments.

S.2.1. Computation and Training Details

Our iSHIFT agent builds on the Qwen2-VL-2B foundation model [31], initialized with its pre-trained weights. The Perception Module uses a frozen DINOv2-L encoder [20] for stable and efficient feature extraction. To control computational cost in our 2.5B model, the action history is truncated to the two most recent actions. Training is performed on a NVIDIA A100 (80GB) GPU with AdamW and DeepSpeed ZeRO Stage 2. Our training strategy contains three phases:

1. **Alignment Phase:** Only the Visual Perception Module's cross-attention projector is unfrozen and trained to integrate visual features (learning rate 2×10^{-3}, batch size 32, gradient accumulation 4).

Algorithm 2 Dataset Enhancement and Training Strategy in iSHIFT

Require: Dataset $\mathcal{D} = \{(I, T, A)\}$ where I is the UI screenshot, T is the textual instruction, and A is the ground truth action
Ensure: Enhanced dataset $\mathcal{D}'$ with structured latent-token sequences

Initialize $\mathcal{D}' \leftarrow \emptyset$
for each sample (I, T, A) in $\mathcal{D}$ **do**
  Determine perception requirement:

$$r = f_{\text{cat}}(A) \in \{\text{Low}, \text{High}\}$$

  if $r = \text{Low}$ **then** ▷ **Fast Path (Direct Reasoning)**
    Construct sequence:

$$\mathcal{S} = [T, \texttt{<bot>}, \underbrace{\langle z_1 \rangle, \ldots, \langle z_n \rangle}_{\text{latent thinking tokens}}, \texttt{<eot>}]$$

    Each $\langle z_i \rangle$ represents an implicit thinking step
  else ▷ **Slow Path (Perception-Aware Reasoning)**
    Construct sequence:

$$\mathcal{S} = [T, \texttt{<bot>}, \underbrace{\langle z_1 \rangle, \ldots, \langle z_n \rangle}_{\text{latent thinking tokens}}, \texttt{<eot>}, \texttt{<bop>}, \texttt{<ctrl>}, \texttt{<eop>}]$$

  end if
  Add enhanced sample $(\mathcal{S}, I, A)$ to $\mathcal{D}'$
end for
return Enhanced dataset $\mathcal{D}'$

2. **Thought Training Phase:** The model learns implicit thought generation using the Android in the Zoo dataset [41] (learning rate 3×10^{-5}, batch size 16, accumulation 4). Since AITZ provides ground truth thought annotations, we use it exactly as released without any modification. This stage teaches the latent tokens their deliberative role, and we observe that the resulting implicit reasoning generalizes with meaningful thought patterns well beyond the AITZ domain.
3. **Fine-tuning Phase:** Using the thought-trained checkpoint, the model learns the full adaptive slow fast strategy on downstream GUI datasets, with the same batchsize and accumulation settings and a learning rate of 2×10^{-5}.

Inference: During inference, the model receives both the instruction and the screenshot of the current GUI state along with previous actions if any. It then performs implicit task assessment to decide the appropriate reasoning path. If the task is simple, the model follows the fast path and directly predicts the action. If the task requires finer visual grounding, it selects the slow path and outputs the corresponding perception tokens. These tokens trigger the visual perception module, which returns localized features for the queried regions. Using these refined features, the model computes precise grounding coordinates and generates the final action.

S.2.2. Hyperparameters

Because iSHIFT is built directly on the Qwen2-VL architecture, nearly all architectural and optimization hyperparameters are inherited from the base model, leaving only a single meaningful hyperparameter to tune: the number of latent thinking tokens. This parameter controls the depth of the model's implicit deliberation, and as demonstrated in our ablation on latent token count (§5) has a tangible impact on performance. Based on this analysis, we set the number of latent tokens to **8**, which offers the optimal balance between reasoning capacity and efficiency.

S.2.3. Prompt Template

We employ a unified prompt template that specifies the instruction, screenshot, previous actions, and the latent tokens according to the Dataset Enhancement strategy (§S.1) described above. As illustrated in Figure S.1, the prompt explicitly structures the input into well defined blocks, enabling the model to reason over task goals, past actions, and visual context in a predictable format. This prompt formulation is shared across all training phases and benchmarks, ensuring that iSHIFT can reliably initiate implicit thinking, activate the slow path when needed, and generate grounded actions with minimal ambiguity.

Fast Path

User:
<image>
Previous Actions: {previous_actions}
Goal: {query}
Predict the next action to be taken according to the Goal
Let's think step by step.
Assistant:
<bot><latent><latent><latent><latent><latent><latent><latent><eot>
User:
Request for additional features if required or answer the question directly based on your observations
Assistant:
Action Decision:action_type:{action_type}, touch_point:[x_1, y_1], lift_point:[x_2, y_2], typed_text:{text}

Slow Path

User:
<image>
Previous Actions: {previous_actions}
Goal: {query}
Predict the next action to be taken according to the Goal
Let's think step by step.
Assistant:
<bot><latent><latent><latent><latent><latent><latent><latent><eot>
User:
Request for additional features if required or answer the question directly based on your observations
Assistant:
<bop><ctrl><eop>
User:
<detection_image>
Assistant:
Action Decision:action_type:{action_type}, touch_point:[x_1, y_1], lift_point:[x_2, y_2], typed_text:{text}

Figure S.1. **slow fast Prompt Formatting.** Illustration of how each dataset sample is reformatted into fast and slow path variants. Fast path samples receive only latent thinking tokens, whereas Slow path samples additionally include latent perception tokens to support precise grounding, following the rule-based classifier described in Algorithm 2.

S.2.4. Dataset Details

AITW [23] is a large-scale Android smartphone interaction environment consisting of approximately 30k natural-language instructions and 715k trajectories. We obtain our train, validation, and test splits from the AutoUI [39] benchmark, following the standardized partitioning used in prior work to ensure consistent and comparable evaluation across models. The action space includes 12 discrete actions: `CLICK`, `TYPE`, `SELECT`, `SCROLL UP`, `SCROLL DOWN`, `SCROLL LEFT`, `SCROLL RIGHT`, `PRESS BACK`, `PRESS HOME`, `PRESS ENTER`, `STATUS TASK COMPLETE`, and `STATUS TASK IMPOSSIBLE`.

Android Control [12] is an Android UI interaction benchmark that provides 13,604 training episodes (74,722 step-level actions) and 2,855 test episodes, offering a controlled environment for evaluating generalization in mobile UI manipulation. We use the official train–test splits provided by the benchmark to ensure consistency with prior work. The dataset defines a structured action space covering nine interaction types: `click`, `long_press`, `type`, directional `scroll`, `navigate_home`, `navigate_back`, `open_app`, `wait`, and `terminate`. These actions enable fine-grained control over device navigation, text entry, and task completion signaling. We modify this dataset to match the action space of AITW for consistent training.

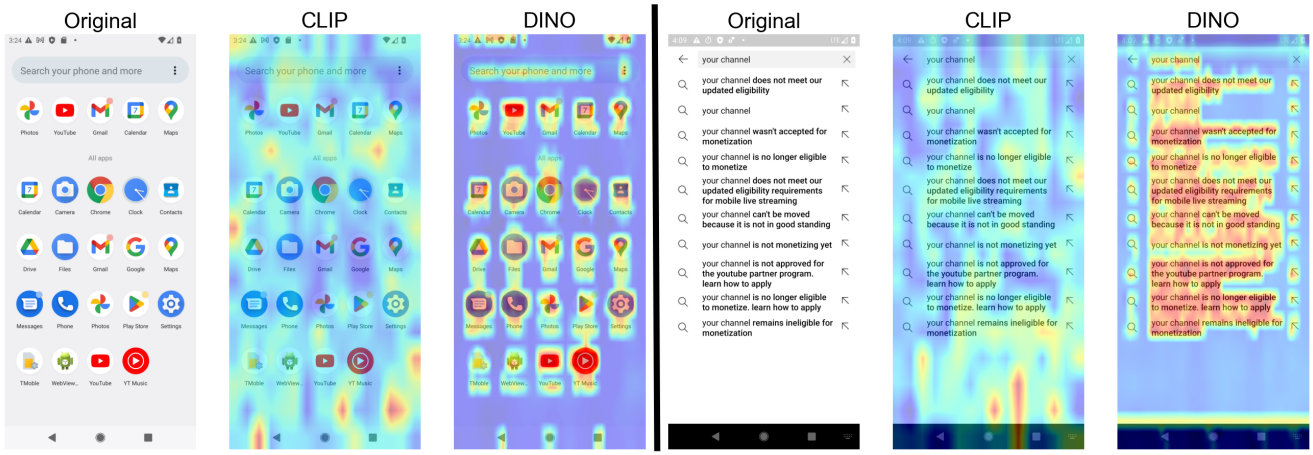


Figure S.2. Attention map comparison between CLIP and DINO visual encoders. DINO produces more localized and structured attention over relevant UI elements, while CLIP exhibits broader, less focused activations.

GUI Odyssey [15] is a large-scale benchmark designed to evaluate generalization in multi-device and multi-application GUI interaction. The dataset contains 8,334 episodes and provides several standardized train–test configurations. In the Random setup, the data is split 80/20, yielding approximately 6,667 training and 1,667 testing episodes. The Task subset allocates data across six task categories in a 6:1 ratio, resulting in roughly 7,144 training and 1,190 test episodes. For the Device configuration, all 1,381 episodes collected on the Pixel Tablet are reserved for testing, while the remaining 6,953 episodes form the training set. Finally, the App subset divides episodes by application frequency with an 85/15 ratio, producing approximately 7,084 training and 1,250 test samples. GUI Odyssey defines a nine-action interaction space consisting of `CLICK`, `LONG_PRESS`, `SCROLL`, `TYPE`, `COMPLETE`, `IMPOSSIBLE`, `HOME` and `BACK`, each with structured arguments specifying positions, gestures, or navigation commands. We modify this dataset to match the action space of AITW for consistent training and evaluation across benchmarks.

GUIAct [4] is a multi-platform benchmark for evaluating task-oriented GUI agents across web and smartphone environments. The dataset provides these standard splits: Web-Single, with 67k training and 1.4k testing samples and Smartphone, with 9,157 training and 2k testing samples. The action space is as follows: `Click`, `Hover`, `Tap`, `Input`, `Type`, `Copy`, `Paste`, `Scroll`, `Drag`, `Swipe`, `Enter` and `Answer`. For consistent evaluation across benchmarks, we normalize this action space by merging semantically equivalent operations.

S.3. Initialization of Latent Tokens

To validate the importance of our implicit thought–training stage, we conducted an ablation in which the latent thought tokens were initialized from scratch and trained only during downstream fine-tuning, without exposure to the Android in the Zoo thought-supervision phase. As shown in Table S.1, removing the thought pretraining leads to consistent performance drops across all evaluation splits, with the largest declines observed in the more open-ended General and G.Apps categories. In contrast, equipping the model with thought pretraining exhibits stronger reasoning consistency and improved robustness, yielding an absolute average gain of 3.3 points. These results confirm that early alignment of latent tokens toward deliberative behavior plays a crucial role in enabling the model to effectively leverage the adaptive slow fast strategy during GUI interaction.

Method	General	Install	G.Apps	WebShop	Single	Avg
w/o Thought Pretraining	66.75	78.34	67.80	68.96	83.34	73.04
with Thought Pretraining	**70.60**	**80.82**	**71.64**	**72.60**	**86.03**	**76.34**

Table S.1. **Effect of latent-token initialization on adaptive slow fast reasoning**. Thought-pretrained latent tokens significantly improve performance, confirming their role in enabling reliable slow-path grounding and overall task success.

S.4. Chain of Thought v/s Latent Thinking

To evaluate the effectiveness and generalization ability of our implicit-thinking approach relative to explicit chain-of-thought (CoT) reasoning, we train on the Android In The Zoo (AITZ) dataset [41] and assess performance on the diverse test splits of Android In The Wild (AITW) [23]. AITZ is constructed from a small subset of AITW and is a publicly available dataset that provides ground truth thought annotations. Therefore, for our CoT baseline, we use the dataset exactly as released without modifying or augmenting any annotation. AITZ contains approximately 18k training examples with detailed thoughts and screen descriptions, while the combined AITW test sets include around 127k samples, forming a challenging out-of-distribution benchmark for assessing generalization of the latent tokens. As shown in Table S.2, implicit thinking consistently outperforms explicit CoT across all AITW splits, achieving gains of 8–11 points on average despite using nearly $8\times$ fewer tokens. Notably, the implicit model delivers the strongest improvements in the most complex domains: General, Google Apps, and WebShop, demonstrating that compact latent reasoning not only generalizes better but also provides significantly lower inference overhead compared to explicit CoT.

Method	General	Install	G.Apps	WebShop	Single	Avg Tokens
Chain of Thought	46.03	64.97	49.81	49.78	66.57	62
Implicit Thinking	**56.54**	**71.73**	**60.28**	**58.75**	**77.22**	8

Table S.2. **Implicit thinking vs. chain-of-thought when trained on AITZ and evaluated on AITW**. Using AITZ (18k thought annotated samples) for training and the full AITW benchmark (127k test samples) for evaluation, implicit reasoning delivers consistent improvements over explicit CoT across all domains, while operating with an order-of-magnitude fewer tokens.

S.5. Analysis on Slow-Fast Switching

We provide direct diagnostics of the learned policy in Fig. S.3. iSHIFT exhibits fewer Click/Non-Click confusions and achieves higher *Click Acc* (**73.5%**) compared to fixed-path baselines that always use Slow (**69.4%**) or Fast (**65.5%**). Notably, the Always-Slow variant shares a similar programmatic perception augmentation as iSHIFT, indicating that the gains are not due to simply observing additional perception tokens, but rather from learning *when* to invoke the slow pathway.

We further validate this via controlled ablations in Table S.3. Inverting the learned switching decisions for 30% of cases leads to a substantial drop in performance (General / Install / Avg: **70.6 / 80.82 / 76.34** $\rightarrow$ **63.8 / 70.25 / 68.28**), demonstrating that the switching policy is critical to performance.

Finally, we compare against a multimodal transformer-based **external controller** that predicts when to invoke enhanced perception. As shown in Table S.3, it achieves **67.23 / 73.96 / 71.52** (Gen / Install / Avg), which remains below iSHIFT's **70.6 / 80.82 / 76.34**. These results support our central claim: the key contribution is not merely adding a perception module, but *jointly learning the switching policy within the model*, aligning control with internal state and downstream action prediction. In this sense, iSHIFT can be interpreted as a form of generalized *tool use*, where the slow pathway acts as an optional perception tool whose invocation is learned end-to-end.

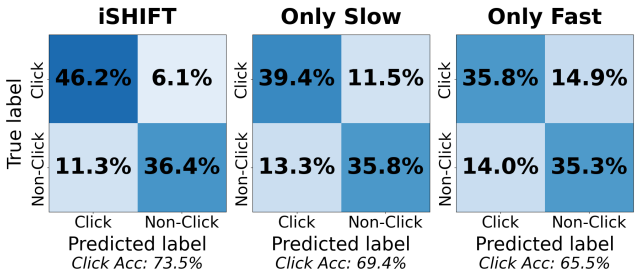


Figure S.3. Action-type confusion matrices (Click vs. Non-Click). *Click Acc* is shown below each matrix.

Method	General	Install	G.Apps	WebShop	Single	Average
External Controller	67.23	73.96	68.79	67.98	79.65	71.52
30% Inverted	63.8	70.25	65.87	64.22	77.28	68.28
iSHIFT	**70.6**	**80.82**	**71.64**	**72.6**	**86.03**	**76.34**

Table S.3. **Effect of learned slow-fast switching on AITW performance.** iSHIFT consistently outperforms both an external controller and a perturbed (30% inverted) switching policy across all domains.

S.6. iSHIFT is architecture agnostic

iSHIFT is **base-model agnostic** and transfers effectively across different backbones. As shown in Table S.4, applying iSHIFT to *Qwen2.5-VL-3B* yields consistent improvements over the base model across all evaluated benchmarks. In particular,

performance increases from **58.79 to 61.48** on Android Control-High, from **41.22 to 79.61** on Control-Low, and from **65.82 to 70.28** on AITW-Average. While *Qwen2.5-VL* is generally a stronger backbone, we adopt *Qwen2-VL* as our primary model due to its superior empirical performance in our setting.

Method	Android Control		AITW Avg
	High	Low	
Qwen2.5-VL-3B	58.79	41.22	65.82
Qwen2.5-VL-3B (iSHIFT)	**61.48**	**79.61**	**70.28**

Table S.4. **iSHIFT generalizes across base models.** On Qwen2.5-VL-3B, iSHIFT consistently improves performance over the base model across benchmarks, increasing Android Control-High from 58.79 to 61.48, Control-Low from 41.22 to 79.61, and AITW Average from 65.82 to 70.28. This demonstrates that iSHIFT is base-model agnostic and transfers effectively to stronger backbones.

S.7. CLIP vs DINO in Visual Perception Module

Figure S.2 compares attention maps of CLIP and DINO on examples from AITW dataset. As observed, CLIP's heatmaps show more diffuse, global attention that loosely corresponds to semantically meaningful areas (like icons or text regions), reflecting its image text training objective. In contrast, DINO produces sharper, object centered attention, precisely outlining icons, buttons, and text lines, demonstrating its strong spatial and structural awareness learned through self supervised vision training. Overall, CLIP captures semantic meaning, while DINO captures visual structure and object boundaries.

S.8. Comparison with RL based Methods

We additionally compare iSHIFT against recent reinforcement learning based GUI agents across multiple benchmarks to contextualize its performance relative to policy optimization approaches. As shown in Table S.5, iSHIFT matches or surpasses strong RL baselines such as DigiRL and DistRL on the custom AITW test subsets, despite using a purely supervised training pipeline without any environment rollouts or reward engineering. Notably, our model provides large gains on the Web Shopping split, highlighting the benefits of adaptive perception in visually dense, multi step tasks. Beyond AITW, iSHIFT also achieves substantial improvements on the Android Control and GUI Odyssey benchmarks (Table S.6), outperforming prior RL agents including UI-R1, GUI-R1, and SWIRL, by wide margins across both high and low complexity scenarios. These results emphasize that our token based slow fast strategy yields stronger generalization and more reliable action grounding than methods relying on reinforcement learning.

Method	General	Web Shopping	Test Set
DigiRL	71.90	67.20	First 96 task samples
Ours	**72.25**	**72.28**	
DistRL	**73.20**	68.50	First 128 task samples
Ours	71.85	**73.19**	

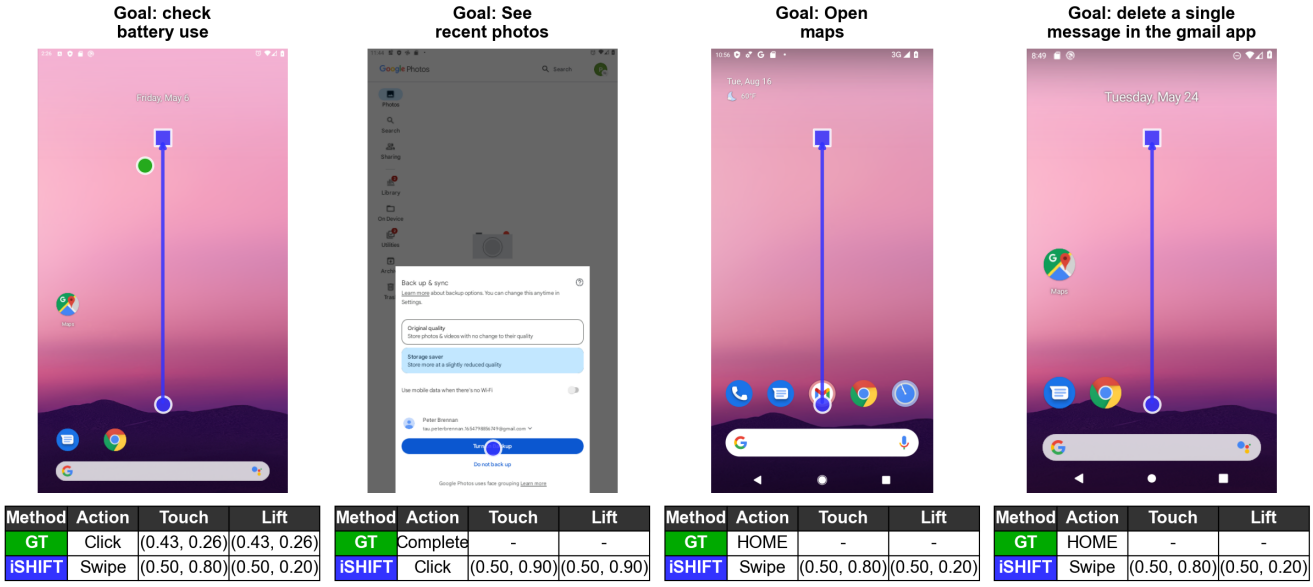
Table S.5. **Comparison with RL based agents on AITW**. Results on the custom AITW test subsets used by DigiRL and DistRL. iSHIFT consistently matches or surpasses RL baselines despite using purely supervised training without rollouts or rewards.

Method	**Android Control**		**GUI Odyssey**
	High	Low	
UI-R1	45.44	66.44	32.49
GUI-R1 3B	46.55	64.41	41.33
GUI-R1 7B	51.67	66.52	38.79
SWIRL	51.24	78.81	51.65
iSHIFT	**65.6**	**87.7**	**73.97**

Table S.6. **Comparison of iSHIFT with prior RL-based GUI agents on Android Control and GUI Odyssey**. iSHIFT achieves the highest performance across all settings, outperforming UI-R1, GUI-R1 (3B and 7B), and SWIRL in both simple and complex interaction scenarios.

S.9. Not Just Following Instructions: iSHIFT Understands

This section presents qualitative examples from AITW dataset where iSHIFT's actions differ from annotated ground truth trajectories yet still achieve the intended goal effectively. These examples demonstrate that iSHIFT is not limited to imitating annotations but can flexibly adapt its interactions based on visual context and task objectives. We categorize such behaviors into three representative types: (i) Cases where the provided annotation is wrong while iSHIFT performs the correct action. (ii) iSHIFT takes shorter and more efficient completion routes, and (iii) iSHIFT takes alternate but valid interaction paths that



Method	Action	Touch	Lift
GT	Click	(0.43, 0.26)	(0.43, 0.26)
iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)

Method	Action	Touch	Lift
GT	Complete	-	-
iSHIFT	Click	(0.50, 0.90)	(0.50, 0.90)

Method	Action	Touch	Lift
GT	HOME	-	-
iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)

Method	Action	Touch	Lift
GT	HOME	-	-
iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)

Figure S.4. Examples where the ground truth annotations are incorrect, while iSHIFT performs the correct action to achieve the goal.

reach the goal. Together, these examples highlight iSHIFT's robustness and ability to generalize beyond the exact sequences observed during supervision.

Handling Annotation Errors Figure S.4 highlights situations where iSHIFT performs the correct interaction even when the annotated ground truth is incomplete or inaccurate. In tasks such as "Check battery use," "Open maps," and "Delete a single message in Gmail," the annotations specify incorrect or missing actions (e.g., HOME or Click), whereas iSHIFT executes valid gestures that reveal the intended interface or open the correct app. Likewise, in "See recent photos," iSHIFT selects the appropriate navigation button, while the annotation prematurely marks the task as complete. These examples suggest that minor inconsistencies in GUI annotations do not prevent iSHIFT from executing valid, goal-directed interactions.

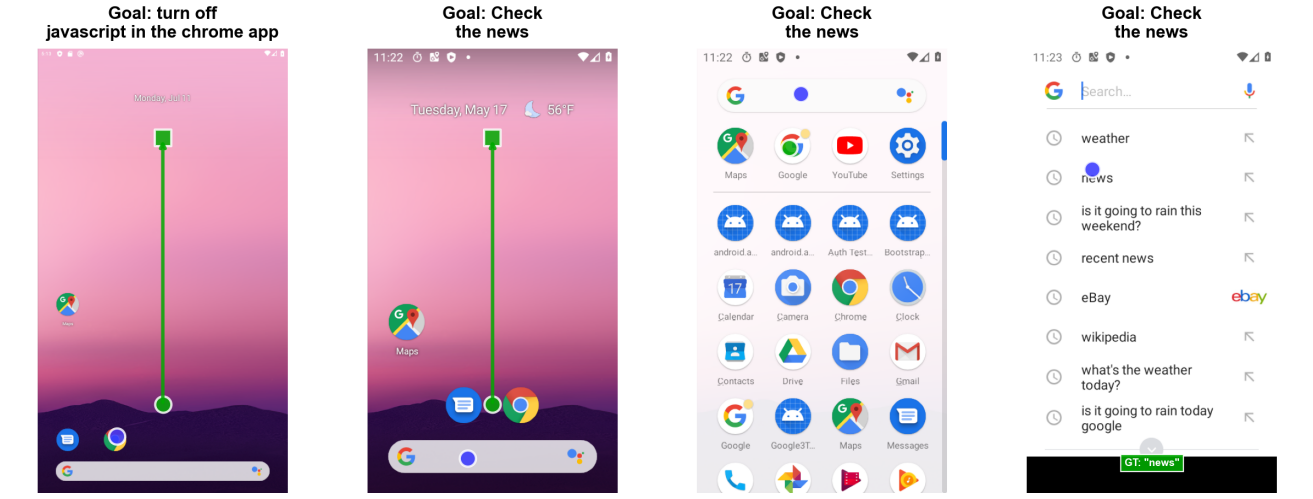
More Efficient Task Completions As shown in Figure S.5, iSHIFT often completes tasks through shorter and more efficient action sequences. For instance, in "Turn off JavaScript in Chrome," it directly clicks the Chrome icon instead of performing an intermediate swipe. In "Check the news" examples, it bypasses the step of opening the app drawer and interacts directly with the Google search bar, selecting an existing suggestion instead of typing a full query. Such behaviors demonstrate that iSHIFT can streamline interactions to minimize unnecessary steps while still reaching the correct goal.

Alternate but Valid Paths Figure S.6 shows cases where iSHIFT follows a different yet valid sequence of actions to complete the task. For "What's the news in Chile?", it clicks on a suggested query instead of pressing ENTER, producing the same result. For "Open a new tab in Chrome," it accesses the tabs option from the extended menu rather than using the tabs menu. Similarly, in the Costco and eBay tasks, it selects the search symbol instead of selecting the nearby search suggestions that lead to equivalent outcomes. These examples indicate that iSHIFT can flexibly adapt its interaction pattern while remaining consistent with the task objective.

S.10. Limitations

In some cases, iSHIFT performs actions that differ from both the ground truth and the intended goal. These deviations typically stem from: (i) the model occasionally selects the fast path when a task benefits from deeper visual reasoning (see Figure S.7), (ii) it may opt for the slow path on simpler interactions (see Figure S.8), and (iii) the Visual Perception Module may occasionally provide slightly imprecise localized features (see Figure S.9). Nonetheless, as shown in Figure S.10 and S.11, such occurrences are rare, indicating that iSHIFT's adaptive control and perception mechanisms remain stable.

Choosing the Fast Path instead of Slow and vice-versa. While iSHIFT is designed to dynamically balance between the slow and fast reasoning paths, occasional mismatches occur where it selects the fast route for tasks requiring deeper perception (see Figure S.7) or the slow route for simpler interactions (see Figure S.8). These cases arise from subtle variations in visual complexity estimation or perception cues. However, as shown in Figure S.10 and S.11, such instances are infrequent, iSHIFT maintains a nearly even and well-calibrated distribution of slow and fast path usage, closely aligning with the ground truth



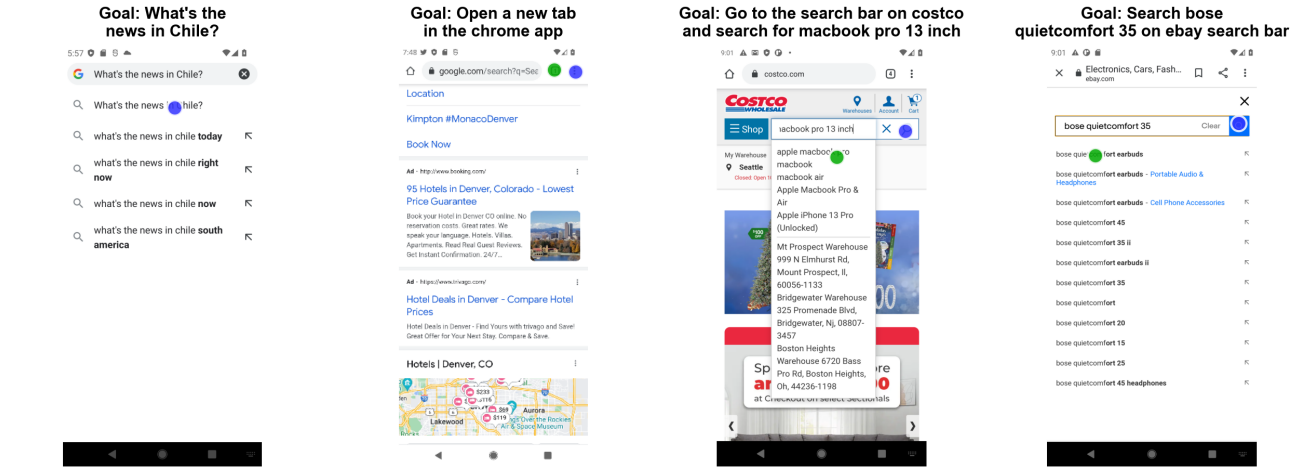
Method	Action	Touch	Lift
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)
iSHIFT	Click	(0.32, 0.87)	(0.32, 0.87)

Method	Action	Touch	Lift
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)
iSHIFT	Click	(0.40, 0.92)	(0.40, 0.92)

Method	Action	Touch	Lift
GT	Click	(0.37, 0.22)	(0.37, 0.22)
iSHIFT	Click	(0.41, 0.10)	(0.41, 0.10)

Method	Action	Touch	Lift
GT	TYPE	-	-
iSHIFT	Click	(0.26, 0.27)	(0.26, 0.27)

Figure S.5. Examples where iSHIFT completes tasks through shorter or more efficient interaction paths than the ground truth trajectories, demonstrating its ability to identify faster yet valid routes to achieve the same goals.



Method	Action	Touch	Lift
GT	ENTER	-	-
iSHIFT	Click	(0.56, 0.14)	(0.56, 0.14)

Method	Action	Touch	Lift
GT	Click	(0.82, 0.05)	(0.82, 0.05)
iSHIFT	Click	(0.93, 0.06)	(0.93, 0.06)

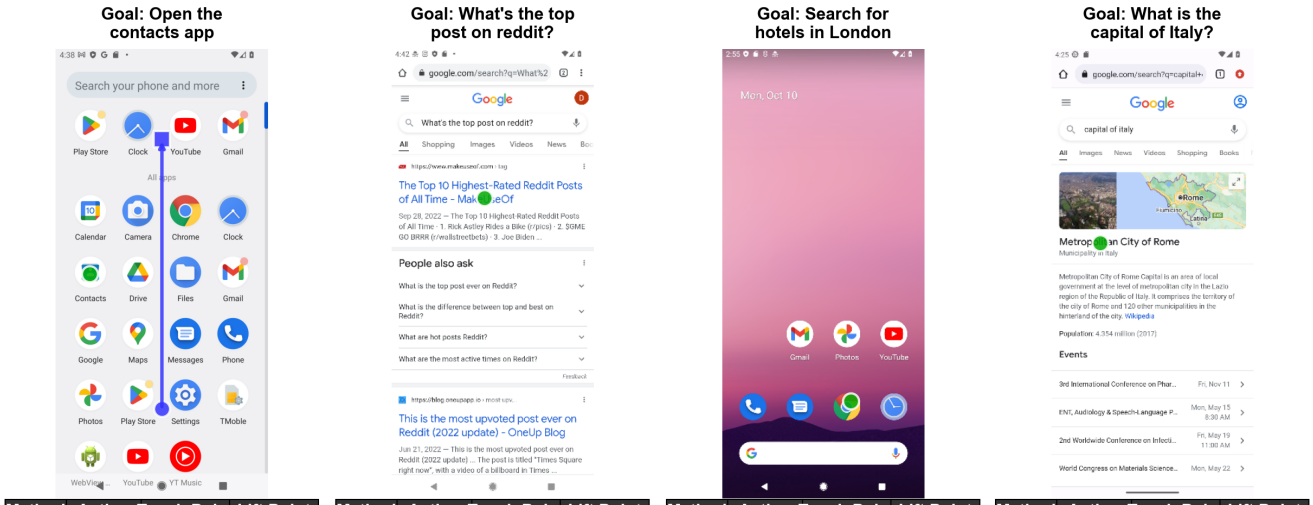
Method	Action	Touch	Lift
GT	Click	(0.57, 0.26)	(0.57, 0.26)
iSHIFT	Click	(0.90, 0.20)	(0.90, 0.20)

Method	Action	Touch	Lift
GT	Click	(0.23, 0.26)	(0.23, 0.26)
iSHIFT	Click	(0.91, 0.18)	(0.91, 0.18)

Figure S.6. Examples where iSHIFT follows alternate but valid interaction paths that differ from the ground truth actions yet successfully achieve the intended goals.

statistics. This indicates that the adaptive control mechanism is generally stable, with only minor room for refinement in boundary cases.

Visual Perception Module. Figure S.9 illustrates cases where iSHIFT's Visual Perception Module shows minor localization differences. In the first example, when opening the Calendar app, iSHIFT's attention is slightly offset from the target icon, leading to a small spatial error. In the second example, when asked to play a YouTube video, it selects one of the two YouTube icons present on the screen, different from the annotated ground truth but still a valid choice.



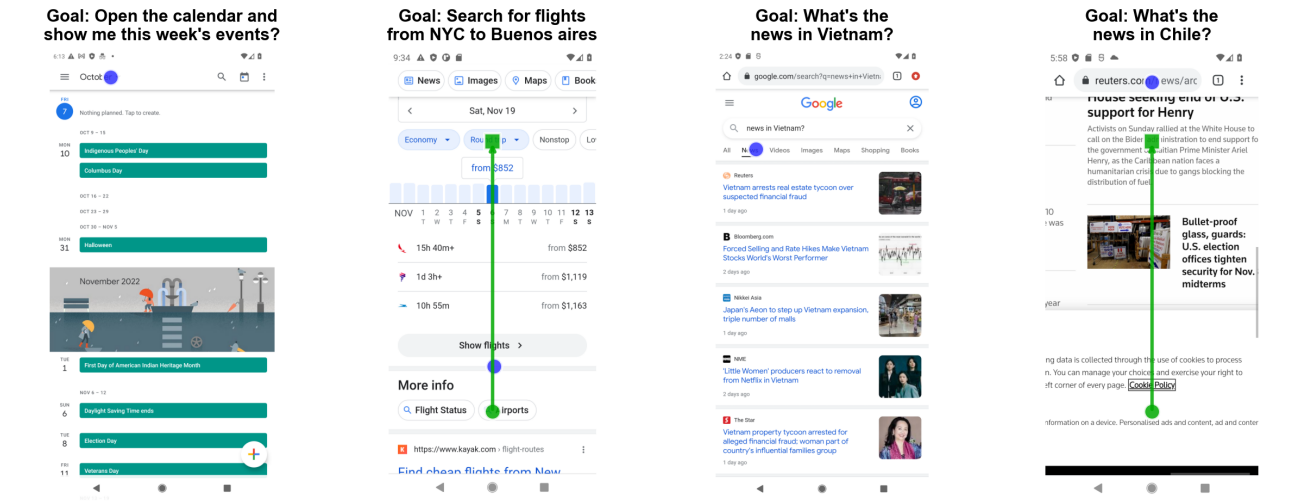
Method	Action	Touch Poin	Lift Point
GT	Click	(0.16, 0.50)	(0.16, 0.50)
iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)

Method	Action	Touch Poin	Lift Point
GT	Click	(0.46, 0.33)	(0.46, 0.33)
iSHIFT	ENTER	-	-

Method	Action	Touch Poin	Lift Point
GT	Click	(0.64, 0.78)	(0.64, 0.78)
iSHIFT	HOME	-	-

Method	Action	Touch Poin	Lift Point
GT	Click	(0.24, 0.43)	(0.24, 0.43)
iSHIFT	Complete	-	-

Figure S.7. Examples where iSHIFT occasionally selects the fast path for tasks that require deeper perception.



Method	Action	Touch Poin	Lift Point
GT	Complete	-	-
iSHIFT	Click	(0.27, 0.06)	(0.27, 0.06)

Method	Action	Touch Poin	Lift Point
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)
iSHIFT	Click	(0.51, 0.70)	(0.51, 0.70)

Method	Action	Touch Poin	Lift Point
GT	Complete	-	-
iSHIFT	Click	(0.19, 0.22)	(0.19, 0.22)

Method	Action	Touch Poin	Lift Point
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)
iSHIFT	Click	(0.50, 0.07)	(0.50, 0.07)

Figure S.8. Examples where iSHIFT occasionally selects the slow path for simpler interactions.

S.11. Qualitative Results and Comparisons

We provide qualitative examples illustrating how iSHIFT behaves across diverse GUI environments and interaction challenges (Figures S.12 - S.20). As shown in our qualitative figures, the model reliably selects between the fast and slow paths and produces actions with high spatial precision. We also compare iSHIFT against prior agents such as ShowUI[13] and SeeClick[5] highlighting fewer failure modes and more stable behavior in cases involving dense UI elements, ambiguous affordances, or fine-grained interaction requirements (Figures S.21 - S.24). These examples offer an intuitive understanding of why our adaptive slow fast strategy yields strong improvements across benchmarks.

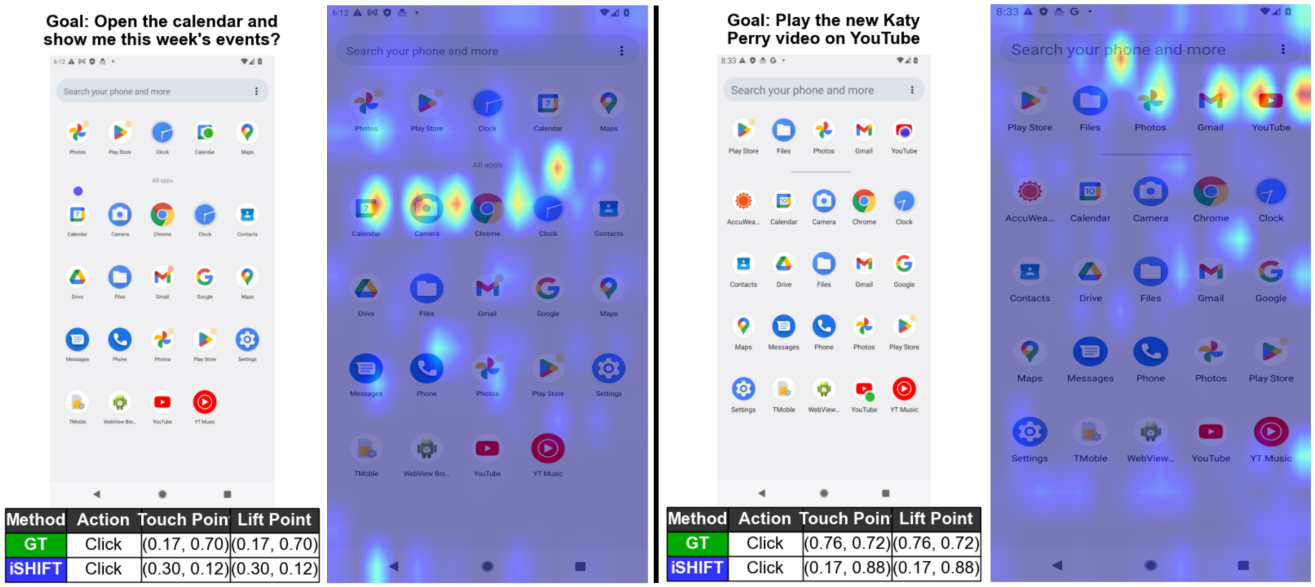


Figure S.9. Examples showing minor localization variations from iSHIFT's Visual Perception Module.

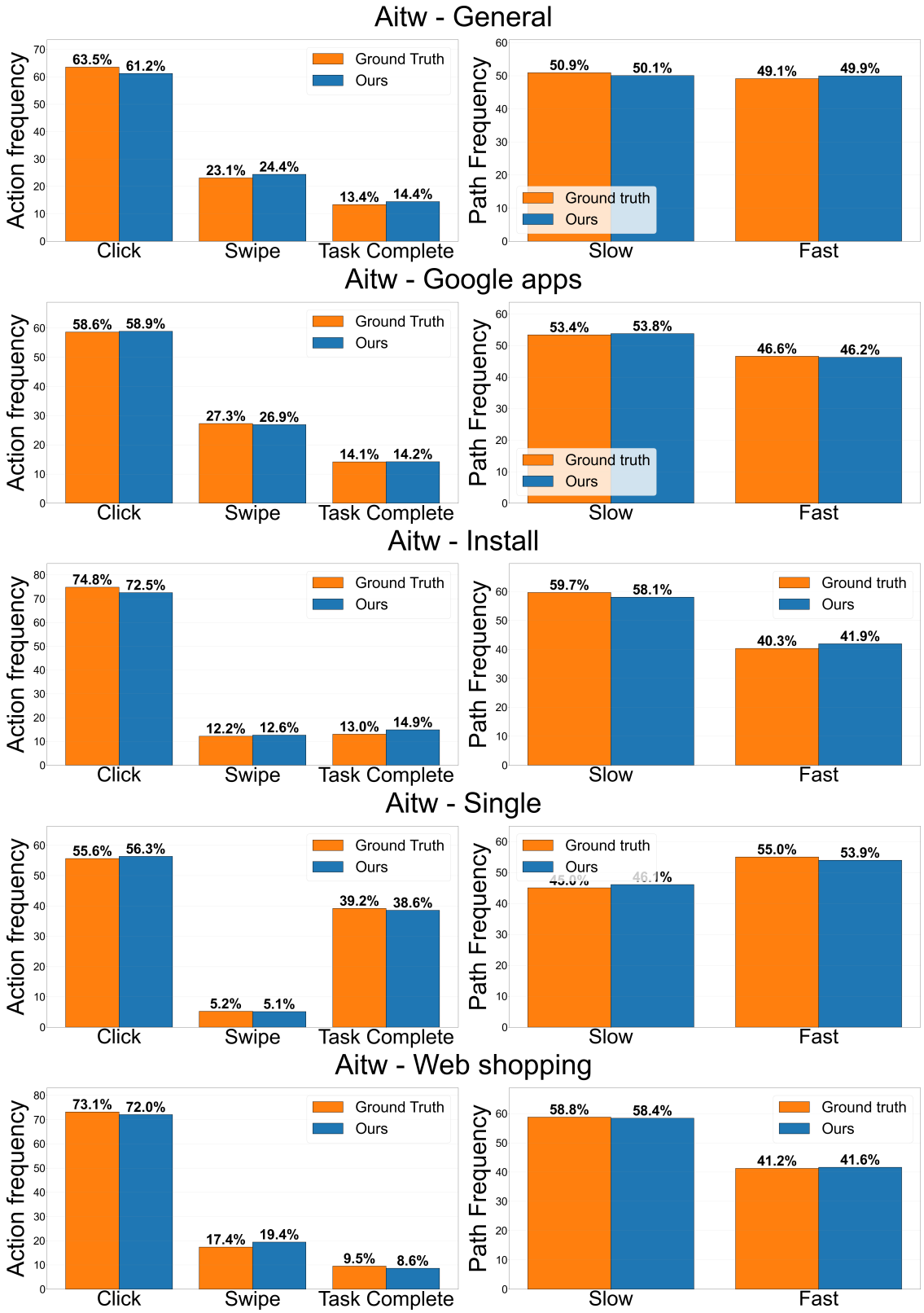


Figure S.10. Action and path distribution on AITW test data. iSHIFT closely matches ground truth action frequencies and accurately predicts both slow and fast paths across all task categories.

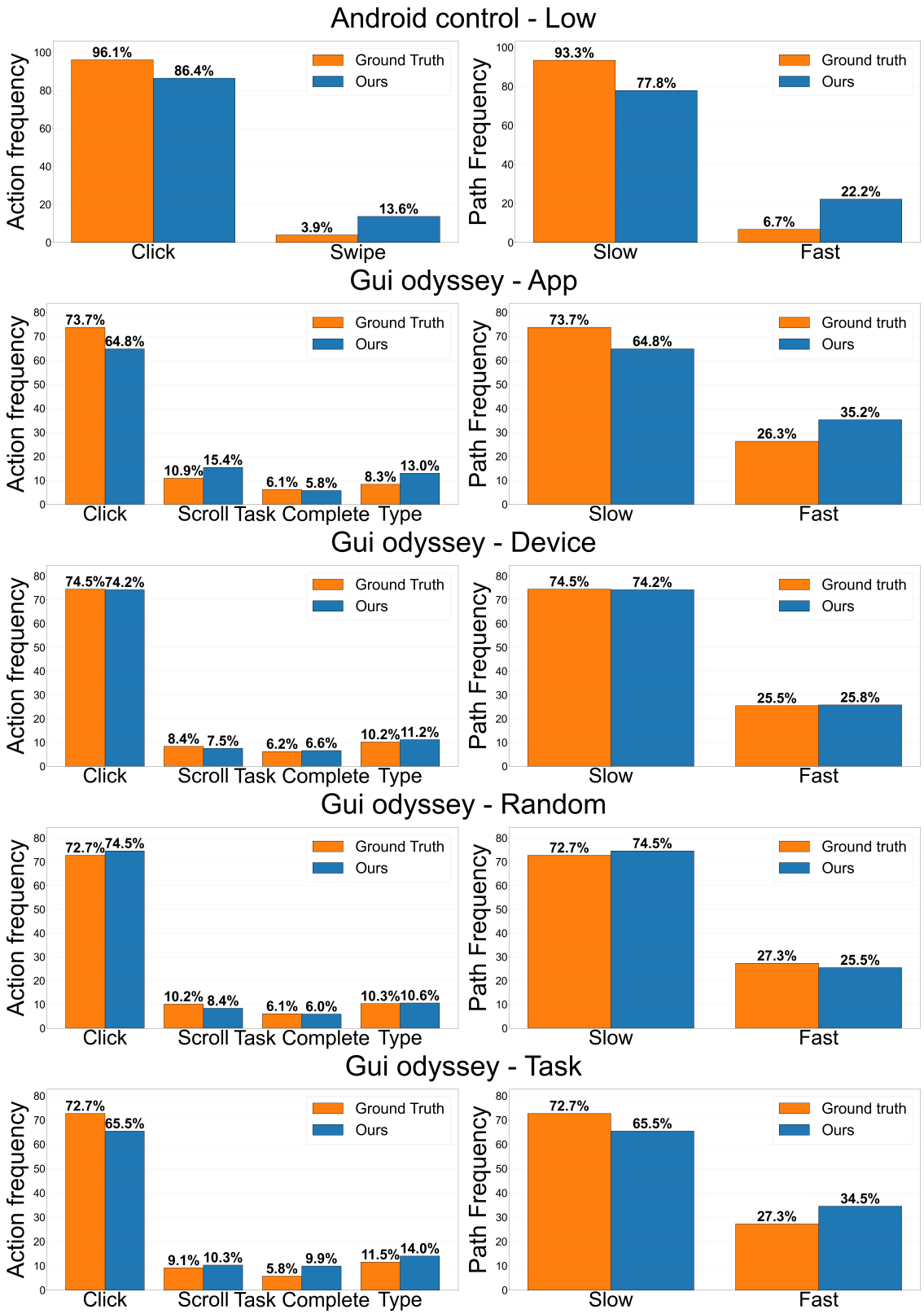
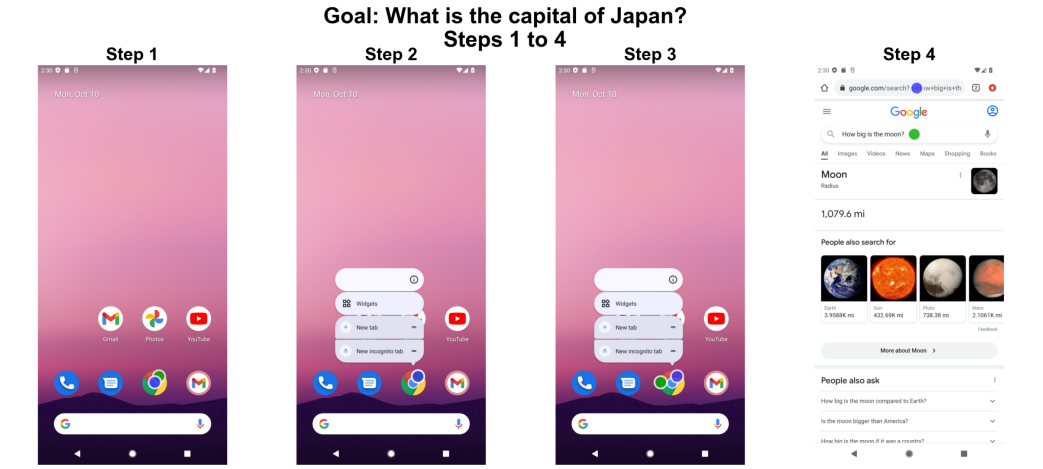
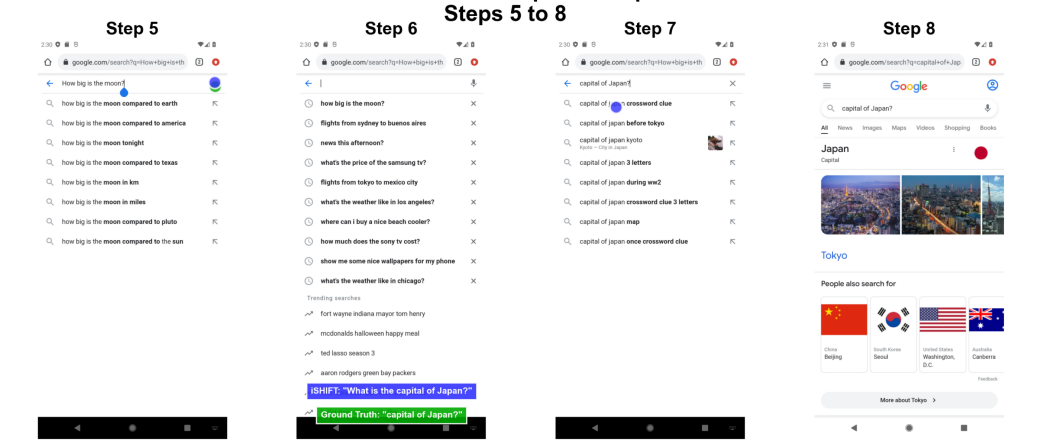


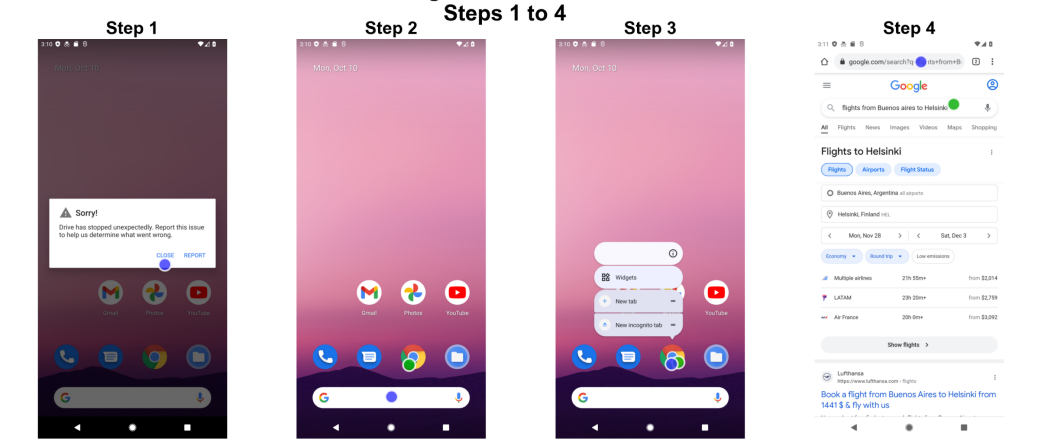
Figure S.11. Action and path distribution on Android Control and GUI odyssey test data. iSHIFT closely matches ground truth action frequencies and accurately predicts both slow and fast paths across all task categories.



Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift
GT	Click	(0.64, 0.78)	(0.64, 0.78)	GT	Click	(0.64, 0.78)	(0.64, 0.78)	GT	Click	(0.55, 0.79)	(0.55, 0.79)	GT	Click	(0.53, 0.17)	(0.53, 0.17)
iSHIFT	Click	(0.62, 0.79)	(0.62, 0.79)	iSHIFT	Click	(0.64, 0.78)	(0.64, 0.78)	iSHIFT	Click	(0.64, 0.78)	(0.64, 0.78)	iSHIFT	Click	(0.54, 0.06)	(0.54, 0.06)

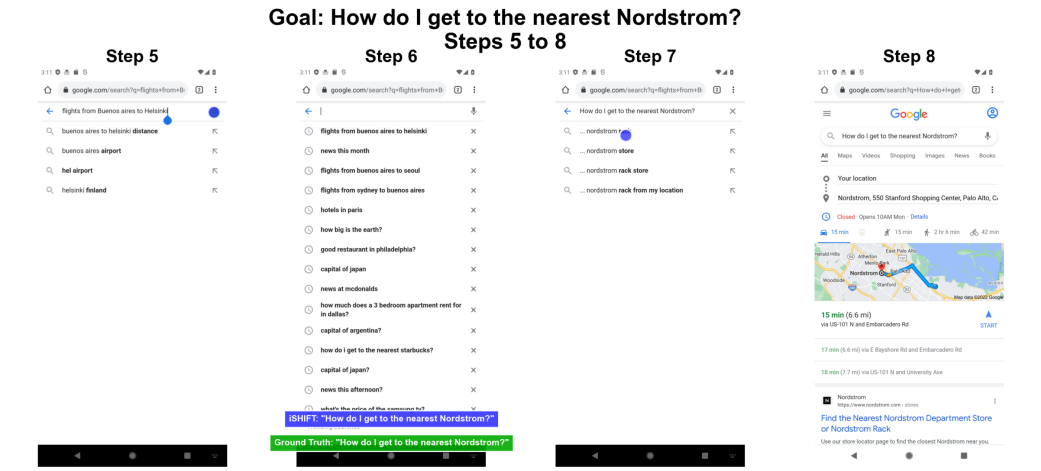


Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift
GT	Click	(0.94, 0.12)	(0.94, 0.12)	GT	TYPE	-	-	GT	Enter	-	-	GT	Comp.	-	-
iSHIFT	Click	(0.93, 0.11)	(0.93, 0.11)	iSHIFT	TYPE	-	-	iSHIFT	Click	(0.32, 0.17)	(0.32, 0.17)	iSHIFT	Comp.	-	-



Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift
GT	Home	-	-	GT	Click	(0.59, 0.81)	(0.59, 0.81)	GT	Click	(0.65, 0.81)	(0.65, 0.81)	GT	Click	(0.73, 0.16)	(0.73, 0.16)
iSHIFT	Click	(0.67, 0.56)	(0.67, 0.56)	iSHIFT	Click	(0.50, 0.89)	(0.50, 0.89)	iSHIFT	Click	(0.61, 0.81)	(0.61, 0.81)	iSHIFT	Click	(0.56, 0.06)	(0.56, 0.06)

Figure S.12. Episodes from the AITW dataset comparing iSHIFT trajectories with the ground-truth demonstrations. iSHIFT closely follows the intended interaction flow, and when deviations occur, the model selects alternative but valid action sequences that still achieve the correct task outcome.

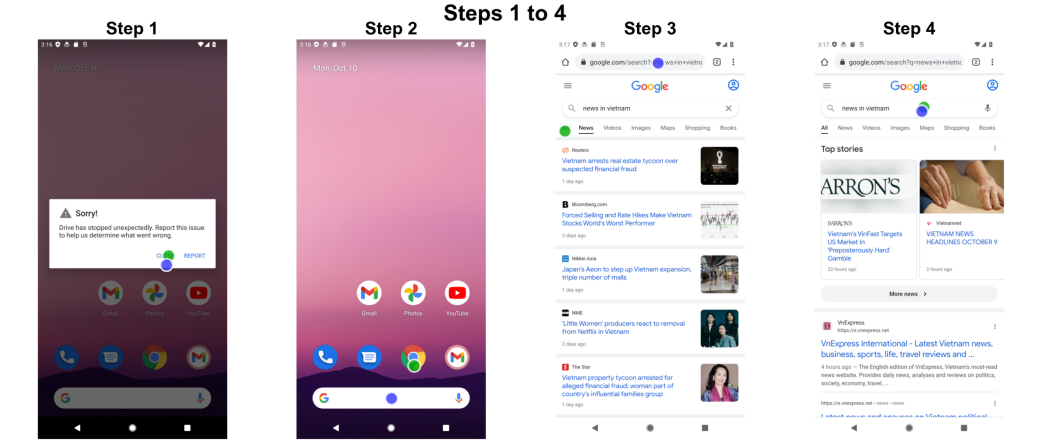


Method	Act.	Touch	Lift
GT	Click	(0.93, 0.11)	(0.93, 0.11)
iSHIFT	Click	(0.93, 0.11)	(0.93, 0.11)

Method	Act.	Touch	Lift
GT	TYPE	-	-
iSHIFT	TYPE	-	-

Method	Act.	Touch	Lift
GT	Enter	-	-
iSHIFT	Click	(0.37, 0.17)	(0.37, 0.17)

Method	Act.	Touch	Lift
GT	Comp.	-	-
iSHIFT	Comp.	-	-

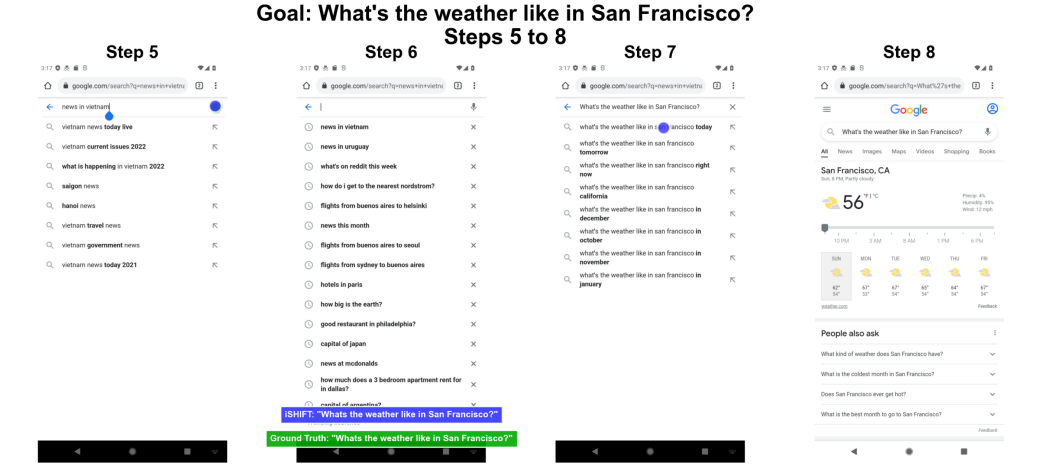


Method	Act.	Touch	Lift
GT	Click	(0.69, 0.54)	(0.69, 0.54)
iSHIFT	Click	(0.68, 0.56)	(0.68, 0.56)

Method	Act.	Touch	Lift
GT	Click	(0.62, 0.82)	(0.62, 0.82)
iSHIFT	Click	(0.50, 0.90)	(0.50, 0.90)

Method	Act.	Touch	Lift
GT	Click	(0.05, 0.23)	(0.05, 0.23)
iSHIFT	Click	(0.54, 0.06)	(0.54, 0.06)

Method	Act.	Touch	Lift
GT	Click	(0.58, 0.17)	(0.58, 0.17)
iSHIFT	Click	(0.56, 0.18)	(0.56, 0.18)



Method	Act.	Touch	Lift
GT	Click	(0.94, 0.11)	(0.94, 0.11)
iSHIFT	Click	(0.94, 0.11)	(0.94, 0.11)

Method	Act.	Touch	Lift
GT	TYPE	-	-
iSHIFT	TYPE	-	-

Method	Act.	Touch	Lift
GT	Enter	-	-
iSHIFT	Click	(0.57, 0.16)	(0.57, 0.16)

Method	Act.	Touch	Lift
GT	Comp.	-	-
iSHIFT	Comp.	-	-

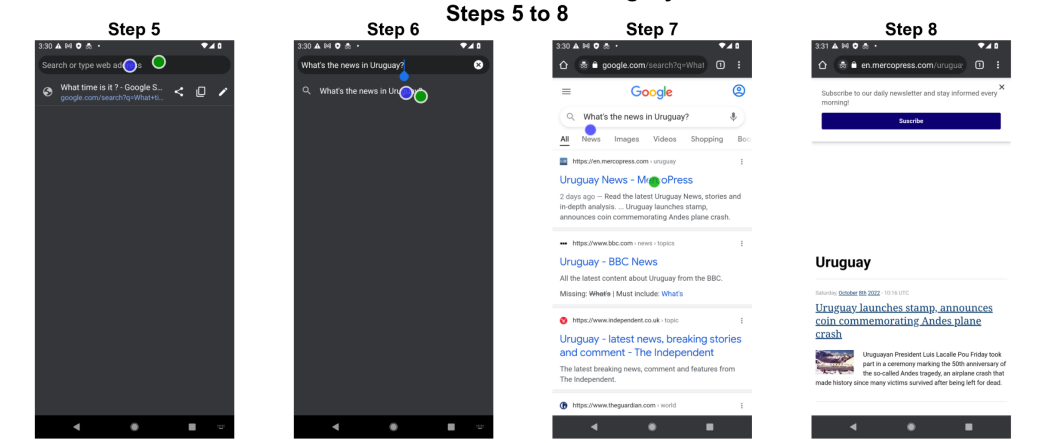
Figure S.13. Episodes from the AITW dataset comparing iSHIFT trajectories with the ground-truth demonstrations. iSHIFT closely follows the intended interaction flow, and when deviations occur, the model selects alternative but valid action sequences that still achieve the correct task outcome.

Method	Act.	Touch	Lift
GT	Home	-	-
iSHIFT	Home	-	-

Method	Act.	Touch	Lift
GT	Click	(0.66, 0.78)	(0.66, 0.78)
iSHIFT	Click	(0.69, 0.78)	(0.69, 0.78)

Method	Act.	Touch	Lift
GT	Click	(0.48, 0.06)	(0.48, 0.06)
iSHIFT	Click	(0.50, 0.06)	(0.50, 0.06)

Method	Act.	Touch	Lift
GT	TYPE	-	-
iSHIFT	TYPE	-	-

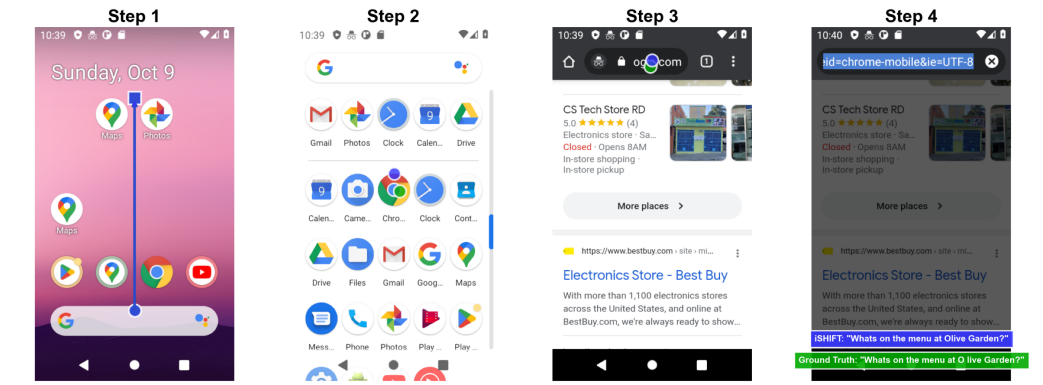


Method	Act.	Touch	Lift
GT	Click	(0.62, 0.05)	(0.62, 0.05)
iSHIFT	Click	(0.48, 0.06)	(0.48, 0.06)

Method	Act.	Touch	Lift
GT	Click	(0.64, 0.14)	(0.64, 0.14)
iSHIFT	Click	(0.56, 0.13)	(0.56, 0.13)

Method	Act.	Touch	Lift
GT	Click	(0.51, 0.36)	(0.51, 0.36)
iSHIFT	Click	(0.19, 0.22)	(0.19, 0.22)

Method	Act.	Touch	Lift
GT	Comp.	-	-
iSHIFT	Comp.	-	-



Method	Act.	Touch	Lift
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)
iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)

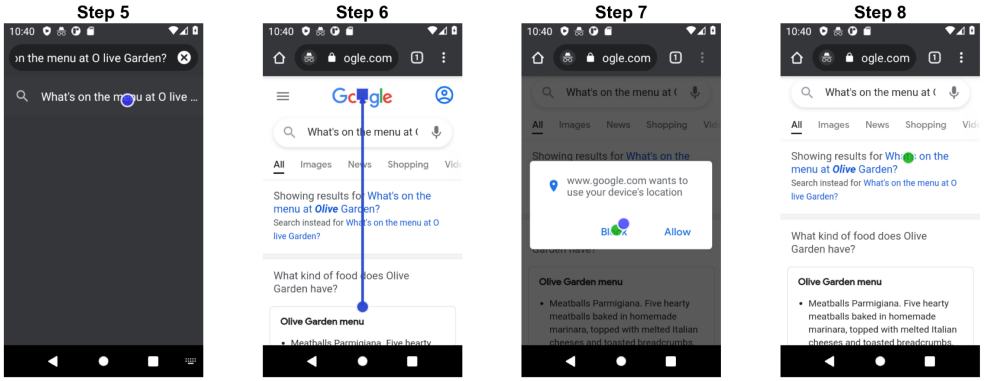
Method	Act.	Touch	Lift
GT	Click	(0.53, 0.46)	(0.53, 0.46)
iSHIFT	Click	(0.50, 0.41)	(0.50, 0.41)

Method	Act.	Touch	Lift
GT	Click	(0.49, 0.11)	(0.49, 0.11)
iSHIFT	Click	(0.50, 0.09)	(0.50, 0.09)

Method	Act.	Touch	Lift
GT	TYPE	-	-
iSHIFT	TYPE	-	-

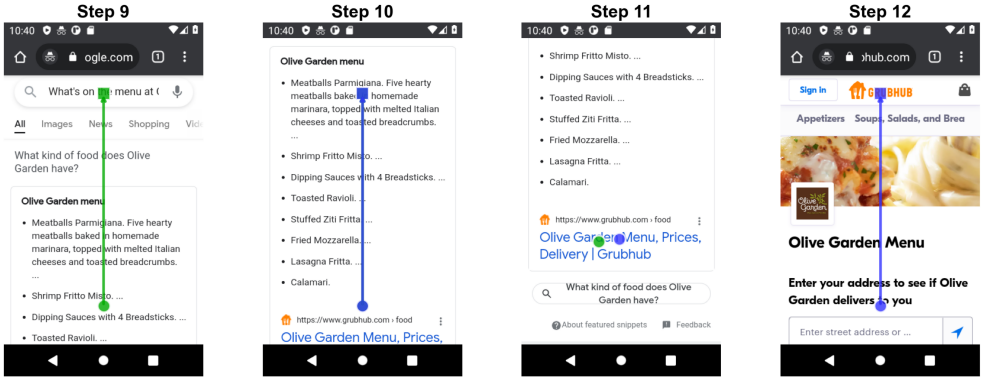
Figure S.14. Episodes from the AITW dataset comparing iSHIFT trajectories with the ground-truth demonstrations. iSHIFT closely follows the intended interaction flow, and when deviations occur, the model selects alternative but valid action sequences that still achieve the correct task outcome.

Goal: What's on the menu at Olive Garden?
Steps 5 to 8



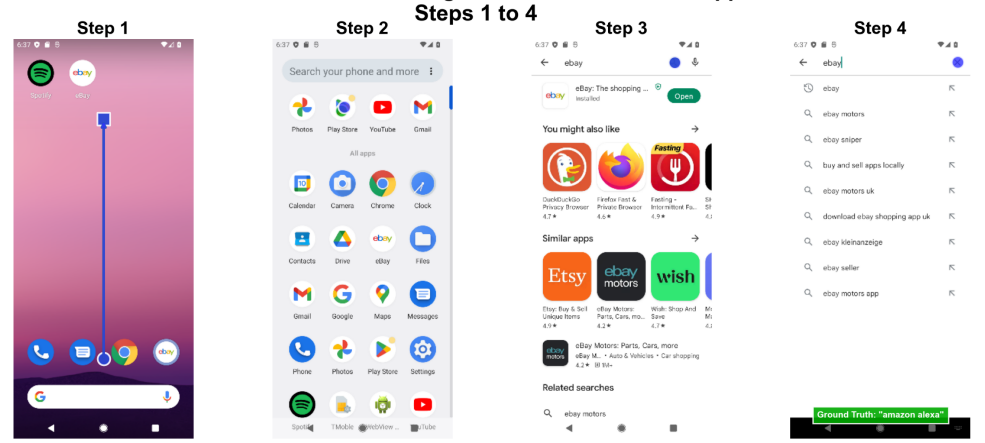
Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift
GT	Enter	-	-	GT	Swipe	(0.50, 0.80)	(0.50, 0.20)	GT	Click	(0.47, 0.58)	(0.47, 0.58)	GT	Click	(0.64, 0.38)	(0.64, 0.38)
iSHIFT	Click	(0.62, 0.22)	(0.62, 0.22)	iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)	iSHIFT	Click	(0.51, 0.57)	(0.51, 0.57)	iSHIFT	Comp.	-	-

Goal: What's on the menu at Olive Garden?
Steps 9 to 12



Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)	GT	Swipe	(0.50, 0.80)	(0.50, 0.20)	GT	Click	(0.39, 0.62)	(0.39, 0.62)	GT	Comp.	-	-
iSHIFT	Comp.	-	-	iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)	iSHIFT	Click	(0.49, 0.61)	(0.49, 0.61)	iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)

Goal: Check the settings for the Amazon Alexa app
Steps 1 to 4



Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)	GT	Click	(0.38, 0.18)	(0.38, 0.18)	GT	Click	(0.79, 0.06)	(0.79, 0.06)	GT	TYPE	-	-
iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)	iSHIFT	Click	(0.40, 0.17)	(0.40, 0.17)	iSHIFT	Click	(0.80, 0.06)	(0.80, 0.06)	iSHIFT	Click	(0.93, 0.06)	(0.93, 0.06)

Figure S.15. Episodes from the AITW dataset comparing iSHIFT trajectories with the ground-truth demonstrations. iSHIFT closely follows the intended interaction flow, and when deviations occur, the model selects alternative but valid action sequences that still achieve the correct task outcome.

Goal: Check the settings for the Amazon Alexa app
Steps 5 to 8

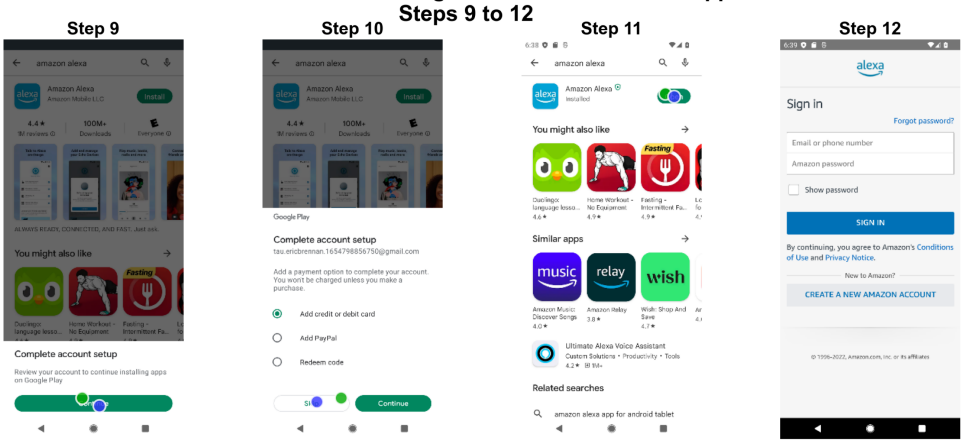
Method	Act.	Touch	Lift
GT	Click	(0.94, 0.06)	(0.94, 0.06)
iSHIFT	Enter	-	-

Method	Act.	Touch	Lift
GT	TYPE	-	-
iSHIFT	TYPE	-	-

Method	Act.	Touch	Lift
GT	Click	(0.61, 0.14)	(0.61, 0.14)
iSHIFT	Click	(0.37, 0.12)	(0.37, 0.12)

Method	Act.	Touch	Lift
GT	Click	(0.79, 0.13)	(0.79, 0.13)
iSHIFT	Click	(0.61, 0.14)	(0.61, 0.14)

Goal: Check the settings for the Amazon Alexa app
Steps 9 to 12



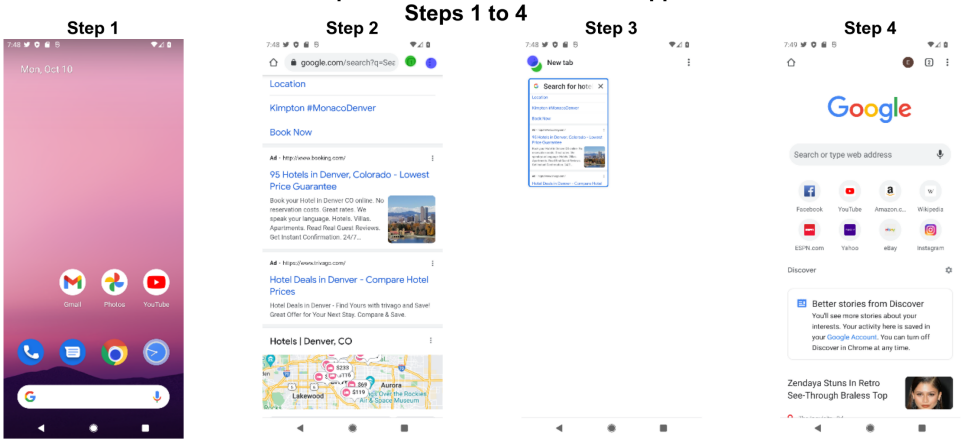
Method	Act.	Touch	Lift
GT	Click	(0.44, 0.90)	(0.44, 0.90)
iSHIFT	Click	(0.53, 0.92)	(0.53, 0.92)

Method	Act.	Touch	Lift
GT	Click	(0.44, 0.90)	(0.44, 0.90)
iSHIFT	Click	(0.30, 0.91)	(0.30, 0.91)

Method	Act.	Touch	Lift
GT	Click	(0.80, 0.13)	(0.80, 0.13)
iSHIFT	Click	(0.85, 0.14)	(0.85, 0.14)

Method	Act.	Touch	Lift
GT	Comp.	-	-
iSHIFT	Comp.	-	-

Goal: Open a new tab in the chrome app
Steps 1 to 4



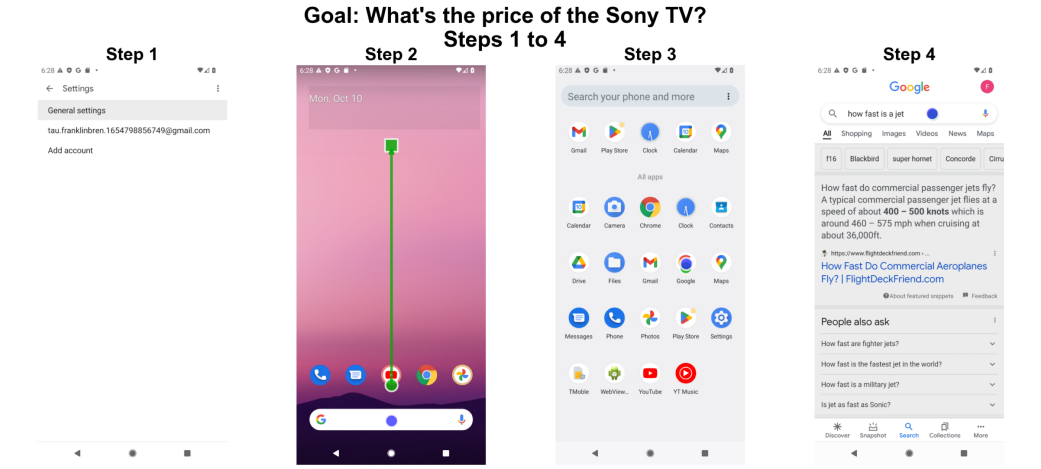
Method	Act.	Touch	Lift
GT	Click	(0.62, 0.79)	(0.62, 0.79)
iSHIFT	Click	(0.62, 0.79)	(0.62, 0.79)

Method	Act.	Touch	Lift
GT	Click	(0.82, 0.05)	(0.82, 0.05)
iSHIFT	Click	(0.93, 0.06)	(0.93, 0.06)

Method	Act.	Touch	Lift
GT	Click	(0.08, 0.07)	(0.08, 0.07)
iSHIFT	Click	(0.06, 0.05)	(0.06, 0.05)

Method	Act.	Touch	Lift
GT	Comp.	-	-
iSHIFT	Comp.	-	-

Figure S.16. Episodes from the AITW dataset comparing iSHIFT trajectories with the ground-truth demonstrations. iSHIFT closely follows the intended interaction flow, and when deviations occur, the model selects alternative but valid action sequences that still achieve the correct task outcome.

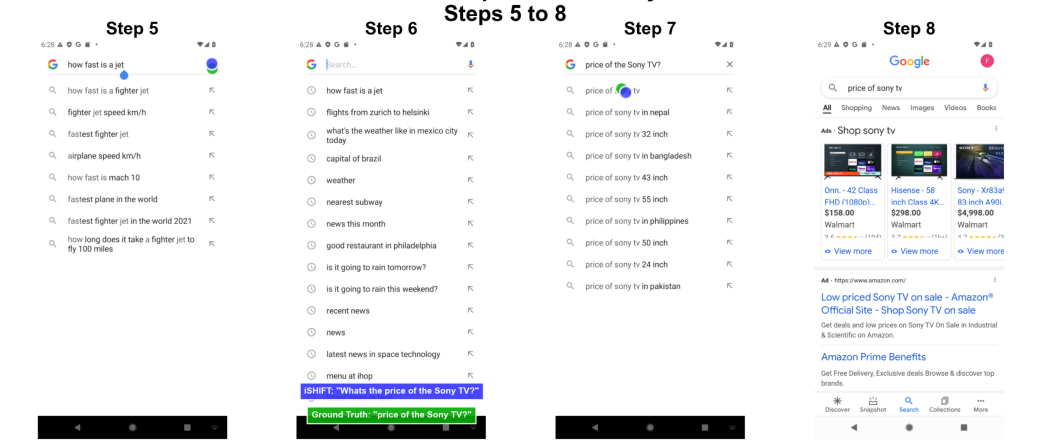


Method	Act.	Touch	Lift
GT	Home	-	-
iSHIFT	Home	-	-

Method	Act.	Touch	Lift
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)
iSHIFT	Click	(0.50, 0.89)	(0.50, 0.89)

Method	Act.	Touch	Lift
GT	Click	(0.69, 0.51)	(0.69, 0.51)
iSHIFT	Click	(0.69, 0.50)	(0.69, 0.50)

Method	Act.	Touch	Lift
GT	Click	(0.62, 0.12)	(0.62, 0.12)
iSHIFT	Click	(0.62, 0.12)	(0.62, 0.12)

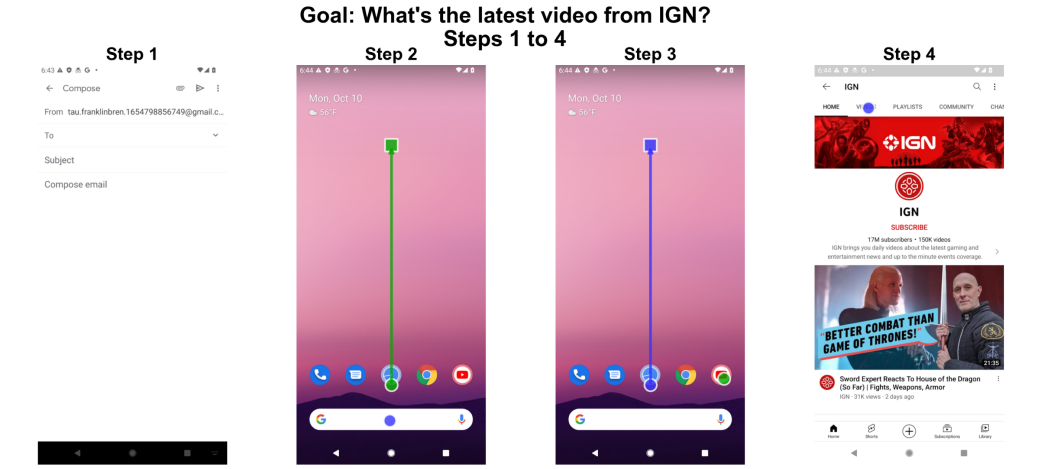


Method	Act.	Touch	Lift
GT	Click	(0.92, 0.07)	(0.92, 0.07)
iSHIFT	Click	(0.92, 0.06)	(0.92, 0.06)

Method	Act.	Touch	Lift
GT	TYPE	-	-
iSHIFT	TYPE	-	-

Method	Act.	Touch	Lift
GT	Click	(0.35, 0.12)	(0.35, 0.12)
iSHIFT	Click	(0.37, 0.13)	(0.37, 0.13)

Method	Act.	Touch	Lift
GT	Comp.	-	-
iSHIFT	Comp.	-	-



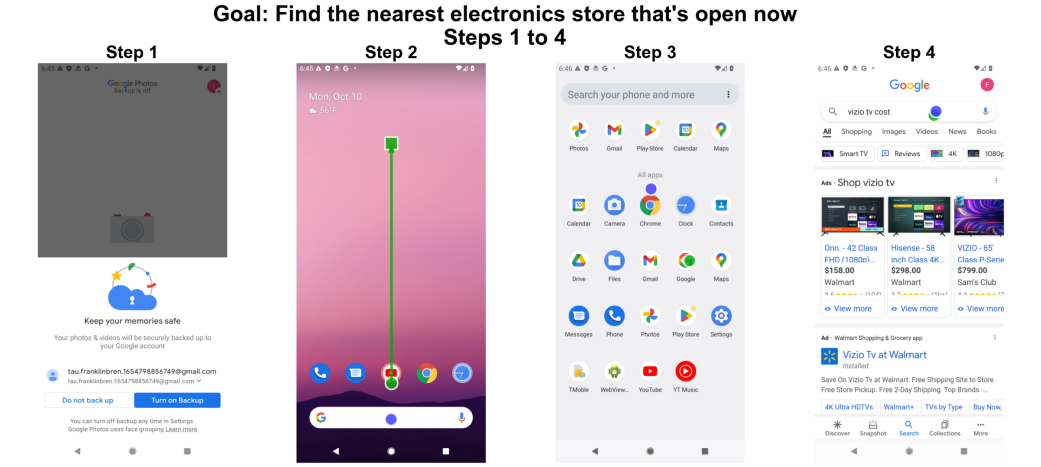
Method	Act.	Touch	Lift
GT	Home	-	-
iSHIFT	Home	-	-

Method	Act.	Touch	Lift
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)
iSHIFT	Click	(0.49, 0.89)	(0.49, 0.89)

Method	Act.	Touch	Lift
GT	Click	(0.89, 0.78)	(0.89, 0.78)
iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)

Method	Act.	Touch	Lift
GT	Comp.	-	-
iSHIFT	Click	(0.28, 0.11)	(0.28, 0.11)

Figure S.17. Episodes from the AITW dataset comparing iSHIFT trajectories with the ground-truth demonstrations. iSHIFT closely follows the intended interaction flow, and when deviations occur, the model selects alternative but valid action sequences that still achieve the correct task outcome.

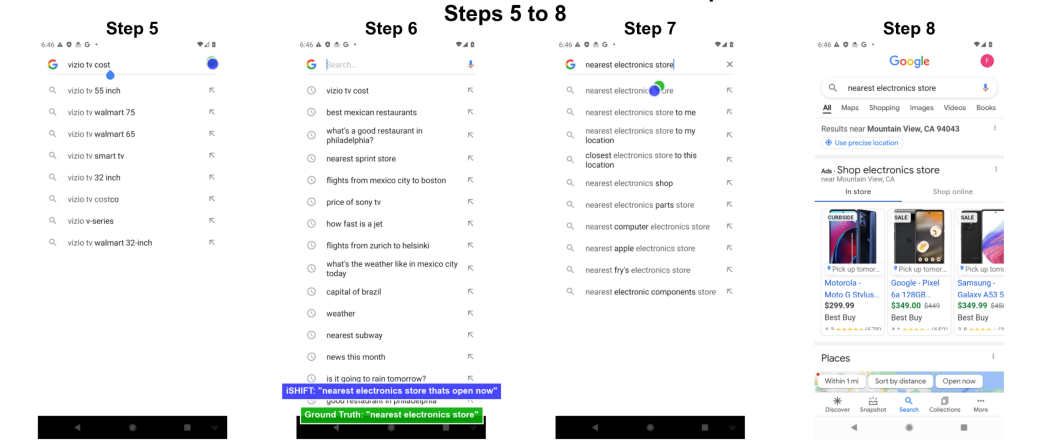


Method	Act.	Touch	Lift
GT	Home	-	-
iSHIFT	Home	-	-

Method	Act.	Touch	Lift
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)
iSHIFT	Click	(0.50, 0.89)	(0.50, 0.89)

Method	Act.	Touch	Lift
GT	Click	(0.70, 0.49)	(0.70, 0.49)
iSHIFT	Click	(0.50, 0.31)	(0.50, 0.31)

Method	Act.	Touch	Lift
GT	Click	(0.63, 0.13)	(0.63, 0.13)
iSHIFT	Click	(0.64, 0.12)	(0.64, 0.12)

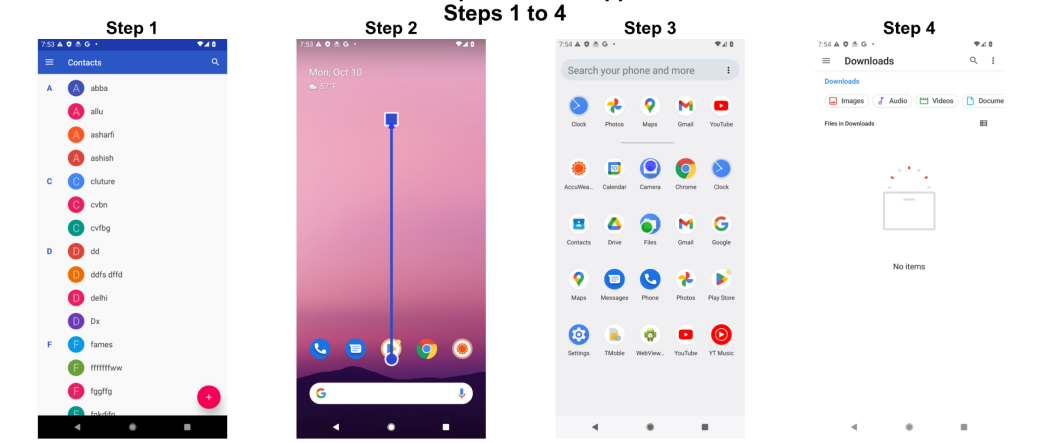


Method	Act.	Touch	Lift
GT	Click	(0.92, 0.06)	(0.92, 0.06)
iSHIFT	Click	(0.92, 0.06)	(0.92, 0.06)

Method	Act.	Touch	Lift
GT	TYPE	-	-
iSHIFT	TYPE	-	-

Method	Act.	Touch	Lift
GT	Click	(0.54, 0.12)	(0.54, 0.12)
iSHIFT	Click	(0.52, 0.13)	(0.52, 0.13)

Method	Act.	Touch	Lift
GT	Comp.	-	-
iSHIFT	Comp.	-	-



Method	Act.	Touch	Lift
GT	Home	-	-
iSHIFT	Home	-	-

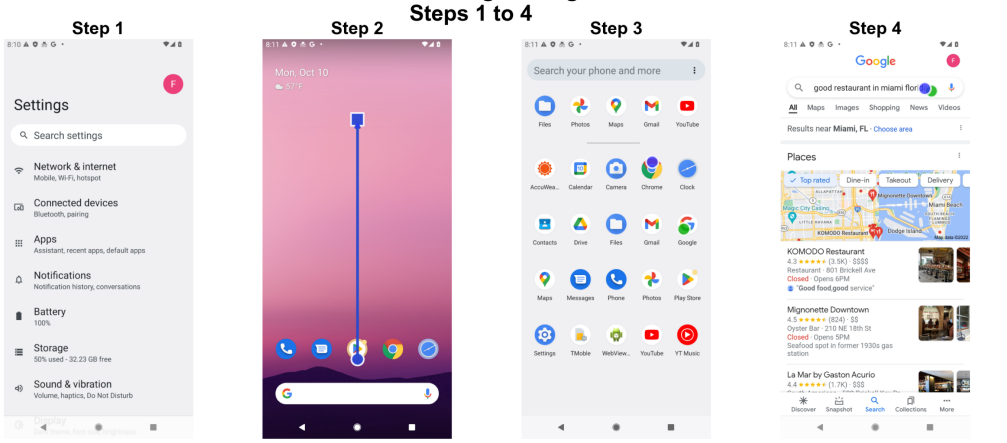
Method	Act.	Touch	Lift
GT	Swipe	(0.50, 0.80)	(0.50, 0.20)
iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)

Method	Act.	Touch	Lift
GT	Click	(0.48, 0.47)	(0.48, 0.47)
iSHIFT	Click	(0.50, 0.32)	(0.50, 0.32)

Method	Act.	Touch	Lift
GT	Comp.	-	-
iSHIFT	Comp.	-	-

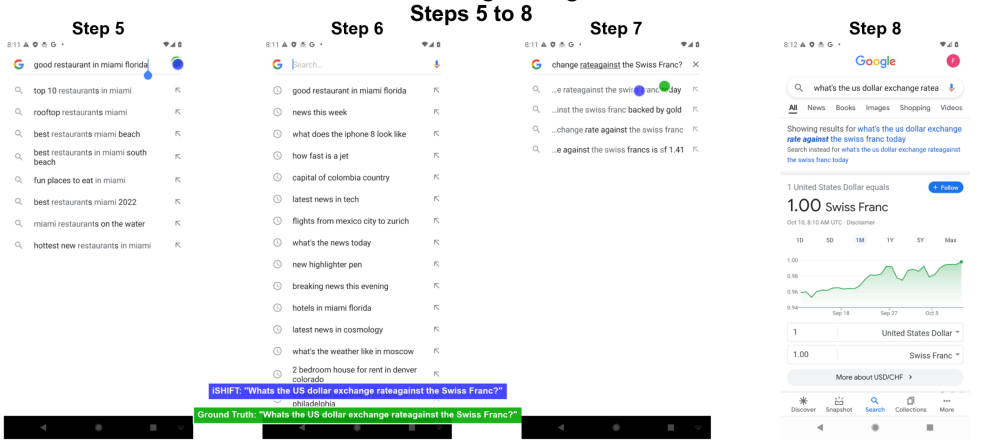
Figure S.18. Episodes from the AITW dataset comparing iSHIFT trajectories with the ground-truth demonstrations. iSHIFT closely follows the intended interaction flow, and when deviations occur, the model selects alternative but valid action sequences that still achieve the correct task outcome.

Goal: What's the US dollar exchange rateagainst the Swiss Franc?
Steps 1 to 4



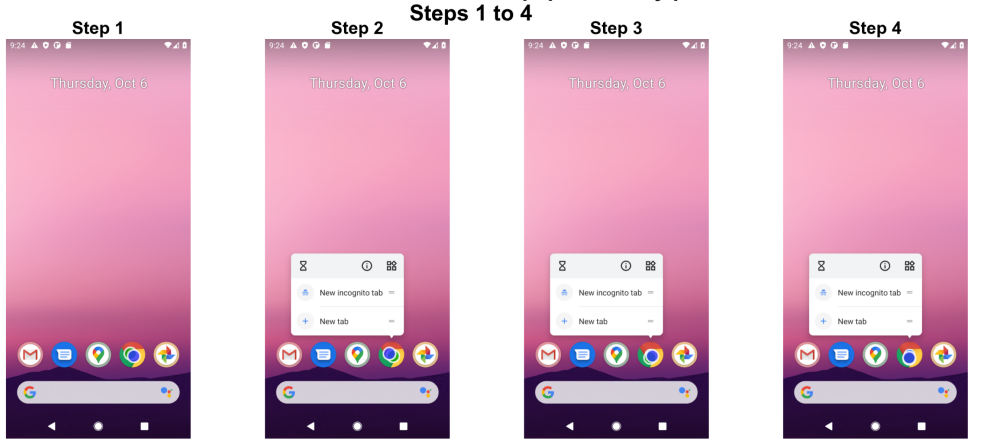
Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift
GT	Home	-	-	GT	Swipe	(0.50, 0.80)	(0.50, 0.20)	GT	Click	(0.85, 0.47)	(0.85, 0.47)	GT	Click	(0.79, 0.13)	(0.79, 0.13)
iSHIFT	Home	-	-	iSHIFT	Swipe	(0.50, 0.80)	(0.50, 0.20)	iSHIFT	Click	(0.69, 0.31)	(0.69, 0.31)	iSHIFT	Click	(0.76, 0.12)	(0.76, 0.12)

Goal: What's the US dollar exchange rateagainst the Swiss Franc?
Steps 5 to 8



Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift
GT	Click	(0.91, 0.05)	(0.91, 0.05)	GT	TYPE	-	-	GT	Click	(0.75, 0.12)	(0.75, 0.12)	GT	Comp.	-	-
iSHIFT	Click	(0.92, 0.06)	(0.92, 0.06)	iSHIFT	TYPE	-	-	iSHIFT	Click	(0.62, 0.13)	(0.62, 0.13)	iSHIFT	Comp.	-	-

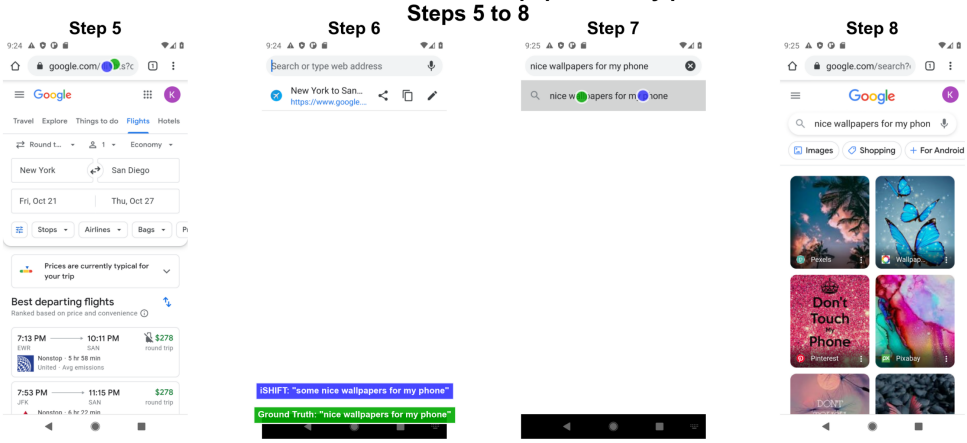
Goal: Show me some nice wallpapers for my phone
Steps 1 to 4



Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift	Method	Act.	Touch	Lift
GT	Click	(0.67, 0.79)	(0.67, 0.79)	GT	Click	(0.69, 0.79)	(0.69, 0.79)	GT	Click	(0.68, 0.80)	(0.68, 0.80)	GT	Click	(0.68, 0.80)	(0.68, 0.80)
iSHIFT	Click	(0.69, 0.79)	(0.69, 0.79)	iSHIFT	Click	(0.67, 0.79)	(0.67, 0.79)	iSHIFT	Click	(0.69, 0.79)	(0.69, 0.79)	iSHIFT	Click	(0.68, 0.80)	(0.68, 0.80)

Figure S.19. Episodes from the AITW dataset comparing iSHIFT trajectories with the ground-truth demonstrations. iSHIFT closely follows the intended interaction flow, and when deviations occur, the model selects alternative but valid action sequences that still achieve the correct task outcome.

Goal: Show me some nice wallpapers for my phone
Steps 5 to 8



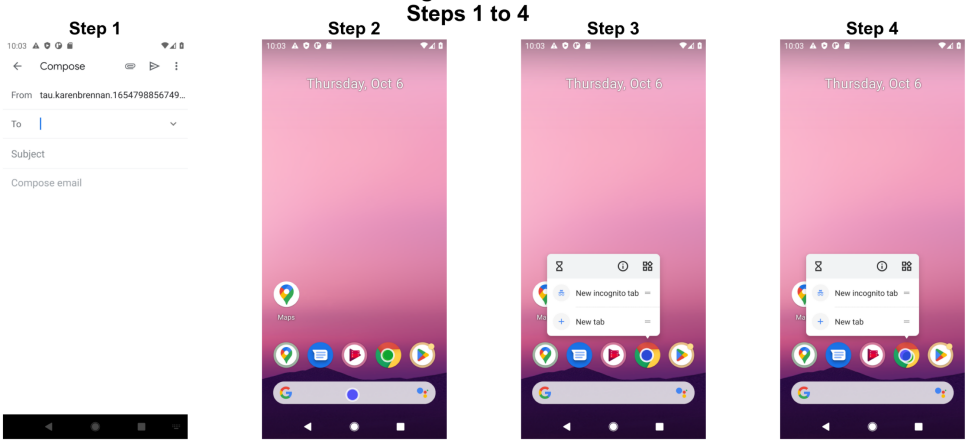
Method	Act.	Touch	Lift
GT	Click	(0.60, 0.06)	(0.60, 0.06)
iSHIFT	Click	(0.56, 0.07)	(0.56, 0.07)

Method	Act.	Touch	Lift
GT	TYPE	-	-
iSHIFT	TYPE	-	-

Method	Act.	Touch	Lift
GT	Click	(0.33, 0.14)	(0.33, 0.14)
iSHIFT	Click	(0.66, 0.14)	(0.66, 0.14)

Method	Act.	Touch	Lift
GT	Comp.	-	-
iSHIFT	Comp.	-	-

Goal: How do I get to the nearest IKEA?
Steps 1 to 4



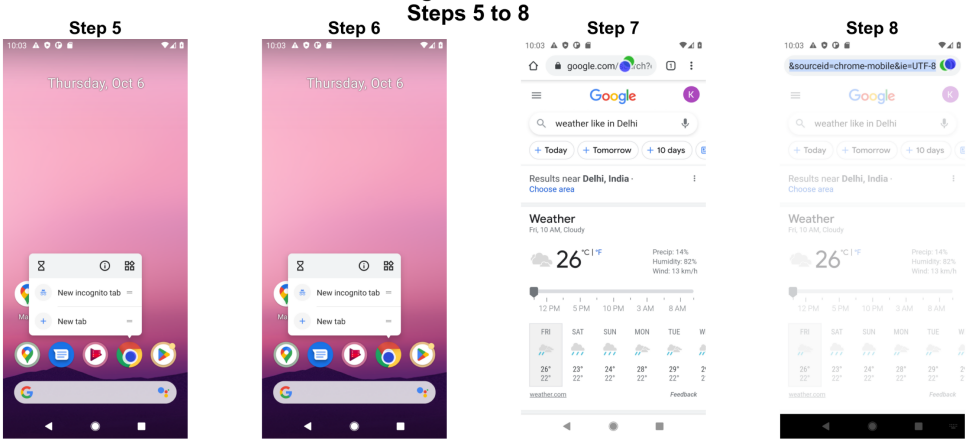
Method	Act.	Touch	Lift
GT	Home	-	-
iSHIFT	Home	-	-

Method	Act.	Touch	Lift
GT	Click	(0.68, 0.79)	(0.68, 0.79)
iSHIFT	Click	(0.49, 0.89)	(0.49, 0.89)

Method	Act.	Touch	Lift
GT	Click	(0.68, 0.79)	(0.68, 0.79)
iSHIFT	Click	(0.68, 0.79)	(0.68, 0.79)

Method	Act.	Touch	Lift
GT	Click	(0.69, 0.79)	(0.69, 0.79)
iSHIFT	Click	(0.68, 0.79)	(0.68, 0.79)

Goal: How do I get to the nearest IKEA?
Steps 5 to 8



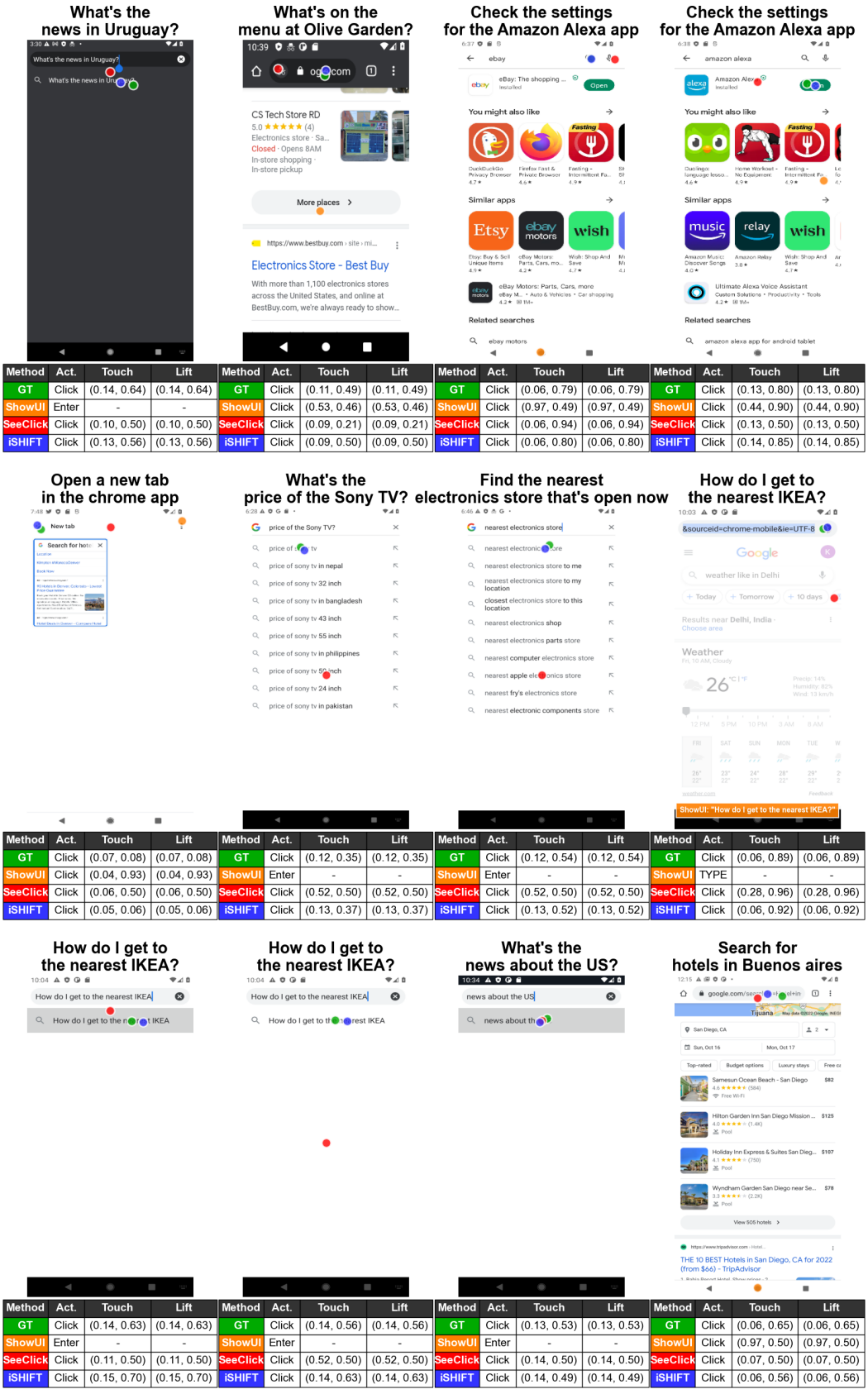
Method	Act.	Touch	Lift
GT	Click	(0.68, 0.80)	(0.68, 0.80)
iSHIFT	Click	(0.69, 0.79)	(0.69, 0.79)

Method	Act.	Touch	Lift
GT	Click	(0.68, 0.80)	(0.68, 0.80)
iSHIFT	Click	(0.68, 0.80)	(0.68, 0.80)

Method	Act.	Touch	Lift
GT	Click	(0.58, 0.06)	(0.58, 0.06)
iSHIFT	Click	(0.56, 0.07)	(0.56, 0.07)

Method	Act.	Touch	Lift
GT	Click	(0.89, 0.06)	(0.89, 0.06)
iSHIFT	Click	(0.92, 0.06)	(0.92, 0.06)

Figure S.20. Episodes from the AITW dataset comparing iSHIFT trajectories with the ground-truth demonstrations. iSHIFT closely follows the intended interaction flow, and when deviations occur, the model selects alternative but valid action sequences that still achieve the correct task outcome.



Method	Act.	Touch	Lift
GT	Click	(0.14, 0.64)	(0.14, 0.64)
ShowUI	Enter	-	-
SeeClick	Click	(0.10, 0.50)	(0.10, 0.50)
iSHIFT	Click	(0.13, 0.56)	(0.13, 0.56)

Method	Act.	Touch	Lift
GT	Click	(0.11, 0.49)	(0.11, 0.49)
ShowUI	Click	(0.53, 0.46)	(0.53, 0.46)
SeeClick	Click	(0.09, 0.21)	(0.09, 0.21)
iSHIFT	Click	(0.09, 0.50)	(0.09, 0.50)

Method	Act.	Touch	Lift
GT	Click	(0.06, 0.79)	(0.06, 0.79)
ShowUI	Click	(0.97, 0.49)	(0.97, 0.49)
SeeClick	Click	(0.06, 0.94)	(0.06, 0.94)
iSHIFT	Click	(0.06, 0.80)	(0.06, 0.80)

Method	Act.	Touch	Lift
GT	Click	(0.13, 0.80)	(0.13, 0.80)
ShowUI	Click	(0.44, 0.90)	(0.44, 0.90)
SeeClick	Click	(0.13, 0.50)	(0.13, 0.50)
iSHIFT	Click	(0.14, 0.85)	(0.14, 0.85)

Method	Act.	Touch	Lift
GT	Click	(0.07, 0.08)	(0.07, 0.08)
ShowUI	Click	(0.04, 0.93)	(0.04, 0.93)
SeeClick	Click	(0.06, 0.50)	(0.06, 0.50)
iSHIFT	Click	(0.05, 0.06)	(0.05, 0.06)

Method	Act.	Touch	Lift
GT	Click	(0.12, 0.35)	(0.12, 0.35)
ShowUI	Enter	-	-
SeeClick	Click	(0.52, 0.50)	(0.52, 0.50)
iSHIFT	Click	(0.13, 0.37)	(0.13, 0.37)

Method	Act.	Touch	Lift
GT	Click	(0.12, 0.54)	(0.12, 0.54)
ShowUI	Enter	-	-
SeeClick	Click	(0.52, 0.50)	(0.52, 0.50)
iSHIFT	Click	(0.13, 0.52)	(0.13, 0.52)

Method	Act.	Touch	Lift
GT	Click	(0.06, 0.89)	(0.06, 0.89)
ShowUI	TYPE	-	-
SeeClick	Click	(0.28, 0.96)	(0.28, 0.96)
iSHIFT	Click	(0.06, 0.92)	(0.06, 0.92)

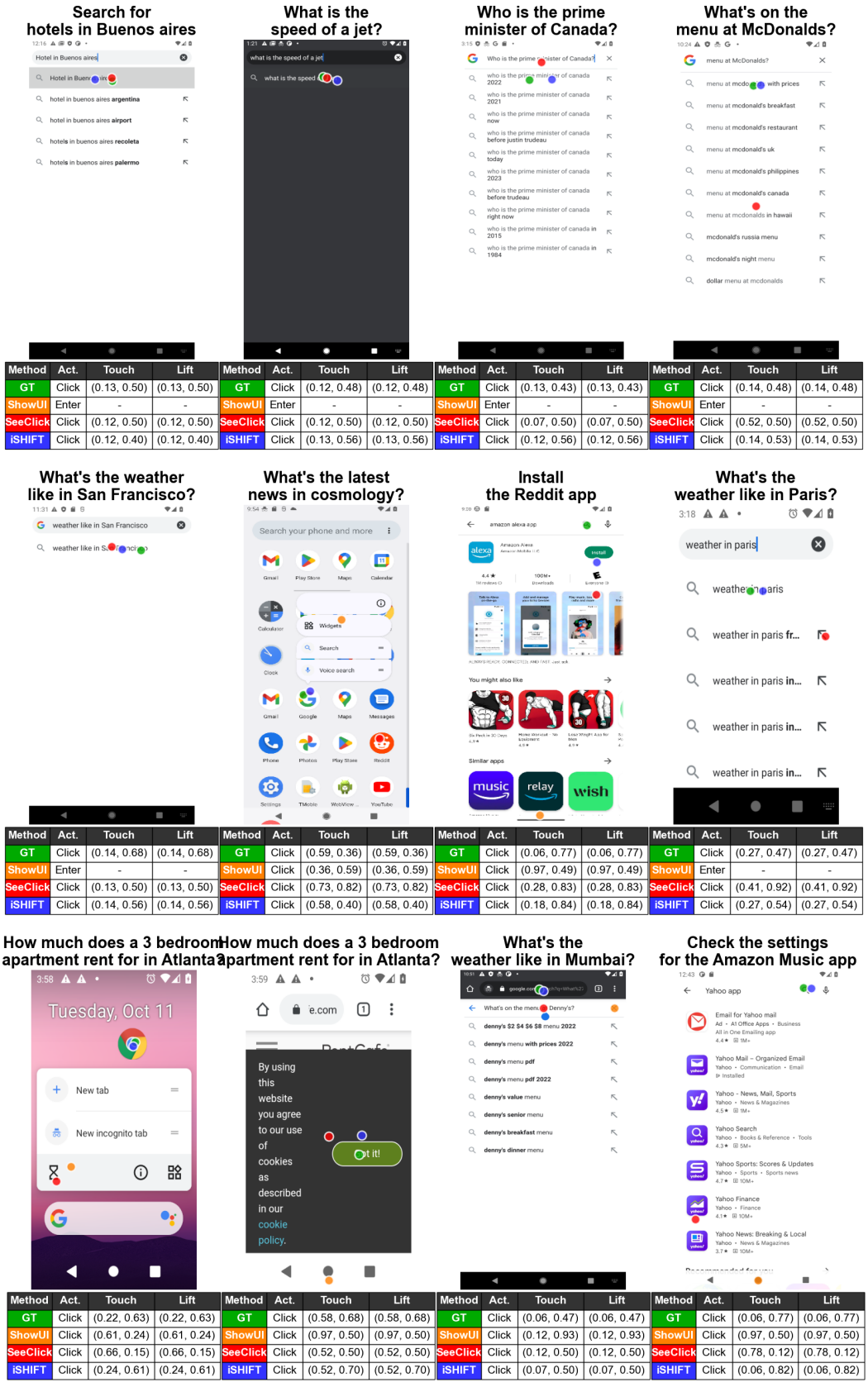
Method	Act.	Touch	Lift
GT	Click	(0.14, 0.63)	(0.14, 0.63)
ShowUI	Enter	-	-
SeeClick	Click	(0.11, 0.50)	(0.11, 0.50)
iSHIFT	Click	(0.15, 0.70)	(0.15, 0.70)

Method	Act.	Touch	Lift
GT	Click	(0.14, 0.56)	(0.14, 0.56)
ShowUI	Enter	-	-
SeeClick	Click	(0.52, 0.50)	(0.52, 0.50)
iSHIFT	Click	(0.14, 0.63)	(0.14, 0.63)

Method	Act.	Touch	Lift
GT	Click	(0.13, 0.53)	(0.13, 0.53)
ShowUI	Enter	-	-
SeeClick	Click	(0.14, 0.50)	(0.14, 0.50)
iSHIFT	Click	(0.14, 0.49)	(0.14, 0.49)

Method	Act.	Touch	Lift
GT	Click	(0.06, 0.65)	(0.06, 0.65)
ShowUI	Click	(0.97, 0.50)	(0.97, 0.50)
SeeClick	Click	(0.07, 0.50)	(0.07, 0.50)
iSHIFT	Click	(0.06, 0.56)	(0.06, 0.56)

Figure S.21. Qualitative comparison between iSHIFT, ShowUI and SeeClick on the AITW dataset. iSHIFT consistently selects correct interaction steps and produces stable trajectories across diverse task scenarios, outperforming competing approaches.



Method	Act.	Touch	Lift
GT	Click	(0.13, 0.50)	(0.13, 0.50)
ShowUI	Enter	-	-
SeeClick	Click	(0.12, 0.50)	(0.12, 0.50)
iSHIFT	Click	(0.12, 0.40)	(0.12, 0.40)

Method	Act.	Touch	Lift
GT	Click	(0.12, 0.48)	(0.12, 0.48)
ShowUI	Enter	-	-
SeeClick	Click	(0.12, 0.50)	(0.12, 0.50)
iSHIFT	Click	(0.13, 0.56)	(0.13, 0.56)

Method	Act.	Touch	Lift
GT	Click	(0.13, 0.43)	(0.13, 0.43)
ShowUI	Enter	-	-
SeeClick	Click	(0.07, 0.50)	(0.07, 0.50)
iSHIFT	Click	(0.12, 0.56)	(0.12, 0.56)

Method	Act.	Touch	Lift
GT	Click	(0.14, 0.48)	(0.14, 0.48)
ShowUI	Enter	-	-
SeeClick	Click	(0.52, 0.50)	(0.52, 0.50)
iSHIFT	Click	(0.14, 0.53)	(0.14, 0.53)

Method	Act.	Touch	Lift
GT	Click	(0.14, 0.68)	(0.14, 0.68)
ShowUI	Enter	-	-
SeeClick	Click	(0.13, 0.50)	(0.13, 0.50)
iSHIFT	Click	(0.14, 0.56)	(0.14, 0.56)

Method	Act.	Touch	Lift
GT	Click	(0.59, 0.36)	(0.59, 0.36)
ShowUI	Click	(0.36, 0.59)	(0.36, 0.59)
SeeClick	Click	(0.73, 0.82)	(0.73, 0.82)
iSHIFT	Click	(0.58, 0.40)	(0.58, 0.40)

Method	Act.	Touch	Lift
GT	Click	(0.06, 0.77)	(0.06, 0.77)
ShowUI	Click	(0.97, 0.49)	(0.97, 0.49)
SeeClick	Click	(0.28, 0.83)	(0.28, 0.83)
iSHIFT	Click	(0.18, 0.84)	(0.18, 0.84)

Method	Act.	Touch	Lift
GT	Click	(0.27, 0.47)	(0.27, 0.47)
ShowUI	Enter	-	-
SeeClick	Click	(0.41, 0.92)	(0.41, 0.92)
iSHIFT	Click	(0.27, 0.54)	(0.27, 0.54)

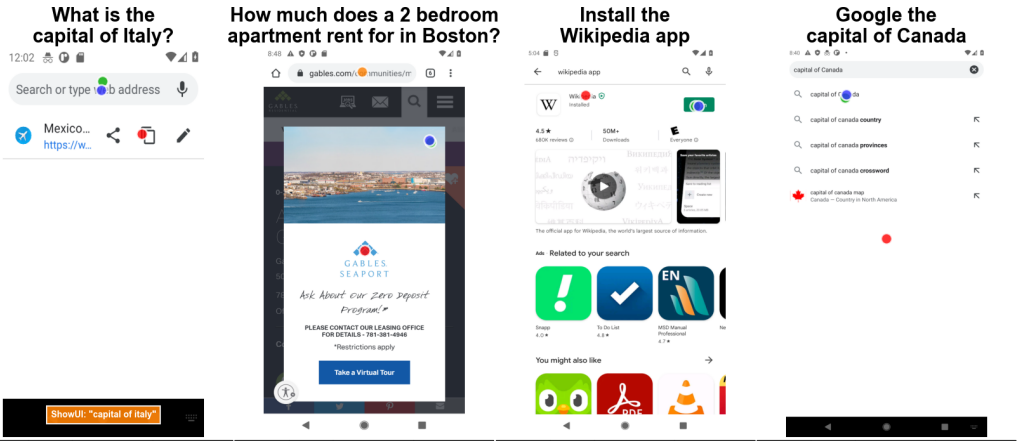
Method	Act.	Touch	Lift
GT	Click	(0.22, 0.63)	(0.22, 0.63)
ShowUI	Click	(0.61, 0.24)	(0.61, 0.24)
SeeClick	Click	(0.66, 0.15)	(0.66, 0.15)
iSHIFT	Click	(0.24, 0.61)	(0.24, 0.61)

Method	Act.	Touch	Lift
GT	Click	(0.58, 0.68)	(0.58, 0.68)
ShowUI	Click	(0.97, 0.50)	(0.97, 0.50)
SeeClick	Click	(0.52, 0.50)	(0.52, 0.50)
iSHIFT	Click	(0.52, 0.70)	(0.52, 0.70)

Method	Act.	Touch	Lift
GT	Click	(0.06, 0.47)	(0.06, 0.47)
ShowUI	Click	(0.12, 0.93)	(0.12, 0.93)
SeeClick	Click	(0.12, 0.50)	(0.12, 0.50)
iSHIFT	Click	(0.07, 0.50)	(0.07, 0.50)

Method	Act.	Touch	Lift
GT	Click	(0.06, 0.77)	(0.06, 0.77)
ShowUI	Click	(0.97, 0.50)	(0.97, 0.50)
SeeClick	Click	(0.78, 0.12)	(0.78, 0.12)
iSHIFT	Click	(0.06, 0.82)	(0.06, 0.82)

Figure S.22. Qualitative comparison between iSHIFT, ShowUI and SeeClick on the AITW dataset. iSHIFT consistently selects correct interaction steps and produces stable trajectories across diverse task scenarios, outperforming competing approaches.

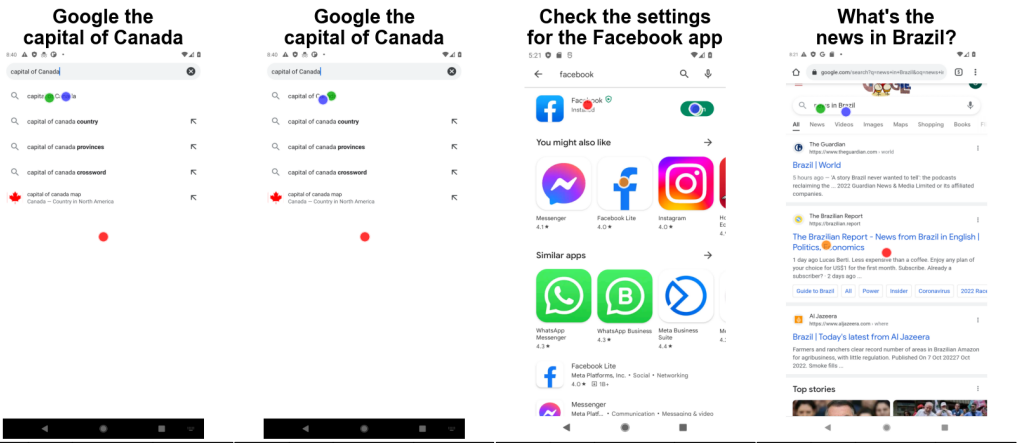


Method	Act.	Touch	Lift
GT	Click	(0.09, 0.50)	(0.09, 0.50)
ShowUI	TYPE	-	-
SeeClick	Click	(0.22, 0.69)	(0.22, 0.69)
iSHIFT	Click	(0.11, 0.49)	(0.11, 0.49)

Method	Act.	Touch	Lift
GT	Click	(0.24, 0.83)	(0.24, 0.83)
ShowUI	Click	(0.06, 0.49)	(0.06, 0.49)
SeeClick	Click	(0.52, 0.50)	(0.52, 0.50)
iSHIFT	Click	(0.24, 0.82)	(0.24, 0.82)

Method	Act.	Touch	Lift
GT	Click	(0.15, 0.85)	(0.15, 0.85)
ShowUI	Comp.	-	-
SeeClick	Click	(0.12, 0.30)	(0.12, 0.30)
iSHIFT	Click	(0.15, 0.87)	(0.15, 0.87)

Method	Act.	Touch	Lift
GT	Click	(0.13, 0.29)	(0.13, 0.29)
ShowUI	Enter	-	-
SeeClick	Click	(0.49, 0.50)	(0.49, 0.50)
iSHIFT	Click	(0.12, 0.30)	(0.12, 0.30)

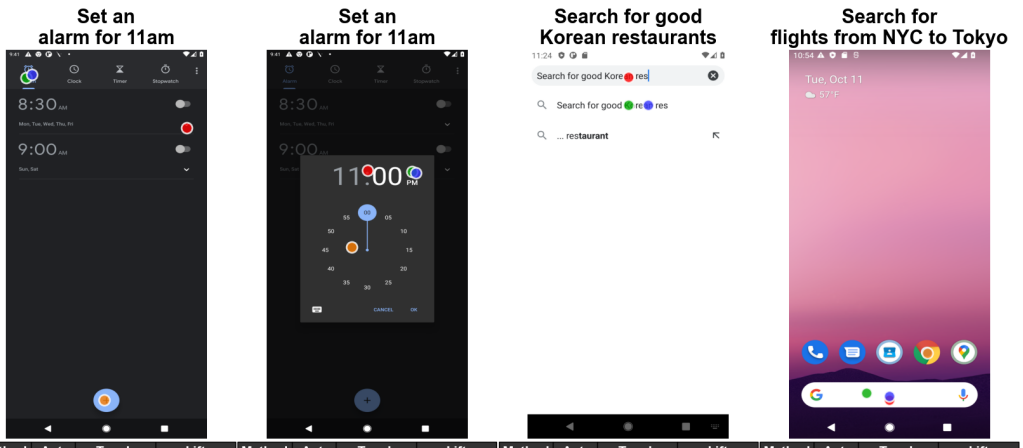


Method	Act.	Touch	Lift
GT	Click	(0.12, 0.23)	(0.12, 0.23)
ShowUI	Enter	-	-
SeeClick	Click	(0.48, 0.50)	(0.48, 0.50)
iSHIFT	Click	(0.12, 0.31)	(0.12, 0.31)

Method	Act.	Touch	Lift
GT	Click	(0.12, 0.33)	(0.12, 0.33)
ShowUI	Enter	-	-
SeeClick	Click	(0.48, 0.50)	(0.48, 0.50)
iSHIFT	Click	(0.13, 0.29)	(0.13, 0.29)

Method	Act.	Touch	Lift
GT	Click	(0.15, 0.85)	(0.15, 0.85)
ShowUI	Click	(0.34, 0.49)	(0.34, 0.49)
SeeClick	Click	(0.14, 0.31)	(0.14, 0.31)
iSHIFT	Click	(0.15, 0.84)	(0.15, 0.84)

Method	Act.	Touch	Lift
GT	Click	(0.15, 0.17)	(0.15, 0.17)
ShowUI	Click	(0.50, 0.20)	(0.50, 0.20)
SeeClick	Click	(0.52, 0.50)	(0.52, 0.50)
iSHIFT	Click	(0.16, 0.30)	(0.16, 0.30)



Method	Act.	Touch	Lift
GT	Click	(0.07, 0.10)	(0.07, 0.10)
ShowUI	Click	(0.90, 0.49)	(0.90, 0.49)
SeeClick	Click	(0.20, 0.90)	(0.20, 0.90)
iSHIFT	Click	(0.06, 0.13)	(0.06, 0.13)

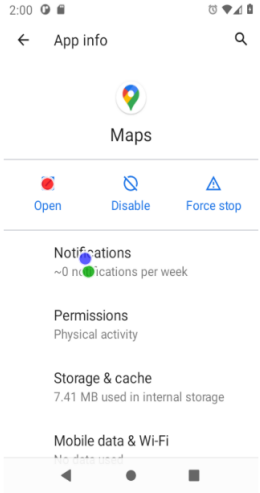
Method	Act.	Touch	Lift
GT	Click	(0.31, 0.72)	(0.31, 0.72)
ShowUI	Click	(0.51, 0.42)	(0.51, 0.42)
SeeClick	Click	(0.31, 0.50)	(0.31, 0.50)
iSHIFT	Click	(0.32, 0.74)	(0.32, 0.74)

Method	Act.	Touch	Lift
GT	Click	(0.14, 0.50)	(0.14, 0.50)
ShowUI	Enter	-	-
SeeClick	Click	(0.07, 0.50)	(0.07, 0.50)
iSHIFT	Click	(0.14, 0.60)	(0.14, 0.60)

Method	Act.	Touch	Lift
GT	Click	(0.88, 0.39)	(0.88, 0.39)
ShowUI	Click	(0.78, 0.69)	(0.78, 0.69)
SeeClick	Click	(0.90, 0.50)	(0.90, 0.50)
iSHIFT	Click	(0.89, 0.50)	(0.89, 0.50)

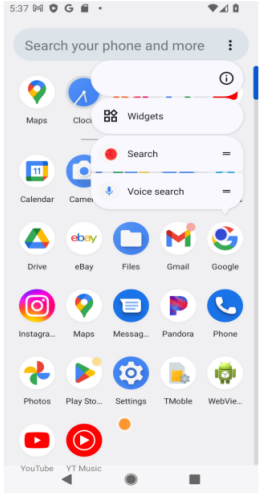
Figure S.23. Qualitative comparison between iSHIFT, ShowUI and SeeClick on the AITW dataset. iSHIFT consistently selects correct interaction steps and produces stable trajectories across diverse task scenarios, outperforming competing approaches.

Check the settings for the Google Maps app



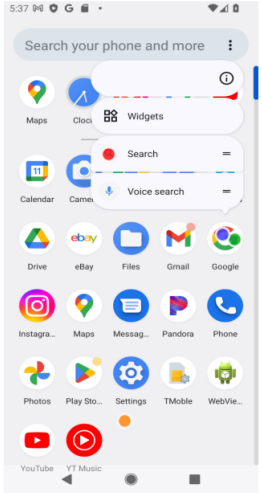
Method	Act.	Touch	Lift
GT	Click	(0.55, 0.33)	(0.55, 0.33)
ShowUI	Comp.	-	-
SeeClick	Click	(0.37, 0.17)	(0.37, 0.17)
iSHIFT	Click	(0.52, 0.32)	(0.52, 0.32)

Search for a new lipgloss



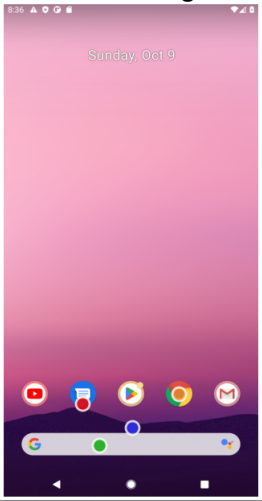
Method	Act.	Touch	Lift
GT	Click	(0.47, 0.85)	(0.47, 0.85)
ShowUI	Click	(0.86, 0.47)	(0.86, 0.47)
SeeClick	Click	(0.31, 0.42)	(0.31, 0.42)
iSHIFT	Click	(0.47, 0.85)	(0.47, 0.85)

Search for a new lipgloss



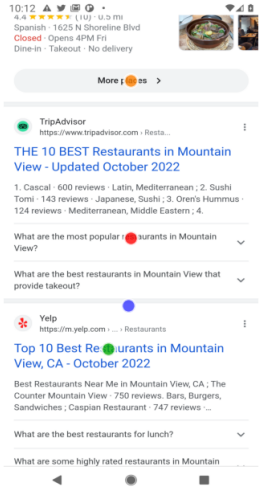
Method	Act.	Touch	Lift
GT	Click	(0.48, 0.90)	(0.48, 0.90)
ShowUI	Click	(0.85, 0.47)	(0.85, 0.47)
SeeClick	Click	(0.31, 0.41)	(0.31, 0.41)
iSHIFT	Click	(0.47, 0.86)	(0.47, 0.86)

What's the price of the Samsung TV?



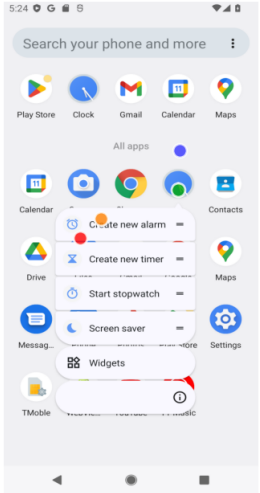
Method	Act.	Touch	Lift
GT	Click	(0.89, 0.38)	(0.89, 0.38)
ShowUI	Click	(0.79, 0.69)	(0.79, 0.69)
SeeClick	Click	(0.81, 0.31)	(0.81, 0.31)
iSHIFT	Click	(0.86, 0.51)	(0.86, 0.51)

What's a good restaurant near me?



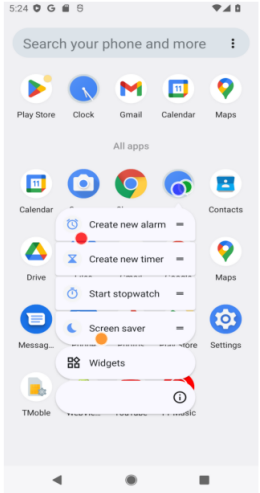
Method	Act.	Touch	Lift
GT	Click	(0.70, 0.41)	(0.70, 0.41)
ShowUI	Click	(0.16, 0.50)	(0.16, 0.50)
SeeClick	Click	(0.48, 0.50)	(0.48, 0.50)
iSHIFT	Click	(0.62, 0.49)	(0.62, 0.49)

Set an alarm for 6pm



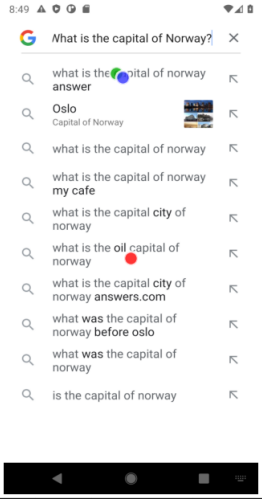
Method	Act.	Touch	Lift
GT	Click	(0.38, 0.68)	(0.38, 0.68)
ShowUI	Click	(0.44, 0.38)	(0.44, 0.38)
SeeClick	Click	(0.48, 0.30)	(0.48, 0.30)
iSHIFT	Click	(0.30, 0.69)	(0.30, 0.69)

Set an alarm for 6pm



Method	Act.	Touch	Lift
GT	Click	(0.37, 0.71)	(0.37, 0.71)
ShowUI	Click	(0.68, 0.38)	(0.68, 0.38)
SeeClick	Click	(0.48, 0.30)	(0.48, 0.30)
iSHIFT	Click	(0.38, 0.68)	(0.38, 0.68)

What is the capital of Norway?



Method	Act.	Touch	Lift
GT	Click	(0.15, 0.44)	(0.15, 0.44)
ShowUI	Enter	-	-
SeeClick	Click	(0.52, 0.50)	(0.52, 0.50)
iSHIFT	Click	(0.15, 0.47)	(0.15, 0.47)

Figure S.24. Qualitative comparison between iSHIFT, ShowUI and SeeClick on the AITW dataset. iSHIFT consistently selects correct interaction steps and produces stable trajectories across diverse task scenarios, outperforming competing approaches.