EARL: Early Intent Recognition in GUI Tasks Using Theory of Mind

Shraddha Pawar [1] **Pramod Kaushik** [1] **Sarath Sivaprasad** [2] **Balavarun Pedapudi** [3] **Mario Fritz** [2]
Shirish Karande [1]

Abstract

Understanding user intent is essential for building better human interaction agents, as it enables personalization, co-creation, and contextual adaptation. However, existing approaches are either restricted to text environments, use human annotation, or just predict future user actions lacking the ability to reason explicitly about user goals. In this work, we introduce EARL (Early Action Reasoning for Latent intent), a theory of mind inspired inference-time algorithm that models user intent as an *inverse planning problem*, inferring latent goals from observed user actions. EARL hypothesizes potential user intent at multiple stages during the course of task execution, enabling timely intervention and personalization. Evaluated on three diverse benchmarks namely Mind2Web, AiTz, and VideoGUI, and using two strong LLMs (Gemini-1.5-Pro and GPT-4o), we show that EARL consistently outperforms CoT-based LLM baselines in accurately deciphering user intent, especially under partial observations.

1 Introduction

Understanding user intent by observing user's actions is key to building effective human-AI interaction systems. As large language models (LLMs) are increasingly deployed as autonomous agents in graphical user interfaces (GUIs), the ability to infer user goals from *partial observations* (early in the trajectory of actions) becomes critical for enabling proactive assistance and collaborative task completion. Recent surveys of GUI agents identify intent modeling as the key missing capability preventing these systems from achieving human-like adaptability (Wang et al., 2024).

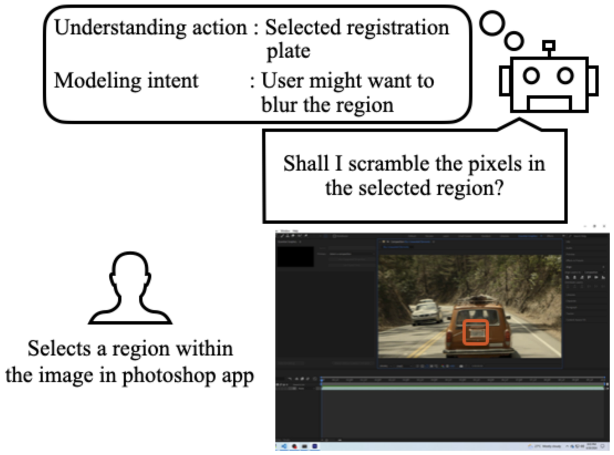

Understanding action : Selected registration plate
Modeling intent : User might want to blur the region
Shall I scramble the pixels in the selected region?
Selects a region within the image in photoshop app

Figure 1: A user opens the photoshop app on a system. Based on this initial action and contextual cues (e.g., the user selecting the number plate of a car), the LLM infers the likely intent: to blur the number plate and proactively suggests the corresponding action. Our method enables such anticipatory behavior by modeling intent from partial observations.

Current approaches to GUI agents suffer from three key limitations. First, most treat user behavior as action sequences to imitate rather than interpreting them to predict the latent intent (Zhang et al., 2024c). Second, methods that do model intent typically require complete task trajectories or predefined goal taxonomies (Wang et al., 2024). Third, even state-of-the-art techniques rely on few-shot prompting with manually curated domain specific examples (Zhang et al., 2024a), limiting their generalization to novel users and contexts. These constraints make existing systems unable to reason about intent during the most practically relevant phase: early interaction when the task is incomplete Figure 1.

As a result, current agents struggle to anticipate intent early and act proactively, which is important in real-world usage. Wang et al. argue (Wang et al., 2024), the field must shift from hand-coded or RL-based black boxes to flexible, interpretable reasoning. Towards filling this gap, we introduce a Theory of Mind (ToM)-inspired approach called EARL (Early Action Reasoning for Latent intent) that formulates

[1] Tata Consultancy Services Research, India [2] CISPA Helmholtz Center for Information Security, Germany [3] IIIT Hyderabad, Hyderabad, India. Correspondence to: Shraddha Pawar <shraddhavijay.pawar@tcs.com>, Pramod Kaushik <pramod.kaushik@tcs.com>, Sarath Sivaprasad <sarath.sivaprasad@cispa.de>, Balavarun Pedapudi <balavarun.pedapudi@research.iiit.ac.in>, Mario Fritz <m.fritz@cispa.de>, Shirish Karande <shirish.karande@tcs.com>.

Workshop on Computer-use Agents @ ICML 2025, Vancouver, Canada. Copyright 2025 by the author(s).

intent recognition as inverse planning problem. During inference, the agent evaluates plausibility of different latent goals that could have generated the observed partial trajectory of actions. With each new state and action observed, the model updates the set of hypothesis about the latent goals and re-samples new ones if necessary. This mechanism enables the tracking of goal states and explicit reasoning on the intent of the user.

To benchmark this method we follow an evaluation scheme where user intent is to be predicted from partial trajectories generated from an unobserved goal. This formulation allows the agent to infer latent user intent from only the early fraction of observed actions. Unlike prior approaches that wait for the full trajectory, this evaluation demands reasoning during the interaction, critical for timely and context-aware adaptation. In short, we treat intent modeling as an inference-time reasoning task with only partial observations.

We evaluate our method across three challenging and diverse benchmarks: Mind2Web, AItW, and VideoGUI, which span desktop and mobile GUIs, open-world web navigation, and multimodal task settings. The evaluation task is: given a fraction of a user's action sequence, the agent is asked to predict the goal that leads to the sequence of actions accurately. Our method outperforms a strong Chain-of-Thought (CoT) baseline in intent prediction accuracy by 2.6–84.1% when given only the early portion (25–75%) of user actions. Beyond accuracy, our method produces more interpretable intent hypotheses and enables earlier model response, demonstrating practical advantages for proactive agent applications. Following are our key contributions:

- We formulate the inverse problem of exploration; intent modeling as an early-stage intent prediction task, where the agent infers latent user goals from partial action trajectories, rather than full execution histories.

- We propose **EARL** (Early Action Reasoning for Latent Intent), an inference-time algorithm that models intent via inverse planning. EARL is entirely dataset-agnostic, zero-shot, and applicable across diverse GUI domains, unlike prior methods that rely on few-shots or environment-specific templates.

- We evaluate EARL on three competitive benchmarks, Mind2Web, AiTz, and VideoGUI, using two strong LLMs (Gemini-1.5-Pro and GPT-4o) as backbones. Compared to CoT-based LLM baseline, EARL consistently improves intent prediction accuracy under partial observations. Specifically, it achieves relative gains in perfect match rate of up to **84.1% at 25% trajectory length** (on VideoGUI), **56.2% at 50%** (on VideoGUI), and **28.9% at 75%** (on AiTz).

2 Related work

Building intelligent human interaction GUI agents requires to go beyond just executing user commands, it demands understanding why a user is performing an action. This insight has driven progress along several complementary fronts. First, modular agent architectures have emerged to handle the complexity of open-ended GUI control, but often remain purely reactive (Wang et al., 2024). Second, LLMs have enabled post hoc summarization of user actions into intentions, yet typically require full trajectories and offer limited real-time support (Zhang et al., 2024a). Third, next-action prediction methods excel at short-horizon reasoning but rarely model high-level goals. Finally, a small but growing body of work has explored early-stage goal inference, often inspired by Theory of Mind, though this remains underdeveloped in HCI and GUI contexts (Rabinowitz et al., 2018).

GUI Agent Architectures for Task Automation. End-to-end GUI agents aim to autonomously execute high-level natural-language instructions by directly interacting with user interfaces. Recent frameworks, such as the unified GUI pipeline by Wang et al. (Wang et al., 2024), employ modular architectures including components like GUI Perceiver, Task Planner, Decision Maker, Memory Retriever, and Executor. Similarly, (Deng et al., 2023a) adopts a two-stage HTML element-ranking approach combined with multi-choice question-answering prompts for complex web navigation tasks. Earlier RL-based agents, such as MiniWoB++ (Liu et al., 2018), have demonstrated effectiveness within controlled browser simulators, but struggle with real-world complexity. However, these architectures primarily focus on maximizing task-completion rates after observing full trajectories, often overlooking explicit user intent modeling. In contrast, our work adds value by embedding an inference-time intention recognition module into these pipelines, enabling goal-aligned reasoning from early task interactions.

LLM-Based Intention Recognition and Summarization. Recent methods have approached intention modeling as summarization tasks, translating entire user-action sequences into succinct natural language intentions. Zhang et al. (Zhang et al., 2024a) introduced SummAct, a hierarchical summarization framework combining sub-goal generation with UI-element attention mechanisms, yielding significant improvements on Mind2Web and MoTIF datasets. Similarly, Ahmed al (Ahmed, 2024) developed Mistral-Intention, fine-tuning LLMs with keyword extraction losses to better capture essential action details, and demonstrated improved performance on diverse GUI environments. However, these methods depend heavily on full trajectory information with human annotation and infer intent post hoc, limiting real-time applicability. Our approach enhances this dimension by framing intention recognition as an inverse planning task, allowing accurate intention inference using only partial action observations. Our approach also does not use any human annotation particular to a dataset making it dataset agnostic.

Next-Action Prediction with LLMs. Next-action prediction tasks condition on previous user interactions and current GUI states to anticipate immediate subsequent actions. Prominent models like SYNAPSE (Zheng et al., 2023), DroidBot-GPT (Wen et al., 2023), and AutoDroid (Wen et al., 2024) employ LLMs enhanced by state abstraction and trajectory-based prompting to achieve high accuracy in predicting immediate next actions. Despite their effectiveness, these methods typically focus on short-term, step-by-step prediction without explicitly modeling broader user goals. Our proposed method significantly differs by leveraging early-stage latent intent inference, thus facilitating not only accurate next-action predictions but also enabling proactive, contextually relevant interventions.

3 TOM guided intent modelling

Accurate intent recognition in GUI interactions hinges on an agent's ability to reason about latent goals from partial and noisy user actions. Traditional approaches are often reliant on post hoc analysis of complete trajectories or rigid goal taxonomies fail to address the dynamic nature of real-world interactions, where users reveal their intentions incrementally. Inspired by human cognitive processes, we frame intent recognition as an inverse planning problem under the Theory of Mind (ToM) framework (Baker et al., 2017). In this framework mental state attribution is treated as performing probabilistic inference using a generative model of a rational agent. An agent's perceptions combined with its prior beliefs determine its current beliefs, and those beliefs together with its goals drive its actions. Consequently, one can recover an agent's beliefs and goals by (a) simulating belief updates forward from its observations and priors, (b) inferring them backward from its observed actions, or (c) jointly integrating both observation-based and action-based information.

3.1 Methodology

In this method we adapt the particle-filter thought-tracing algorithm of Kim et al. (Kim et al., 2025) to infer a user's latent goals in GUI environments, formalized in Algorithm 1. We instantiate this algorithm in a GUI intent inference algorithm called **EARL** (*Early Action Reasoning for Latent Intent*), which maintains a belief distribution over possible user goals as the user interacts with the interface. Each candidate goal is represented as a natural language hypothesis, treated as a weighted "particle" that evolves over time. Given an input trajectory $E = \{(s_1, a_1), \ldots, (s_T, a_T)\}$, where s_t and a_t denote the GUI screen and action at step t, the agent begins by invoking INITIALIZE to generate a diverse hypothesis set H_1 of $N = 4$ candidate goals from the initial observation. For each subsequent step t, EARL first applies PROPAGATE to carry forward the hypotheses from H_{t-1} into the new state s_t, preserving their semantic content. It then invokes UPDATEWEIGHTS, which evaluates each hypothesis $g_i \in H_t$ based on how well it explains the newly observed action a_t in the context of s_t, assigning qualitative likelihoods such as *very likely*, *likely*, *uncertain*, or *unlikely*. These belief strengths guide future reasoning by increasing the weight of plausible hypotheses and downweighting those that are inconsistent with the user's behavior.

If the updated belief set H_t becomes dominated by low-likelihood or semantically overlapping hypotheses, the agent invokes RESAMPLE to restore diversity. This involves pruning particles that are consistently rated as *unlikely*, and duplicating or paraphrasing stronger candidates to maintain a balanced distribution. Through this iterative process, EARL incrementally refines its understanding of the user's latent intent, enabling robust goal inference even when the true objective is revealed gradually across a multi-step interaction. Figure 2 illustrates this process: the agent begins with a diverse set of coarse-grained hypotheses (e.g., "apply color grading," "blur license plate," "crop car") and incrementally filters and reweights them based on observed user actions. As finer-grained cues emerge—such as selecting the license plate—the belief in irrelevant goals is downweighted, and the most consistent hypothesis (e.g., "blur license plate") is selected as the predicted intent.

At designated checkpoints (e.g., after observing 25%, 50%, or 75% of the full trajectory), EARL summarizes the current belief state using SUMMARIZEBELIEFTRACE, recording the highest-weighted hypothesis at that point as the predicted intent $\hat{g}$. This mechanism allows EARL to infer intents at multiple stages throughout the interaction, rather than deferring to a final decision at 100%. Belief summarization and intent reporting occur cumulatively, reflecting both newly observed context and the agent's evolving hypothesis distribution. All reasoning is guided by a structured system prompt that defines the agent's behavior and goal-tracking strategy, enabling consistent and generalizable inference through a single language model interface (see Appendix A2).

Throughout the remainder of this paper, we refer to this GUI-based, particle-filtering algorithm as **EARL** (*Early Action Reasoning for Latent Intent*)

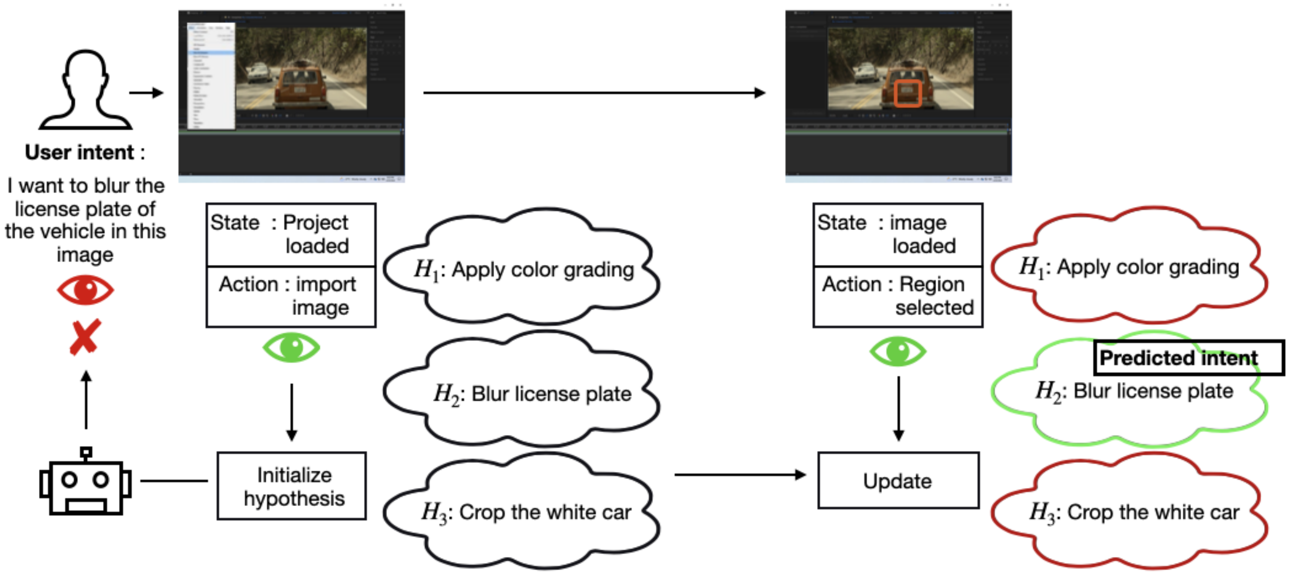

User intent :
I want to blur the license plate of the vehicle in this image
State : Project loaded
Action : import image
Initialize hypothesis
H_1: Apply color grading
H_2: Blur license plate
H_3: Crop the white car
State : image loaded
Action : Region selected
Update
H_1: Apply color grading
Predicted intent
H_2: Blur license plate
H_3: Crop the white car

Figure 2: A user opens the photoshop application on the system and imports an image of car. Based on this initial action and contextual cues (for e.g. the image of a car), the LLM initiates a chain of hypothesis that includes cropping the white car to blurring license plate to applying color grading. The user's further action of selecting the license plate of the car narrows the hypothesis down to more fine grained hypothesis suggestion of blurring the license plate matching the intent of the user.

3.2 Evaluation

We evaluate the accuracy of predicted user intents at different points in the interaction trajectory using a semantic goal entailment framework (see Appendix A4). Rather than performing exact text comparison, we assess how well the predicted intent semantically aligns with the annotated ground truth intent. This approach is motivated by the need to account for diverse phrasing, varying levels of specificity, and the compositional nature of user goals in GUI-based tasks. Our evaluation methodology is inspired by the goal alignment judgment setup proposed by Berkovitch et al. (Berkovitch et al., 2025).

Given a ground truth intent A and a predicted intent B, the model is tasked with determining whether fulfilling B would fully, partially, or not at all satisfy the requirements of A. To accomplish this, we use an LLM to perform pairwise comparison between A and B and classify the relationship into one of three categories: MATCH, PARTIAL MATCH, or NON MATCH.

A prediction is considered a MATCH if B expresses the same underlying goal as A, possibly using different language or added specificity, but still preserving all essential constraints and outcomes. For example, if A is "Search for the capital of Argentina" and B is "Search for the capital of Argentina using Google," then B satisfies A and is classified as a MATCH. In contrast, a PARTIAL MATCH indicates that the predicted goal is related to the ground truth but either omits required constraints or introduces new ones that are not implied by A. For instance, predicting "Change background to blue and add a gradient" for a ground truth "Change background to blue" qualifies as a PARTIAL MATCH due to the added gradient constraint. Lastly, a NON MATCH reflects a semantic misalignment, where the predicted intent either diverges functionally from the annotated goal or is too underspecified to support the same outcome.

Each predicted intent is compared to the gold intent using Gemini-1.5-Pro, which is prompted with structured instructions to return exactly one label among the three. The classification is then mapped to a numerical score, 1.0 for MATCH, 0.5 for PARTIAL MATCH, and 0.0 for NON MATCH. This scoring method allows for nuanced assessment of semantic similarity, particularly in settings where user intent unfolds gradually over interaction sequences. By incorporating goal entailment rather than exact matching, we enable fair and flexible comparison across models that generate free-form intent predictions.

4 Experiment Setup

To assess the effectiveness of early intent prediction in GUI-based environments, we conduct experiments using two advanced large language models: **Gemini-1.5-Pro** and **GPT-4o**. Both models are prompted to infer the user's goal at intermediate points in the interaction sequence, specifically after observing 25%, 50%, and 75% of the full GUI trajectory. These predictions are evaluated against ground truth intents using, **Gemini-1.5-Pro**, configured as a semantic

Algorithm 1 EARL: GUI-Based Intent Inference via Goal Hypothesis Propagation

```
Input: GUI trajectory $E = \{(s_1, a_1), \ldots, (s_T, a_T)\}$; checkpoint indices $C \subseteq \{1, \ldots, T\}$
Output: Predicted intents $\{\hat{g}_p\}$ at checkpoints; belief traces $\{\tilde{\tau}_p\}$
for $t = 1$ to $T$ do
    $(s_t, a_t) \leftarrow$ screen and action at step $t$
    if $t = 1$ then
        $H_t \leftarrow \textsc{Initialize}(s_t, a_t, N)$
    else
        $H_t \leftarrow \textsc{Propagate}(H_{t-1}, s_t, a_t)$
        $H_t \leftarrow \textsc{UpdateWeights}(H_t, a_t)$
        if $H_t$ is low-confidence or semantically redundant then
            $H_t \leftarrow \textsc{Resample}(H_t)$
        end if
    end if
    if $t \in C$ then
        $\tilde{\tau}_p \leftarrow \textsc{SummarizeBeliefTrace}(\{H_1, \ldots, H_t\})$
        $\hat{g}_p \leftarrow$ highest-weighted hypothesis in $\tilde{\tau}_p$
        Output: $(t, \hat{g}_p)$
    end if
end for
Return all $\{\hat{g}_p\}$ and $\{\tilde{\tau}_p\}$
```

evaluator. The evaluator model is tasked with determining the degree to which each predicted intent aligns with the true user goal, using a structured entailment prompt.

Our experiments span three benchmark datasets. **Mind2Web** (Deng et al., 2023b) is a web-based task dataset covering diverse domains such as booking, navigation, and search. **AiTz** (Zhang et al., 2024b) captures Android app interaction trajectories and includes a variety of task types across domains like app installation, Google apps, web shopping, and general settings. **VideoGUI** (Lin et al., 2024) is a curated collection of GUI tutorials drawn from creative software, where users perform visually grounded operations such as editing, animation, and object manipulation. Each dataset is evaluated independently. We specifically selected these datasets because they satisfy the structural requirements for intent inference: each task provides an explicit goal paired with a sequence of user interactions, where each step includes both the observed screen state (as a screenshot) and the corresponding action taken (see Appendix A1). This format enables step-wise intent modeling grounded in visual context and behavioral data, which is essential for evaluating partial intent prediction and belief evolution over time. Only tasks with a minimum of six steps in their interaction trajectory are included to ensure that sufficient context is available for intent inference at the earliest 25% checkpoint. This filtering results in a total of **865 valid tasks**, distributed across datasets as follows: **439 tasks** from AiTz (aggregated across four subdomains), **345 tasks** from Mind2Web, and **81 tasks** from VideoGUI.

As a baseline, we use a standard chain-of-thought (CoT) prompting strategy (see Appendix A3) in which the model is shown a partial sequence of screenshots and actions and asked to generate the user's intent in natural language. Unlike our proposed EARL method, CoT does not maintain or refine goal hypotheses over time, and instead treats each prediction independently based on the local prompt context. To ensure robustness and account for stochasticity in LLM behavior, we evaluate both models (GPT-4o and Gemini-1.5-Pro) using both CoT and EARL prompting strategies. Each model–dataset–method combination is run across three independent trials. We set the decoding temperature of the language model to zero for all prediction and evaluation steps to ensure deterministic outputs across runs.

To quantify performance, we report two complementary evaluation metrics. The first is the **Perfect Match Rate**, which reflects the percentage of predictions at each checkpoint that are judged to be a complete semantic match with the ground truth intent. This evaluation is based on whether the predicted goal would fully satisfy the user's intended outcome, even if the phrasing differs. These results are summarized in Table 1, which provides a compact overview of high-confidence performance across datasets and checkpoints. The table allows for a direct comparison between the EARL model and the baseline prompting approach, highlighting the relative gains in perfect goal inference at early stages of the interaction.

The second evaluation metric, the **Weighted Mean Score**, offers a continuous and interpretable measure of a model's average semantic alignment across a set of predictions. While the Perfect Match Rate only captures the fraction of predictions that are exactly correct, this metric considers the entire distribution of predictions,including those that are partially or completely incorrect, by assigning different weights to different levels of alignment.

To compute this score, each predicted intent is first labeled as a Match, Partial Match, or Non Match, depending on how well it satisfies the ground truth. We then count the number of predictions in each of these three categories. The final score is calculated by multiplying these counts with predefined weights (1.0 for Match, 0.5 for Partial Match, and 0.0 for Non Match), summing the results, and dividing by the total number of predictions. This can be formally expressed as:

$$\text{Mean Score} = \frac{w_m \cdot M + w_p \cdot P + w_n \cdot N}{M + P + N}$$

where M, P, and N are the number of predictions labeled as Match, Partial Match, and Non Match, respectively. The weights are defined as $w_m = 1.0$, $w_p = 0.5$, and $w_n = 0.0$.

This produces a single scalar value between 0.0 and 1.0, where higher values indicate stronger semantic accuracy overall. By rewarding partially correct predictions while penalizing incorrect ones, the Weighted Mean Score provides a more comprehensive view of model performance.

Dataset	LLM	Checkpoint	EARL	Baseline	Gain (%)
AiTz	GPT-4o	25%	**4.1**	3.04	**34.9%**
		50%	**17.01**	12.53	**35.7%**
		75%	**44.34**	34.4	**28.9%**
	Gemini	25%	**3.3**	2.85	**15.8%**
		50%	**12.98**	11.5	**12.9%**
		75%	**37.02**	32.35	**14.4%**
Mind2Web	GPT-4o	25%	1.74	**2.32**	-25.0%
		50%	**8.89**	8.21	**8.3%**
		75%	**24.15**	21.45	**12.6%**
	Gemini	25%	**1.19**	1.16	**2.6%**
		50%	**6.38**	4.93	**29.4%**
		75%	**17.68**	16.23	**8.9%**
VideoGUI	GPT-4o	25%	**14.4**	7.82	**84.1%**
		50%	**22.22**	16.87	**31.7%**
		75%	**35.39**	32.1	**10.2%**
	Gemini	25%	**11.11**	6.17	**80.1%**
		50%	**15.43**	9.88	**56.2%**
		75%	**16.67**	14.81	**12.5%**

Table 1: Perfect match rate (%) at different trajectory checkpoints for each dataset and backbone (GPT-4o and Gemini). EARL outperforms the CoT baseline across most datasets.

5 Results and Discussion

5.1 Overview of Quantitative Trends

Table 1 quantifies the Perfect Match Rate (%) for each model–dataset combination at 25%, 50%, and 75% checkpoints. EARL yields significant gains over the CoT baseline—up to **84.1%** on VideoGUI (GPT-4o, 25%) and **56.2%** on VideoGUI (Gemini, 50%), with consistent improvements observed in most configurations. Table 2 further analyzes semantic alignment by reporting the percentage of Match, Partial Match, and Non Match predictions, as well as the resulting weighted mean score. EARL consistently achieves higher mean scores across checkpoints, indicating stronger overall alignment with user goals, even when perfect matches are not obtained.

Figure 3 presents a comparative view of model performance across all datasets and checkpoints. Figure 3 shows the Match (%) rate across the three datasets AiTz, Mind2Web, and VideoGUI—for both EARL and the Chain-of-Thought (CoT) baseline, using two LLMs: GPT-4o and Gemini. EARL consistently achieves higher match accuracy, with especially large margins at earlier checkpoints (25% and 50%). Figure 3 illustrates the trend of the weighted mean score, highlighting EARL's consistent semantic advantage across models and datasets.

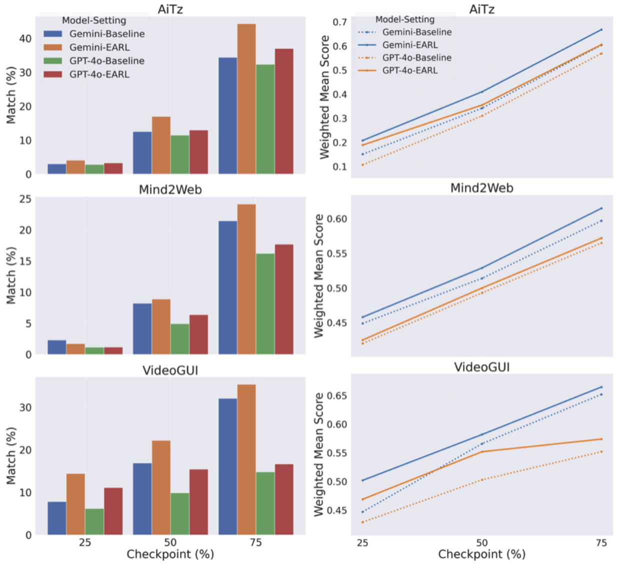


Figure 3: Comparison of the performance of EARL with the CoT baseline. (a) shows the exact match rate for CoT and EARL across datasets and checkpoints. Across all three checkpoints and three datasets, both the LLMs (GPT-4o and Gemini-1.5-Pro) with EARL show better performance in deciphering user intent from partial trajectories compared to the CoT baseline (b) highlights EARL's consistent gain in semantic alignment (weighted mean score) over CoT across all three checkpoints and datasets for both LLMs. The inference time algorithm EARL is shown in solid lines whereas the CoT baseline is shown in dotted lines.

5.2 Comparative Analysis of EARL and Chain-of-Thought Baseline

Although both EARL and the Chain-of-Thought (CoT) baseline operate over sequences of GUI screenshots and user actions, they differ significantly in how they perform inference under partial observation. CoT treats each checkpoint as an isolated input window: it consumes the visible trajectory up to that point and produces a single-shot intent prediction without modeling uncertainty or maintaining evolving beliefs. EARL, in contrast, maintains a set of competing intent hypotheses that are updated incrementally based on the user's observed behavior. This distinction becomes especially important in early-stage prediction, where the user's goal is not yet fully revealed and must be inferred from subtle or indirect cues.

One observed example involves a creative editing workflow where the user first imports a background image, overlays text, and selects a mask layer. At this early point (25%), the CoT baseline often predicts narrow intents such as “add text” or “mask the text layer,” based solely on visible actions. EARL, however, begins with a diverse hypothesis set including goals like “mask background,” “create text reveal effect,” or “prepare composite layer,” and evaluates each based on how well it fits the current interface state. As the user continues by adding a blur and enabling track matte mode (50%), CoT tends to focus on local transformations like “apply blur effect,” while EARL reweights its belief distribution, downranking simpler operations and strengthening the hypothesis for a “text reveal” effect. By 75%, when glow is applied and the preview is rendered, EARL converges on a temporally grounded goal like “create glowing text reveal using track matte,” while CoT remains focused on fragmented predictions without integrating the broader intent.

Table 2: Semantic alignment results across datasets and models. For each checkpoint percentage (CP), we show the percentages of Non-Matches (NM), Partial-Matches (PM), and perfect Matches (M), along with the weighted mean score. Bold scores indicate where EARL outperforms Baseline.

Dataset	Model	CP (%)	Method	NM (%)	PM (%)	M (%)	Mean
AiTz	GPT-4o	25	Baseline	81.44	15.72	2.85	0.107
			EARL	65.49	31.21	3.30	**0.189**
		50	Baseline	49.43	39.07	11.50	0.310
			EARL	41.91	45.10	12.98	**0.355**
		75	Baseline	18.56	49.09	32.35	0.569
			EARL	15.72	47.27	37.02	**0.606**
	Gemini	25	Baseline	72.89	24.07	3.04	0.151
			EARL	62.57	33.33	4.10	**0.208**
		50	Baseline	44.12	43.36	12.53	0.342
			EARL	35.00	47.99	17.01	**0.410**
		75	Baseline	13.59	52.01	34.40	0.604
			EARL	10.78	44.87	44.34	**0.668**
Mind2Web	GPT-4o	25	Baseline	16.23	81.06	1.16	0.420
			EARL	16.60	82.61	1.19	**0.425**
		50	Baseline	6.38	88.70	4.93	0.493
			EARL	6.67	86.96	6.38	**0.500**
		75	Baseline	3.19	80.58	16.23	0.565
			EARL	3.19	79.13	17.68	**0.572**
	Gemini	25	Baseline	12.46	85.22	2.32	0.449
			EARL	10.05	88.21	1.74	**0.458**
		50	Baseline	5.31	86.47	8.21	0.514
			EARL	3.09	88.02	8.89	**0.529**
		75	Baseline	2.13	76.43	21.45	0.597
			EARL	1.16	74.69	24.15	**0.615**
VideoGUI	GPT-4o	25	Baseline	20.37	73.46	6.17	0.429
			EARL	17.28	71.60	11.11	**0.469**
		50	Baseline	9.26	80.86	9.88	0.503
			EARL	4.94	79.63	15.43	**0.552**
		75	Baseline	4.32	80.64	14.81	0.552
			EARL	1.85	81.48	16.67	**0.574**
	Gemini	25	Baseline	18.52	73.66	7.82	0.447
			EARL	13.99	71.60	14.40	**0.502**
		50	Baseline	3.70	79.42	16.87	0.566
			EARL	5.76	72.02	22.22	**0.582**
		75	Baseline	1.65	66.26	32.10	0.652
			EARL	2.47	62.14	35.39	**0.665**

A similar trend appears in mobile device settings tasks. At 25%, after navigating to display settings and toggling adaptive brightness, CoT typically outputs intents like “adjust brightness,” closely reflecting the immediate interface state. EARL, meanwhile, retains a broader hypothesis set that includes “enable adaptive brightness,” “manually reduce screen brightness,” and “optimize power consumption.” As the user proceeds to open battery settings and explore power-saving features (50%), CoT updates toward “view battery settings” or “turn on battery saver,” still reflecting only the latest action. EARL instead adjusts belief toward a goal like “optimize power using adaptive settings,” connecting earlier and current actions. By 75%, EARL strengthens its belief in higher-level goals such as “reduce power consumption through adaptive features,” whereas CoT continues to predict isolated steps, missing the evolving underlying intent.

This contrast illustrates a key behavioral distinction between the two approaches: while CoT generates predictions grounded primarily in the most recent observed actions, EARL incrementally integrates cues from the full interaction history to model latent user goals with greater semantic depth. Its evolving belief refinement enables more coherent interpretation of partially observed tasks, particularly in settings where intent must be inferred before all goal-revealing actions have been completed.

5.3 Limitations Due to Ground Truth Specificity and Goal Timing

While EARL consistently outperforms the CoT baseline across benchmarks, the magnitude of improvement is not uniformly large across all datasets. A key limiting factor lies in the nature of the annotated ground truth intents, particularly in how specific or temporally delayed they are.

In the AiTz dataset, some ground truth intents are phrased as highly specific instructions rather than generalized, latent goals. For example, an intent like “login with username ’john’ and password ’xyz123’ ” encodes a literal procedural step tied directly to interface elements, rather than the broader goal of “log in” or “access account.” In such cases, both EARL and CoT are constrained: since neither model is instructed to reproduce literal field values, they often receive a PARTIAL MATCH or NON MATCH even when the inferred goal is behaviorally appropriate. These over-specified ground truths limit the space for abstraction and reduce EARL’s advantage in progressive inference.

In contrast, the Mind2Web dataset introduces a different challenge. Many of its tasks involve complex, multi-part goals that are only fully specified at the end of the trajectory. For instance, a task might ask the user to “find a permanent job in Logistics within 20 miles of New York, zip 11005, in the middle-income range for a high school diploma holder.” Early-stage actions in such tasks, like keyword search, location filter, or salary sorting, do not yet reflect the complete structure of the target goal. As a result, EARL may form plausible partial hypotheses like “search for jobs near New York,” but its predictions at early checkpoints cannot match the full specificity required by the annotated intent. This constrains its performance gains over CoT, which often makes similarly partial predictions in such contexts.

These characteristics overly literal goal definitions and delayed intent anchoring pose fundamental challenges to early intent modeling. They limit the measurable benefit of belief refinement strategies like those employed by EARL, particularly when the ground truth formulation does not reward abstraction or progressive inference. In contrast, datasets such as AiTz (excluding its over-specified cases) and VideoGUI, where the user’s goal unfolds gradually through interaction, e.g., setting up a visual effect or completing a feature-rich workflow—better support EARL’s evolving hypothesis mechanism and allow it to demonstrate stronger alignment and earlier goal anticipation compared to the CoT baseline.

5.4 Theoretical and Practical Implications

EARL’s success in early intent recognition underscores the value of inverse planning frameworks for modeling latent user goals in GUI interactions. Central to this is the ”inversion problem” articulated by Mullainathan & Kleinberg (2023), who argue that AI systems must prioritize inferring mental states (e.g., goals, beliefs) over merely predicting actions. They demonstrate that systems focused solely on behavioral prediction risk optimizing for superficial proxies rather than the underlying intent, leading to brittle, context-blind solutions. For instance, an agent trained to predict ”mouse clicks” might learn to mimic common interaction patterns (e.g., frequent clicks on a ”submit” button) without understanding why the user clicked, resulting in failures when faced with novel goals or interface changes. EARL addresses this gap by explicitly reasoning about the latent goals that drive user actions, treating intent recognition as an inverse problem where partial observations are explained through probabilistic hypothesis refinement.

This approach bridges classical inverse planning (Baker et al., 2009), which models actions as rational means to achieve hidden goals, with modern LLM-based reasoning. While earlier ToM-inspired systems (Rabinowitz et al., 2018) focused on synthetic environments, EARL demonstrates how particle filtering and hypothesis rejuvenation can adapt these principles to noisy, open-world GUI tasks. For example, EARL’s ability to refine hypotheses like ”color grading” → ”blur license plate” in Figure 2 mirrors human belief updating during collaborative tasks, as observed in cognitive science (Baker et al., 2017).

Practically, EARL’s gains on VideoGUI (84.1% at 25% trajectory length) highlight the benefits of modeling intent incrementally. Proactive systems like EARL can reduce user effort in creative workflows (e.g., suggesting blur tools early in Figure 1), addressing a key limitation of post hoc summarization methods (Zhang et al., 2024a). However, challenges remain: overly specific ground truths (e.g., AiTz’s literal login instructions) constrain abstraction, echoing Mullainathan & Kleinberg (2023)’s warnings about goal misalignment between humans and algorithms.

Future work could integrate multimodal cues (e.g., gaze tracking (Lee et al., 2023)) to enrich hypothesis generation, while addressing ethical risks like unintended inference of sensitive goals. By prioritizing interpretable mental state reasoning over opaque action prediction, EARL advances the vision of GUI agents that collaborate with humans, rather than merely executing commands.

6 Conclusion

We introduce EARL (Early Action Reasoning for Latent Intent), a framework for proactive intent recognition in GUI tasks inspired by Theory of Mind (ToM). By integrating particle filtering with hypothesis re-sampling, EARL maintains a belief distribution over candidate goals, updating them incrementally as new actions are observed. Our experiments across three benchmarks (Mind2Web, AiTz, VideoGUI) demonstrate that EARL’s hypothesis-driven reasoning outperforms action-prediction paradigm CoT. This capability addresses the “inversion problem” in AI—systems must infer why users act, not just what they do, to avoid brittle, context-blind solutions.Looking ahead, EARL’s modular architecture invites extensions, such as integrating gaze or speech cues to enrich hypotheses, or fine-tuning LLMs for goal abstraction. Our work underscores that modeling mental states is not merely complementary to action prediction, it is foundational for human-AI partnership.

7 Impact statement

EARL enables proactive GUI agents through early intent inference, benefiting accessibility and complex workflows. While enhancing responsiveness, it raises privacy/autonomy concerns—we mitigate these via transparent operation and user correction mechanisms. Our dataset-agnostic approach reduces bias, though diverse user evaluation remains critical. By focusing on interpretable goal inference (not action mimicry), EARL advances collaborative, human-centered AI. By focusing on interpretable goal inference (not action mimicry), EARL advances collaborative, human-centered AI. This work paves the way for more equitable digital interfaces that adapt to users’ cognitive needs rather than requiring adaptation to rigid systems, particularly perticularly empowering technology-novice users through anticipatory support.

References

Ahmed, M. Leveraging large language models for latent intention recognition and next action prediction. Master's thesis, 2024.

Baker, C. L., Saxe, R., and Tenenbaum, J. B. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

Baker, C. L., Jara-Ettinger, J., Saxe, R., and Tenenbaum, J. B. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):0064, 2017.

Berkovitch, O., Caduri, S., Kahlon, N., Efros, A., Caciularu, A., and Dagan, I. Identifying user goals from ui trajectories, 2025. URL https://arxiv.org/abs/2406.14314.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023a.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2web: Towards a generalist agent for the web, 2023b. URL https://arxiv.org/abs/2306.06070.

Kim, H., Sclar, M., Zhi-Xuan, T., Ying, L., Levine, S., Liu, Y., Tenenbaum, J. B., and Choi, Y. Hypothesis-driven theory-of-mind reasoning for large language models. *arXiv preprint arXiv:2502.11881*, 2025.

Lee, M., Singh, G., and Awadalla, H. Eyetom: Gaze-augmented theory of mind inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12345–12355, 2023.

Lin, K. Q., Li, L., Gao, D., WU, Q., Yan, M., Yang, Z., Wang, L., and Shou, M. Z. Videogui: A benchmark for gui automation from instructional videos, 2024. URL https://arxiv.org/abs/2406.10227.

Liu, E. Z., Guu, K., Pasupat, P., Shi, T., and Liang, P. Reinforcement learning on web interfaces using workflow-guided exploration. *arXiv preprint arXiv:1802.08802*, 2018.

Mullainathan, S. and Kleinberg, J. The inversion problem: Why algorithms should infer mental states and not just predict behavior. *Penn Institute for Computational Science*, 2023. URL https://upenn.edu/papers/mullainathan-kleinberg-inversion-problem.pdf. [Hypothetical paper for illustrative purposes].

Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., and Botvinick, M. Machine theory of mind. In *International conference on machine learning*, pp. 4218–4227. PMLR, 2018.

Wang, S., Liu, W., Chen, J., Zhou, Y., Gan, W., Zeng, X., Che, Y., Yu, S., Hao, X., Shao, K., et al. Gui agents with foundation models: A comprehensive survey. *arXiv preprint arXiv:2411.04890*, 2024.

Wen, H., Wang, H., Liu, J., and Li, Y. Droidbot-gpt: Gpt-powered ui automation for android. *arXiv preprint arXiv:2304.07061*, 2023.

Wen, H., Li, Y., Liu, G., Zhao, S., Yu, T., Li, T. J.-J., Jiang, S., Liu, Y., Zhang, Y., and Liu, Y. Autodroid: Llm-powered task automation in android. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pp. 543–557, 2024.

Zhang, G., Ahmed, M., Hu, Z., and Bulling, A. Summact: Uncovering user intentions through interactive behaviour summarisation. *arXiv preprint arXiv:2410.08356*, 2024a.

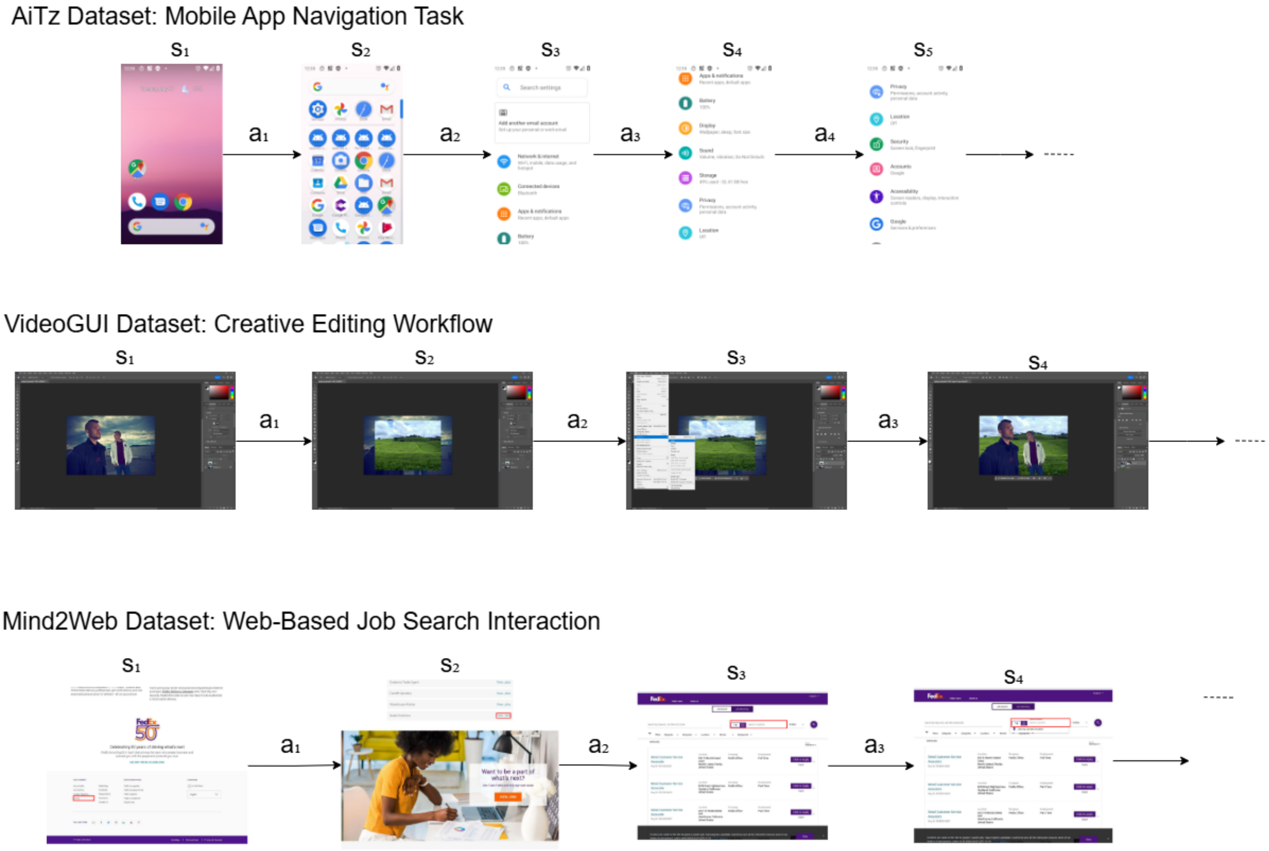
Zhang, J., Wu, J., Teng, Y., Liao, M., Xu, N., Xiao, X., Wei, Z., and Tang, D. Android in the zoo: Chain-of-action-thought for gui agents, 2024b. URL https://arxiv.org/abs/2403.02713.

Zhang, S., Zhang, Z., Chen, K., Ma, X., Yang, M., Zhao, T., and Zhang, M. Dynamic planning for llm-based graphical user interface automation. *arXiv preprint arXiv:2410.00467*, 2024c.

Zheng, L., Wang, R., Wang, X., and An, B. Synapse: Trajectory-as-exemplar prompting with memory for computer control. *arXiv preprint arXiv:2306.07863*, 2023.

8 Appendix

A1 Example Trajectories Across Datasets


AiTz Dataset: Mobile App Navigation Task

VideoGUI Dataset: Creative Editing Workflow

Mind2Web Dataset: Web-Based Job Search Interaction

Figure 4: Examples of early interaction trajectories from each dataset. Top: a mobile settings navigation task from the AiTz dataset. Middle: a creative editing sequence from the VideoGUI dataset. Bottom: a web-based job search scenario from the Mind2Web dataset. Each example highlights the sequence of GUI states and user actions, with dotted arrows denoting continuation beyond the shown steps.

A2 EARL Prompt Used for Intent Prediction

Prompt Template: EARL for Intent Prediction

Given a sequence of GUI steps, infer what the user is trying to accomplish based on what they see and do.
You will be given a sequence of GUI steps. Each step includes:

- A screenshot showing the interface before the action is taken
- A natural language description of the action performed (e.g., ”click Gmail icon”, ”type ’resume’ in search bar”)

Track a set of evolving goal hypotheses over time using the following steps:

`<thinking>`
Follow this process internally. Do not include or print anything from this section in your output.
EARL Algorithm:

1. Initialize (only at the first step): Start with 4 diverse hypotheses about the user’s goal based on the initial screen and action.
2. Propogate: For all later steps, carry forward the existing hypotheses and refine them using the latest screen and action.
3. Update Weights: Assess how well each hypothesis explains the user’s current action and assign a qualitative likelihood (e.g., very likely, likely, uncertain, unlikely, very unlikely).
4. Resample: If many hypotheses have very low likelihoods, remove them and duplicate stronger ones to maintain a set of 4 plausible alternatives.
5. Rejuvenate: If the hypotheses are too semantically similar, paraphrase or diversify the top ones.
    - Drop hypotheses that are low-weight or nearly identical in meaning
    - Rephrase high-weight but redundant ones to maintain diversity in possible interpretations

`</thinking>`

6. Hypotheses Summarization (after processing all steps):
Once the full sequence has been processed and belief updates are complete, summarize the user’s belief evolution using the following format:

`<thought_trace_summary>`
context: [What the user now sees on screen]
action: [What they did and why]
believes: [What they currently want to achieve, based on behavior so far]
`</thought_trace_summary>`

Only summarize meaningful turning points where the visible state or belief clearly shifts. This summarization is not done during the loop, but only after all steps have been processed.

Final Intent Output:
Based on the complete thought trace, infer what motivated the entire sequence — the goal that explains the user’s behavior across all screens and actions.
Use this format:

```
<Intent> [User’s intent/goal in present tense]</Intent>
```

Note: Only output the `<thought_trace_summary>` and `<Intent>` tags. Do not include the `<thinking>` section or any explanatory text.

A3 Baseline Prompt Using Chain-of-Thought

Prompt Template: Chain-of-Thought Baseline

You will receive a series of screenshots and actions representing a user's interactions on a website or application. Each item contains:

- An action performed by the user
- A screenshot depicting the state of the website before that action

Think step by step about what the user is doing and why.

Your output must follow this structure:

step-by-step description:
Provide a numbered list where each entry corresponds directly to a specific screenshot, detailing the user's actions and the visual context provided by the screenshots.

concise task:
Summarize the user's overall goal that motivated the sequence of actions based on the step-by-step description.

Use the following format to express the inferred intent:

```
<Intent> [User’s intent/goal in present tense] </Intent>
```

A4 Prompt for Comparing Predicted and True Goals

Prompt Template: Evaluate Predicted Goal (B) Against True Goal (A)

You are an expert evaluator of goal alignment in GUI-based tasks.
Your job is to assess how well two user goals align with each other in terms of their intended outcome, specificity, and completeness. Both goals describe what a user wants to achieve through interaction with a graphical interface.
A = {a}
B = {b}

CHOOSE ONE RELATION

MATCH

- B expresses the same actionable goal as A
- B includes all essential constraints, targets, or outcomes that A requires
- Even if B is phrased differently or more specific, completing B would fully satisfy A
- If A is more general and B is more specific, but performing A would still accomplish everything that B requires — this counts as a MATCH
- Example:
  A: ”Search for the capital of Argentina”
  B: ”Search for the capital of Argentina using Google”
  → MATCH: B adds specificity, but A still covers all of B

PARTIAL MATCH

- B reflects the same high-level intent as A (e.g., animate a shape, enable a setting, search for information)
- But either:
  - B omits one or more meaningful constraints that A specifies (e.g., a specific object, color, or location)
  - Or B adds extra constraints not present in A, such that performing B might not guarantee A is satisfied
- Example:
  A: ”Add a title to the video”
  B: ”Add a title to the video using a red font and position it at the top-left corner”
  → PARTIAL MATCH: B adds visual and positional constraints (red font, top-left) not mentioned in A — performing A does not guarantee these requirements are satisfied

NON MATCH

- B expresses a different goal or intent than A
- Or B is too vague, too incomplete, or completely unrelated to satisfy A in any meaningful way

RESPONSE FORMAT

Return exactly one word wrapped in an <output> tag:

```
<output>MATCH</output>
<output>PARTIAL MATCH</output>
<output>NON MATCH</output>
```